UNIT I – INTRODUCTION TO DBMS

Introduction to DBMS

- Data can be defined as raw fact that could be recorded.

  (Text, number, video, speech, image, audio)
- Database is a organized collection of related data which has an implicit meaning.
- Information: Arranged form of data is called information.
- A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database.
- Some of databses are MySQL, SQL, Server, Oracle, dBASE, FoxPro, MS-Access etc.

Importance of DBMS

- It manages data efficiently and allows user to manage data easily.
- It stores, organizes and manages a large amount of information within a single software application.
- It provide a highly efficient method for handling multiple types of data.
- It is tedious and time consuming to handle data manually.
- Multiple users can use the system at the same time in different ways.

Advantages of DBMS

- **Redundancy control:** Redundancy refers to repeated instances of the same data.
  - A student may have his phone number and address stored in multiple places like in accounts, administration, library
  - It causes wastage of disk space, anomalies, and inconsistency
  - Database system used normalization and keys to control redundancy
- **Inconsistency Control:** values of same data may no longer agree
  - The address of student may be different in different files
  - Normalization and constraints help to control inconsistency

Advantages of DBMS

- **Flexibility: database schema can be easily changed to incorporate any new changes in requirements.**
- **Backup and recovery: DBMS provides facility of backing up the data and recovering from hardware or software failures**
- **Integrity constraints: database system satisfies integrity constraints like primary keys, data type, etc. to ensure accuracy consistency and reliability of databases.**
- **Concurrent/multiple users: multiple users can access the system at the same time**
    - **Like in QFX, air ticket reservation system**

Disadvantages of DBMS

- Cost of Hardware and Software of a DBMS is quite high which increases the budget of your organization.
- Most database management systems are often complex systems, so special training may need for user.
- In some organizations, all data is integrated into a single database which can be damaged because of electric failure or database is corrupted on the storage media.
- Use of the same program at a time by many users sometimes lead to the loss of some data.
- DBMS can't perform sophisticated calculations

Database Applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions
- Database today are core component of all computer based applications

Database Users

- Users are differentiated by the way they interact with the system. They are classified as:
- Naïve users – They interact with the system by invoking one of the permanent application programs that have been written previously
  - They do not have to know about the implementation details
  - Examples, people accessing database over the web, bank tellers, clerical staff
- Application programmers – are the computer professionals who develop application programs and user interfaces
  - They can use any of the high level languages for the purpose
- Sophisticated users – interact with the system either using a database query language or by using tools such as data analysis software
  - Engineers, Scientist, Analysts
- Specialized users – sophisticated users who write specialized database applications like knowledge-base and expert systems

Data Models

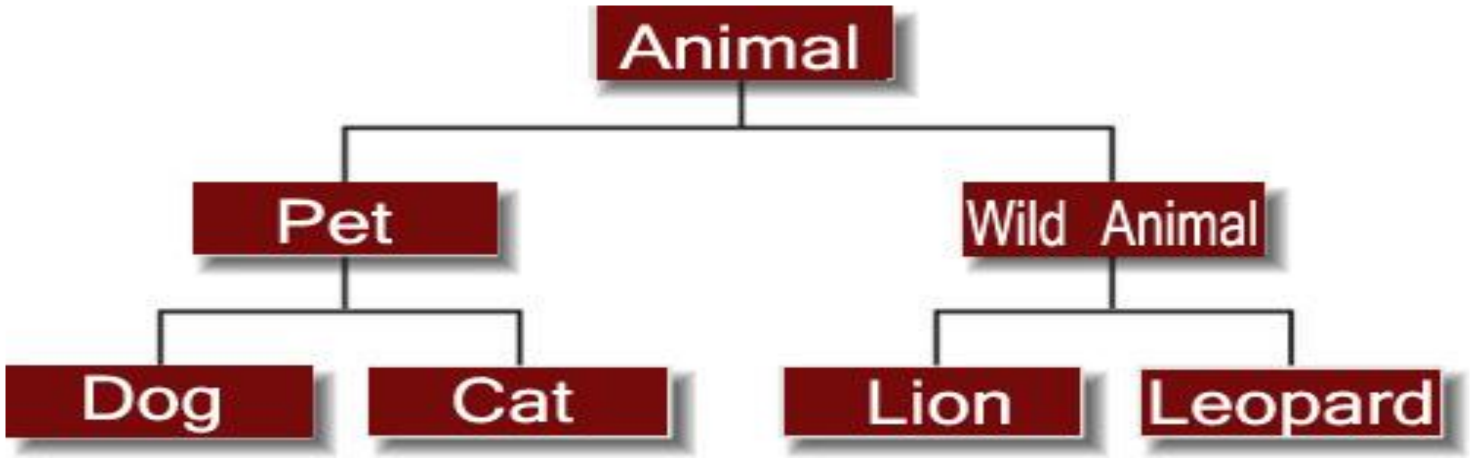
- Data model may be defined as conceptual framework for describing Data, Data relationships, Data semantics and Data constraints.
- Provides a way to describe the design of a database at the physical, logical and view level
- Data models show that how the data is connected and stored in the system. It shows the relationship between data and how data is presented to the user
- Data model is an abstract thing for describing data, data-relationship and constraints.

Data Models: Types

- Relational Model
  - A collection of tables to represent both data and the relationships among the data.
  - Each table has multiple columns and each column has a unique name.
  - Recorded values are atomic values
- E-R Model
  - Entity Relationship Model is a high-level data model
  - ER Model is diagrammatical representation of the conceptual design of a database
  - Based on the perception of a real world objects called entities, and of relationships among these objects.

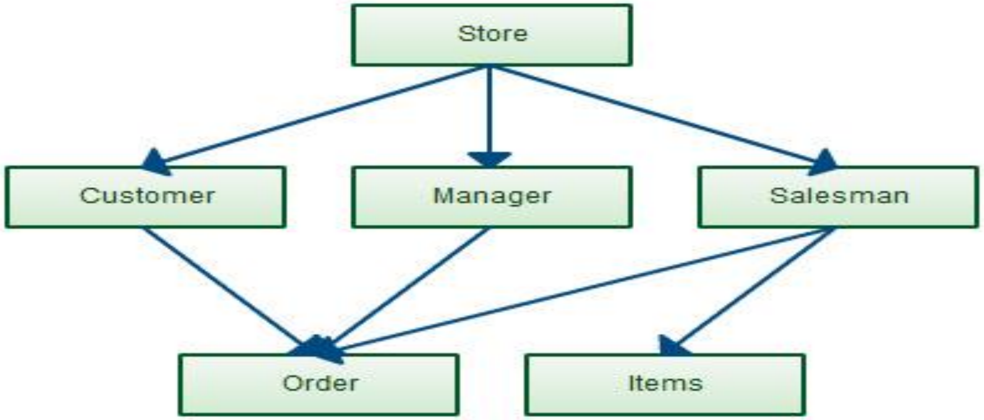
Data Models: Types

- Hierarchical Model
  - It represents the data in a hierarchical tree structure.
  - Hierarchical model has one parent entity with several children entity and the process goes on
  - But at the top we should have only one entity called root (parent-child relationship)
  - It has one to many or many to one relationship



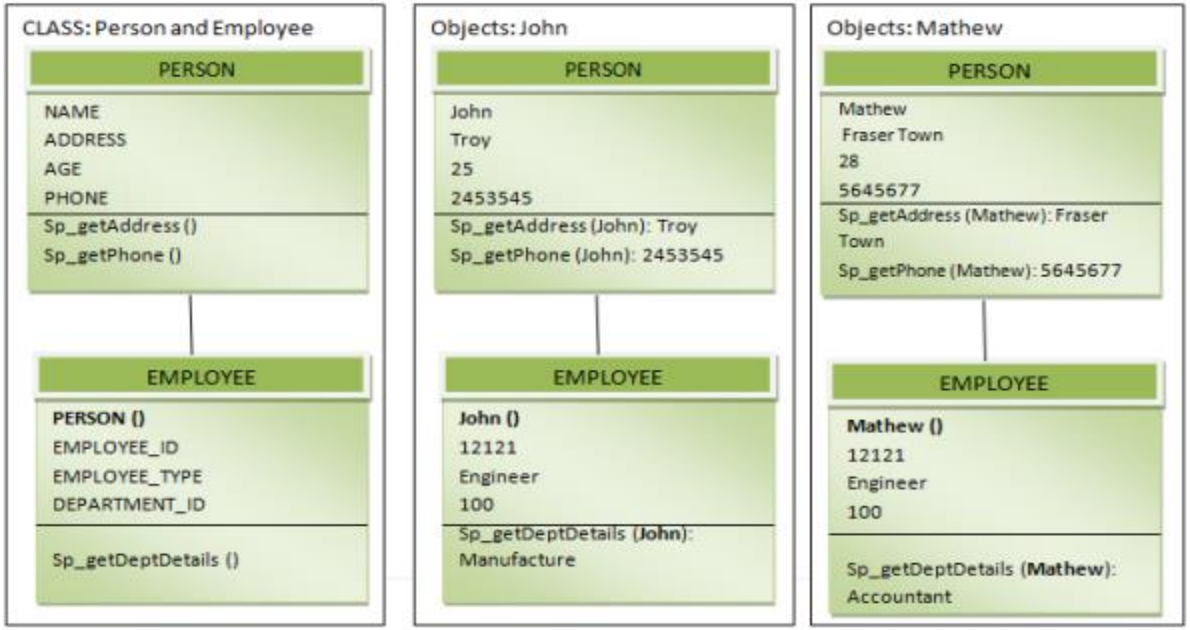
Data Models: Types

- Network Model
  - It is like Hierarchical Model, but the only difference is that it allows a child to have more than one parent
  - Some entities in the graph can be accessed through several paths
  - It has many to many relationships as well



Data Models: Types

- Object Oriented Model
  - It stores the data in the form of objects and classes
  - Extension of the E-R model with notions of encapsulation, methods and object attributes
  - This model handles more complex applications, such as GIS, multimedia and scientific modeling



Data Instances and Schemas

Instance (Database state)

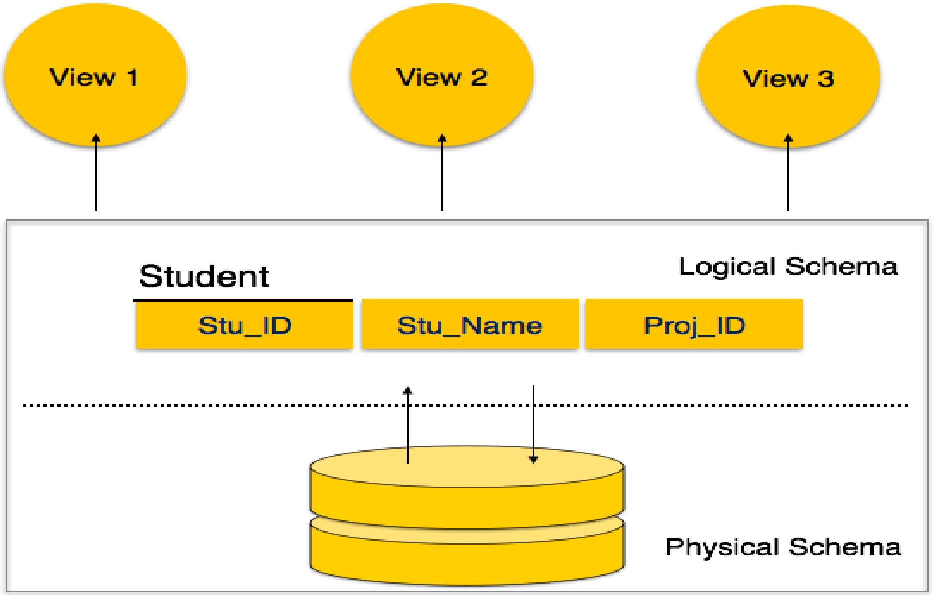
- the actual content of the database at a particular point in time
- Because the databases change over time, the database instance differ at different times

Schema

- refers to the overall skeleton of the database, structural description of database.
  - ➢ Physical schema: database design at the physical level
  - ➢ Describes how data is represented and stored in secondary storage using a particular database management system
  - ➢ Logical schema: database design at the logical level.
  - ➢ What data is stored and relationship among data
  - ➢ Defines relational tables, views and constraints

Data Instances and Schemas

- View schema: View schema defines database at view level that usually describes end-user interaction with database systems.



Data Independence

- The separation of data from the programs that use the data such that any changes made to the data does not need changes in the program is data independence
  - Application program should be free from the way data is represented and stored
- Physical level data independence: It refers to the ability to modify the physical schema without any alterations to the conceptual or logical schema, done for optimization
- E.g., Conceptual structure of the database would not be affected by any change in storage size of the database system server.
  - Changing from sequential to random access files
  - Utilizing new storage devices.

Data Independence

- Logical level data independence: It refers characteristic of being able to modify the logical schema without affecting the external schema or application program.
- The user view of the data would not be affected by any changes to the conceptual view of the data.
- These changes may include insertion or deletion of attributes, altering table structures or relationships to the logical schema etc. does not affect user’s interactivity.

Three schema architecture/ Data abstraction

- **The system hides certain details of how the data are stored and maintained and such view is an abstract view.**
- **The Database System provides users with an abstract view of the data.**
- **Data Abstraction:- The database designers use the complex data structure to represent the data in the database and developer hides the complexity from user from several level of abstraction such as physical level, logical level, and view level. This process is called data abstraction.**
- **It hides how exactly data is stored and maintained.**

Three schema architecture/ Data abstraction

- **Physical level (Internal Level)**
  - **This is the lowest level of data abstraction.**
  - **It tells us how the data is actually stored and organized in memory.**
  - **The access methods like sequential or random access.**
  - **Size of memory, and the type of memory.**
- **Logical Level (Conceptual level)**
  - **Conceptual structure of database, entities and attributes.**
  - **It tells what data is exactly stored in the database like the data type.**
  - **The relationship among those data.**

Three schema architecture/ Data abstraction

- **View level (External Level)**
  - This is the highest level of data abstraction.
  - This level exists to ease the accessibility of the database by an individual user.
  - Simplifies the interaction with the system
  - The system may provide many views for the same database
  - Only a part of the actual database is viewed by the user
- The main purpose of data abstraction is achieving data independence in order to save time, complexity and cost required when database is modified or altered.

Data Abstraction: Three layers

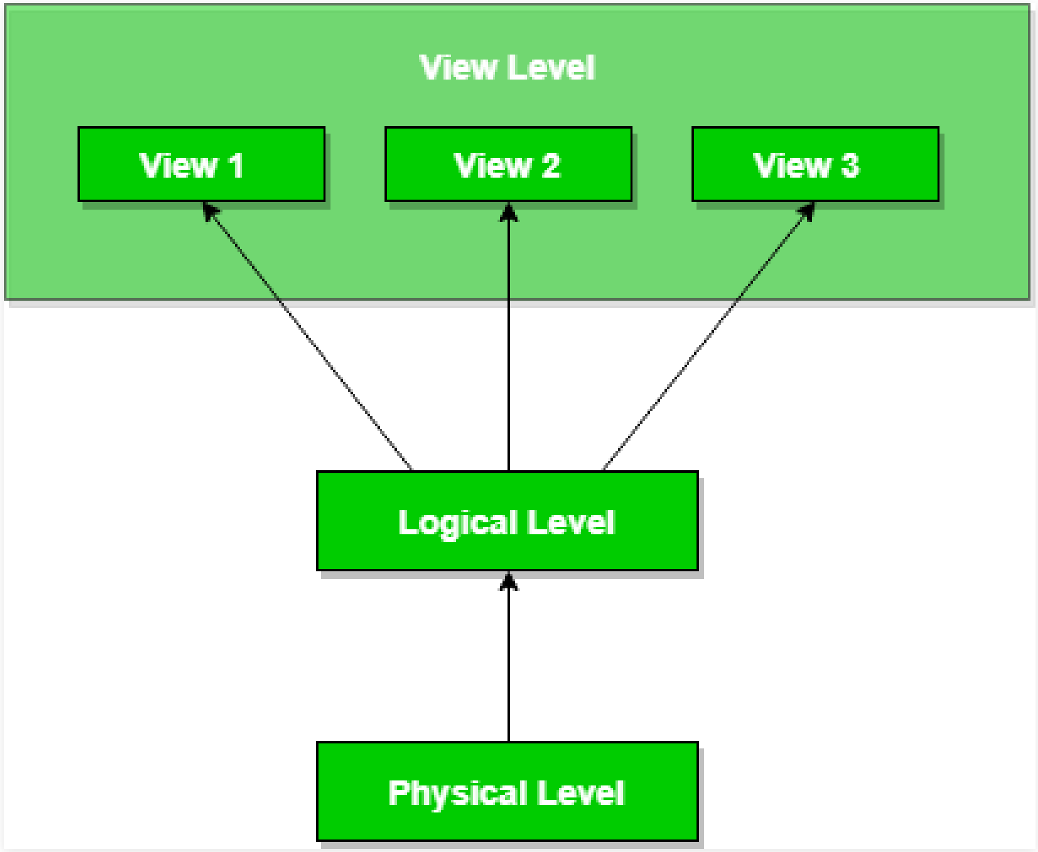


Figure 1: Three layers of data abstraction

Database Languages

- **DBMS provides following languages: Data-Definition Language (DDL), Data-Manipulation Language (DML), Transaction control Language (TCL), Data Control Language (DCL), Storage definition language(SDL) and View definition language (VDL).**
- **DDL: is the language used to define and create database schema. It is used in conceptual level.**
  - **These statements define the implementation details of the database schema hence usually hidden from the users**
  - **CREATE , ALTER, DROP , RENAME and COMMENT are example of DDL statements**
  - **They are also used to specify the data structure and integrity constraints**
  - **Some of the integrity constraints are:**
    - **Domain constraints: decide what type and range of values an attribute can take**
    - **Referential integrity: shows relation between two tables, ensures if a value appears in one table also appears in another table.**

Because each table in a database must have a primary key, this primary key can appear in other tables because of its relationship to data within those tables.

Database Languages

- **Assertion: conditions database must always follow. For example age should be in int and name in string.**
- **Authorization: privileges given to different users (read, insert, update or delete authorization)**

Database Languages

DML: is the language used by the users to access or manipulate data organized by an appropriate data model (Query Language)

- **The types of access or manipulations are:**
  - **Retrieval of information stored in the database (SELECT)**
  - **Insertion of new information into the database (INSERT)**
  - **Deletion of information in the database (DELETE)**
  - **Modification of information stored in the database (UPDATE)**

The common DML commands: SELECT, INSERT, UPDATE, DELETE.

Database Languages

DCL: It stands for data control language. It is used to retrieve the stored or saved data. DCL execution is transactional. For example:

GRANT: it is used to give user access privileges to a database.

REVOKE: it is used to take back permissions from the user.

TCL: It stands for transactional control language and used to manage transaction in the database. These are used to manage the changes made to the data in a table by DML statements. For example:

COMMIT: It is used to save the transaction on the database.

ROLLBACK: It is used to restore the database to the database to original since the last commit.

Database Languages

- VDL: It stands for view definition language and it represents user views and their mapping to the conceptual schema.
- SDL: It stands for storage definition language and it specifies the mapping between two schemas.

Interface

- A database management system interface is a user interface that allows user to input queries to a database without using the query language itself. There are following types of interfaces:

1. Menu based
2. Mobile apps
3. Form based
4. GUI

Interface

5. Natural language
6. Keyboard
7. Search input and output.
8. Parametric users
9. DBA

Menu-based interfaces for web clients:

These interfaces present the user with list of options called menus that is used for user to request any service.

Interface

- Basic advantages of using menus is that they removes the tension of remembering specific command and syntax of any query language.

Form based interface: A form based interfaces displays a form to each user, then user fill out all the given information and matches to the stored data then allows to user to access service.

Interface

- Graphical user interface: It uses any graphical symbol or diagram to interact with system.
- Natural language interfaces: These interface accept any requests written in english or some other languages .

Interface for parametric users: parametric users such as bank tellers often have a small set of operation that must perform repeatedly so special type of interface is used for them.

Interface

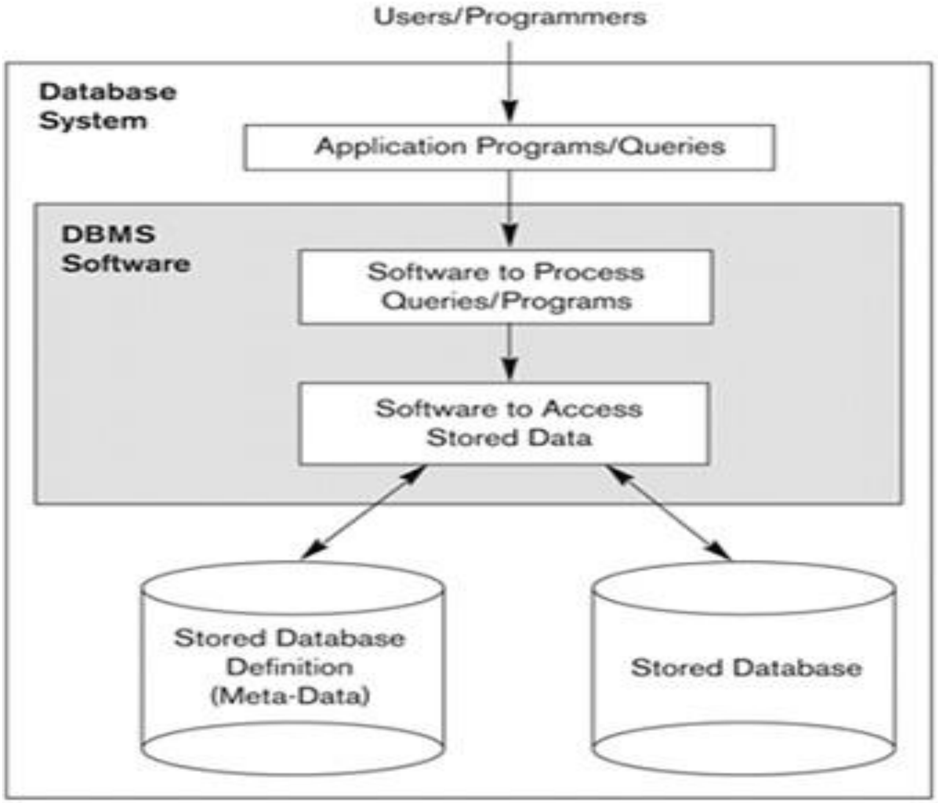
- DBA: it stands for database system administrator. There are many commands that are specially for DBA. For example creating account, access control etc.

Database system environment

- Database system environment deals with the components of a organization that defines and regulate the collection, storage, management and use of data within a database environment.
- Database system is composed of different major parts. Some database system components are as follow:

1. Hardware
2. Software
3. People
4. Procedure
5. Data

database system environment


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Database System Utility

- Backup Utility: in case of crashes and disasters this utility helps to take copies of the database periodically.
- Monitoring Tools: This tool monitors the performance of internal scheme that can be changed and optimize the data access.
- Data loading utility: Data loading utility helps to load data from an external format without writing program.

Database System Utility

- Recovery Utility: From the database backup and history of transactions this utility helps in reconstructing the correct state of database.
- File organization: this utilities helps in restructuring of data from one type to another.

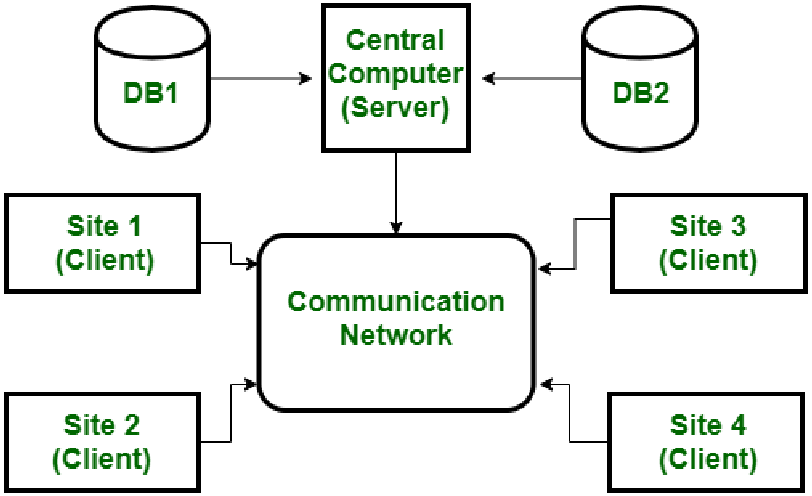
Application Environment and Communication facilities

- The database itself runs in software environment, including the operating system and any programming languages and utilities.
- The software depends on underlying hardware to run, including a processing unit, memory and storage.

(Description of components)

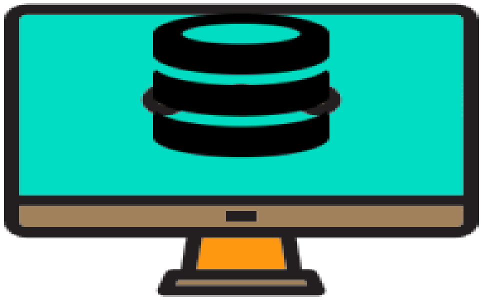
Centralized Architecture

- A centralized architecture for DBMS is one in which all data is stored on a single server and all clients connect to that server in order to access and manipulate the data.



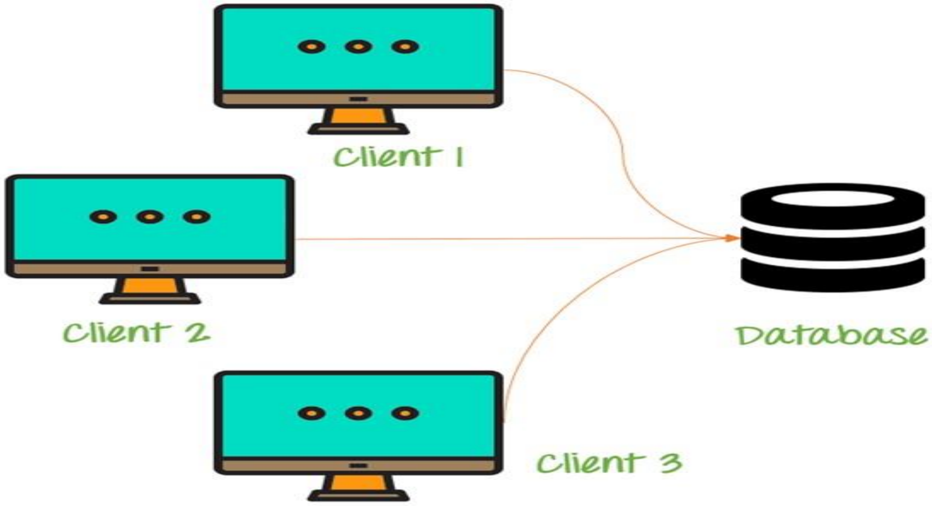
Database Architecture

- DBMS architecture helps in design, development, implementation, and maintenance of a database
- Selecting the correct Database Architecture helps in quick and secure access to this data.
- Main aim is to separate user application from physical database
- 1-tier architecture:
  - The Client, Server, and Database all reside on the same machine.
  - Anytime you install a DB in your system and access it to practice SQL queries it is 1 tier architecture.



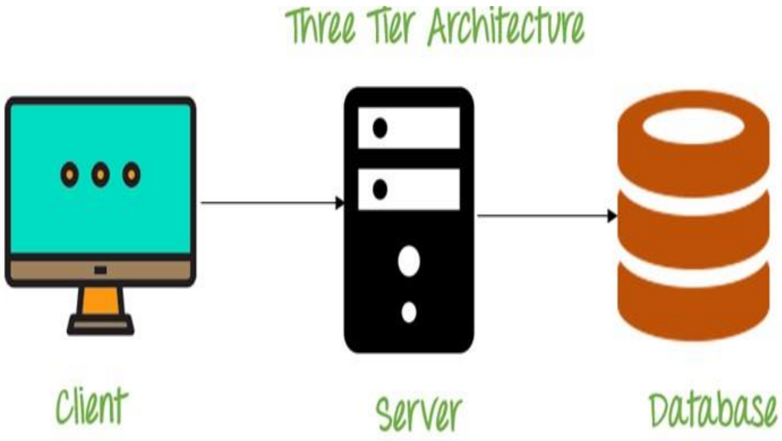
Database Architecture

- 2-tier architecture:
  - The two-tier architecture is like client server application.
  - The direct communication takes place between client and server.
  - Client system sends the request to the Server system and the Server system processes the request and sends back the data to the Client System
  - Desktop applications

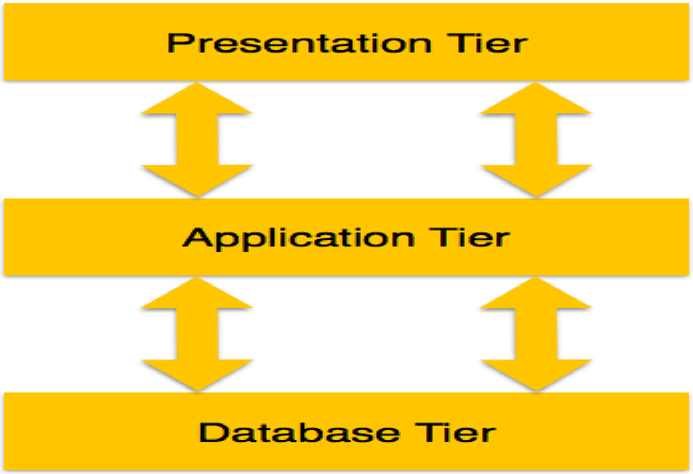


Database Architecture

- 3-tier architecture: Web applications
  - Database Tier – At this tier, the database resides along with its query processing languages i.e. dbms itself
  - Application Tier – the application layer sits in the middle and acts as a mediator between the end-user and the database.
    - At this tier, the application server and the programs that access the database reside.
  - User (Presentation) Tier – End-users operate on this tier and they know nothing about any existence of the database beyond this layer.



Three Tier Architecture
Client
Server
Database



Presentation Tier
Application Tier
Database Tier