DIWAKAR EDUCATION HUB

Database Management Systems Unit – 4

As per updated syllabus

DIWAKAR EDUCATION HUB

2020

THE LEARN WITH EXPERTIES

Database Management System

Database Management System or DBMS in short refers to the technology of storing and retrieving usersí data with utmost efficiency along with appropriate security measures.

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

Database Management System or DBMS in short refers to the technology of storing and retrieving usersí data with utmost efficiency along with appropriate security measures.

Characteristics

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics −

- **Real-world entity** − A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- **Relation-based tables** − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.

DIWAKAR EDUCATION HUB

- **Isolation of data and application** − A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.
- **Less redundancy** − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.
- **Consistency** − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- **Query Language** − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- **ACID Properties** − DBMS follows the concepts of **A**tomicity, **C**onsistency, **I**solation, and **D**urability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.
- **Multiuser and Concurrent Access** − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.
- **Multiple views** − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a

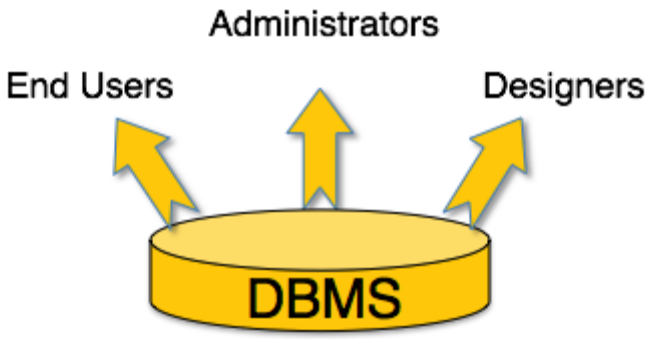
DIWAKAR EDUCATION HUB

person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.

- **Security** − Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

Users

A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows −



- **Administrators** − Administrators maintain the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators

DIWAK EDUCATION HUB

also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.

- **Designers** − Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.
- **End Users** − End users are those who actually reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

Applications of DBMS

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information. Following are the important characteristics and applications of DBMS.

- **ACID Properties** − DBMS follows the concepts of **A**tomicity, **C**onsistency, **I**solation, and **D**urability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.
- **Multiuser and Concurrent Access** − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.

DIWAKAR EDUCATION HUB

- **Multiple views** − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.

- **Security** − Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

DBMS - Architecture

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. An n-tier architecture divides the whole system into related but independent **n** modules, which can be independently modified, altered, changed, or replaced.

In 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it. Any changes done here will directly be done on the DBMS itself. It does not provide handy tools for end-users. Database designers and programmers normally prefer to use single-tier architecture.

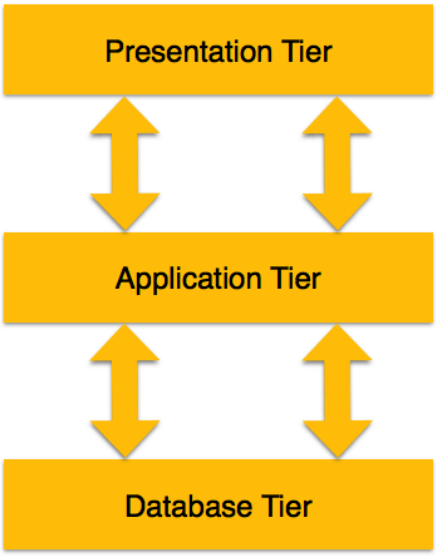
If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS by means of an application. Here the application tier is

DIWAKAR EDUCATION HUB

entirely independent of the database in terms of operation, design, and programming.

3-tier Architecture

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.



- **Database (Data) Tier** – At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.
- **Application (Middle) Tier** – At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.
- **User (Presentation) Tier** – End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this

DIWAKAR EDUCATION HUB

layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

Multiple-tier database architecture is highly modifiable, as almost all its components are independent and can be changed independently.

Data Models

Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.

The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific, hence they were prone to introduce lots of duplication and update anomalies.

Entity-Relationship Model

Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

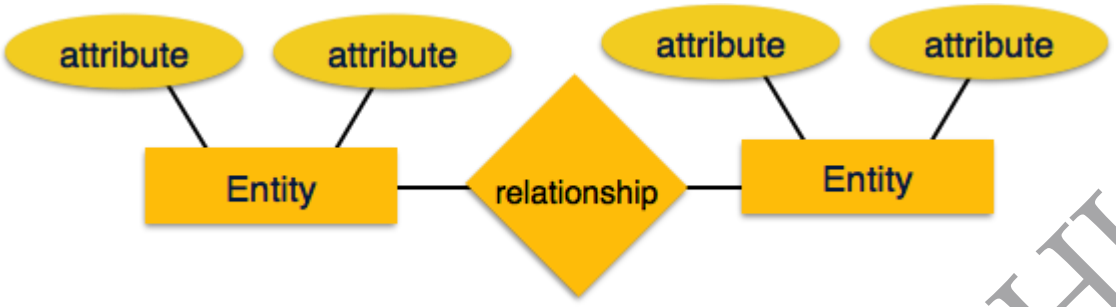
ER Model is best used for the conceptual design of a database.

ER Model is based on –

- **Entities** and their *attributes.*
- **Relationships** among entities.

These concepts are explained below.

DIWAKAR EDUCATION HUB



- **Entity** – An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- **Relationship** – The logical association among entities is called ***relationship***. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

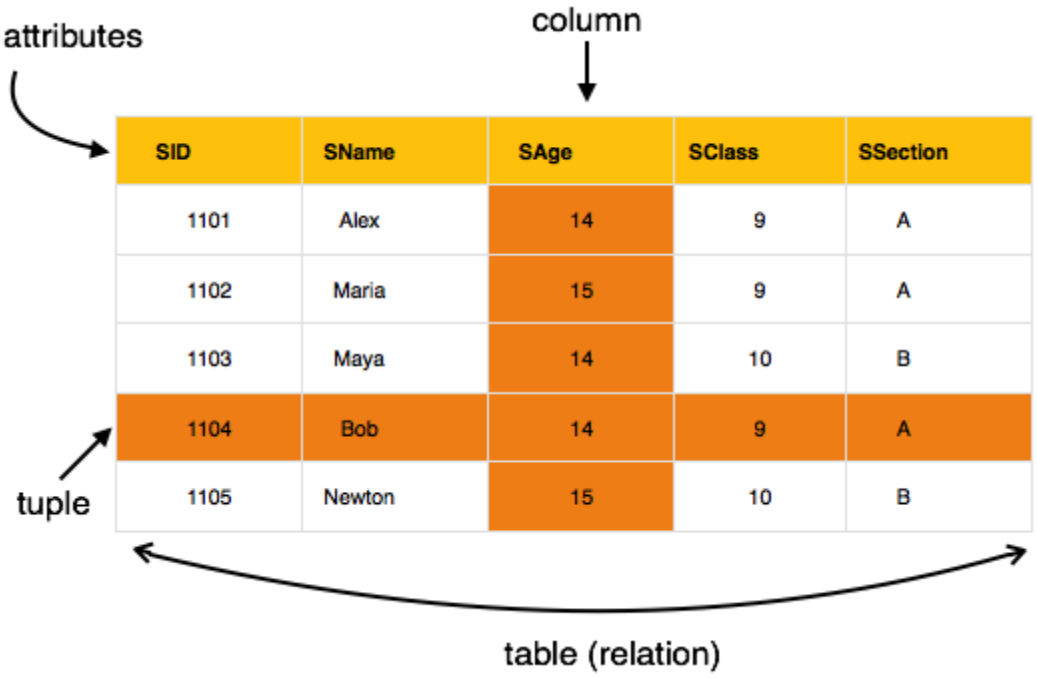
Mapping cardinalities –

- one to one
- one to many
- many to one
- many to many

Relational Model

The most popular data model in DBMS is the Relational Model. It is more scientific a model than others. This model is based on first-order predicate logic and defines a table as an **n-ary relation**.

DIWAKAR EDUCATION HUB



The main highlights of this model are –

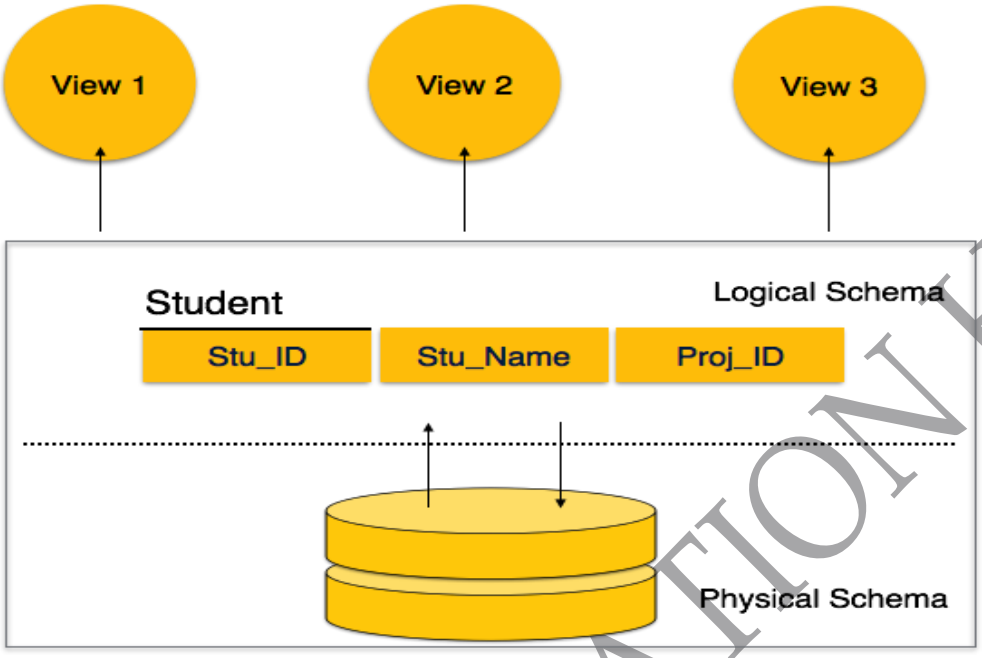
- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from a same domain.

Data Schemas

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

DIWAKAR EDUCATION HUB



A database schema can be divided broadly into two categories –

- **Physical Database Schema** – This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- **Logical Database Schema** – This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Database Instance

It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.

A database instance is a state of operational database with data at any given time. It contains a snapshot of the database. Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state, by diligently

DIWAKAR EDUCATION HUB

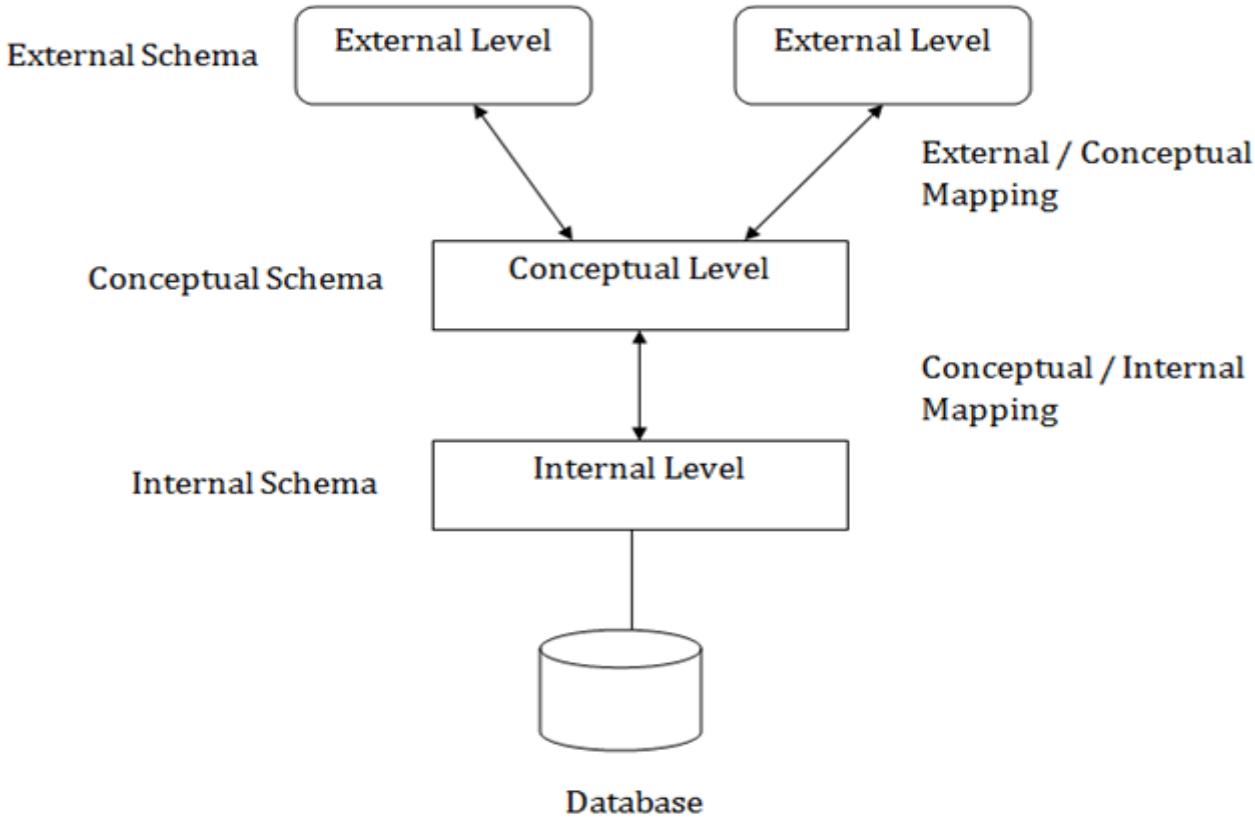
following all the validations, constraints, and conditions that the database designers have imposed.

Three schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:

DIWAKAR EDUCATION HUB



In the above diagram:

- It shows the DBMS architecture.
- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.
- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

1. Internal Level

DIWAKAR EDUCATION

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

2. Conceptual Level

- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

3. External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

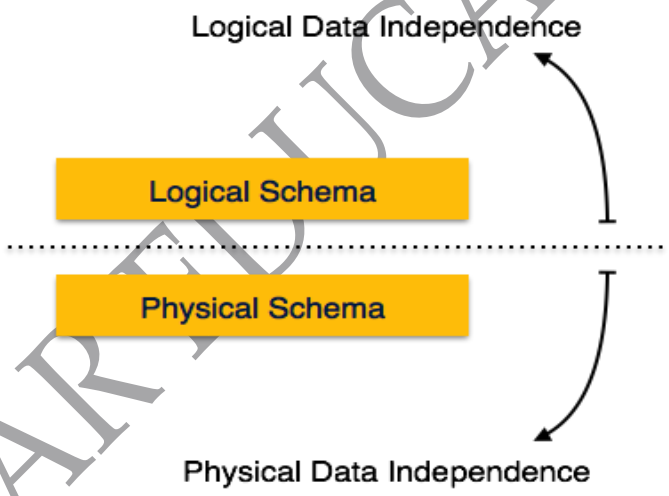
DIWAKAR EDUCATION HUB

Data Independence

If a database system is not multi-layered, then it becomes difficult to make any changes in the database system. Database systems are designed in multi-layers as we learnt earlier.

Data Independence

A database system normally contains a lot of data in addition to users' data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.



Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.

Logical Data Independence

Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.

DIWAKAR EDUCATION HUB

Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

Physical Data Independence

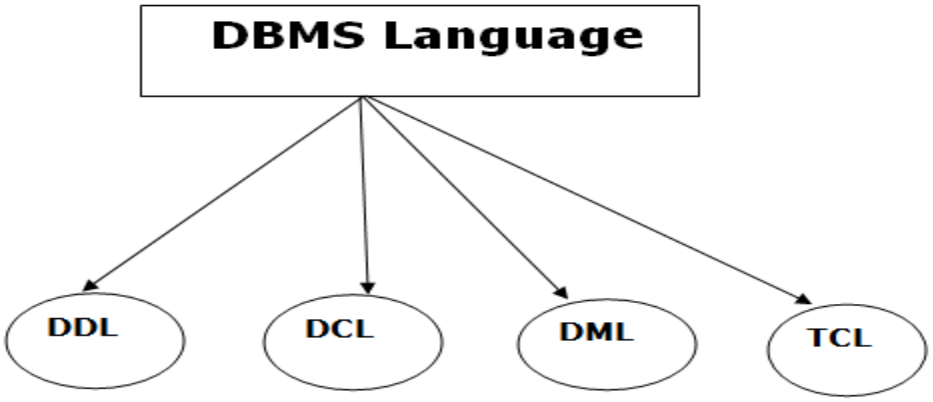
All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.

For example, in case we want to change or upgrade the storage system itself – suppose we want to replace hard-disks with SSD – it should not have any impact on the logical data or schemas.

Database Language

- A DBMS has appropriate languages and interfaces to express database queries and updates.
- Database languages can be used to read, store and update the data in the database.

Types of Database Language



DIW REDUCATION HUB

1. Data Definition Language

- **DDL** stands for **D**ata **D**efinition **L**anguage. It is used to define database structure or pattern.
- It is used to create schema, tables, indexes, constraints, etc. in the database.
- Using the DDL statements, you can create the skeleton of the database.
- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Some tasks that come under DDL:

- **Create:** It is used to create objects in the database.
- **Alter:** It is used to alter the structure of the database.
- **Drop:** It is used to delete objects from the database.
- **Truncate:** It is used to remove all records from a table.
- **Rename:** It is used to rename an object.
- **Comment:** It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

2. Data Manipulation Language

DML stands for **D**ata **M**anipulation **L**anguage. It is used for accessing and manipulating data in a database. It handles user requests.

Some tasks that come under DML:

- **Select:** It is used to retrieve data from a database.
- **Insert:** It is used to insert data into a table.
- **Update:** It is used to update existing data within a table.
- **Delete:** It is used to delete all records from a table.
- **Merge:** It performs UPSERT operation, i.e., insert or update operations.

DIWAKAR EDUCATION HUB

- **Call:** It is used to call a structured query language or a Java subprogram.
- **Explain Plan:** It has the parameter of explaining data.
- **Lock Table:** It controls concurrency.

3. Data Control Language

- **DCL** stands for **D**ata **C**ontrol **L**anguage. It is used to retrieve the stored or saved data.
- The DCL execution is transactional. It also has rollback parameters.

(But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Some tasks that come under DCL:

- **Grant:** It is used to give user access privileges to a database.
- **Revoke:** It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:

CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

4. Transaction Control Language

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

Some tasks that come under TCL:

- **Commit:** It is used to save the transaction on the database.
- **Rollback:** It is used to restore the database to original since the last Commit.

DBMS Interface

A database management system (DBMS) interface is a user interface which allows for the ability to input queries to a database without using the query language

DIWAKAR EDUCATION HUB

itself. A DBMS interface could be a web client, a local client that runs on a desktop computer, or even a mobile app.

A database management system stores data and responds to queries using a query language, such as SQL. A DBMS interface provides a way to query data without having to use the query language, which can be complicated.

The typical way to do this is to create some kind of form that shows what kinds of queries users can make. Web-based forms are increasingly common with the popularity of MySQL, but the traditional way to do it has been local desktop apps. It is also possible to create mobile applications. These interfaces provide a friendlier way of accessing data rather than just using the command line.

User-friendly interfaces provide by DBMS may include the following:

1. **Menu-Based Interfaces for Web Clients or Browsing –**
   These interfaces present the user with lists of options (called menus) that lead the user through the formation of a request. Basic advantage of using menus is that they removes the tension of remembering specific commands and syntax of any query language, rather than query is basically composed step by step by collecting or picking options from a menu that is basically shown by the system. Pull-down menus are a very popular technique in *Web based interfaces*. They are also often used in *browsing interface* which allow a user to look through the contents of a database in an exploratory and unstructured manner.
2. **Forms-Based Interfaces –**
   A forms-based interface displays a form to each user. Users can fill out all of the form entries to insert a new data, or they can fill out only certain entries, in which case the DBMS will redeem same type of data for other remaining entries. This type of forms are usually designed or created and programmed for the users that have no expertise in operating system. Many DBMSs have *forms specification languages* which are special languages that help specify such forms.

DIWAKAR EDUCATION HUB

Example: SQL* Forms is a form-based language that specifies queries using a form designed in conjunction with the relational database schema.b>

3. **Graphical User Interface –**
   A GUI typically displays a schema to the user in diagrammatic form.The user then can specify a query by manipulating the diagram. In many cases, GUI's utilize both menus and forms. Most GUIs use a pointing device such as mouse, to pick certain part of the displayed schema diagram.
4. **Natural language Interfaces –**
   These interfaces accept request written in English or some other language and attempt to understand them. A Natural language interface has its own schema, which is similar to the database conceptual schema as well as a dictionary of important words.

   The natural language interface refers to the words in its schema as well as to the set of standard words in a dictionary to interpret the request.If the interpretation is successful, the interface generates a high-level query corresponding to the natural language and submits it to the DBMS for processing, otherwise a dialogue is started with the user to clarify any provided condition or request. The main disadvantage with this is that the capabilities of this type of interfaces are not that much advance.

5. **Speech Input and Output –**
   There is an limited use of speech say it for a query or an answer to a question or being a result of a request it is becoming commonplace Applications with limited vocabularies such as inquiries for telephone directory, flight arrival/departure, and bank account information are allowed speech for input and output to enable ordinary folks to access this information.

   The Speech input is detected using a predefined words and used to set up the parameters that are supplied to the queries. For output, a similar conversion from text or numbers into speech take place.

DIWAKAR EDUCATION HUB

6. **Interfaces for DBA –**
   Most database system contains privileged commands that can be used only by the DBA's staff. These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, reorganizing the storage structures of a databases.

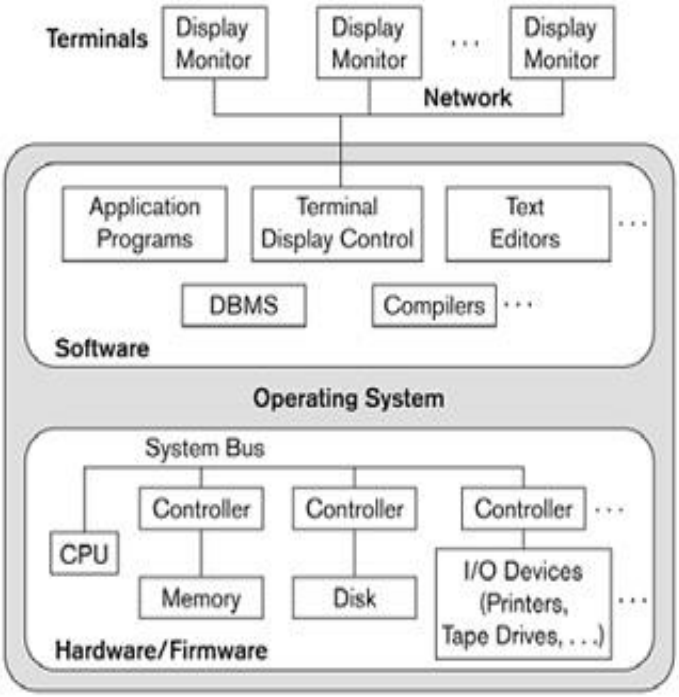
Centralized and Client-Server DBMS Architectures:

Centralized DBMS:

a) Merge everything into single system including- Hardware, DBMS software, application programs, and user interface processing software.

b) User can still connect by a remote terminal – but all processing is done at centralized site.

Physical Centralized Architecture:



A physical centralized architecture.

Architectures for DBMS have pursued trends similar to those generating computer system architectures. Earlier architectures utilized mainframes

DIWA EDUCATION HUB

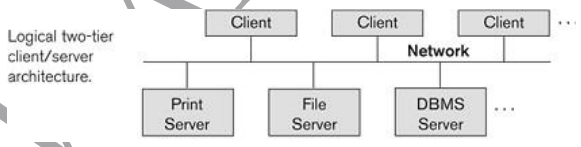
computers to provide the main processing for all system functions including user application programs as well as user interface programs as well all DBMS functionality. The reason was that the majority of users accessed such systems via computer terminals that did not have processing power and only provided display capabilities. Thus all processing was performed remotely on the computer system and only display information and controls were sent from the computer to the display terminals which were connected to central computer via a variety of types of communication networks.

As prices of hardware refused most users replaced their terminals with PCs and workstations. At first database systems utilized these computers similarly to how they have used is play terminals so that DBMS itself was still a Centralized DBMS in which all the DBMS functionality application program execution and user interface processing were carried out on one Machine.

Basic 2-tier Client-Server Architectures:

- Specialized Servers with Specialized functions
- Print server
- File server
- DBMS server
- Web server
- Email server
- Clients are able to access the specialized servers as needed

Logical two-tier client server architecture:



Clients:

- Offer appropriate interfaces through a client software module to access as well as utilize the various server resources.

DIVAKAR EDUCATION HUB

- Clients perhaps diskless machines or PCs or Workstations with disks with only the client software installed.
- Connected to the servers by means of some form of a network. (LAN- local area network, wireless network and so on.)

DBMS Server:

- Provides database query as well as transaction services to the clients
- Relational DBMS servers are habitually called query servers, SQL servers, or transaction servers
- Applications running on clients use an Application Program Interface (API) to access server databases via standard interface such as:

ODBC- Open Database Connectivity standard
JDBC- for Java programming access
Client and server should install appropriate client module and server module software for ODBC or JDBC

Two Tier Client-Server Architecture:

a) A client program may perhaps connect to several DBMSs sometimes called the data sources.

b) In general data sources are able to be files or other non-DBMS software that manages data. Other variations of clients are likely- example in some object DBMSs more functionality is transferred to clients including data dictionary functions, optimization as well as recovery across multiple servers etc.

Three Tier Client-Server Architecture:

a) Common for Web applications.

b) Intermediate Layer entitled Application Server or Web Server.

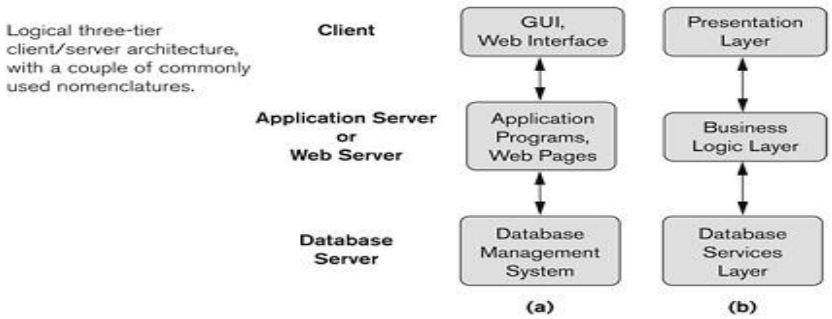
c) Stores the web connectivity software as well as the business logic part of the application used to access the corresponding data from the database server.

DIWAKAR EDUCATION HUB

d) Acts like a conduit for sending moderately processed data between the database server and the client.

e) Three-tier Architecture is able to Enhance Security:

- Database server merely accessible via middle tier.
- Clients can't directly access database server.



Classification of DBMS's:

• Based on the data model used
• Traditional- Network, Relational, Hierarchical.
• Emerging- Object-oriented and Object-relational.
• Other classifications
• Single-user (typically utilized with personal computers) v/s multi-user (most DBMSs).
• Centralized (utilizes a single computer with one database) v/s distributed (uses multiple computers and multiple databases)

Variations of Distributed DBMSs (DDBMSs):

- Homogeneous DDBMS
- Heterogeneous DDBMS
- Federated or Multi-database Systems
- Distributed Database Systems have at the present come to be known as client-server based database systems because
- They don't support a totally distributed environment however rather a set of database servers supporting a set of clients.

DIWAKAR EDUCATION HUB

Cost considerations for DBMSs:

- Cost Range- from free open-source systems to configurations costing millions of dollars
- Instances of free relational DBMSs- MySQL, PostgreSQL and others.

Data Modelling

Data modeling (data modelling) is the process of creating a data model for the data to be stored in a Database. This data model is a conceptual representation of Data objects, the associations between different data objects and the rules. Data modeling helps in the visual representation of data and enforces business rules, regulatory compliances, and government policies on the data. Data Models ensure consistency in naming conventions, default values, semantics, security while ensuring quality of the data.

Data Model

Data model is defined as an abstract model that organizes data description, data semantics and consistency constraints of data. Data model emphasizes on what data is needed and how it should be organized instead of what operations will be performed on data. Data Model is like architect's building plan which helps building conceptual models and set relationship between data items.

The two types of Data Models techniques are

1. Entity Relationship (E-R) Model
2. UML (Unified Modelling Language)

Why use Data Model?

The primary goal of using data model are:

DIWAKAR EDUCATION HUB

- Ensures that all data objects required by the database are accurately represented. Omission of data will lead to creation of faulty reports and produce incorrect results.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

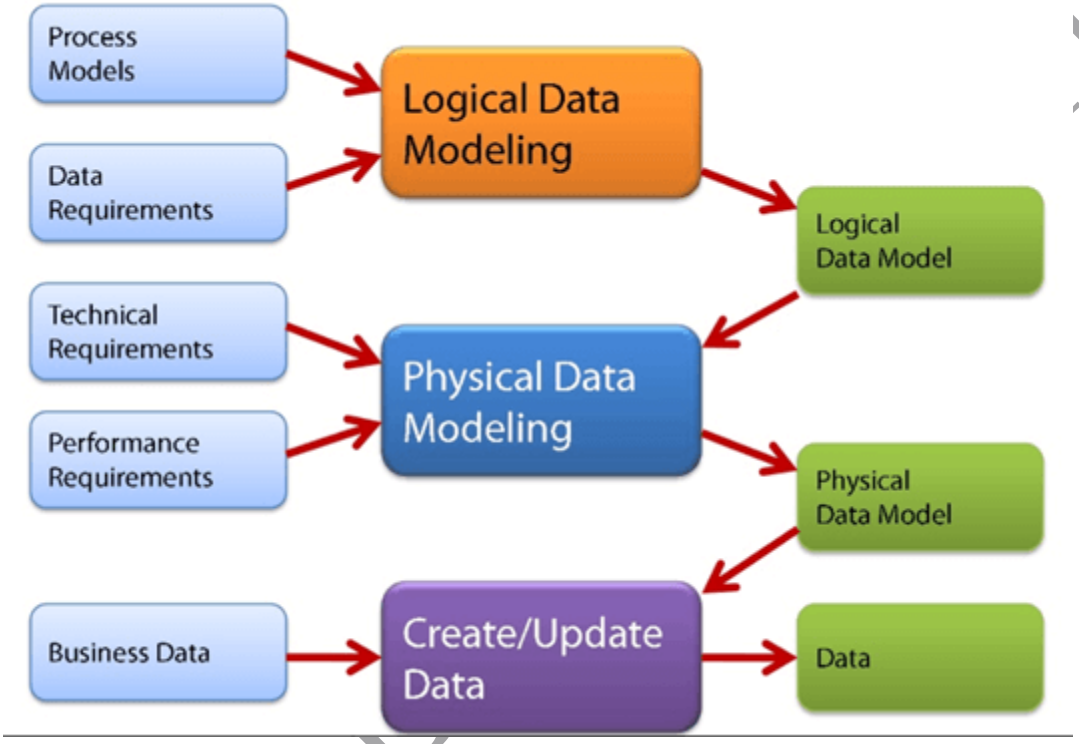
Types of Data Models

Types of Data Models : There are mainly three different types of data models: conceptual data models, logical data models and physical data models and each one has a specific purpose. The data models are used to represent the data and how it is stored in the database and to set the relationship between data items.

1. **Conceptual Data Model:** This Data Model defines **WHAT** the system contains. This model is typically created by Business stakeholders and Data Architects. The purpose is to organize, scope and define business concepts and rules.
2. **Logical Data Model:** Defines **HOW** the system should be implemented regardless of the DBMS. This model is typically created by Data Architects and Business Analysts. The purpose is to developed technical map of rules and data structures.

DIWAKAR EDUCATION HUB

3. **Physical Data Model**: This Data Model describes **HOW** the system will be implemented using a specific DBMS system. This model is typically created by DBA and developers. The purpose is actual implementation of the database.



Types of Data Model

Conceptual Data Model

A **Conceptual Data Model** is an organized view of database concepts and their relationships. The purpose of creating conceptual data model is to establish entities, their attributes and relationships. In this data modeling level, there is hardly any detail available of the actual database structure. Business stakeholders and data architects typically create a conceptual data model.

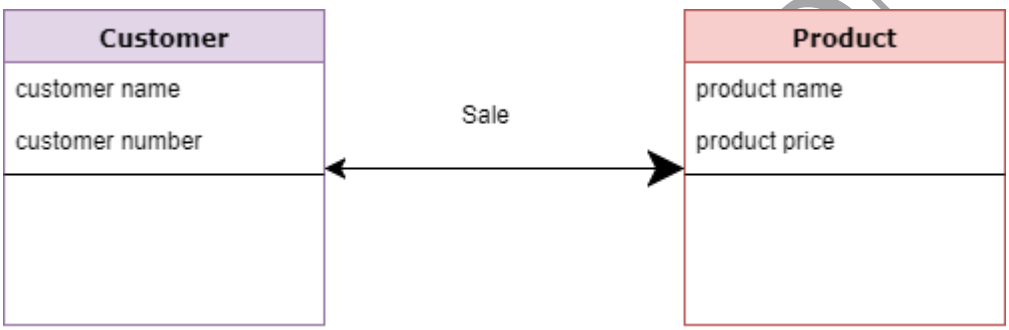
The 3 basic tenants of Conceptual Data Model are

- **Entity**: A real-world thing
- **Attribute**: Characteristics or properties of an entity

- **Relationship**: Dependency or association between two entities

Data model example:

- Customer and Product are two entities. Customer number and name are attributes of the Customer entity
- Product name and price are attributes of product entity
- Sale is the relationship between the customer and product



Conceptual Data Model

Characteristics of a conceptual data model

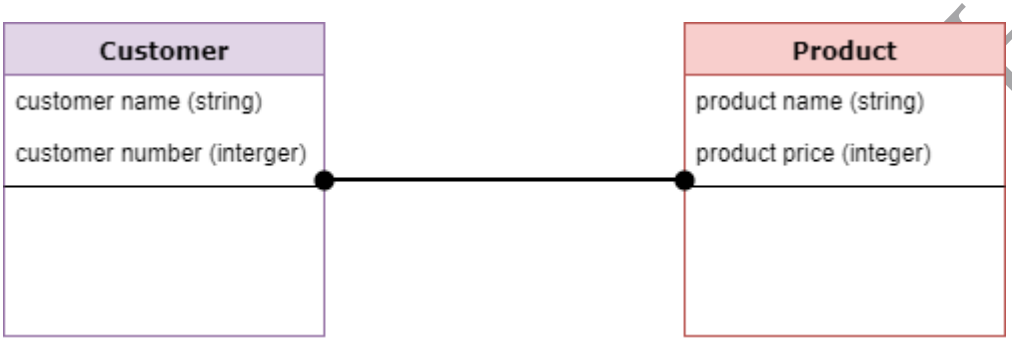
- Offers Organisation-wide coverage of the business concepts.
- This type of Data Models are designed and developed for a business audience.
- The conceptual model is developed independently of hardware specifications like data storage capacity, location or software specifications like DBMS vendor and technology. The focus is to represent data as a user will see it in the "real world."

Conceptual data models known as Domain models create a common vocabulary for all stakeholders by establishing basic concepts and scope.

Logical Data Model

DIWAKAR EDUCATION HUB

The **Logical Data Model** is used to define the structure of data elements and to set relationships between them. Logical data model adds further information to the conceptual data model elements. The advantage of using Logical data model is to provide a foundation to form the base for the Physical model. However, the modeling structure remains generic.



Logical Data Model

At this Data Modeling level, no primary or secondary key is defined. At this Data modeling level, you need to verify and adjust the connector details that were set earlier for relationships.

Characteristics of a Logical data model

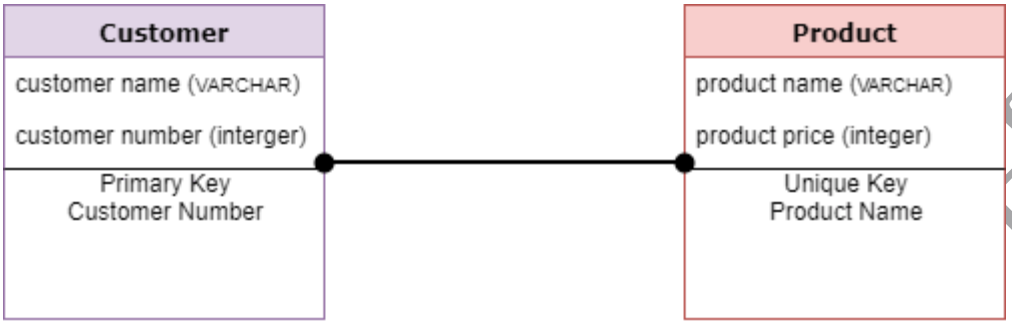
- Describes data needs for a single project but could integrate with other logical data models based on the scope of the project.
- Designed and developed independently from the DBMS.
- Data attributes will have datatypes with exact precisions and length.
- Normalization processes to the model is applied typically till 3NF.

Physical Data Model

A **Physical Data Model** describes database specific implementation of the data model. It offers database abstraction and helps generate schema. This is because of the richness of meta-data offered by a Physical Data Model. Physical data

DIWAKAR EDUCATION HUB

model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers and other RDBMS features.



Physical Data Model

Characteristics of a physical data model:

- The physical data model describes data need for a single project or application though it maybe integrated with other physical data models based on project scope.
- Data Model contains relationships between tables that which addresses cardinality and nullability of the relationships.
- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project.
- Columns should have exact datatypes, lengths assigned and default values.
- Primary and Foreign keys, views, indexes, access profiles, and authorizations, etc. are defined.

Advantages and Disadvantages of Data Model:

Advantages of Data model:

- The main goal of a designing data model is to make certain that data objects offered by the functional team are represented accurately.

DIWAKAR EDUCATION HUB

- The data model should be detailed enough to be used for building the physical database.
- The information in the data model can be used for defining the relationship between tables, primary and foreign keys, and stored procedures.
- Data Model helps business to communicate the within and across organizations.
- Data model helps to documents data mappings in ETL process
- Help to recognize correct sources of data to populate the model

Disadvantages of Data model:

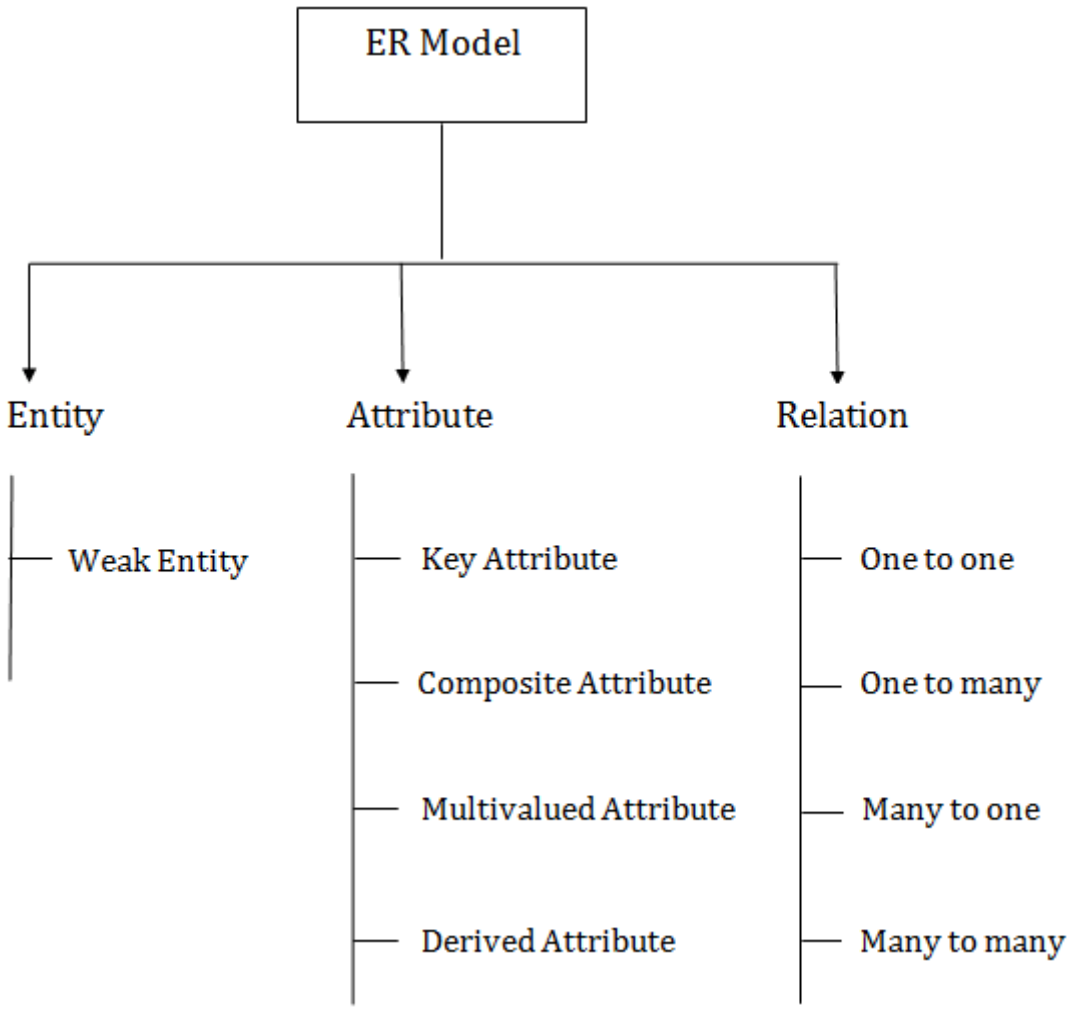
- To develop Data model one should know physical data stored characteristics.
- This is a navigational system produces complex application development, management. Thus, it requires a knowledge of the biographical truth.
- Even smaller change made in structure require modification in the entire application.
- There is no set data manipulation language in DBMS.

Entity Relationship (E-R) Model

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

Component of ER Diagram

DIWAKAR EDUCATION HUB



ER Diagram

ER Model is represented by means of an ER diagram. Any object, for example, entities, attributes of an entity, relationship sets, and attributes of relationship sets, can be represented with the help of an ER diagram.

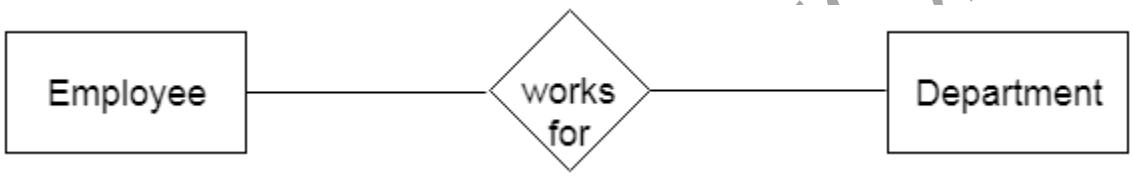
Entity

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

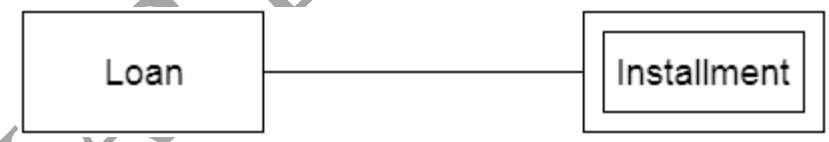
An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



a. Weak Entity

An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



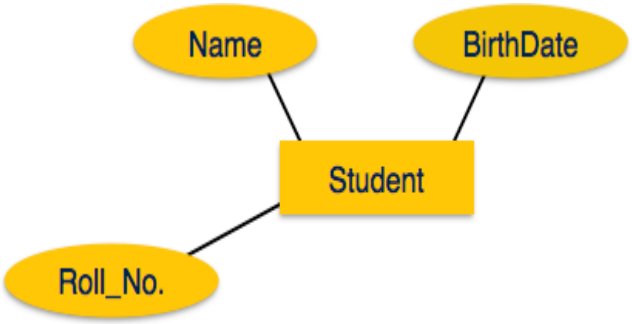
Attributes

Entities are represented by means of their properties, called **attributes**. All attributes have values. For example, a student entity may have name, class, and age as attributes.

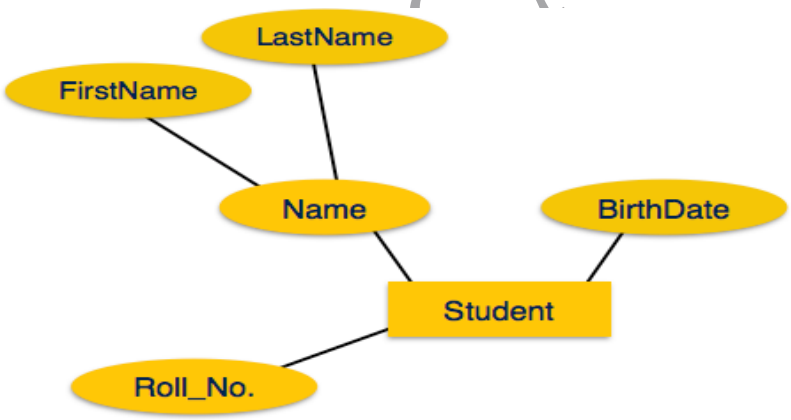
There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

DIWAKAR EDUCATION HUB

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).

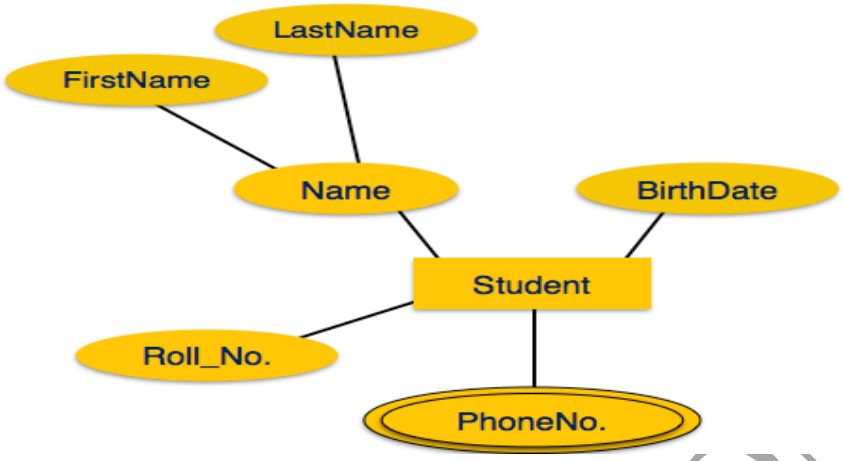


If the attributes are **composite**, they are further divided in a tree like structure. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.

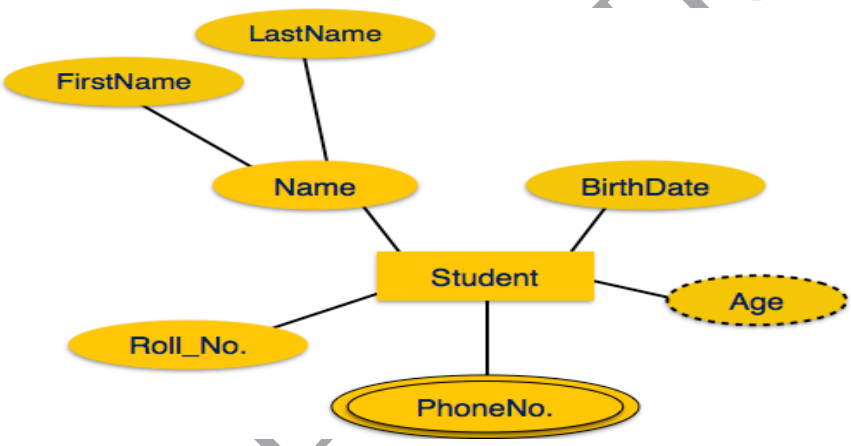


Multivalued attributes are depicted by double ellipse.

DIWAKAR EDUCATION HUB



Derived attributes are depicted by dashed ellipse.



Types of Attributes

- **Simple attribute** – Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute** – Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute** – Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department

DIWAKAR EDUCATION HUB

should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data_of_birth.

- **Single-value attribute** − Single-value attributes contain single value. For example − Social_Security_Number.
- **Multi-value attribute** − Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc.

These attribute types can come together in a way like –

- simple single-valued attributes
- simple multi-valued attributes
- composite single-valued attributes
- composite multi-valued attributes

Entity-Set and Keys

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.

For example, the roll_number of a student makes him/her identifiable among students.

- **Super Key** − A set of attributes (one or more) that collectively identifies an entity in an entity set.
- **Candidate Key** − A minimal super key is called a candidate key. An entity set may have more than one candidate key.
- **Primary Key** − A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

***Relational Database Design (RDD)* mean**

DIWAKAR EDUCATION HUB

Relational database design (RDD) models information and data into a set of tables with rows and columns. Each row of a relation/table represents a record, and each column represents an attribute of data. The Structured Query Language (SQL) is used to manipulate relational databases. The design of a relational database is composed of four stages, where the data are modeled into a set of related tables. The stages are:

- Define relations/attributes
- Define primary keys
- Define relationships
- Normalization

Relational Database Design (RDD)

Relational databases differ from other databases in their approach to organizing data and performing transactions. In an RDD, the data are organized into tables and all types of data access are carried out via controlled transactions. Relational database design satisfies the ACID (atomicity, consistency, integrity and durability) properties required from a database design. Relational database design mandates the use of a database server in applications for dealing with data management problems.

The four stages of an RDD are as follows:

- Relations and attributes: The various tables and attributes related to each table are identified. The tables represent entities, and the attributes represent the properties of the respective entities.
- Primary keys: The attribute or set of attributes that help in uniquely identifying a record is identified and assigned as the primary key
- Relationships: The relationships between the various tables are established with the help of foreign keys. Foreign keys are attributes occurring in a table that are primary keys of another table. The types of relationships that can exist between the relations (tables) are:

DIWAKAR EDUCATION HUB

- One to one
- One to many
- Many to many

An entity-relationship diagram can be used to depict the entities, their attributes and the relationship between the entities in a diagrammatic way.

- Normalization: This is the process of optimizing the database structure. Normalization simplifies the database design to avoid redundancy and confusion. The different normal forms are as follows:
  - First normal form
  - Second normal form
  - Third normal form
  - Boyce-Codd normal form
  - Fifth normal form

By applying a set of rules, a table is normalized into the above normal forms in a linearly progressive fashion. The efficiency of the design gets better with each higher degree of normalization.

Relationship

The association among entities is called a relationship. For example, an employee **works_at** a department, a student **enrolls** in a course. Here, Works_at and Enrolls are called relationships.

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

DIWAKAR EDUCATION HUB



Types of relationship are as follows:

a. One-to-One Relationship

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

For example, A female can marry to one male, and a male can marry to one female.

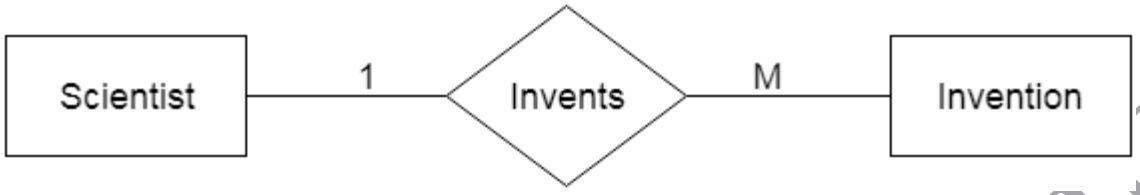


b. One-to-many relationship

When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

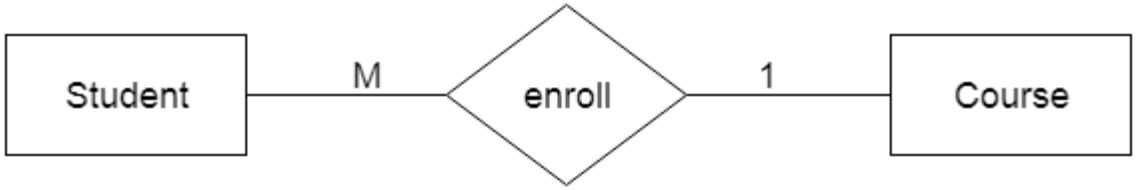
DIWAKAR EDUCATION HUB



c. Many-to-one relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

For example, Student enrolls for only one course, but a course can have many students.



d. Many-to-many relationship

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

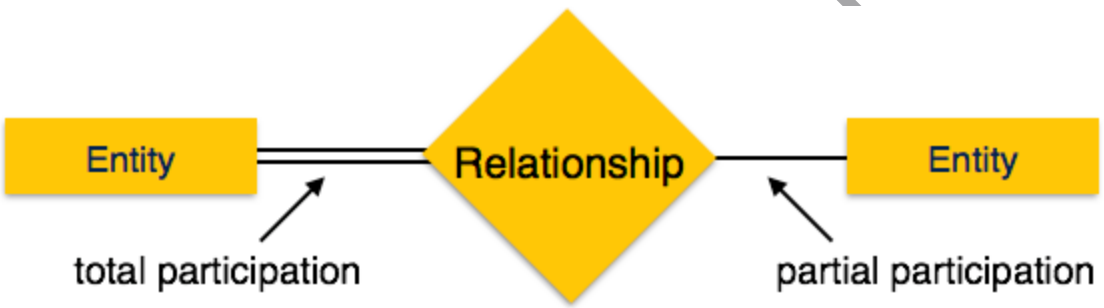
For example, Employee can assign by many projects and project can have many employees.

DIWAKAR EDUCATION HUB



Participation Constraints

- **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines.



Relationship Set

A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called **descriptive attributes**.

Degree of Relationship

The number of participating entities in a relationship defines the degree of the relationship.

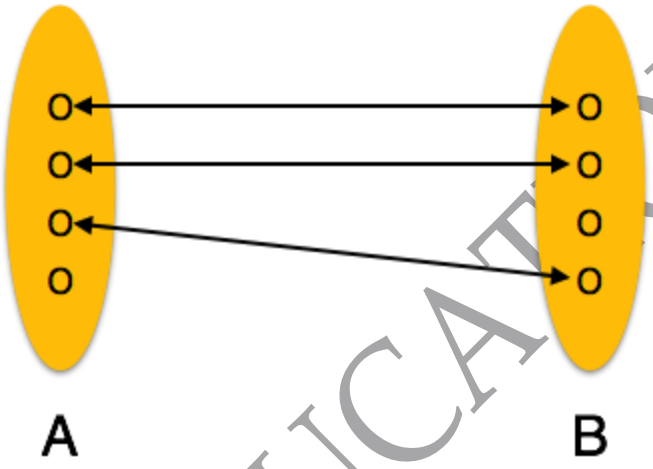
- Binary = degree 2
- Ternary = degree 3
- n-ary = degree

DIWAKAR EDUCATION HUB

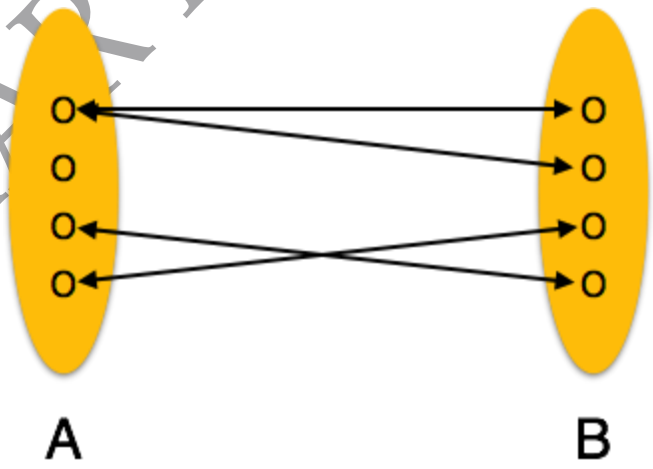
Mapping Cardinalities

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

- **One-to-one** − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

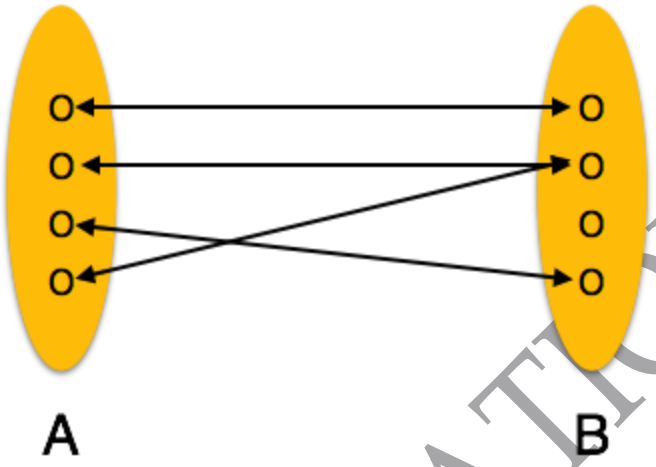


- **One-to-many** − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

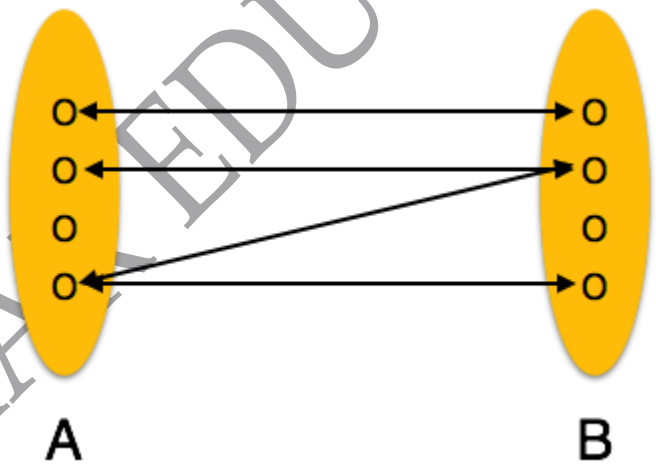


DIWAKAR EDUCATION HUB

- **Many-to-one** – More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



- **Many-to-many** – One entity from A can be associated with more than one entity from B and vice versa.



Notation of ER diagram

Database can be represented using the notations. In ER diagram, many notations are used to express the cardinality. These notations are as follows:

DIWAKAR EDUCATION HUB

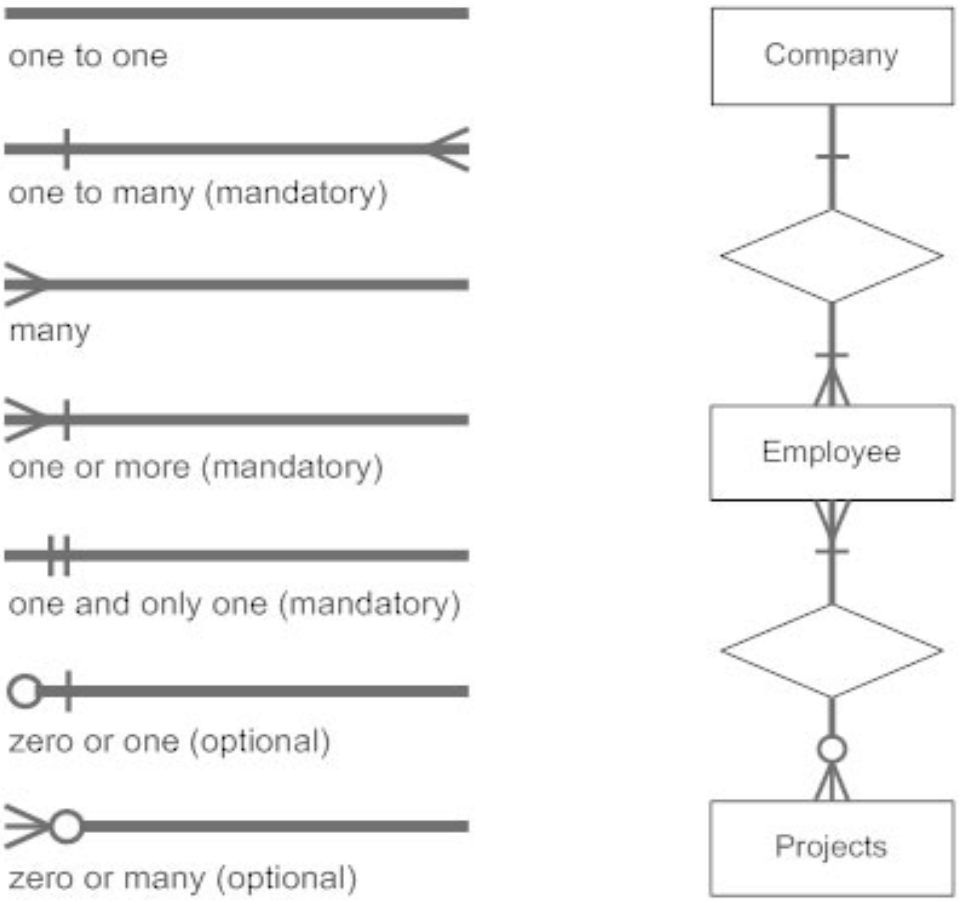


Fig: Notations of ER diagram

Relational Model concept

Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has a name or attribute.

Domain: It contains a set of atomic values that an attribute can take.

Attribute: It contains the name of a column in a particular table. Each attribute Ai must have a domain, dom(Ai)

DIWAKAR

Relational instance: In the relational database system, the relational instance is represented by a finite set of tuples. Relation instances do not have duplicate tuples.

Relational schema: A relational schema contains the name of the relation and name of all columns or attributes.

Relational key: In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.

Example: STUDENT Relation

NAME	ROLL_NO	PHONE_NO	ADDRESS	AGE
Ram	14795	7305758992	Noida	24
Shyam	12839	9026288936	Delhi	35
Laxman	33289	8583287182	Gurugram	20
Mahesh	27857	7086819134	Ghaziabad	27
Ganesh	17282	9028 9i3988	Delhi	40

- In the given table, NAME, ROLL_NO, PHONE_NO, ADDRESS, and AGE are the attributes.
- The instance of schema STUDENT has 5 tuples.
- t3 = <Laxman, 33289, 8583287182, Gurugram, 20>

Properties of Relations

- Name of the relation is distinct from all other relations.
- Each relation cell contains exactly one atomic (single) value
- Each attribute contains a distinct name
- Attribute domain has no significance
- tuple has no duplicate value
- Order of tuple can have a different sequence

Constraints on Relational database model

On modeling the design of the relational database we can put some restrictions like what values are allowed to be inserted in the relation, what kind of modifications and deletions are allowed in the relation. These are the restrictions we impose on the relational database.

In models like ER models, we did not have such features.

Constraints in the databases can be categorized into 3 main categories:

1. Constraints that are applied in the data model is called **Implicit constraints**.
2. Constraints that are directly applied in the schemas of the data model, by specifying them in the DDL(Data Definition Language). These are called as **schema-based constraints or Explicit constraints**.
3. Constraints that cannot be directly applied in the schemas of the data model. We call these Application based or **semantic constraints**.

So here we will deal with **Implicit constraints**.

Mainly Constraints on the relational database are of 4 types:

1. Domain constraints
2. Key constraints
3. Entity Integrity constraints
4. Referential integrity constraints

DIWAKAR EDUCATION HUB

1. Domain constraints :

1. Every domain must contain atomic values(smallest indivisible units) it means composite and multi-valued attributes are not allowed.
2. We perform datatype check here, which means when we assign a data type to a column we limit the values that it can contain. Eg. If we assign the datatype of attribute age as int, we cant give it values other then int datatype.

Example:

EID	Name	Phone
01	Bikash Dutta	123456789 234456678

Explanation:
In the above relation, Name is a composite attribute and Phone is a multi-values attribute, so it is violating domain constraint.

2. Key Constraints or Uniqueness Constraints :

1. These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.
2. A relation can have multiple keys or candidate keys(minimal superkey), out of which we choose one of the keys as primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with less number of attributes.
3. Null values are not allowed in the primary key, hence Not Null constraint is also a part of key constraint.

DIWAKAR EDUCATION HUB

Example:

EID	Name	Phone
01	Bikash	6026526747
02	Paul	7002494274
01	Tuhin	9234567892

Explanation:
In the above table, EID is the primary key, and first and the last tuple has the same value in EID ie 01, so it is violating the key constraint.

3. Entity Integrity Constraints :

1. Entity Integrity constraints says that no primary key can take NULL value, since using primary key we identify each tuple uniquely in a relation.

Example:

EID	Name	Phone
01	Bikash	7002494274
02	Paul	6026526747
NULL	Sony	9234567892

Explanation:
In the above relation, EID is made primary key, and the primary key cant take NULL values but in the third tuple, the primary key is null, so it is a violating Entity Integrity constraints.

4. Referential Integrity Constraints :

DIWAKAR EDUCATION HUB

1. The Referential integrity constraints is specified between two relations or tables and used to maintain the consistency among the tuples in two relations.
2. This constraint is enforced through foreign key, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2, then the foreign key of R1 is said to reference or refer to the primary key of relation R2.
3. The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take NULL values, but can't be empty.

Example:

EID	Name	DNO
01	Divine	12
02	Dino	22
04	Vivian	14

DNO	Place
12	Jaipur
13	Mumbai
14	Delhi

Explanation:

In the above, DNO of the first relation is the foreign key, and DNO in the second relation is the primary key. DNO = 22 in the foreign key of the first table is not allowed since DNO = 22
is not defined in the primary key of the second relation. Therefore Referential integrity constraints is violated here

Relational Language

DIWAKAR EDUCATION HUB

Relational language is a type of programming language in which the programming logic is composed of relations and the output is computed based on the query applied. Relational language works on relations among data and entities to compute a result. Relational language includes features from and is similar to functional programming language.

Relational language is primarily based on the relational data model, which governs relational database software and systems. In the relational model's programming context, the procedures are replaced by the relations among values. These relations are applied over the processed arguments or values to construct an output. The resulting output is mainly in the form of an argument or property. The side effects emerging from this programming logic are also handled by the procedures or relations.

Relational language is primarily based on the relational data model, which governs relational database software and systems. In the relational model's programming context, the procedures are replaced by the relations among values. These relations are applied over the processed arguments or values to construct an output. The resulting output is mainly in the form of an argument or property. The side effects emerging from this programming logic are also handled by the procedures or relations.

Relational Databases and Schemas

A relational database schema is an arrangement of relation states in such a manner that every relational database state fulfills the integrity constraints set on a relational database schema.

A **relational schema** outlines the database relationships and structure in a relational database program. It can be displayed graphically or written in the Structured Query Language (SQL) used to build tables in a relational database.

A relational schema contains the name of the relation and name of all columns or attributes.

DIWAKAR EDUCATION HUB

A relation schema represents name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.

a relational database schema is an arrangement of integrity constraints. Thus, in the context of relational database schema following points deserve a particular consideration:

1. A specific characteristic, that bears the same real-world concept, may appear in more than one relationship with the same or a different name. For example, in Employees relation, Employee Id (EmpId) is represented in Vouchers as AuthBy and PrepBy.
2. The specific real-world concept that appears more than once in a relationship should be represented by different names. For example, an employee is represented as subordinate or junior by using EmpId and as a superior or senior by using SuperId, in the employee's relation.
3. The integrity constraints that are specified on database schema shall apply to every database state of that schema.

Understanding a Relational Schema

A relational schema for a database is an outline of how data is organized. It can be a graphic illustration or another kind of chart used by programmers to understand how each table is laid out, including the columns and the types of data they hold and how tables connect. It can also be written in SQL code.

A database schema usually specifies which columns are primary keys in tables and which other columns have special constraints such as being required to have unique values in each record. It also usually specifies which columns in which tables contain references to data in other tables, often by including primary keys from other table records so that rows can be easily joined. These are called foreign key columns. For example, a customer order table may contain a

DIWAKAR EDUCATION HUB

customer number column that is a foreign key referencing the primary key of the customer table.

Relational Model Diagram

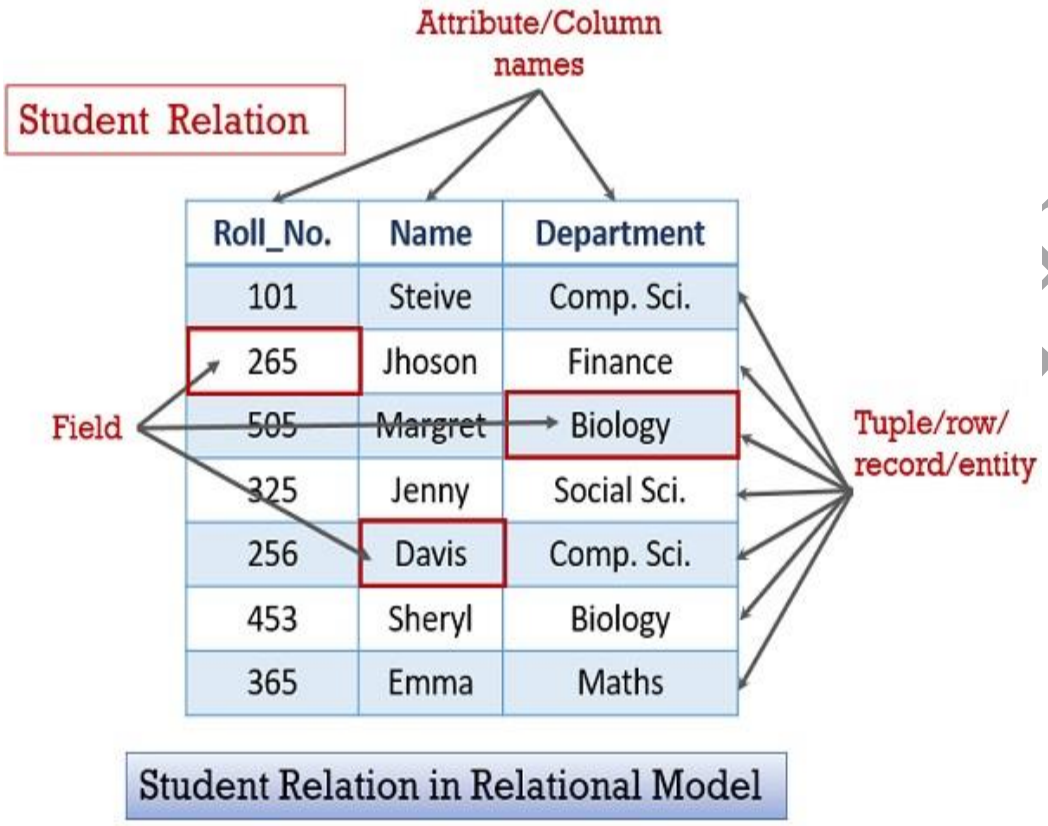
The figure below indicates a relation in a relational model.

Roll_No.	Name	Department
101	Steive	Comp. Sci.
265	Jhoson	Finance
505	Margret	Biology
325	Jenny	Social Sci.
256	Davis	Comp. Sci.
453	Sheryl	Biology
365	Emma	Maths

Student Relation in Relational Model

It is a Student relation and it is having entries of 5 students (tuples) in it. The figure below will help you identify the relation, attributes, tuples and field in a relational model.

DIWAKAR EDUCATION HUB



Update Operations, and Dealing with Constraint Violations

The operations of the relational model can be categorized into *retrievals* and *updates.* The relational algebra operations, which can be used to specify **retrievals**, are discussed in detail in Chapter 6. A relational algebra expression forms a new relation after applying a number of algebraic operators to an existing set of relations; its main use is for querying a database to retrieve information. The user formulates a query that specifies the data of interest, and a new relation is formed by applying relational operators to retrieve this data. That **result relation** becomes the answer to (or result of) the user's query. Chapter 6 also introduces the language called relational calculus, which is used to define the new relation declaratively without giving a specific order of operations.

In this section, we concentrate on the database **modification** or **update** operations. There are three basic operations that can change the states of relations in the data-base: Insert, Delete, and

DIWAKAR EDUCATION HUB

Update (or Modify). They insert new data, delete old data, or modify existing data records. **Insert** is used to insert one or more new tuples in a relation, **Delete** is used to delete tuples, and **Update** (or **Modify**) is used to change the values of some attributes in existing tuples. Whenever these operations are applied, the integrity constraints specified on the relational database schema should not be violated. In this section we discuss the types of constraints that may be violated by each of these operations and the types of actions that may be taken if an operation causes a violation. We use the database shown in Figure 3.6 for examples and discuss only key constraints, entity integrity constraints, and the referential integrity constraints shown.

1. The Insert Operation

The **Insert** operation provides a list of attribute values for a new tuple *t* that is to be inserted into a relation *R*. Insert can violate any of the four types of constraints dis-cussed in the previous section. Domain constraints can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type. Key constraints can be violated if a key value in the new tuple *t* already exists in another tuple in the relation *r*(*R*). Entity integrity can be violated if any part of the primary key of the new tuple *t* is NULL. Referential integrity can be violated if the value of any foreign key in *t* refers to a tuple that does not exist in the referenced relation. Here are some examples to illustrate this discussion.

Operation:

Insert <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.

Result: This insertion violates the entity integrity constraint (NULL for the primary key Ssn), so it is rejected.

Operation:

DIWAKAR EDUCATION HUB

Insert <‘Alicia’, ‘J’, ‘Zelaya’, ‘999887777’, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, ‘987654321’, 4> into EMPLOYEE.

Result: This insertion violates the key constraint because another tuple with the same Ssn value already exists in the EMPLOYEE relation, and so it is rejected.

Operation:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’, ‘1960-04-05’, ‘6357 Windswept, Katy, TX’, F, 28000, ‘987654321’, 7> into EMPLOYEE.

Result: This insertion violates the referential integrity constraint specified on Dno in EMPLOYEE because no corresponding referenced tuple exists in

DEPARTMENT with Dnumber = 7.

Operation:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, NULL, 4> into EMPLOYEE.

Result: This insertion satisfies all constraints, so it is acceptable.

If an insertion violates one or more constraints, the default option is to *reject the insertion.* In this case, it would be useful if the DBMS could provide a reason to the user as to why the insertion was rejected. Another option is to attempt to *correct the reason for rejecting the insertion,* but this is typically not used for violations caused by Insert; rather, it is used more often in correcting violations for Delete and Update. In the first operation, the DBMS could ask the user to provide a value for Ssn, and could then accept the insertion if a valid Ssn value is provided. In opera-tion 3, the DBMS could either ask the user to change the value of Dno to some valid value (or set it to NULL), or it could ask the user to insert a DEPARTMENT tuple with Dnumber = 7 and could accept the original insertion only after such an operation was accepted. Notice that in the latter case the insertion violation can **cascade** back to the EMPLOYEE relation if the user

DIWAKAR EDUCATION HUB

attempts to insert a tuple for department 7 with a value for Mgr_ssn that does not exist in the EMPLOYEE relation.

2. The Delete Operation

The **Delete** operation can violate only referential integrity. This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database. To specify deletion, a condition on the attributes of the relation selects the tuple (or tuples) to be deleted. Here are some examples.

Operation:

Delete the WORKS_ON tuple with Essn = '999887777' and Pno = 10. *Result*: This deletion is acceptable and deletes exactly one tuple.

Operation:

Delete the EMPLOYEE tuple with Ssn = '999887777'.

Result: This deletion is not acceptable, because there are tuples in WORKS_ON that refer to this tuple. Hence, if the tuple in EMPLOYEE is

deleted, referential integrity violations will result.

Operation:

Delete the EMPLOYEE tuple with Ssn = '333445555'.

Result: This deletion will result in even worse referential integrity violations, because the tuple involved is referenced by tuples from the EMPLOYEE,

DEPARTMENT, WORKS_ON, and DEPENDENT relations.

Several options are available if a deletion operation causes a violation. The first option, called **restrict,** is to *reject the deletion.* The second option, called **cascade,** is to *attempt to cascade (or propagate) the deletion* by deleting tuples that reference the tuple that is being deleted. For example, in operation 2,

DIWAKAR EDUCATION HUB

the DBMS could automati-cally delete the offending tuples from WORKS_ON with Essn = '999887777'. A third option, called **set null** or **set default,** is to *modify the referencing attribute values* that cause the violation; each such value is either set to NULL or changed to reference another default valid tuple. Notice that if a referencing attribute that causes a viola-tion is *part of the primary key,* it *cannot* be set to NULL; otherwise, it would violate entity integrity.

Combinations of these three options are also possible. For example, to avoid having operation 3 cause a violation, the DBMS may automatically delete all tuples from WORKS_ON and DEPENDENT with Essn = '333445555'. Tuples in EMPLOYEE with Super_ssn = '333445555' and the tuple in DEPARTMENT with Mgr_ssn = '333445555' can have their Super_ssn and Mgr_ssn values changed to other valid values or to NULL. Although it may make sense to delete automatically the WORKS_ON and DEPENDENT tuples that refer to an EMPLOYEE tuple, it may not make sense to delete other EMPLOYEE tuples or a DEPARTMENT tuple.

In general, when a referential integrity constraint is specified in the DDL, the DBMS will allow the database designer to *specify which of the options* applies in case of a violation of the constraint. We discuss how to specify these options in the SQL DDL in Chapter 4.

3. The Update Operation

The **Update** (or **Modify**) operation is used to change the values of one or more attributes in a tuple (or tuples) of some relation *R*. It is necessary to specify a condition on the attributes of the relation to select the tuple (or tuples) to be modified. Here are some examples.

Operation:

Update the salary of the EMPLOYEE tuple with Ssn = '999887777' to 28000. *Result*: Acceptable.

Operation:

DIWAKAR EDUCATION HUB

Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 1. *Result*: Acceptable.

Operation:

Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 7. *Result*: Unacceptable, because it violates referential integrity.

Operation:

Update the Ssn of the EMPLOYEE tuple with Ssn = '999887777' to '987654321'.

Result: Unacceptable, because it violates primary key constraint by repeating a value that already exists as a primary key in another tuple; it violates refer-ential integrity constraints because there are other relations that refer to the existing value of Ssn.

Updating an attribute that is *neither part of a primary key nor of a foreign key* usually causes no problems; the DBMS need only check to confirm that the new value is of the correct data type and domain. Modifying a primary key value is similar to delet-ing one tuple and inserting another in its place because we use the primary key to identify tuples. Hence, the issues discussed earlier in both Sections 3.3.1 (Insert) and 3.3.2 (Delete) come into play. If a foreign key attribute is modified, the DBMS must make sure that the new value refers to an existing tuple in the referenced relation (or is set to NULL). Similar options exist to deal with referential integrity violations caused by Update as those options discussed for the Delete operation. In fact, when a referential integrity constraint is specified in the DDL, the DBMS will allow the user to choose separate options to deal with a violation caused by Delete and a vio-lation caused by Update.

4. The Transaction Concept

A database application program running against a relational database typically executes one or more *transactions*. A **transaction** is an executing program that includes some database operations, such as reading from the database, or

DIWAKAR EDUCATION HUB

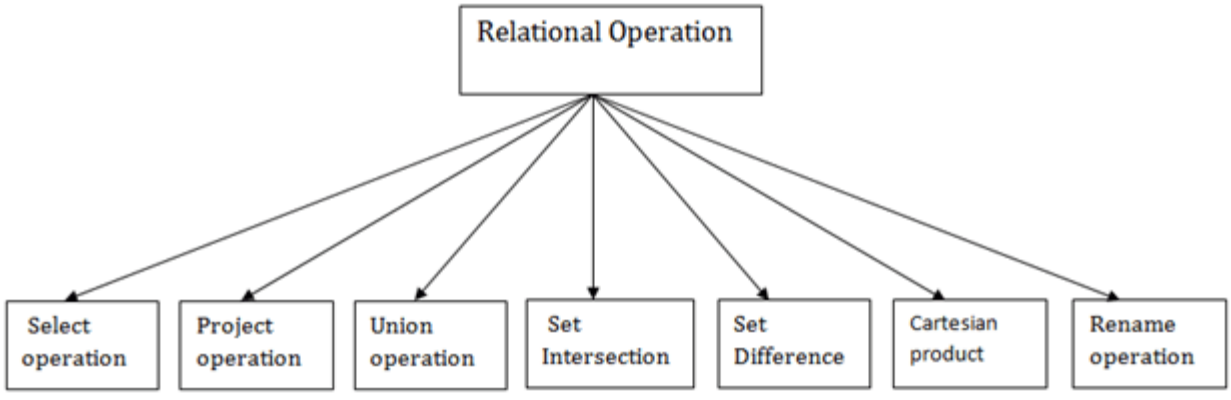
applying insertions, deletions, or updates to the database. At the end of the transaction, it must leave the database in a valid or consistent state that satisfies all the constraints spec-ified on the database schema. A single transaction may involve any number of retrieval operations (to be discussed as part of relational algebra and calculus in Chapter 6, and as a part of the language SQL in Chapters 4 and 5), and any number of update operations. These retrievals and updates will together form an atomic unit of work against the database. For example, a transaction to apply a bank with-drawal will typically read the user account record, check if there is a sufficient bal-ance, and then update the record by the withdrawal amount.

A large number of commercial applications running against relational databases in **online transaction processing (OLTP**) systems are executing transactions at rates that reach several hundred per second.

Relational Algebra

Relational algebra is a procedural query language. It gives a step by step process to obtain the result of the query. It uses operators to perform queries.

Types of Relational operation



1. Select Operation:

DT REDUCATION HUB

- The select operation selects tuples that satisfy a given predicate.
- It is denoted by sigma (σ).

1. Notation: σ p(r)

Where:

σ is used for selection prediction
r is used for relation
p is used as a propositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

For example: LOAN Relation

BRANCH_NAME	LOAN_NO	AMOUNT
Downtown	L-17	1000
Redwood	L-23	2000
Perryride	L-15	1500
Downtown	L-14	1500
Mianus	L-13	500
Roundhill	L-11	900
Perryride	L-16	1300

Input:

1. σ BRANCH_NAME="perryride" (LOAN)

Output:

BRANCH_NAME	LOAN_NO	AMOUNT
Perryride	L-15	1500
Perryride	L-16	1300

2. Project Operation:

- This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table.
- It is denoted by ∏.

1. Notation: ∏ A1, A2, An (r)

Where

A1, **A2**, **A3** is used as an attribute name of relation **r**.

Example: CUSTOMER RELATION

NAME	STREET	CITY
Jones	Main	Harrison
Smith	North	Rye
Hays	Main	Harrison
Curry	North	Rye

Johnson	Alma	Brooklyn
Brooks	Senator	Brooklyn

Input:

1. ∏ NAME, CITY (CUSTOMER)

Output:

NAME	CITY
Jones	Harrison
Smith	Rye
Hays	Harrison
Curry	Rye
Johnson	Brooklyn
Brooks	Brooklyn

3. Union Operation:

- Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
- It eliminates the duplicate tuples. It is denoted by ∪.

1. Notation: R ∪ S

A union operation must hold the following condition:

DIWAKAR EDUCATION HUB

- R and S must have the attribute of the same number.
- Duplicate tuples are eliminated automatically.

Example:

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Johnson	A-101
Smith	A-121
Mayes	A-321
Turner	A-176
Johnson	A-273
Jones	A-472
Lindsay	A-284

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Jones	L-17
Smith	L-23

Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17

Input:

1. ∏ CUSTOMER_NAME (BORROW) ∪ ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Johnson
Smith
Hayes
Turner
Jones
Lindsay
Jackson

Curry
Williams
Mayes

4. Set Intersection:

- Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S.
- It is denoted by intersection ∩.

1. Notation: R ∩ S

Example: Using the above DEPOSITOR table and BORROW table

Input:

1. ∏ CUSTOMER_NAME (BORROW) ∩ ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Smith
Jones

5. Set Difference:

- Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in R but not in S.
- It is denoted by intersection minus (-).

DIWAKAR EDUCATION HUB

1. Notation: R - S

Example: Using the above DEPOSITOR table and BORROW table

Input:

1. ∏ CUSTOMER_NAME (BORROW) - ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Jackson
Hayes
Willians
Curry

6. Cartesian product

- The Cartesian product is used to combine each row in one table with each row in the other table. It is also known as a cross product.
- It is denoted by X.

1. Notation: E X D

Example:

EMPLOYEE

EMP_ID	EMP_NAME	EMP_DEPT

DIWAKAR EDUCATION HUB

1	Smith	A
2	Harry	C
3	John	B

DEPARTMENT

DEPT_NO	DEPT_NAME
A	Marketing
B	Sales
C	Legal

Input:

1. EMPLOYEE X DEPARTMENT

Output:

EMP_ID	EMP_NAME	EMP_DEPT	DEPT_NO	DEPT_NAME
1	Smith	A	A	Marketing
1	Smith	A	B	Sales
1	Smith	A	C	Legal
2	Harry	C	A	Marketing

2	Harry	C	B	Sales
2	Harry	C	C	Legal
3	John	B	A	Marketing
3	John	B	B	Sales
3	John	B	C	Legal

7. Rename Operation:

The rename operation is used to rename the output relation. It is denoted by **rho** (ρ).

Example: We can use the rename operator to rename STUDENT relation to STUDENT1.

1. ρ(STUDENT1, STUDENT)

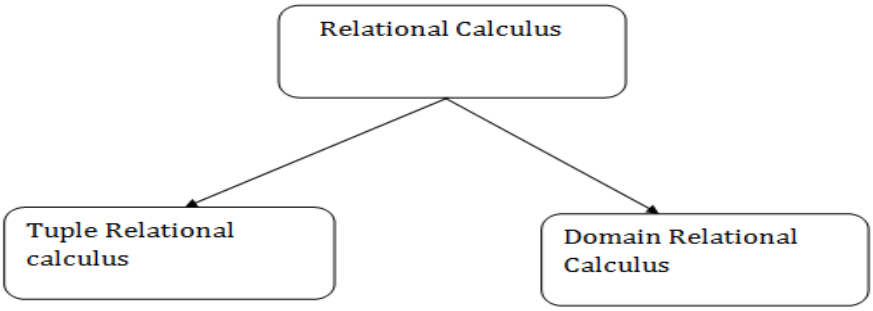
Note: Apart from these common operations Relational algebra can be used in Join operations.

Relational Calculus

- Relational calculus is a non-procedural query language. In the non-procedural query language, the user is concerned with the details of how to obtain the end results.
- The relational calculus tells what to do but never explains how to do.

Types of Relational calculus:

DIWAKAR EDUCATION



1. Tuple Relational Calculus (TRC)

- The tuple relational calculus is specified to select the tuples in a relation. In TRC, filtering variable uses the tuples of a relation.
- The result of the relation can have one or more tuples.

Notation:

1. {T | P (T)} or {T | Condition (T)}

Where

T is the resulting tuples

P(T) is the condition used to fetch T.

For example:

1. { T.name | Author(T) AND T.article = 'database' }

OUTPUT: This query selects the tuples from the AUTHOR relation. It returns a tuple with 'name' from Author who has written an article on 'database'.

TRC (tuple relation calculus) can be quantified. In TRC, we can use Existential (∃) and Universal Quantifiers (∀).

For example:

DIWAKAR EDUCATION HUB

1. { R| ∃T ∈ Authors(T.article='database' AND R.name=T.name)}

Output: This query will yield the same result as the previous one.

2. Domain Relational Calculus (DRC)

- The second form of relation is known as Domain relational calculus. In domain relational calculus, filtering variable uses the domain of attributes.
- Domain relational calculus uses the same operators as tuple calculus. It uses logical connectives ∧ (and), ∨ (or) and ¬ (not).
- It uses Existential (∃) and Universal Quantifiers (∀) to bind the variable.

Notation:

1. { a1, a2, a3, ..., an | P (a1, a2, a3, ... ,an)}

Where

a1, a2 are attributes
P stands for formula built by inner attributes

For example:

1. {< article, page, subject > | ∈ javatpoint ∧ subject = 'database'}

Output: This query will yield the article, page, and subject from the relational javatpoint, where the subject is a database.

Codd Rules

Dr Edgar F. Codd, after his extensive research on the Relational Model of database systems, came up with twelve rules of his own, which according to him, a database must obey in order to be regarded as a true relational database.

These rules can be applied on any database system that manages stored data using only its relational capabilities. This is a foundation rule, which acts as a base for all the other rules.

DIWAKAR EDUCATION HUB

Rule 1: Information Rule

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data.

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following − data is missing, data is not known, or data is not applicable.

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an online catalog, known as **data dictionary**, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

Rule 5: Comprehensive Data Sub-Language Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, must also be updatable by the system.

DIWAKAR EDUCATION HUB

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

Rule 9: Logical Data Independence

The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.

Rule 10: Integrity Independence

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

Rule 11: Distribution Independence

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

Rule 12: Non-Subversion Rule

DIWAKAR EDUCATION HUB

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

SQL

SQL is a programming language for Relational Databases. It is designed over relational algebra and tuple relational calculus. SQL comes as a package with all major distributions of RDBMS.

SQL comprises both data definition and data manipulation languages. Using the data definition properties of SQL, one can design and modify database schema, whereas data manipulation properties allows SQL to store and retrieve data from database.

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

Rules:

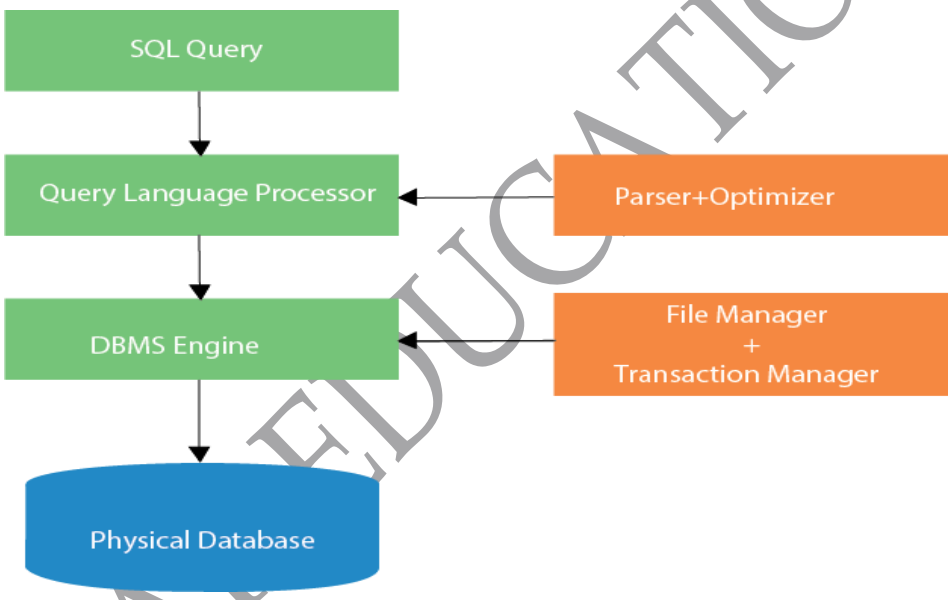
SQL follows the following rules:

- Structure query language is not case sensitive. Generally, keywords of SQL are written in uppercase.
- Statements of SQL are dependent on text lines. We can use a single SQL statement on one or multiple text line.
- Using the SQL statements, you can perform most of the actions in a database.
- SQL depends on tuple relational calculus and relational algebra.

DIWAKAR EDUCATION HUB

SQL process:

- When an SQL command is executing for any RDBMS, then the system figure out the best way to carry out the request and the SQL engine determines that how to interpret the task.
- In the process, various components are included. These components can be optimization Engine, Query engine, Query dispatcher, classic, etc.
- All the non-SQL queries are handled by the classic query engine, but SQL query engine won't handle logical files.



Characteristics of SQL

- SQL is easy to learn.
- SQL is used to access data from relational database management systems.
- SQL can execute queries against the database.
- SQL is used to describe the data.
- SQL is used to define the data in the database and manipulate it when needed.
- SQL is used to create and drop the database and table.

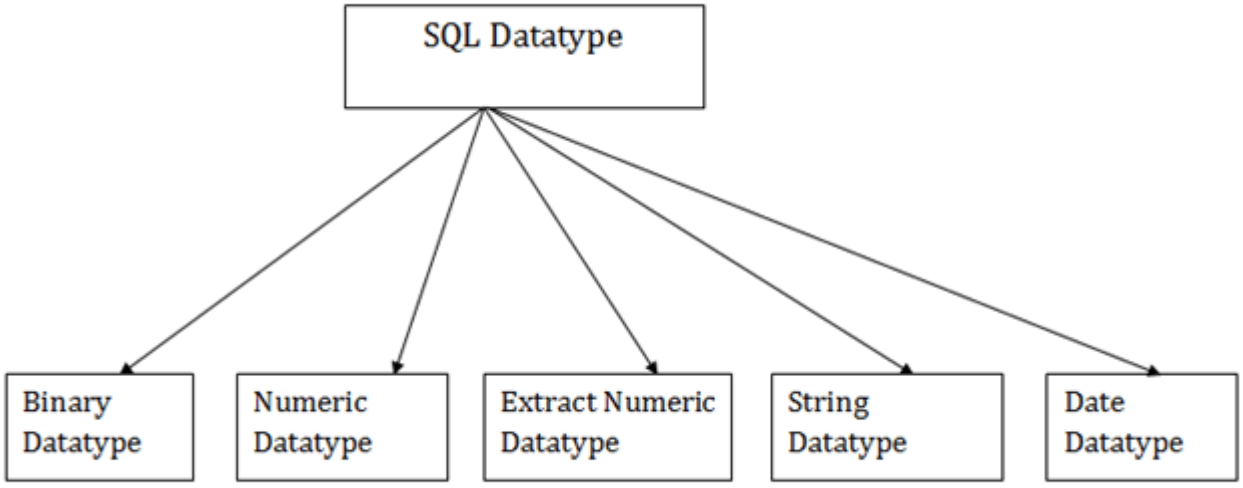
DIWAKAR EDUCATION HUB

- SQL is used to create a view, stored procedure, function in a database.
- SQL allows users to set permissions on tables, procedures, and views.

SQL Datatype

- SQL Datatype is used to define the values that a column can contain.
- Every column is required to have a name and data type in the database table.

Datatype of SQL:



1. Binary Datatypes

There are Three types of binary Datatypes which are given below:

Data Type	Description
binary	It has a maximum length of 8000 bytes. It contains fixed-length binary data.

varbinary	It has a maximum length of 8000 bytes. It contains variable-length binary data.
image	It has a maximum length of 2,147,483,647 bytes. It contains variable-length binary data.

2. Approximate Numeric Datatype :

The subtypes are given below:

Data type	From	To	Description
float	-1.79E + 308	1.79E + 308	It is used to specify a floating-point value e.g. 6.2, 2.9 etc.
real	-3.40e + 38	3.40E + 38	It specifies a single precision floating point number

3. Exact Numeric Datatype

The subtypes are given below:

Data type	Description

DIWAKAR EDUCATION HUB

int	It is used to specify an integer value.
smallint	It is used to specify small integer value.
bit	It has the number of bits to store.
decimal	It specifies a numeric value that can have a decimal number.
numeric	It is used to specify a numeric value.

4. Character String Datatype

The subtypes are given below:

Data type	Description
char	It has a maximum length of 8000 characters. It contains Fixed-length non-unicode characters.
varchar	It has a maximum length of 8000 characters. It contains variable-length non-unicode characters.
text	It has a maximum length of 2,147,483,647 characters. It contains variable-length non-unicode characters.

5. Date and time Datatypes

The subtypes are given below:

Datatype	Description
date	It is used to store the year, month, and days value.
time	It is used to store the hour, minute, and second values.
timestamp	It stores the year, month, day, hour, minute, and the second value.

SQL INSERT Statement

The SQL INSERT statement is used to insert a single or multiple data in a table. In SQL, You can insert the data in two ways:

1. Without specifying column name
2. By specifying column name

Sample Table

EMPLOYEE

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30

DIVAKAR EDUCATION

2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36

1. Without specifying column name

If you want to specify all column values, you can specify or ignore the column values.

Syntax

1. INSERT INTO TABLE_NAME
2. VALUES (value1, value2, value 3, ... Value N);

Query

1. INSERT INTO EMPLOYEE VALUES (6, 'Marry', 'Canada', 600000, 48);

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26

3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

2. By specifying column name

To insert partial column values, you must have to specify the column names.

Syntax

1. INSERT INTO TABLE_NAME
2. [(col1, col2, col3,.... col N)]
3. VALUES (value1, value2, value 3, ... Value N);

Query

1. INSERT INTO EMPLOYEE (EMP_ID, EMP_NAME, AGE) VALUES (7, 'Jack', 40);

Output: After executing this query, the table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26

DIVAKAR EDUCATION HUB

3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48
7	Jack	null	null	40

Note: In SQL INSERT query, if you add values for all columns then there is no need to specify the column name. But, you must be sure that you are entering the values in the same order as the column exists.

SQL Update Statement

The SQL UPDATE statement is used to modify the data that is already in the database. The condition in the WHERE clause decides that which row is to be updated.

Syntax

1. UPDATE table_name
2. SET column1 = value1, column2 = value2, ...
3. WHERE condition;

Sample Table

EMPLOYEE

DIWAKAR EDUCATION HUB

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Updating single record

Update the column EMP_NAME and set the value to 'Emma' in the row where SALARY is 500000.

Syntax

1. UPDATE table_name
2. SET column_name = value
3. WHERE condition;

Query

1. UPDATE EMPLOYEE

DIWAKAR EDUCATION HUB

2. SET EMP_NAME = 'Emma'
3. WHERE SALARY = 500000;

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Emma	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Updating multiple records

If you want to update multiple columns, you should separate each field assigned with a comma. In the EMPLOYEE table, update the column EMP_NAME to 'Kevin' and CITY to 'Boston' where EMP_ID is 5.

Syntax

1. UPDATE table_name

DIWAKAR HUB

2. SET column_name = value1, column_name2 = value2
3. WHERE condition;

Query

1. UPDATE EMPLOYEE
2. SET EMP_NAME = 'Kevin', City = 'Boston'
3. WHERE EMP_ID = 5;

Output

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Kevin	Boston	200000	36
6	Marry	Canada	600000	48

Without use of WHERE clause

If you want to update all row from a table, then you don't need to use the WHERE clause. In the EMPLOYEE table, update the column EMP_NAME as 'Harry'.

DUCATION HUB

Syntax

1. UPDATE table_name
2. SET column_name = value1;

Query

1. UPDATE EMPLOYEE
2. SET EMP_NAME = 'Harry';

Output

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Harry	Chicago	200000	30
2	Harry	Austin	300000	26
3	Harry	Denver	100000	42
4	Harry	Washington	500000	29
5	Harry	Los angels	200000	36
6	Harry	Canada	600000	48

SQL DELETE Statement

DIGITION HUB

The SQL DELETE statement is used to delete rows from a table. Generally, DELETE statement removes one or more records form a table.

Syntax

1. DELETE FROM table_name WHERE some_condition;

Sample Table

EMPLOYEE

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Deleting Single Record

Delete the row from the table EMPLOYEE where EMP_NAME = 'Kristen'. This will delete only the fourth row.

Query

1. DELETE FROM EMPLOYEE

2. WHERE EMP_NAME = 'Kristen';

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Deleting Multiple Record

Delete the row from the EMPLOYEE table where AGE is 30. This will delete two rows(first and third row).

Query

1. DELETE FROM EMPLOYEE WHERE AGE= 30;

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE

DIWAKAR EDUCATION HUB

2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Delete all of the records

Delete all the row from the EMPLOYEE table. After this, no records left to display. The EMPLOYEE table will become empty.

Syntax

DELETE * FROM table_name;

or

DELETE FROM table_name;

Query

1. DELETE FROM EMPLOYEE;

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE

Note: Using the condition in the WHERE clause, we can delete single as well as multiple records. If you want to delete all the records from the table, then you don't need to use the WHERE clause.

DIVAKAR EDUCATION

Views in SQL

- Views in SQL are considered as a virtual table. A view also contains rows and columns.
- To create the view, we can select the fields from one or more tables present in the database.
- A view can either have specific rows based on certain condition or all the rows of a table.

Sample table:

Student_Detail

STU_ID	NAME	ADDRESS
1	Stephan	Delhi
2	Kathrin	Noida
3	David	Ghaziabad
4	Alina	Gurugram

Student_Marks

STU_ID	NAME	MARKS	AGE
1	Stephan	97	19

DIGITAL EDUCATION HUB

2	Kathrin	86	21
3	David	74	18
4	Alina	90	20
5	John	96	18

1. Creating view

A view can be created using the **CREATE VIEW** statement. We can create a view from a single table or multiple tables.

Syntax:

1. CREATE VIEW view_name AS
2. SELECT column1, column2.....
3. FROM table_name
4. WHERE condition;

2. Creating View from a single table

In this example, we create a View named DetailsView from the table Student_Detail.

Query:

1. CREATE VIEW DetailsView AS
2. SELECT NAME, ADDRESS
3. FROM Student_Details

DIWAKAR EDUCATION HUB

4. WHERE STU_ID < 4;

Just like table query, we can query the view to view the data.

1. SELECT * FROM DetailsView;

Output:

NAME	ADDRESS
Stephan	Delhi
Kathrin	Noida
David	Ghaziabad

3. Creating View from multiple tables

View from multiple tables can be created by simply include multiple tables in the SELECT statement.

In the given example, a view is created named MarksView from two tables Student_Detail and Student_Marks.

Query:

1. CREATE VIEW MarksView AS
2. SELECT Student_Detail.NAME, Student_Detail.ADDRESS, Student_Marks.MARKS
3. FROM Student_Detail, Student_Mark
4. WHERE Student_Detail.NAME = Student_Marks.NAME;

DIWAKAR EDUCATION HUB

To display data of View MarksView:

1. SELECT * FROM MarksView;

NAME	ADDRESS	MARKS
Stephan	Delhi	97
Kathrin	Noida	86
David	Ghaziabad	74
Alina	Gurugram	90

4. Deleting View

A view can be deleted using the Drop View statement.

Syntax

1. DROP VIEW view_name;

Example:

If we want to delete the View **MarksView**, we can do this as:

1. DROP VIEW MarksView;

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events −

- A **database manipulation (DML)** statement (DELETE, INSERT, or UPDATE)

DIWAKAR EDU

- A **database definition (DDL)** statement (CREATE, ALTER, or DROP).
- A **database operation** (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

Triggers can be written for the following purposes −

- Generating some derived column values automatically
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

Creating Triggers

The syntax for creating a trigger is −

```
CREATE [OR REPLACE ] TRIGGER trigger_name

{BEFORE | AFTER | INSTEAD OF }

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF col_name]

ON table_name

[REFERENCING OLD AS o NEW AS n]
```

DIWAKAR EDUCATION HUB

```
[FOR EACH ROW]
WHEN (condition)
DECLARE
  Declaration-statements
BEGIN
  Executable-statements
EXCEPTION
  Exception-handling-statements
END;
```

Where,

- CREATE [OR REPLACE] TRIGGER trigger_name − Creates or replaces an existing trigger with the *trigger_name*.
- {BEFORE | AFTER | INSTEAD OF} − This specifies when the trigger will be executed. The INSTEAD OF clause is used for creating trigger on a view.
- {INSERT [OR] | UPDATE [OR] | DELETE} − This specifies the DML operation.
- [OF col_name] − This specifies the column name that will be updated.
- [ON table_name] − This specifies the name of the table associated with the trigger.
- [REFERENCING OLD AS o NEW AS n] − This allows you to refer new and old values for various DML statements, such as INSERT, UPDATE, and DELETE.
- [FOR EACH ROW] − This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected. Otherwise the trigger will execute

DIWAKAR EDUCATION HUB

just once when the SQL statement is executed, which is called a table level trigger.

- WHEN (condition) − This provides a condition for rows for which the trigger would fire. This clause is valid only for row-level triggers.

Example

To start with, we will be using the CUSTOMERS table we had created and used in the previous chapters −

Select * from customers;

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00

The following program creates a **row-level** trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values −

```
CREATE OR REPLACE TRIGGER display_salary_changes

BEFORE DELETE OR INSERT OR UPDATE ON customers

FOR EACH ROW

WHEN (NEW.ID > 0)
```

DIWAKAR EDUCATION HUB

```
DECLARE
  sal_diff number;
BEGIN
  sal_diff := :NEW.salary  - :OLD.salary;
  dbms_output.put_line('Old salary: ' || :OLD.salary);
  dbms_output.put_line('New salary: ' || :NEW.salary);
  dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
```

When the above code is executed at the SQL prompt, it produces the following result –

Trigger created.

The following points need to be considered here –

- OLD and NEW references are not available for table-level triggers, rather you can use them for record-level triggers.
- If you want to query the table in the same trigger, then you should use the AFTER keyword, because triggers can query the table or change it again only after the initial changes are applied and the table is back in a consistent state.
- The above trigger has been written in such a way that it will fire before any DELETE or INSERT or UPDATE operation on the table, but you can write your trigger on a single or multiple operations, for example BEFORE DELETE, which will fire whenever a record will be deleted using the DELETE operation on the table.

DIWAKAR EDUCATION HUB

Triggering a Trigger

Let us perform some DML operations on the CUSTOMERS table. Here is one INSERT statement, which will create a new record in the table −

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (7, 'Kriti', 22, 'HP', 7500.00 );
```

When a record is created in the CUSTOMERS table, the above create trigger, **display_salary_changes** will be fired and it will display the following result −

```
Old salary:

New salary: 7500

Salary difference:
```

Because this is a new record, old salary is not available and the above result comes as null. Let us now perform one more DML operation on the CUSTOMERS table. The UPDATE statement will update an existing record in the table −

```
UPDATE customers

SET salary = salary + 500

WHERE id = 2;
```

When a record is updated in the CUSTOMERS table, the above create trigger, **display_salary_changes** will be fired and it will display the following result −

```
Old salary: 1500

New salary: 2000

Salary difference: 500
```

DIWAKAR EDUCATION HUB

SQL injection (SQLi)

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.

In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.

Impact of a successful SQL injection attack

A successful SQL injection attack can result in unauthorized access to sensitive data, such as passwords, credit card details, or personal user information. Many high-profile data breaches in recent years have been the result of SQL injection attacks, leading to reputational damage and regulatory fines. In some cases, an attacker can obtain a persistent backdoor into an organization's systems, leading to a long-term compromise that can go unnoticed for an extended period.

SQL injection examples

There are a wide variety of SQL injection vulnerabilities, attacks, and techniques, which arise in different situations. Some common SQL injection examples include:

- **Retrieving hidden data**, where you can modify an SQL query to return additional results.
- **Subverting application logic**, where you can change a query to interfere with the application's logic.
- **UNION attacks**, where you can retrieve data from different database tables.

DIWAKAR EDUCATION HUB

- **Examining the database**, where you can extract information about the version and structure of the database.
- **Blind SQL injection**, where the results of a query you control are not returned in the application's responses.

How to detect SQL injection vulnerabilities

The majority of SQL injection vulnerabilities can be found quickly and reliably using Burp Suite's web vulnerability scanner.

SQL injection can be detected manually by using a systematic set of tests against every entry point in the application. This typically involves:

- Submitting the single quote character ' and looking for errors or other anomalies.
- Submitting some SQL-specific syntax that evaluates to the base (original) value of the entry point, and to a different value, and looking for systematic differences in the resulting application responses.
- Submitting Boolean conditions such as OR 1=1 and OR 1=2, and looking for differences in the application's responses.
- Submitting payloads designed to trigger time delays when executed within an SQL query, and looking for differences in the time taken to respond.
- Submitting OAST payloads designed to trigger an out-of-band network interaction when executed within an SQL query, and monitoring for any resulting interactions.

SQL injection in different parts of the query

Most SQL injection vulnerabilities arise within the WHERE clause of a SELECT query. This type of SQL injection is generally well-understood by experienced testers.

DIWAKAR EDUCATION HUB

But SQL injection vulnerabilities can in principle occur at any location within the query, and within different query types. The most common other locations where SQL injection arises are:

- In UPDATE statements, within the updated values or the WHERE clause.
- In INSERT statements, within the inserted values.
- In SELECT statements, within the table or column name.
- In SELECT statements, within the ORDER BY clause.

Second-order SQL injection

First-order SQL injection arises where the application takes user input from an HTTP request and, in the course of processing that request, incorporates the input into an SQL query in an unsafe way.

In second-order SQL injection (also known as stored SQL injection), the application takes user input from an HTTP request and stores it for future use. This is usually done by placing the input into a database, but no vulnerability arises at the point where the data is stored. Later, when handling a different HTTP request, the application retrieves the stored data and incorporates it into an SQL query in an unsafe way.

Second-order SQL injection often arises in situations where developers are aware of SQL injection vulnerabilities, and so safely handle the initial placement of the input into the database. When the data is later processed, it is deemed to be safe, since it was previously placed into the database safely. At this point, the data is handled in an unsafe way, because the developer wrongly deems it to be trusted.

Database-specific factors

Some core features of the SQL language are implemented in the same way across popular database platforms, and so many ways of detecting and exploiting SQL injection vulnerabilities work identically on different types of database.

DIWAKAR EDUCATION HUB

However, there are also many differences between common databases. These mean that some techniques for detecting and exploiting SQL injection work differently on different platforms. For example:

- Syntax for string concatenation.
- Comments.
- Batched (or stacked) queries.
- Platform-specific APIs.
- Error messages.

How to prevent SQL injection

Most instances of SQL injection can be prevented by using parameterized queries (also known as prepared statements) instead of string concatenation within the query.

The following code is vulnerable to SQL injection because the user input is concatenated directly into the query:

```
String query = "SELECT * FROM products WHERE category = '"+ input + "'";

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(query);
```

This code can be easily rewritten in a way that prevents the user input from interfering with the query structure:

```
PreparedStatement statement = connection.prepareStatement("SELECT * FROM products WHERE category = ?");

statement.setString(1, input);

ResultSet resultSet = statement.executeQuery();
```

Parameterized queries can be used for any situation where untrusted input appears as data within the query, including the WHERE clause and values in

DIWAKAR EDUCATION HUB

an INSERT or UPDATE statement. They can't be used to handle untrusted input in other parts of the query, such as table or column names, or the ORDER BY clause. Application functionality that places untrusted data into those parts of the query will need to take a different approach, such as white-listing permitted input values, or using different logic to deliver the required behavior.

For a parameterized query to be effective in preventing SQL injection, the string that is used in the query must always be a hard-coded constant, and must never contain any variable data from any origin. Do not be tempted to decide case-by-case whether an item of data is trusted, and continue using string concatenation within the query for cases that are considered safe. It is all too easy to make mistakes about the possible origin of data, or for changes in other code to violate assumptions about what data is tainted.

Functional Dependency

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

$$X \rightarrow Y$$

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

For example:

Assume we have an employee table with attributes: Emp_Id, Emp_Name, Emp_Address.

Here Emp_Id attribute can uniquely identify the Emp_Name attribute of employee table because if we know the Emp_Id, we can tell that employee name associated with it.

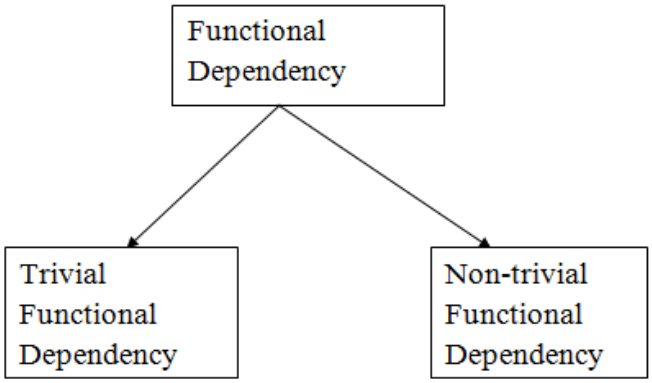
Functional dependency can be written as:

$$Emp\_Id \rightarrow Emp\_Name$$

DIWAKAR EDUCATION HUB

We can say that Emp_Name is functionally dependent on Emp_Id.

Types of Functional dependency



1. Trivial functional dependency

- A → B has trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial like: A → A, B → B

Example:

1. Consider a table with two columns Employee_Id and Employee_Name.
2. {Employee_id, Employee_Name} → Employee_Id is a trivial functional de pendency as
3. Employee_Id is a subset of {Employee_Id, Employee_Name}.
4. Also, Employee_Id → Employee_Id and Employee_Name → Employee_N ame are trivial dependencies too.

2. Non-trivial functional dependency

- A → B has a non-trivial functional dependency if B is not a subset of A.
- When A intersection B is NULL, then A → B is called as complete non-trivial.

DIWAKAR EDUCATION HUB

Example:

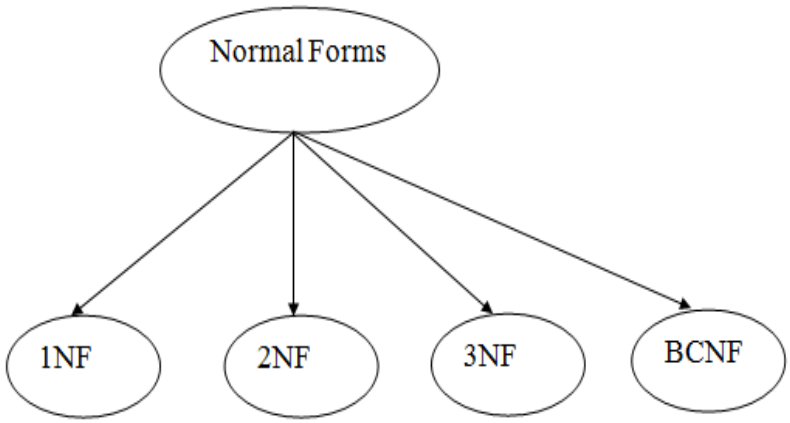
1. ID → Name,
2. Name → DOB

Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Types of Normal Forms

There are the four types of normal forms:



Normal Form	Description

DIWAKAR EDUCATION HUB

1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
4NF	A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
5NF	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

Transaction

- The transaction is a set of logically related operation. It contains a group of tasks.
- A transaction is an action or series of actions. It is performed by a single user to perform operations for accessing the contents of the database.

Example: Suppose an employee of bank transfers Rs 800 from X's account to Y's account. This small transaction contains several low-level tasks:

X's Account

1. Open_Account(X)
2. Old_Balance = X.balance
3. New_Balance = Old_Balance - 800
4. X.balance = New_Balance

DIWAKAR EDUCATION HUB

5. Close_Account(X)

Y's Account

1. Open_Account(Y)
2. Old_Balance = Y.balance
3. New_Balance = Old_Balance + 800
4. Y.balance = New_Balance
5. Close_Account(Y)

Operations of Transaction:

Following are the main operations of transaction:

Read(X): Read operation is used to read the value of X from the database and stores it in a buffer in main memory.

Write(X): Write operation is used to write the value back to the database from the buffer.

An example to debit transaction from an account which consists of following operations:

1. 1. R(X);
2. 2. X = X - 500;
3. 3. W(X);

Assume the value of X before starting of the transaction is 4000.

- The first operation reads X's value from database and stores it in a buffer.
- The second operation will decrease the value of X by 500. So buffer will contain 3500.
- The third operation will write the buffer's value to the database. So X's final value will be 3500.

DIWAKAR EDUCATION HUB

But it may be possible that because of the failure of hardware, software or power, etc. that transaction may fail before finished all the operations in the set.

For example: If in the above transaction, the debit transaction fails after executing operation 2 then X's value will remain 4000 in the database which is not acceptable by the bank.

To solve this problem, we have two important operations:

Commit: It is used to save the work done permanently.

Rollback: It is used to undo the work done.

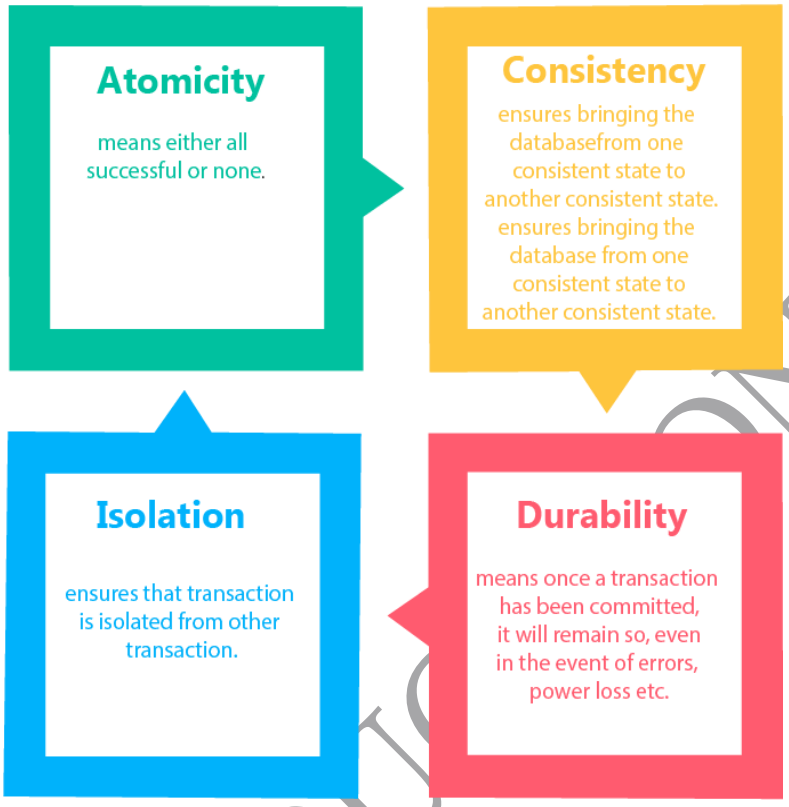
Transaction property

The transaction has the four properties. These are used to maintain consistency in a database, before and after the transaction.

Property of Transaction

1. Atomicity
2. Consistency
3. Isolation
4. Durability

DIWAKAR EDUCATION HUB



Atomicity

- It states that all operations of the transaction take place at once if not, the transaction is aborted.
- There is no midway, i.e., the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.

Atomicity involves the following two operations:

Abort: If a transaction aborts then all the changes made are not visible.

Commit: If a transaction commits then all the changes made are visible.

Consistency

DIWAKAR EDUCATION HUB

- The integrity constraints are maintained so that the database is consistent before and after the transaction.
- The execution of a transaction will leave a database in either its prior stable state or a new stable state.
- The consistent property of database states that every transaction sees a consistent database instance.
- The transaction is used to transform the database from one consistent state to another consistent state.

Isolation

- It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed.
- In isolation, if the transaction T1 is being executed and using the data item X, then that data item can't be accessed by any other transaction T2 until the transaction T1 ends.
- The concurrency control subsystem of the DBMS enforced the isolation property.

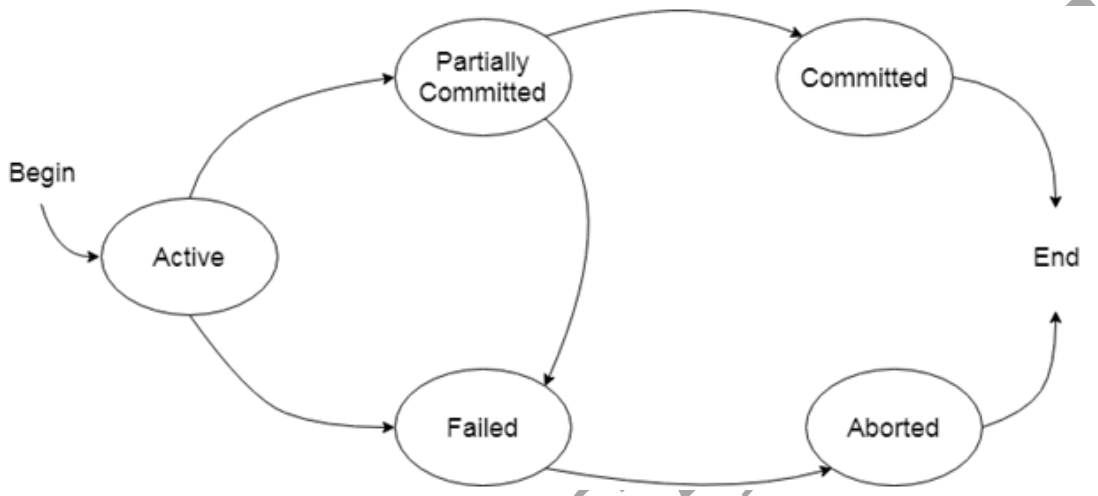
Durability

- The durability property is used to indicate the performance of the database's consistent state. It states that the transaction made the permanent changes.
- They cannot be lost by the erroneous operation of a faulty transaction or by the system failure. When a transaction is completed, then the database reaches a state known as the consistent state. That consistent state cannot be lost, even in the event of a system's failure.
- The recovery subsystem of the DBMS has the responsibility of Durability property.

DIWAKAR EDUCATION HUB

States of Transaction

In a database, the transaction can be in one of the following states -



Active state

- The active state is the first state of every transaction. In this state, the transaction is being executed.
- For example: Insertion or deletion or updating a record is done here. But all the records are still not saved to the database.

Partially committed

- In the partially committed state, a transaction executes its final operation, but the data is still not saved to the database.
- In the total mark calculation example, a final display of the total marks step is executed in this state.

Committed

DIWAKAR EDUCATION HUB

A transaction is said to be in a committed state if it executes all its operations successfully. In this state, all the effects are now permanently saved on the database system.

Failed state

- If any of the checks made by the database recovery system fails, then the transaction is said to be in the failed state.
- In the example of total mark calculation, if the database is not able to fire a query to fetch the marks, then the transaction will fail to execute.

Aborted

- If any of the checks fail and the transaction has reached a failed state then the database recovery system will make sure that the database is in its previous consistent state. If not then it will abort or roll back the transaction to bring the database into a consistent state.
- If the transaction fails in the middle of the transaction then before executing the transaction, all the executed transactions are rolled back to its consistent state.
- After aborting the transaction, the database recovery module will select one of the two operations:

  1. Re-start the transaction
  2. Kill the transaction

Desirable Properties of Transactions

Any transaction must maintain the ACID properties, viz. Atomicity, Consistency, Isolation, and Durability.

- **Atomicity** − This property states that a transaction is an atomic unit of processing, that is, either it is performed in its entirety or not performed at all. No partial update should exist.

DIWAKAR EDUCATION HUB

- **Consistency** − A transaction should take the database from one consistent state to another consistent state. It should not adversely affect any data item in the database.
- **Isolation** − A transaction should be executed as if it is the only one in the system. There should not be any interference from the other concurrent transactions that are simultaneously running.
- **Durability** − If a committed transaction brings about a change, that change should be durable in the database and not lost in case of any failure.

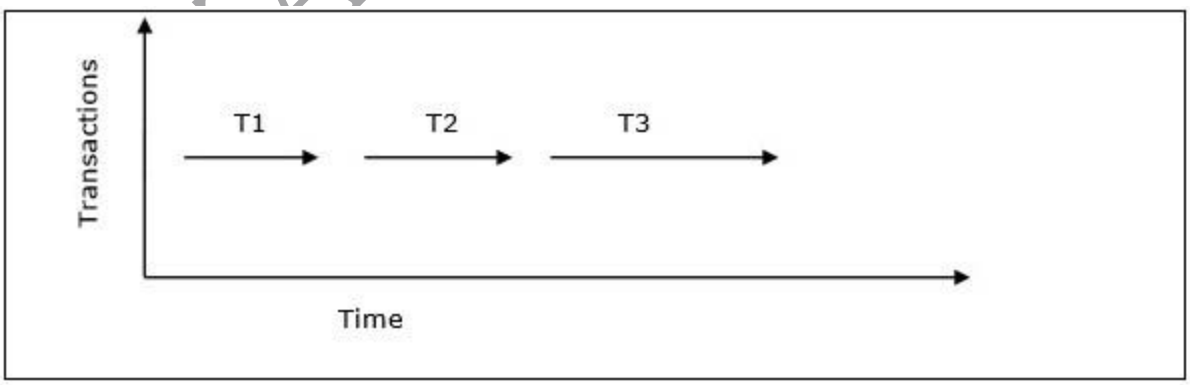
Schedules and Conflicts

In a system with a number of simultaneous transactions, a **schedule** is the total order of execution of operations. Given a schedule S comprising of n transactions, say T1, T2, T3…….Tn; for any transaction Ti, the operations in Ti must execute as laid down in the schedule S.

Types of Schedules

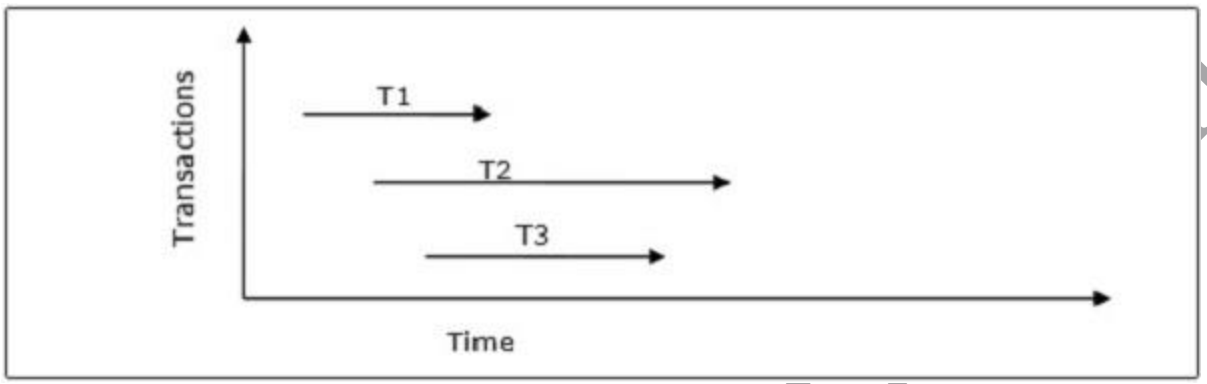
There are two types of schedules −

- **Serial Schedules** − In a serial schedule, at any point of time, only one transaction is active, i.e. there is no overlapping of transactions. This is depicted in the following graph −



DIWAKAR EDUCATION HUB

- **Parallel Schedules** − In parallel schedules, more than one transactions are active simultaneously, i.e. the transactions contain operations that overlap at time. This is depicted in the following graph −



Conflicts in Schedules

In a schedule comprising of multiple transactions, a **conflict** occurs when two active transactions perform non-compatible operations. Two operations are said to be in conflict, when all of the following three conditions exists simultaneously −

- The two operations are parts of different transactions.
- Both the operations access the same data item.
- At least one of the operations is a write_item() operation, i.e. it tries to modify the data item.

Serializability

A **serializable schedule** of 'n' transactions is a parallel schedule which is equivalent to a serial schedule comprising of the same 'n' transactions. A serializable schedule contains the correctness of serial schedule while ascertaining better CPU utilization of parallel schedule.

Equivalence of Schedules

Equivalence of two schedules can be of the following types −

DIWAKAR EDUCATION HUB

- **Result equivalence** − Two schedules producing identical results are said to be result equivalent.
- **View equivalence** − Two schedules that perform similar action in a similar manner are said to be view equivalent.
- **Conflict equivalence** − Two schedules are said to be conflict equivalent if both contain the same set of transactions and has the same order of conflicting pairs of operations.

Concurrency Control

- In the concurrency control, the multiple transactions can be executed simultaneously.
- It may affect the transaction result. It is highly important to maintain the order of execution of those transactions.

Problems of concurrency control

Several problems can occur when concurrent transactions are executed in an uncontrolled manner. Following are the three problems in concurrency control.

1. Lost updates
2. Dirty read
3. Unrepeatable read

1. Lost update problem

- When two transactions that access the same database items contain their operations in a way that makes the value of some database item incorrect, then the lost update problem occurs.
- If two transactions T1 and T2 read a record and then update it, then the effect of updating of the first record will be overwritten by the second update.

DIWAKAR EDUCATION HUB

Example:

Transaction-X	Time	Transaction-Y
——	t1	——
Read A	t2	——
——	t3	Read A
Update A	t4	——
——	t5	Update A
——	t6	——

Here,

- At time t2, transaction-X reads A's value.
- At time t3, Transaction-Y reads A's value.
- At time t4, Transactions-X writes A's value on the basis of the value seen at time t2.
- At time t5, Transactions-Y writes A's value on the basis of the value seen at time t3.
- So at time T5, the update of Transaction-X is lost because Transaction y overwrites it without looking at its current value.
- Such type of problem is known as Lost Update Problem as update made by one transaction is lost here.

2. Dirty Read

- The dirty read occurs in the case when one transaction updates an item of the database, and then the transaction fails for some reason. The updated database item is accessed by another transaction before it is changed back to the original value.
- A transaction T1 updates a record which is read by T2. If T1 aborts then T2 now has values which have never formed part of the stable database.

DIWAKAR EDUCATION

Example:

Transaction-X	Time	Transaction-Y
——	t1	——
——	t2	Update A
Read A	t3	——
——	t4	Rollback
——	t5	——

- At time t2, transaction-Y writes A's value.
- At time t3, Transaction-X reads A's value.
- At time t4, Transactions-Y rollbacks. So, it changes A's value back to that of prior to t1.
- So, Transaction-X now contains a value which has never become part of the stable database.
- Such type of problem is known as Dirty Read Problem, as one transaction reads a dirty value which has not been committed.

3. Inconsistent Retrievals Problem

- Inconsistent Retrievals Problem is also known as unrepeatable read. When a transaction calculates some summary function over a set of data while the other transactions are updating the data, then the Inconsistent Retrievals Problem occurs.
- A transaction T1 reads a record and then does some other processing during which the transaction T2 updates the record. Now when the transaction T1 reads the record, then the new value will be inconsistent with the previous value.

Example:

Suppose two transactions operate on three accounts.

DIWAKAR EDUCATION

Account-1	Account-2	Account-3
Balance = 200	Balance = 250	Balance = 150

Transaction-X	Time	Transaction-Y
——	t1	——
Read Balance of Acc-1 sum <-- 200 Read Balance of Acc-2	t2	——
Sum <-- Sum + 250 = 450	t3	——
——	t4	Read Balance of Acc-3
——	t5	Update Balance of Acc-3 150 --> 150 - 50 --> 100
——	t6	Read Balance of Acc-1
——	t7	Update Balance of Acc-1 200 --> 200 + 50 --> 250
Read Balance of Acc-3	t8	COMMIT
Sum <-- Sum + 250 = 550	t9	——

- Transaction-X is doing the sum of all balance while transaction-Y is transferring an amount 50 from Account-1 to Account-3.
- Here, transaction-X produces the result of 550 which is incorrect. If we write this produced result in the database, the database will become an inconsistent state because the actual sum is 600.
- Here, transaction-X has seen an inconsistent state of the database.

Concurrency Control Protocol

DIWAKAR

Concurrency control protocols ensure atomicity, isolation, and serializability of concurrent transactions. The concurrency control protocol can be divided into three categories:

1. Lock based protocol
2. Time-stamp protocol
3. Validation based protocol

Query Processing in DBMS

Query Processing is the activity performed in extracting data from the database. In query processing, it takes various steps for fetching the data from the database. The steps involved are:

1. Parsing and translation
2. Optimization
3. Evaluation

The query processing works in the following way:

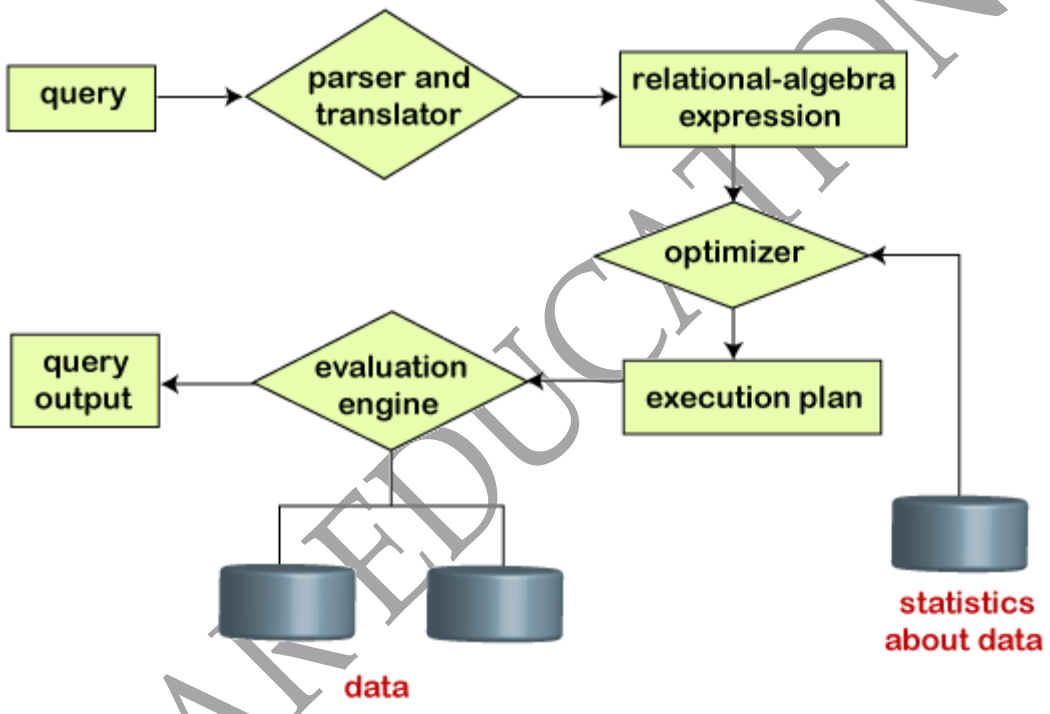
Parsing and Translation

As query processing includes certain activities for data retrieval. Initially, the given user queries get translated in high-level database languages such as SQL. It gets translated into expressions that can be further used at the physical level of the file system. After this, the actual evaluation of the queries and a variety of query - optimizing transformations and takes place. Thus before processing a query, a computer system needs to translate the query into a human-readable and understandable language. Consequently, SQL or Structured Query Language is the best suitable choice for humans. But, it is not perfectly suitable for the internal representation of the query to the system. Relational algebra is well suited for the internal representation of a query. The translation process in query processing is similar to the parser of a query. When a user executes any query, for generating

DIWAKAR EDUCATION HUB

the internal form of the query, the parser in the system checks the syntax of the query, verifies the name of the relation in the database, the tuple, and finally the required attribute value. The parser creates a tree of the query, known as 'parse-tree.' Further, translate it into the form of relational algebra. With this, it evenly replaces all the use of the views when used in the query.

Thus, we can understand the working of a query processing in the below-described diagram:



Steps in query processing

Suppose a user executes a query. As we have learned that there are various methods of extracting the data from the database. In SQL, a user wants to fetch the records of the employees whose salary is greater than or equal to 10000. For doing this, the following query is undertaken:

select emp_name from Employee where salary>10000;

DIWAKAR EDUCATION HUB

Thus, to make the system understand the user query, it needs to be translated in the form of relational algebra. We can bring this query in the relational algebra form as:

- $\sigma_{salary>10000}$ (π_{salary} (Employee))
- π_{salary} ($\sigma_{salary>10000}$ (Employee))

After translating the given query, we can execute each relational algebra operation by using different algorithms. So, in this way, a query processing begins its working.

Evaluation

For this, with addition to the relational algebra translation, it is required to annotate the translated relational algebra expression with the instructions used for specifying and evaluating each operation. Thus, after translating the user query, the system executes a query evaluation plan.

Query Evaluation Plan

- In order to fully evaluate a query, the system needs to construct a query evaluation plan.
- The annotations in the evaluation plan may refer to the algorithms to be used for the particular index or the specific operations.
- Such relational algebra with annotations is referred to as **Evaluation Primitives**. The evaluation primitives carry the instructions needed for the evaluation of the operation.
- Thus, a query evaluation plan defines a sequence of primitive operations used for evaluating a query. The query evaluation plan is also referred to as **the query execution plan**.
- A **query execution engine** is responsible for generating the output of the given query. It takes the query execution plan, executes it, and finally makes the output for the user query.

DIWAKAR EDUCATION HUB

Optimization

- The cost of the query evaluation can vary for different types of queries. Although the system is responsible for constructing the evaluation plan, the user does need not to write their query efficiently.
- Usually, a database system generates an efficient query evaluation plan, which minimizes its cost. This type of task performed by the database system and is known as Query Optimization.
- For optimizing a query, the query optimizer should have an estimated cost analysis of each operation. It is because the overall operation cost depends on the memory allocations to several operations, execution costs, and so on.

Finally, after selecting an evaluation plan, the system evaluates the query and produces the output of the query.

Steps for Query Optimization

Query optimization involves three steps, namely query tree generation, plan generation, and query plan code generation.

Step 1 – Query Tree Generation

A query tree is a tree data structure representing a relational algebra expression. The tables of the query are represented as leaf nodes. The relational algebra operations are represented as the internal nodes. The root represents the query as a whole.

During execution, an internal node is executed whenever its operand tables are available. The node is then replaced by the result table. This process continues for all internal nodes until the root node is executed and replaced by the result table.

For example, let us consider the following schemas −

EMPLOYEE

DIWAKAR EDUCATION HUB

EmpID	EName	Salary	DeptNo

DEPARTMENT

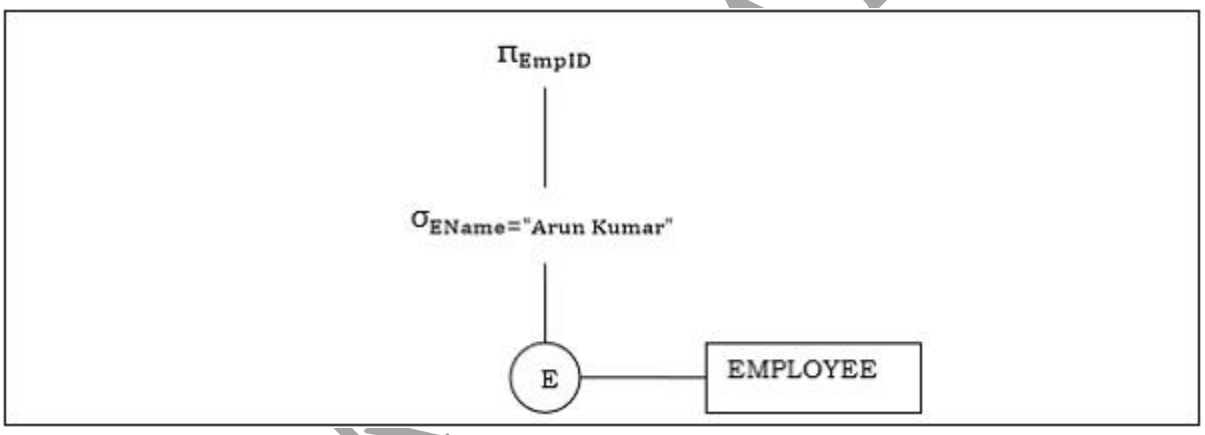
DNo	DName	L

Example 1

Let us consider the query as the following.

$$\pi_{EmpID} (\sigma_{EName = \small "ArunKumar"} {(EMPLOYEE)})$$

The corresponding query tree will be –



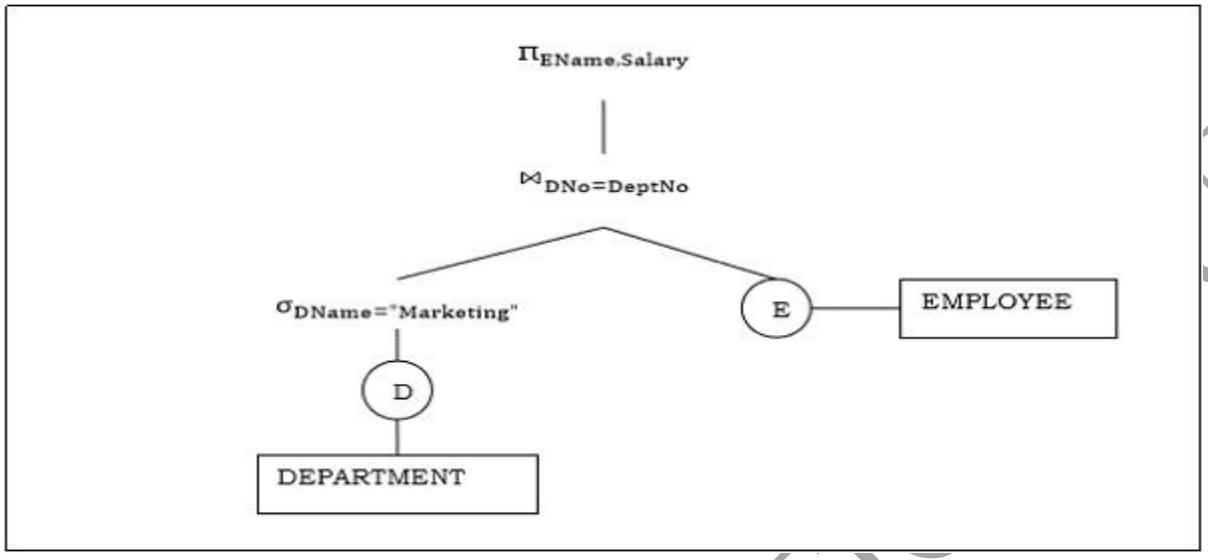
Example 2

consider another query involving a join.

$\pi_{EName, Salary} (\sigma_{DName = \small "Marketing"} {(DEPARTMENT)}) \bowtie_{DNo=DeptNo}{(EMPLOYEE)}$

Following is the query tree for the above query.

DIWAKAR EDUCATION HUB



Step 2 – Query Plan Generation

After the query tree is generated, a query plan is made. A query plan is an extended query tree that includes access paths for all operations in the query tree. Access paths specify how the relational operations in the tree should be performed. For example, a selection operation can have an access path that gives details about the use of B+ tree index for selection.

Besides, a query plan also states how the intermediate tables should be passed from one operator to the next, how temporary tables should be used and how operations should be pipelined/combined.

Step 3– Code Generation

Code generation is the final step in query optimization. It is the executable form of the query, whose form depends upon the type of the underlying operating system. Once the query code is generated, the Execution Manager runs it and produces the results.

Approaches to Query Optimization

Among the approaches for query optimization, exhaustive search and heuristics-based algorithms are mostly used.

DIWAKAR EDUCATION HUB

Exhaustive Search Optimization

In these techniques, for a query, all possible query plans are initially generated and then the best plan is selected. Though these techniques provide the best solution, it has an exponential time and space complexity owing to the large solution space. For example, dynamic programming technique.

Heuristic Based Optimization

Heuristic based optimization uses rule-based optimization approaches for query optimization. These algorithms have polynomial time and space complexity, which is lower than the exponential complexity of exhaustive search-based algorithms. However, these algorithms do not necessarily produce the best query plan.

Some of the common heuristic rules are −

- Perform select and project operations before join operations. This is done by moving the select and project operations down the query tree. This reduces the number of tuples available for join.
- Perform the most restrictive select/project operations at first before the other operations.
- Avoid cross-product operation since they result in very large-sized intermediate tables.

Database Recovery Techniques

Crash Recovery

DBMS is a highly complex system with hundreds of transactions being executed every second. The durability and robustness of a DBMS depends on its complex architecture and its underlying hardware and system software. If it fails or crashes amid transactions, it is expected that the system would follow some sort of algorithm or techniques to recover lost data.

DIWAKAR EDUCATION HUB

Failure Classification

To see where the problem has occurred, we generalize a failure into various categories, as follows −

Transaction failure

A transaction has to abort when it fails to execute or when it reaches a point from where it can't go any further. This is called transaction failure where only a few transactions or processes are hurt.

Reasons for a transaction failure could be –

- **Logical errors** − Where a transaction cannot complete because it has some code error or any internal error condition.
- **System errors** − Where the database system itself terminates an active transaction because the DBMS is not able to execute it, or it has to stop because of some system condition. For example, in case of deadlock or resource unavailability, the system aborts an active transaction.

System Crash

There are problems − external to the system − that may cause the system to stop abruptly and cause the system to crash. For example, interruptions in power supply may cause the failure of underlying hardware or software failure.

Examples may include operating system errors.

Disk Failure

In early days of technology evolution, it was a common problem where hard-disk drives or storage drives used to fail frequently.

Disk failures include formation of bad sectors, unreachability to the disk, disk head crash or any other failure, which destroys all or a part of disk storage.

Storage Structure

DIWAKAR EDUCATION HUB

We have already described the storage system. In brief, the storage structure can be divided into two categories −

- **Volatile storage** − As the name suggests, a volatile storage cannot survive system crashes. Volatile storage devices are placed very close to the CPU; normally they are embedded onto the chipset itself. For example, main memory and cache memory are examples of volatile storage. They are fast but can store only a small amount of information.
- **Non-volatile storage** − These memories are made to survive system crashes. They are huge in data storage capacity, but slower in accessibility. Examples may include hard-disks, magnetic tapes, flash memory, and non-volatile (battery backed up) RAM.

Recovery and Atomicity

When a system crashes, it may have several transactions being executed and various files opened for them to modify the data items. Transactions are made of various operations, which are atomic in nature. But according to ACID properties of DBMS, atomicity of transactions as a whole must be maintained, that is, either all the operations are executed or none.

When a DBMS recovers from a crash, it should maintain the following −

- It should check the states of all the transactions, which were being executed.
- A transaction may be in the middle of some operation; the DBMS must ensure the atomicity of the transaction in this case.
- It should check whether the transaction can be completed now or it needs to be rolled back.
- No transactions would be allowed to leave the DBMS in an inconsistent state.

DIWAKAR EDUCATION HUB

There are two types of techniques, which can help a DBMS in recovering as well as maintaining the atomicity of a transaction −

- Maintaining the logs of each transaction, and writing them onto some stable storage before actually modifying the database.
- Maintaining shadow paging, where the changes are done on a volatile memory, and later, the actual database is updated.

Log-based Recovery

Log is a sequence of records, which maintains the records of actions performed by a transaction. It is important that the logs are written prior to the actual modification and stored on a stable storage media, which is failsafe.

Log-based recovery works as follows −

- The log file is kept on a stable storage media.
- When a transaction enters the system and starts execution, it writes a log about it.

$$<T_n, \text{Start}>$$

- When the transaction modifies an item X, it write logs as follows −

$$<T_n, X, V_1, V_2>$$

It reads T_n has changed the value of X, from V_1 to V_2.

- When the transaction finishes, it logs −

$$<T_n, \text{commit}>$$

The database can be modified using two approaches −

- **Deferred database modification** − All logs are written on to the stable storage and the database is updated when a transaction commits.

DIWAKAR EDUCATION HUB

- **Immediate database modification** − Each log follows an actual database modification. That is, the database is modified immediately after every operation.

Recovery with Concurrent Transactions

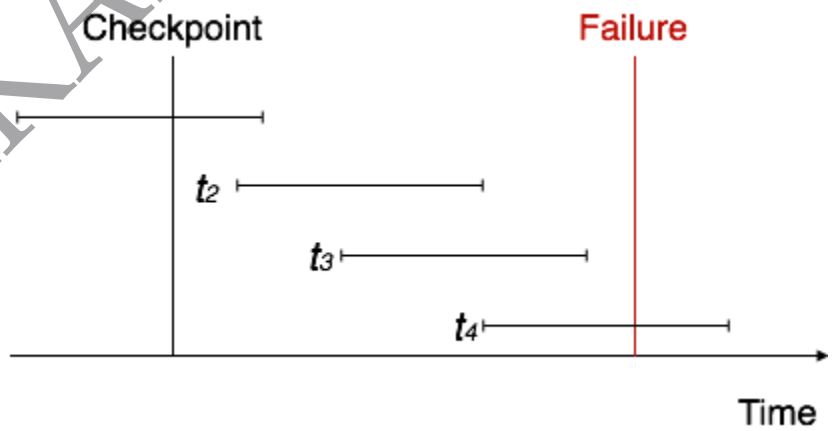
When more than one transaction are being executed in parallel, the logs are interleaved. At the time of recovery, it would become hard for the recovery system to backtrack all logs, and then start recovering. To ease this situation, most modern DBMS use the concept of 'checkpoints'.

Checkpoint

Keeping and maintaining logs in real time and in real environment may fill out all the memory space available in the system. As time passes, the log file may grow too big to be handled at all. Checkpoint is a mechanism where all the previous logs are removed from the system and stored permanently in a storage disk. Checkpoint declares a point before which the DBMS was in consistent state, and all the transactions were committed.

Recovery

When a system with concurrent transactions crashes and recovers, it behaves in the following manner −



DIWAKAR EDUCATION HUB

- The recovery system reads the logs backwards from the end to the last checkpoint.
- It maintains two lists, an undo-list and a redo-list.
- If the recovery system sees a log with <T_n, Start> and <T_n, Commit> or just <T_n, Commit>, it puts the transaction in the redo-list.
- If the recovery system sees a log with <T_n, Start> but no commit or abort log found, it puts the transaction in undo-list.

All the transactions in the undo-list are then undone and their logs are removed. All the transactions in the redo-list and their previous logs are removed and then redone before saving their logs.

Object and Object-Relational Databases

Object-Relational Database (ORD)

An object-relational database (ORD) is a database management system (DBMS) that's composed of both a relational database (RDBMS) and an object-oriented database (OODBMS). ORD supports the basic components of any object-oriented database model in its schemas and the query language used, such as objects, classes and inheritance.

An object-relational database may also be known as an object relational database management systems (ORDBMS).

ORD is said to be the middleman between relational and object-oriented databases because it contains aspects and characteristics from both models. In ORD, the basic approach is based on RDB, since the data is stored in a traditional database and manipulated and accessed using queries written in a query language like SQL. However, ORD also showcases an object-oriented characteristic in that the database is considered an object store, usually for software that is written in an object-oriented programming language. Here, APIs are used to store and access the data as objects.

DIWAKAR EDUCATION HUB

One of ORD's aims is to bridge the gap between conceptual data modeling techniques for relational and object-oriented databases like the entity-relationship diagram (ERD) and object-relational mapping (ORM). It also aims to connect the divide between relational databases and the object-oriented modeling techniques that are usually used in programming languages like Java, C# and C++.

Traditional RDBMS products concentrate on the efficient organization of data that is derived from a limited set of data-types. On the other hand, an ORDBMS has a feature that allows developers to build and innovate their own data types and methods, which can be applied to the DBMS. With this, ORDBMS intends to allow developers to increase the abstraction with which they view the problem area.

Database Security

DB2 database and functions can be managed by two different modes of security controls:

1. Authentication
2. Authorization

Authentication

Authentication is the process of confirming that a user logs in only in accordance with the rights to perform the activities he is authorized to perform. User authentication can be performed at operating system level or database level itself. By using authentication tools for biometrics such as retina and figure prints are in use to keep the database from hackers or malicious users.

The database security can be managed from outside the db2 database system. Here are some type of security authentication process:

- Based on Operating System authentications.

DIWAKAR EDUCATION HUB

- Lightweight Directory Access Protocol (LDAP)

For DB2, the security service is a part of operating system as a separate product. For Authentication, it requires two different credentials, those are userid or username, and password.

Authorization

You can access the DB2 Database and its functionality within the DB2 database system, which is managed by the DB2 Database manager. Authorization is a process managed by the DB2 Database manager. The manager obtains information about the current authenticated user, that indicates which database operation the user can perform or access.

Here are different ways of permissions available for authorization:

Primary permission: Grants the authorization ID directly.

Secondary permission: Grants to the groups and roles if the user is a member

Public permission: Grants to all users publicly.

Context-sensitive permission: Grants to the trusted context role.

Authorization can be given to users based on the categories below:

- System-level authorization
- System administrator [SYSADM]
- System Control [SYSCTRL]
- System maintenance [SYSMAINT]
- System monitor [SYSMON]

Authorities provide of control over instance-level functionality. Authority provide to group privileges, to control maintenance and authority operations. For instance, database and database objects.

- Database-level authorization

DIWAKAR EDUCATION HUB

- Security Administrator [SECADM]
- Database Administrator [DBADM]
- Access Control [ACCESSCTRL]
- Data access [DATAACCESS]
- SQL administrator. [SQLADM]
- Workload management administrator [WLMADM]
- Explain [EXPLAIN]

Authorities provide controls within the database. Other authorities for database include with LDAD and CONNECT.

- **Object-Level Authorization**: Object-Level authorization involves verifying privileges when an operation is performed on an object.
- **Content-based Authorization**: User can have read and write access to individual rows and columns on a particular table using Label-based access Control [LBAC].

DB2 tables and configuration files are used to record the permissions associated with authorization names. When a user tries to access the data, the recorded permissions verify the following permissions:

- Authorization name of the user
- Which group belongs to the user
- Which roles are granted directly to the user or indirectly to a group
- Permissions acquired through a trusted context.

While working with the SQL statements, the DB2 authorization model considers the combination of the following permissions:

- Permissions granted to the primary authorization ID associated with the SQL statements.
- Secondary authorization IDs associated with the SQL statements.
- Granted to PUBLIC
- Granted to the trusted context role.

DIWAKAR EDUCATION HUB

Instance level authorities

Some instance related authorities.

System administration authority (SYSADM)

It is highest level administrative authority at the instance-level. Users with SYSADM authority can execute some databases and database manager commands within the instance. Users with SYSADM authority can perform the following operations:

- Upgrade a Database
- Restore a Database
- Update Database manager configuration file.

System control authority (SYSCTRL)

It is the highest level in System control authority. It provides to perform maintenance and utility operations against the database manager instance and its databases. These operations can affect system resources, but they do not allow direct access to data in the database.

Users with SYSCTRL authority can perform the following actions:

- Updating the database, Node, or Distributed Connect Service (DCS) directory
- Forcing users off the system-level
- Creating or Dropping a database-level
- Creating, altering, or dropping a table space
- Using any table space
- Restoring Database

System maintenance authority (SYSMAINT)

It is a second level of system control authority. It provides to perform maintenance and utility operations against the database manager instance and its

DIWAKAR EDUCATION HUB

databases. These operations affect the system resources without allowing direct access to data in the database. This authority is designed for users to maintain databases within a database manager instance that contains sensitive data.

Only Users with SYSMAINT or higher level system authorities can perform the following tasks:

- Taking backup
- Restoring the backup
- Roll forward recovery
- Starting or stopping instance
- Restoring tablespaces
- Executing db2trc command
- Taking system monitor snapshots in case of an Instance level user or a database level user.

A user with SYSMAINT can perform the following tasks:

- Query the state of a tablespace
- Updating log history files
- Reorganizing of tables
- Using RUNSTATS (Collection catalog statistics)

System monitor authority (SYSMON)

With this authority, the user can monitor or take snapshots of database manager instance or its database. SYSMON authority enables the user to run the following tasks:

- GET DATABASE MANAGER MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET SNAPSHOT
- LIST
    - LIST ACTIVE DATABASES
    - LIST APPLICATIONS

DIWAKAR EDUCATION HUB

- LIST DATABASE PARTITION GROUPS
- LIST DCS APPLICATIONS
- LIST PACKAGES
- LIST TABLES
- LIST TABLESPACE CONTAINERS
- LIST TABLESPACES
- LIST UTITLITIES

- RESET MONITOR
- UPDATE MONITOR SWITCHES

Database authorities

Each database authority holds the authorization ID to perform some action on the database. These database authorities are different from privileges. Here is the list of some database authorities:

ACCESSCTRL: allows to grant and revoke all object privileges and database authorities.

BINDADD: Allows to create a new package in the database.

CONNECT: Allows to connect to the database.

CREATETAB: Allows to create new tables in the database.

CREATE_EXTERNAL_ROUTINE: Allows to create a procedure to be used by applications and the users of the databases.

DATAACCESS: Allows to access data stored in the database tables.

DBADM: Act as a database administrator. It gives all other database authorities except ACCESSCTRL, DATAACCESS, and SECADM.

EXPLAIN: Allows to explain query plans without requiring them to hold the privileges to access the data in the tables.

DIWAKAR EDUCATION HUB

IMPLICIT_SCHEMA: Allows a user to create a schema implicitly by creating an object using a CREATE statement.

LOAD: Allows to load data into table.

QUIESCE_CONNECT: Allows to access the database while it is quiesce (temporarily disabled).

SECADM: Allows to act as a security administrator for the database.

SQLADM: Allows to monitor and tune SQL statements.

WLMADM: Allows to act as a workload administrator

Privileges

SETSESSIONUSER

Authorization ID privileges involve actions on authorization IDs. There is only one privilege, called the SETSESSIONUSER privilege. It can be granted to user or a group and it allows to session user to switch identities to any of the authorization IDs on which the privileges are granted. This privilege is granted by user SECADM authority.

Schema privileges

This privileges involve actions on schema in the database. The owner of the schema has all the permissions to manipulate the schema objects like tables, views, indexes, packages, data types, functions, triggers, procedures and aliases. A user, a group, a role, or PUBLIC can be granted any user of the following privileges:

- **CREATEIN**: allows to create objects within the schema
- **ALTERIN**: allows to modify objects within the schema.

DROPIN

This allows to delete the objects within the schema.

DIWAKAR EDUCATION HUB

Table space privileges

These privileges involve actions on the tablespaces in the database. User can be granted the USE privilege for the tablespaces. The privileges then allow them to create tables within tablespaces. The privilege owner can grant the USE privilege with the command WITH GRANT OPTION on the tablespace when tablespace is created. And SECADM or ACCESSCTRL authorities have the permissions to USE privileges on the tablespace.

Table and view privileges

The user must have CONNECT authority on the database to be able to use table and view privileges. The privileges for tables and views are as given below:

CONTROL

It provides all the privileges for a table or a view including drop and grant, revoke individual table privileges to the user.

ALTER

It allows user to modify a table.

DELETE

It allows the user to delete rows from the table or view.

INDEX

It allows the user to insert a row into table or view. It can also run import utility.

REFERENCES

It allows the users to create and drop a foreign key.

SELECT

It allows the user to retrieve rows from a table or view.

DIWAKAR EDUCATION HUB

UPDATE

It allows the user to change entries in a table, view.

Package privileges

User must have CONNECT authority to the database. Package is a database object that contains the information of database manager to access data in the most efficient way for a particular application.

CONTROL

It provides the user with privileges of rebinding, dropping or executing packages. A user with this privileges is granted to BIND and EXECUTE privileges.

BIND

It allows the user to bind or rebind that package.

EXECUTE

Allows to execute a package.

Index privileges

This privilege automatically receives CONTROL privilege on the index.

Sequence privileges

Sequence automatically receives the USAGE and ALTER privileges on the sequence.

Routine privileges

It involves the action of routines such as functions, procedures, and methods within a database.

The enhanced data model offers rich features, but breaks backward compatibility.

DIWAKAR EDUCATION HUB

The classic model is simple, well-understood, and had been around for a long time. The enhanced data model offers many new features for structuring data. Data producers must choose which data model to use.

Reasons to use the classic model:

- Data using the classic model can be read by all existing netCDF software.
- Writing programs for classic model data is easier.
- Most or all existing netCDF conventions are targeted at the classic model.
- Many great features, like compression, parallel I/O, large data sizes, etc., are available within the classic model.

Reasons to use the ehanced model:

- Complex data structures can be represented very easily in the data, leading to easier programming.
- If exisiting HDF5 applications produce or use these data, and depend on user-defined types, unsigned types, strings, or groups, then the enhanced model is required.
- In performance-critical applications, the enhanced model may provide significant benefits.

Temporal Databases

Temporal data strored in a temporal database is different from the data stored in non-temporal database in that a time period attached to the data expresses when it was valid or stored in the database. As mentioned above, conventional databases consider the data stored in it to be valid at time instant now, they do not keep track of past or future database states. By attaching a time period to the data, it becomes possible to store different database states.
A first step towards a temporal database thus is to timestamp the data. This allows the distinction of different database states. One approach is that a temporal database may timestamp entities with time periods. Another approach is the timestamping of the property values of the entities. In the relational data

DIWAKAR EDUCATION HUB

model, tuples are timestamped, where as in object-oriented data models, objects and/or attribute values may be timestamped.

What time period do we store in these timestamps? As we mentioned already, there are mainly two different notions of time which are relevant for temporal databases. One is called the **valid time**, the other one is the **transaction time**. Valid time denotes the time period during which a fact is true with respect to the real world. Transaction time is the time period during which a fact is stored in the database. Note that these two time periods do not have to be the same for a single fact. Imagine that we come up with a temporal database storing data about the 18th century. The valid time of these facts is somewhere between 1700 and 1799, where as the transaction time starts when we insert the facts into the database, for example, January 21, 1998.

Assume we would like to store data about our employees with respect to the real world. Then, the following table could result:

EmpID	Name	Department	Salary	ValidTimeStart	ValidTimeEnd
10	John	Research	11000	1985	1990
10	John	Sales	11000	1990	1993
10	John	Sales	12000	1993	INF
11	Paul	Research	10000	1988	1995
12	George	Research	10500	1991	INF
13	Ringo	Sales	15500	1988	INF

The above valid-time table stores the history of the employees with respect to the real world. The attributes **ValidTimeStart** and **ValidTimeEnd** actually represent a time interval which is closed at its lower and open at its upper bound. Thus, we see that during the time period [1985 - 1990), employee John was working in the

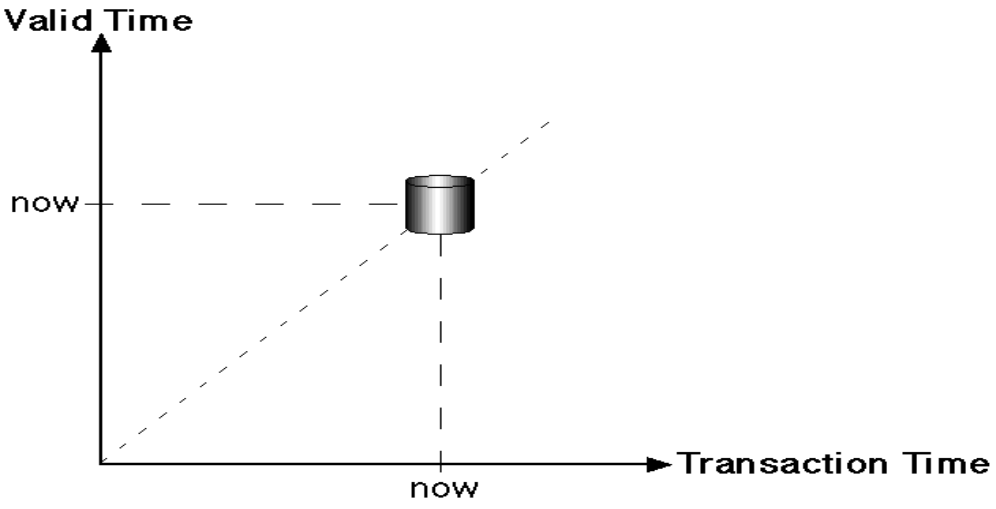
DIWAKAR EDUCATION HUB

research department, having a salary of 11000. Then he changed to the sales department, still earning 11000. In 1993, he got a salary raise to 12000. The upper bound INF denotes that the tuple is valid until further notice. Note that it is now possible to store information about past states. We see that Paul was employed from 1988 until 1995. In the corresponding non-temporal table, this information was (physically) deleted when Paul left the company.

Different Forms of Temporal Databases

The two different notions of time - valid time and transaction time - allow the distinction of different forms of temporal databases. A **historical database** stores data with respect to valid time, a **rollback database** stores data with respect to transaction time. A **bitemporal database** stores data with respect to both valid time and transaction time.

As mentioned above, commercial DBMS are said to store only a single state of the real world, usually the most recent state. Such databases usually are called **snapshot databases**. A snapshot database in the context of valid time and transaction time is depicted in the following picture:



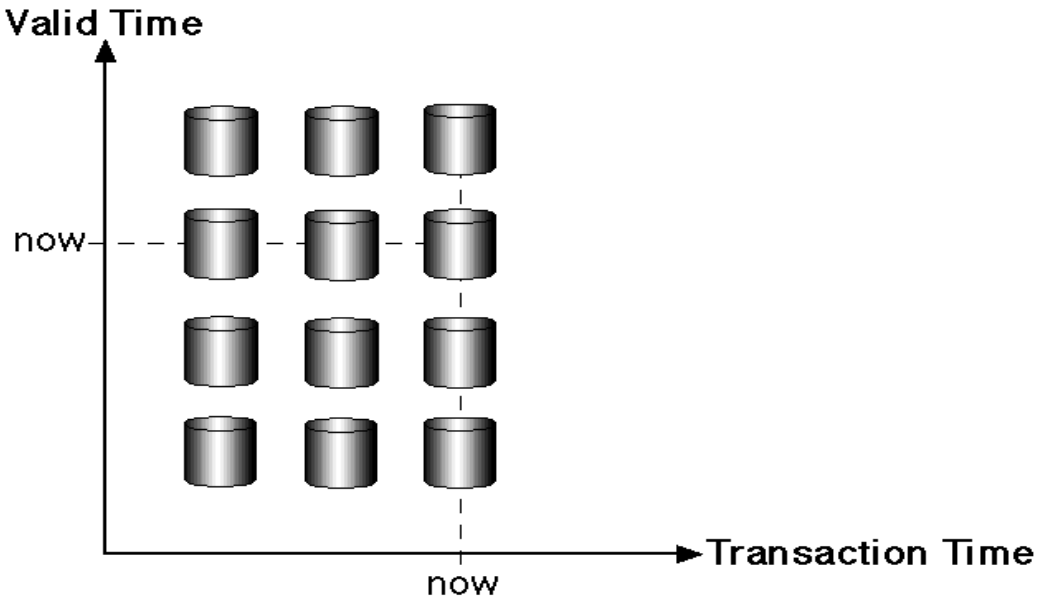
On the other hand, a **bitemporal DBMS** such as TimeDB stores the history of data with respect to both valid time and transaction time. Note that the history of

DIW EDUCATION HUB

when data was stored in the database (transaction time) is limited to past and present database states, since it is managed by the system directly which does not know anything about future states.

A table in the bitemporal relational DBMS TimeDB may either be a snapshot table (storing only current data), a valid-time table (storing when the data is valid wrt. the real world), a transaction-time table (storing when the data was recorded in the database) or a bitemporal table (storing both valid time and transaction time). An extended version of SQL allows to specify which kind of table is needed when the table is created. Existing tables may also be altered (schema versioning). Additionally, it supports **temporal queries**, **temporal modification statements** and **temporal constraints**.

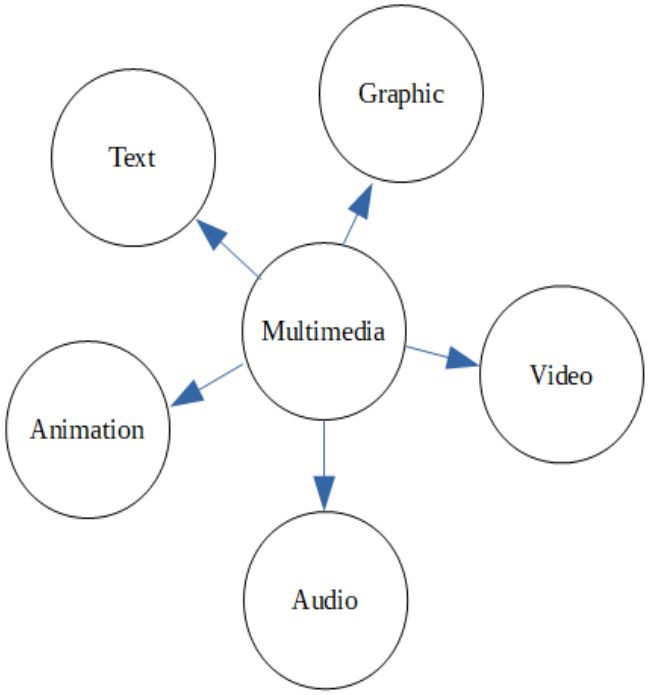
The states stored in a bitemporal database are sketched in the picture below. Of course, a temporal DBMS such as TimeDB does not store each database state separately as depicted in the picture below. It stores valid time and/or transaction time for each tuple, as described above.



DIW EDUCATION HUB

Multimedia Databases

The multimedia databases are used to store multimedia data such as images, animation, audio, video along with text. This data is stored in the form of multiple file types like .txt(text), .jpg(images), .swf(videos), .mp3(audio) etc.



Contents of the Multimedia Database

The multimedia database stored the multimedia data and information related to it. This is given in detail as follows –

Media data

This is the multimedia data that is stored in the database such as images, videos, audios, animation etc.

Media format data

The Media format data contains the formatting information related to the media data such as sampling rate, frame rate, encoding scheme etc.

Media keyword data

This contains the keyword data related to the media in the database. For an image the keyword data can be date and time of the image, description of the image etc.

Media feature data

Th Media feature data describes the features of the media data. For an image, feature data can be colours of the image, textures in the image etc.

Challenges of Multimedia Database

There are many challenges to implement a multimedia database. Some of these are:

- Multimedia databases contains data in a large type of formats such as .txt(text), .jpg(images), .swf(videos), .mp3(audio) etc. It is difficult to convert one type of data format to another.
- The multimedia database requires a large size as the multimedia data is quite large and needs to be stored successfully in the database.
- It takes a lot of time to process multimedia data so multimedia database is slow.

Mobile Databases

Mobile databases are separate from the main database and can easily be transported to various places. Even though they are not connected to the main database, they can still communicate with the database to share and exchange data.

The mobile database includes the following components −

- The main system database that stores all the data and is linked to the mobile database.
- The mobile database that allows users to view information even while on the move. It shares information with the main database.

DIWAKAR EDUCATION HUB

- The device that uses the mobile database to access data. This device can be a mobile phone, laptop etc.
- A communication link that allows the transfer of data between the mobile database and the main database.

Advantages of Mobile Databases

Some advantages of mobile databases are −

- The data in a database can be accessed from anywhere using a mobile database. It provides wireless database access.
- The database systems are synchronized using mobile databases and multiple users can access the data with seamless delivery process.
- Mobile databases require very little support and maintenance.
- The mobile database can be synchronized with multiple devices such as mobiles, computer devices, laptops etc.

Disadvantages of Mobile Databases

Some disadvantages of mobile databases are −

- The mobile data is less secure than data that is stored in a conventional stationary database. This presents a security hazard.
- The mobile unit that houses a mobile database may frequently lose power because of limited battery. This should not lead to loss of data in database.

Deductive Database

A **deductive database** is a database system that makes conclusions about its data based on a set of well-defined rules and facts. This type of database was developed to combine logic programming with relational database management systems. Usually, the language used to define the rules and facts is the logical programming language Datalog.

DIWAKAR EDUCATION HUB

A **Deductive Database** is a type of database that can make conclusions or we can say deductions using a sets of well defined rules and fact that are stored in the database. In today's world as we deal with a large amount of data, this deductive database provides a lot of advantages. It helps to combine the *RDBMS* with logic programming. To design a deductive database a purely declarative programming language called Datalog is used.

The implementations of deductive databases can be seen in LDL (Logic Data Language), NAIL (Not Another Implementation of Logic), CORAL, and VALIDITY. The use of LDL and VALIDITY in a variety of business/industrial applications are as follows.

1. LDL Applications:
This system has been applied to the following application domains:

- **Enterprise modeling:**
  Data related to an enterprise may result in an extended ER model containing hundreds of entities and relationship and thousands of attributes.This domain involves modeling the structure, processes, and constraints within an enterprise.

- **Hypothesis testing or data dredging:**
  This domain involves formulating a hypothesis, translating in into an LDL rule set and a query, and then executing the query against given data to test the hypothesis. This has been applied to genome data analysis in the field of microbiology, where data dredging consists of identifying the DNA sequences from low-level digitized auto radio graphs from experiments performed on E.Coli Bacteria.

- **Software reuse:**
  A small fraction of the software for an application is rule-based and encoded in LDL (bulk is developed in standard procedural code). The rules give rise to a knowledge base that contains, *A definition of each C module used in system*and *A set of rules that defines ways in which modules can*

DIWAKAR EDUCATION HUB

export/import functions, constraints and so on. The "Knowledge base" can be used to make decisions that pertain to the reuse of software subsets. This is being experimented within banking software.

2. VALIDITY Applications:

Validity combines deductive capabilities with the ability to manipulate complex objects (OIDs, inheritance, methods, etc). It provides a DOOD data model and language called DEL (Datalog Extended Language), an engine working along a client-server model and a set of tools for schema and rule editing, validation, and querying.

The following are some application areas of the VALIDITY system:

- **Electronic commerce:**
  In electronic commerce, complex customers profiles have to be matched against target descriptions. The matching process is also described by rules, and computed predicates deal with numeric computations. The declarative nature of DEl makes the formulation of the matching algorithm easy.

- **Rules-governed processes:**
  In a rules-governed process, well defined rules define the actions to be performed. In those process some classes are modeled as DEL classes. The main advantage of VALIDITY is the ease with which new regulations are taken into account.

- **Knowledge discovery:**
  The goal of knowledge discovery is to find new data relationships by analyzing existing data. An application prototype developed by University of Illinois utilizes already existing minority student data that has been enhanced with rules in DEL.

- **Concurrent Engineering:**
  A concurrent engineering applications deals with large amounts of centralized data, shared by several participants. An application prototype has been developed in the area of civil engineering. The design data is

DIWAKAR EDUCATION HUB

modeled using the object-oriented power of the DEL language. DEL is able to handle transformation of rules into constraints, and it can also handle any closed formula as an integrity constraint.

XML - Databases

XML Database is used to store huge amount of information in the XML format. As the use of XML is increasing in every field, it is required to have a secured place to store the XML documents. The data stored in the database can be queried using **XQuery**, serialized, and exported into a desired format.

XML Database Types

There are two major types of XML databases −

- XML- enabled
- Native XML (NXD)

XML - Enabled Database

XML enabled database is nothing but the extension provided for the conversion of XML document. This is a relational database, where data is stored in tables consisting of rows and columns. The tables contain set of records, which in turn consist of fields.

Native XML Database

Native XML database is based on the container rather than table format. It can store large amount of XML document and data. Native XML database is queried by the **XPath**-expressions.

Native XML database has an advantage over the XML-enabled database. It is highly capable to store, query and maintain the XML document than XML-enabled database.

Example

Following example demonstrates XML database −

```
<?xml version = "1.0"?>
<contact-info>
```


DIWAKAR EDUCATION HUB

```
  <contact1>
    <name>Tanmay Patil</name>
    <company>TutorialsPoint</company>
    <phone>(011) 123-4567</phone>
  </contact1>

  <contact2>
    <name>Manisha Patil</name>
    <company>TutorialsPoint</company>
    <phone>(011) 789-4567</phone>
  </contact2>
</contact-info>
```

Here, a table of contacts is created that holds the records of contacts (contact1 and contact2), which in turn consists of three entities – *name, company* and *phone*.

Internet Database Applications

Internet Database Applications are programs that are built to run on Internet browsers and communicate with database servers. Internet Database Applications are usually developed using very few graphics and are built using XHTML forms and Style Sheets.

Most companies are starting to migrate from the old fashioned desktop database applications to web based Internet Database Applications in XHTML format.

Below are some of the benefits of Internet Database Applications:

- **Powerful and Scalable** - Internet Database Applications are more robust, agile and able to expand and scale up more easily.
  Database servers that are built to serve Internet applications are designed to handle millions of concurrent connections and complex SQL queries.
  A good example is Facebook, which uses database servers that are able to handle millions of inquiries and complex SQL queries.
  Internet database applications use the same type of database server that is

DIWAKAR EDUCATION HUB

designed to run Facebook. The database servers that are built to serve desktop applications usually can handle only a limited number of connections and are not able to deal with complex SQL queries.

- **Web Based** - Internet Database Applications are web based applications, therefore the data can be accessed using a browser at any location.
- **Security** - Database servers have been fortified with preventive features and security protocols have been implemented to combat today's cyber security threats and vulnerabilities.
- **Open Source**, Better Licensing Terms and Cost Savings - There are many powerful database servers that are open source. This means that there is no licensing cost. Many large enterprise sites are using Open Source Database Servers, such as Facebook, Yahoo, YouTube, Flickr, Wikipedia, etc. . Open Source also creates less dependence on vendors, which is a big advantage because that provides more product quality control and lower cost. Open source also offers easier customization and is experiencing a fast growing adoption rate, especially by the large and influential enterprises.
- **Abundant Features** - There are many open source programming languages (such as PHP, Python, Ruby) and hundreds of powerful open source libraries, tools and plug-ins specifically built to interact with today's database servers.

Geographical information system (GIS)

Geographical information system (GIS) is basically defined as a systematic integration of hardware and software for capturing, storing, displaying, updating manipulating and analyzing spatial data. GIS can also be viewed as an interdisciplinary area that incorporates many distinct fields of study such as:

1. Geodesy that is basically projection,

surveying, cartography and so on.

2. Remote Sensing

DIWAKAR EDUCATION HUB

3. Photogrammetry

4. Environmental Science

5. City Planning

6. Cognitive Science

As a result GIS relies on progress made in fields such as computer science, databases, statistics, and artificial intelligence. All the different problems and question that arises from the integration of multiple disciplines make a more than a simple tool.

Requirements for GIS –
Geographic Information requires a means of integration between different sources of data at different level of accuracy. System basically deals with the aspects of daily life, so it must be updated daily to keep it current and reliable. Much of the Information Stored in GIS are for practical use requires a special means of retrieval and manipulation.

GIS system and application basically deals with information that can be viewed as data with specific meaning and context rather than simple data.

Components of GIS system –
GIS system can be viewed as an integration of three components are hardware and software, data, people. Lets discuss them one by one:

1. **Hardware and software –**
   Hardware relates to device used by end users such as graphic devices or plotters and scanners. Data storage and manipulation is done using a range of processor. With the development of the Internet and Web based application, Web servers have become part of many system's architecture, hence most GIS's follows 3-Tier architecture.

DIWAKAR EDUCATION HUB

Software parts relates to the processes used to define, store and manipulate the data and hence it is akin to DBMS. Different models are used to provide efficient means of storage retrieval and manipulation of data.

2. **Data –**
   Geographic data are basically divided into two main groups are vector and raster.

Vector data/layers in GIS refers to discrete objects represented by points, lines and polygons. Lines are formed by connecting two or more points and polygons are closed set of Lines. Layers represent geometries that share a common set of attributes. Objects within a layer have mutual topology. Vector sources include digitized maps, features extracted from image surveys and many more.

Raster data is a continuous grid of cells in two dimension or the equivalent of cubic cells in three dimension. Raster data are divided conceptually into categorical and continuous. In a categorical raster every cell value is linked to a category in a separate table.Examples Soil type, vegetation types.land suitability, and so on. Continuous raster images usually describes continuous phenomena in space such as Digital Elevation Model where each pixel is an elevation value. Unlike categorical raster, a continuous raster doesn't have an attribute/category table attached. Typical Raster sources are aerial images, satellite images and scanned map images.

3. **People –**
   People are involved in all phases of development of a GIS system and in collecting data. They include cartographers and surveyors who create the maps and survey the land and the geographical features. They also include system users who collect the data, upload the data to system, manipulate the system and analyze the results.

Genome Data Management

DIWAKAR EDUCATION HUB

GENOME is a prototype **database management system** (**DBMS**)/user interface system designed to manage complex biological **data**, allowing users to more fully analyze and understand relationships in human **genome data**. The system is designed to allow the establishment of a net- work of searchable **data** sources.

Characteristics of Biological Data (Genome Data Management)

There are many characteristics of biological data. All these characteristics make the management of biological information a particularly challenging problem. Here mainly we will focus on characteristics of biological information and multidisciplinary field called bioinformatics. Bioinformatics, now a days has emerged with graduate degree programs in several universities.

Characteristics of Biological Information:

- There is a high amount and range of variability in data.
  There should be a flexibility in biological systems so that it can handle data types and values. Placing constraints on data types must be limited with such a wide range of possible data values. There can be a loss of information when there is exclusion of such values.
- There will be a difference in representation of the same data by different biologists.
  This can be done even using the same system. There is a multiple ways to model any given entity with the results often reflecting the particular focus of the scientist.
  There should be a linking of data elements in a network of schemas.
- Defining the complex queries and also important to the biologists.
  Complex queries must be supported by biological systems. Knowledge of the data structure is needed for the average users because with the help of this knowledge average user can construct a complex query across data sets on their own. For this systems must provide some tools for building these queries.

DIWAKAR EDUCATION HUB

- When compared with most other domains or applications, biological data becomes highly complex.
  Such data must ensure that no information is lost during biological data modelling and such data must be able to represent a complex substructure of data as well as relationships. An additional context is provided by the structure of the biological data for interpretation of the information.

- There is a rapid change in schemas of biological databases.
  There should be a support of schema evolution and data object migration so that there can be an improved information flow between generations or releases of databases.
  The relational database systems support the ability to extend the schema and a frequent occurrence in the biological setting.

- Most biologists are not likely to have knowledge of internal structure of the database or about schema design.
  Users need an information which can be displayed in a manner such that it can be applicable to the problem which they are trying to address. Also the data structure should be reflected in an easy and understandable manner. An information regarding the meaning of the schema is not provided to the user because of the failure by the relational schemas. A present search interfaces is provided by the web interfaces, which may limit access into the database.

- There is no need of the write access to the database by the users of biological data, instead they only require read access.
  There is limitation of write access to the privileged users called curators. There are only small numbers of users which require write access but a wide variety of read access patterns are generated by the users into the databases.

- Access to “old” values of the data are required by the users of biological data most often while verifying the previously reported results.
  Hence system of archives must support the changes to the values of the

DIWAKAR EDUCATION HUB

data in the database. Access to both the most recent version of data value and its previous version are important in the biological domain.

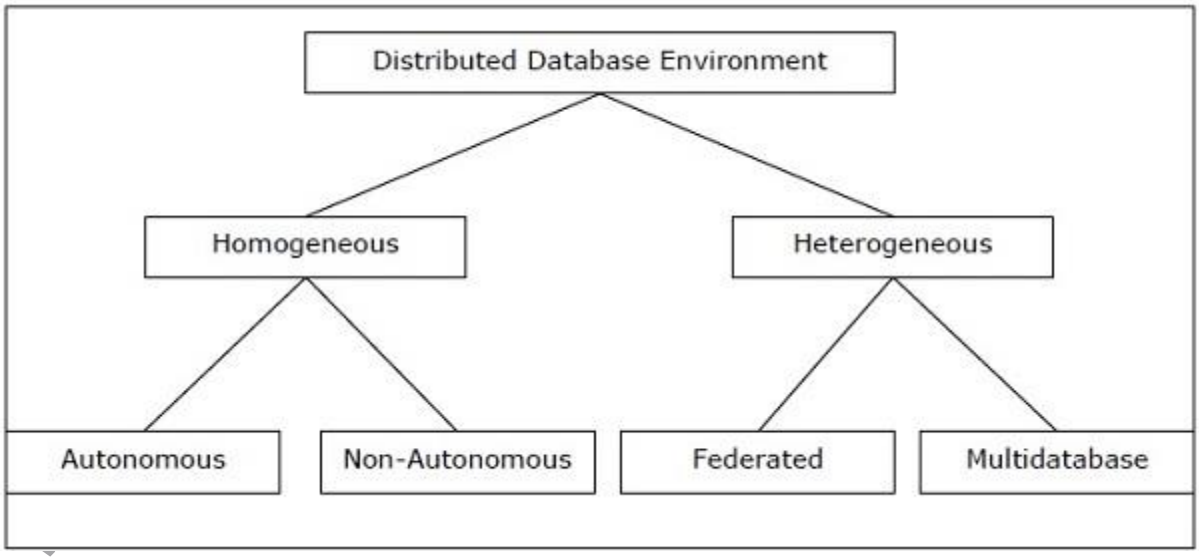
- Added meaning is given by the context of data for its use in biological applications.
  Whenever appropriate, context must be maintained and conveyed to the user. For the maximization of the interpretation of a biological data value, it should be possible to integrate as many contexts as possible.

Distributed databases

Distributed databases can be classified into homogeneous and heterogeneous databases having further divisions.

Types of Distributed Databases

Distributed databases can be broadly classified into homogeneous and heterogeneous distributed database environments, each with further sub-divisions, as shown in the following illustration.



Homogeneous Distributed Databases

In a homogeneous distributed database, all the sites use identical DBMS and operating systems. Its properties are –

DIGITAL EDUCATION HUB

- The sites use very similar software.
- The sites use identical DBMS or DBMS from the same vendor.
- Each site is aware of all other sites and cooperates with other sites to process user requests.
- The database is accessed through a single interface as if it is a single database.

Types of Homogeneous Distributed Database

There are two types of homogeneous distributed database −

- **Autonomous** − Each database is independent that functions on its own. They are integrated by a controlling application and use message passing to share data updates.
- **Non-autonomous** − Data is distributed across the homogeneous nodes and a central or master DBMS co-ordinates data updates across the sites.

Heterogeneous Distributed Databases

In a heterogeneous distributed database, different sites have different operating systems, DBMS products and data models. Its properties are −

- Different sites use dissimilar schemas and software.
- The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.
- Query processing is complex due to dissimilar schemas.
- Transaction processing is complex due to dissimilar software.
- A site may not be aware of other sites and so there is limited co-operation in processing user requests.

Types of Heterogeneous Distributed Databases

DIWAKAR EDUCATION HUB

- **Federated** − The heterogeneous database systems are independent in nature and integrated together so that they function as a single database system.
- **Un-federated** − The database systems employ a central coordinating module through which the databases are accessed.

Distributed DBMS Architectures

DDBMS architectures are generally developed depending on three parameters −

- **Distribution** − It states the physical distribution of data across the different sites.
- **Autonomy** − It indicates the distribution of control of the database system and the degree to which each constituent DBMS can operate independently.
- **Heterogeneity** − It refers to the uniformity or dissimilarity of the data models, system components and databases.

Architectural Models

Some of the common architectural models are −

- Client - Server Architecture for DDBMS
- Peer - to - Peer Architecture for DDBMS
- Multi - DBMS Architecture

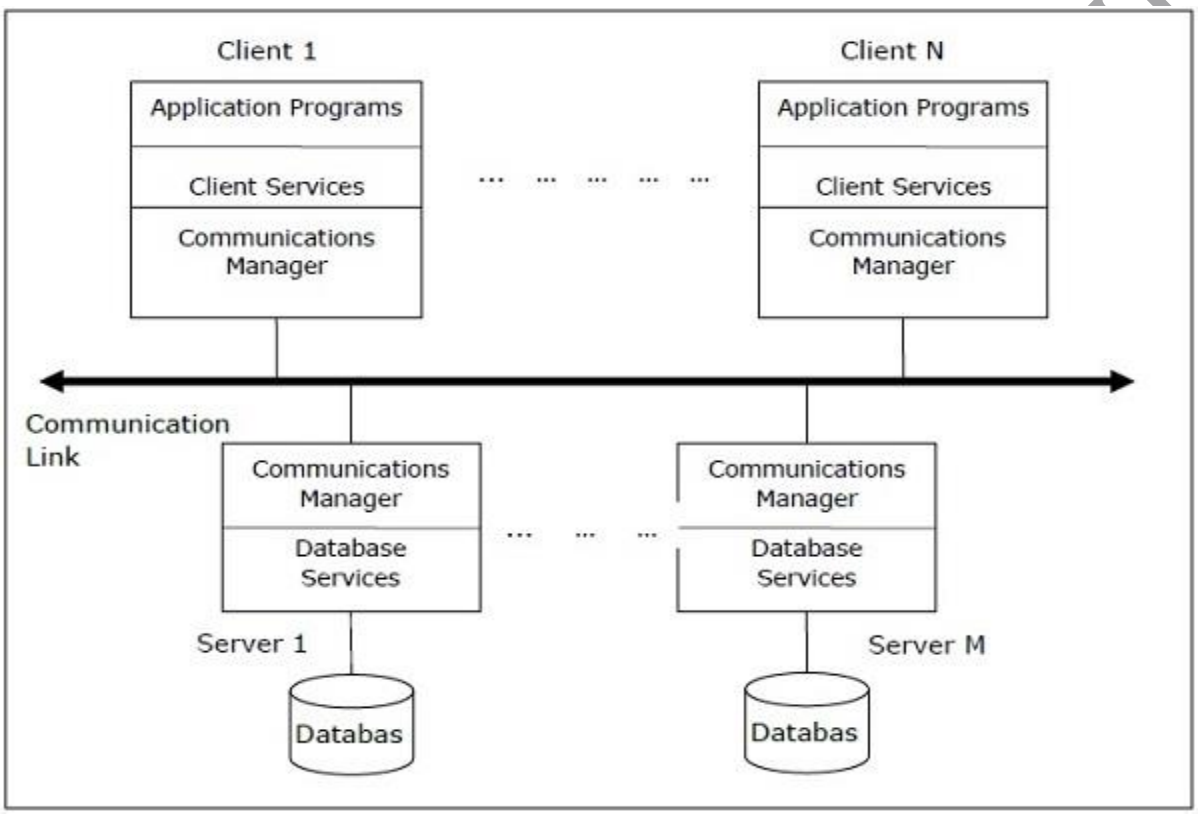
Client - Server Architecture for DDBMS

This is a two-level architecture where the functionality is divided into servers and clients. The server functions primarily encompass data management, query processing, optimization and transaction management. Client functions include mainly user interface. However, they have some functions like consistency checking and transaction management.

DIWAKAR EDUCATION HUB

The two different client - server architecture are –

- Single Server Multiple Client
- Multiple Server Multiple Client (shown in the following diagram)



Peer- to-Peer Architecture for DDBMS

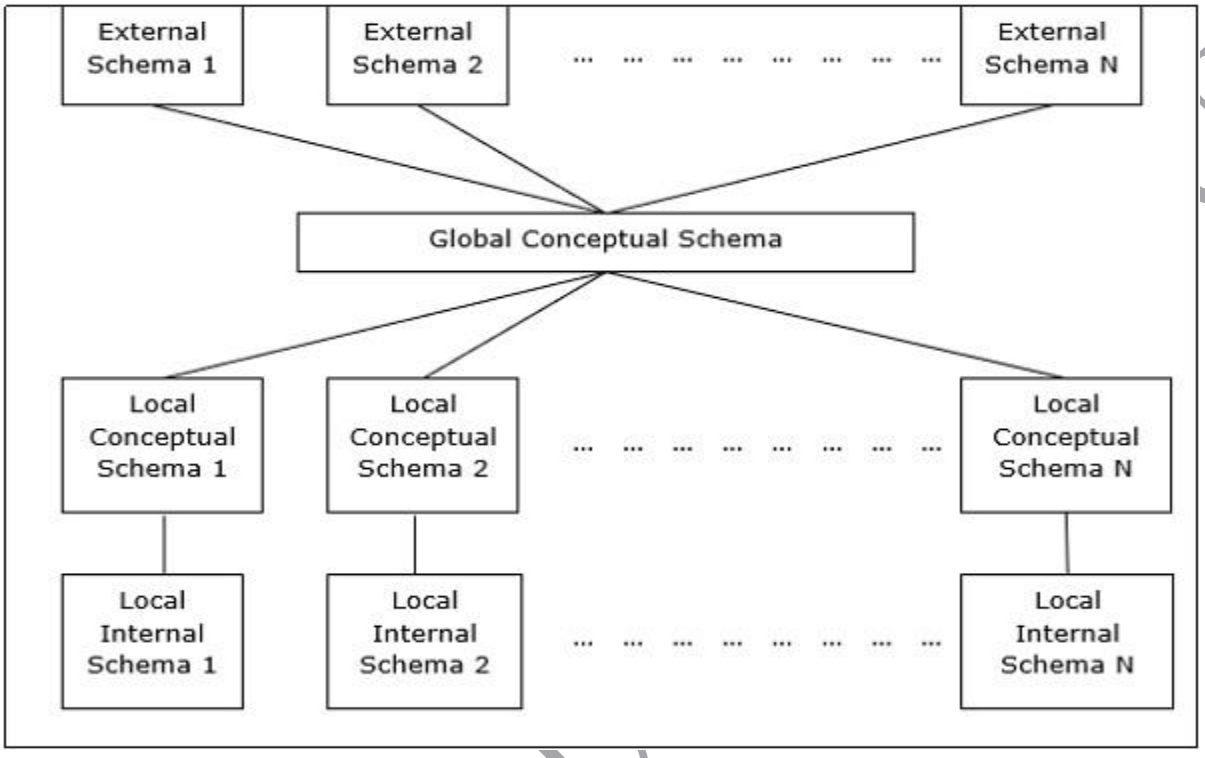
In these systems, each peer acts both as a client and a server for imparting database services. The peers share their resource with other peers and co-ordinate their activities.

This architecture generally has four levels of schemas –

- **Global Conceptual Schema** – Depicts the global logical view of data.
- **Local Conceptual Schema** – Depicts logical data organization at each site.
- **Local Internal Schema** – Depicts physical data organization at each site.

DIWAKAR

- **External Schema** – Depicts user view of data.



Multi - DBMS Architectures

This is an integrated database system formed by a collection of two or more autonomous database systems.

Multi-DBMS can be expressed through six levels of schemas –

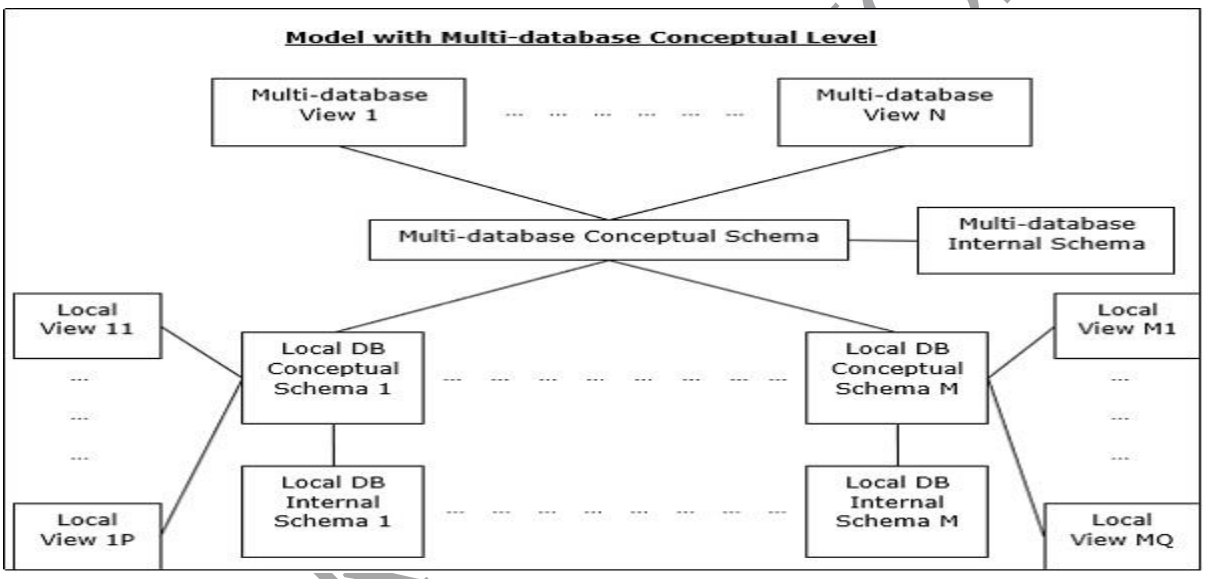
- **Multi-database View Level** – Depicts multiple user views comprising of subsets of the integrated distributed database.
- **Multi-database Conceptual Level** – Depicts integrated multi-database that comprises of global logical multi-database structure definitions.
- **Multi-database Internal Level** – Depicts the data distribution across different sites and multi-database to local data mapping.
- **Local database View Level** – Depicts public view of local data.

DIWAKAR EDUCATION HUB

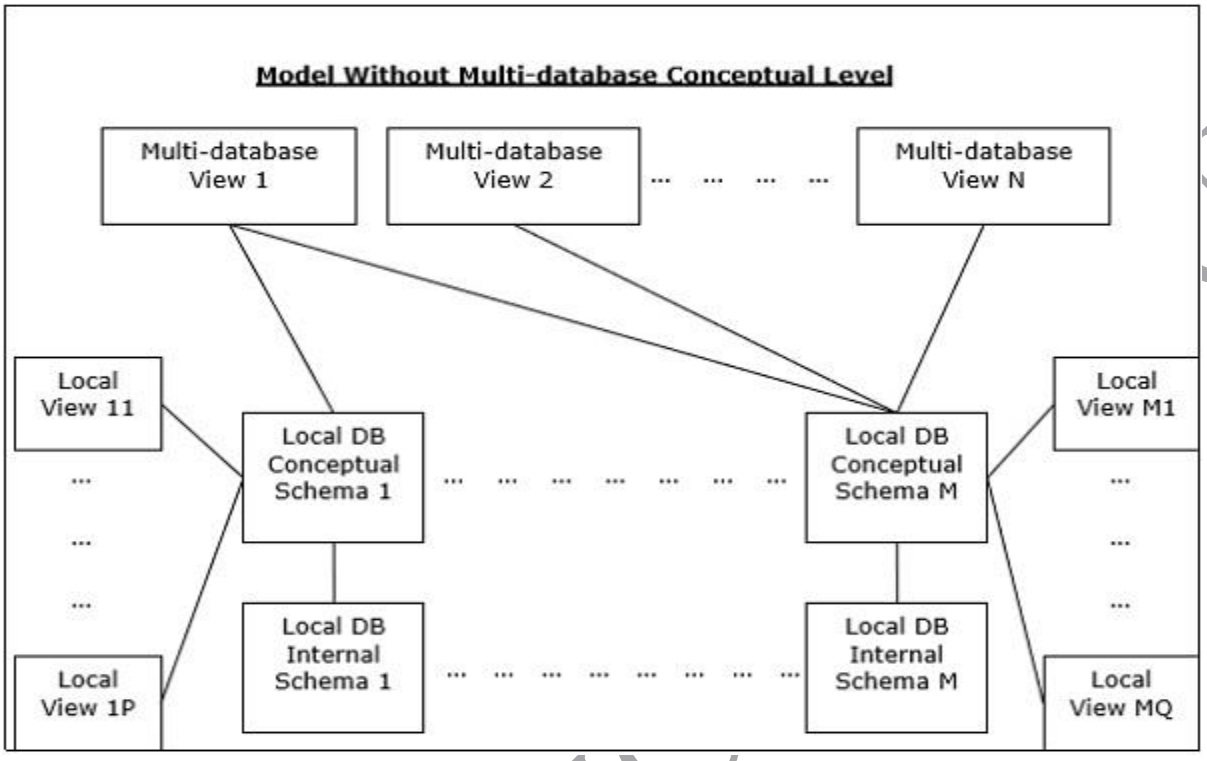
- **Local database Conceptual Level** − Depicts local data organization at each site.
- **Local database Internal Level** − Depicts physical data organization at each site.

There are two design alternatives for multi-DBMS −

- Model with multi-database conceptual level.
- Model without multi-database conceptual level.



DIWAKAR EDUCATION HUB



Design Alternatives

The distribution design alternatives for the tables in a DDBMS are as follows −

- Non-replicated and non-fragmented
- Fully replicated
- Partially replicated
- Fragmented
- Mixed

Non-replicated & Non-fragmented

In this design alternative, different tables are placed at different sites. Data is placed so that it is at a close proximity to the site where it is used most. It is most suitable for database systems where the percentage of queries needed to join information in tables placed at different sites is low. If an appropriate distribution strategy is adopted, then this design alternative helps to reduce the communication cost during data processing.

DIWAKAR EDUCATION HUB

Fully Replicated

In this design alternative, at each site, one copy of all the database tables is stored. Since, each site has its own copy of the entire database, queries are very fast requiring negligible communication cost. On the contrary, the massive redundancy in data requires huge cost during update operations. Hence, this is suitable for systems where a large number of queries is required to be handled whereas the number of database updates is low.

Partially Replicated

Copies of tables or portions of tables are stored at different sites. The distribution of the tables is done in accordance to the frequency of access. This takes into consideration the fact that the frequency of accessing the tables vary considerably from site to site. The number of copies of the tables (or portions) depends on how frequently the access queries execute and the site which generate the access queries.

Fragmented

In this design, a table is divided into two or more pieces referred to as fragments or partitions, and each fragment can be stored at different sites. This considers the fact that it seldom happens that all data stored in a table is required at a given site. Moreover, fragmentation increases parallelism and provides better disaster recovery. Here, there is only one copy of each fragment in the system, i.e. no redundant data.

The three fragmentation techniques are −

- Vertical fragmentation
- Horizontal fragmentation
- Hybrid fragmentation

Mixed Distribution

DIWAKAR EDUCATION HUB

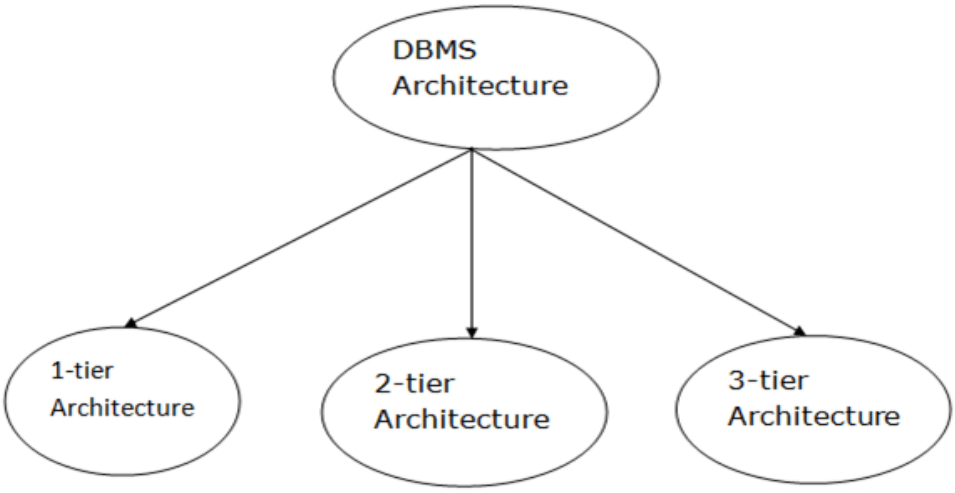
This is a combination of fragmentation and partial replications. Here, the tables are initially fragmented in any form (horizontal or vertical), and then these fragments are partially replicated across the different sites according to the frequency of accessing the fragments.

DBMS Architecture

In client server computing, the clients requests a resource and the server provides that resource. A server may serve multiple clients at the same time while a client is in contact with only one server.

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture



DIW EDUCATION HUB

Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: 2-tier architecture and 3-tier architecture.

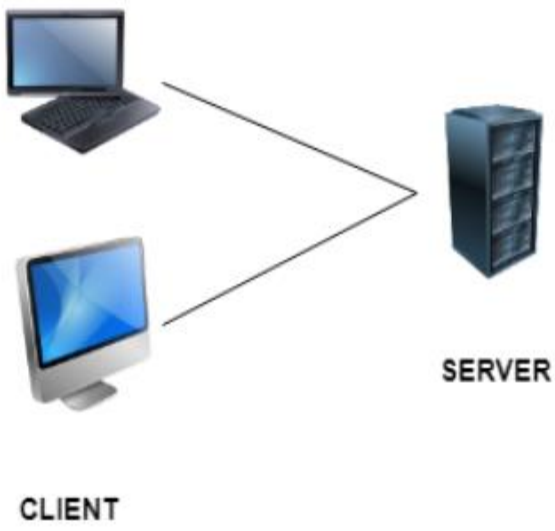
The different structures for two tier and three tier are given as follows −

Two - Tier Client/Server Architecture

The two tier architecture primarily has two parts, a client tier and a server tier.The client tier sends a request to the server tier and the server tier responds with the desired information.

An example of a two tier client/server structure is a web server. It returns the required web pages to the clients that requested them.

An illustration of the two-tier client/server structure is as follows −



Advantages of Two - Tier Client/Server Architecture

Some of the advantages of the two-tier client/server structure are −

- This structure is quite easy to maintain and modify.

- The communication between the client and server in the form of request response messages is quite fast.

Disadvantages of Two - Tier Client/Server Architecture

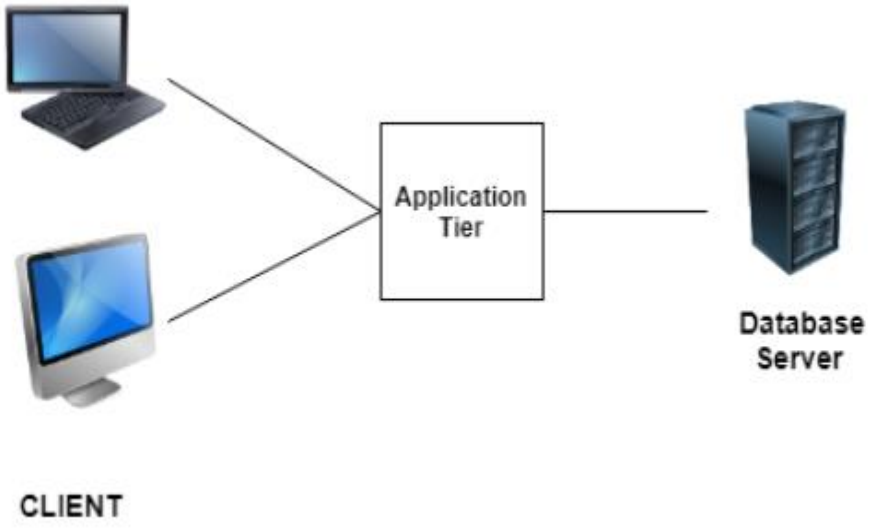
A major disadvantage of the two-tier client/server structure is −

- If the client nodes are increased beyond capacity in the structure, then the server is not able to handle the request overflow and performance of the system degrades.

Three - Tier Client/Server Architecture

The three tier architecture has three layers namely client, application and data layer. The client layer is the one that requests the information. In this case it could be the GUI, web interface etc. The application layer acts as an interface between the client and data layer. It helps in communication and also provides security. The data layer is the one that actually contains the required data.

An illustration of the three-tier client/server structure is as follows −



Advantages of Three - Tier Client/Server Architecture

Some of the advantages of the three-tier client/server structure are −

- The three tier structure provides much better service and fast performance.
- The structure can be scaled according to requirements without any problem.
- Data security is much improved in the three tier structure.

Disadvantages of Three - Tier Client/Server Architecture

A major disadvantage of the three-tier client/server structure is −

- Three - tier client/server structure is quite complex due to advanced features.

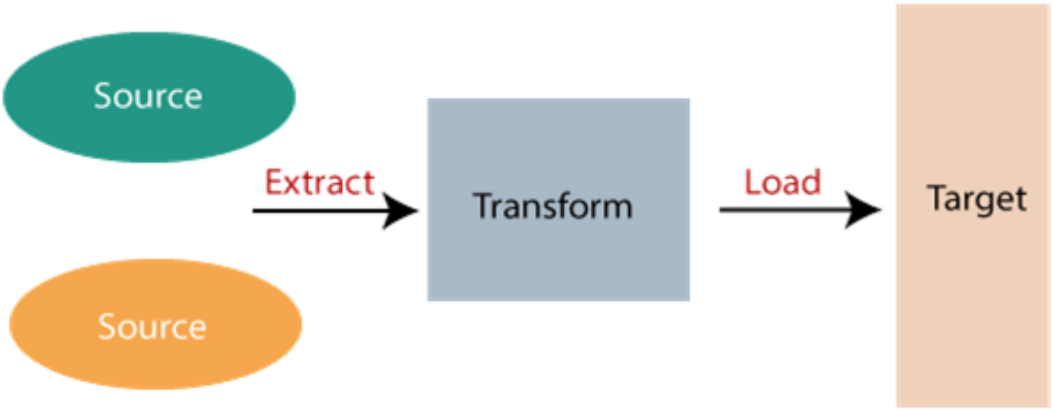
Data Mining Vs Data Warehousing

Data warehouse refers to the process of compiling and organizing data into one common database, whereas **data mining** refers to the process of extracting useful data from the databases. The data mining process depends on the data compiled in the data warehousing phase to recognize meaningful patterns. A data warehousing is created to support management systems.

Data Warehouse:

A **Data Warehouse** refers to a place where data can be stored for useful mining. It is like a quick computer system with exceptionally huge data storage capacity. Data from the various organization's systems are copied to the Warehouse, where it can be fetched and conformed to delete errors. Here, advanced requests can be made against the warehouse storage of data.

DIWAKAR EDUCATION HUB



Data Warehousing Process

Data warehouse combines data from numerous sources which ensure the data quality, accuracy, and consistency. Data warehouse boosts system execution by separating analytics processing from transnational databases. Data flows into a data warehouse from different databases. A data warehouse works by sorting out data into a pattern that depicts the format and types of data. Query tools examine the data tables using patterns.

Data warehouses and **databases** both are relative data systems, but both are made to serve different purposes. A data warehouse is built to store a huge amount of historical data and empowers fast requests over all the data, typically using **Online Analytical Processing** (OLAP). A database is made to store current transactions and allow quick access to specific transactions for ongoing business processes, commonly known as **Online Transaction Processing** (OLTP).

Important Features of Data Warehouse

The Important features of Data Warehouse are given below:

1. Subject Oriented

A data warehouse is subject-oriented. It provides useful data about a subject instead of the company's ongoing operations, and these subjects can be customers, suppliers, marketing, product, promotion, etc. A data warehouse

DIWAKAR EDUCATION HUB

usually focuses on modeling and analysis of data that helps the business organization to make data-driven decisions.

2. Time-Variant:

The different data present in the data warehouse provides information for a specific period.

3. Integrated

A data warehouse is built by joining data from heterogeneous sources, such as social databases, level documents, etc.

4. Non- Volatile

It means, once data entered into the warehouse cannot be change.

Advantages of Data Warehouse:

- More accurate data access
- Improved productivity and performance
- Cost-efficient
- Consistent and quality data

Data Mining:

Data mining refers to the analysis of data. It is the computer-supported process of analyzing huge sets of data that have either been compiled by computer systems or have been downloaded into the computer. In the data mining process, the computer analyzes the data and extract useful information from it. It looks for hidden patterns within the data set and try to predict future behavior. Data mining is primarily used to discover and indicate relationships among the data sets.

DIWAKAR EDUCATION HUB



Data mining aims to enable business organizations to view business behaviors, trends relationships that allow the business to make data-driven decisions. It is also known as knowledge Discover in Database (KDD). Data mining tools utilize AI, statistics, databases, and machine learning systems to discover the relationship between the data. Data mining tools can support business-related questions that traditionally time-consuming to resolve any issue.

Important features of Data Mining:

The important features of Data Mining are given below:

- It utilizes the Automated discovery of patterns.
- It predicts the expected results.
- It focuses on large data sets and databases
- It creates actionable information.

Advantages of Data Mining:

i. Market Analysis:

Data Mining can predict the market that helps the business to make the decision. For example, it predicts who is keen to purchase what type of products.

DIWAKAR EDUCATION HUB

ii. Fraud detection:

Data Mining methods can help to find which cellular phone calls, insurance claims, credit, or debit card purchases are going to be fraudulent.

iii. Financial Market Analysis:

Data Mining techniques are widely used to help **Model Financial Market**

iv. Trend Analysis:

Analyzing the current existing trend in the marketplace is a strategic benefit because it helps in cost reduction and manufacturing process as per market demand.

Differences between Data Mining and Data Warehousing:

Data Mining	Data Warehousing
Data mining is the process of determining data patterns.	A data warehouse is a database system designed for analytics.
Data mining is generally considered as the process of extracting useful data from a large set of data.	Data warehousing is the process of combining all the relevant data.
Business entrepreneurs carry data mining with the help of engineers.	Data warehousing is entirely carried out by the engineers.
In data mining, data is analyzed	In data warehousing, data is stored

DUCATION HUB

repeatedly.	periodically.
Data mining uses pattern recognition techniques to identify patterns.	Data warehousing is the process of extracting and storing data that allow easier reporting.
One of the most amazing data mining technique is the detection and identification of the unwanted errors that occur in the system.	One of the advantages of the data warehouse is its ability to update frequently. That is the reason why it is ideal for business entrepreneurs who want up to date with the latest stuff.
The data mining techniques are cost-efficient as compared to other statistical data applications.	The responsibility of the data warehouse is to simplify every type of business data.
The data mining techniques are not 100 percent accurate. It may lead to serious consequences in a certain condition.	In the data warehouse, there is a high possibility that the data required for analysis by the company may not be integrated into the warehouse. It can simply lead to loss of data.
Companies can benefit from this analytical tool by equipping suitable and accessible knowledge-based data.	Data warehouse stores a huge amount of historical data that helps users to analyze different periods and trends to make future predictions.

Data Warehouse Modeling

Data warehouse modeling is the process of designing the schemas of the detailed and summarized information of the data warehouse. The goal of data warehouse modeling is to develop a schema describing the reality, or at least a part of the fact, which the data warehouse is needed to support.

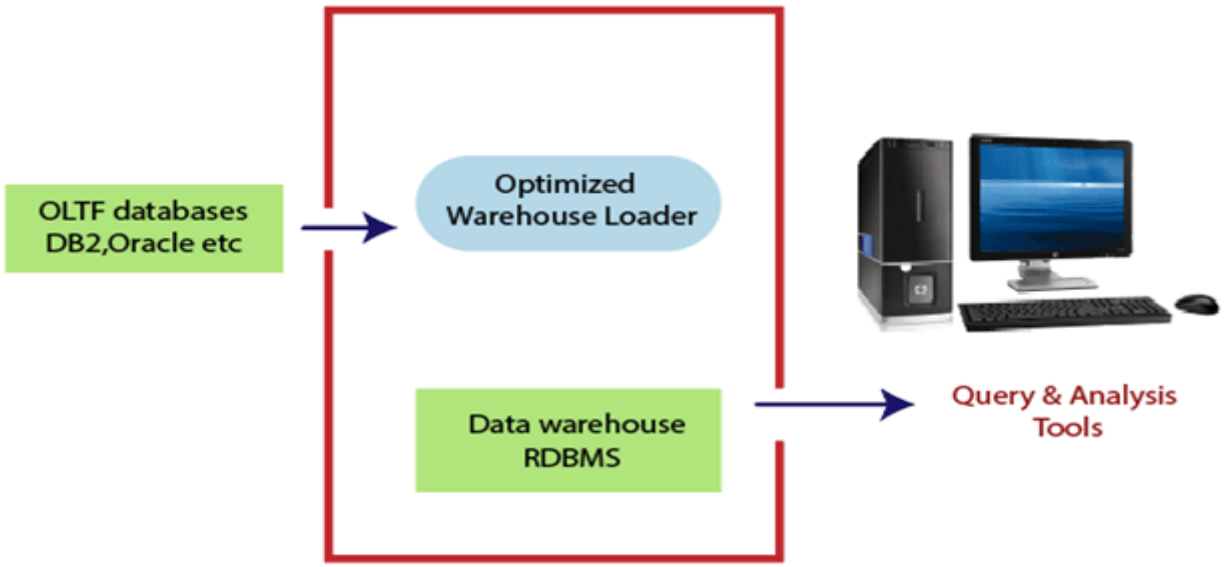
Data warehouse modeling is an essential stage of building a data warehouse for two main reasons. Firstly, through the schema, data warehouse clients can visualize the relationships among the warehouse data, to use them with greater ease. Secondly, a well-designed schema allows an effective data warehouse structure to emerge, to help decrease the cost of implementing the warehouse and improve the efficiency of using it.

Data modeling in data warehouses is different from data modeling in operational database systems. The primary function of data warehouses is to support DSS processes. Thus, the objective of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.

In contrast, data modeling in operational database systems targets efficiently supporting simple transactions in the database such as retrieving, inserting, deleting, and changing data. Moreover, data warehouses are designed for the customer with general information knowledge about the enterprise, whereas operational database systems are more oriented toward use by software specialists for creating distinct applications.

Data Warehouse model is illustrated in the given diagram.

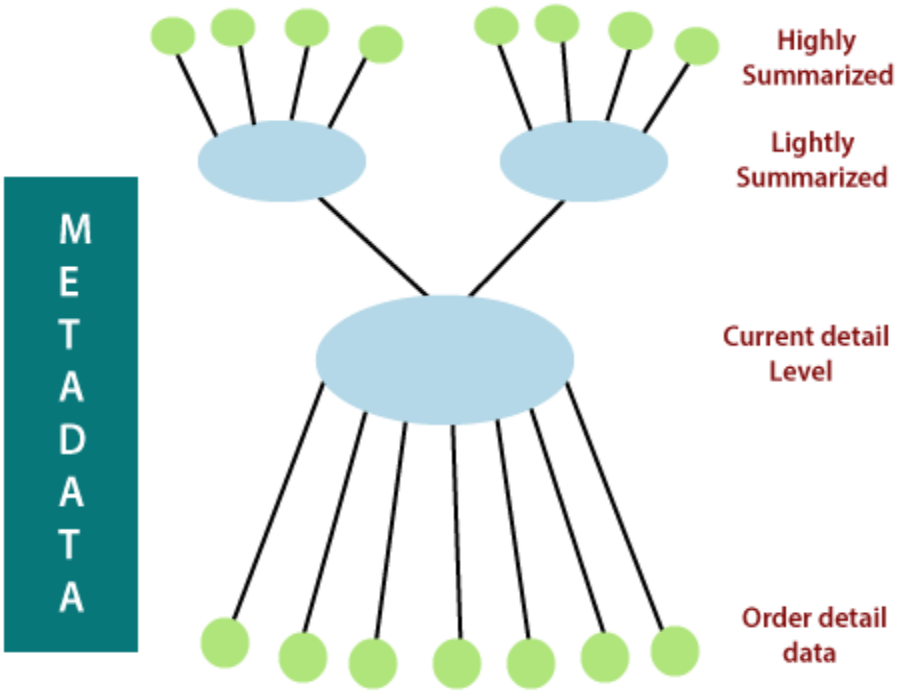
DIWAKAR EDUCATION HUB



Data Warehouse Model

The data within the specific warehouse itself has a particular architecture with the emphasis on various levels of summarization, as shown in figure:

DIWAKAR EDUCA



The Structure of data inside the data warehouse

The current detail record is central in importance as it:

- Reflects the most current happenings, which are commonly the most stimulating.
- It is numerous as it is saved at the lowest method of the Granularity.
- It is always (almost) saved on disk storage, which is fast to access but expensive and difficult to manage.

Older detail data is stored in some form of mass storage, and it is infrequently accessed and kept at a level detail consistent with current detailed data.

Lightly summarized data is data extract from the low level of detail found at the current, detailed level and usually is stored on disk storage. When building the data warehouse have to remember what unit of time is summarization done over and also the components or what attributes the summarized data will contain.

DIWAKAR EDUCATION HUB

Highly summarized data is compact and directly available and can even be found outside the warehouse.

Metadata is the final element of the data warehouses and is really of various dimensions in which it is not the same as file drawn from the operational data, but it is used as:-

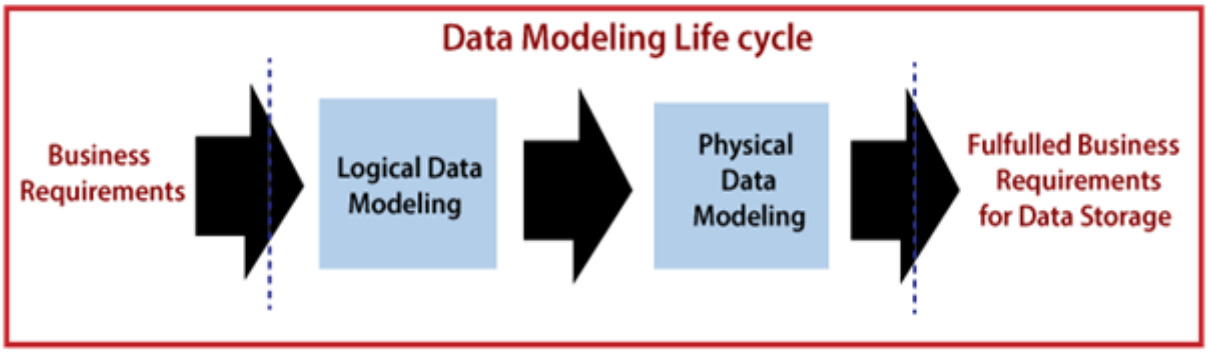
- A directory to help the DSS investigator locate the items of the data warehouse.
- A guide to the mapping of record as the data is changed from the operational data to the data warehouse environment.
- A guide to the method used for summarization between the current, accurate data and the lightly summarized information and the highly summarized data, etc.

Data Modeling Life Cycle

In this section, we define a data modeling life cycle. It is a straight forward process of transforming the business requirements to fulfill the goals for storing, maintaining, and accessing the data within IT systems. The result is a logical and physical data model for an enterprise data warehouse.

The objective of the data modeling life cycle is primarily the creation of a storage area for business information. That area comes from the logical and physical data modeling stages, as shown in Figure:

DIWAKAR EDUCATION HUB



A generic data modeling life cycle

Conceptual Data Model

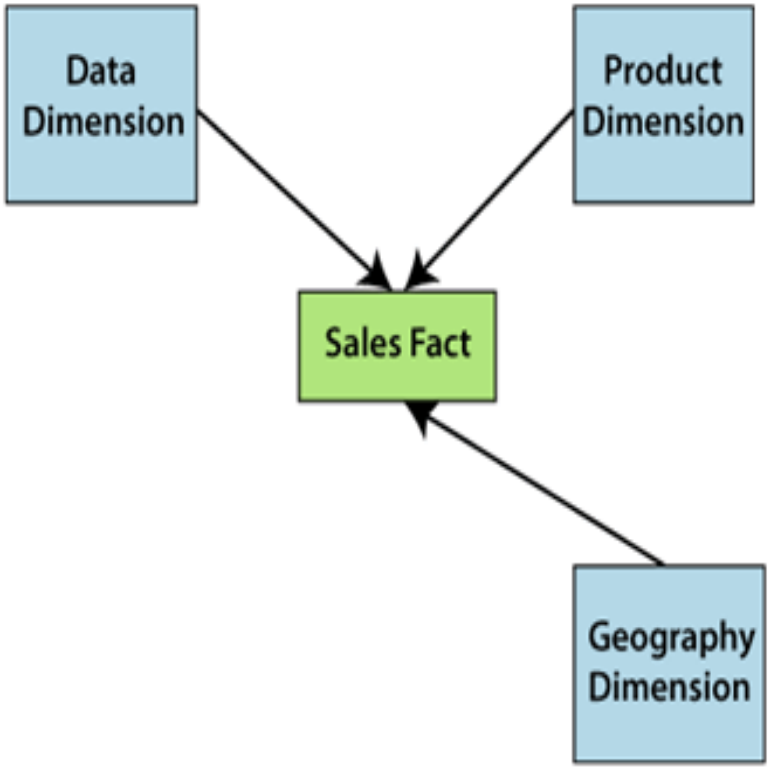
A conceptual data model recognizes the highest-level relationships between the different entities.

Characteristics of the conceptual data model

- It contains the essential entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.

We can see that the only data shown via the conceptual data model is the entities that define the data and the relationships between those entities. No other data, as shown through the conceptual data model.

DIWAKAR EDUCATION



Example of Conceptual Data Model

Logical Data Model

A logical data model defines the information in as much structure as possible, without observing how they will be physically achieved in the database. The primary objective of logical data modeling is to document the business data structures, processes, rules, and relationships by a single view - the logical data model.

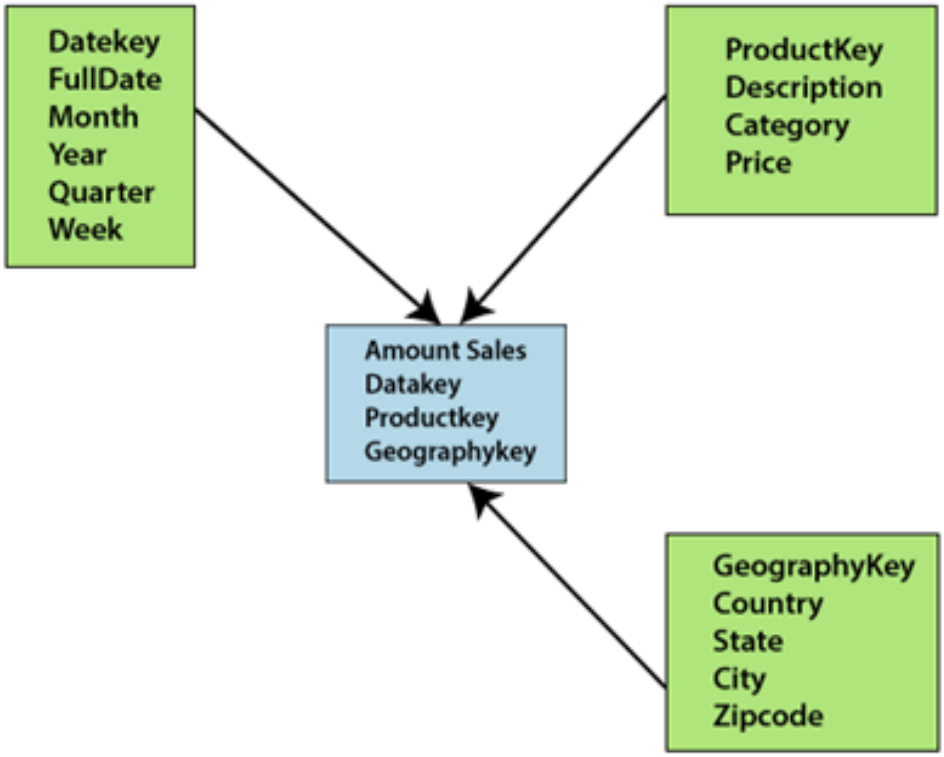
Features of a logical data model

- It involves all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is stated.
- Referential Integrity is specified (FK Relation).

DIWAKAR L HUB

The phase for designing the logical data model which are as follows:

- Specify primary keys for all entities.
- List the relationships between different entities.
- List all attributes for each entity.
- Normalization.
- No data types are listed



Example of Logical Data Model

Physical Data Model

Physical data model describes how the model will be presented in the database. A physical database model demonstrates all table structures, column names, data types, constraints, primary key, foreign key, and relationships between tables. The purpose of physical data modeling is the mapping of the logical data model to

the physical structures of the RDBMS system hosting the data warehouse. This contains defining physical RDBMS structures, such as tables and data types to use when storing the information. It may also include the definition of new data structures for enhancing query performance.

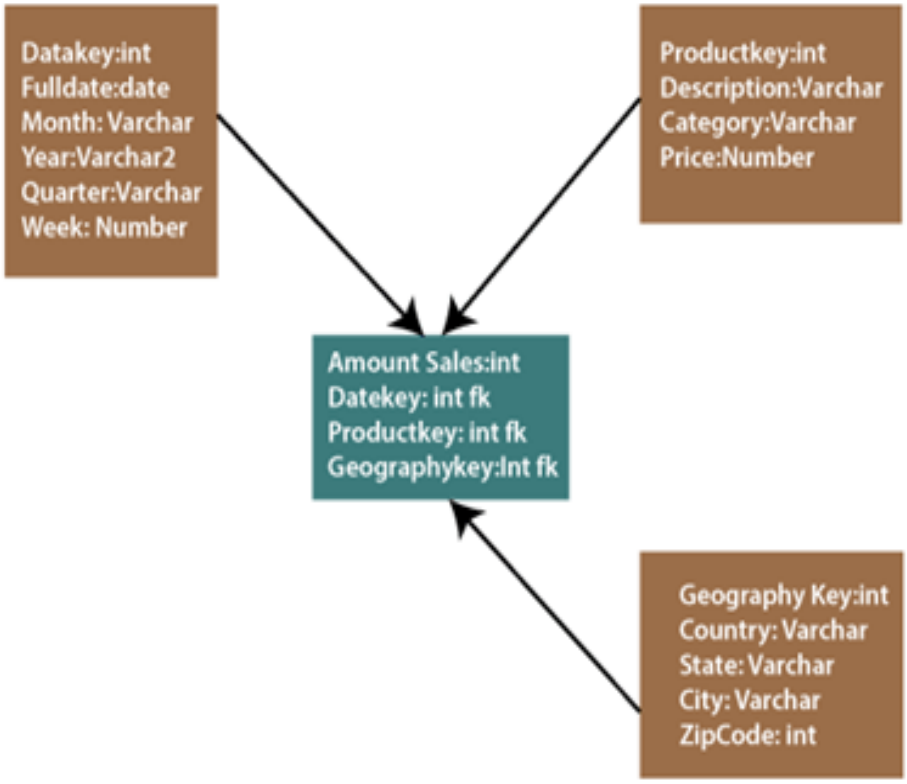
Characteristics of a physical data model

- Specification all tables and columns.
- Foreign keys are used to recognize relationships between tables.

The steps for physical data model design which are as follows:

- Convert entities to tables.
- Convert relationships to foreign keys.
- Convert attributes to columns.

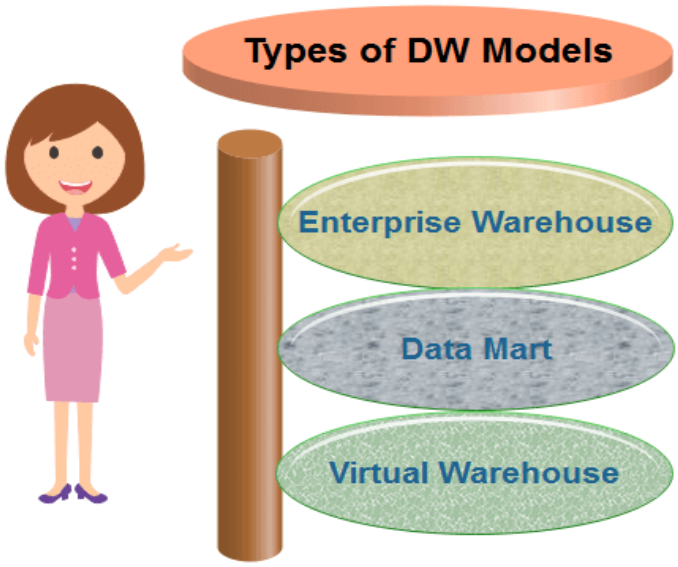
DIWAKAR EDUCATION HUB



Example of Physical Data Model

Types of Data Warehouse Models

DIWAKAR



Enterprise Warehouse

An Enterprise warehouse collects all of the records about subjects spanning the entire organization. It supports corporate-wide data integration, usually from one or more operational systems or external data providers, and it's cross-functional in scope. It generally contains detailed information as well as summarized information and can range in estimate from a few gigabyte to hundreds of gigabytes, terabytes, or beyond.

An enterprise data warehouse may be accomplished on traditional mainframes, UNIX super servers, or parallel architecture platforms. It required extensive business modeling and may take years to develop and build.

Data Mart

A data mart includes a subset of corporate-wide data that is of value to a specific collection of users. The scope is confined to particular selected subjects. For example, a marketing data mart may restrict its subjects to the customer, items, and sales. The data contained in the data marts tend to be summarized.

Data Marts is divided into two parts:

DIWAKAR EDUCATION HUB

Independent Data Mart: Independent data mart is sourced from data captured from one or more operational systems or external data providers, or data generally locally within a different department or geographic area.

Dependent Data Mart: Dependent data marts are sourced exactly from enterprise data-warehouses.

Virtual Warehouses

Virtual Data Warehouses is a set of perception over the operational database. For effective query processing, only some of the possible summary vision may be materialized. A virtual warehouse is simple to build but required excess capacity on operational database servers.

Concept Hierarchy

A **concept hierarchy** defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts. Consider a concept hierarchy for the dimension *location*. City values for *location* include Vancouver, Toronto, New York, and Chicago. Each city, however, can be mapped to the province or state to which it belongs. For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois. The provinces and states can in turn be mapped to the country (e.g., Canada or the United States) to which they belong. These mappings form a concept hierarchy for the dimension *location*, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries). This concept hierarchy is illustrated in Figure 4.9.

DIWAKAR EDUCATION HUB

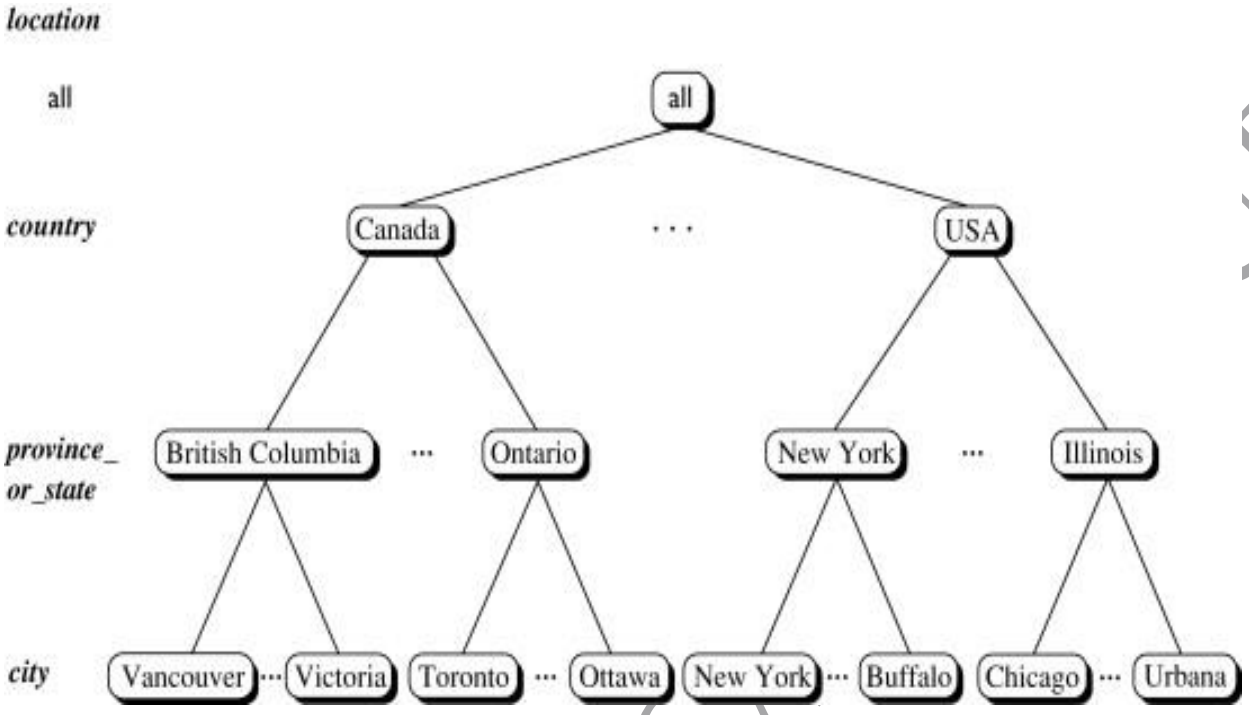


Figure 4.9. A concept hierarchy for *location*. Due to space limitations, not all of the hierarchy nodes are shown, indicated by ellipses between nodes.

Many concept hierarchies are implicit within the database schema. For example, suppose that the dimension *location* is described by the attributes *number, street, city, province_or_state, zip_code*, and *country*. These attributes are related by a total order, forming a concept hierarchy such as *"street < city < province_or_state < country."* This hierarchy is shown in Figure 4.10(a). Alternatively, the attributes of a dimension may be organized in a partial order, forming a lattice. An example of a partial order for the *time* dimension based on the attributes *day, week, month, quarter*, and *year* is *"day <{month < quarter; week} < year."*[1] This lattice structure is shown in Figure 4.10(b). A concept hierarchy that is a total or partial order among attributes in a database schema is called a **schema hierarchy**. Concept hierarchies that are common to many applications (e.g., *for time*) may be predefined in the data mining system. Data mining systems should provide users with the flexibility to tailor predefined hierarchies according to their particular needs. For example, users may want to define a fiscal year starting on April 1 or an academic year starting on September 1.

DIWAKAR EDUCA B

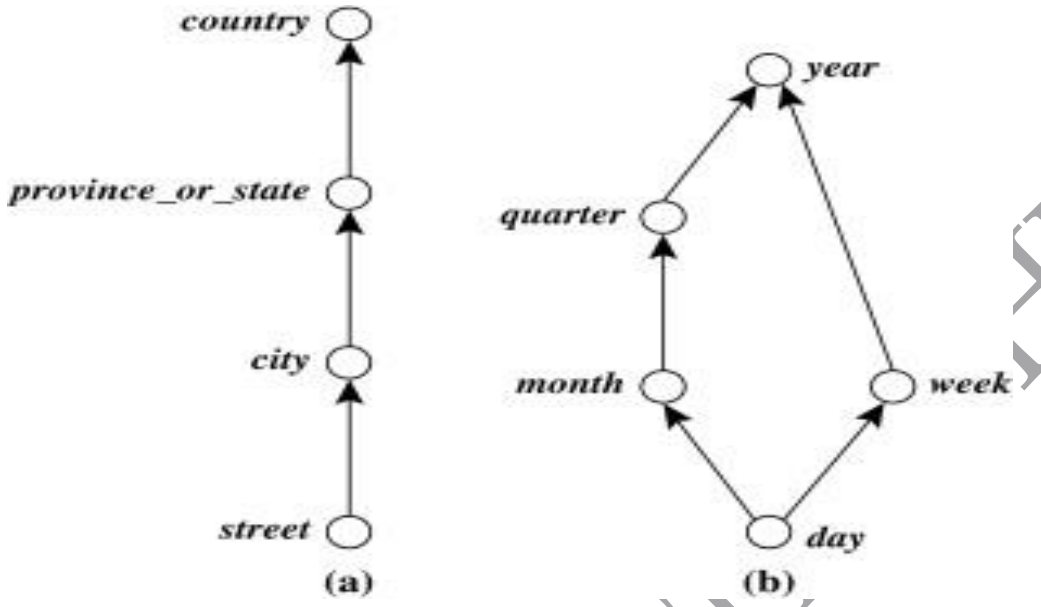


Figure 4.10. Hierarchical and lattice structures of attributes in warehouse dimensions: (a) a hierarchy for *location* and (b) a lattice for *time*.

Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a **set-grouping hierarchy**. A total or partial order can be defined among groups of values. An example of a set-grouping hierarchy is shown in Figure 4.11 for the dimension *price*, where an interval ($*X*…$*Y*] denotes the range from $*X* (exclusive) to $*Y* (inclusive).

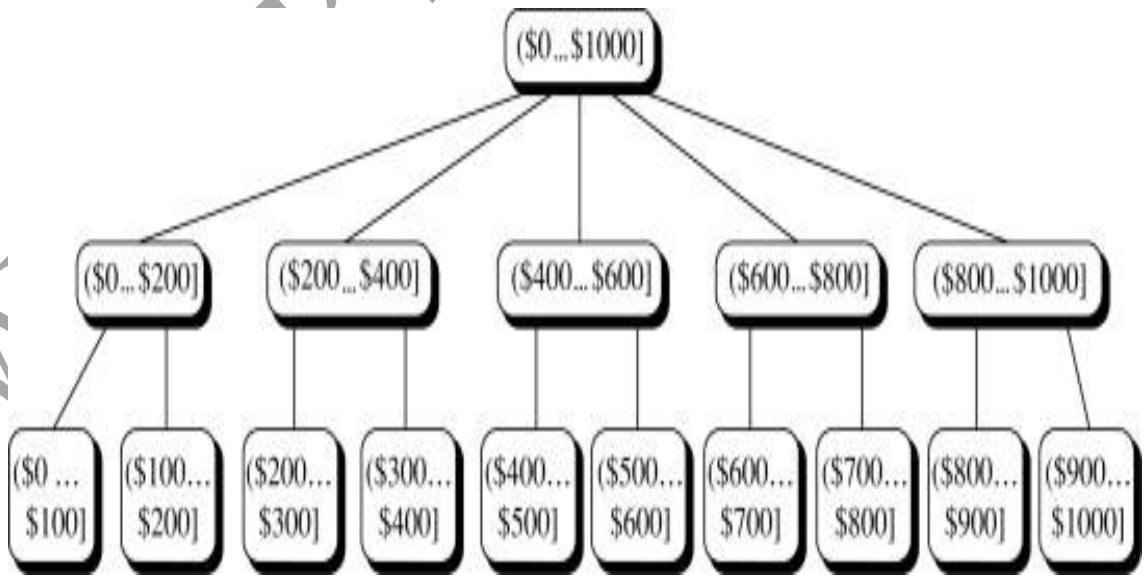


Figure 4.11. A concept hierarchy for *price*.

There may be more than one concept hierarchy for a given attribute or dimension, based on different user viewpoints. For instance, a user may prefer to organize *price* by defining ranges for *inexpensive, moderately_priced*, and *expensive*.

Concept hierarchies may be provided manually by system users, domain experts, or knowledge engineers, or may be automatically generated based on statistical analysis of the data distribution. The automatic generation of concept hierarchies is discussed in Chapter 3 as a preprocessing step in preparation for data mining.

OLTP and OLAP: The two terms look similar but refer to different kinds of systems. Online transaction processing (OLTP) captures, stores, and processes data from transactions in real time. Online analytical processing (OLAP) uses complex queries to analyze aggregated historical data from OLTP systems.

OLTP

An OLTP system captures and maintains transaction data in a database. Each transaction involves individual database records made up of multiple fields or columns. Examples include banking and credit card activity or retail checkout scanning.

In OLTP, the emphasis is on fast processing, because OLTP databases are read, written, and updated frequently. If a transaction fails, built-in system logic ensures data integrity.

OLAP

OLAP applies complex queries to large amounts of historical data, aggregated from OLTP databases and other sources, for data mining, analytics, and **business intelligence** projects. In OLAP, the emphasis is on response time to these complex queries. Each query involves one or more columns of data aggregated from many rows. Examples include year-over-year financial performance or marketing lead

DIWAKAR EDUCATION HUB

generation trends. OLAP databases and **data warehouses** give analysts and decision-makers the ability to use custom reporting tools to turn data into information. Query failure in OLAP does not interrupt or delay transaction processing for customers, but it can delay or impact the accuracy of business intelligence insights.

OLTP vs. OLAP: side-by-side comparison

OLTP is operational, while OLAP is informational. A glance at the key features of both kinds of processing illustrates their fundamental differences, and how they work together.

	OLTP	OLAP
Characteristics	Handles a large number of small transactions	Handles large volumes of data with complex queries
Query types	Simple standardized queries	Complex queries
Operations	Based on INSERT, UPDATE, DELETE commands	Based on SELECT commands to aggregate data for reporting
Response time	Milliseconds	Seconds, minutes, or hours depending on the

		amount of data to process
Design	Industry-specific, such as retail, manufacturing, or banking	Subject-specific, such as sales, inventory, or marketing
Source	Transactions	Aggregated data from transactions
Purpose	Control and run essential business operations in real time	Plan, solve problems, support decisions, discover hidden insights
Data updates	Short, fast updates initiated by user	Data periodically refreshed with scheduled, long-running batch jobs
Space requirements	Generally small if historical data is archived	Generally large due to aggregating large datasets
Backup and recovery	Regular backups required to ensure business continuity	Lost data can be reloaded from OLTP database as

	and meet legal and governance requirements	needed in lieu of regular backups
Productivity	Increases productivity of end users	Increases productivity of business managers, data analysts, and executives
Data view	Lists day-to-day business transactions	Multi-dimensional view of enterprise data
User examples	Customer-facing personnel, clerks, online shoppers	Knowledge workers such as data analysts, business analysts, and executives
Database design	Normalized databases for efficiency	Denormalized databases for analysis

OLTP provides an immediate record of current business activity, while OLAP generates and validates insights from that data as it's compiled over time. That historical perspective empowers accurate forecasting, but as with all business

intelligence, the insights generated with OLAP are only as good as the data pipeline from which they emanate.

Association rules

Association rules are if-then statements that help to show the probability of relationships between data items within large data sets in various types of databases. Association rule mining has a number of applications and is widely used to help discover sales correlations in transactional data or in medical data sets.

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently.

Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

TID	ITEMS
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer

DIWAKAR EDUCATION HUB

TID	ITEMS
5	Bread, Milk, Diaper, Coke

Before starting first see the basic definitions.

Support Count( ) – Frequency of occurrence of a itemset.

Here ({Milk, Bread, Diaper})=2

Frequent Itemset – An itemset whose support is greater than or equal to minsup threshold.

Association Rule – An implication expression of the form X -> Y, where X and Y are any 2 itemsets.

Example: {Milk, Diaper}->{Beer}

Rule Evaluation Metrics –

- **Support(s) –**
  The number of transactions that include items in the {X} and {Y} parts of the rule as a percentage of the total number of transaction.It is a measure of how frequently the collection of items occur together as a percentage of all transactions.
- **Support = (X+Y) total –**
  It is interpreted as fraction of transactions that contain both X and Y.
- **Confidence(c) –**
  It is the ratio of the no of transactions that includes all items in {B} as well as the no of transactions that includes all items in {A} to the no of transactions that includes all items in {A}.

DIWAKAR EDUCATION HUB

- **Conf(X=>Y) = Supp(X Y) Supp(X) –**
  It measures how often each item in Y appears in transactions that contains items in X also.
- **Lift(l) –**
  The lift of the rule X=>Y is the confidence of the rule divided by the expected confidence, assuming that the itemsets X and Y are independent of each other.The expected confidence is the confidence divided by the frequency of {Y}.
- **Lift(X=>Y) = Conf(X=>Y) Supp(Y) –**
  Lift value near 1 indicates X and Y almost often appear together as expected, greater than 1 means they appear together more than expected and less than 1 means they appear less than expected.Greater lift values indicate stronger association.

Example – From the above table, {Milk, Diaper}=>{Beer}

s= ({Milk, Diaper, Beer}) |T|

= 2/5

= 0.4

c= (Milk, Diaper, Beer) (Milk, Diaper)

= 2/3

= 0.67

l= Supp({Milk, Diaper, Beer}) Supp({Milk, Diaper})*Supp({Beer})

= 0.4/(0.6*0.6)

= 1.11

The Association rule is very useful in analyzing datasets. The data is collected using bar-code scanners in supermarkets. Such databases consists of a large

DIWAKAR EDUCATION HUB

number of transaction records which list all items bought by a customer on a single purchase. So the manager could know if certain groups of items are consistently purchased together and use this data for adjusting store layouts, cross-selling, promotions based on statistics.

Classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.

Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm.

The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating.

In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These

DIWAKAR EDUCATION HUB

relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model. See "Testing a Classification Model".

Scoring a classification model results in class assignments and probabilities for each case. For example, a model that classifies customers as low, medium, or high value would also predict the probability of each classification for each customer.

Classification has many applications in customer segmentation, business modeling, marketing, credit analysis, and biomedical and drug response modeling.

A Sample Classification Problem

Suppose we want to predict which of our customers are likely to increase spending if given an affinity card. You could build a model using demographic data about customers who have used an affinity card in the past. Since we want to predict either a positive or a negative response (will or will not increase spending), we will build a binary classification model.

This example uses classification model, dt_sh_clas_sample, which is created by one of the Oracle Data Mining sample programs (described in *Oracle Data Mining Administrator's Guide*). Figure 5-1 shows six columns and ten rows from the case table used to build the model. A target value of 1 has been assigned to customers who increased spending with an affinity card; a value of 0 has been assigned to customers who did not increase spending.

Figure 5-1 Sample Build Data for Classification

DIWAKAR EDUCATION HUB

case ID | predictors | target

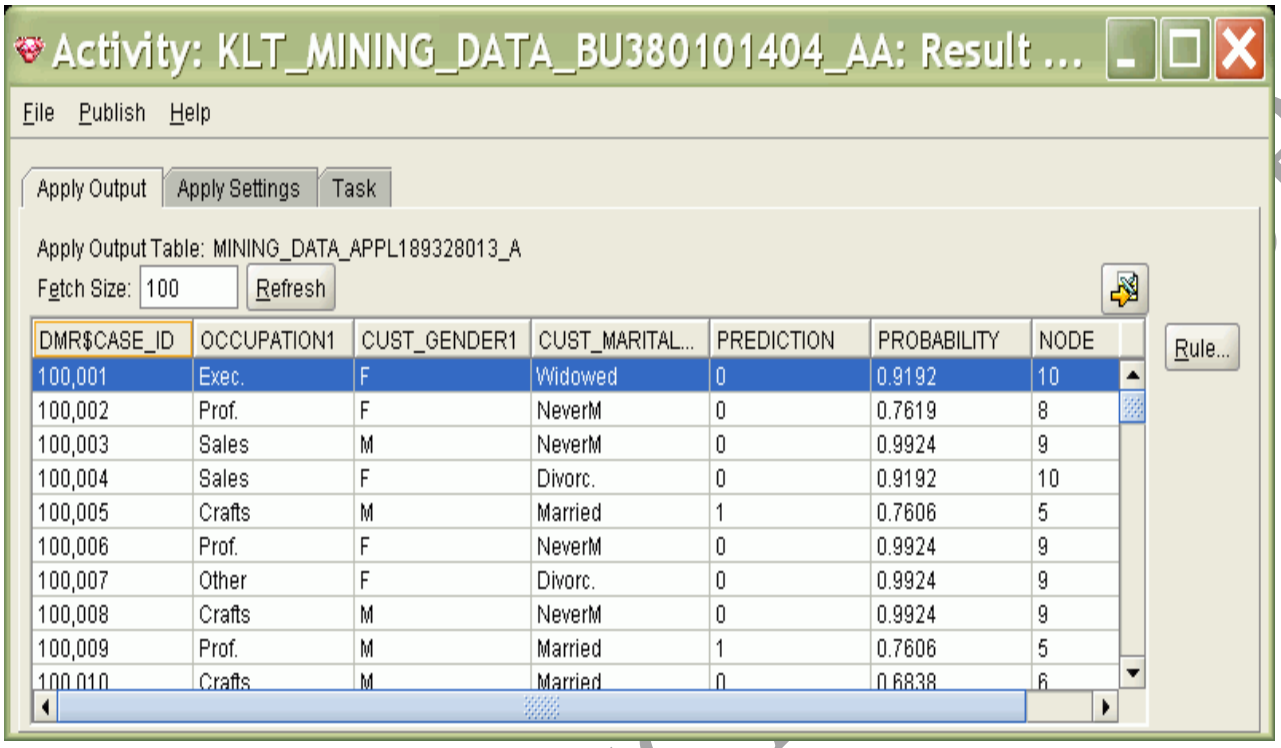
CUST_ID	CUST_GENDER	EDUCATION	OCCUPATION	AGE	AFFINITY_CARD
101501	F	Masters	Prof.	41	0
101502	M	Bach.	Sales	27	0
101503	F	HS-grad	Cleric.	20	0
101504	M	Bach.	Exec.	45	1
101505	M	Masters	Sales	34	1
101506	M	HS-grad	Other	38	0
101507	M	< Bach.	Sales	28	0
101508	M	HS-grad	Sales	19	0
101509	M	Bach.	Other	52	0
101510	M	Bach.	Sales	27	1

After undergoing testing (see "Testing a Classification Model"), the model can be applied to the data set that you wish to mine.

Figure 5-2 shows some of the predictions generated when the model is applied to the customer data set provided with the Oracle Data Mining sample programs. It displays several of the predictors along with the prediction (1=will increase spending; 0=will not increase spending) and the probability of the prediction for each customer.

Figure 5-2 Classification Results in Oracle Data Miner

DIWAKAR EDUCATION HUB



DMR$CASE_ID	OCCUPATION1	CUST_GENDER1	CUST_MARITAL...	PREDICTION	PROBABILITY	NODE
100,001	Exec.	F	Widowed	0	0.9192	10
100,002	Prof.	F	NeverM	0	0.7619	8
100,003	Sales	M	NeverM	0	0.9924	9
100,004	Sales	F	Divorc.	0	0.9192	10
100,005	Crafts	M	Married	1	0.7606	5
100,006	Prof.	F	NeverM	0	0.9924	9
100,007	Other	F	Divorc.	0	0.9924	9
100,008	Crafts	M	NeverM	0	0.9924	9
100,009	Prof.	M	Married	1	0.7606	5
100,010	Crafts	M	Married	0	0.6838	6

Description of "Figure 5-2 Classification Results in Oracle Data Miner"

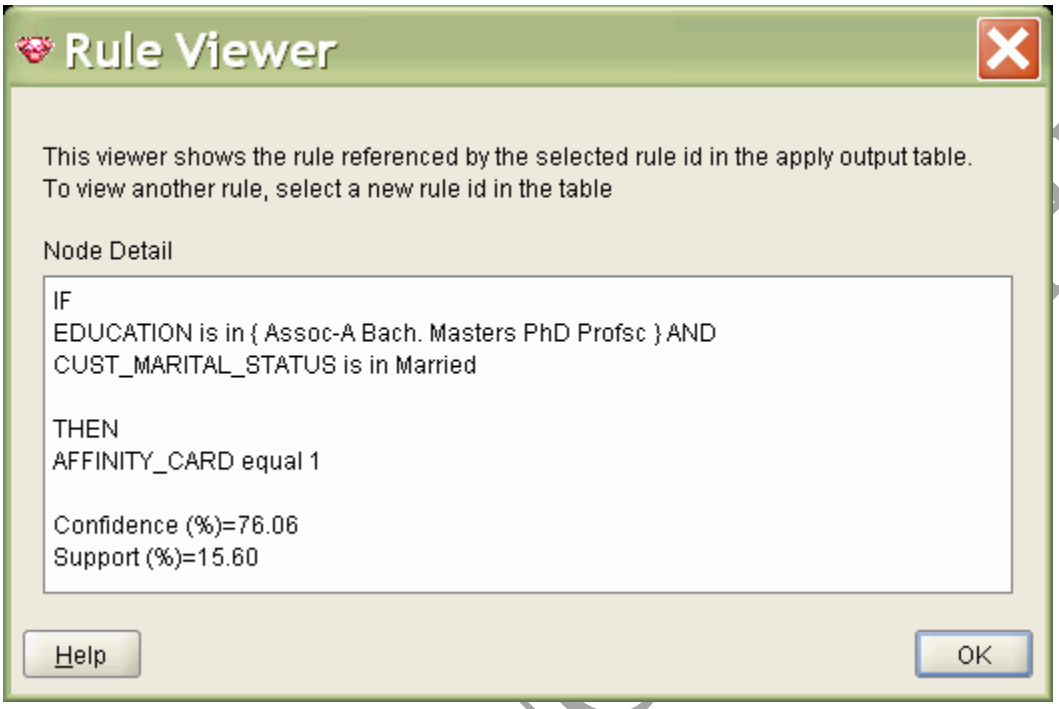
Note:

Oracle Data Miner displays the generalized case ID in the DMR$CASE_ID column of the apply output table. A "1" is appended to the column name of each predictor that you choose to include in the output. The predictions (affinity card usage in Figure 5-2) are displayed in the PREDICTION column. The probability of each prediction is displayed in the PROBABILITY column. For decision trees, the node is displayed in the NODE column.

Since this classification model uses the Decision Tree algorithm, rules are generated with the predictions and probabilities. With the Oracle Data Miner Rule Viewer, you can see the rule that produced a prediction for a given node in the tree. Figure 5-3 shows the rule for node 5. The rule states that married customers who have a college degree (Associates, Bachelor, Masters, Ph.D., or professional) are likely to increase spending with an affinity card.

DIWAKAR EDUCATION HUB

Figure 5-3 Decision Tree Rules for Classification



Description of "Figure 5-3 Decision Tree Rules for Classification"

Testing a Classification Model

A classification model is tested by applying it to test data with known target values and comparing the predicted values with the known values.

The test data must be compatible with the data used to build the model and must be prepared in the same way that the build data was prepared. Typically the build data and test data come from the same historical data set. A percentage of the records is used to build the model; the remaining records are used to test the model.

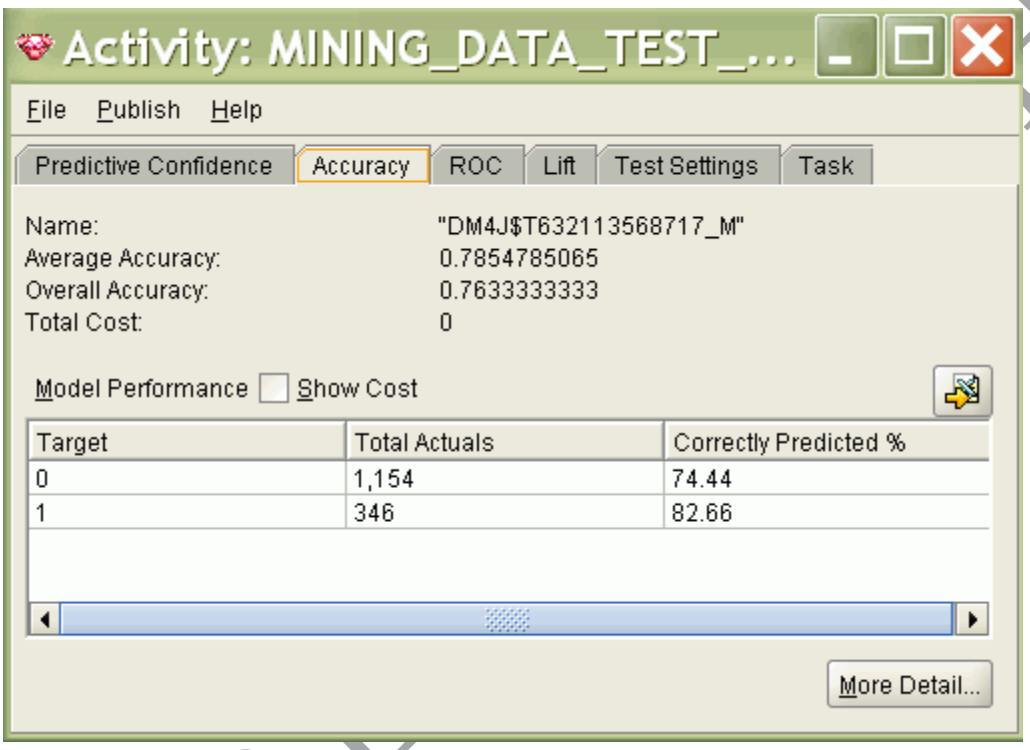
Test metrics are used to assess how accurately the model predicts the known values. If the model performs well and meets the business requirements, it can then be applied to new data to predict the future.

Accuracy

DIWAKAR EDUCATION HUB

Accuracy refers to the percentage of correct predictions made by the model when compared with the actual classifications in the test data. Figure 5-4 shows the accuracy of a binary classification model in Oracle Data Miner.

Figure 5-4 Accuracy of a Binary Classification Model



Description of "Figure 5-4 Accuracy of a Binary Classification Model"

Confusion Matrix

A confusion matrix displays the number of correct and incorrect predictions made by the model compared with the actual classifications in the test data. The matrix is *n*-by-*n*, where *n* is the number of classes.

Figure 5-5 shows a confusion matrix for a binary classification model. The rows present the number of actual classifications in the test data. The columns present the number of predicted classifications made by the model.

Figure 5-5 Confusion Matrix for a Binary Classification Model

		PREDICTED CLASS	
		affinity_card = 1	affinity_card = 0
ACTUAL CLASS	affinity_card = 1	516	25
	affinity_card = 0	10	725

Description of "Figure 5-5 Confusion Matrix for a Binary Classification Model"

In this example, the model correctly predicted the positive class for affinity_card 516 times and incorrectly predicted it 25 times. The model correctly predicted the negative class for affinity_card 725 times and incorrectly predicted it 10 times. The following can be computed from this confusion matrix:

- The model made 1241 **correct predictions** (516 + 725).
- The model made 35 **incorrect predictions** (25 + 10).
- There are 1276 **total scored cases** (516 + 25 + 10 + 725).
- The **error rate** is 35/1276 = 0.0274.
- The **overall accuracy rate** is 1241/1276 = 0.9725.

Clustering

Clustering analysis finds clusters of data objects that are similar in some sense to one another. The members of a cluster are more like each other than they are like members of other clusters. The goal of clustering analysis is to find high-quality clusters such that the inter-cluster similarity is low and the intra-cluster similarity is high.

Clustering, like classification, is used to segment the data. Unlike classification, clustering models segment data into groups that were not previously defined. Classification models segment data by assigning it to previously-defined classes, which are specified in a target. Clustering models do not use a target.

DIWAKAR EDUCATION HUB

Clustering is useful for exploring data. If there are many cases and no obvious groupings, clustering algorithms can be used to find natural groupings. Clustering can also serve as a useful data-preprocessing step to identify homogeneous groups on which to build supervised models.

Clustering can also be used for anomaly detection. Once the data has been segmented into clusters, you might find that some cases do not fit well into any clusters. These cases are anomalies or outliers.

Interpreting Clusters

Since known classes are not used in clustering, the interpretation of clusters can present difficulties. How do you know if the clusters can reliably be used for business decision making?

You can analyze clusters by examining information generated by the clustering algorithm. Oracle Data Mining generates the following information about each cluster:

- **Position in the cluster hierarchy**, described in "Cluster Rules"
- **Rule for the position in the hierarchy**, described in "Cluster Rules"
- **Attribute histograms**, described in "Attribute Histograms"
- **Cluster centroid**, described in "Centroid of a Cluster"

As with other forms of data mining, the process of clustering may be iterative and may require the creation of several models. The removal of irrelevant attributes or the introduction of new attributes may improve the quality of the segments produced by a clustering model.

How are Clusters Computed?

There are several different approaches to the computation of clusters. Clustering algorithms may be characterized as:

DIWAKAR EDUCATION HUB

- Hierarchical — Groups data objects into a hierarchy of clusters. The hierarchy can be formed top-down or bottom-up. Hierarchical methods rely on a distance function to measure the similarity between clusters.

Note:

The clustering algorithms supported by Oracle Data Mining perform hierarchical clustering.

- Partitioning — Partitions data objects into a given number of clusters. The clusters are formed in order to optimize an objective criterion such as distance.
- Locality-based — Groups neighboring data objects into clusters based on local conditions.
- Grid-based — Divides the input space into hyper-rectangular cells, discards the low-density cells, and then combines adjacent high-density cells to form clusters.

Cluster Rules

Oracle Data Mining performs hierarchical clustering. The leaf clusters are the final clusters generated by the algorithm. Clusters higher up in the hierarchy are intermediate clusters.

Rules describe the data in each cluster. A rule is a conditional statement that captures the logic used to split a parent cluster into child clusters. A rule describes the conditions for a case to be assigned with some probability to a cluster. For example, the following rule applies to cases that are assigned to cluster 19:

IF

```
OCCUPATION in Cleric. AND OCCUPATION in Crafts

  AND OCCUPATION in Exec.
```

DIWAKAR EDUCATION HUB

AND OCCUPATION in Prof.

CUST_GENDER in M

COUNTRY_NAME in United States of America

CUST_MARITAL_STATUS in Married

AFFINITY_CARD in 1.0

EDUCATION in < Bach.

AND EDUCATION in Bach.

AND EDUCATION in HS-grad

AND EDUCATION in Masters

CUST_INCOME_LEVEL in B: 30,000 - 49,999

AND CUST_INCOME_LEVEL in E: 90,000 - 109,999

AGE lessOrEqual 0.7

AND AGE greaterOrEqual 0.2

THEN

Cluster equal 19.0

Support and Confidence

Support and confidence are metrics that describe the relationships between clustering rules and cases.

Support is the percentage of cases for which the rule holds.

Confidence is the probability that a case described by this rule will actually be assigned to the cluster.

Number of Clusters

DIWAKAR EDUCATION HUB

The CLUS_NUM_CLUSTERS build setting specifies the maximum number of clusters that can be generated by a clustering algorithm.

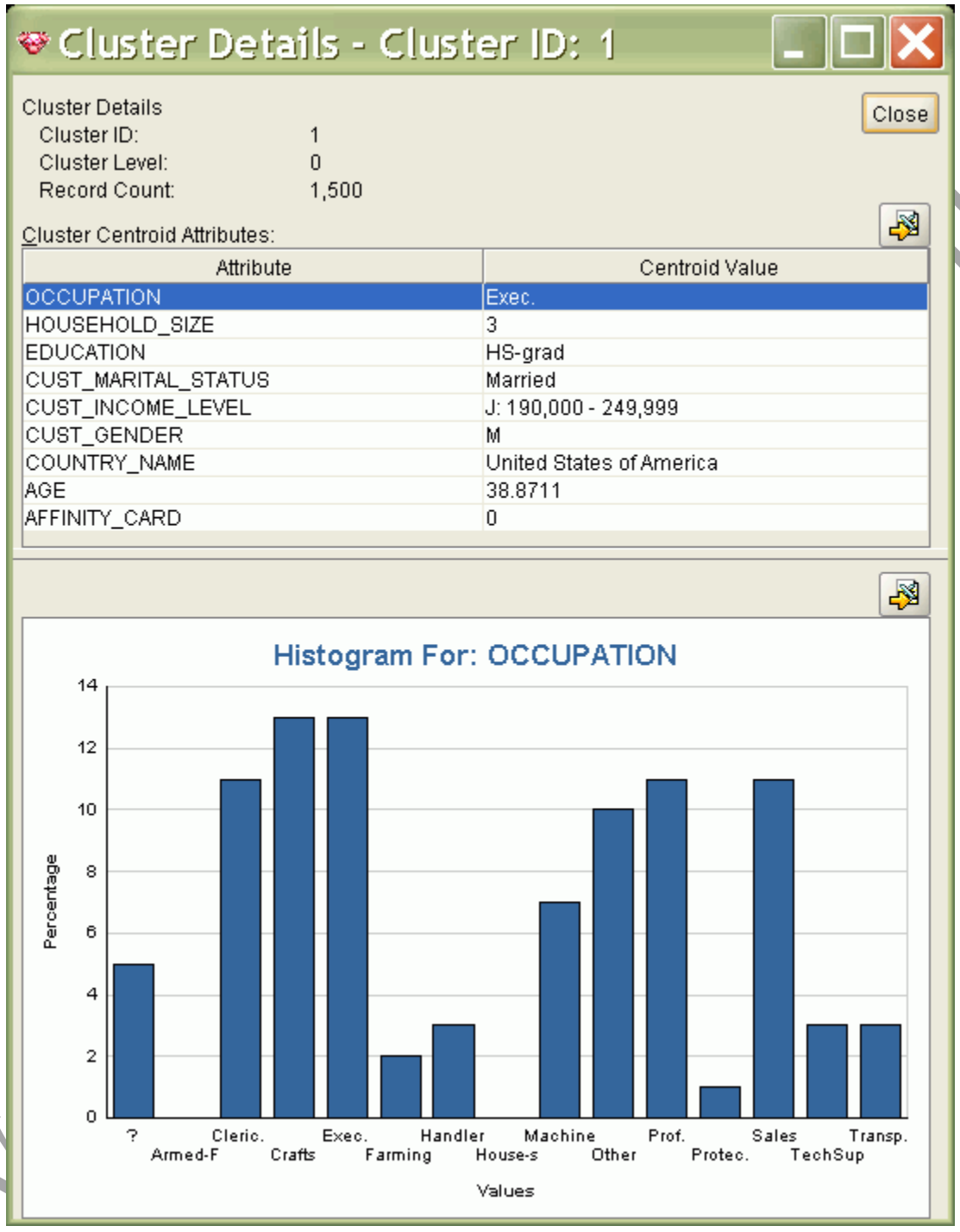
Attribute Histograms

In Oracle Data Miner, a **histogram** represents the distribution of the values of an attribute in a cluster. Figure 7-1 shows a histogram for the distribution of occupations in a cluster of customer data.

In this cluster, about 13% of the customers are craftsmen; about 13% are executives, 2% are farmers, and so on. None of the customers in this cluster are in the armed forces or work in housing sales.

Figure 7-1 Histogram in Oracle Data Miner

DIWAKAR EDUCATION HUB



Description of "Figure 7-1 Histogram in Oracle Data Miner"

Centroid of a Cluster

The **centroid** represents the most typical case in a cluster. For example, in a data set of customer ages and incomes, the centroid of each cluster would be a customer of average age and average income in that cluster. If the data set included gender, the centroid would have the gender most frequently represented in the cluster. Figure 7-1 shows the centroid values for a cluster.

The centroid is a prototype. It does not necessarily describe any given case assigned to the cluster. The attribute values for the centroid are the mean of the numerical attributes and the mode of the categorical attributes.

Scoring New Data

Oracle Data Mining supports the scoring operation for clustering. In addition to generating clusters from the build data, clustering models create a Bayesian probability model that can be used to score new data.

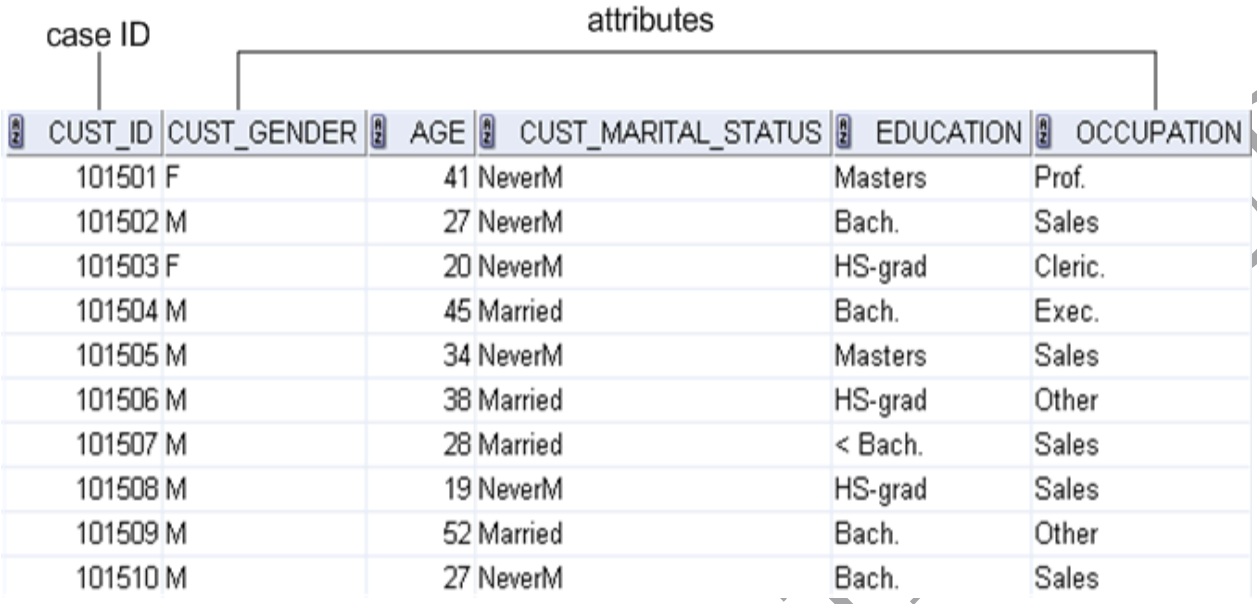
Sample Clustering Problems

These examples use the clustering model km_sh_clus_sample, created by one of the Oracle Data Mining sample programs, to show how clustering might be used to find natural groupings in the build data or to score new data.

Figure 7-2 shows six columns and ten rows from the case table used to build the model. Note that no column is designated as a target.

Figure 7-2 Build Data for Clustering

DIWAKAR EDUCATION HUB



CUST_ID	CUST_GENDER	AGE	CUST_MARITAL_STATUS	EDUCATION	OCCUPATION
101501	F	41	NeverM	Masters	Prof.
101502	M	27	NeverM	Bach.	Sales
101503	F	20	NeverM	HS-grad	Cleric.
101504	M	45	Married	Bach.	Exec.
101505	M	34	NeverM	Masters	Sales
101506	M	38	Married	HS-grad	Other
101507	M	28	Married	< Bach.	Sales
101508	M	19	NeverM	HS-grad	Sales
101509	M	52	Married	Bach.	Other
101510	M	27	NeverM	Bach.	Sales

Regression

Regression is a data mining function that predicts a number. Profit, sales, mortgage rates, house values, square footage, temperature, or distance could all be predicted using regression techniques. For example, a regression model could be used to predict the value of a house based on location, number of rooms, lot size, and other factors.

A regression task begins with a data set in which the target values are known. For example, a regression model that predicts house values could be developed based on observed data for many houses over a period of time. In addition to the value, the data might track the age of the house, square footage, number of rooms, taxes, school district, proximity to shopping centers, and so on. House value would be the target, the other attributes would be the predictors, and the data for each house would constitute a case.

In the model build (training) process, a regression algorithm estimates the value of the target as a function of the predictors for each case in the build data. These relationships between predictors and target are summarized in a model, which can then be applied to a different data set in which the target values are unknown.

DIWAKAR EDUCATION

Regression models are tested by computing various statistics that measure the difference between the predicted values and the expected values. The historical data for a regression project is typically divided into two data sets: one for building the model, the other for testing the model.

Regression modeling has many applications in trend analysis, business planning, marketing, financial forecasting, time series prediction, biomedical and drug response modeling, and environmental modeling.

How Does Regression Work?

You do not need to understand the mathematics used in regression analysis to develop and use quality regression models for data mining. However, it is helpful to understand a few basic concepts.

Regression analysis seeks to determine the values of parameters for a function that cause the function to best fit a set of data observations that you provide. The following equation expresses these relationships in symbols. It shows that regression is the process of estimating the value of a continuous target (*y*) as a function (*F*) of one or more predictors ($\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$), a set of parameters (θ_1 , θ_2 , ..., θ_n), and a measure of error (*e*).

$y = F(\mathbf{x},\theta) + e$

The predictors can be understood as independent variables and the target as a dependent variable. The error, also called the **residual**, is the difference between the expected and predicted value of the dependent variable. The regression parameters are also known as **regression coefficients**.

The process of training a regression model involves finding the parameter values that minimize a measure of the error, for example, the sum of squared errors.

There are different families of regression functions and different ways of measuring the error.

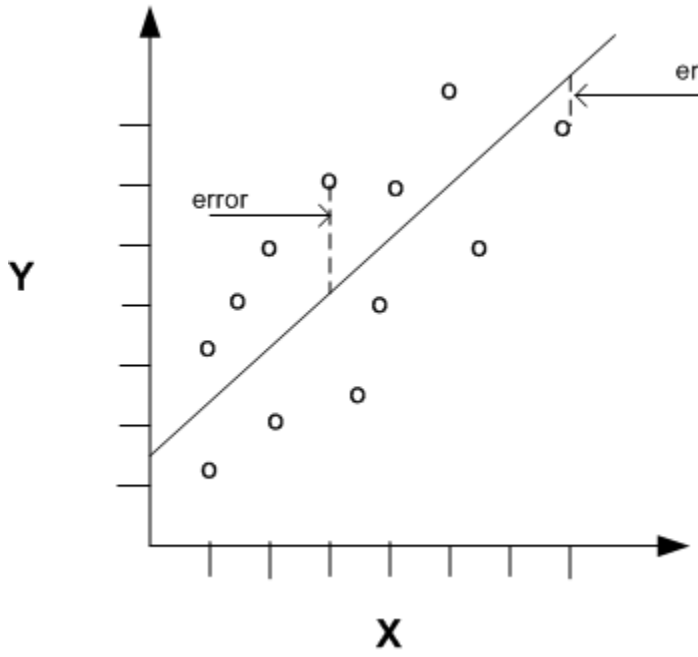
Linear Regression

DIWAKAR EDUCATION HUB

A linear regression technique can be used if the relationship between the predictors and the target can be approximated with a straight line.

Regression with a single predictor is the easiest to visualize. Simple linear regression with a single predictor is shown in Figure 4-1.

Figure 4-1 Linear Regression With a Single Predictor



Description of "Figure 4-1 Linear Regression With a Single Predictor"

Linear regression with a single predictor can be expressed with the following equation.

$y = \theta_2 \mathbf{x} + \theta_1 + e$

The regression parameters in simple linear regression are:

- The **slope** of the line (θ_2) — the angle between a data point and the regression line
- The ***y* intercept** (θ_1) — the point where **x** crosses the *y* axis (**x** = 0)

DIWAKAR EDUCATION HUB

Multivariate Linear Regression

The term **multivariate linear regression** refers to linear regression with two or more predictors ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$). When multiple predictors are used, the regression line cannot be visualized in two-dimensional space. However, the line can be computed simply by expanding the equation for single-predictor linear regression to include the parameters for each of the predictors.

$y = \theta_1 + \theta_2\mathbf{x}_1 + \theta_3\mathbf{x}_2 + ... \theta_n \mathbf{x}_{n-1} + e$

Regression Coefficients

In multivariate linear regression, the regression parameters are often referred to as coefficients. When you build a multivariate linear regression model, the algorithm computes a coefficient for each of the predictors used by the model. The coefficient is a measure of the impact of the predictor **x** on the target *y*. Numerous statistics are available for analyzing the regression coefficients to evaluate how well the regression line fits the data. ("Regression Statistics".)

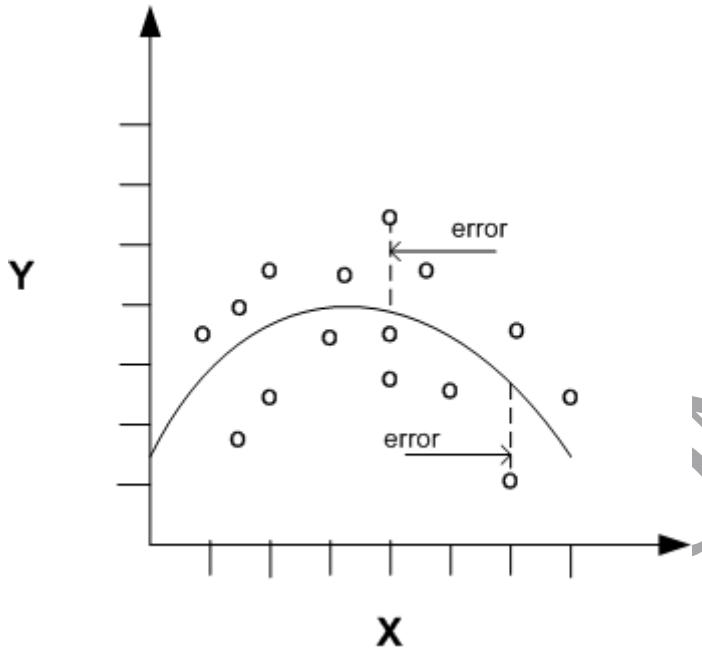
Nonlinear Regression

Often the relationship between **x** and *y* cannot be approximated with a straight line. In this case, a nonlinear regression technique may be used. Alternatively, the data could be preprocessed to make the relationship linear.

Nonlinear regression models define *y* as a function of **x** using an equation that is more complicated than the linear regression equation. In Figure 4-2, **x** and *y* have a nonlinear relationship.

Figure 4-2 Nonlinear Regression With a SIngle Predictor

DIWAKAR EDUCATION HUB



Description of "Figure 4-2 Nonlinear Regression With a SIngle Predictor"

Multivariate Nonlinear Regression

The term **multivariate nonlinear regression** refers to nonlinear regression with two or more predictors ($\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$). When multiple predictors are used, the nonlinear relationship cannot be visualized in two-dimensional space.

Confidence Bounds

A regression model predicts a numeric target value for each case in the scoring data. In addition to the predictions, some regression algorithms can identify confidence bounds, which are the upper and lower boundaries of an interval in which the predicted value is likely to lie.

When a model is built to make predictions with a given confidence, the confidence interval will be produced along with the predictions. For example, a model might predict the value of a house to be $500,000 with a 95% confidence that the value will be between $475,000 and $525,000.

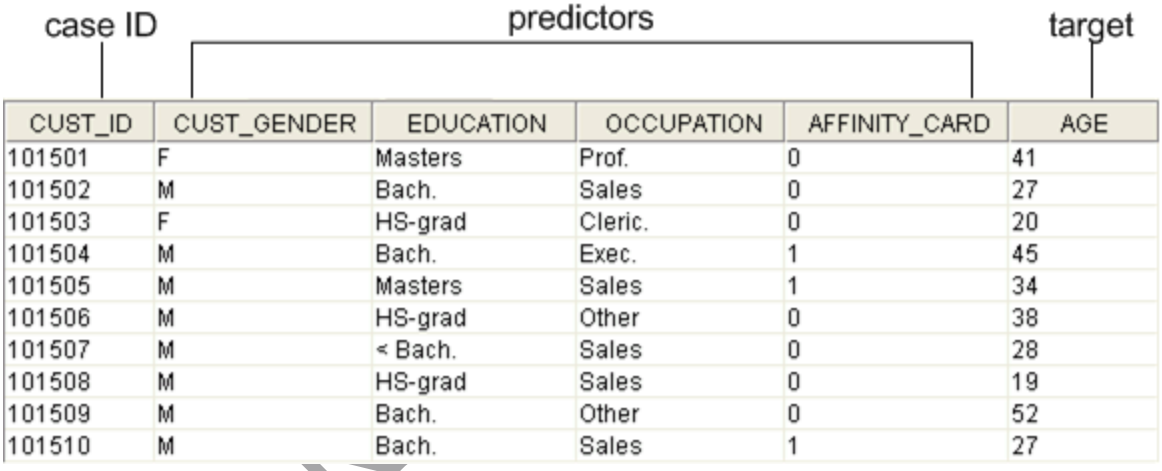
A Sample Regression Problem

DIWAKAR EDUCATION HUB

Suppose you want to learn more about the purchasing behavior of customers of different ages. You could build a model to predict the ages of customers as a function of various demographic characteristics and shopping patterns. Since the model will predict a number (age), we will use a regression algorithm.

This example uses the regression model, svmr_sh_regr_sample, which is created by one of the Oracle Data Mining sample programs. Figure 4-3 shows six columns and ten rows from the case table used to build the model.
The affinity_card column can contain either a 1, indicating frequent use of a preferred-buyer card, or a 0, which indicates no use or infrequent use.

Figure 4-3 Sample Build Data for Regression



CUST_ID	CUST_GENDER	EDUCATION	OCCUPATION	AFFINITY_CARD	AGE
101501	F	Masters	Prof.	0	41
101502	M	Bach.	Sales	0	27
101503	F	HS-grad	Cleric.	0	20
101504	M	Bach.	Exec.	1	45
101505	M	Masters	Sales	1	34
101506	M	HS-grad	Other	0	38
101507	M	< Bach.	Sales	0	28
101508	M	HS-grad	Sales	0	19
101509	M	Bach.	Other	0	52
101510	M	Bach.	Sales	1	27

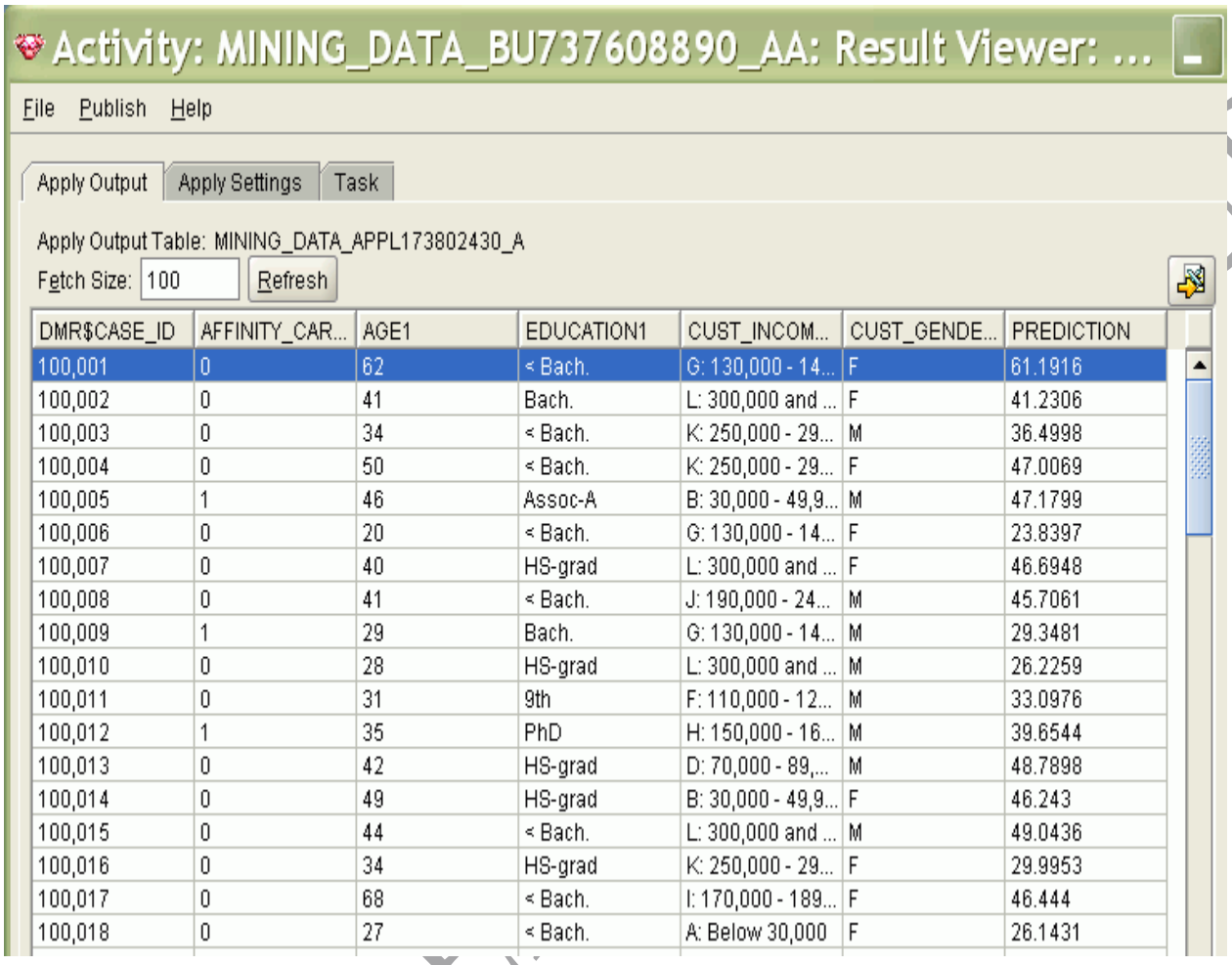
Description of "Figure 4-3 Sample Build Data for Regression"

After undergoing testing (see "Testing a Regression Model"), the model can be applied to the data set that you wish to mine.

Figure 4-4 shows some of the predictions generated when the model is applied to the customer data set provided with the Oracle Data Mining sample programs. Several of the predictors are displayed along with the predicted age for each customer.

Figure 4-4 Regression Results in Oracle Data Miner

DIWAKAR TION HUB



DMR$CASE_ID	AFFINITY_CAR...	AGE1	EDUCATION1	CUST_INCOM...	CUST_GENDE...	PREDICTION
100,001	0	62	< Bach.	G: 130,000 - 14...	F	61.1916
100,002	0	41	Bach.	L: 300,000 and ...	F	41.2306
100,003	0	34	< Bach.	K: 250,000 - 29...	M	36.4998
100,004	0	50	< Bach.	K: 250,000 - 29...	F	47.0069
100,005	1	46	Assoc-A	B: 30,000 - 49,9...	M	47.1799
100,006	0	20	< Bach.	G: 130,000 - 14...	F	23.8397
100,007	0	40	HS-grad	L: 300,000 and ...	F	46.6948
100,008	0	41	< Bach.	J: 190,000 - 24...	M	45.7061
100,009	1	29	Bach.	G: 130,000 - 14...	M	29.3481
100,010	0	28	HS-grad	L: 300,000 and ...	M	26.2259
100,011	0	31	9th	F: 110,000 - 12...	M	33.0976
100,012	1	35	PhD	H: 150,000 - 16...	M	39.6544
100,013	0	42	HS-grad	D: 70,000 - 89,...	M	48.7898
100,014	0	49	HS-grad	B: 30,000 - 49,9...	F	46.243
100,015	0	44	< Bach.	L: 300,000 and ...	M	49.0436
100,016	0	34	HS-grad	K: 250,000 - 29...	F	29.9953
100,017	0	68	< Bach.	I: 170,000 - 189...	F	46.444
100,018	0	27	< Bach.	A: Below 30,000	F	26.1431

Description of "Figure 4-4 Regression Results in Oracle Data Miner"

Note:

Oracle Data Miner displays the generalized case ID in the DMR$CASE_ID column of the apply output table. A "1" is appended to the column name of each predictor that you choose to include in the output. The predictions (the predicted ages in Figure 4-4) are displayed in the PREDICTION column.

Testing a Regression Model

A regression model is tested by applying it to test data with known target values and comparing the predicted values with the known values.

DIWAKAR

The test data must be compatible with the data used to build the model and must be prepared in the same way that the build data was prepared. Typically the build data and test data come from the same historical data set. A percentage of the records is used to build the model; the remaining records are used to test the model.

Test metrics are used to assess how accurately the model predicts these known values. If the model performs well and meets the business requirements, it can then be applied to new data to predict the future.

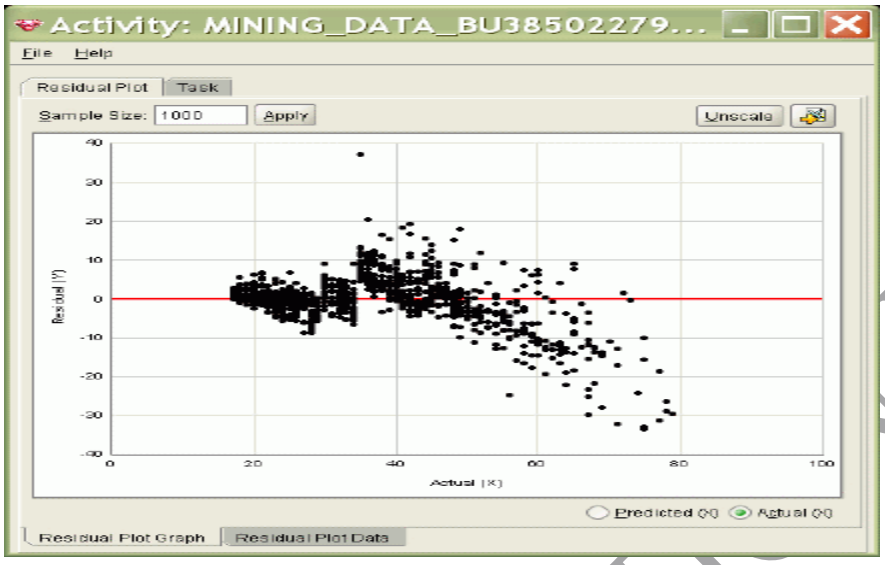
Residual Plot

A residual plot is a scatter plot where the x-axis is the predicted value of x, and the y-axis is the residual for x. The residual is the difference between the actual value of x and the predicted value of x.

Figure 4-5 shows a residual plot for the regression results shown in Figure 4-4. Note that most of the data points are clustered around 0, indicating small residuals. However, the distance between the data points and 0 increases with the value of x, indicating that the model has greater error for people of higher ages.

Figure 4-5 Residual Plot in Oracle Data Miner

DIWAKAR EDUCATION HUB



Description of "Figure 4-5 Residual Plot in Oracle Data Miner"

Regression Statistics

The Root Mean Squared Error and the Mean Absolute Error are commonly used statistics for evaluating the overall quality of a regression model. Different statistics may also be available depending on the regression methods used by the algorithm.

Root Mean Squared Error

The Root Mean Squared Error (RMSE) is the square root of the average squared distance of a data point from the fitted line.

This SQL expression calculates the RMSE.

SQRT(AVG((*predicted_value* - *actual_value*) * (*predicted_value* - *actual_value*)))

This formula shows the RMSE in mathematical symbols. The large sigma character represents summation; *j* represents the current predictor, and *n* represents the number of predictors.

DIWAKAR EDUCATION HUB

$$\mathbf{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Description of the illustration rmse

Mean Absolute Error

The Mean Absolute Error (MAE) is the average of the absolute value of the residuals (error). The MAE is very similar to the RMSE but is less sensitive to large errors.

This SQL expression calculates the MAE.

AVG(ABS(*predicted_value* - *actual_value*))

This formula shows the MAE in mathematical symbols. The large sigma character represents summation; *j* represents the current predictor, and *n* represents the number of predictors.

$$\mathbf{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

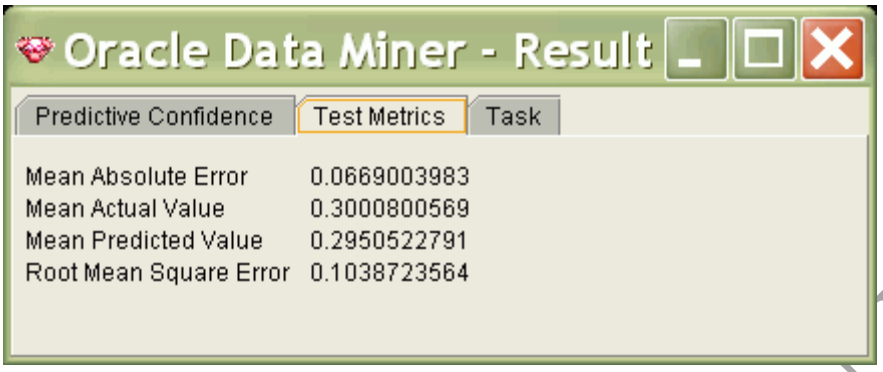
Description of the illustration mae

Test Metrics in Oracle Data Miner

Oracle Data Miner calculates the regression test metrics shown in Figure 4-6.

Figure 4-6 Test Metrics for a Regression Model

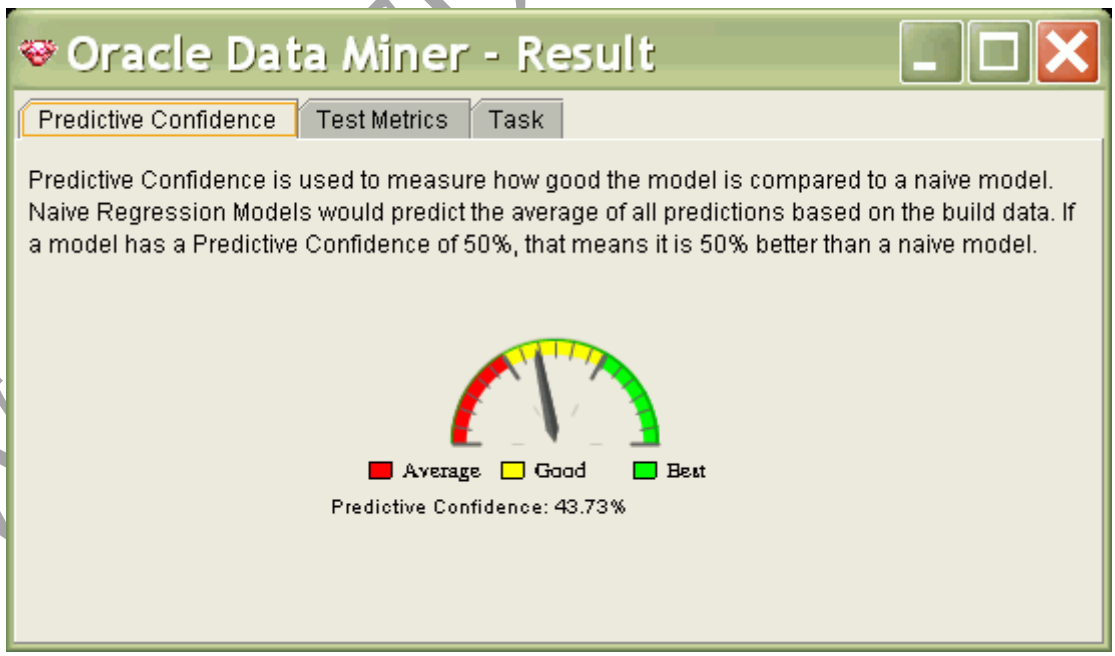
DIWAKAR EDUCATION HUB



Description of "Figure 4-6 Test Metrics for a Regression Model"

Oracle Data Miner calculates the predictive confidence for regression models. Predictive confidence is a measure of the improvement gained by the model over chance. If the model were "naive" and performed no analysis, it would simply predict the average. Predictive confidence is the percentage increase gained by the model over a naive model. Figure 4-7 shows a predictive confidence of 43%, indicating that the model is 43% better than a naive model.

Figure 4-7 Predictive Confidence for a Regression Model



Description of "Figure 4-7 Predictive Confidence for a Regression Model"

DIW EDUCATION HUB

Regression Algorithms

Oracle Data Mining supports two algorithms for regression. Both algorithms are particularly suited for mining data sets that have very high dimensionality (many attributes), including transactional and unstructured data.

- **Generalized Linear Models (GLM)**

GLM is a popular statistical technique for linear modeling. Oracle Data Mining implements GLM for regression and for binary classification.

GLM provides extensive coefficient statistics and model statistics, as well as row diagnostics. GLM also supports confidence bounds.

- **Support Vector Machines (SVM)**

SVM is a powerful, state-of-the-art algorithm for linear and nonlinear regression. Oracle Data Mining implements SVM for regression and other mining functions.

SVM regression supports two kernels: the Gaussian kernel for nonlinear regression, and the linear kernel for linear regression. SVM also supports active learning.

Support Vector Machines

Support Vector Machines (SVM) is a powerful, state-of-the-art algorithm with strong theoretical foundations based on the Vapnik-Chervonenkis theory. SVM has strong **regularization** properties. Regularization refers to the generalization of the model to new data.

Advantages of SVM

SVM models have similar functional form to neural networks and radial basis functions, both popular data mining techniques. However, neither of these algorithms has the well-founded theoretical approach to regularization that forms

DIWAKAR EDUCATION HUB

the basis of SVM. The quality of generalization and ease of training of SVM is far beyond the capacities of these more traditional methods.

SVM can model complex, real-world problems such as text and image classification, hand-writing recognition, and bioinformatics and biosequence analysis.

SVM performs well on data sets that have many attributes, even if there are very few cases on which to train the model. There is no upper limit on the number of attributes; the only constraints are those imposed by hardware. Traditional neural nets do not perform well under these circumstances.

Advantages of SVM in Oracle Data Mining

Oracle Data Mining has its own proprietary implementation of SVM, which exploits the many benefits of the algorithm while compensating for some of the limitations inherent in the SVM framework. Oracle Data Mining SVM provides the scalability and usability that are needed in a production quality data mining system.

Usability

Usability is a major enhancement, because SVM has often been viewed as a tool for experts. The algorithm typically requires data preparation, tuning, and optimization. Oracle Data Mining minimizes these requirements. You do not need to be an expert to build a quality SVM model in Oracle Data Mining. For example:

- Data preparation is not required in most cases.
- Default tuning parameters are generally adequate.

Scalability

When dealing with very large data sets, sampling is often required. However, sampling is not required with Oracle Data Mining SVM, because the algorithm itself uses stratified sampling to reduce the size of the training data as needed.

DIWAKAR EDUCATION HUB

Oracle Data Mining SVM is highly optimized. It builds a model incrementally by optimizing small working sets toward a global solution. The model is trained until convergence on the current working set, then the model adapts to the new data. The process continues iteratively until the convergence conditions are met. The Gaussian kernel uses caching techniques to manage the working sets.

Oracle Data Mining SVM supports **active learning**, an optimization method that builds a smaller, more compact model while reducing the time and memory resources required for training the model. See "Active Learning".

Kernel-Based Learning

SVM is a kernel-based algorithm. A **kernel** is a function that transforms the input data to a high-dimensional space where the problem is solved. Kernel functions can be linear or nonlinear.

Oracle Data Mining supports linear and Gaussian (nonlinear) kernels.

In Oracle Data Mining, the **linear kernel** function reduces to a linear equation on the original attributes in the training data. A linear kernel works well when there are many attributes in the training data.

The **Gaussian kernel** transforms each case in the training data to a point in an *n*-dimensional space, where *n* is the number of cases. The algorithm attempts to separate the points into subsets with homogeneous target values. The Gaussian kernel uses nonlinear separators, but within the kernel space it constructs a linear equation.

Active Learning

Active learning is an optimization method for controlling model growth and reducing model build time. Without active learning, SVM models grow as the size of the build data set increases, which effectively limits SVM models to small and medium size training sets (less than 100,000 cases). Active learning provides a

DIWAKAR EDUCATION HUB

way to overcome this restriction. With active learning, SVM models can be built on very large training sets.

Active learning forces the SVM algorithm to restrict learning to the most informative training examples and not to attempt to use the entire body of data. In most cases, the resulting models have predictive accuracy comparable to that of a standard (exact) SVM model.

Active learning provides a significant improvement in both linear and Gaussian SVM models, whether for classification, regression, or anomaly detection. However, active learning is especially advantageous for the Gaussian kernel, because nonlinear models can otherwise grow to be very large and can place considerable demands on memory and other system resources.

Tuning an SVM Model

SVM has built-in mechanisms that automatically choose appropriate settings based on the data. You may need to override the system-determined settings for some domains.

The build settings described in Table 18-1 are available for configuring SVM models. Settings pertain to regression, classification, and anomaly detection unless otherwise specified.

Table 18-1 Build Settings for Support Vector Machines

Setting Name	Configures....	Description
SVMS_KERNEL_FUNCTION	Kernel	Linear or Gaussian. The algorithm automatically uses the kernel function that is most appropriate to the

DIWAKAR EDUCATION HUB

Setting Name	Configures....	Description
		data. SVM uses the linear kernel when there are many attributes (more than 100) in the training data, otherwise it uses the Gaussian kernel.. The number of attributes does not correspond to the number of columns in the training data. SVM explodes categorical attributes to binary, numeric attributes. In addition, Oracle Data Mining interprets each row in a nested column as a separate attribute.
SVMS_STD_DEV	Standard deviation for Gaussian kernel	Controls the spread of the Gaussian

Setting Name	Configures....	Description
		kernel function. SVM uses a data-driven approach to find a standard deviation value that is on the same scale as distances between typical cases.
SVMS_KERNEL_CACHE_SIZE	Cache size for Gaussian kernel	Amount of memory allocated to the Gaussian kernel cache maintained in memory to improve model build time. The default cache size is 50 MB.
SVMS_ACTIVE_LEARNING	Active learning	Whether or not to use active learning. This setting is especially important for nonlinear (Gaussian) SVM models. By default, active

Setting Name	Configures....	Description
		learning is enabled.
SVMS_COMPLEXITY_FACTOR	Complexity factor	Regularization setting that balances the complexity of the model against model robustness to achieve good generalization on new data. SVM uses a data-driven approach to finding the complexity factor.
SVMS_CONVERGENCE_TOLERANCE	Convergence tolerance	The criterion for completing the model training process. The default is 0.001.
SVMS_EPSILON	Epsilon factor for regression	Regularization setting for regression, similar to complexity factor. Epsilon specifies the allowable residuals, or noise, in the data.

Setting Name	Configures....	Description
SVMS_OUTLIER_RATE	Outliers for anomaly detection	The expected outlier rate in anomaly detection. The default rate is 0.1.

Data Preparation for SVM

The SVM algorithm operates natively on numeric attributes. The algorithm automatically "explodes" categorical data into a set of binary attributes, one per category value. For example, a character column for marital status with values married or single would be transformed to two numeric attributes: married and single. The new attributes could have the value 1 (true) or 0 (false).

When there are missing values in columns with simple data types (not nested), SVM interprets them as missing at random. The algorithm automatically replaces missing categorical values with the mode and missing numerical values with the mean.

When there are missing values in nested columns, SVM interprets them as sparse. The algorithm automatically replaces sparse numerical data with zeros and sparse categorical data with zero vectors.

Normalization

SVM requires the normalization of numeric input. Normalization places the values of numeric attributes on the same scale and prevents attributes with a large original scale from biasing the solution. Normalization also minimizes the likelihood of overflows and underflows. Furthermore, normalization brings the numerical attributes to the same scale (0,1) as the exploded categorical data.

DIWAKAR EDUCATION HUB

SVM and Automatic Data Preparation

The SVM algorithm automatically handles missing value treatment and the transformation of categorical data, but normalization and outlier detection must be handled by ADP or prepared manually. ADP performs min-max normalization for SVM.

Note:

Oracle Corporation recommends that you use Automatic Data Preparation with SVM. The transformations performed by ADP are appropriate for most models.

SVM Classification

SVM classification is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. SVM finds the vectors ("support vectors") that define the separators giving the widest separation of classes.

SVM classification supports both binary and multiclass targets.

Class Weights

In SVM classification, weights are a biasing mechanism for specifying the relative importance of target values (classes).

SVM models are automatically initialized to achieve the best average prediction across all classes. However, if the training data does not represent a realistic distribution, you can bias the model to compensate for class values that are under-represented. If you increase the weight for a class, the percent of correct predictions for that class should increase.

The Oracle Data Mining APIs use priors to specify class weights for SVM. To use priors in training a model, you create a priors table and specify its name as a build setting for the model.

DIWAKAR EDUCATION HUB

Priors are associated with probabilistic models to correct for biased sampling procedures. SVM uses priors as a weight vector that biases optimization and favors one class over another.

One-Class SVM

Oracle Data Mining uses SVM as the one-class classifier for anomaly detection. When SVM is used for anomaly detection, it has the classification mining function but no target.

One-class SVM models, when applied, produce a prediction and a probability for each case in the scoring data. If the prediction is 1, the case is considered typical. If the prediction is 0, the case is considered anomalous. This behavior reflects the fact that the model is trained with normal data.

You can specify the percentage of the data that you expect to be anomalous with the SVMS_OUTLIER_RATE build setting. If you have some knowledge that the number of ÒsuspiciousÓ cases is a certain percentage of your population, then you can set the outlier rate to that percentage. The model will identify approximately that many ÒrareÓ cases when applied to the general population. The default is 10%, which is probably high for many anomaly detection problems.

SVM Regression

SVM uses an epsilon-insensitive loss function to solve regression problems.

SVM regression tries to find a continuous function such that the maximum number of data points lie within the epsilon-wide insensitivity tube. Predictions falling within epsilon distance of the true target value are not interpreted as errors.

The epsilon factor is a regularization setting for SVM regression. It balances the margin of error with model robustness to achieve the best generalization to new data.

DIWAKAR EDUCATION HUB

K Nearest Neighbors - Classification

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

Algorithm

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of

DI JCATION HUB

standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

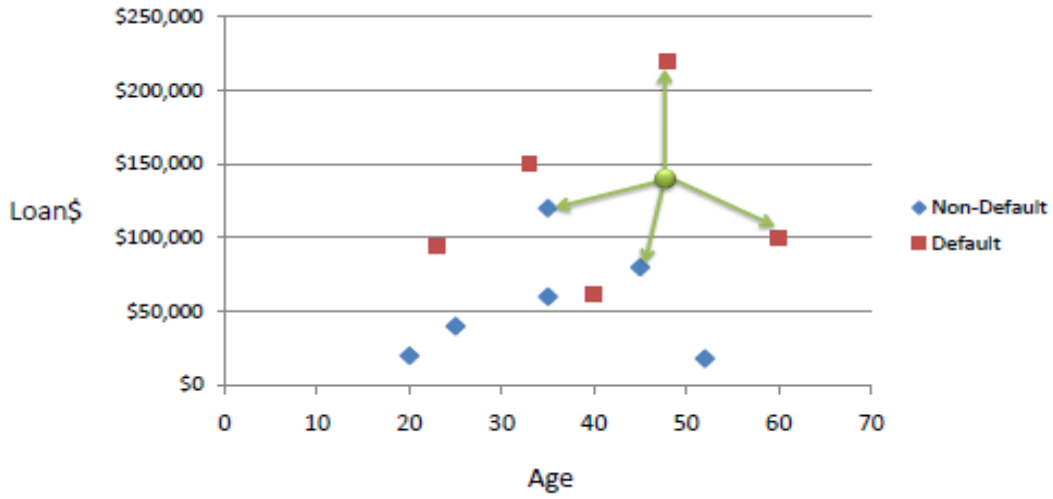
X	Y	Distance
Male	Male	0
Male	Female	1

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

***Example*:**

Consider the following data concerning credit default. Age and Loan are two numerical variables (predictors) and Default is the target.

DIWAKAR EDUCATION HUB



We can now use the training set to classify an unknown case (Age=48 and Loan=$142,000) using Euclidean distance. If K=1 then the nearest neighbor is the last case in the training set with Default=Y.

D = Sqrt[(48-33)^2 + (142000-150000)^2] = 8000.01 >> Default=Y

DIWAKAR EDUCATION HUB

Age	Loan	Default	Distance	
25	$40,000	N	102000	
35	$60,000	N	82000	
45	$80,000	N	62000	
20	$20,000	N	122000	
35	$120,000	N	22000	2
52	$18,000	N	124000	
23	$95,000	Y	47000	
40	$62,000	Y	80000	
60	$100,000	Y	42000	3
48	$220,000	Y	78000	
33	$150,000	Y	8000	1
48	**$142,000**	**?**		

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

With K=3, there are two Default=Y and one Default=N out of three closest neighbors. The prediction for the unknown case is again Default=Y.

Standardized Distance

One major drawback in calculating distance measures directly from the training set is in the case where variables have different measurement scales or there is a mixture of numerical and categorical variables. For example, if one variable is based on annual income in dollars, and the other is based on age in years then income will have a much higher influence on the distance calculated. One solution is to standardize the training set as shown below.

DIWAKAR EDUCATION HUB

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	**0.61**	**?**	

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

Hidden Markov Model (HMM)

A hidden Markov model (HMM) is a kind of statistical model that is a variation on the Markov chain. In a hidden Markov model, there are "hidden" states, or unobserved, in contrast to a standard Markov chain where all states are visible to the observer. Hidden Markov models are used for machine learning and data mining tasks including speech, handwriting and gesture recognition.

Hidden Markov Model (HMM)

The hidden Markov model was developed by the mathematician L.E. Baum and his colleagues in the 1960s. Like the popular Markov chain, the hidden Markov model attempts to predict the future state of a variable using probabilities based on the current and past state. The key difference between a Markov chain and the hidden Markov model is that the state in the latter is not directly visible to an observer, even though the output is.

Hidden Markov models are used for machine learning and data mining tasks. Some of these include speech recognition, handwriting recognition, part-of-speech tagging and bioinformatics.

Dependency Modeling

DIWAKAR EDUCATION HUB

Dependency Modeling consists of finding a model which describes significant dependencies between variables.
Dependency models exist at two levels:

1. The *structural* level of the model specifies (often graphically) which variables are locally dependent on each other, and
2. The *quantitative* level of the model specifies the strengths of the dependencies using some numerical scale.

Link Analysis

Link analysis is a data analysis technique used in network theory that is used to evaluate the relationships or connections between network nodes. These relationships can be between various types of objects (nodes), including people, organizations and even transactions.

Link analysis is essentially a kind of knowledge discovery that can be used to visualize data to allow for better analysis, especially in the context of links, whether Web links or relationship links between people or between different entities. Link analysis is often used in search engine optimization as well as in intelligence, in security analysis and in market and medical research.

Link Analysis

Link analysis is literally about analyzing the links between objects, whether they are physical, digital or relational. This requires diligent data gathering. For example, in the case of a website where all of the links and backlinks that are present must be analyzed, a tool has to sift through all of the HTML codes and various scripts in the page and then follow all the links it finds in order to determine what sort of links are present and whether they are active or dead. This information can be very important for search engine optimization, as it allows the analyst to determine whether the search engine is actually able to find and index the website.

DIWAKAR EDUCATION HUB

In networking, link analysis may involve determining the integrity of the connection between each network node by analyzing the data that passes through the physical or virtual links. With the data, analysts can find bottlenecks and possible fault areas and are able to patch them up more quickly or even help with network optimization.

Link analysis has three primary purposes:

- Find matches for known patterns of interests between linked objects.
- Find anomalies by detecting violated known patterns.
- Find new patterns of interest (for example, in social networking and marketing and business intelligence).

Social Network Analysis (SNA)

Social network analysis (SNA) is a process of quantitative and qualitative analysis of a social network. SNA measures and maps the flow of relationships and relationship changes between knowledge-possessing entities. Simple and complex entities include websites, computers, animals, humans, groups, organizations and nations.

The SNA structure is made up of node entities, such as humans, and ties, such as relationships. The advent of modern thought and computing facilitated a gradual evolution of the social networking concept in the form of highly complex, graph-based networks with many types of nodes and ties. These networks are the key to procedures and initiatives involving problem solving, administration and operations.

Social Network Analysis (SNA)

SNA usually refers to varied information and knowledge entities, but most actual studies focus on human (node) and relational (tie) analysis. The tie value is social

DIWAKAR EDUCATION HUB

capital.

SNA is often diagrammed with points (nodes) and lines (ties) to present the intricacies related to social networking. Professional researchers perform analysis using software and unique theories and methodologies.

SNA research is conducted in either of the following ways:

- Studying the complete social network, including all ties in a defined population.
- Studying egocentric components, including all ties and personal communities, which involves studying relationship between the focal points in the network and the social ties they make in their communities.

A snowball network forms when alters become egos and can create, or nominate, additional alters. Conducting snowball studies is difficult, due to logistical limitations. The abstract SNA concept is complicated further by studying hybrid networks, in which complete networks may create unlisted alters available for ego observation. Hybrid networks are analogous to employees affected by outside consultants, where data collection is not thoroughly defined.

Three analytical tendencies make SNA distinctive, as follows:

- Groups are not assumed to be societal building blocks.
- Studies focus on how ties affect individuals and other relationships, versus discrete individuals, organizations or states.
- Studies focus on structure, the composition of ties and how they affect societal norms, versus assuming that socialized norms determine behavior.

Sequence mining

DIWAKAR EDUCATION HUB

Sequence mining has already proven to be quite beneficial in many domains such as marketing analysis or Web click-stream analysis. A sequence s is defined as a set of ordered items denoted by $\langle s1,s2,\cdots,sn\rangle$. In activity recognition problems, the sequence is typically ordered using timestamps. The goal of sequence mining is to discover interesting patterns in data with respect to some subjective or objective measure of how interesting it is. Typically, this task involves discovering frequent sequential patterns with respect to a frequency support measure.

The task of discovering all the *frequent* sequences is not a trivial one. In fact, it can be quite challenging due to the combinatorial and exponential search space [19]. Over the past decade, a number of sequence mining methods have been proposed that handle the exponential search by using various heuristics. The first sequence mining algorithm was called GSP , which was based on the a priori approach for mining frequent itemsets. GSP makes several passes over the database to count the support of each sequence and to generate candidates. Then, it prunes the sequences with a support count below the minimum support.

Many other algorithms have been proposed to extend the GSP algorithm. One example is the PSP algorithm, which uses a prefix-based tree to represent candidate patterns [38]. FREESPAN [26] and PREFIXSPAN are among the first algorithms to consider a projection method for mining sequential patterns, by recursively projecting sequence databases into smaller projected databases. SPADE is another algorithm that needs only three passes over the database to discover sequential patterns. SPAM was the first algorithm to use a vertical bitmap representation of a database. Some other algorithms focus on discovering specific types of frequent patterns. For example, BIDE is an efficient algorithm for mining frequent closed sequences without candidate maintenance; there are also methods for constraint-based sequential pattern mining

Big Data

According to Gartner, the definition of Big Data –

DIWAKAR EDUCATION HUB

"Big data" is high-volume, velocity, and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making."

This definition clearly answers the "What is Big Data?" question – Big Data refers to complex and large data sets that have to be processed and analyzed to uncover valuable information that can benefit businesses and organizations.

However, there are certain basic tenets of Big Data that will make it even simpler to answer what is Big Data:

- It refers to a massive amount of data that keeps on growing exponentially with time.
- It is so voluminous that it cannot be processed or analyzed using conventional data processing techniques.
- It includes data mining, data storage, data analysis, data sharing, and data visualization.
- The term is an all-comprehensive one including data, data frameworks, along with the tools and techniques used to process and analyze the data.

Types of Big Data

Now that we are on track with what is big data, let's have a look at the types of big data:

Structured

Structured is one of the types of big data and By structured data, we mean data that can be processed, stored, and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms. **For instance, the employee table in a company database will be structured as the employee details, their job positions, their salaries, etc.,** will be present in an organized manner.

DIWAKAR EDUCATION HUB

Unstructured

Unstructured data refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyze unstructured data. Email is an example of unstructured data. Structured and unstructured are two important types of big data.

Semi-structured

Semi structured is the third type of big data. Semi-structured data pertains to the data containing both the formats mentioned above, that is, structured and unstructured data. To be precise, it refers to the data that although has not been classified under a particular repository (database), yet contains vital information or tags that segregate individual elements within the data. Thus we come to the end of types of data. Lets discuss the characteristics of data.

Characteristics of Big Data

Back in 2001, Gartner analyst Doug Laney listed the **3 'V's of Big Data – Variety, Velocity, and Volume.** Let's discuss the characteristics of big data.
These characteristics, isolatedly, are enough to know what is big data. Let's look at them in depth:

1) Variety

Variety of Big Data refers to structured, unstructured, and semistructured data that is gathered from multiple sources. While in the past, data could only be collected from spreadsheets and databases, today data comes in an array of forms such as emails, PDFs, photos, videos, audios, SM posts, and so much more. Variety is one of the important characteristics of big data.

2) Velocity

Velocity essentially refers to the speed at which data is being created in real-time. In a broader prospect, it comprises the rate of change, linking of incoming data sets at varying speeds, and activity bursts.

DIWAKAR EDUCATION HUB

3) Volume

Volume is one of the characteristics of big data. We already know that Big Data indicates huge 'volumes' of data that is being generated on a daily basis from various sources like social media platforms, business processes, machines, networks, human interactions, etc. Such a large amount of data are stored in data warehouses. Thus comes to the end of characteristics of big data.

Advantages of Big Data (Features)

- One of the biggest advantages of Big Data is predictive analysis. Big Data analytics tools can predict outcomes accurately, thereby, allowing businesses and organizations to make better decisions, while simultaneously optimizing their operational efficiencies and reducing risks.
- By harnessing data from social media platforms using Big Data analytics tools, businesses around the world are streamlining their digital marketing strategies to enhance the overall consumer experience. Big Data provides insights into the customer pain points and allows companies to improve upon their products and services.
- Being accurate, Big Data combines relevant data from multiple sources to produce highly actionable insights. Almost 43% of companies lack the necessary tools to filter out irrelevant data, which eventually costs them millions of dollars to hash out useful data from the bulk. Big Data tools can help reduce this, saving you both time and money.
- Big Data analytics could help companies generate more sales leads which would naturally mean a boost in revenue. Businesses are using Big Data analytics tools to understand how well their products/services are doing in the market and how the customers are responding to them. Thus, the can understand better where to invest their time and money.

DIWAKAR EDUCATION HUB

- With Big Data insights, you can always stay a step ahead of your competitors. You can screen the market to know what kind of promotions and offers your rivals are providing, and then you can come up with better offers for your customers. Also, Big Data insights allow you to learn customer behavior to understand the customer trends and provide a highly 'personalized' experience to them.

Who is using Big Data? 5 Applications

The people who're using Big Data know better that, what is Big Data. Let's look at some such industries:

1) Healthcare

Big Data has already started to create a huge difference in the healthcare sector. With the help of predictive analytics, medical professionals and HCPs are now able to provide personalized healthcare services to individual patients. Apart from that, fitness wearables, telemedicine, remote monitoring – all powered by Big Data and AI – are helping change lives for the better.

2) Academia

Big Data is also helping enhance education today. Education is no more limited to the physical bounds of the classroom – there are numerous online educational courses to learn from. Academic institutions are investing in digital courses powered by Big Data technologies to aid the all-round development of budding learners.

3) Banking

The banking sector relies on Big Data for fraud detection. Big Data tools can efficiently detect fraudulent acts in real-time such as misuse of credit/debit cards, archival of inspection tracks, faulty alteration in customer stats, etc.

4) Manufacturing

DIWAKAR EDUCATION HUB

According to TCS Global Trend Study, the most significant benefit of Big Data in manufacturing is improving the supply strategies and product quality. In the manufacturing sector, Big data helps create a transparent infrastructure, thereby, predicting uncertainties and incompetencies that can affect the business adversely.

5) IT

One of the largest users of Big Data, IT companies around the world are using Big Data to optimize their functioning, enhance employee productivity, and minimize risks in business operations. By combining Big Data technologies with ML and AI, the IT sector is continually powering innovation to find solutions even for the most complex of problems.

6. Retail

Big Data has changed the way of working in traditional brick and mortar retail stores. Over the years, retailers have collected vast amounts of data from local demographic surveys, POS scanners, RFID, customer loyalty cards, store inventory, and so on. Now, they've started to leverage this data to create personalized customer experiences, boost sales, increase revenue, and deliver outstanding customer service.

Retailers are even using smart sensors and Wi-Fi to track the movement of customers, the most frequented aisles, for how long customers linger in the aisles, among other things. They also gather social media data to understand what customers are saying about their brand, their services, and tweak their product design and marketing strategies accordingly.

7. Transportation

Big Data Analytics holds immense value for the transportation industry. In countries across the world, both private and government-run transportation companies use Big Data technologies to optimize route planning, control traffic,

DIWAKAR EDUCATION HUB

manage road congestion, and improve services. Additionally, transportation services even use Big Data to revenue management, drive technological innovation, enhance logistics, and of course, to gain the upper hand in the market.

Big Data Case studies

1. Walmart

Walmart leverages Big Data and *Data Mining* to create personalized product recommendations for its customers. With the help of these two emerging technologies, Walmart can uncover valuable patterns showing the most frequently bought products, most popular products, and even the most popular product bundles (products that complement each other and are usually purchased together).

Based on these insights, Walmart creates attractive and customized recommendations for individual users. By effectively implementing Data Mining techniques, the retail giant has successfully increased the conversion rates and improved its customer service substantially. Furthermore, Walmart uses *Hadoop* and NoSQL technologies to allow customers to access real-time data accumulated from disparate sources.

2. American Express

The credit card giant leverages enormous volumes of customer data to identify indicators that could depict user loyalty. It also uses Big Data to build advanced predictive models for analyzing historical transactions along with 115 different variables to predict potential customer churn. Thanks to Big Data solutions and tools, American Express can identify 24% of the accounts that are highly likely to close in the upcoming four to five months.

3. General Electric

DIWAKAR EDUCATION HUB

In the words of Jeff Immelt, Chairman of General Electric, in the past few years, GE has been successful in bringing together the best of both worlds – "the physical and analytical worlds." GE thoroughly utilizes Big Data. Every machine operating under General Electric generates data on how they work. The GE analytics team then crunches these colossal amounts of data to extract relevant insights from it and redesign the machines and their operations accordingly.

Today, the company has realized that even minor improvements, no matter how small, play a crucial role in their company infrastructure. According to GE stats, Big Data has the potential to boost productivity by 1.5% in the US, which compiled over a span of 20 years could increase the average national income by a staggering 30%!

4. Uber

Uber is one of the major cab service providers in the world. It leverages customer data to track and identify the most popular and most used services by the users. Once this data is collected, Uber uses data analytics to analyze the usage patterns of customers and determine which services should be given more emphasis and importance.

Apart from this, Uber uses Big Data in another unique way. Uber closely studies the demand and supply of its services and changes the cab fares accordingly. It is the surge pricing mechanism that works something like this – suppose when you are in a hurry, and you have to book a cab from a crowded location, Uber will charge you double the normal amount!

5. Netflix

Netflix is one of the most popular on-demand online video content streaming platform used by people around the world. Netflix is a major proponent of the recommendation engine. It collects customer data to understand the specific needs, preferences, and taste patterns of users. Then it uses this data to predict

DIWAKAR EDUCATION HUB

what individual users will like and create personalized content recommendation lists for them.

Today, Netflix has become so vast that it is even creating unique content for users. Data is the secret ingredient that fuels both its recommendation engines and new content decisions. The most pivotal data points used by Netflix include titles that users watch, user ratings, genres preferred, and how often users stop the playback, to name a few. Hadoop, Hive, and Pig are the three core components of the data structure used by Netflix.

6. Procter & Gamble

Procter & Gamble has been around us for ages now. However, despite being an "old" company, P&G is nowhere close to old in its ways. Recognizing the potential of Big Data, P&G started implementing Big Data tools and technologies in each of its business units all over the world. The company's primary focus behind using Big Data was to utilize real-time insights to drive smarter decision making.

To accomplish this goal, P&G started collecting vast amounts of structured and unstructured data across R&D, supply chain, customer-facing operations, and customer interactions, both from company repositories and online sources. The global brand has even developed Big Data systems and processes to allow managers to access the latest industry data and analytics.

7. IRS

Yes, even government agencies are not shying away from using Big Data. The US Internal Revenue Service actively uses Big Data to prevent identity theft, fraud, and untimely payments (people who should pay taxes but don't pay them in due time).

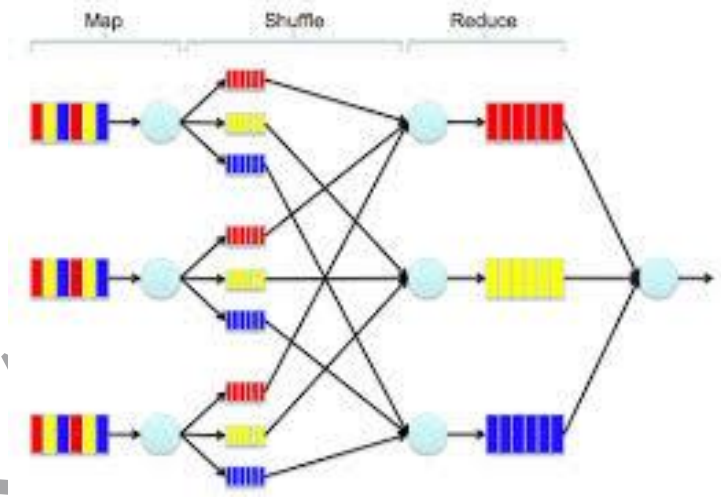
The IRS even harnesses the power of Big Data to ensure and enforce compliance with tax rules and laws. As of now, the IRS has successfully averted fraud and scams involving billions of dollars, especially in the case of identity theft. In the past three years, it has also recovered over US$ 2 billion.

DIWAKAR EDUCATION HUB

Introduction to MapReduce

MapReduce is a programming model for processing large data sets with a parallel , distributed algorithm on a cluster (source: Wikipedia). Map Reduce when coupled with HDFS can be used to handle big data. The fundamentals of this HDFS-MapReduce system, which is commonly referred to as Hadoop.

The basic unit of information, used in MapReduce is a (Key,value) pair. All types of structured and unstructured data need to be translated to this basic unit, before feeding the data to MapReduce model. As the name suggests, MapReduce model consist of two separate routines, namely Map-function and Reduce-function. This article will help you understand the step by step functionality of Map-Reduce model.The computation on an input (i.e. on a set of pairs) in MapReduce model occurs in three stages:



Step 1 : The map stage

Step 2 : The shuffle stage

Step 3 : The reduce stage.

Semantically, the map and shuffle phases distribute the data, and the reduce phase performs the computation. In this article we will discuss about each of these stages in detail.

DIWAK ATION HUB

[stextbox id="section"] The Map stage [/stextbox]

MapReduce logic, unlike other data frameworks, is not restricted to just structured datasets. It has an extensive capability to handle unstructured data as well. Map stage is the critical step which makes this possible. Mapper brings a structure to unstructured data. For instance, if I want to count the number of photographs on my laptop by the location (city), where the photo was taken, I need to analyze unstructured data. The mapper makes (key, value) pairs from this data set. In this case, key will be the location and value will be the photograph. After mapper is done with its task, we have a structure to the entire data-set.

In the map stage, the mapper takes a single (key, value) pair as input and produces any number of (key, value) pairs as output . It is important to think of the map operation as stateless, that is, its logic operates on a single pair at a time (even if in practice several input pairs are delivered to the same mapper). To summarize, for the map phase, the user simply designs a map function that maps an input (key, value) pair to any number (even none) of output pairs. Most of the time, the map phase is simply used to specify the desired location of the input value by changing its key.

[stextbox id="section"] The shuffle stage [/stextbox]

The shuffle stage is automatically handled by the MapReduce framework, i.e. the engineer has nothing to do for this stage. The underlying system implementing MapReduce routes all of the values that are associated with an individual key to the same reducer.

[stextbox id="section"] The Reduce stage [/stextbox]

In the reduce stage, the reducer takes all of the values associated with a single key k and outputs any number of (key, value) pairs. This highlights one of the sequential aspects of MapReduce computation: all of the maps need to finish before the reduce stage can begin. Since the reducer has access to all the values with the same key, it can perform sequential computations on these values. In the

DIWAKAR EDUCATION HUB

reduce step, the parallelism is exploited by observing that reducers operating on different keys can be executed simultaneously. To summarize, for the reduce phase, the user designs a function that takes in input a list of values associated with a single key and outputs any number of pairs. Often the output keys of a reducer equal the input key (in fact, in the original MapReduce paper the output key *must* equal to the input key, but Hadoop relaxed this constraint).

Overall, a program in the MapReduce paradigm can consist of many rounds (usually called *jobs*) of different map and reduce functions, performed sequentially one after another.

[stextbox id="section"] An example [/stextbox]

Let's consider an example to understand Map-Reduce in depth. We have the following 3 sentences :

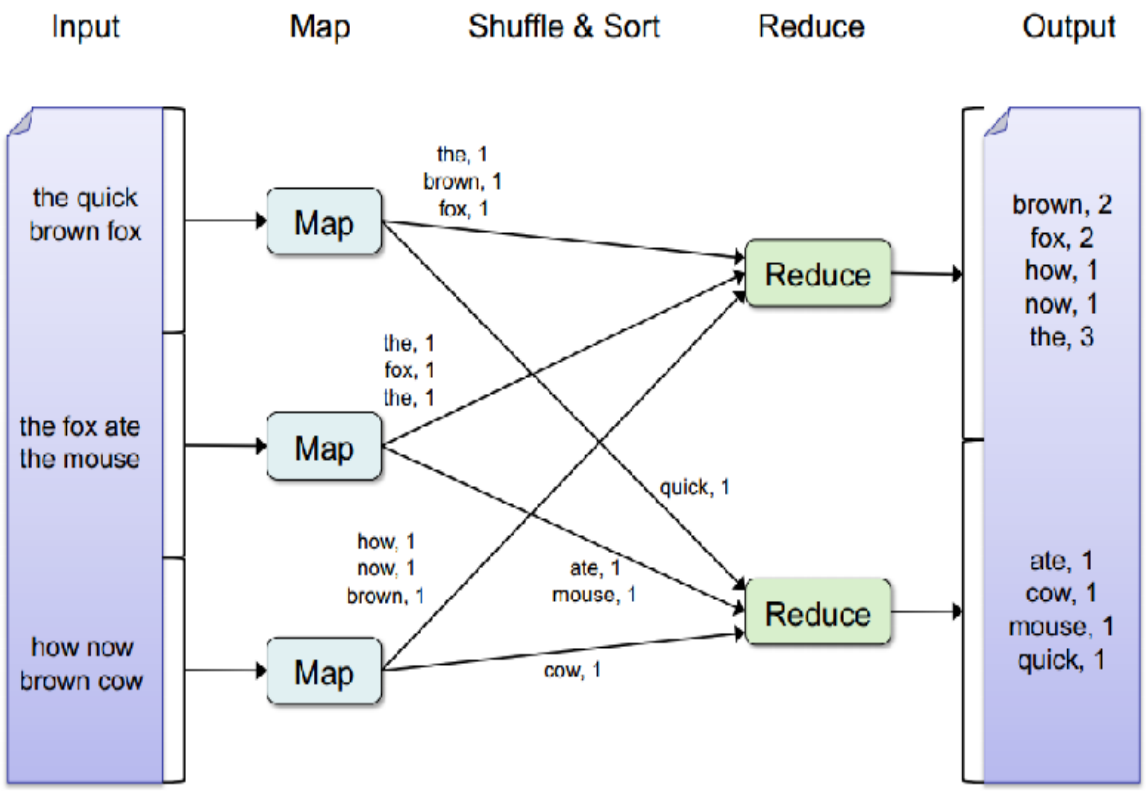
1. The quick brown fox

2. The fox ate the mouse

3. How now brown cow

Our objective is to count the frequency of each word in all the sentences. Imagine that each of these sentences acquire huge memory and hence are allotted to different data nodes. Mapper takes over this unstructured data and creates key value pairs. In this case key is the word and value is the count of this word in the text available at this data node. For instance, the 1st Map node generates 4 key-value pairs : (the,1), (brown,1),(fox,1), (quick,1). The first 3 key-value pairs go to the first Reducer and the last key-value go to the second Reducer.

DIWAKAR EDUCATION HUB



Similarly, the 2nd and 3rd map functions do the mapping for the other two sentences. Through shuffling, all the similar words come to the same end. Once, the key value pairs are sorted, the reducer function operates on this structured data to come up with a summary.

[stextbox id=”section”] End Notes : [/stextbox]

Let’s take some example of Map-Reduce function usage in the industry :

- **At Google:**

– Index building for Google Search
– Article clustering for Google News
– Statistical machine translation

- **At Yahoo!:**

– Index building for Yahoo! Search
– Spam detection for Yahoo! Mail

DIWAKAR EDU HUB

- **At Facebook:**

– Data mining
– Ad optimization
– Spam detection Example

- **At Amazon:**

– Product clustering
– Statistical machine translation

The constraint of using Map-reduce function is that user has to follow a logic format. This logic is to generate key-value pairs using Map function and then summarize using Reduce function. But luckily most of the data manipulation operations can be tricked into this format. In the next article we will take some example like how to do data-set merging, matrix multiplication, matrix transpose, etc. using Map-Reduce.

Introduction to Hadoop

Hadoop is a complete eco-system of open source projects that provide us the framework to deal with big data. Let's start by brainstorming the possible challenges of dealing with big data (on traditional systems) and then look at the capability of Hadoop solution.

Following are the challenges I can think of in dealing with big data :

1. High capital investment in procuring a server with high processing capacity.

2. Enormous time taken

3. In case of long query, imagine an error happens on the last step. You will waste so much time making these iterations.

4. Difficulty in program query building

Here is how Hadoop solves all of these issues :

DIWAKAR EDUCATION HUB

1. High capital investment in procuring a server with high processing capacity: Hadoop clusters work on normal commodity hardware and keep multiple copies to ensure reliability of data. A maximum of 4500 machines can be connected together using Hadoop.

2. Enormous time taken : The process is broken down into pieces and executed in parallel, hence saving time. A maximum of 25 Petabyte (1 PB = 1000 TB) data can be processed using Hadoop.

3. In case of long query, imagine an error happens on the last step. You will waste so much time making these iterations : Hadoop builds back up data-sets at every level. It also executes query on duplicate datasets to avoid process loss in case of individual failure. These steps makes Hadoop processing more precise and accurate.

4. Difficulty in program query building : Queries in Hadoop are as simple as coding in any language. You just need to change the way of thinking around building a query to enable parallel processing.

Background of Hadoop

With an increase in the penetration of internet and the usage of the internet, the data captured by Google increased exponentially year on year. Just to give you an estimate of this number, in 2007 Google collected on an average 270 PB of data every month. The same number increased to 20000 PB everyday in 2009. Obviously, Google needed a better platform to process such an enormous data. Google implemented a programming model called MapReduce, which could process this 20000 PB per day. Google ran these MapReduce operations on a special file system called Google File System (GFS). Sadly, GFS is not an open source.

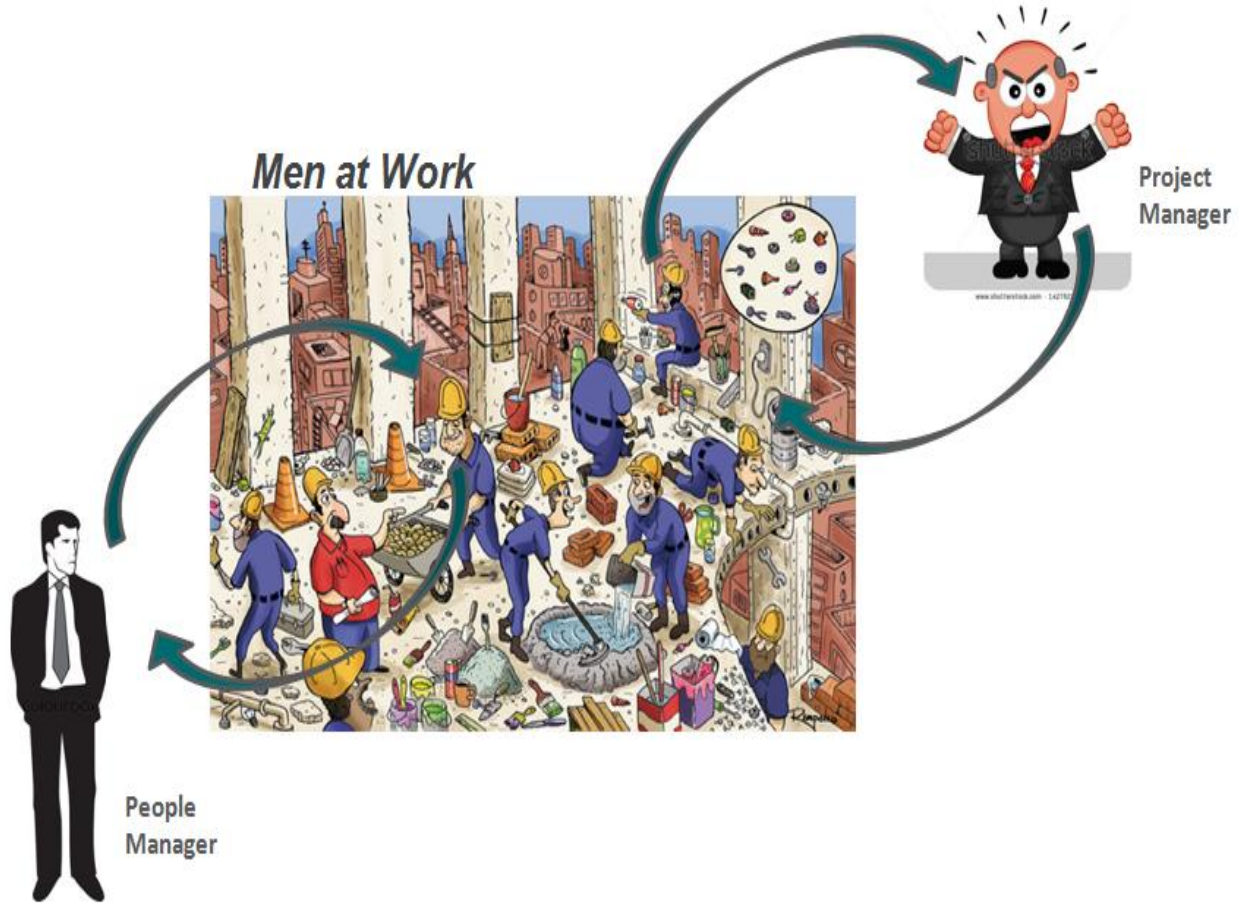
Doug cutting and Yahoo! reverse engineered the model GFS and built a parallel Hadoop Distributed File System (HDFS). The software or framework that supports

DIWAKAR EDUCATION HUB

HDFS and MapReduce is known as Hadoop. Hadoop is an open source and distributed by Apache.

Framework of Hadoop Processing

Let's draw an analogy from our daily life to understand the working of Hadoop. The bottom of the pyramid of any firm are the people who are individual contributors. They can be analyst, programmers, manual labors, chefs, etc. Managing their work is the project manager. The project manager is responsible for a successful completion of the task. He needs to distribute labor, smoothen the coordination among them etc. Also, most of these firms have a people manager, who is more concerned about retaining the head count.



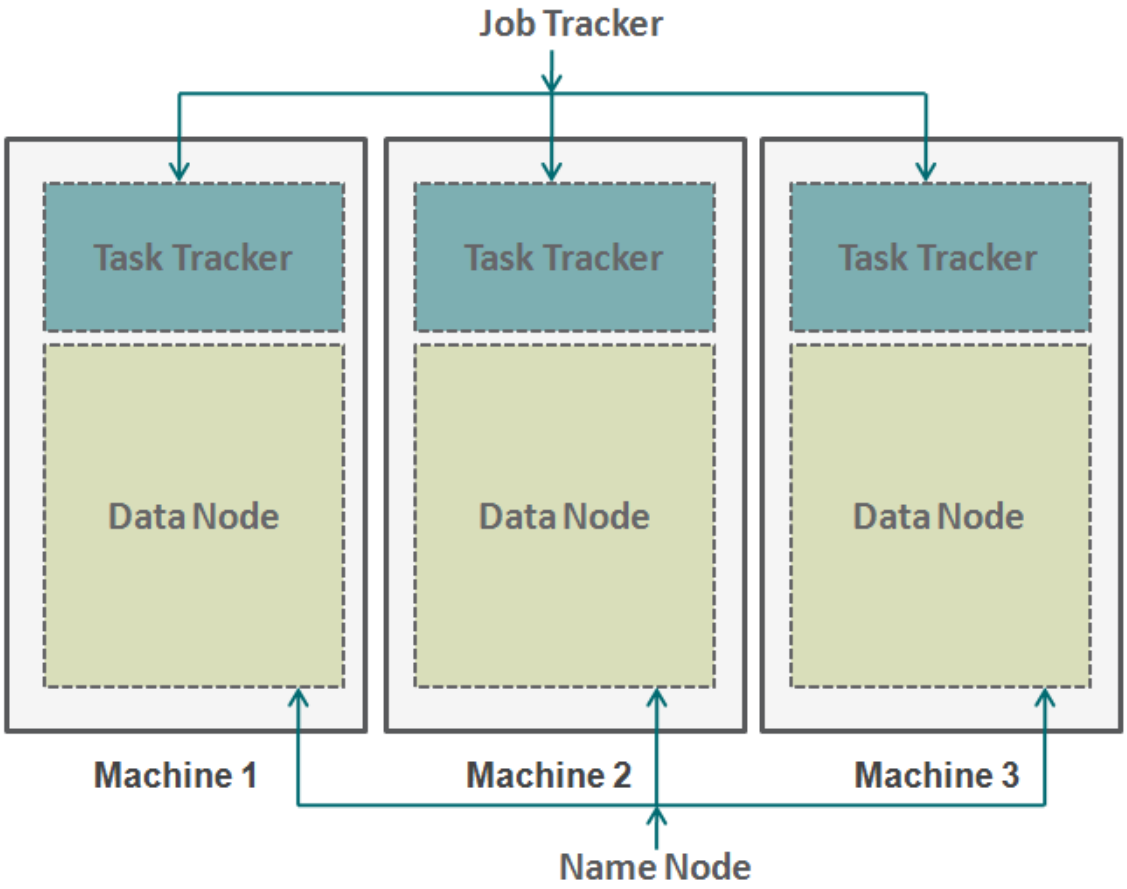
DIGITION HUB

Hadoop works in a similar format. On the bottom we have machines arranged in parallel. These machines are analogous to individual contributor in our analogy. Every machine has a data node and a task tracker. Data node is also known as HDFS (Hadoop Distributed File System) and Task tracker is also known as map-reducers.

Data node contains the entire set of data and Task tracker does all the operations. You can imagine task tracker as your arms and leg, which enables you to do a task and data node as your brain, which contains all the information which you want to process. These machines are working in silos and it is very essential to coordinate them. The Task trackers (Project manager in our analogy) in different machines are coordinated by a Job Tracker. Job Tracker makes sure that each operation is completed and if there is a process failure at any node, it needs to assign a duplicate task to some task tracker. Job tracker also distributes the entire task to all the machines.

A name node on the other hand coordinates all the data nodes. It governs the distribution of data going to each machine. It also checks for any kind of purging which have happened on any machine. If such purging happens, it finds the duplicate data which was sent to other data node and duplicates it again. You can think of this name node as the people manager in our analogy which is concerned more about the retention of the entire dataset.

DIWAKAR EDUCATION HUB



When not to use Hadoop ?

Till now, we have seen how Hadoop has made handling big data possible. But in some scenarios Hadoop implementation is not recommended. Following are some of those scenarios :

1. Low Latency data access : Quick access to small parts of data
2. Multiple data modification : Hadoop is a better fit only if we are primarily concerned about reading data and not writing data.
3. Lots of small files : Hadoop is a better fit in scenarios, where we have few but large files.

DIWAKAR EL

Distributed File System (DFS)

A distributed file system (DFS) is a file system with data stored on a server. The data is accessed and processed as if it was stored on the local client machine. The DFS makes it convenient to share information and files among users on a network in a controlled and authorized way. The server allows the client users to share files and store data just like they are storing the information locally. However, the servers have full control over the data and give access control to the clients.

Distributed File System (DFS)

There has been exceptional growth in network-based computing recently and client/server-based applications have brought revolutions in this area. Sharing storage resources and information on the network is one of the key elements in both local area networks (LANs) and wide area networks (WANs). Different technologies have been developed to bring convenience to sharing resources and files on a network; a distributed file system is one of the processes used regularly.

One process involved in implementing the DFS is giving access control and storage management controls to the client system in a centralized way, managed by the servers. Transparency is one of the core processes in DFS, so files are accessed, stored, and managed on the local client machines while the process itself is actually held on the servers. This transparency brings convenience to the end user on a client machine because the network file system efficiently manages all the processes. Generally, a DFS is used in a LAN, but it can be used in a WAN or over the Internet.

A DFS allows efficient and well-managed data and storage sharing options on a network compared to other options. Another option for users in network-based computing is a shared disk file system. A shared disk file system puts the access control on the client's systems so the data is inaccessible when the client system goes offline. DFS is fault-tolerant and the data is accessible even if some of the network nodes are offline.

DIWAKAR EDUCATION HUB

A DFS makes it possible to restrict access to the file system depending on access lists or capabilities on both the servers and the clients, depending on how the protocol is designed.

HDFS

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly faulttolerant and designed using low-cost hardware.

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

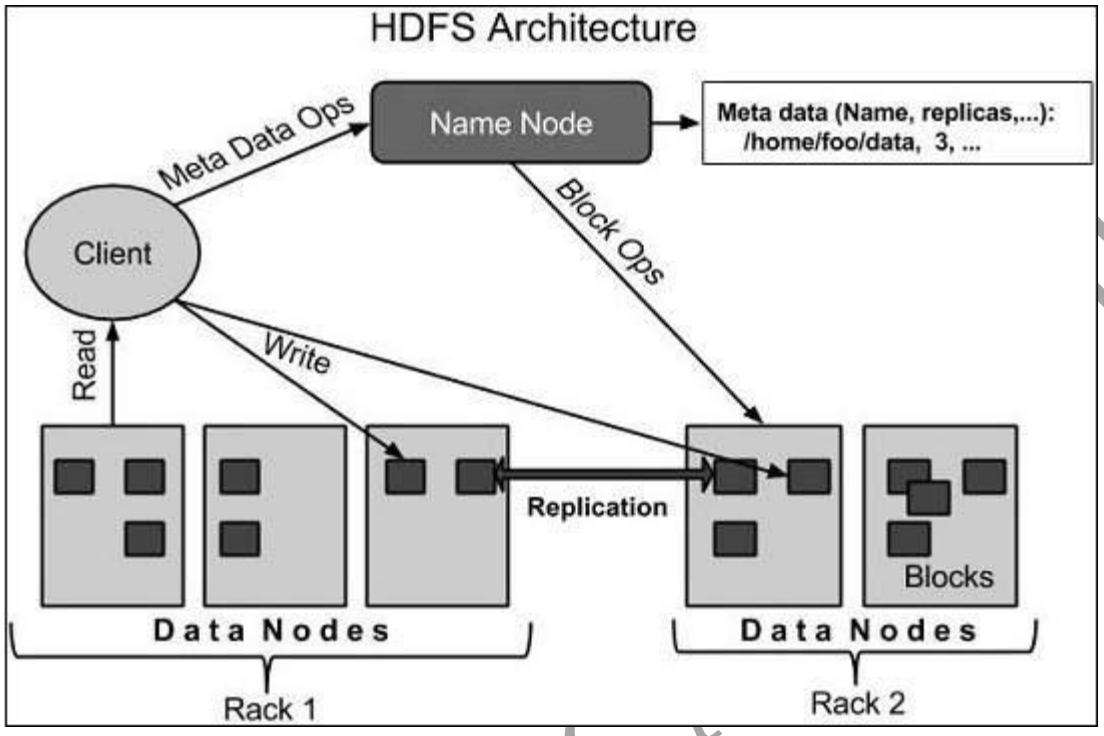
Features of HDFS

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

HDFS Architecture

Given below is the architecture of a Hadoop File System.

DIWAKAR EDUCATION HUB



HDFS follows the master-slave architecture and it has the following elements.

Namenode

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks −

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

Datanode

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a

DIWAKAR EDUCATION HUB

cluster, there will be a datanode. These nodes manage the data storage of their system.

- Datanodes perform read-write operations on the file systems, as per client request.
- They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

Block

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goals of HDFS

Fault detection and recovery − Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

Huge datasets − HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

Hardware at data − A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

NoSQL

NoSQL databases (aka "not only SQL") are non tabular, and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph.

DIWAKAR EDUCATION HUB

They provide flexible schemas and scale easily with large amounts of data and high user loads.

What is NoSQL?

When people use the term "NoSQL database", they typically use it to refer to any non-relational database. Some say the term "NoSQL" stands for "non SQL" while others say it stands for "not only SQL." Either way, most agree that NoSQL databases are databases that store data in a format other than relational tables.

A common misconception is that NoSQL databases or non-relational databases don't store relationship data well. NoSQL databases can store relationship data—they just store it differently than relational databases do. In fact, when compared with SQL databases, many find modeling relationship data in NoSQL databases to be *easier* than in SQL databases, because related data doesn't have to be split between tables.

NoSQL data models allow related data to be nested within a single data structure.

NoSQL databases emerged in the late 2000s as the cost of storage dramatically decreased. Gone were the days of needing to create a complex, difficult-to-manage data model simply for the purposes of reducing data duplication. Developers (rather than storage) were becoming the primary cost of software development, so NoSQL databases optimized for developer productivity.

The benefits of NoSQL database

Data Models

NoSQL databases often leverage data models more tailored to specific use cases, making them better at supporting those workloads than relational databases. For example, key-value databases support simple queries very efficiently while graph databases are the best for queries that involve identifying complex relationships between separate pieces of data.

Performance

DIWAKAR EDUCATION HUB

NoSQL databases can often perform better than SQL/relational databases for your use case. For example, if you're using a document database and are storing all the information about an object in the same document (so that it matches the objects in your code), the database only needs to go to one place for those queries. In a SQL database, the same query would likely involve joining multiple tables and records, which can dramatically impact performance while also slowing down how quickly developers write code.

Scalability

SQL/relational databases were originally designed to scale up and although there are ways to get them to scale out, those solutions are often bolt-ons, complicated, expensive to manage, and hard to evolve. Some core SQL functionality also only really works well when everything is on one server. In contrast, NoSQL databases are designed from the ground up to scale-out horizontally, making it much easier to maintain performance as your workload grows beyond the limits of a single server.

Data Distribution

Because NoSQL databases are designed from the ground up as distributed systems, they can more easily support a variety of business requirements. For example, suppose the business needs a globally distributed application that provides excellent performance to users all around the world. NoSQL databases can allow you to deploy a single distributed cluster to support that application and ensure low latency access to data from anywhere. This approach also makes it much easier to comply with data sovereignty mandates required by modern privacy regulations.

Reliability

NoSQL databases ensure high availability and uptime with native replication and built-in failover for self-healing, resilient database clusters. Similar failover systems can be set up for SQL databases but since the functionality is not native

DIWAKAR EDUCATION HUB

to the underlying database, this often means more resources to deploy and maintain a separate clustering layer that then takes longer to identify and recover from underlying systems failures.

Flexibility

NoSQL databases are better at allowing users to test new ideas and update data structures. For example, MongoDB, the leading document database, stores data in flexible, JSON-like documents, meaning fields can vary from document to document and the data structures can be easily changed over time, as application requirements evolve. This is a better fit for modern microservices architectures where developers are continuously integrating and deploying new application functionality.

Queries Optimization

Queries can be executed in many different ways. All paths lead to the same query result. The Query optimizer evaluates the possibilities and selects the efficient plan. Efficiency is measured in latency and throughput, depending on the workload. The cost of Memory, CPU, disk usage is added to the cost of a plan in a cost-based optimizer.

Now, most NoSQL databases have SQL-like query language support. So, a good optimizer is mandatory. When you don't have a good optimizer, developers have to live with feature restrictions and DBAs have to live with performance issues.

Database Optimizer

A query optimizer chooses an optimal index and access paths to execute the query. At a very high level, SQL optimizers decide the following before creating the execution tree:

1. Query rewrite based on heuristics, cost or both.
2. Index selection.

DIWAKAR EDUCATION HUB

- Selecting the optimal index(es) for each of the table (keyspaces in Couchbase N1QL, collection in case of MongoDB)
- Depending on the index selected, choose the predicates to push down, see the query is covered or not, decide on sort and pagination strategy.

3. Join reordering
4. Join type

Queries Optimization

Query optimization is the science and the art of applying equivalence rules to rewrite the tree of operators evoked in a query and produce an optimal plan. A plan is optimal if it returns the answer in the least time or using the least space. There are well known syntactic, logical, and semantic equivalence rules used during optimization. These rules can be used to select an optimal plan among semantically equivalent plans by associating a cost with each plan and selecting the lowest overall cost. The cost associated with each plan is generated using accurate metrics such as the cardinality or the number of result tuples in the output of each operator, the cost of accessing a source and obtaining results from that source, and so on. One must also have a cost formula that can calculate the processing cost for each implementation of each operator. The *overall cost* is typically defined as the total time needed to evaluate the query and obtain all of the answers.

The characterization of an optimal, low-cost plan is a difficult task. The complexity of producing an optimal, low-cost plan for a relational query is NP-complete. However, many efforts have produced reasonable heuristics to solve this problem. Both dynamic programming and randomized optimization based on simulated annealing provide good solutions.

A BIS could be improved significantly by exploiting the traditional database technology for optimization extended to capture the complex metrics presented

DIWAKAR EDUCATION HUB

in Section 4.4.1. Many of the systems presented in this book address optimization at different levels. K2 uses rewriting rules and a cost model. P/FDM combines traditional optimization strategies, such as query rewriting and selection of the best execution plan, with a query-shipping approach. DiscoveryLink performs two types of optimization: query rewriting followed by a cost-based optimization plan. KIND is addressing the use of domain knowledge into executable meta-data. The knowledge of biological resources can be used to identify the best plan with query (Q) defined in Section 4.4.2 as illustrated in the following.

The two possible plans illustrated in Figures 4.1 and 4.2 do not have the same cost. Evaluation costs depend on factors including the number of accesses to each data source, the size (cardinality) of each relation or data source involved in the query, the number of results returned or the selectivity of the query, the number of queries that are submitted to the sources, and the order of accessing sources.

Each access to a data source retrieves many documents that need to be parsed. Each object returned may generate further accesses to (other) sources. Web accesses are costly and should be as limited as possible. A plan that limits the number of accesses is likely to have a lower cost. Early selection is likely to limit the number of accesses. For example, the call to PubMed in the plan illustrated in Figure 4.1 retrieves 81,840 citations, whereas the call to GenBank in the plan in Figure 4.2 retrieves 1616 sequences. (Note that the statistics and results cited in this paper were gathered between April 2001 and April 2002 and may no longer be up to date.) If each of the retrieved documents (from PubMed or GenBank) generated an additional access to the second source, clearly the second plan has the potential to be much less expensive when compared to the first plan.

The size of the data sources involved in the query may also affect the cost of the evaluation plan. As of May 4, 2001, Swiss-Prot contained 95,674 entries whereas PubMed contained more than 11 million citations; these are the values of cardinality for the corresponding relations. A query submitted to PubMed (as used in the first plan) retrieves 727,545 references that mention *brain*, whereas it retrieves 206,317 references that mention *brain* and were published since 1995.

DIWAKAR EDUCATION HUB

This is the selectivity of the query. In contrast, the query submitted to Swiss-Prot in the second plan returns 126 proteins annotated with *calcium channel.*

In addition to the previously mentioned characteristics of the resources, the order of accessing sources and the use of different capabilities of sources also affects the total cost of the plan. The first plan accesses PubMed and extracts values for identifiers of records in Swiss-Prot from the results. It then passes these values to the query on Swiss-Prot via the join operator. To pass each value, the plan may have to send multiple calls to the Swiss-Prot source, one for each value, and this can be expensive. However, by passing these values of identifiers to Swiss-Prot, the Swiss-Prot source has the potential to constrain the query, and this could reduce the number of results returned from Swiss-Prot. On the other hand, the second plan submits queries in parallel to both PubMed and Swiss-Prot. It does not pass values of identifiers of Swiss-Prot records to Swiss-Prot; consequently, more results may be returned from Swiss-Prot. The results from both PubMed and Swiss-Prot have to be processed (joined) locally, and this could be computationally expensive. Recall that for this plan, 206,317 PubMed references and 126 proteins from Swiss-Prot are processed locally. However, the advantage is that a single query has been submitted to Swiss-Prot in the second plan. Also, both sources are accessed in parallel.

Although it has not been described previously, there is a third plan that should be considered for this query. This plan would first retrieve those proteins annotated with *calcium channel* from Swiss-Prot and extract MEDLINE identifiers from these records. It would then pass these identifiers to PubMed and restrict the results to those matching the keyword *brain.* In this particular case, this third plan has the potential to be the least costly. It submits one sub-query to Swiss-Prot, and it will not download 206,317 PubMed references. Finally, it will not join 206,317 PubMed references and 126 proteins from Swiss-Prot locally.

Optimization has an immediate impact in the overall performance of the system. The consequences of the inefficiency of a system to execute users' queries may

DIWAKAR EDUCATION HUB

affect the satisfaction of users as well as the capabilities of the system to return any output to the user.

NoSQL Database

Databases can be divided in 3 types:

1. RDBMS (Relational Database Management System)
2. OLAP (Online Analytical Processing)
3. NoSQL (recently developed database)

NoSQL Database

NoSQL Database is used to refer a non-SQL or non relational database.

It provides a mechanism for storage and retrieval of data other than tabular relations model used in relational databases. NoSQL database doesn't use tables for storing data. It is generally used to store big data and real-time web applications.

History behind the creation of NoSQL Databases

In the early 1970, Flat File Systems are used. Data were stored in flat files and the biggest problems with flat files are each company implement their own flat files and there are no standards. It is very difficult to store data in the files, retrieve data from files because there is no standard way to store data.

Then the relational database was created by E.F. Codd and these databases answered the question of having no standard way to store data. But later relational database also get a problem that it could not handle big data, due to this problem there was a need of database which can handle every types of problems then NoSQL database was developed.

Advantages of NoSQL

- It supports query language.

DIWAKAR EDUCATION HUB

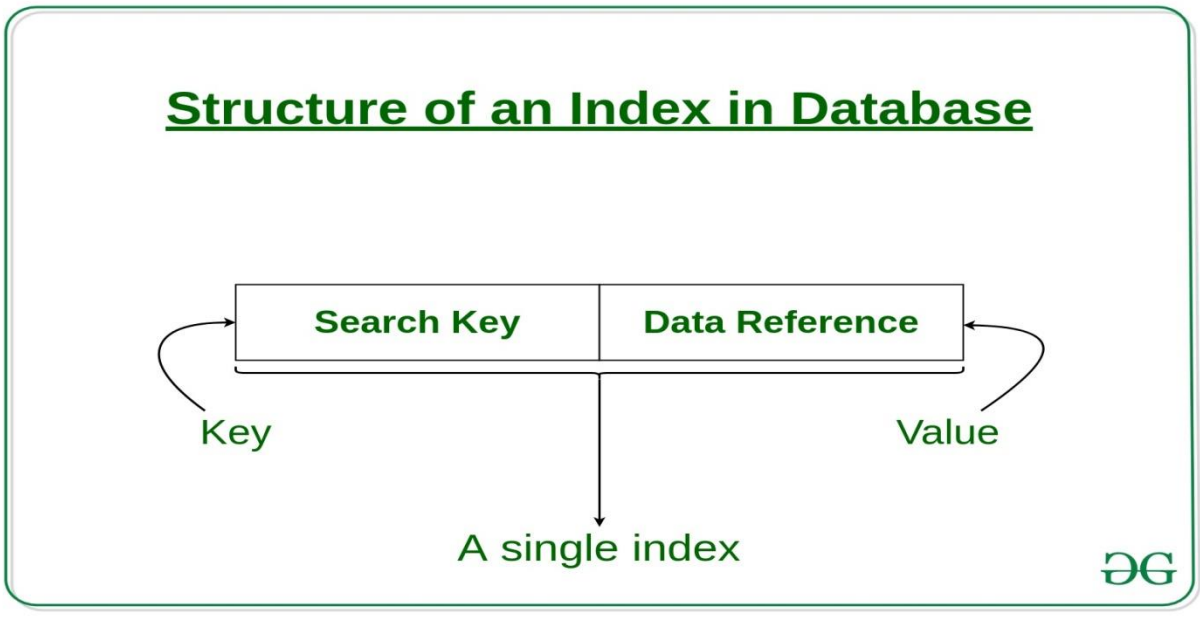
- It provides fast performance.
- It provides horizontal scalability.

Indexing data sets

Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. It is a data structure technique which is used to quickly locate and access the data in a database.

Indexes are created using a few database columns.

- The first column is the **Search key** that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly.
  Note: The data may or may not be stored in sorted order.
- The second column is the **Data Reference** or **Pointer** which contains a set of pointers holding the address of the disk block where that particular key value can be found.



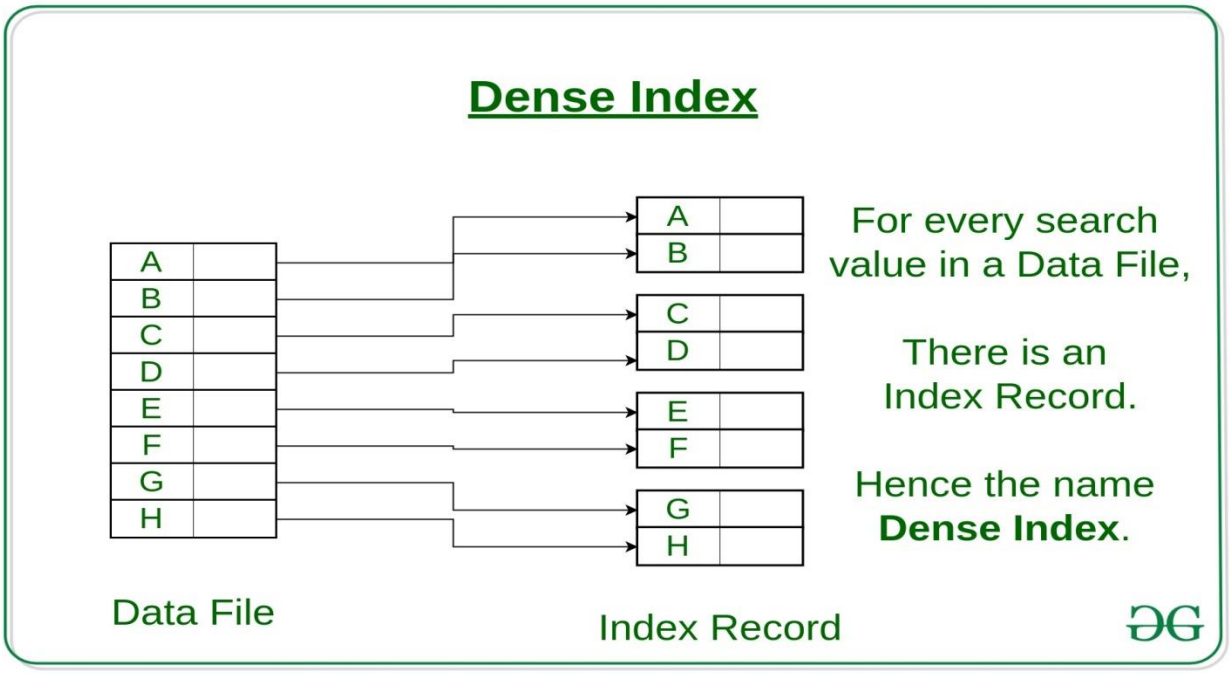
The indexing has various attributes:

- **Access Types**: This refers to the type of access such as value based search, range access, etc.
- **Access Time**: It refers to the time needed to find particular data element or set of elements.
- **Insertion Time**: It refers to the time taken to find the appropriate space and insert a new data.
- **Deletion Time**: Time taken to find an item and delete it as well as update the index structure.
- **Space Overhead**: It refers to the additional space required by the index.

In general, there are two types of file organization mechanism which are followed by the indexing methods to store the data:

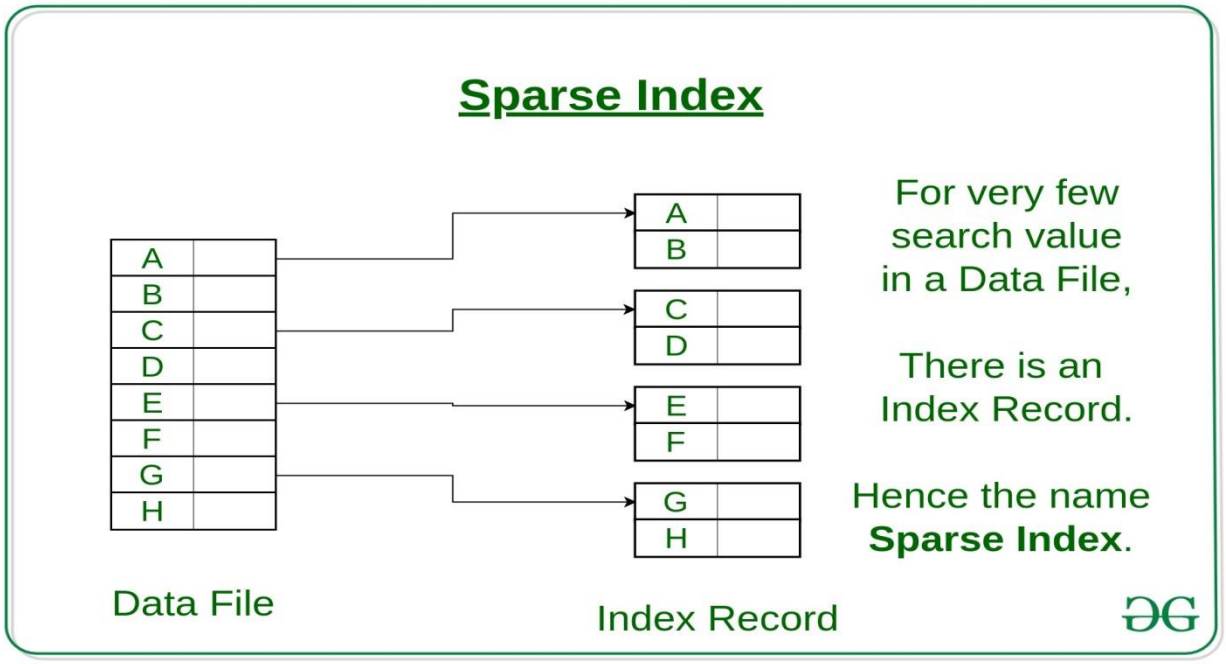
1. **Sequential File Organization or Ordered Index File:** In this, the indices are based on a sorted ordering of the values. These are generally fast and a more traditional type of storing mechanism. These Ordered or Sequential file organization might store the data in a dense or sparse format:
    - **Dense Index:**
        - For every search key value in the data file, there is an index record.
        - This record contains the search key and also a reference to the first data record with that search key value.

DIWAKAR EDUCATION HUB



- **Sparse Index:**
  - The index record appears only for a few items in the data file. Each item points to a block as shown.
  - To locate a record, we find the index record with the largest search key value less than or equal to the search key value we are looking for.
  - We start at that record pointed to by the index record, and proceed along with the pointers in the file (that is, sequentially) until we find the desired record.

DIWAKAR EDUCA



2. **Hash File organization:** Indices are based on the values being distributed uniformly across a range of buckets. The buckets to which a value is assigned is determined by a function called a hash function.

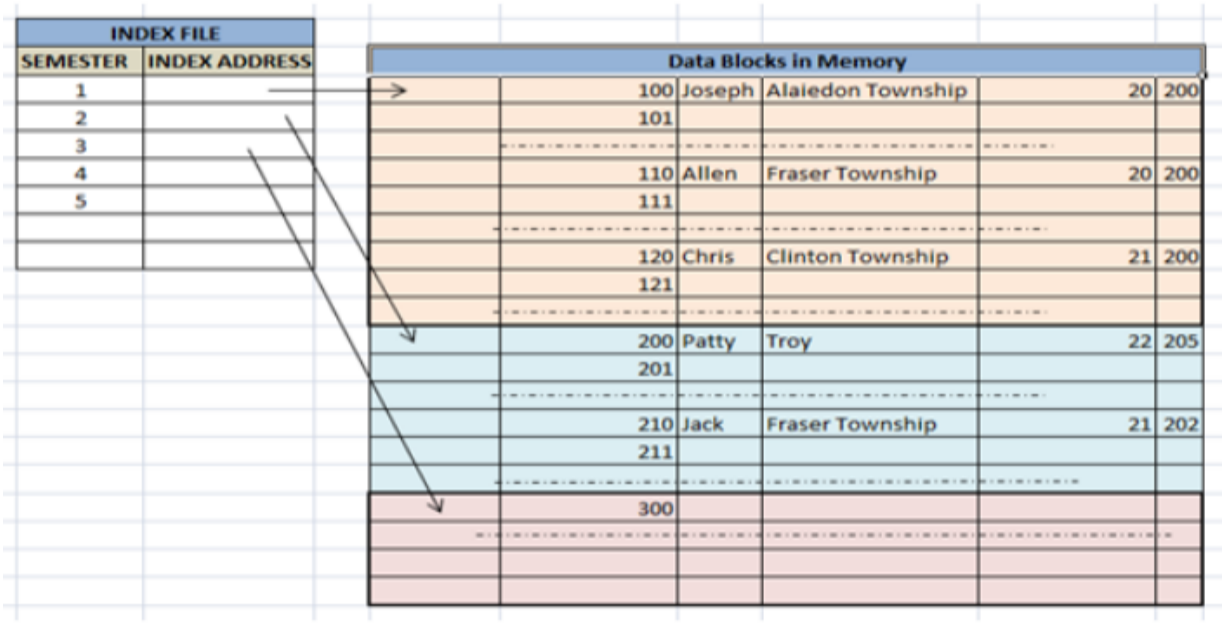
There are primarily three methods of indexing:

- Clustered Indexing
- Non-Clustered or Secondary Indexing
- Multilevel Indexing

1. **Clustered Indexing**
   When more than two records are stored in the same file these types of storing known as cluster indexing. By using the cluster indexing we can reduce the cost of searching reason being multiple records related to the same thing are stored at one place and it also gives the frequent joing of more than two tables(records).
   Clustering index is defined on an ordered data file. The data file is ordered on a non-key field. In some cases, the index is created on non-primary key

DIWAKAR EDUCA

columns which may not be unique for each record. In such cases, in order to identify the records faster, we will group two or more columns together to get the unique values and create index out of them. This method is known as the clustering index. Basically, records with similar characteristics are grouped together and indexes are created for these groups.
For example, students studying in each semester are grouped together. i.e. 1st Semester students, 2nd semester students, 3rd semester students etc are grouped.



Clustered index sorted according to first name (Search key)

Primary Indexing:

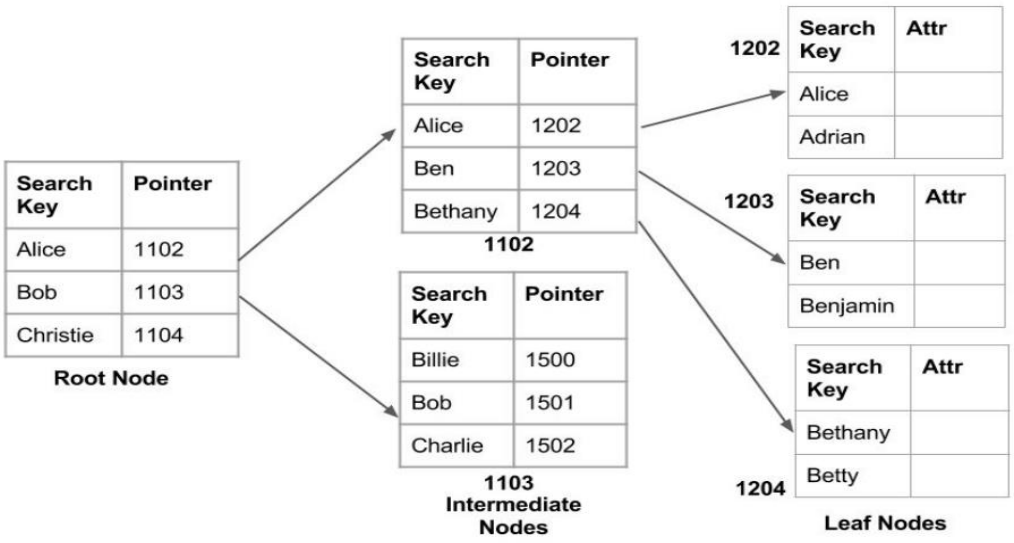
This is a type of Clustered Indexing wherein the data is sorted according to the search key and the primary key of the database table is used to create the index. It is a default format of indexing where it induces sequential file organization. As primary keys are unique and are stored in a sorted manner, the performance of the searching operation is quite efficient.

2. **Non-clustered or Secondary Indexing**
   A non clustered index just tells us where the data lies, i.e. it gives us a list of virtual pointers or references to the location where the data is actually

DIWAKAR EDUCATION HUB

stored. Data is not physically stored in the order of the index. Instead, data is present in leaf nodes. For eg. the contents page of a book. Each entry gives us the page number or location of the information stored. The actual data here(information on each page of the book) is not organized but we have an ordered reference(contents page) to where the data points actually lie. We can have only dense ordering in the non-clustered index as sparse ordering is not possible because data is not physically organized accordingly.

It requires more time as compared to the clustered index because some amount of extra work is done in order to extract the data by further following the pointer. In the case of a clustered index, data is directly present in front of the index.

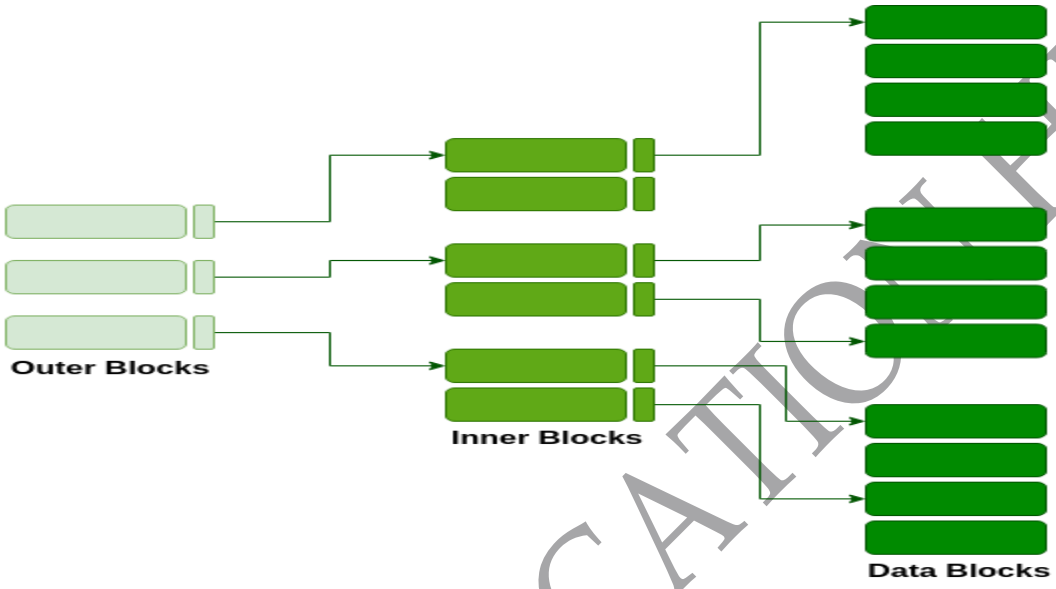


Non clustered index

3. **Multilevel Indexing**

With the growth of the size of the database, indices also grow. As the index is stored in the main memory, a single-level index might become too large a size to store with multiple disk accesses. The multilevel indexing segregates the main block into various smaller blocks so that the same can stored in a

single block. The outer blocks are divided into inner blocks which in turn are pointed to the data blocks. This can be easily stored in the main memory with fewer overheads.



NOSQL in Cloud

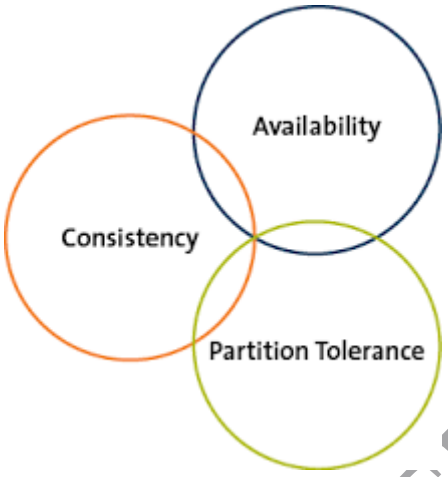
With the current move to cloud computing, the need to scale applications presents itself as a challenge for storing data. If you are using a traditional relational database you may find yourself working on a complex policy for distributing your database load across multiple database instances. This solution will often present a lot of problems and probably won't be great at elastically scaling.

As an alternative you could consider a cloud-based NoSQL database. Over the past few weeks I have been analysing a few such offerings, each of which promises to scale as your application grows, without requiring you to think about how you might distribute the data and load.

Specifically I have been looking at Amazon's DynamoDB, Google's Cloud Datastore and Cloud BigTable. I chose to take a look into these 3 databases because we have existing applications running in Google and Amazon's clouds and I can see the advantage these databases can offer. In this post I'll report on what I've learnt.

DIWAKAR EDUCATION HUB

Consistency, Availability & Partition Tolerance



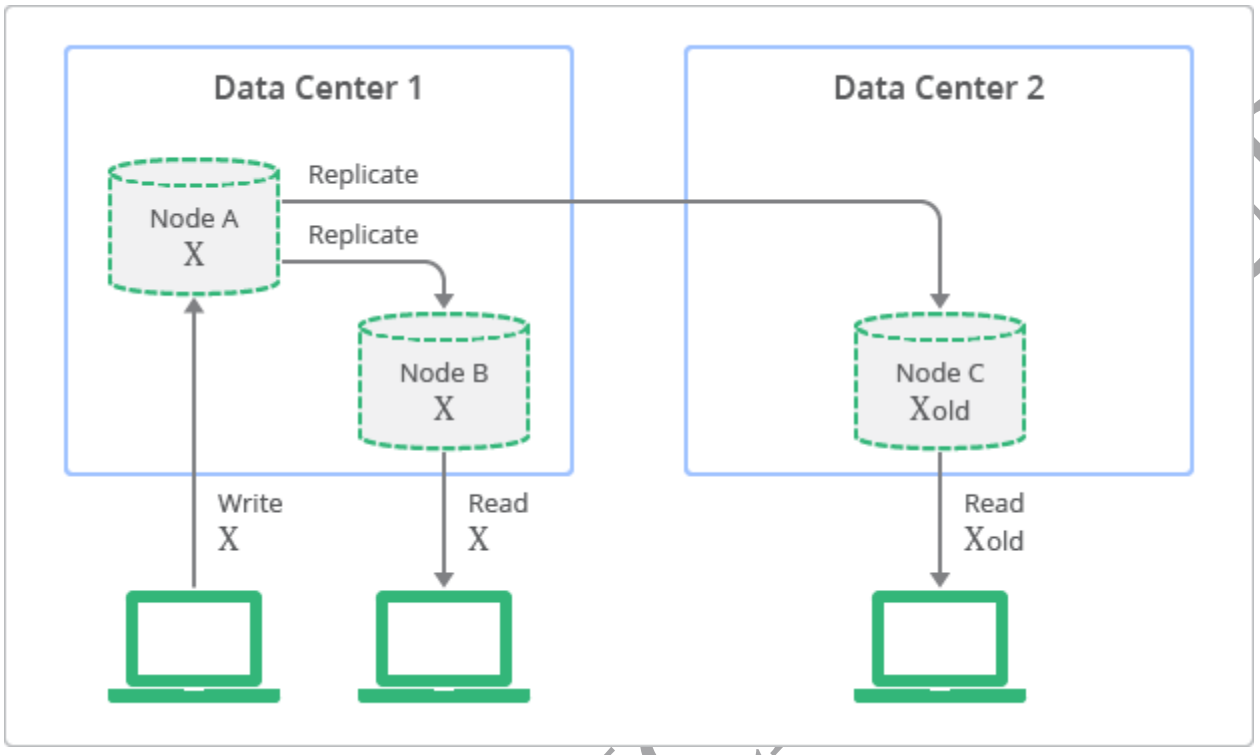
Firstly — and most importantly — it's necessary to understand that distributed NoSQL databases achieve high scalability in comparison to a traditional RDBMS by making some important tradeoffs.

A good starting-place for thinking about this is the *CAP Theorem*, which states that a distributed database can — at most — provide two of the following: Consistency, Availability and Partition Tolerance. We define each of these as follows:

- **Consistency:** All nodes contain the same data
- **Availability:** Every request should receive a response
- **Partition Tolerance:** Losing a node should not affect the system

Eventually Consistent Operations

DIWAKAR EDUCATION HUB



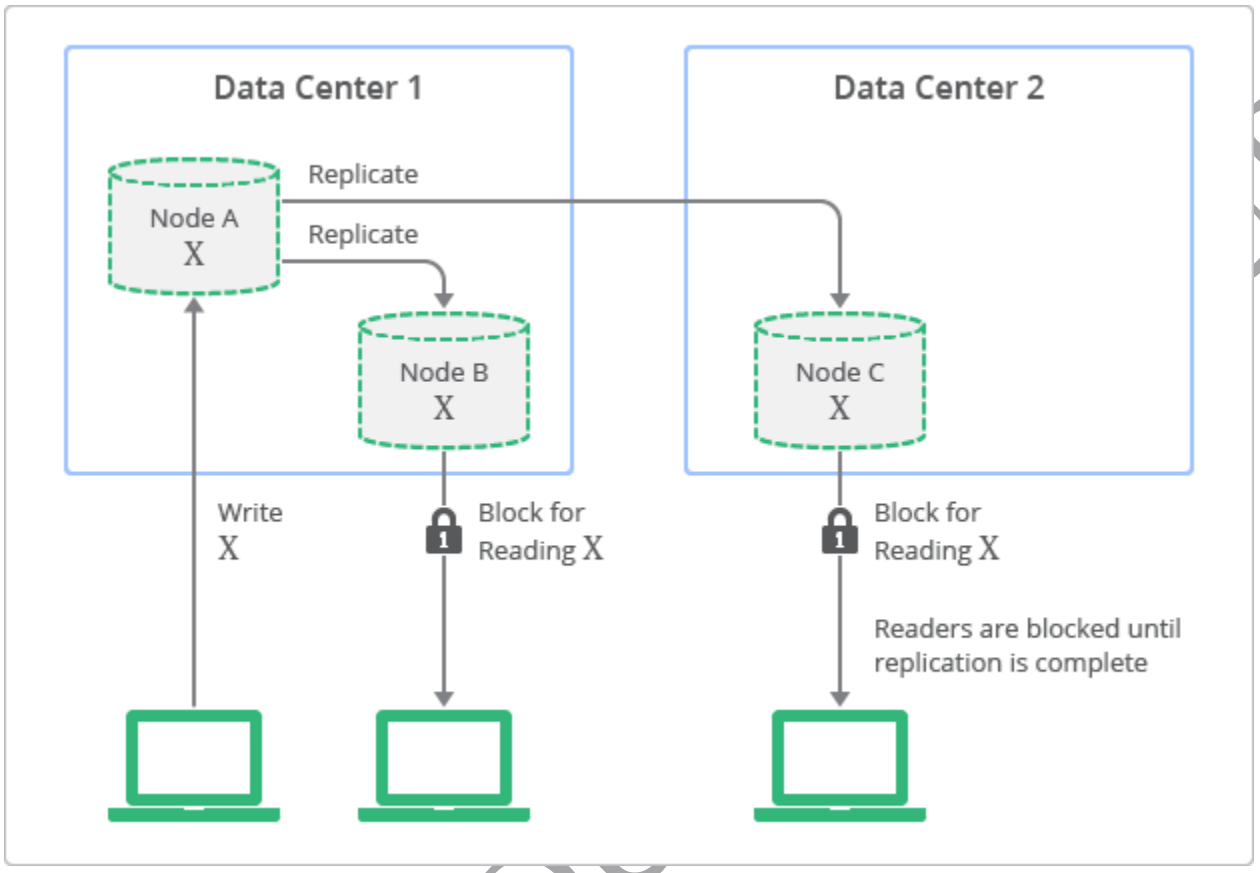
All three NoSQL databases I looked at provide Availability and Partition Tolerance for eventually-consistent operations. In most cases these two properties will suffice.

For example, if a user posts to a social media website and it takes a second or two for everyone's request to pick up the change, then it's not usually an issue.

This happens due to write operations writing to multiple nodes before the data is eventually replicated across all of the nodes, which usually occurs within one second. Read operations are then read from only one node.

Strongly Consistent Operations

DIWAKAR EDUCATION HUB



All three databases also provide strongly consistent operations which guarantee that the latest version of the data will always be returned.

DynamoDB achieves this by ensuring that writes are written out to the majority of nodes before a success result is returned. Reads are also done in a similar way — results will not return until the record is read from more then half of the nodes. This is to ensure that the result will be the latest copy of the record.

All this occurs at the expense of availability, where a node being inaccessible can prevent the verification of the data's consistency if it occurs a short time after the write operation. Google achieves this behaviour in a slightly different way by using a locking mechanism where a read can't be completed on a node until it has the latest copy of the data. This model is required when you need to guarantee the consistency of your data. For example, you would not want a financial transaction being calculated on an old version of the data.

DIWAKAR EDUCATION HUB

OK, now that we've got the hard stuff out of the way, let's move onto some of the more practical questions that might come up when using a cloud-based database.

Local Development

Having a database in the cloud is cool, but how does it work if you've got a team of developers, each of whom needs to run their own copy of the database locally? Fortunately, DynamoDB, BigTable and Cloud Datastore all have the option of downloading and running a local development server. All three local development environments are really easy to download and get started with. They are designed to provide you with an interface that matches the production environment.

Java Object Mapping

If you are going to be using Java to develop your application, you might be used to using frameworks like Hibernate or JPA to automatically map RDBMS rows to objects. How does this work with NoSQL databases?

DynamoDB provides an intuitive way of mapping Java classes to objects in DynamoDB Tables. You simply annotate the Java object as a DynamoDB Table and then annotate your instance variable getters with the appropriate annotations.

```
@DynamoDBTable(tableName="users")
public class User {

@DynamoDBHashKey(attributeName="username")
public String getUsername(){
   return username;
}
public void setUsername(String username){
   this.username = username;
}

@DynamoDBAttribute(attributeName = "email")
```

DIWAKAR EDUCATION HUB

```
public String getEmail(){
   return email;
}
public void setEmail(String email){
   this.email = email;
}
```

Querying

An important thing to understand about all of these NoSQL databases is that they don't provide a full-blown query language.

Instead, you need to use their APIs and SDKs to access the database. By using simple query and scan operations you can retrieve zero or more records from a given table. Since each of the three databases I looked at provide a slightly different way of indexing the tables, the range of features in this space varies.

DynamoDB for example provides multiple secondary indexes, meaning there is the ability to efficiently scan any indexed column. This is not a feature in either of Google's NoSQL offerings.

Furthermore, unlike SQL databases, none of these NoSQL databases give you a means of doing table joins, or even having foreign keys. Instead, this is something that your application has to manage itself.

That's said, one of the main advantages in my opinion of NoSQL is that there is no fixed schema. As your needs change you can dynamically add new attributes to records in your table.

For example, using Java and DynamoDB, you can do the following, which will return a list of users that have the same username as a given user:

```
User user = new User(username);
DynamoDBQueryExpression<User> queryExpression =
new DynamoDBQueryExpression<User>().withHashKeyValues(user);
```

DIWAKAR EDUCATION HUB

```
List<User> itemList =
Properties.getMapper().query(User.class, queryExpression);
```

Distributed Database Design

The main benefit of NoSQL databases is their ability to scale, and to do so in an almost seamless way. But, just like a SQL database, a poorly-designed NoSQL database can give you slow query response times. This is why you need to consider your database design carefully.

In order to spread the load across multiple nodes, distributed databases need to spread the stored data across multiple nodes. This is done in order for the load to be balanced. The flip-side of this is that if frequently-accessed data is on a small subset of nodes, you will not be making full use of the available capacity.

Consequently, you need to be careful of which columns you select as indexes. Ideally you want to spread your load across the whole table as opposed to accessing only a portion of your data.

A good design can be achieved by picking a hash key that is likely to be randomly accessed. For example if you have a users table and choose the username as the hash key it will be likely that load will distributed across all of the nodes. This is due to the likeliness that users will be randomly accessed.

In contrast to this, it would, for example, be a poor design to use the date as the hash key for a table that contains forum posts. This is due to the likeliness that most of the requests will be for the records on the current day so the node or nodes containing these records will likely be a small subset of all the nodes. This scenario can cause your requests to be throttled or hang.

Pricing

DIWAKAR EDUCATION HUB



Since Google does not have a data centre in Australia, I will only be looking at pricing in the US.

DynamoDB is priced on storage and provisioned read/write capacity. In the Oregon region storage is charged at $0.25 per GB/Month and at $0.0065 per hour for every 10 units of Write Capacity and the same price for every 50 units of read capacity.

Google Cloud Datastore has a similar pricing model. With storage priced at $0.18 per GB of data per month and $0.06 per 100,000 read operations. Write operations are charged at the same rate. Datastore also have a Free quota of 50,000 read and 50,000 write operations per day. Since Datastore is a Beta product it currently has a limit of 100 million operations per day, however you can request the limit to be increased.

The pricing model for Google Bigtable is significantly different. With Bigtable you are charged at a rate of $0.65 per instance/hour. With a minimum of 3 instances required, some basic arithmetic gives us a starting price for Bigtable of $142.35 per month. You are then charged at $0.17 per GB/Month for SSD-backed storage. A cheaper HDD-backed option priced at $0.026 per GB/Month is yet to be released.

Finally you are charged for external network usage. This ranges between 8 and 23 cents per GB of traffic depending on the location and amount of data transferred. Traffic to other Google Cloud Platform services in the same region/zone is free.

DIWAKAR EDUCATION HUB

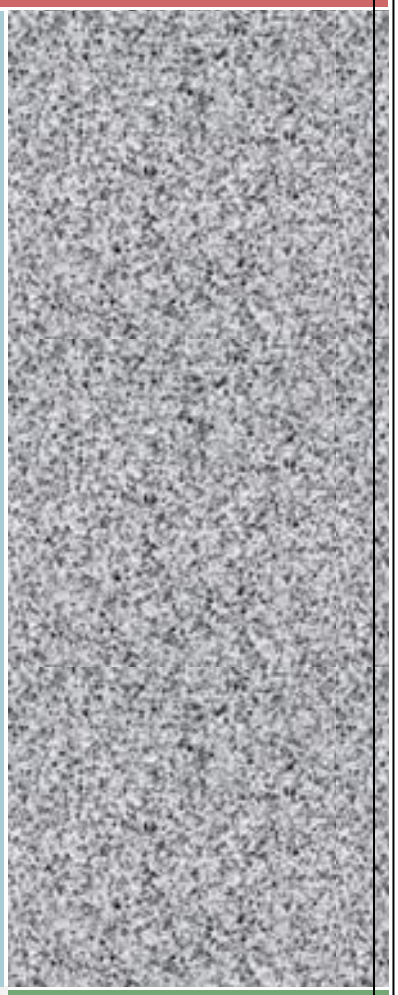
DIWAKAR EDUCATION HUB

DIWAKAR EDUCATION HUB

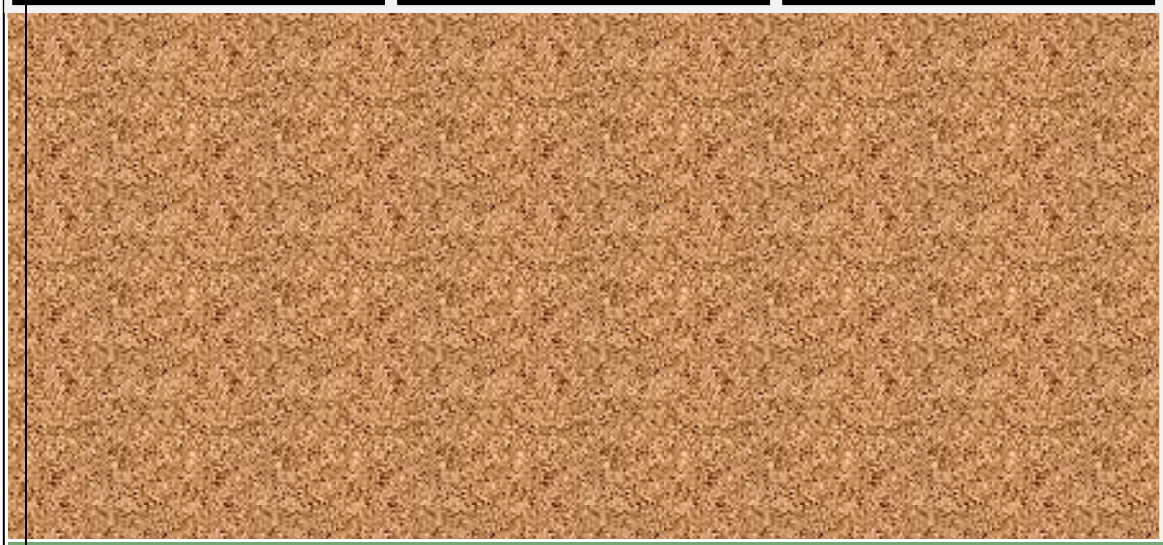
Database Management Systems
Unit – 4 MCQs

As per updated syllabus

DIWAKAR EDUCATION HUB



2020



THE LEARN WITH EXPERTIES

1. A relational database consists of a collection of
a) Tables
b) Fields
c) Records
d) Keys

Answer: a
Explanation: Fields are the column of the relation or tables. Records are each row in a relation. Keys are the constraints in a relation.

2. A _____ in a table represents a relationship among a set of values.
a) Column
b) Key
c) Row
d) Entry

Answer: c
Explanation: Column has only one set of values. Keys are constraints and row is one whole set of attributes. Entry is just a piece of data.

3. The term _____ is used to refer to a row.
a) Attribute
b) Tuple
c) Field
d) Instance

Answer: b
Explanation: Tuple is one entry of the relation with several attributes which are fields.

4. The term attribute refers to a _____ of a table.
a) Record
b) Column
c) Tuple
d) Key

Answer: b
Explanation: Attribute is a specific domain in the relation which has entries of all tuples.

5. For each attribute of a relation, there is a set of permitted values, called the _____ of that attribute.
a) Domain
b) Relation
c) Set
d) Schema

Answer: a
Explanation: The values of the attribute should be present in the domain. Domain is a set of values permitted.

6. Database _____ which is the logical design of the database, and the database _____ which is a snapshot of the data in the database at a given instant in time.
a) Instance, Schema
b) Relation, Schema
c) Relation, Domain
d) Schema, Instance

Answer: d
Explanation: Instance is an instance of time and schema is a representation.

7. Course(course_id,sec_id,semester)
Here the course_id,sec_id and semester are _____ and course is a _____
a) Relations, Attribute
b) Attributes, Relation
c) Tuple, Relation
d) Tuple, Attributes

Answer: b
Explanation: The relation course has a set of attributes course_id,sec_id,semester .

8. Department (dept name, building, budget) and Employee (employee_id, name, dept name, salary)
Here the dept_name attribute appears in both the relations. Here using common attributes in relation schema is one way of relating _____ relations.
a) Attributes of common
b) Tuple of common
c) Tuple of distinct
d) Attributes of distinct

Answer: c
Explanation: Here the relations are connected by the common attributes.

9. A domain is atomic if elements of the domain are considered to be _____ units.
a) Different
b) Indivisbile
c) Constant
d) Divisible

Answer: b
Explanation: None.

10. The tuples of the relations can be of _____ order.
a) Any
b) Same
c) Sorted
d) Constant

Answer: a
Explanation: The values only count. The order of the tuples does not matter.

11. Which one of the following is a set of one or more attributes taken collectively to uniquely identify a record?
a) Candidate key
b) Sub key
c) Super key
d) Foreign key

Answer: c
Explanation: Super key is the superset of all the keys in a relation.

12. Consider attributes ID, CITY and NAME. Which one of this can be considered as a super key?
a) NAME
b) ID
c) CITY
d) CITY, ID

Answer: b
Explanation: Here the id is the only attribute which can be taken as a key. Other attributes are not uniquely identified.

13. The subset of a super key is a candidate key under what condition?
a) No proper subset is a super key
b) All subsets are super keys
c) Subset is a super key
d) Each subset is a super key

Answer: a
Explanation: The subset of a set cannot be the same set. Candidate key is a set from a super key which cannot be the whole of the super set.

14. A _____ is a property of the entire relation, rather than of the individual tuples in which each tuple is unique.
a) Rows
b) Key
c) Attribute
d) Fields

Answer: b
Explanation: Key is the constraint which specifies uniqueness.

15. Which one of the following attribute can be taken as a primary key?
a) Name
b) Street
c) Id
d) Department

Answer: c
Explanation: The attributes name, street and department can repeat for some tuples. But the id attribute has to be unique. So it forms a primary key.

16. Which one of the following cannot be taken as a primary key?
a) Id
b) Register number
c) Dept_id
d) Street

Answer: d
Explanation: Street is the only attribute which can occur more than once.

17. An attribute in a relation is a foreign key if the _____ key from one relation is used as an attribute in that relation.
a) Candidate
b) Primary
c) Super
d) Sub

Answer: b
Explanation: The primary key has to

be referred in the other relation to form a foreign key in that relation.

18. The relation with the attribute which is the primary key is referenced in another relation. The relation which has the attribute as a primary key is called
a) Referential relation
b) Referencing relation
c) Referenced relation
d) Referred relation

Answer: b
Explanation: None.

19. The _____ is the one in which the primary key of one relation is used as a normal attribute in another relation.
a) Referential relation
b) Referencing relation
c) Referenced relation
d) Referred relation

Answer: c
Explanation: None.

20. A _____ integrity constraint requires that the values appearing in specified attributes of any tuple in the referencing relation also appear in specified attributes of at least one tuple in the referenced relation.
a) Referential
b) Referencing
c) Specific
d) Primary

Answer: a
Explanation: A relation, say r1, may include among its attributes the primary key of another relation, say r2. This attribute is called a foreign key from r1, referencing r2. The relation r1 is also called the referencing relation of the foreign key dependency, and r2 is called the referenced relation of the foreign key.

21. Using which language can a user request information from a database?
a) Query
b) Relational
c) Structural
d) Compiler

Answer: a
Explanation: Query language is a method through which the database entries can be accessed.

22. Student(ID, name, dept name, tot_cred)
In this query which attributes form the primary key?
a) Name
b) Dept
c) Tot_cred
d) ID

Answer: d
Explanation: The attributes name, dept and tot_cred can have same values unlike ID.

23. Which one of the following is a procedural language?
a) Domain relational calculus
b) Tuple relational calculus
c) Relational algebra
d) Query language

Answer: c
Explanation: Domain and Tuple relational calculus are non-procedural language. Query language is a method through which database entries can be accessed.

24. The_____ operation allows the combining of two relations by merging pairs of tuples, one from each relation, into a single tuple.
a) Select
b) Join
c) Union
d) Intersection

Answer: b
Explanation: Join finds the common tuple in the relations and combines it.

25. The result which operation contains all pairs of tuples from the two relations, regardless of whether their attribute values match.
a) Join
b) Cartesian product
c) Intersection
d) Set difference

Answer: b
Explanation: Cartesian product is the multiplication of all the values in the attributes.

26. Which one of the following is used to define the structure of the relation, deleting relations and relating schemas?
a) DML(Data Manipulation Langauge)
b) DDL(Data Definition Langauge)
c) Query
d) Relational Schema

Answer: b
Explanation: Data Definition language is the language which performs all the operation in defining structure of relation.

27. Which one of the following provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database?
a) DML(Data Manipulation Langauge)
b) DDL(Data Definition Langauge)
c) Query
d) Relational Schema

Answer: a
Explanation: DML performs the change in the values of the relation.

28.

CREATE TABLE employee (name **VARCHAR**, id **INTEGER**)

What type of statement is this?
a) DML
b) DDL
c) View
d) Integrity constraint

Answer: b
Explanation: Data Definition language is the language which performs all the operation in defining structure of relation.

29.

SELECT * **FROM** employee

What type of statement is this?
a) DML
b) DDL
c) View
d) Integrity constraint

Answer: a
Explanation: Select operation just shows the required fields of the relation. So it forms a DML.

30. The basic data type char(n) is a _____ length character string and varchar(n) is _____ length character.
a) Fixed, equal
b) Equal, variable
c) Fixed, variable
d) Variable, equal

Answer: c
Explanation: Varchar changes its length accordingly whereas char has a specific length which has to be filled by either letters or spaces.

31. An attribute A of datatype varchar(20) has the value “Avi”. The attribute B of datatype char(20) has value ”Reed”. Here attribute A has ____ spaces and attribute B has ____ spaces.
a) 3, 20
b) 20, 4
c) 20, 20
d) 3, 4

Answer: a
Explanation: Varchar changes its length accordingly whereas char has a specific length which has to be filled by either letters or spaces.

32. To remove a relation from an SQL database, we use the _____ command.

DIWAKAR EDUCATION HUB

a) Delete
b) Purge
c) Remove
d) Drop table

Answer: d
Explanation: Drop table deletes the whole structure of the relation .purge removes the table which cannot be obtained again.

33.

DELETE FROM r; //r - relation

This command performs which of the following action?
a) Remove relation
b) Clear relation entries
c) Delete fields
d) Delete rows

Answer: b
Explanation: Delete command removes the entries in the table.

34.

INSERT INTO instructor **VALUES** (10211, 'Smith', 'Biology', 66000);

What type of statement is this?
a) Query
b) DML
c) Relational
d) DDL

Answer: b
Explanation: The values are manipulated. So it is a DML.

35. Updates that violate _____ are disallowed.
a) Integrity constraints
b) Transaction control
c) Authorization
d) DDL constraints

Answer: a
Explanation: Integrity constraint has to be maintained in the entries of the relation.

36.

Name
Annie
Bob
Callie
Derek

Which of these query will display the the table given above ?
a) Select employee from name
b) Select name

DIWAKAR EDUCATION HUB

c) Select name from employee
d) Select employee

Answer: c
Explanation: The field to be displayed is included in select and the table is included in the from clause.

37. Here which of the following displays the unique values of the column?

```
SELECT _________ dept_name
FROM instructor;
```

a) All
b) From
c) Distinct
d) Name

Answer: c
Explanation: Distinct keyword selects only the entries that are unique.

38. The _____ clause allows us to select only those rows in the result relation of the ____ clause that satisfy a specified predicate.
a) Where, from
b) From, select
c) Select, from
d) From, where

Answer: a
Explanation: Where selects the rows on a particular condition. From gives the relation which involves the operation.

39. The query given below will not give an error. Which one of the following has to be replaced to get the desired output?

```
SELECT ID, name, dept name, salary
* 1.1
WHERE instructor;
```

a) Salary*1.1
b) ID
c) Where
d) Instructor

Answer: c
Explanation: Where selects the rows on a particular condition. From gives the relation which involves the operation. Since Instructor is a relation it has to have from clause.

40. The _____ clause is used to list the attributes desired in the result of a query.
a) Where
b) Select
c) From
d) Distinct

Answer: b
Explanation: None

41. This Query can be replaced by which one of the following?

```
SELECT name, course_id
FROM instructor, teaches
WHERE instructor_ID= teaches_ID;
```

a) Select name,course_id from teaches,instructor where instructor_id=course_id;
b) Select name, course_id from instructor natural join teaches;
c) Select name, course_id from instructor;
d) Select course_id from instructor join teaches;

Answer: b
Explanation: Join clause joins two tables by matching the common column.

42.

```
SELECT * FROM employee WHERE salary>10000 AND dept_id=101;
```

Which of the following fields are displayed as output?
a) Salary, dept_id
b) Employee
c) Salary
d) All the field of employee relation

Answer: d
Explanation: Here * is used to select all the fields of the relation.

43.

Employee_id	Name	Salary
1001	Annie	6000
1009	Ross	4500
1018	Zeith	7000

This is Employee table.
Which of the following employee_id will be displayed for the given query?

```
SELECT * FROM employee WHERE employee_id>1009;
```

a) 1009, 1001, 1018
b) 1009, 1018
c) 1001
d) 1018

Answer: d
Explanation: Greater than symbol does not include the given value unlike >=.

44.

```
SELECT name ____ instructor name, course id
FROM instructor, teaches
WHERE instructor.ID= teaches.ID;
```

Which keyword must be used here to rename the field name?
a) From
b) Rename
c) As

d) Join

Answer: c
Explanation: As keyword is used to rename.

45.

```
SELECT * FROM employee WHERE
dept_name="Comp Sci";
```

In the SQL given above there is an error . Identify the error.
a) Dept_name
b) Employee
c) “Comp Sci”
d) From

Answer: c
Explanation: For any string operations single quoted(‘) must be used to enclose.

46.

```
SELECT emp_name
FROM department
WHERE dept_name LIKE ’ _____
Computer Science’;
```

Which one of the following has to be added into the blank to select the dept_name which has Computer Science as its ending string?
a) %
b) _
c) | |
d) $

Answer: a
Explanation: The % character matches any substring.

47. ’_ _ _ ’ matches any string of _____ three characters. ’_ _ _ %’ matches any string of at _____ three characters.
a) Atleast, Exactly
b) Exactly, Atleast
c) Atleast, All
d) All, Exactly

Answer: b
Explanation: None.

48.

```
SELECT name
FROM instructor
WHERE dept name = ’Physics’
ORDER BY name;
```

By default, the order by clause lists items in _____ order.
a) Descending
b) Any
c) Same
d) Ascending

Answer: d
Explanation: Specification of descending order is essential but it not for ascending.

49.

```
SELECT *
FROM instructor
ORDER BY salary ____, name ___;
```

To display the salary from greater to smaller and name in ascending order which of the following options should be used?
a) Ascending, Descending
b) Asc, Desc
c) Desc, Asc
d) Descending, Ascending

Answer: c
Explanation: None.

50. The union operation is represented by
a) ∩
b) U
c) –
d) *

Answer: b
Explanation: Union operator combines the relations.

51. The intersection operator is used to get the _____ tuples.
a) Different
b) Common
c) All
d) Repeating

Answer: b
Explanation: Intersection operator ignores unique tuples and takes only common ones.

52. The union operation automatically _____ unlike the select clause.
a) Adds tuples
b) Eliminates unique tuples
c) Adds common tuples
d) Eliminates duplicate

Answer: d
Explanation: None.

53. If we want to retain all duplicates, we must write _____ in place of union.
a) Union all
b) Union some
c) Intersect all
d) Intersect some

Answer: a
Explanation: Union all will combine all the tuples including duplicates.

54.

```
(SELECT course id
FROM SECTION
WHERE semester = 'Fall' AND YEAR= 2009)
EXCEPT
(SELECT course id
FROM SECTION
WHERE semester = 'Spring' AND YEAR= 2010);
```

This query displays
a) Only tuples from second part

b) Only tuples from the first part which has the tuples from second part
c) Tuples from both the parts
d) Tuples from first part which do not have second part

Answer: d
Explanation: Except keyword is used to ignore the values.

55. For like predicate which of the following is true.

i) % matches zero **OF** more characters.
ii) _ matches exactly one **CHARACTER**.

a) i-only
b) ii-only
c) i & ii
d) None of the mentioned

Answer: a
Explanation:% is used with like and _ is used to fill in the character.

56. The number of attributes in relation is called as its
a) Cardinality
b) Degree
c) Tuples
d) Entity

Answer: b
Explanation: None.

57. _____ clause is an additional filter that is applied to the result.
a) Select
b) Group-by
c) Having
d) Order by

Answer: c
Explanation: Having is used to provide additional aggregate filtration to the query.

58. _____ joins are SQL server default
a) Outer
b) Inner
c) Equi
d) None of the mentioned

Answer: b
Explanation: It is optional to give the inner keyword with the join as it is default.

59. The _____ is essentially used to search for patterns in target string.
a) Like Predicate
b) Null Predicate
c) In Predicate
d) Out Predicate

Answer: a
Explanation: Like predicate matches the string in the given pattern.

60. Aggregate functions are functions that take a _____ as input and return a single value.
a) Collection of values
b) Single value
c) Aggregate value
d) Both Collection of values & Single value

Answer: a
Explanation: None.

61.

```
SELECT _____
FROM instructor
WHERE dept name= 'Comp. Sci.';
```

Which of the following should be used to find the mean of the salary ?
a) Mean(salary)
b) Avg(salary)
c) Sum(salary)
d) Count(salary)

Answer: b
Explanation: Avg() is used to find the mean of the values.

62.

```
SELECT COUNT (____ ID)
FROM teaches
WHERE semester = 'Spring' AND
YEAR = 2010;
```

If we do want to eliminate duplicates, we use the keyword _____in the aggregate expression.
a) Distinct
b) Count
c) Avg
d) Primary key

Answer: a
Explanation: Distinct keyword is used to select only unique items from the relation.

63. All aggregate functions except _____ ignore null values in their input collection.
a) Count(attribute)
b) Count(*)
c) Avg
d) Sum

Answer: b
Explanation: * is used to select all values including null.

64. A Boolean data type that can take values true, false, and_____
a) 1
b) 0
c) Null
d) Unknown

Answer: d
Explanation: Unknown values do not take null value but it is not known.

65. The ____ connective tests for set membership, where the set is a collection of values produced by a

select clause. The ____ connective tests for the absence of set membership.
a) Or, in
b) Not in, in
c) In, not in
d) In, or

Answer: c
Explanation: In checks, if the query has the value but not in checks if it does not have the value.

66. The phrase “greater than at least one” is represented in SQL by _____
a) < all
b) < some
c) > all
d) > some

Answer: d
Explanation: >some takes atlest one value above it .

67. Which of the following is used to find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester .
a)

```
SELECT course id
FROM SECTION AS S
WHERE semester = ’Fall’ AND YEAR= 2009 AND
EXISTS (SELECT *
FROM SECTION AS T
WHERE semester = ’Spring’ AND YEAR= 2010 AND
S.course id= T.course id);
```

b)

```
SELECT name
FROM instructor
WHERE salary > SOME (SELECT salary
FROM instructor
WHERE dept name = ’Biology’);
```

c)

```
SELECT COUNT (DISTINCT ID)
FROM takes
WHERE (course id, sec id, semester, YEAR) IN (SELECT course id, sec id, semester, YEAR
FROM teaches
WHERE teaches.ID= 10101);
```

d)

```
(SELECT course id
FROM SECTION
WHERE semester = ’Spring’ AND YEAR= 2010)
```

Answer: a
Explanation: None.

68. We can test for the nonexistence of tuples in a subquery by using the _____ construct.
a) Not exist

b) Not exists
c) Exists
d) Exist

Answer: b
Explanation: Exists is used to check for the existence of tuples.

69.

SELECT dept_name, ID, avg (salary)
FROM instructor
GROUP BY dept_name;
This statement **IS** erroneous because

a) Avg(salary) should not be selected
b) Dept_id should not be used in group by clause
c) Misplaced group by clause
d) Group by clause is not valid in this query

Answer: b
Explanation: Any attribute that is not present in the group by clause must appear only inside an aggregate function if it appears in the select clause, otherwise the query is treated as erroneous.

70. SQL applies predicates in the _____ clause after groups have been formed, so aggregate functions may be used.
a) Group by
b) With
c) Where
d) Having

Answer: b
Explanation: The with clause provides away of defining a temporary relation whose definition is available only to the query in which the with clause occurs.

71. Aggregate functions can be used in the select list or the_____clause of a select statement or subquery. They cannot be used in a _____ clause.
a) Where, having
b) Having, where
c) Group by, having
d) Group by, where

Answer: b
Explanation: To include aggregate functions having clause must be included after where.

72. The _____ keyword is used to access attributes of preceding tables or subqueries in the from clause.
a) In
b) Lateral
c) Having
d) With

Answer: b
Explanation:
Eg : **SELECT** name, salary, avg salary
FROM instructor I1, lateral
(**SELECT** avg(salary) **AS** avg salary

```
        FROM instructor I2
        WHERE I2.dept name= I1.dept
name);
```

Without the lateral clause, the subquery cannot access the correlation variable
I1 from the outer query.

73. Which of the following creates a temporary relation for the query on which it is defined?
a) With
b) From
c) Where
d) Select

Answer: a
Explanation: The with clause provides a way of defining a temporary relation whose definition is available only to the query in which the with clause occurs.

74. A Delete command operates on _____ relation.
a) One
b) Two
c) Several
d) Null

Answer: a
Explanation: Delete can delete from only one table at a time.

75.

Delete from r where P;

The above command
a) Deletes a particular tuple from the relation
b) Deletes the relation
c) Clears all entries from the relation
d) All of the mentioned

Answer: a
Explanation: Here P gives the condition for deleting specific rows.

76. Which one of the following deletes all the entries but keeps the structure of the relation.
a) Delete from r where P;
b) Delete from instructor where dept name= 'Finance';
c) Delete from instructor where salary between 13000 and 15000;
d) Delete from instructor;

Answer: d
Explanation: Absence of condition deletes all rows.

77. _____ are useful in SQL update statements, where they can be used in the set clause.
a) Multiple queries
b) Sub queries
c) Update
d) Scalar subqueries

Answer: d
Explanation: None.

78. The problem of ordering the update in multiple updates is avoided using
a) Set
b) Where
c) Case
d) When

Answer: c
Explanation: The case statements can add the order of updating tuples.

79. Which of the following creates a virtual relation for storing the query?
a) Function
b) View
c) Procedure
d) None of the mentioned

Answer: b
Explanation: Any such relation that is not part of the logical model, but is made visible to a user as a virtual relation, is called a view.

80. Which of the following is the syntax for views where v is view name?
a) Create view v as “query name”;
b) Create “query expression” as view;
c) Create view v as “query expression”;
d) Create view “query expression”;

Answer: c
Explanation: <query expression> is any legal query expression. The view name is represented by v.

81.

```
SELECT course_id
FROM physics_fall_2009
WHERE building= ’Watson’;
```

Here the tuples are selected from the view.Which one denotes the view.
a) Course_id
b) Watson
c) Building
d) physics_fall_2009

Answer: c
Explanation: View names may appear in a query any place where a relation name may appear.

82. Materialised views make sure that
a) View definition is kept stable
b) View definition is kept up-to-date
c) View definition is verified for error
d) View is deleted after specified time

Answer: b
Explanation: None.

83. Updating the value of the view
a) Will affect the relation from which it is defined
b) Will not change the view definition
c) Will not affect the relation from which it is defined

d) Cannot determine

Answer: a
Explanation: None.

84. SQL view is said to be updatable (that is, inserts, updates or deletes can be applied on the view) if which of the following conditions are satisfied by the query defining the view?
a) The from clause has only one database relation
b) The query does not have a group by or having clause
c) The select clause contains only attribute names of the relation and does not have any expressions, aggregates, or distinct specification
d) All of the mentioned

Answer: d
Explanation: All of the conditions must be satisfied to update the view in sql.

85. Which of the following is used at the end of the view to reject the tuples which do not satisfy the condition in where clause?
a) With
b) Check
c) With check
d) All of the mentioned

Answer: c
Explanation: Views can be defined with a with check option clause at the end of the view definition; then, if a tuple inserted into the view does not satisfy the view's where clause condition, the insertion is rejected by the database system.

86. For the view Create view instructor_info as

```
        SELECT ID, name, building
        FROM instructor,
department
        WHERE instructor.dept
name= department.dept name;
```

If we insert tuple into the view as insert into instructor info values ('69987', 'White', 'Taylor');
What will be the values of the other attributes in instructor and department relations?
a) Default value
b) Null
c) Error statement
d) 0

Answer: b
Explanation: The values take null if there is no constraint in the attribute else it is an Erroneous statement.

87.

```
CREATE VIEW faculty AS
  SELECT ID, name, dept name
```

```
FROM instructor;
```

Find the error in this query.
a) Instructor
b) Select
c) View …as
d) None of the mentioned

Answer: d
Explanation: Syntax is – create view v as <query expression>;.

88. A _____ consists of a sequence of query and/or update statements.
a) Transaction
b) Commit
c) Rollback
d) Flashback

Answer: a
Explanation: Transaction is a set of operation until commit.

89. Which of the following makes the transaction permanent in the database?
a) View
b) Commit
c) Rollback
d) Flashback

Answer: b
Explanation: Commit work commits the current transaction.

90. In order to undo the work of transaction after last commit which one should be used?
a) View
b) Commit
c) Rollback
d) Flashback

Answer: c
Explanation: Rollback work causes the current transaction to be rolled back; that is, it undoes all the updates performed by the SQL statements in the transaction.

91. Consider the following action:

```
TRANSACTION.....
Commit;
ROLLBACK;
```

What does Rollback do?
a) Undoes the transactions before commit
b) Clears all transactions
c) Redoes the transactions before commit
d) No action

Answer: d
Explanation: Once a transaction has executed commit work, its effects can no longer be undone by rollback work.

92. In case of any shut down during transaction before commit which of the following statement is done automatically?

a) View
b) Commit
c) Rollback
d) Flashback

Answer: c
Explanation: Once a transaction has executed commit work, its effects can no longer be undone by rollback work.

93. In order to maintain the consistency during transactions, database provides
a) Commit
b) Atomic
c) Flashback
d) Retain

Answer: b
Explanation: By atomic, either all the effects of the transaction are reflected in the database, or none are (after rollback).

94. Transaction processing is associated with everything below except
a) Conforming an action or triggering a response
b) Producing detail summary or exception report
c) Recording a business activity
d) Maintaining a data

Answer: a
Explanation: None.

95. A transaction completes its execution is said to be
a) Committed
b) Aborted
c) Rolled back
d) Failed

Answer: a
Explanation: A complete transaction always commits.

96. Which of the following is used to get back all the transactions back after rollback?
a) Commit
b) Rollback
c) Flashback
d) Redo

Answer: c
Explanation: None.

97. _____ will undo all statements up to commit?
a) Transaction
b) Flashback
c) Rollback
d) Abort

Answer: c
Explanation: Flashback will undo all the statements and Abort will terminate the operation.

98. To include integrity constraint in an existing relation use :
a) Create table

b) Modify table
c) Alter table
d) Drop table

Answer: c
Explanation: SYNTAX – alter table table-name add constraint, where constraint can be any constraint on the relation.

99. Which of the following is not an integrity constraint?
a) Not null
b) Positive
c) Unique
d) Check 'predicate'

Answer: b
Explanation: Positive is a value and not a constraint.

100.

```
CREATE TABLE Employee(Emp_id NUMERIC NOT NULL, Name VARCHAR(20) , dept_name VARCHAR(20), Salary NUMERIC UNIQUE(Emp_id,Name));
INSERT INTO Employee VALUES(1002, Ross, CSE, 10000)
INSERT INTO Employee VALUES(1006,Ted,Finance, );
INSERT INTO Employee VALUES(1002,Rita,Sales,20000);
```

What will be the result of the query?
a) All statements executed
b) Error in create statement
c) Error in insert into Employee values(1006,Ted,Finance,);
d) Error in insert into Employee values(1008,Ross,Sales,20000);

Answer: d
Explanation: The not null specification prohibits the insertion of a null value for the attribute. The unique specification says that no two tuples in the relation can be equal on all the listed attributes.

101.

```
CREATE TABLE Manager(ID NUMERIC,Name VARCHAR(20),budget NUMERIC,Details VARCHAR(30));
```

Inorder to ensure that the value of budget is non-negative which of the following should be used?
a) Check(budget>0)
b) Check(budget<0)
c) Alter(budget>0)
d) Alter(budget<0)

Answer: a
Explanation: A common use of the check clause is to ensure that attribute values satisfy specified conditions, in effect creating a powerful type system.

102. Foreign key is the one in which the _____ of one relation is referenced in another relation.
a) Foreign key
b) Primary key
c) References
d) Check constraint

Answer: b
Explanation: The foreign-key declaration specifies that for each course tuple, the department name specified in the tuple must exist in the department relation.

103.

```
CREATE TABLE course
( . . .
FOREIGN KEY (dept name)
REFERENCES department
. . . );
```

Which of the following is used to delete the entries in the referenced table when the tuple is deleted in course table?
a) Delete
b) Delete cascade
c) Set null
d) All of the mentioned

Answer: b
Explanation: The delete “cascades” to the course relation, deletes the tuple that refers to the department that was deleted.

104. Domain constraints, functional dependency and referential integrity are special forms of _____
a) Foreign key
b) Primary key
c) Assertion
d) Referential constraint

Answer: c
Explanation: An assertion is a predicate expressing a condition we wish the database to always satisfy.

105. Which of the following is the right syntax for the assertion?
a) Create assertion ‘assertion-name’ check ‘predicate’;
b) Create assertion check ‘predicate’ ‘assertion-name’;
c) Create assertions ‘predicates’;
d) All of the mentioned

Answer: a
Explanation: None.

106. Data integrity constraints are used to:
a) Control who is allowed access to the data
b) Ensure that duplicate records are not entered into the table
c) Improve the quality of data entered for a specific property (i.e., table column)
d) Prevent users from changing the values stored in the table

Answer: c
Explanation: None.

107. Which of the following can be addressed by enforcing a referential integrity constraint?
a) All phone numbers must include the area code
b) Certain fields are required (such as the email address, or phone number) before the record is accepted
c) Information on the customer must be known before anything can be sold to that customer
d) When entering an order quantity, the user must input a number and not some text (i.e., 12 rather than 'a dozen')

Answer: c
Explanation: The information can be referred to and obtained.

108. Dates must be specified in the format
a) mm/dd/yy
b) yyyy/mm/dd
c) dd/mm/yy
d) yy/dd/mm

Answer: b
Explanation: yyyy/mm/dd is the default format in sql.

109. A _____ on an attribute of a relation is a data structure that allows the database system to find those tuples in the relation that have a specified value for that attribute efficiently, without scanning through all the tuples of the relation.
a) Index
b) Reference
c) Assertion
d) Timestamp

Answer: a
Explanation: Index is the reference to the tuples in a relation.

110.

Create index studentID_index on student(ID);

Here which one denotes the relation for which index is created?
a) StudentID_index
b) ID
c) StudentID
d) Student

Answer: d
Explanation: The statement creates an index named studentID index on the attribute ID of the relation student.

111. Which of the following is used to store movie and image files?
a) Clob
b) Blob
c) Binary

d) Image

Answer: b
Explanation: SQL therefore provides large-object data types for character data (clob) and binary data (blob). The letters “lob” in these data types stand for “Large OBject”.

112. The user defined data type can be created using
a) Create datatype
b) Create data
c) Create definetype
d) Create type

Answer: d
Explanation: The create type clause can be used to define new types.Syntax : create type Dollars as numeric(12,2) final; .

113. Values of one type can be converted to another domain using which of the following?
a) Cast
b) Drop type
c) Alter type
d) Convert

Answer: a
Explanation: Example of cast :cast (department.budget to numeric(12,2)). SQL provides drop type and alter type clauses to drop or modify types that have been created earlier.

114.

CREATE DOMAIN YearlySalary **NUMERIC**(8,2)
CONSTRAINT salary **VALUE** test _____;

In order to ensure that an instructor’s salary domain allows only values greater than a specified value use:
a) Value>=30000.00
b) Not null;
c) Check(value >= 29000.00);
d) Check(value)

Answer: c
Explanation: Check(value ‘condition’) is the syntax.

115. Which of the following closely resembles Create view?
a) Create table . . .like
b) Create table . . . as
c) With data
d) Create view as

Answer: b
Explanation: The ‘create table . . . as’ statement closely resembles the create view statement and both are defined by using queries. The main difference is that the contents of the table are set when the table is created, whereas the contents of a view always reflect the current query result.

116. In contemporary databases, the top level of the hierarchy consists of _____ each of which can contain _____
a) Catalogs, schemas
b) Schemas, catalogs
c) Environment, schemas
d) Schemas, Environment

Answer: a
Explanation: None.

117. Which of the following statements creates a new table temp instructor that has the same schema as an instructor.
a) create table temp_instructor;
b) Create table temp_instructor like instructor;
c) Create Table as temp_instructor;
d) Create table like temp_instructor;

Answer: b
Explanation: None.

118. The database administrator who authorizes all the new users, modifies the database and takes grants privilege is
a) Super user
b) Administrator
c) Operator of operating system
d) All of the mentioned

Answer: d
Explanation: The authorizations provided by the administrator to the user is a privilege.

119. Which of the following is a basic form of grant statement?
a)

```
GRANT 'privilege list'
ON 'relation name or view name'
TO 'user/role list';
```

b)

```
GRANT 'privilege list'
ON 'user/role list'
TO 'relation name or view name';
```

c)

```
GRANT 'privilege list'
TO 'user/role list'
```

d)

```
GRANT 'privilege list'
ON 'relation name or view name'
ON 'user/role list';
```

Answer: a
Explanation: The privilege list allows the granting of several privileges in one command .

120. Which of the following is used to provide privilege to only a particular attribute?
a) Grant select on employee to Amit
b) Grant update(budget) on

department to Raj
c) Grant update(budget,salary,Rate) on department to Raj
d) Grant delete to Amit

Answer: b
Explanation: This grant statement gives user Raj update authorization on the budget attribute of the department relation.

121. Which of the following statement is used to remove the privilege from the user Amir?
a) Remove update on department from Amir
b) Revoke update on employee from Amir
c) Delete select on department from Raj
d) Grant update on employee from Amir

Answer: b
Explanation: revoke on from ;

122. Which of the following is used to provide delete authorization to instructor?
a)

```
CREATE ROLE instructor ;
GRANT DELETE TO instructor;
```

b)

```
CREATE ROLE instructor;
GRANT SELECT ON takes
TO instructor;
```

c)

```
CREATE ROLE instructor;
GRANT DELETE ON takes
TO instructor;
```

d) All of the mentioned

Answer: c
Explanation: The role is first created and the authorization is given on relation takes to the role.

123. Which of the following is true regarding views?
a) The user who creates a view cannot be given update authorization on a view without having update authorization on the relations used to define the view
b) The user who creates a view cannot be given update authorization on a view without having update authorization on the relations used to define the view
c) If a user creates a view on which no authorization can be granted, the system will allow the view creation request
d) A user who creates a view receives all privileges on that view

Answer: c
Explanation: A user who creates a

view does not necessarily receive all privileges on that view.

124. If we wish to grant a privilege and to allow the recipient to pass the privilege on to other users, we append the _____ clause to the appropriate grant command.
a) With grant
b) Grant user
c) Grant pass privelege
d) With grant option

Answer: d
Explanation: None.

125. In authorization graph, if DBA provides authorization to u1 which inturn gives to u2 which of the following is correct?
a) If DBA revokes authorization from u1 then u2 authorization is also revoked
b) If u1 revokes authorization from u2 then u2 authorization is revoked
c) If DBA & u1 revokes authorization from u1 then u2 authorization is also revoked
d) If u2 revokes authorization then u1 authorization is revoked

Answer: c
Explanation: A user has an authorization if and only if there is a path from the root of the authorization graph down to the node representing the user.

126. Which of the following is used to avoid cascading of authorizations from the user?
a) Granted by current role
b) Revoke select on department from Amit, Satoshi restrict;
c) Revoke grant option for select on department from Amit;
d) Revoke select on department from Amit, Satoshi cascade;

Answer: b
Explanation: The revoke statement may specify restrict in order to prevent cascading revocation. The keyword cascade can be used instead of restrict to indicate that revocation should cascade.

127. The granting and revoking of roles by the user may cause some confusions when that user role is revoked. To overcome the above situation
a) The privilege must be granted only by roles
b) The privilege is granted by roles and users
c) The user role cannot be removed once given
d) By restricting the user access to the roles

Answer: a
Explanation: The current role associated with a session can be set by executing set role name. The

specified role must have been granted to the user, else the set role statement fails.

128. A _____ is a special kind of a store procedure that executes in response to certain action on the table like insertion, deletion or updation of data.
a) Procedures
b) Triggers
c) Functions
d) None of the mentioned

Answer: b
Explanation: Triggers are automatically generated when a particular operation takes place.

129. Triggers are supported in
a) Delete
b) Update
c) Views
d) All of the mentioned

Answer: c
Explanation: The triggers run after an insert, update or delete on a table. They are not supported for views.

130. The CREATE TRIGGER statement is used to create the trigger. THE _____ clause specifies the table name on which the trigger is to be attached. The _____ specifies that this is an AFTER INSERT trigger.
a) for insert, on
b) On, for insert
c) For, insert
d) None of the mentioned

Answer: b
Explanation: The triggers run after an insert, update or delete on a table. They are not supported for views.

131. What are the after triggers?
a) Triggers generated after a particular operation
b) These triggers run after an insert, update or delete on a table
c) These triggers run after an insert, views, update or delete on a table
d) All of the mentioned

Answer: b
Explanation: AFTER TRIGGERS can be classified further into three types as: AFTER INSERT Trigger, AFTER UPDATE Trigger, AFTER DELETE Trigger.

132. The variables in the triggers are declared using
a) –
b) @
c) /
d) /@

Answer: b
Explanation: Example : declare @empid int; where empid is the variable.

133. The default extension for an Oracle SQL*Plus file is:
a) .txt
b) .pls
c) .ora
d) .sql

Answer: d
Explanation: Example :None.

134. Which of the following is NOT an Oracle-supported trigger?
a) BEFORE
b) DURING
c) AFTER
d) INSTEAD OF

Answer: b
Explanation: Example: During trigger is not possible in any database.

135. What are the different in triggers?
a) Define, Create
b) Drop, Comment
c) Insert, Update, Delete
d) All of the mentioned

Answer: c
Explanation: Triggers are not possible for create, drop.

136. Triggers _____ enabled or disabled
a) Can be
b) Cannot be
c) Ought to be
d) Always

Answer: a
Explanation: Triggers can be manipulated.

137. Which prefixes are available to Oracle triggers?
a) : new only
b) : old only
c) Both :new and : old
d) Neither :new nor : old

Answer: c
Explanation: None.

138. OLAP stands for
a) Online analytical processing
b) Online analysis processing
c) Online transaction processing
d) Online aggregate processing

Answer: a
Explanation: OLAP is the manipulation of information to support decision making.

139. Data that can be modeled as dimension attributes and measure attributes are called _____ data.
a) Multidimensional
b) Singledimensional
c) Measured
d) Dimensional

Answer: a
Explanation: Given a relation used for

data analysis, we can identify some of its attributes as measure attributes, since they measure some value, and can be aggregated upon.Dimension attribute define the dimensions on which measure attributes, and summaries of measure attributes, are viewed.

140. The generalization of cross-tab which is represented visually is _____ which is also called as data cube.
a) Two dimensional cube
b) Multidimensional cube
c) N-dimensional cube
d) Cuboid

Answer: a
Explanation: Each cell in the cube is identified for the values for the three dimensional attributes.

141. The process of viewing the cross-tab (Single dimensional) with a fixed value of one attribute is
a) Slicing
b) Dicing
c) Pivoting
d) Both Slicing and Dicing

Answer: a
Explanation: The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Dice selects two or more dimensions from a given cube and provides a new sub-cube.

142. The operation of moving from finer-granularity data to a coarser granularity (by means of aggregation) is called a _____
a) Rollup
b) Drill down
c) Dicing
d) Pivoting

Answer: a
Explanation: The opposite operation—that of moving fromcoarser-granularity data to finer-granularity data—is called a drill down.

143. In SQL the cross-tabs are created using
a) Slice
b) Dice
c) Pivot
d) All of the mentioned

Answer: a
Explanation: Pivot (sum(quantity) for color in ('dark','pastel','white')).

144.

{ (item name, color, clothes size), (item name, color), (item name, clothes size), (color, clothes size), (item name), (color), (clothes size), () }

This can be achieved by using which of the following ?
a) group by rollup
b) group by cubic
c) group by
d) none of the mentioned

Answer: d
Explanation: 'Group by cube' is used .

145. What do data warehouses support?
a) OLAP
b) OLTP
c) OLAP and OLTP
d) Operational databases

Answer: a
Explanation: None.

146.

```
SELECT item name, color, clothes
SIZE, SUM(quantity)
FROM sales
GROUP BY rollup(item name, color,
clothes SIZE);
```

How many grouping is possible in this rollup?
a) 8
b) 4
c) 2
d) 1

Answer: b
Explanation: { (item name, color, clothes size), (item name, color), (item name), () }.

147. Which one of the following is the right syntax for DECODE?
a) DECODE (search, expression, result [, search, result]… [, default])
b) DECODE (expression, result [, search, result]… [, default], search)
c) DECODE (search, result [, search, result]… [, default], expression)
d) DECODE (expression, search, result [, search, result]… [, default])

Answer: d
Explanation: None.

148. Relational Algebra is a _____ query language that takes two relations as input and produces another relation as an output of the query.
a) Relational
b) Structural
c) Procedural
d) Fundamental

Answer: c
Explanation: This language has fundamental and other operations which are used on relations.

149. Which of the following is a fundamental operation in relational algebra?
a) Set intersection
b) Natural join

c) Assignment
d) None of the mentioned

Answer: d
Explanation: The fundamental operations are select, project, union, set difference, Cartesian product, and rename.

150. Which of the following is used to denote the selection operation in relational algebra?
a) Pi (Greek)
b) Sigma (Greek)
c) Lambda (Greek)
d) Omega (Greek)

Answer: b
Explanation: The select operation selects tuples that satisfy a given predicate.

151. For select operation the _____ appear in the subscript and the _____ argument appears in the paranthesis after the sigma.
a) Predicates, relation
b) Relation, Predicates
c) Operation, Predicates
d) Relation, Operation

Answer: a
Explanation: None.

152. The _____ operation, denoted by −, allows us to find tuples that are in one relation but are not in another.
a) Union
b) Set-difference
c) Difference
d) Intersection

Answer: b
Explanation: The expression r − s produces a relation containing those tuples in r but not in s.

153. Which is a unary operation:
a) Selection operation
b) Primitive operation
c) Projection operation
d) Generalized selection

Answer: d
Explanation: Generalization Selection takes only one argument for operation.

154. Which is a join condition contains an equality operator:
a) Equijoins
b) Cartesian
c) Natural
d) Left

Answer: a
Explanation: None.

155. In precedence of set operators, the expression is evaluated from
a) Left to left
b) Left to right
c) Right to left

d) From user specification

Answer: b
Explanation: The expression is evaluated from left to right according to the precedence.

156. Which of the following is not outer join?
a) Left outer join
b) Right outer join
c) Full outer join
d) All of the mentioned

Answer: d
Explanation: The FULL OUTER JOIN keyword combines the result of both LEFT and RIGHT joins.

157. The assignment operator is denoted by
a) ->
b) <-
c) =
d) ==

Answer: b
Explanation: The result of the expression to the right of the ← is assigned to the relation variable on the left of the ←.

158. Find the ID, name, dept name, salary for instructors whose salary is greater than $80,000 .
a) {t | t ε instructor ∧ t[salary] > 80000}
b) ∃ t ∈ r (Q(t))
c) {t | ∃ s ε instructor (t[ID] = s[ID]∧ s[salary] > 80000)}
d) None of the mentioned

Answer: a
Explanation: This expression is in tuple relational format.

159. A query in the tuple relational calculus is expressed as:
a) {t | P() | t}
b) {P(t) | t }
c) {t | P(t)}
d) All of the mentioned

Answer: c
Explanation: The tuple relational calculus, is a nonprocedural query language. It describes the desired information without giving a specific procedure for obtaining that information.

160.

{t | ∃ s ε instructor (t[name] = s[name]
∧ ∃ u ε department (u[dept name] = s[dept name]
∧ u[building] = “Watson”))}

Which of the following best describes the query?
a) Finds the names of all instructors whose department is in the Watson building

b) Finds the names of all department is in the Watson building
c) Finds the name of the dapartment whose instructor and building is Watson
d) Returns the building name of all the departments

Answer: a
Explanation: This query has two "there exists" clauses in our tuple-relational-calculus expression, connected by and (∧).

161. Which of the following symbol is used in the place of except?
a) ^
b) V
c) ¬
d) ~

Answer: c
Explanation: The query ¬P negates the value of P.

162. "Find all students who have taken all courses offered in the Biology department." The expressions that matches this sentence is :
a) Ǝ t ε r (Q(t))
b) ∀ t ε r (Q(t))
c) ¬ t ε r (Q(t))
d) ~ t ε r (Q(t))

Answer: b
Explanation: ∀ is used denote "for all" in SQL.

163. An _____ is a set of entities of the same type that share the same properties, or attributes.
a) Entity set
b) Attribute set
c) Relation set
d) Entity model

Answer: a
Explanation: An entity is a "thing" or "object" in the real world that is distinguishable from all other objects.

164. Entity is a _____
a) Object of relation
b) Present working model
c) Thing in real world
d) Model of relation

Answer: c
Explanation: For example, each person in a university is an entity.

165. The descriptive property possessed by each entity set is _____
a) Entity
b) Attribute
c) Relation
d) Model

Answer: b
Explanation: Possible attributes of

the instructor entity set are ID, name, dept name, and salary.

166. The function that an entity plays in a relationship is called that entity's _____

a) Participation
b) Position
c) Role
d) Instance

Answer: c
Explanation: A relationship is an association among several entities.

167. The attribute *name* could be structured as an attribute consisting of first name, middle initial, and last name. This type of attribute is called
a) Simple attribute
b) Composite attribute
c) Multivalued attribute
d) Derived attribute

Answer: b
Explanation: Composite attributes can be divided into subparts (that is, other attributes).

168. The attribute AGE is calculated from DATE_OF_BIRTH. The attribute AGE is
a) Single valued
b) Multi valued
c) Composite
d) Derived

Answer: d
Explanation: The value for this type of attribute can be derived from the values of other related attributes or entities.

169. Not applicable condition can be represented in relation entry as
a) NA
b) 0
c) NULL
d) Blank Space

Answer: c
Explanation: NULL always represents that the value is not present.

170. Which of the following can be a multivalued attribute?
a) Phone_number
b) Name
c) Date_of_birth
d) All of the mentioned

Answer: a
Explanation: Name and Date_of_birth cannot hold more than 1 value.

171. Which of the following is a single valued attribute
a) Register_number
b) Address
c) SUBJECT_TAKEN
d) Reference

Answer: a
Explanation: None.

172. In a relation between the entities the type and condition of the relation should be specified. That is called as_____attribute.
a) Desciptive
b) Derived
c) Recursive
d) Relative

Answer: a
Explanation: Consider the entity sets student and section, which participate in a relationship set takes. We may wish to store a descriptive attribute grade with the relationship to record the grade that a student got in the class.

172. _____ express the number of entities to which another entity can be associated via a relationship set.
a) Mapping Cardinality
b) Relational Cardinality
c) Participation Constraints
d) None of the mentioned

Answer: a
Explanation: Mapping cardinality is also called as cardinality ratio.

173. An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.This is called as
a) One-to-many
b) One-to-one
c) Many-to-many
d) Many-to-one

Answer: b
Explanation: Here one entity in one set is related to one one entity in other set.

174. An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A.
a) One-to-many
b) One-to-one
c) Many-to-many
d) Many-to-one

Answer: d
Explanation: Here more than one entity in one set is related to one one entity in other set.

175. Data integrity constraints are used to:
a) Control who is allowed access to the data
b) Ensure that duplicate records are not entered into the table
c) Improve the quality of data entered for a specific property
d) Prevent users from changing the values stored in the table

Answer: c
Explanation: The data entered will be in a particular cell (i.e., table column).

176. Establishing limits on allowable property values, and specifying a set of acceptable, predefined options that can be assigned to a property are examples of:
a) Attributes
b) Data integrity constraints
c) Method constraints
d) Referential integrity constraints

Answer: b
Explanation: Only particular value satisfying the constraints are entered in the column.

177. Which of the following can be addressed by enforcing a referential integrity constraint?
a) All phone numbers must include the area code
b) Certain fields are required (such as the email address, or phone number) before the record is accepted
c) Information on the customer must be known before anything can be sold to that customer
d) Then entering an order quantity, the user must input a number and not some text (i.e., 12 rather than 'a dozen')

Answer: c
Explanation: None.

178. _____ is a special type of integrity constraint that relates two relations & maintains consistency across the relations.
a) Entity Integrity Constraints
b) Referential Integrity Constraints
c) Domain Integrity Constraints
d) Domain Constraints

Answer: b
Explanation: None.

179. Which one of the following uniquely identifies the elements in the relation?
a) Secondary Key
b) Primary key
c) Foreign key
d) Composite key

Answer: b
Explanation: Primary key checks for not null and uniqueness constraint.

180. Drop Table cannot be used to drop a table referenced by a _____ constraint.
a) Local Key
b) Primary Key
c) Composite Key
d) Foreign Key

Answer: d
Explanation: Foreign key is used when primary key of one relation is used in another relation.

181. _____ is preferred method for enforcing data integrity
a) Constraints

b) Stored Procedure
c) Triggers
d) Cursors

Answer: a
Explanation: Constraints are specified to restrict entries in the relation.

182. Which of the following gives a logical structure of the database graphically?
a) Entity-relationship diagram
b) Entity diagram
c) Database diagram
d) Architectural representation

Answer: a
Explanation: E-R diagrams are simple and clear—qualities that may well account in large part for the widespread use of the E-R model.

183. The entity relationship set is represented in E-R diagram as
a) Double diamonds
b) Undivided rectangles
c) Dashed lines
d) Diamond

Answer: d
Explanation: Dashed lines link attributes of a relationship set to the relationship set.

184. The Rectangles divided into two parts represents
a) Entity set
b) Relationship set
c) Attributes of a relationship set
d) Primary key

Answer: a
Explanation: The first part of the rectangle, contains the name of the entity set. The second part contains the names of all the attributes of the entity set.

185. Consider a directed line(->) from the relationship set advisor to both entity sets instructor and student. This indicates _____ cardinality
a) One to many
b) One to one
c) Many to many
d) Many to one

Answer: b
Explanation: This indicates that an instructor may advise at most one student, and a student may have at most one advisor.

186. We indicate roles in E-R diagrams by labeling the lines that connect _____ to _____
a) Diamond , diamond
b) Rectangle, diamond
c) Rectangle, rectangle
d) Diamond, rectangle

Answer: d
Explanation: Diamond represents a

relationship set and rectangle represents a entity set.

187. An entity set that does not have sufficient attributes to form a primary key is termed a _____
a) Strong entity set
b) Variant set
c) Weak entity set
d) Variable set

Answer: c
Explanation: An entity set that has a primary key is termed a strong entity set.

188. For a weak entity set to be meaningful, it must be associated with another entity set, called the
a) Identifying set
b) Owner set
c) Neighbour set
d) Strong entity set

Answer: a
Explanation: Every weak entity must be associated with an identifying entity; that is, the weak entity set is said to be existence dependent on the identifying entity set. The identifying entity set is said to own the weak entity set that it identifies. It is also called as owner entity set.

189. Weak entity set is represented as
a) Underline
b) Double line
c) Double diamond
d) Double rectangle

Answer: c
Explanation: An entity set that has a primary key is termed a strong entity set.

190. If you were collecting and storing information about your music collection, an album would be considered a(n) _____
a) Relation
b) Entity
c) Instance
d) Attribute

Answer: b
Explanation: An entity set is a logical container for instances of an entity type and instances of any type derived from that entity type.

191. What term is used to refer to a specific record in your music database; for instance; information stored about a specific album?
a) Relation
b) Instance
c) Table
d) Column

Answer: b
Explanation: The environment of database is said to be an instance. A database instance or an 'instance' is

made up of the background processes needed by the database.

192. The total participation by entities is represented in E-R diagram as
a) Dashed line
b) Double line
c) Double rectangle
d) Circle

Answer: b
Explanation: It is used to represent the relation between several attributes.

193. Given the basic ER and relational models, which of the following is INCORRECT?
a) An attribute of an entity can have more than one value
b) An attribute of an entity can be composite
c) In a row of a relational table, an attribute can have more than one value
d) In a row of a relational table, an attribute can have exactly one value or a NULL value

Answer: c
Explanation: It is possible to have several values for a single attribute provide it is a multi-valued attribute.

194. Which of the following indicates the maximum number of entities that can be involved in a relationship?
a) Minimum cardinality
b) Maximum cardinality
c) ERD
d) Greater Entity Count

Answer: b
Explanation: In SQL (Structured Query Language), the term cardinality refers to the uniqueness of data values contained in a particular column (attribute) of a database table.

195. In E-R diagram generalization is represented by
a) Ellipse
b) Dashed ellipse
c) Rectangle
d) Triangle

Answer: d
Explanation: Ellipse represents attributes, rectangle represents entity.

196. The entity set person is classified as student and employee. This process is called _____
a) Generalization
b) Specialization
c) Inheritance
d) Constraint generalization

Answer: b
Explanation: The process of designating subgroupings within an entity set is called specialization.

197. Which relationship is used to represent a specialization entity?
a) ISA
b) AIS
c) ONIS
d) WHOIS

Answer: a
Explanation: In terms of an E-R diagram, specialization is depicted by a hollow arrow-head pointing from the specialized entity to the other entity.

198. The refinement from an initial entity set into successive levels of entity subgroupings represents a _____ design process in which distinctions are made explicit.
a) Hierarchy
b) Bottom-up
c) Top-down
d) Radical

Answer: c
Explanation: The design process may also proceed in a bottom-up manner, in which multiple entity sets are synthesized into a higher-level entity set on the basis of common features.

199. There are similarities between the instructor entity set and the secretary entity set in the sense that they have several attributes that are conceptually the same across the two entity sets: namely, the identifier, name, and salary attributes. This process is called
a) Commonality
b) Specialization
c) Generalization
d) Similarity

Answer: c
Explanation: Generalization is used to emphasize the similarities among lower-level entity sets and to hide the differences.

200. If an entity set is a lower-level entity set in more than one ISA relationship, then the entity set has
a) Hierarchy
b) Multilevel inheritance
c) Single inheritance
d) Multiple inheritance

Answer: d
Explanation: The attributes of the higher-level entity sets are said to be inherited by the lower-level entity sets.

201. A _____ constraint requires that an entity belong to no more than one lower-level entity set.
a) Disjointness
b) Uniqueness
c) Special
d) Relational

Answer: a
Explanation: For example, student

entity can satisfy only one condition for the student type attribute; an entity can be either a graduate student or an undergraduate student, but cannot be both.

202. Consider the employee work-team example, and assume that certain employees participate in more than one work team. A given employee may therefore appear in more than one of the team entity sets that are lower level entity sets of employee. Thus, the generalization is _____

a) Overlapping
b) Disjointness
c) Uniqueness
d) Relational

Answer: a
Explanation: In overlapping generalizations, the same entity may belong to more than one lower-level entity set within a single generalization.

203. In the _____ normal form, a composite attribute is converted to individual attributes.
a) First
b) Second
c) Third
d) Fourth

Answer: a
Explanation: The first normal form is used to eliminate the duplicate information.

204. A table on the many side of a one to many or many to many relationship must:
a) Be in Second Normal Form (2NF)
b) Be in Third Normal Form (3NF)
c) Have a single attribute key
d) Have a composite key

Answer: d
Explanation: The relation in second normal form is also in first normal form and no partial dependencies on any column in primary key.

205. Tables in second normal form (2NF):
a) Eliminate all hidden dependencies
b) Eliminate the possibility of a insertion anomalies
c) Have a composite key
d) Have all non key fields depend on the whole primary key

Answer: a
Explanation: The relation in second normal form is also in first normal form and no partial dependencies on any column in primary key.

206. Which-one ofthe following statements about normal forms is FALSE?
a) BCNF is stricter than 3 NF
b) Lossless, dependency -preserving

decomposition into 3 NF is always possible
c) Loss less, dependency – preserving decomposition into BCNF is always possible
d) Any relation with two attributes is BCNF

Answer: c
Explanation: We say that the decomposition is a lossless decomposition if there is no loss of information by replacing r (R) with two relation schemas r1(R1) andr2(R2).

207. Functional Dependencies are the types of constraints that are based on_____
a) Key
b) Key revisited
c) Superset key
d) None of the mentioned

Answer: a
Explanation: Key is the basic element needed for the constraints.

208. Which is a bottom-up approach to database design that design by examining the relationship between attributes:
a) Functional dependency
b) Database modeling
c) Normalization
d) Decomposition

Answer: c
Explanation: Normalisation is the process of removing redundancy and unwanted data.

209. Which forms simplifies and ensures that there are minimal data aggregates and repetitive groups:
a) 1NF
b) 2NF
c) 3NF
d) All of the mentioned

Answer: c
Explanation: The first normal form is used to eliminate the duplicate information.

210. Which forms has a relation that possesses data about an individual entity:
a) 2NF
b) 3NF
c) 4NF
d) 5NF

Answer: c
Explanation: A Table is in 4NF if and only if, for every one of its non-trivial multivalued dependencies X \twoheadrightarrow Y, X is a superkey—that is, X is either a candidate key or a superset thereof.

211. Which forms are based on the concept of functional dependency:
a) 1NF

b) 2NF
c) 3NF
d) 4NF

Answer: c
Explanation: The table is in 3NF if every non-prime attribute of R is non-transitively dependent (i.e. directly dependent) on every superkey of R.

212.

Empdt1(empcode, name, street, city, state, pincode).

For any pincode, there is only one city and state. Also, for given street, city and state, there is just one pincode. In normalization terms, empdt1 is a relation in
a) 1 NF only
b) 2 NF and hence also in 1 NF
c) 3NF and hence also in 2NF and 1NF
d) BCNF and hence also in 3NF, 2NF and 1NF

Answer: b
Explanation: The relation in second normal form is also in first normal form and no partial dependencies on any column in primary key.

213. We can use the following three rules to find logically implied functional dependencies. This collection of rules is called
a) Axioms
b) Armstrong's axioms
c) Armstrong
d) Closure

Answer: b
Explanation: By applying these rules repeatedly, we can find all of F+, given F.

214. An approach to website design with the emphasis on converting visitors to outcomes required by the owner is referred to as:
a) Web usability
b) Persuasion
c) Web accessibility
d) None of the mentioned

Answer: b
Explanation: In computing, graphical user interface is a type of user interface that allows users to interact with electronic devices.

125. A method of modelling and describing user tasks for an interactive application is referred to as:
a) Customer journey
b) Primary persona
c) Use case
d) Web design persona

Answer: c
Explanation: The actions in GUI are usually performed through direct

manipulation of the graphical elements.

216. Information architecture influences:
a) Answer choice
b) Site structure
c) Labeling
d) Navigation design

Answer: b
Explanation: The actions in GUI are usually performed through direct manipulation of the graphical elements.

217. Also known as schematics, a way of illustrating the layout of an individual webpage is a:
a) Wireframe
b) Sitemap
c) Card sorting
d) Blueprint

Answer: a
Explanation: An application programming interface specifies how some software components should interact with each other.

218. A graphical or text depiction of the relationship between different groups of content on a website is referred to as a:
a) Wireframe
b) Blueprint
c) Sitemap
d) Card sorting

Answer: c
Explanation: An application programming interface specifies how some software components should interact with each other.

219. Blueprints are intended to:
a) Prototype of the screen layout showing navigation and main design elements
b) Show the grouping of pages and user journeys
c) Indicate the structure of a site during site design and as a user feature
d) Prototype typical customer journeys or clickstreams through a website

Answer: c
Explanation: A blueprint is a reproduction of a technical drawing, documenting an architecture or an engineering design, using a contact print process.

220. Storyboards are intended to:
a) Indicate the structure of a site during site design and as a user feature
b) Prototype of the screen layout showing navigation and main design elements
c) Integrate consistently available components on the webpage (e.g.

navigation, search boxes)
d) Prototype typical customer journeys or click streams through a website

Answer: d
Explanation: An application programming interface specifies how some software components should interact with each other.

221. Which of the following occupies boot record of hard and floppy disks and activated during computer startup?
a) Worm
b) Boot sector virus
c) Macro virus
d) Virus

Answer: b
Explanation: A blueprint is a reproduction of a technical drawing, documenting an architecture or an engineering design, using a contact print process.

222. In ordered indices the file containing the records is sequentially ordered, a _____ is an index whose search key also defines the sequential order of the file.
a) Clustered index
b) Structured index
c) Unstructured index
d) Nonclustered index

Answer: a
Explanation: Clustering index are also called primary indices; the term primary index may appear to denote an index on a primary key, but such indices can in fact be built on any search key.

223. Indices whose search key specifies an order different from the sequential order of the file are called _____ indices.
a) Nonclustered
b) Secondary
c) All of the mentioned
d) None of the mentioned

Answer: c
Explanation: Nonclustering index is also called secondary indices.

224. An _____ consists of a search-key value and pointers to one or more records with that value as their search-key value.
a) Index entry
b) Index hash
c) Index cluster
d) Index map

Answer: a
Explanation: The pointer to a record consists of the identifier of a disk block and an offset within the disk block to identify the record within the block.

225. In a _____ clustering index, the index record contains the search-key value and a pointer to the first data record with that search-key value and the rest of the records will be in the sequential pointers.
a) Dense
b) Sparse
c) Straight
d) Continuous

Answer: a
Explanation: In a dense nonclustering index, the index must store a list of pointers to all records with the same search-key value.

226. In a _____ index, an index entry appears for only some of the search-key values.
a) Dense
b) Sparse
c) Straight
d) Continuous

Answer: a
Explanation: Sparse indices can be used only if the relation is stored in sorted order of the search key, that is if the index is a clustering index.

227. Incase the indices values are larger, index is created for these values of the index. This is called
a) Pointed index
b) Sequential index
c) Multilevel index
d) Multiple index

Answer: c
Explanation: Indices with two or more levels are called multilevel indices.

228. A search key containing more than one attribute is referred to as a _____ search key.
a) Simple
b) Composite
c) Compound
d) Secondary

Answer: b
Explanation: The structure of the index is the same as that of any other index, the only difference being that the search key is not a single attribute, but rather is a list of attributes.

229. In B+ tree the node which points to another node is called
a) Leaf node
b) External node
c) Final node
d) Internal node

Answer: d
Explanation: Nonleaf nodes are also referred to as internal nodes.

230. Insertion of a large number of entries at a time into an index is referred to as _____ of the

index.
a) Loading
b) Bulk insertion
c) Bulk loading
d) Increase insertion

Answer: c
Explanation: Bulk loading is used to improve efficiency and scalability.

231. While inserting the record into the index, if the search-key value does not appear in the index.
a) The system adds a pointer to the new record in the index entry
b) The system places the record being inserted after the other records with the same search-key values
c) The system inserts an index entry with the search-key value in the index at the appropriate position
d) None of the mentioned

Answer: c
Explanation: If the index entry stores pointers to all records with the same search key value, the system adds a pointer to the new record in the index entry.

232. A(n) _____ can be used to preserve the integrity of a document or a message.
a) Message digest
b) Message summary
c) Encrypted message
d) None of the mentioned

Answer: c
Explanation: Encryption algorithms are used to keep the contents safe.

233. A hash function must meet _____ criteria.
a) Two
b) Three
c) Four
d) None of the mentioned

Answer: b
Explanation: Only if the criteria is fulfilled the values are hashed.

234. What is the main limitation of Hierarchical Databases?
a) Limited capacity (unable to hold much data)
b) Limited flexibility in accessing data
c) Overhead associated with maintaining indexes
d) The performance of the database is poor

Answer: b
Explanation: In this, the data items are placed in a tree like hierarchical structure.

235. The property (or set of properties) that uniquely defines each row in a table is called the:
a) Identifier
b) Index

c) Primary key
d) Symmetric key

Answer: c
Explanation: Primary is used to uniquely identify the tuples.

236. The separation of the data definition from the program is known as:
a) Data dictionary
b) Data independence
c) Data integrity
d) Referential integrity

Answer: b
Explanation: Data dictionary is the place where the meaning of the data are organized.

237. Bitmap indices are a specialized type of index designed for easy querying on _____
a) Bit values
b) Binary digits
c) Multiple keys
d) Single keys

Answer: c
Explanation: Each bitmap index is built on a single key.

238. A _____ on the attribute A of relation r consists of one bitmap for each value that A can take.
a) Bitmap index
b) Bitmap
c) Index
d) Array

Answer: a
Explanation: A bitmap is simply an array of bits.

239.

```
SELECT *
FROM r
WHERE gender = 'f' AND income level = 'L2';
```

In this selection, we fetch the bitmaps for gender value f and the bitmap for income level value L2, and perform an _____ of the two bitmaps.
a) Union
b) Addition
c) Combination
d) Intersection

Answer: d
Explanation: We compute a new bitmap where bit i has value 1 if the ith bit of the two bitmaps are both 1, and has a value 0 otherwise.

240. To identify the deleted records we use the _____
a) Existence bitmap
b) Current bitmap
c) Final bitmap
d) Deleted bitmap

Answer: a
Explanation: The bitmaps which are deleted are denoted by 0.

241. What is the purpose of the index in sql server?
a) To enhance the query performance
b) To provide an index to a record
c) To perform fast searches
d) All of the mentioned

Answer: d
Explanation: A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes.

242. How many types of indexes are there in sql server?
a) 1
b) 2
c) 3
d) 4

Answer: b
Explanation: They are clustered index and non clustered index.

243. How non clustered index point to the data?
a) It never points to anything
b) It points to a data row
c) It is used for pointing data rows containing key values
d) None of the mentioned

Answer: c
Explanation: Nonclustered indexes have a structure separate from the data rows. A nonclustered index contains the nonclustered index key values and each key value entry has a pointer to the data row that contains the key value.

244. Which one is true about clustered index?
a) Clustered index is not associated with table
b) Clustered index is built by default on unique key columns
c) Clustered index is not built on unique key columns
d) None of the mentioned

Answer: b
Explanation: Nonclustered indexes have a structure separate from the data rows. A nonclustered index contains the nonclustered index key values and each key value entry has a pointer to the data row that contains the key value.

245. What is true about indexes?
a) Indexes enhance the performance even if the table is updated frequently
b) It makes harder for sql server engines to work to work on index which have large keys
c) It doesn’t make harder for sql server engines to work to work on

index which have large keys
d) None of the mentioned

Answer: b
Explanation: Indexes tend to improve the performance.

246. A collection of data designed to be used by different people is called a/an
a) Organization
b) Database
c) Relationship
d) Schema

Answer: b
Explanation: Database is a collection of related tables.

247. Which of the following is the oldest database model?
a) Relational
b) Deductive
c) Physical
d) Network

Answer: d
Explanation: The network model is a database model conceived as a flexible way of representing objects and their relationships.

248. Which of the following schemas does define a view or views of the database for particular users?
a) Internal schema
b) Conceptual schema
c) Physical schema
d) External schema

Answer: d
Explanation: An externally-defined schema can provide access to tables that are managed on any PostgreSQL, Microsoft SQL Server, SAS, Oracle, or MySQL database.

249. Which of the following are the process of selecting the data storage and data access characteristics of the database?
a) Logical database design
b) Physical database design
c) Testing and performance tuning
d) Evaluation and selecting

Answer: b
Explanation: The physical design of the database optimizes performance while ensuring data integrity by avoiding unnecessary data redundancies.

250. Which of the following terms does refer to the correctness and completeness of the data in a database?
a) Data security
b) Data constraint
c) Data independence
d) Data integrity

Answer: d
Explanation: ACID property is satisfied by transaction in database.

251. The relationship between DEPARTMENT and EMPLOYEE is a
a) One-to-one relationship
b) One-to-many relationship
c) Many-to-many relationship
d) Many-to-one relationship

Answer: b
Explanation: One entity department is related to several employees.

252. A table can be logically connected to another table by defining a
a) Super key
b) Candidate key
c) Primary key
d) Unique key

Answer: c
Explanation: A superkey is a combination of attributes that can be uniquely used to identify a database record.

253. If the state of the database no longer reflects a real state of the world that the database is supposed to capture, then such a state is called
a) Consistent state
b) Parallel state
c) Durable state
d) Inconsistent state

Answer: d
Explanation: SQL data consistency is that whenever a transaction is performed, it sees a consistent database.

254. Ensuring isolation property is the responsibility of the
a) Recovery-management component of the DBMS
b) Concurrency-control component of the DBMS
c) Transaction-management component of the DBMS
d) Buffer management component in DBMS

Answer: b
Explanation: Concurrency control ensures that correct results for concurrent operations are generated while getting those results as quickly as possible.

255. _____ is a procedural extension of Oracle – SQL that offers language constructs similar to those in imperative programming languages.
a) SQL
b) PL/SQL
c) Advanced SQL
d) PQL

Answer: b
Explanation: PL/SQL is an imperative 3GL that was designed specifically for the seamless processing of SQL commands.

256. _____ combines the data manipulating power of SQL with the data processing power of Procedural languages.
a) PL/SQL
b) SQL
c) Advanced SQL
d) PQL

Answer: a
Explanation: PL/SQL is an imperative 3GL that was designed specifically for the seamless processing of SQL commands.

257. _____ has made PL/SQL code run faster without requiring any additional work on the part of the programmer.
a) SQL Server
b) My SQL
c) Oracle
d) SQL Lite

Answer: c
Explanation: An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information.

258. A line of PL/SQL text contains groups of characters known as
a) Lexical Units
b) Literals
c) Textual Units
d) Identifiers

Answer: a
Explanation: Lexical items can be generally understood to convey a single meaning, much as a lexeme, but are not limited to single words.

259. We use _____ name PL/SQL program objects and units.
a) Lexical Units
b) Literals
c) Delimiters
d) Identifiers

Answer: d
Explanation: The database object name is referred to as its identifier.

260. Consider money is transferred from (1)account-A to account-B and (2) account-B to account-A. Which of the following form a transaction?
a) Only 1
b) Only 2
c) Both 1 and 2 individually
d) Either 1 or 2

Answer: c
Explanation: The term transaction refers to a collection of operations that form a single logical unit of work.

261. A transaction is delimited by statements (or function calls) of the form _____
a) Begin transaction and end transaction
b) Start transaction and stop transaction
c) Get transaction and post transaction
d) Read transaction and write transaction

Answer: a
Explanation: The transaction consists of all operations executed between the begin transaction and end transaction.

262. Identify the characteristics of transactions
a) Atomicity
b) Durability
c) Isolation
d) All of the mentioned

Answer: d
Explanation: Because of the above three properties, transactions are an ideal way of structuring interaction with a database.

263. Which of the following has "all-or-none" property?
a) Atomicity
b) Durability
c) Isolation
d) All of the mentioned

Answer: a
Explanation: Either all operations of the transaction are reflected properly in the database, or none are.

264. The database system must take special actions to ensure that transactions operate properly without interference from concurrently executing database statements. This property is referred to as
a) Atomicity
b) Durability
c) Isolation
d) All of the mentioned

Answer: c
Explanation: Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions Ti and Tj, it appears to Ti that either Tj finished execution before Ti started or Tj started execution after Ti finished.

265. The property of a transaction that persists all the crashes is
a) Atomicity
b) Durability
c) Isolation
d) All of the mentioned

Answer: b
Explanation: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

266. _____ states that only valid data will be written to the database.
a) Consistency
b) Atomicity
c) Durability
d) Isolation

Answer: a
Explanation: If for some reason, a transaction is executed that violates the database's consistency rules, the entire transaction will be rolled back and the database will be restored to a state consistent with those rules.

267. Transaction processing is associated with everything below except
a) Producing detail summary or exception reports
b) Recording a business activity
c) Confirming an action or triggering a response
d) Maintaining a data

Answer: c
Explanation: Collections of operations that form a single logical unit of work are called transactions.

268. The Oracle RDBMS uses the ____ statement to declare a new transaction start and its properties.
a) BEGIN
b) SET TRANSACTION
c) BEGIN TRANSACTION
d) COMMIT

Answer: b
Explanation: Commit is used to store all the transactions.

269. ____ means that the data used during the execution of a transaction cannot be used by a second transaction until the first one is completed.
a) Consistency
b) Atomicity
c) Durability
d) Isolation

Answer: d
Explanation: Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions Ti and Tj, it appears to Ti that either Tj finished execution before Ti started or Tj started execution after Ti finished.

270. In SQL, which command is used to issue multiple CREATE TABLE, CREATE VIEW and GRANT statements in a single transaction?
a) CREATE PACKAGE

b) CREATE SCHEMA
c) CREATE CLUSTER
d) All of the mentioned

Answer: b
Explanation: A database schema of a database system is its structure described in a formal language supported by the database management system and refers to the organization of data as a blueprint of how a database is constructed.

271. In SQL, the CREATE TABLESPACE is used
a) To create a place in the database for storage of scheme objects, rollback segments, and naming the data files to comprise the tablespace
b) To create a database trigger
c) To add/rename data files, to change storage
d) All of the mentioned

Answer: a
Explanation: Triggers are used to initialize the actions for an activity.

272. Which character function can be used to return a specified portion of a character string?
a) INSTR
b) SUBSTRING
c) SUBSTR
d) POS

Answer: c
Explanation: SUBSTR are used to match the particular characters in a string.

273. Which of the following is TRUE for the System Variable $date$?
a) Can be assigned to a global variable
b) Can be assigned to any field only during design time
c) Can be assigned to any variable or field during run time
d) Can be assigned to a local variable

Answer: b
Explanation: A database schema of a database system is its structure described in a formal language supported by the database management system and refers to the organization of data as a blueprint of how a database is constructed.

275. What are the different events in Triggers?
a) Define, Create
b) Drop, Comment
c) Insert, Update, Delete
d) Select, Commit

Answer: c
Explanation: A database trigger is a procedural code that is automatically executed in response to certain

events on a particular table or view in a database.

276. Which of the following is not a property of transactions?
a) Atomicity
b) Concurrency
c) Isolation
d) Durability

Answer: d
Explanation: ACID properties are the properties of transactions.

277. SNAPSHOT is used for (DBA)
a) Synonym
b) Tablespace
c) System server
d) Dynamic data replication

Answer: d
Explanation: Snapshot gets the instance of the database at that time.

278. Isolation of the transactions is ensured by
a) Transaction management
b) Application programmer
c) Concurrency control
d) Recovery management

Answer: c
Explanation: ACID properties are the properties of transactions.

279. Constraint checking can be disabled in existing _____ and _____ constraints so that any data you modify or add to the table is not checked against the constraint.
a) CHECK, FOREIGN KEY
b) DELETE, FOREIGN KEY
c) CHECK, PRIMARY KEY
d) PRIMARY KEY, FOREIGN KEY

Answer: a
Explanation: Check and foreign constraints are used to constraint the table data.

280. In order to maintain transactional integrity and database consistency, what technology does a DBMS deploy?
a) Triggers
b) Pointers
c) Locks
d) Cursors

Answer: c
Explanation: Locks are used to maintain database consistency.

281. A lock that allows concurrent transactions to access different rows of the same table is known as a
a) Database-level lock
b) Table-level lock
c) Page-level lock
d) Row-level lock

Answer: d
Explanation: Locks are used to maintain database consistency.

282. Which of the following are introduced to reduce the overheads caused by the log-based recovery?
a) Checkpoints
b) Indices
c) Deadlocks
d) Locks

Answer: a
Explanation: Checkpoints are introduced to reduce overheads caused by the log-based recovery.

283. Which of the following protocols ensures conflict serializability and safety from deadlocks?
a) Two-phase locking protocol
b) Time-stamp ordering protocol
c) Graph based protocol
d) None of the mentioned

Answer: b
Explanation: Time-stamp ordering protocol ensures conflict serializability and safety from deadlocks.

284. A system is in a _____ state if there exists a set of transactions such that every transaction in the set is waiting for another transaction in the set.
a) Idle
b) Waiting
c) Deadlock
d) Ready

Answer: c
Explanation: When one data item is waiting for another data item in a transaction then system is in deadlock.

285. The deadlock state can be changed back to stable state by using _____ statement.
a) Commit
b) Rollback
c) Savepoint
d) Deadlock

Answer: b
Explanation: Rollback is used to rollback to the point before lock is obtained.

286. What are the ways of dealing with deadlock?
a) Deadlock prevention
b) Deadlock recovery
c) Deadlock detection
d) All of the mentioned

Answer: d
Explanation: Deadlock prevention is also called as deadlock recovery. Prevention is commonly used if the probability that the system would enter a deadlock state is relatively high; otherwise, detection and recovery are more efficient.

287. The most recent version of standard SQL prescribed by the

American National Standards Institute is
a) SQL 2016
b) SQL 2002
c) SQL – 4
d) SQL2

Answer: a
Explanation: SQL-2016 is the most recent version of standard SQL prescribed by the ANSI.

288. ANSI-standard SQL allows the use of special operators in conjunction with the WHERE clause. A special operator used to check whether an attribute value is null is
a) BETWEEN
b) IS NULL
c) LIKE
d) IN

Answer: b
Explanation: Exists is used to check whether an attribute value is null or not in conjunction with the where clause.

289. The method of access that uses key transformation is called as
a) Direct
b) Hash
c) Random
d) Sequential

Answer: b
Explanation: Hash technique uses particular hash key value.

290. Why do we need concurrency control on B+ trees ?
a) To remove the unwanted data
b) To easily add the index elements
c) To maintain accuracy of index
d) All of the mentioned

Answer: c
Explanation: Indices do not have to be treated like other database structures.

291. How many techniques are available to control concurrency on B+ trees?
a) One
b) Three
c) Four
d) None of the mentioned

Answer: d
Explanation: Two techniques are present.

292. In crabbing protocol locking
a) Goes down the tree and back up
b) Goes up the tree and back down
c) Goes down the tree and releases
d) Goes up the tree and releases

Answer: a
Explanation: It moves in a crab like manner.

293. The deadlock can be handled by
a) Removing the nodes that are deadlocked
b) Restarting the search after releasing the lock
c) Restarting the search without releasing the lock
d) Resuming the search

Answer: b
Explanation: Crabbing protocol moves in a crab like manner.

294. The recovery scheme must also provide
a) High availability
b) Low availability
c) High reliability
d) High durability

Answer: a
Explanation: It must minimize the time for which the database is not usable after a failure.

295. Which one of the following is a failure to a system
a) Boot crash
b) Read failure
c) Transaction failure
d) All of the mentioned

Answer: c
Explanation: Types of system failure are transaction failure, system crash and disk failure.

296. Which of the following belongs to transaction failure
a) Read error
b) Boot error
c) Logical error
d) All of the mentioned

Answer: c
Explanation: Types of system transaction failure are logical and system error.

297. The system has entered an undesirable state (for example, deadlock), as a result of which a transaction cannot continue with its normal execution. This is
a) Read error
b) Boot error
c) Logical error
d) System error

Answer: c
Explanation: The transaction, can be re-executed at a later time.

298. The transaction can no longer continue with its normal execution because of some internal condition, such as bad input, data not found, overflow, or resource limit exceeded. This is
a) Read error
b) Boot error
c) Logical error
d) System error

Answer: c
Explanation: The transaction, can be re-executed at a later time.

299. The assumption that hardware errors and bugs in the software bring the system to a halt, but do not corrupt the nonvolatile storage contents, is known as the
a) Stop assumption
b) Fail assumption
c) Halt assumption
d) Fail-stop assumption

Answer: d
Explanation: Well-designed systems have numerous internal checks, at the hardware and the software level, that bring the system to a halt when there is an error. Hence, the fail-stop assumption is a reasonable one.

300. Which kind of failure loses its data in head crash or failure during a transfer operation.
a) Transaction failure
b) System crash
c) Disk failure
d) All of the mentioned

Answer: c
Explanation: Copies of the data on other disks, or archival backups on tertiary media, such as DVD or tapes, are used to recover from the failure.

301. The log is a sequence of _____ recording all the update activities in the database.
a) Log records
b) Records
c) Entries
d) Redo

Answer: a
Explanation: The most widely used structure for recording database modifications is the log.

302. In the _____ scheme, a transaction that wants to update the database first creates a complete copy of the database.
a) Shadow copy
b) Shadow Paging
c) Update log records
d) All of the mentioned

Answer: a
Explanation: If at any point the transaction has to be aborted, the system merely deletes the new copy. The old copy of the database has not been affected.

303. The _____ scheme uses a page table containing pointers to all pages; the page table itself and all updated pages are copied to a new location.

DIWAKAR EDUCATION HUB

a) Shadow copy
b) Shadow Paging
c) Update log records
d) All of the mentioned

Answer: b
Explanation: Any page which is not updated by a transaction is not copied, but instead the new page table just stores a pointer to the original page.

304. The current copy of the database is identified by a pointer, called _____ which is stored on disk.
a) Db-pointer
b) Update log
c) Update log records
d) All of the mentioned

Answer: a
Explanation: Any page which is not updated by a transaction is not copied, but instead the new page table just stores a pointer to the original page.

305. If a transaction does not modify the database until it has committed, it is said to use the _____ technique.
a) Deferred-modification
b) Late-modification
c) Immediate-modification
d) Undo

Answer: a
Explanation: Deferred modification has the overhead that transactions need to make local copies of all updated data items; further, if a transaction reads a data item that it has updated, it must read the value from its local copy.

306. If database modifications occur while the transaction is still active, the transaction is said to use the _____technique.
a) Deferred-modification
b) Late-modification
c) Immediate-modification
d) Undo

Answer: c
Explanation: We say a transaction modifies the database if it performs an update on a disk buffer, or on the disk itself; updates to the private part of main memory do not count as database modifications.

307. _____ using a log record sets the data item specified in the log record to the old value.
a) Deferred-modification
b) Late-modification

DIWAKAR EDUCATION HUB

c) Immediate-modification
d) Undo

Answer: d
Explanation: Undo brings the previous contents.

308. In the _____ phase, the system replays updates of all transactions by scanning the log forward from the last checkpoint.
a) Repeating
b) Redo
c) Replay
d) Undo

Answer: b
Explanation: Undo brings the previous contents.

309. In order to reduce the overhead in retrieving the records from the storage space we use
a) Logs
b) Log buffer
c) Medieval space
d) Lower records

Answer: b
Explanation: The output to stable storage is in units of blocks.

310. The order of log records in the stable storage _____ as the order in which they were written to the log buffer.
a) Must be exactly the same
b) Can be different
c) Is opposite
d) Can be partially same

Answer: a
Explanation: As a result of log buffering, a log record may reside in only main memory (volatile storage) for a considerable time before it is output to stable storage.

311. Before a block of data in main memory can be output to the database, all log records pertaining to data in that block must have been output to stable storage. This is
a) Read-write logging
b) Read-ahead logging
c) Write-ahead logging
d) None of the mentioned

Answer: c
Explanation: The WAL rule requires only that the undo information in the log has been output to stable storage, and it permits the redo information to be written later.

312. Writing the buffered log to _____ is sometimes referred to as a log force.
a) Memory

DIWAKAR EDUCATION HUB

b) Backup
c) Redo memory
d) Disk

Answer: d
Explanation: If there are insufficient log records to fill the block, all log records in main memory are combined into a partially full block and are output to stable storage.

313. The silicon chips used for data processing are called
a) RAM chips
b) ROM chips
c) Micro processors
d) PROM chips

Answer: d
Explanation: PROM is Programmable Read Only Memory.

314. Which of the following is used for manufacturing chips?
a) Control bus
b) Control unit
c) Parity unit
d) Semiconductor

Answer: d
Explanation: A semiconductor is a material which has electrical conductivity between that of a conductor such as copper and that of an insulator such as glass.

315. What was the name of the first commercially available microprocessor chip?
a) Intel 308
b) Intel 33
c) Intel 4004
d) Motorola 639

Answer: c
Explanation: The Intel 4004 is a 4-bit central processing unit (CPU) released by Intel Corporation in 1971

316. Which lock should be obtained to prevent a concurrent transaction from executing a conflicting read, insert or delete operation on the same key value.
a) Higher-level lock
b) Lower-level lock
c) Read only lock
d) Read write

Answer: a
Explanation: Operations acquire lower-level locks while they execute, but release them when they complete; the corresponding transaction must however retain a higher-level lock in a two-phase manner to prevent concurrent

DIWAKAR EDUCATION HUB

transactions from executing conflicting actions.

317. Once the lower-level lock is released, the operation cannot be undone by using the old values of updated data items, and must instead be undone by executing a compensating operation; such an operation is called
a) Logical operation
b) Redo operation
c) Logical undo operation
d) Undo operation

Answer: a
Explanation: It is important that the lower-level locks acquired during an operation are sufficient to perform a subsequent logical undo of the operation.

318. The remote backup site is sometimes also called the
a) Primary Site
b) Secondary Site
c) Tertiary Site
d) None of the mentioned

Answer: b
Explanation: We can achieve high availability by performing transaction processing at one site, called the primary site, and having a remote backup site where all the data from the primary site are replicated.

319. Remote backup system must be _____ with the primary site.
a) Synchronised
b) Separated
c) Connected
d) Detached but related

Answer: a
Explanation: We can achieve high availability by performing transaction processing at one site, called the primary site, and having a remote backup site where all the data from the primary site are replicated.

320. The backup is taken by
a) Erasing all previous records
b) Entering the new records
c) Sending all log records from primary site to the remote backup site
d) Sending selected records from primary site to the remote backup site

Answer: c
Explanation: We can achieve high availability by performing transaction processing at one site, called the primary site, and having a remote

DIWAKAR EDUCATION HUB

backup site where all the data from the primary site are replicated.

321. When the _____ the backup site takes over processing and becomes the primary.
a) Secondary fails
b) Backup recovers
c) Primary fails
d) None of the mentioned

Answer: c
Explanation: When the original primary site recovers, it can either play the role of remote backup, or take over the role of primary site again.

322. The simplest way of transferring control is for the old primary to receive _____ from the old backup site.
a) Undo logs
b) Redo Logs
c) Primary Logs
d) All of the mentioned

Answer: c
Explanation: If control must be transferred back, the old backup site can pretend to have failed, resulting in the old primary taking over.

323. In the _____ phase, the system replays updates of all transactions by scanning the log forward from the last checkpoint.
a) Repeating
b) Redo
c) Replay
d) Undo

Answer: b
Explanation: Undo brings the previous contents.

324. The actions which are played in the order while recording it is called _____ history.
a) Repeating
b) Redo
c) Replay
d) Undo

Answer: a
Explanation: Undo brings the previous contents.

325. A special redo-only log record < Ti, Xj, V1> is written to the log, where V1 is the value being restored to data item Xj during the rollback. These log records are sometimes called
a) Log records
b) Records
c) Compensation log records
d) Compensation redo records

DIWAKAR EDUCATION HUB

Answer: c
Explanation: Such records do not need undo information since we never need to undo such an undo operation.

326. The process of designating sub groupings within the entity set is called as _____
a) Specialization
b) Division
c) Aggregation
d) Finalization

Answer: a
Explanation: The process of designating sub-groupings within the entity set is called as specialization. Specialization allows us to distinguish among entities.

327. State true or false: Specialization can be applied only once
a) True
b) False

Answer: a
Explanation: We can apply specialization multiple times to refine a design. An entity set may also be specialized by more than one distinguishing feature.

328. Which of the following is the specialization that permits multiple sets
a) Superclass specialization
b) Disjoint specialization
c) Overlapping specialization
d) None of the mentioned

Answer: c
Explanation: Overlapping specialization is the type of specialization that permits multiple sets. But disjoint specialization does not permit multiple sets. Disjoint specialization permits at most one set.

329. The similarities between the entity set can be expressed by which of the following features?
a) Specialization
b) Generalization
c) Uniquation
d) Inheritance

Answer: b
Explanation: The similarities between the entity set can be expressed by the generalization feature. It is a containment o the relationship that exists between a higher level entity set and one or more lower level entity sets.

DIWAKAR EDUCATION HUB

330. Higher level entity sets are designated by the term _____
a) Sub class
b) Super class
c) Parent class
d) Root class

Answer: b
Explanation: Higher level entity sets can also be designated by the term super class. In the similar manner lower level entity sets can also be designated by the term sub class.

331. State true or false: The attributes of the higher level entity sets are inherited by the attributes of the lower level entity sets
a) True
b) False

Answer: a
Explanation: The attributes of the higher level entity sets are inherited by the attributes of the lower level entity sets. But the inverse is not true in this case.

332. Which of the following is not a generalization constraint?
a) Condition-defined
b) User defined
c) Disjoint
d) Machine defined

Answer: d
Explanation: Machine definition is not a generalization constraint. Condition defined, user defined, disjoint and overlapping are 4 generalization constraints.

333. Condition defined generalization constraint is also said to be _____
a) Attribute defined
b) Constraint defined
c) Value defined
d) Undefined

Answer: a
Explanation: Condition defined generalization constraint is also said to be attribute defined.

334. Which of the following is not a feature of a good relational design?
a) Specifying primary keys
b) Specifying foreign keys
c) Preserving integrity constraints
d) Allowing redundancy of attributes

Answer: d
Explanation: Allowing redundancy of attributes makes it very difficult for data extraction. So, It is not a good relational design feature.

335. The dependency rules specified by the database designer are known

DIWAKAR EDUCATION HUB

as _____
a) Designer dependencies
b) Database rules
c) Functional dependencies
d) None of the mentioned

Answer: c
Explanation: The dependency rules specified by the database designer are known as functional dependencies. The normal forms are based on functional dependencies.

336. If the decomposition is unable to represent certain important facts about the relation, then such a decomposition is called as?
a) Lossless decomposition
b) Lossy decomposition
c) Insecure decomposition
d) Secure decomposition

Answer: b
Explanation: If the decomposition is unable to represent certain important facts about the relation, then such a decomposition is called as lossy decomposition. Lossy decompositions should be avoided as they result in the loss of data.

337. An instance of a relation that satisfies all real world constraints is known as?
a) Proper relation
b) Ideal relation
c) Perfect relation
d) Legal relation

Answer: d
Explanation: A relation that satisfies all the real world constraints is called as a legal relation. An instance of a legal relation is called as a legal instance.

338. If K → R then K is said to be the _____ of R
a) Candidate key
b) Foreign key
c) Super key
d) Domain

Answer: c
Explanation: If K → R then k is said to be the superkey of R i.e. K uniquely identifies every tuple in the relation R.

339. X → Y holds on a schema k(K) if?
a) At least one legal instance satisfies the functional dependency
b) No legal instance satisfies the functional dependency
c) Each and every legal instance satisfies the functional dependency
d) None of the mentioned

DIWAKAR EDUCATION HUB

Answer: c
Explanation: X → Y holds on a schema k(K) if each and every legal instance satisfies the functional dependency. Even if one instance does not satisfy the functional dependency X→ Y does not hold on a schema.

340. X→ Y is trivial if?
a) X ⊂ Y
b) Y ⊂ X
c) X ⊇ Y
d) None of the mentioned

Answer: a
Explanation: X→ Y is said to be trivial if X is a subset of Y. Thus X ⊂ Y implies X→Y is trivial.

341. Which of the following is not a condition for X→ Y in Boyce codd normal form?
a) X → Y is trivial
b) X is the superkey for the relational schema R
c) Y is the superkey for the relational schema R
d) All of the mentioned

Answer: c
Explanation: Y does not need to be a superkey of the relation for the given functional dependency to satisfy BCNF. X→ Y must be trivial and X must be the superkey of the relation R.

342. Which of the following is used to express database consistency?
a) Primary keys
b) Functional dependencies
c) Check clause
d) All of the mentioned

Answer: d
Explanation: Primary keys, Functional dependencies, Check clause are all used to express database consistency.

343. _____ introduces the Management Data Warehouse (MDW) to SQL Server Management Studio for streamlined performance troubleshooting.
a) SQL Server 2005
b) SQL Server 2008
c) SQL Server 2012
d) SQL Server 2014

Answer: b
Explanation: MDW is a set of components that enable a database developer or administrator to quickly track down problems that could be causing performance degradation.

DIWAKAR EDUCATION HUB

344. Point out the correct statement.
a) MDW consist of three components
b) SQL Server Express instances can be targets
c) Setting up the MDW is a one-step process
d) All of the mentioned

Answer: a
Explanation: MDW consists of three components: Data Collector, MDW database and MDW reports.

345. Which of the following mode allows for the collection and uploading of data to occur on demand?
a) Non-cached mode
b) Cached mode
c) Mixed mode
d) All of the mentioned

Answer: a
Explanation: In non-cached mode, collection and upload are on the same schedule.

346. Which of the following scenario favours cached mode?
a) Continuous collection of data
b) Less frequent uploads
c) Data collection and uploading of jobs on different schedules
d) All of the mentioned

Answer: d
Explanation: Cached mode uses separate schedules for collection and upload.

347. Point out the wrong statement.
a) The Data Collection is performed primarily through SSIS packages that control the collection frequency on the target
b) You should change the database name after creation
c) Do not change any of the job specifications for the data collection and upload jobs
d) None of the mentioned

Answer: b
Explanation: You should not change the database name after creation, because all of the jobs created to manage the database collection refer to the database by the original name and will generate errors if the name is changed.

348. Which of the following is the best Practice and Caveat for Management Data Warehouse?
a) Use a centralized server for the MDW database

DIWAKAR EDUCATION HUB

b) The XML parameters for a single T-SQL collection item can have multiple <Query> elements
c) Use a distributed server for the MDW database
d) All of the mentioned

Answer: a
Explanation: Centralized server allows you to use a single point for viewing reports for multiple instances.

349. _____ stores information about how the management data warehouse reports should group and aggregate performance counters.
a) core.snapshots_internal
b) core.supported_collector_types_internal
c) core.wait_categories
d) core.performance_counter_report_group_items

Answer: d
Explanation: core.wait_categories contains the categories used to group wait types according to wait_type characteristic.

350. Which of the following table is used in the management data warehouse schema that is required for the Server Activity?
a) snapshots.query_stat
b) snapshots.os_latch_stats
c) snapshots.active_sessions
d) all of the mentioned

Answer: b
Explanation: snapshots.os_latch_stats is a System level resource table.

351. Which of the following is syntax for sp_add_collector_type procedure?
a) core.sp_add_collector [@collector_type_uid =] ‘collector_type_uid’
b) core.sp_add_collector_type [@collector_type_uid =].
c) core.sp_add_collector_type [@collector_type_uid =] ‘collector_type_uid’
d) none of the mentioned

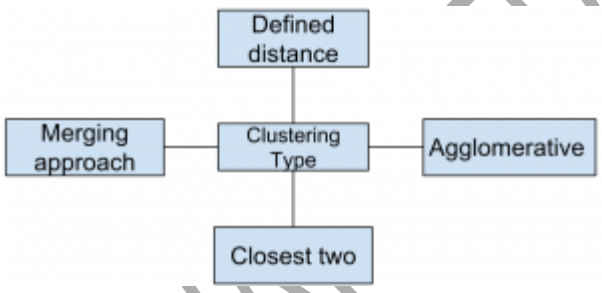
Answer: c
Explanation: core.sp_add_collector_type adds a new entry to the core.supported_collector_types view in the management data warehouse database.

DIWAKAR EDUCATION HUB

352. What does collector_type_id stands for in the following code snippet?
core.sp_remove_collector_type [@collector_type_uid =] 'collector_type_uid'
a) uniqueidentifier
b) membership role
c) directory
d) none of the mentioned

Answer: a
Explanation: collector_type_uid is the GUID for the collector type.

353. Which of the following clustering type has characteristic shown in the below figure?



a) Partitional
b) Hierarchical
c) Naive bayes
d) None of the mentioned

Answer: b
Explanation: Hierarchical clustering groups data over a variety of scales by creating a cluster tree or dendrogram.

354. Point out the correct statement.
a) The choice of an appropriate metric will influence the shape of the clusters
b) Hierarchical clustering is also called HCA
c) In general, the merges and splits are determined in a greedy manner
d) All of the mentioned

Answer: d
Explanation: Some elements may be close to one another according to one distance and farther away according to another.

355. Which of the following is finally produced by Hierarchical Clustering?
a) final estimate of cluster centroids
b) tree showing how close things are to each other
c) assignment of each point to clusters
d) all of the mentioned

Answer: b
Explanation: Hierarchical clustering is an agglomerative approach.

356. Which of the following is required by K-means clustering?

DIWAKAR EDUCATION HUB

a) defined distance metric
b) number of clusters
c) initial guess as to cluster centroids
d) all of the mentioned

Answer: d
Explanation: K-means clustering follows partitioning approach.

357. Point out the wrong statement.
a) k-means clustering is a method of vector quantization
b) k-means clustering aims to partition n observations into k clusters
c) k-nearest neighbor is same as k-means
d) none of the mentioned

Answer: c
Explanation: k-nearest neighbor has nothing to do with k-means.

358. Which of the following combination is incorrect?
a) Continuous – euclidean distance
b) Continuous – correlation similarity
c) Binary – manhattan distance
d) None of the mentioned

Answer: d
Explanation: You should choose a distance/similarity that makes sense for your problem.

359. Hierarchical clustering should be primarily used for exploration.
a) True
b) False

Answer: a
Explanation: Hierarchical clustering is deterministic.

360. Which of the following function is used for k-means clustering?
a) k-means
b) k-mean
c) heatmap
d) none of the mentioned

Answer: a
Explanation: K-means requires a number of clusters.

361. Which of the following clustering requires merging approach?
a) Partitional
b) Hierarchical
c) Naive Bayes
d) None of the mentioned

Answer: b
Explanation: Hierarchical clustering requires a defined distance as well.

362. K-means is not deterministic and it also consists of number of iterations.

DIWAKAR EDUCATION HUB

a) True
b) False

Answer: a
Explanation: K-means clustering produces the final estimate of cluster centroids.

363. Which of the following term is appropriate to the below figure?



a) Large Data
b) Big Data
c) Dark Data
d) None of the mentioned

Answer: b
Explanation: Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate.

364. Point out the correct statement.
a) Machine learning focuses on prediction, based on known properties learned from the training data
b) Data Cleaning focuses on prediction, based on known properties learned from the training data
c) Representing data in a form which both mere mortals can understand and get valuable insights is as much a science as much as it is art
d) None of the mentioned

Answer: d
Explanation: Visualization is becoming a very important aspect.

365. Which of the following characteristic of big data is relatively more concerned to data science?
a) Velocity
b) Variety
c) Volume
d) None of the mentioned

Answer: b
Explanation: Big data enables organizations to store, manage, and manipulate vast amounts of disparate data at the right speed and at the right time.

366. Which of the following analytical capabilities are provided by information management company?
a) Stream Computing
b) Content Management
c) Information Integration
d) All of the mentioned

DIWAKAR EDUCATION HUB

Answer: d
Explanation: With stream computing, store less, analyze more and make better decisions faster.

367. Point out the wrong statement.
a) The big volume indeed represents Big Data
b) The data growth and social media explosion have changed how we look at the data
c) Big Data is just about lots of data
d) All of the mentioned

Answer: c
Explanation: Big Data is actually a concept providing an opportunity to find new insight into your existing data as well guidelines to capture and analysis your future data.

368. Which of the following step is performed by data scientist after acquiring the data?
a) Data Cleansing
b) Data Integration
c) Data Replication
d) All of the mentioned

Answer: a
Explanation: Data cleansing, data cleaning or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database.

369. 3V's are not sufficient to describe big data.
a) True
b) False

Answer: a
Explanation: IBM data scientists break big data into four dimensions: volume, variety, velocity and veracity.

370: Which of the following applied on warehouse?

a) write only

b) read only

c) both a & b

d) none of these

Answer:B

371: Data can be store , retrive and updated in …

a) SMTOP

b) OLTP

c) FTP

d) OLAP

Answer:B

DIWAKAR EDUCATION HUB

372: Which of the following is a good alternative to the star schema?

a) snow flake schema

b) star schema

c) star snow flake schema

d) fact constellation

Answer :D

373: Patterns that can be discovered from a given database are which type…

a) More than one type

b) Multiple type always

c) One type only

d) No specific type

Answer :A

374:Background knowledge is…

a) It is a form of automatic learning.

b) A neural network that makes use of a hidden layer

c) The additional acquaintance used by a learning algorithm to facilitate the learning process

d) None of these

Answer:C

375: Which of the following is true for Classification?

a) A subdivision of a set

b) A measure of the accuracy

c) The task of assigning a classification

d) All of these

Answer:A

376: Data mining is?

a) time variant non-volatile collection of data

b) The actual discovery phase of a knowledge

c) The stage of selecting the right data

d) None of these

Answer -:B

377: ——- is not a data mining functionality?

A) Clustering and Analysis

B) Selection and interpretation

C) Classification and regression

DIWAKAR EDUCATION HUB

D) Characterization and Discrimination

Answer -:B

378: Which of the following can also applied to other forms?

a) Data streams & Sequence data

b) Networked data

c) Text & Spatial data

d) All of these

Answer -:D

379:Which of the following is general characteristics or features of a target class of data?

a) Data selection

b) Data discrimination

c) Data Classification

c) Data Characterization

Answer -:D

380: ——— is the out put of KDD…

a) Query

b) Useful Information

c) Data

d) information

Answer -:B

381: What is noise?

a) component of a network

b) context of KDD and data mining

c) aspects of a data warehouse

d) None of these

Answer -:B

382.Firms that are engaged in sentiment mining are analyzing data collected from?

A. social media sites.
B. in-depth interviews.
C. focus groups.
D. experiments.
E. observations.

Answer -:A. social media sites.

Which of the following forms of data mining assigns records to one of a predefined set of classes?

(A). Classification

(B). Clustering

(C). Both A and B

(D). None

DIWAKAR EDUCATION HUB

Answer -:(B). Clustering

383. What is the adaptive system management?

a) machine language techniques

b) machine learning techniques

c) machine procedures techniques

d) none of these

Answer -:B

384. An essential process used for applying intelligent methods to extract the data patterns is named as …

a) data mining

b) data analysis

c) data implementation

d) data computation

Answer -:A

385. Classification and regression are the properties of…

a) data analysis

b) data manipulation'

c) data mining

d) none of these

Answer -:C

386. A class of learning algorithm that tries to find an optimum classification of a set of examples using the probabilistic theory is named as …

a) Bayesian classifiers

b) Dijkstra classifiers

c) doppler classifiers

d) all of these

Answer -:A

387. Which of the following can be used for finding deep knowledge?

a) stacks

b) algorithms

c) clues

d) none of these

Answer -:C

388. We define a _____ as a subdivison of a set of examples into a number of classes.

a) kingdom

b) tree

c) classification

DIWAKAR EDUCATION HUB

d) array

Answer -:C

389. Group of similar objects that differ significantly from other objects is named as …

a) classification

b) cluster

c) community

d) none of these

Answer -:B

390. Combining different type of methods or information is ….

a) analysis

b) computation

c) stack

d) hybrid

Answer -:D

391. What i sthe name of database having a set of databases from different vendors, possibly using different database paradigms?

a) homogeneous database

b) heterogeneous database

c) hybrid database

d) none of these

Answer -:B

392. What is the strategic value of data mining?

a) design sensitive

b) cost sensitive

c) technical sensitive

d) time sensitive

Answer -:D

393. The amount of information with in data as opposed to the amount of redundancy or noise is known as …

a) paragraph content

b) text content

c) information content

d) none of these

Answer -:C

394. What is inductive learning?

a) learning by hypothesis

b) learning by analyzing

c) learning by generalizing

DIWAKAR EDUCATION HUB

d) none of these

Answer -:C

395: Which of the following applied on warehouse?

a) write only

b) read only

c) both a & b

d) none of these

Answer -:B

396: Data can be store , retrive and updated in …

a) SMTOP

b) OLTP

c) FTP

d) OLAP

Answer -:B

397: Which of the following is a good alternative to the star schema?

a) snow flake schema

b) star schema

c) star snow flake schema

d) fact constellation

Answer -:D

398: Patterns that can be discovered from a given database are which type…

a) More than one type

b) Multiple type always

c) One type only

d) No specific type

Answer -:A

399:Background knowledge is…

a) It is a form of automatic learning.

b) A neural network that makes use of a hidden layer

c) The additional acquaintance used by a learning algorithm to facilitate the learning process

d) None of these

Answer -:C

400: Which of the following is true for Classification?

a) A subdivision of a set

b) A measure of the accuracy

DIWAKAR EDUCATION HUB

c) The task of assigning a classification

d) All of these

Answer -:A

401. As companies move past the experimental phase with Hadoop, many cite the need for additional capabilities, including

_____

a) Improved data storage and information retrieval
b) Improved extract, transform and load features for data integration
c) Improved data warehousing functionality
d) Improved security, workload management, and SQL support

Answer: d
Explanation: Adding security to Hadoop is challenging because all the interactions do not follow the classic client-server pattern.

402. Point out the correct statement.
a) Hadoop do need specialized hardware to process the data
b) Hadoop 2.0 allows live stream processing of real-time data
c) In Hadoop programming framework output files are divided into lines or records
d) None of the mentioned

Answer: b
Explanation: Hadoop batch processes data distributed over a number of computers ranging in 100s and 1000s.

403. According to analysts, for what can traditional IT systems provide a foundation when they're integrated with big data technologies like Hadoop?
a) Big data management and data mining
b) Data warehousing and business intelligence
c) Management of Hadoop clusters
d) Collecting and storing unstructured data

Answer: a
Explanation: Data warehousing integrated with Hadoop would give a better understanding of data.

404. Hadoop is a framework that works with a variety of related tools. Common cohorts include

_____

a) MapReduce, Hive and HBase
b) MapReduce, MySQL and Google Apps

DIWAKAR EDUCATION HUB

c) MapReduce, Hummer and Iguana
d) MapReduce, Heron and Trumpet

Answer: a
Explanation: To use Hive with HBase you'll typically want to launch two clusters, one to run HBase and the other to run Hive.

405. Point out the wrong statement.
a) Hardtop processing capabilities are huge and its real advantage lies in the ability to process terabytes & petabytes of data
b) Hadoop uses a programming model called "MapReduce", all the programs should confirm to this model in order to work on Hadoop platform
c) The programming model, MapReduce, used by Hadoop is difficult to write and test
d) All of the mentioned

Answer: c
Explanation: The programming model, MapReduce, used by Hadoop is simple to write and test.

406. What was Hadoop named after?
a) Creator Doug Cutting's favorite circus act
b) Cutting's high school rock band
c) The toy elephant of Cutting's son
d) A sound Cutting's laptop made during Hadoop development

Answer: c
Explanation: Doug Cutting, Hadoop creator, named the framework after his child's stuffed toy elephant.

407. All of the following accurately describe Hadoop, EXCEPT _____
a) Open-source
b) Real-time
c) Java-based
d) Distributed computing approach

Answer: b
Explanation: Apache Hadoop is an open-source software framework for distributed storage and distributed processing of Big Data on clusters of commodity hardware.

408. _____ can best be described as a programming model used to develop Hadoop-based applications that can process massive amounts of data.
a) MapReduce
b) Mahout
c) Oozie
d) All of the mentioned

DIWAKAR EDUCATION HUB

Answer: a
Explanation: MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm.

409. _____ has the world's largest Hadoop cluster.
a) Apple
b) Datamatics
c) Facebook
d) None of the mentioned

Answer: c
Explanation: Facebook has many Hadoop clusters, the largest among them is the one that is used for Data warehousing.

410. Facebook Tackles Big Data With _____ based on Hadoop.
a) 'Project Prism'
b) 'Prism'
c) 'Project Big'
d) 'Project Data'

Answer: a
Explanation: Prism automatically replicates and moves data wherever it's needed across a vast network of computing facilities.

411. IBM and _____ have announced a major initiative to use Hadoop to support university courses in distributed computer programming.
a) Google Latitude
b) Android (operating system)
c) Google Variations
d) Google

Answer: d
Explanation: Google and IBM Announce University Initiative to Address Internet-Scale.

412. Point out the correct statement.
a) Hadoop is an ideal environment for extracting and transforming small volumes of data
b) Hadoop stores data in HDFS and supports data compression/decompression
c) The Giraph framework is less useful than a MapReduce job to solve graph and machine learning
d) None of the mentioned

Answer: b
Explanation: Data compression can be achieved using compression algorithms like bzip2, gzip, LZO, etc. Different algorithms can be used in

DIWAKAR EDUCATION HUB

different scenarios based on their capabilities.

413. What license is Hadoop distributed under?
a) Apache License 2.0
b) Mozilla Public License
c) Shareware
d) Commercial

Answer: a
Explanation: Hadoop is Open Source, released under Apache 2 license.

414. Sun also has the Hadoop Live CD _____ project, which allows running a fully functional Hadoop cluster using a live CD.
a) OpenOffice.org
b) OpenSolaris
c) GNU
d) Linux

Answer: b
Explanation: The OpenSolaris Hadoop LiveCD project built a bootable CD-ROM image.

415. Which of the following genres does Hadoop produce?
a) Distributed file system
b) JAX-RS
c) Java Message Service
d) Relational Database Management System

Answer: a
Explanation: The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to the user.

416. What was Hadoop written in?
a) Java (software platform)
b) Perl
c) Java (programming language)
d) Lua (programming language)

Answer: c
Explanation: The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command-line utilities written as shell-scripts.

417. A _____ serves as the master and there is only one NameNode per cluster.
a) Data Node
b) NameNode
c) Data block
d) Replication

Answer: b
Explanation: All the metadata related to HDFS including the information

DIWAKAR EDUCATION HUB

about data nodes, files stored on HDFS, and Replication, etc. are stored and maintained on the NameNode.

418. Point out the correct statement.
a) DataNode is the slave/worker node and holds the user data in the form of Data Blocks
b) Each incoming file is broken into 32 MB by default
c) Data blocks are replicated across different nodes in the cluster to ensure a low degree of fault tolerance
d) None of the mentioned

Answer: a
Explanation: There can be any number of DataNodes in a Hadoop Cluster.

419. HDFS works in a _____ fashion.
a) master-worker
b) master-slave
c) worker/slave
d) all of the mentioned

Answer: a
Explanation: NameNode servers as the master and each DataNode servers as a worker/slave

420. _____ NameNode is used when the Primary NameNode goes down.
a) Rack
b) Data
c) Secondary
d) None of the mentioned

Answer: c
Explanation: Secondary namenode is used for all time availability and reliability.

421. Point out the wrong statement.
a) Replication Factor can be configured at a cluster level (Default is set to 3) and also at a file level
b) Block Report from each DataNode contains a list of all the blocks that are stored on that DataNode
c) User data is stored on the local file system of DataNodes
d) DataNode is aware of the files to which the blocks stored on it belong to

Answer: d
Explanation: NameNode is aware of the files to which the blocks stored on it belong to.

422. Which of the following scenario may not be a good fit for HDFS?

DIWAKAR EDUCATION HUB

a) HDFS is not suitable for scenarios requiring multiple/simultaneous writes to the same file
b) HDFS is suitable for storing data related to applications requiring low latency data access
c) HDFS is suitable for storing data related to applications requiring low latency data access
d) None of the mentioned

Answer: a
Explanation: HDFS can be used for storing archive data since it is cheaper as HDFS allows storing the data on low cost commodity hardware while ensuring a high degree of fault-tolerance.

423. The need for data replication can arise in various scenarios like _____
a) Replication Factor is changed
b) DataNode goes down
c) Data Blocks get corrupted
d) All of the mentioned

Answer: d
Explanation: Data is replicated across different DataNodes to ensure a high degree of fault-tolerance.

424. _____ is the slave/worker node and holds the user data in the form of Data Blocks.
a) DataNode
b) NameNode
c) Data block
d) Replication

Answer: a
Explanation: A DataNode stores data in the [HadoopFileSystem]. A functional filesystem has more than one DataNode, with data replicated across them.

425. HDFS provides a command line interface called _____ used to interact with HDFS.
a) "HDFS Shell"
b) "FS Shell"
c) "DFS Shell"
d) None of the mentioned

Answer: b
Explanation: The File System (FS) shell includes various shell-like commands that directly interact with the Hadoop Distributed File System (HDFS).

426. HDFS is implemented in _____ programming language.
a) C++
b) Java

DIWAKAR EDUCATION HUB

c) Scala
d) None of the mentioned

Answer: b
Explanation: HDFS is implemented in Java and any computer which can run Java can host a NameNode/DataNode on it.

427. For YARN, the _____ Manager UI provides host and port information.
a) Data Node
b) NameNode
c) Resource
d) Replication

Answer: c
Explanation: All the metadata related to HDFS including the information about data nodes, files stored on HDFS, and Replication, etc. are stored and maintained on the NameNode.

428. Which of the following is not a NoSQL database?
a) SQL Server
b) MongoDB
c) Cassandra
d) None of the mentioned

Answer: a
Explanation: Microsoft SQL Server is a relational database management system developed by Microsoft.

429. Point out the correct statement.
a) Documents can contain many different key-value pairs, or key-array pairs, or even nested documents
b) MongoDB has official drivers for a variety of popular programming languages and development environments
c) When compared to relational databases, NoSQL databases are more scalable and provide superior performance
d) All of the mentioned

Answer: d
Explanation: There are also a large number of unofficial or community-supported drivers for other programming languages and frameworks.

430. Which of the following is a NoSQL Database Type?
a) SQL
b) Document databases
c) JSON
d) All of the mentioned

Answer: b
Explanation: Document databases

DIWAKAR EDUCATION HUB

pair each key with a complex data structure known as a document.

431. Which of the following is a wide-column store?
a) Cassandra
b) Riak
c) MongoDB
d) Redis

Answer: a
Explanation: Wide-column stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.

432. Point out the wrong statement.
a) Non Relational databases require that schemas be defined before you can add data
b) NoSQL databases are built to allow the insertion of data without a predefined schema
c) NewSQL databases are built to allow the insertion of data without a predefined schema
d) All of the mentioned

Answer: a
Explanation: There's also no way, using a relational database, to effectively address data that's completely unstructured or unknown in advance.

DIWAKAR EDUCATION HUB