Syllabus: **Introduction**: Database system, Characteristics (Database Vs File System), Database Users (Actors on Scene, Workers behind the scene), Advantages of Data base systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Introduction

Databases and database technology are having a major impact on the growing use of computers. It is fair to say that database play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, law, education, and library science, to name a few. The word ***database*** is in such common use that we must begin by defining what a database is.

What is Data

Data is any sort of information which is stored in computer memory. This information can later be used for a website, an application or any other client to store for future purpose. The most common information is User information in the form of user personal, address and banking information.

Let’s consider Facebook, it stores our personal data, images, posts, comments and many more things.

Banking application also stores user data, their transactions details, funds summary etc. All this information is data, but when it put together and store in a structural way, it becomes informational data.

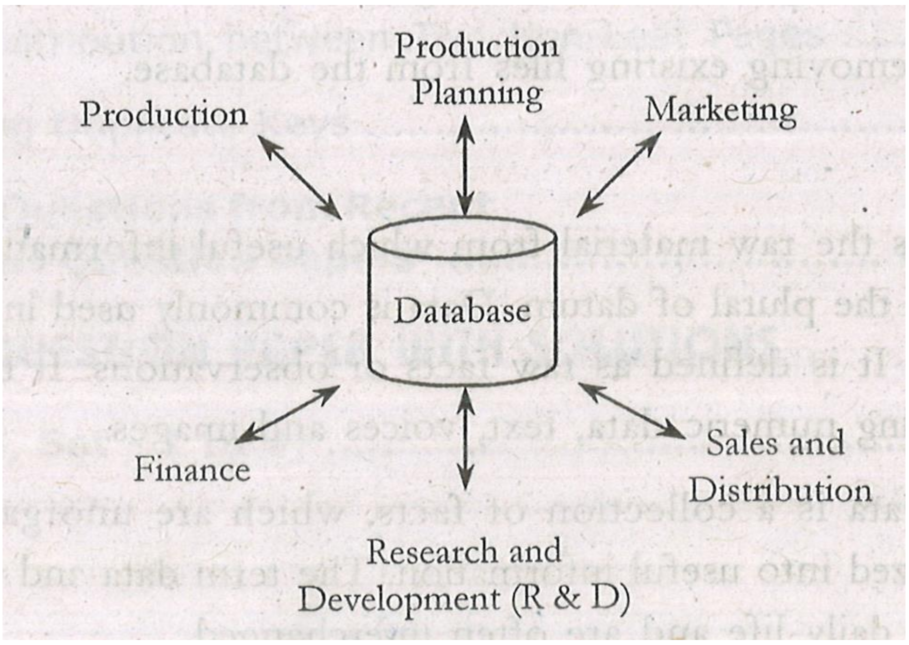
But, how do these applications or websites get data? When you post a status on Facebook, perform a banking transaction online or upload a selfie on Instagram, you are actually sending **data** to the site or to be precise their server. So, we can say any information transmitted or transferred is actually data. Servers filter out the necessary data and store it in **Database**.

What is a Database

A **database** is a collection of related data, so that it can be easily accessed and managed. You can organize data into tables, rows, columns, and index it to make it easier to find

Relevant information. **Database handlers** create a database in such a way that only one set of software program provides access of data to all the users. The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.

There are many **dynamic websites** on the World Wide Web now-a-days which are handled through databases. For example, consider the names, tele-phone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database.



What is DBMS?

A **database management system** (**DBMS**) is a collection of programs that enables users to create and maintain a database. The DBMS is a *general-purpose software system* that facilitates the processes of ***defining***, ***constructing***, ***manipulating***, and ***sharing*** databases among various users and applications.

Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called **meta-data**.

Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS.

Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data.

Sharing a database allows multiple users and programs to access the database simultaneously.

An **application program** accesses the database by sending queries or requests for data to the DBMS. A **query** typically causes some data to be retrieved; a **transaction** may cause some data to be read and some data to be written into the database.

Other important functions provided by the DBMS include ***protecting*** the database and ***maintaining*** it over a long period of time.

Protection includes system protection against hardware or software malfunction (or crashes) and security protection against unauthorized or malicious access.

A typical large database may have a life cycle of many years, so the DBMS must be able to **maintain** the database system by allowing the system to evolve as requirements change over time.

Here are some examples of popular DBMS used these days:

- MySql
- Oracle
- SQL Server
- IBM DB2
- PostgreSQL
- Amazon SimpleDB (cloud based) etc.

Database Users

For a small personal database one person defines, constructs and manipulates the database and there is no sharing. In large organizations many people involved in design, use and maintenance of a large database with hundred of users.

We have two kinds of database users

1. Actors on the scene.
2. Workers behind the scene.

Actors on the scene

Those who actually use and control the database content and those who design, develop and maintain database applications called **"Actors on the Scene"**, and these people involve the day-to-day use of a large database.

We have four categories: -

- **Database Administrators (DBA)**
- **Database Designers**
- **End Users**
- **System analyst and application programmers.**

1. Database administrators (DBA):- The responsibility of DBA is to administrate the resource of databases. He is responsible for authorizing access the database. He will coordinate and monitor the database and acquire software and hardware resources needed. He is responsible for maintaining database security.

2. Database designers: - These are responsible for identifying the data is to be stored in databases and choosing appropriate structures to represent and store this data. These are responsible to communicate database users to know their requirements.

3. End users: - These are the people who need access to database for querying, updating and generating reports. The database exists primarily for their use only. There are four categories of end users:

- **Casual end user**
- **Naive or parametric end user**
- **Sophisticated end user**
- **Standalone end user**

Casual end user:- These users occasionally access the database, but they may need different information each time. These users use a sophisticated database query language to specify their requests.

Ex: through browsers.

Naive or parametric end users:- These users constantly querying and updating the database.

Ex: - 1. Bank account holders continuously checking the balances

2. Reservation agents continuously checking the tickets availability.

Sophisticated end user:- These include engineers, scientists, business analysts who use DBMS to implement their own applications to meet their complex requirements.

Standalone end users:- These users maintain personal database by using ready-made programming packages that provide easy-to-use and menu based or graphical based interface.

Ex:- user of tax package stores a variety of personal financial data for tax purpose.

4. System analyst and application programmers:- system analyst determine the requirements or specifications of end users especially naive or parametric end users. Application programmers implement these specifications as programs. Then they test, debug, document and update these programs.

Workers behind the scene

Those who design and develop the DBMS software and related tools, and the computer systems operators called **"Workers behind the Scene"**. These are the people who are not interested in the database content itself.

There are three categories: -

- **Database system designers and implementers.**
- **Tool developers.**
- **Operators and maintenance personnel.**

1. Database system designers and implementers:- These people will design and implement the DBMS modules and interfaces as a software package. These will implement the modules including the implementation of catalog, query processing, interface processing, buffering data, controlling concurrency, recovery and security.

2. Tool developers:- These people will implement the tools that facilitate database modelling and design, database system design, and improved performance. Tools are optional packages that are often purchased separately.

3. Operators and maintenance personnel:- These are responsible for actual running and maintenance of the hardware and software environment for the database system.

Characteristics of Database Management

A database management system has following characteristics:

- **Data stored into Tables**
- **Reduced Redundancy**
- **Data Consistency**
- **Support multiple users**
- **Query Language**
- **Security**
- **Transaction processing**

1. **Data stored into Tables:** Data is never directly stored into the database. Data is stored into tables, created inside the database. DBMS also allows having relationships between tables which makes the data more meaningful and connected. You can easily understand what type of data is stored where by looking at all the tables created in a database.
2. **Reduced Redundancy:** In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem. But DBMS follows **Normalisation** which divides the data in such a way that repetition is minimum.
3. **Data Consistency:** Data that is being continuosly updated and added, maintaining the consistency of data can become a challenge. But DBMS handles it all by itself.
4. **Support Multiple user:** DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.
5. **Query Language:** DBMS provides users with a simple Query language, using which data can be easily fetched, inserted, deleted and updated in a database.
6. **Security:** The DBMS also takes care of the security of data, protecting the data from un-authorized access.
7. **Transaction processing:** A database management system must include concurrency control subsystems. This feature ensures that data remains consistent and valid during transaction processing even if several users update the same information.

File System vs. DBMS

There are following differences between File system and DBMS:

FILE SYSTEM	DBMS
A file system is a software that manages and organizes the files in a storage medium. It controls how data is stored and retrieved.	DBMS or Database Management System is a software application. It is used for accessing, creating, and managing databases.
The file system provides the details of data representation and storage of data.	DBMS gives an abstract view of data that hides the details
Storing and retrieving of data can't be done efficiently in a file system.	DBMS is efficient to use as there are a wide variety of methods to store and retrieve data.
It does not offer data recovery processes.	There is a backup recovery for data in DBMS.
The file system doesn't have a crash recovery mechanism.	DBMS provides a crash recovery mechanism
Protecting a file system is very difficult.	DBMS offers good protection mechanism.
The file system offers lesser security.	Database Management System offers high security.
Not provide support for complicated transactions.	Easy to implement complicated transactions.
The centralization process is hard in File Management System.	Centralization is easy to achieve in the DBMS system.
There is no efficient query processing in the file system.	You can easily query data in a database using the SQL language.
These systems don't offer concurrency.	DBMS system provides a concurrency facility.

Advantages of Database systems

Compared to the File Based Data Management System, Database Management System has many advantages. Some of these advantages are given below,

- **Reduction of Redundancies**
- **Data independence and efficient access**
- **Data integrity**
- **Data security**
- **Reduced Application Development Time**
- **Conflict Resolution**
- **Data Administration**
- **Concurrent Access**
- **Crash Recovery**

Reduction of Redundancies

- ✓ Redundancy means duplication (Making the same copy again). Reduction of redundancy means avoiding duplication of data and reducing the total amount of storage space required.
- ✓ It also reduces the extra processing time to search the required data in a large mass of data.
- ✓ Another advantage of avoiding duplication is the elimination of the inconsistencies (difficult) in searching the exact data file required.

Data independence and efficient access

- ✓ Database programs in the database are independent of their storage details.
- ✓ The conceptual schema provides physical storage details and external schema provides logical storage details i.e., the conceptual schema provides independence from external schema. It means physical storage details are independent from logical storage details.
- ✓ DBMS strongly provides the efficient access retrieval of the stored information, including support for very large files, index structures in query optimization.

Data integrity

- ✓ Data integrity means that the data values entered in the database must be checked to ensure that they fall within a specified range and are of correct format (type).

For example, the value for the age of an employee must be in the range of 16 and 55.

- ✓ Data integrity also checks that if we are referring any field, then that field must exist. For example, a user is not allowed to transfer funds from a existing savings account to an non-existent savings account.

Data security

- ✓ Data is of vital importance to an organization and must be confidential. Such confidential data must be secured strongly and may not be accessed by any unauthorized person.
- ✓ DBA (Database Administrator) should ensure that proper and different access permissions are given to different types of users. For example, a manager can access the salary details of employees in his department only.

Reduced Application Development Time

- ✓ Since the DBMS provides several pre-defined functions like concurrency control, crash recovery. High-level query facilities etc., only application specified code needs to be written by the users.

Conflict Resolution

- ✓ The DBA should resolve the conflicts among various users to access the same data file.
- ✓ The DBA chooses the best file structure and access method to get optimal performance for using critical applications.

Data Administration

- ✓ DBMS facilitates maintenance and administration of data by providing a common base for a large collection of data that is shared by several users.
- ✓ In addition, the DBA ensures the fine-tuning of data representation periodic back-ups, ensures proper permissions of data access, monitoring all jobs etc.

Concurrent Access

- ✓ Many users access a single program concurrently (at the same time) as if their programs were running in isolation.

- ✓ The DBMS executes the actions of the program in such a way that the concurrent access is permitted, but the conflicting operations are not permitted to proceed concurrently.

Crash Recovery

- ✓ The DBMS maintain a continuous log (record) of the changes made to the data, so that, if there is any system crash by power failure, it can restore the database to a previously stored consistent state.
- ✓ That is the actions of incomplete transactions are undone, so that the database stores only the actions of complete transactions after recovery from a crash.

Database applications

Database management systems are used by many individuals either directly or indirectly. Some of the applications of DBMS are listed below,

- ❖ **Railway Reservation System**
- ❖ **Banking**
- ❖ **Universities and colleges**
- ❖ **Social Media Sites**
- ❖ **Telecommunications**
- ❖ **Online Shopping**
- ❖ **Human Resource Management**
- ❖ **Airline Reservation system**

Railway Reservation System

Database is required to keep record of ticket booking, train's status. Also if trains get late then people get to know it through database update.

Banking

We make thousands of transactions through banks daily and we can do this without going to the bank. So how banking has become so easy that by sitting at home we can send or get money through banks. That is all possible just because of DBMS that manages all the bank transactions.

Universities and colleges

Universities and colleges maintain all the records through DBMS. Like Student's registrations details, results, courses and grades all the information is stored in database.

Social Media Sites

Social media websites are used to share our views and connect with our friends. Millions of users are using social media accounts like facebook, twitter and Google. All the information of users is stored in DBMS.

Telecommunications

Database is used to store the details of number of call made to generate the bill. For prepaid customers, it stores the available credit.

Online Shopping

These sites store the product information, your addresses and credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

Human Resource Management

Human resource management department keeps records of each employee's salary, tax and work through DBMS.

Airline Reservation system

Same as railway reservation system, airline also needs DBMS to keep records of flights status.

Data Models

A data model is a collection of concepts that can be used to describe the structure of a database. Structure of a database means the data types, relationships that should hold the data. Most data models also include a set of basic operations for specifying retrievals and data updating.

An example of a user-defined operation could be a COMPUTE_GPA, which can be applied to a STUDENT object.

Other operations like insert, delete, modify or retrieve any kind of object are often included in the basic data model operations. Basically there are two types of data models.

1. Object based logical models

a) Entity-relationship model (E-R model)
b) Object oriented model

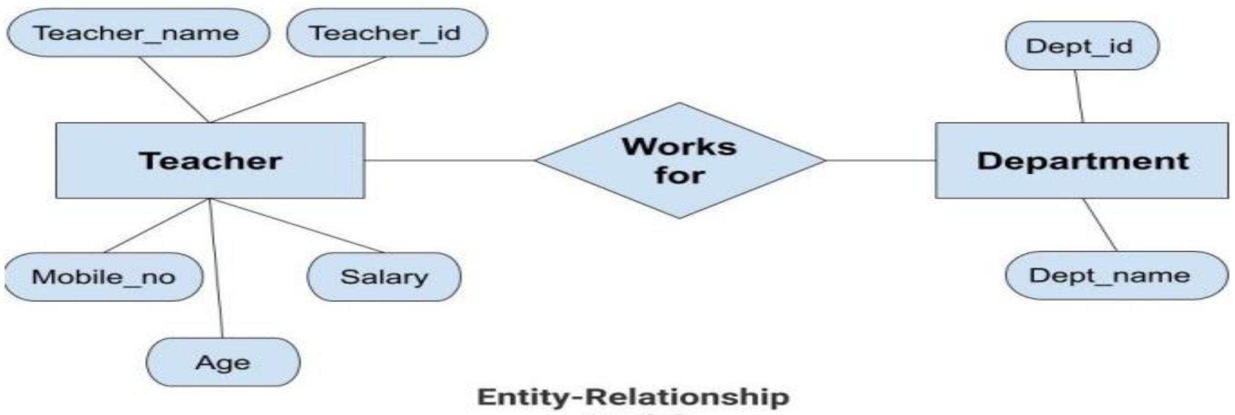
2. Record based logical models

a) Hierarchical model
b) Network model
c) Relational model

A) Entity relationship model (E-R model):- Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand. It is also very easy for the developers to understand the system by just looking at the ER diagram. We use the ER diagram as a visual tool to represent an ER Model.

ER diagram has the following three components:

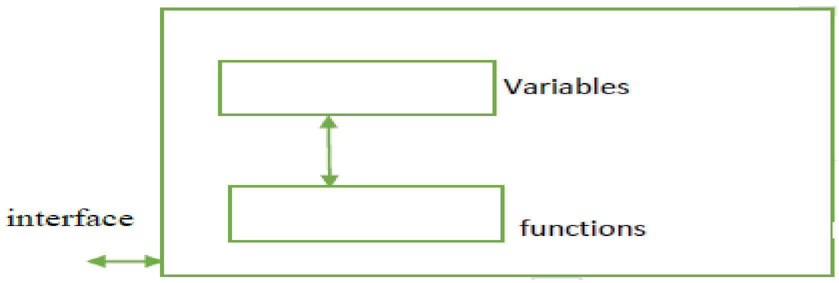
- **Entities:** Entity is a real-world thing. It can be a person, place, or even a concept. Example: Teachers, Students, Course, Building, Department, etc are some of the entities of a School Management System.
- **Attributes:** An entity contains a real-world property called attribute. This is the characteristics of that attribute. Example: The entity teacher has the property like teacher id, salary, age, etc.
- **Relationship:** Relationship tells how two attributes are related. Example: Teacher works for a department.



Entity-Relationship Model

In the above diagram, the entities are Teacher and Department. The attributes of Teacher entity are Teacher_Name, Teacher_id, Age, Salary and Mobile_Number. The attributes of entity Department entity are Dept_id, Dept_name. The two entities are connected using the relationship. Here, each teacher works for a department.

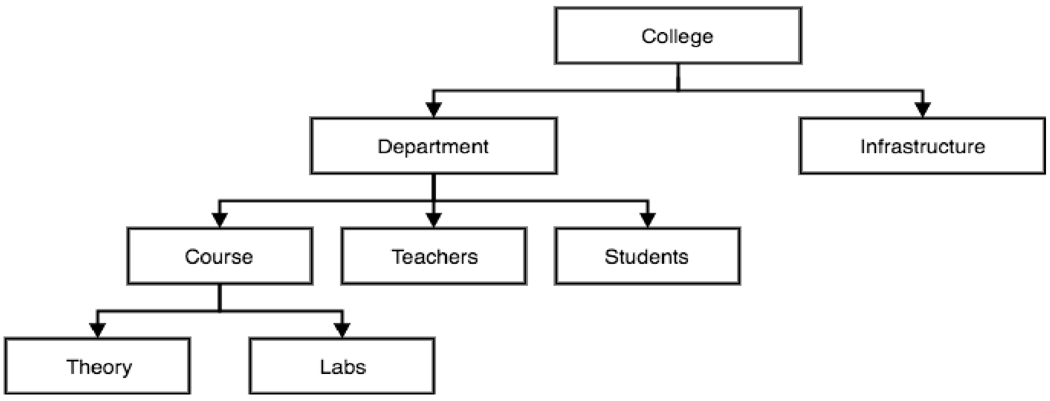
b) Object oriented model: - Like E-R model, this model also models a database as a collection of objects. An object body encapsulates data (variables) as well as methods (functions) to manipulate the data. The objects that contain same type of data and same type of functions are generated together a class. So class is a type definition for proposed objects. The only way an object 'A' can access the data items of another object 'B' is by invoking the methods of 'B'. 'A' can reach this by making method calls to 'B' through B's interface.



The structure of object oriented database is method as a set of classes and database will comprise objects belonging to those classes.

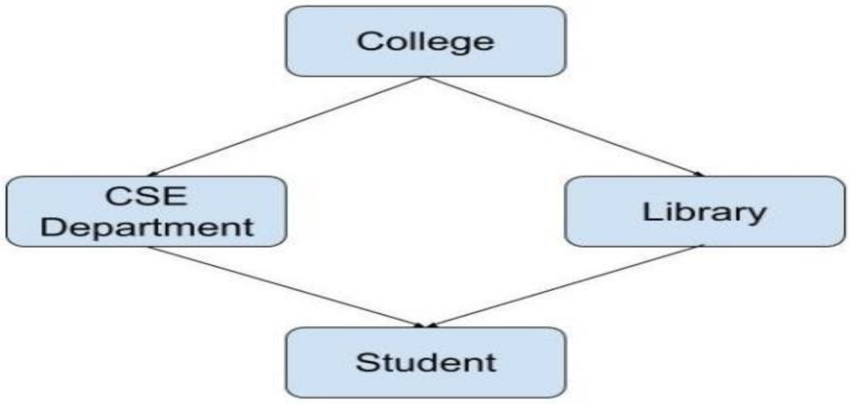
2.Record based logical models:- These models describes data at the logical level. In these models data is stored as collection of records of different types. Each record type can have a fixed number of fields and each field is usually of fixed length.

a) Hierarchical Model: Hierarchical Model was the first DBMS model. This model organizes the data in the hierarchical tree structure. The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node. This model easily represents some of the real-world relationships like food recipes, sitemap of a website etc. for example; one department can have many courses, many professors and off-course many students.



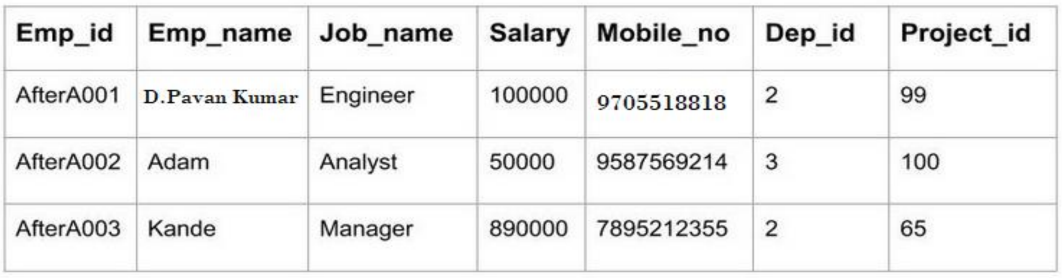
b) Network Model: This model is an extension of the hierarchical model. It was the most popular model before the relational model. This model is the same as the hierarchical model; the only difference is that a record can have more than one parent. It replaces the hierarchical tree with a graph.

Example: In the example below we can see that node student has two parents i.e. CSE Department and Library. This was earlier not possible in the hierarchical model.



c) Relational Model: Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called relations in the relational model.

Example: In this example, we have an Employee table.



Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	D.Pavan Kumar	Engineer	100000	9705518818	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

Features of Relational Model

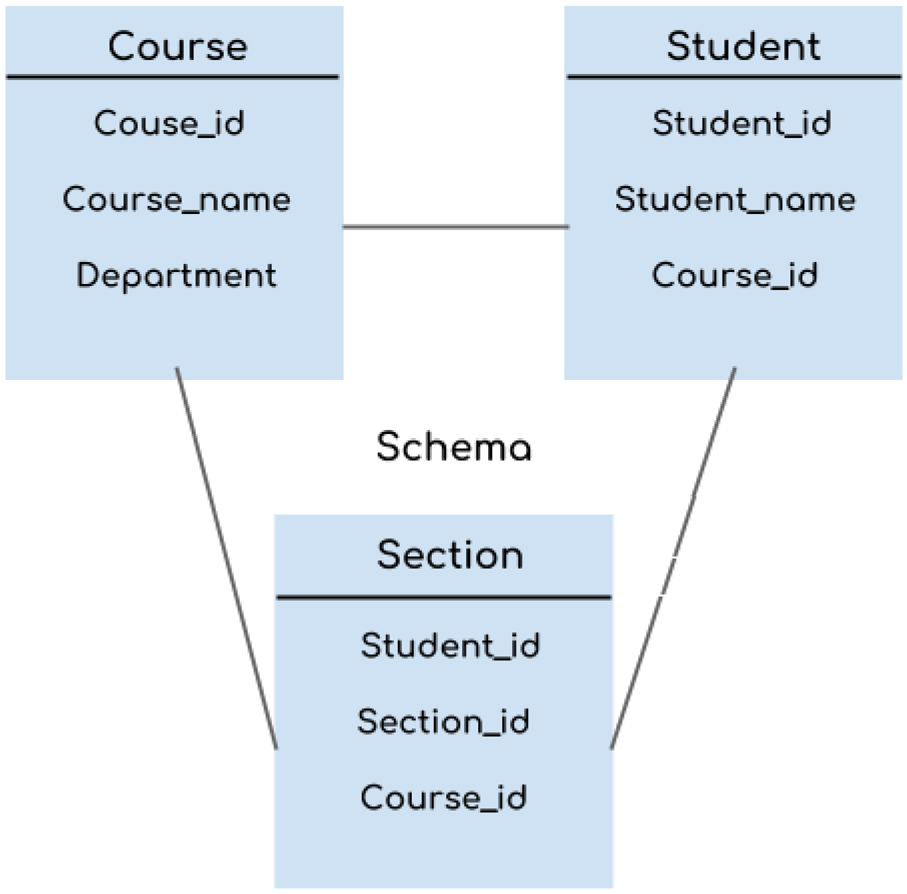
Tuples: Each row in the table is called tuple. A row contains all the information about any instance of the object. In the above example, each row has all the information about any specific individual like the first row has information about John.

Attribute or field: Attributes are the property which defines the table or relation. The values of the attribute should be from the same domain. In the above example, we have different attributes of the employee like Salary, Mobile_no, etc.

Concepts of Schema

Design of a database is called the schema. Schema is of three types: Physical schema, logical schema and view schema.

For example: In the following diagram, we have a schema that shows the relationship between three tables: Course, Student and Section. The diagram only shows the design of the database, it doesn't show the data present in those tables. Schema is only a structural view (design) of a database as shown in the diagram below.



The design of a database at physical level is called **physical schema**, how the data stored in blocks of storage is described at this level.

Design of database at logical level is called **logical schema**, programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).

Design of database at view level is called **view schema**. This generally describes end user interaction with database systems.

Concepts of Instance

Definition of instance: The data stored in database at a particular moment of time is called instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.

For example, let's say we have a single table student in the database, today the table has 100 records, so today the instance of the database has 100 records. Let's say we are going to add another 100 records in this table by tomorrow so the instance of database tomorrow will have 200 records in table. In short, at a particular moment the data stored in database is called the instance that changes over time when we add or delete data from the database.

3-tier architecture of DBMS

Data abstraction:- Abstraction is nothing but hiding complex details and exposing the essential things. Database developers hide internal complexity through several levels of abstraction.

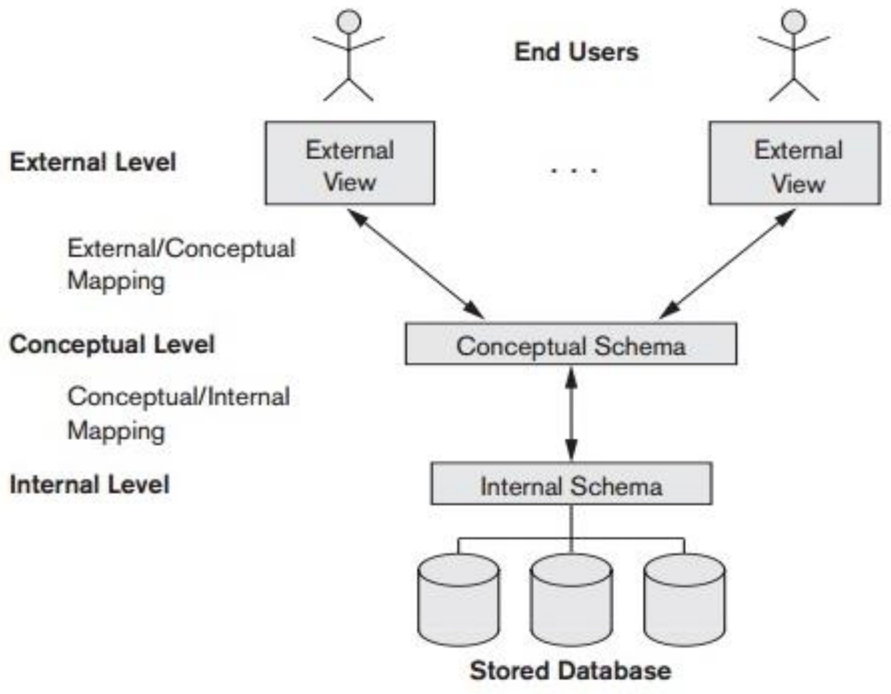
The goal of the three-schema architecture, is to separate the user applications from the physical database. The data abstracted at each of these levels is described by a word schema. A schema means a plan that explains the records and relationships existing between them, at each level.

schemas can be defined at the following three levels:

- **Internal level (or) internal schema**
- **Conceptual level (or) conceptual schema**
- **External level (or) external schema**

Internal level (or) internal schema

The **internal level** has an **internal schema**, which describes the **physical storage** structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.



Conceptual level (or) conceptual schema

The **conceptual level** has a **conceptual schema**, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented. This implementation conceptual schema is often based on a conceptual schema design in a high-level data model.

External level (or) external schema

The **external** or **view level** includes a number of **external schemas** or **user views**. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on an external schema design in a high-level data model.

Data Independence

The three-schema architecture can be used to further explain the concept of **data independence**, which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. We can define two types of data independence:

- ❖ **Logical data independence**
- ❖ **Physical data independence**

Logical data independence

Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the conceptual schema to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item). In the last case, external schemas that refer only to the remaining data should not be affected. After the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before. Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.

Physical data independence

Physical data independence is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well. Changes to the internal schema may be needed because some physical files were reorganized.

for example, by creating additional access structures to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to change the conceptual schema.

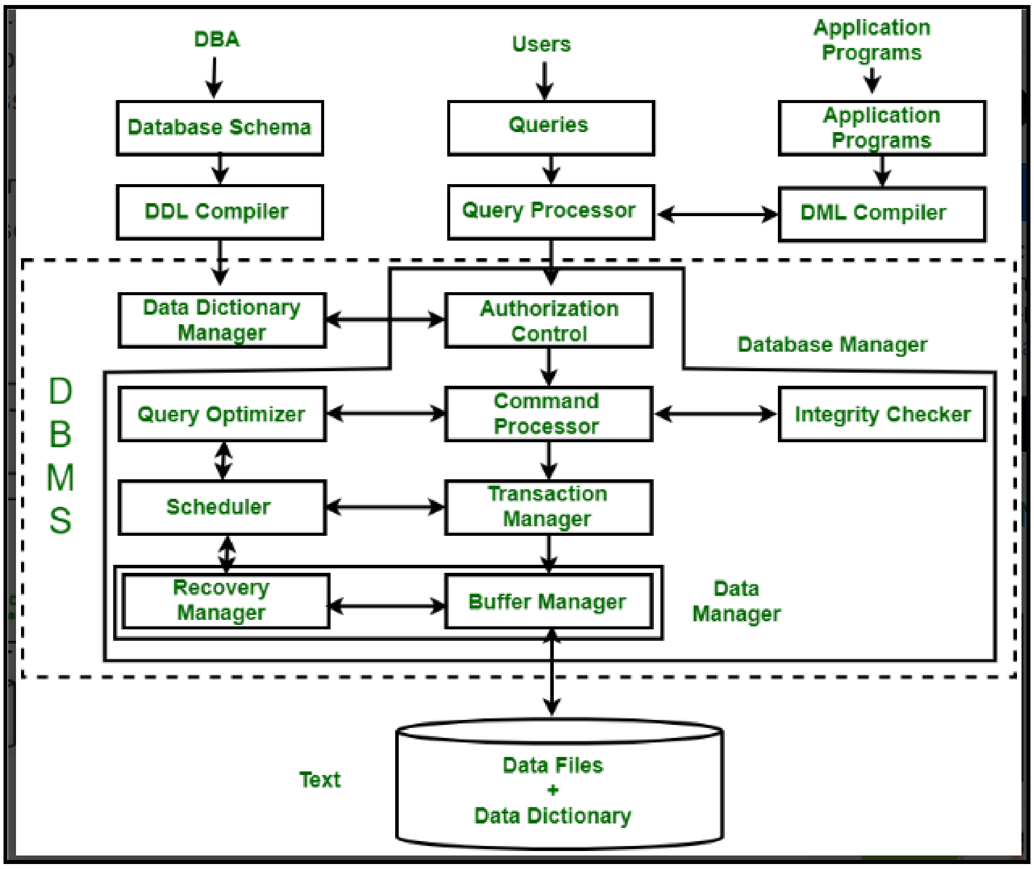
Database system structure

Database Management System (DBMS) is a software that allows access to data stored in a database and provides an easy and effective method of -

- ❖ **Defining the information.**
- ❖ **Storing the information.**
- ❖ **Manipulating the information.**

- **Protecting the information from system crashes or data theft.**
- **Differentiating access permissions for different users.**

The database system is divided into three components: Query Processor, Storage Manager, and Disk Storage. These are explained as following below.



1.Query Processor:

It interprets the requests (**queries**) received from end user via an application program into instructions. It also executes the user request which is received from the DML compiler.

Query Processor contains the following components –

DML Compiler

It processes the DML statements into low level instruction (machine language), so that they can be executed.

DDL Interpreter

It processes the DDL statements into a set of tables containing meta data (data about data).

Embedded DML Pre-compiler

It processes DML statements embedded in an application program into procedural calls.

Query Optimizer

It executes the instruction generated by DML Compiler.

2. Storage Manager:

Storage Manager is a program that provides an interface between the data stored in the database and the queries received. It is also known as Database Control System. It maintains the consistency and integrity of the database by applying the constraints and executes the DCL statements. It is responsible for updating, storing, deleting, and retrieving data in the database.
It contains the following components –

Authorization Manager

It ensures role-based access control, i.e.. checks whether the particular person is privileged to perform the requested operation or not.

Integrity Manager

It checks the integrity constraints when the database is modified.

Transaction Manager

It controls concurrent access by performing the operations in a scheduled way that it receives the transaction. Thus, it ensures that the database remains in the consistent state before and after the execution of a transaction.

File Manager

It manages the file space and the data structure used to represent information in the database.

Buffer Manager

It is responsible for cache memory and the transfer of data between the secondary storage and main memory.

3. Disk Storage:

It contains the following components –

Data Files -

It stores the data.

Data Dictionary

It contains the information about the structure of any database object. It is the repository of information that governs the metadata.

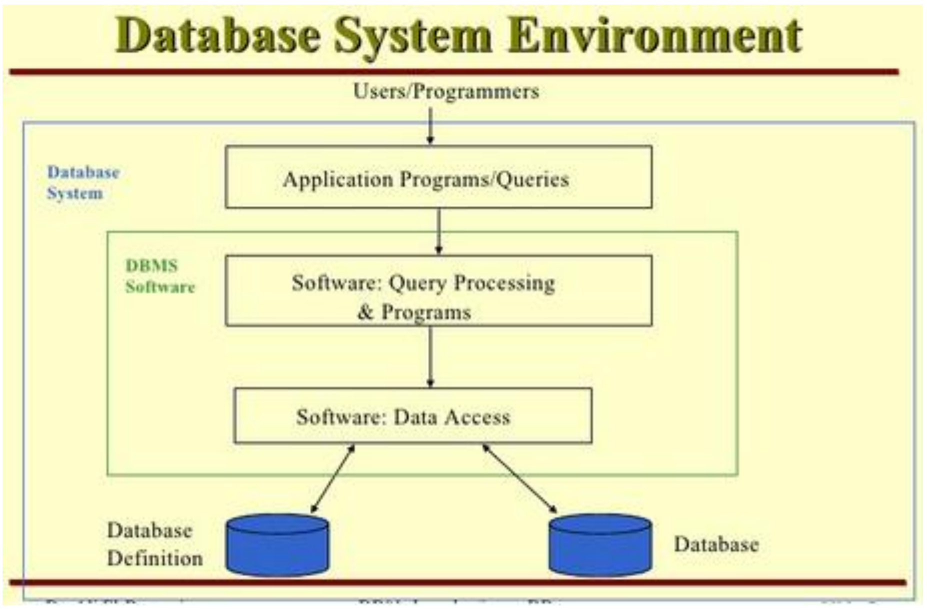
Indices

It provides faster retrieval of data item.

Database system Environment

The term database system refers to the components of an organization that defines and regulate the collection, storage, management and use of data within a database environment. The database management system can be divided into five major components, they are:

- **Hardware**
- **Software**
- **Data**
- **Procedures**
- **Database Access Language**



Hardware

When we say Hardware, we mean computer, hard disks, I/O channels for data, and any other physical component involved before any data is successfully stored into the memory.

When we run Oracle or MySQL on our personal computer, then our computer's Hard Disk, our Keyboard using which we type in all the commands, our computer's RAM, ROM all become a part of the DBMS hardware.

Software

This is the main component, as this is the program which controls everything. The DBMS software is more like a wrapper around the physical database, which provides us with an easy-to-use interface to store, access and update data.

The DBMS software is capable of understanding the Database Access Language and interpret it into actual database commands to execute them on the DB.

Data

Data is that resource, for which DBMS was designed. The motive behind the creation of DBMS was to store and utilize data.

In a typical Database, the user saved Data is present and Meta data is stored.

Metadata is data about the data. This is information stored by the DBMS to better understand the data stored in it.

For example: When I store my Name in a database, the DBMS will store when the name was stored in the database, what is the size of the name, is it stored as related data to some other data, or is it independent, all this information is metadata.

Procedures

Procedures refer to general instructions to use a database management system. This includes procedures to setup and install a DBMS, to login and logout of DBMS software, to manage databases, to take backups, generating reports etc.

Database Access Language

Database Access Language is a simple language designed to write commands to access, insert, update and delete data stored in any database.

A user can write commands in the Database Access Language and submit it to the DBMS for execution, which is then translated and executed by the DBMS.

User can create new databases, tables, insert data, fetch stored data, update data and delete the data using the access language.

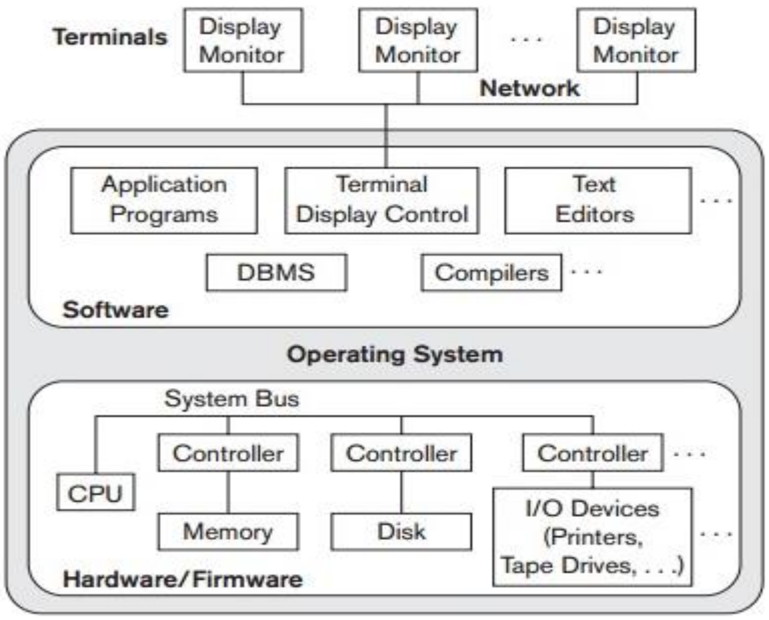
Centralized and Client Server architecture for the database

Centralized and Client Server architecture for the database classified into four different types:

- ❖**Centralized DBMSs Architecture**
- ❖**Basic Client/Server Architectures**
- ❖**Two-Tier Client/Server Architectures for DBMSs**
- ❖**Three-Tier Architectures for Web Applications**

Centralized DBMSs Architecture

Earlier architectures used mainframe computers to provide the main processing for all system functions, including user application programs and user interface programs, as well as all the DBMS functionality. The reason was that most users accessed such systems via computer terminals that did not have processing power and only provided display capabilities. Therefore, 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 the central computer via various types of communications networks.



Most users replaced their terminals with PCs and workstations. At first, database systems used these computers similarly to how they had used display terminals, so that the DBMS itself was still a centralized DBMS in which all the DBMS functionality, application program execution, and user inter-face processing were carried out on one machine. Figure illustrates the physical components in a centralized architecture. Gradually, DBMS systems started to exploit the available processing power at the user side, which led to client/server DBMS architectures.

Basic Client/Server Architectures:

The client/server architecture was developed to deal with computing environments in which a large number of PCs, workstations, file servers, printers, database servers, Web servers, e-mail servers, and other software and equipment are connected via a network. The idea is to define **specialized servers** with specific functionalities. For example, it is possible to connect a number of PCs or small workstations as clients to a **file server** that maintains the files of the client machines. Another machine can be designated as a **printer server** by being connected to various printers; all print requests by the clients are forwarded to this machine. **Web servers** or **e-mail servers** also fall into the specialized server category.

The **client machines** provide the user with the appropriate interfaces to utilize these servers, as well as with local processing power to run local applications. This concept can be carried over to other software packages, with specialized programs such as a **CAD** (computer-aided design) package being stored on specific server machines and being made accessible to multiple clients.

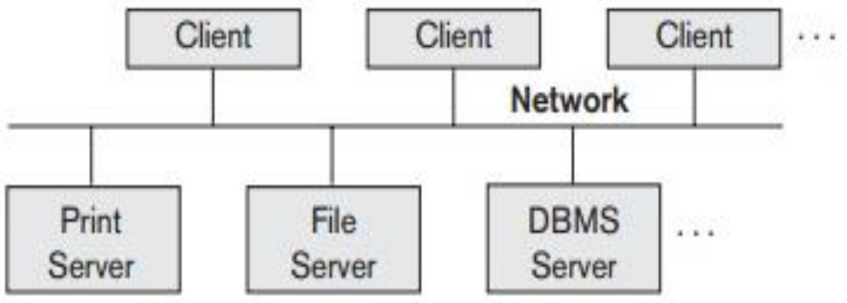


Figure illustrates client/server architecture at the logical level.

A **client** in this framework is typically a user machine that provides user interface capabilities and local processing. When a client requires access to additional functionality— such as database access—that does not exist at that machine, it connects to a server that provides the needed functionality.

A **server** is a system containing both hard-ware and software that can provide services to the client machines, such as file access, printing, archiving, or database access. In general, some machines install only client software, others only server software, and still others may include both client and server software.

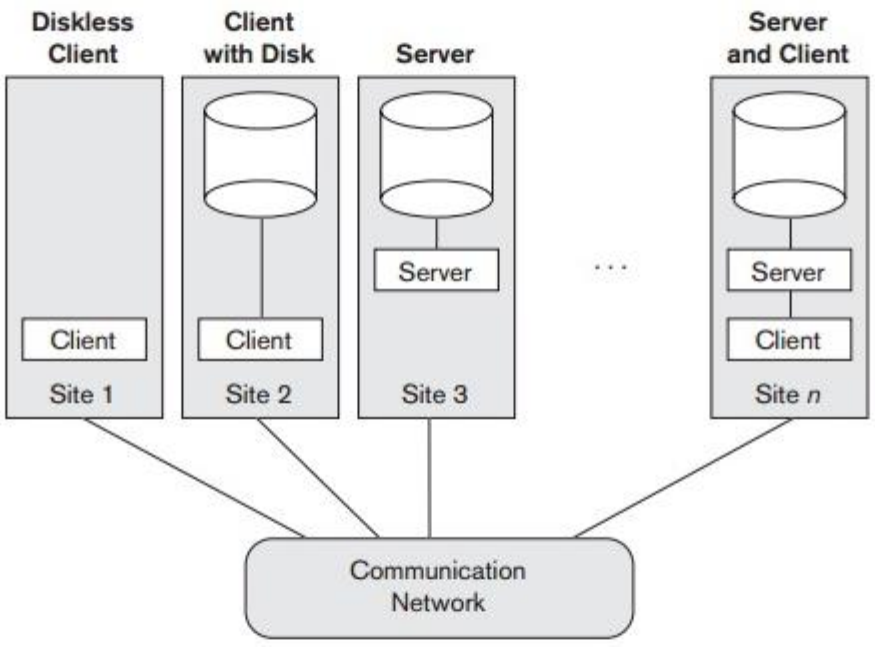
Two-Tier Client/Server Architectures for DBMSs

In relational database management systems (RDBMSs), many of which started as centralized systems, the system components that were first moved to the client side were the user interface and application programs. Because **SQL** provided a standard language for RDBMSs, this created a logical dividing point between client and server. Hence, the query and transaction functionality related to SQL processing remained on the server side. The server is often called a **query server** or **transaction server** because it provides these two functionalities.

The user interface programs and application programs can run on the client side. When DBMS access is required, the program establishes a connection to the DBMS (which is on the server side); once the connection is created, the client program can communicate with the DBMS. A standard called **Open Database Connectivity**

(**ODBC**) provides an application programming interface (**API**), which allows client-side programs to call the DBMS, as long as both client and server machines have the necessary software installed.

The different approach to two-tier client/server architecture was taken by some object-oriented DBMSs, where the software modules of the DBMS were divided between client and server in a more integrated way.



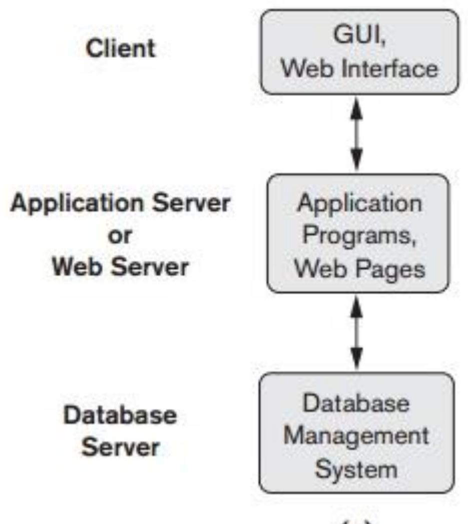
For example, the **server level** may include the part of the DBMS software responsible for handling data storage on disk pages, local concurrency control and recovery, buffering and caching of disk pages, and other such functions.

The **client level** may handle the user interface; data dictionary functions; DBMS interactions with programming language compilers; global query optimization, concurrency control, and recovery across multiple servers; structuring of complex objects from the data in the buffers; and other such functions.

The architectures described here are called **two-tier architectures** because the soft-ware components are distributed over two systems: client and server.

Three-Tier Architectures for Web Applications

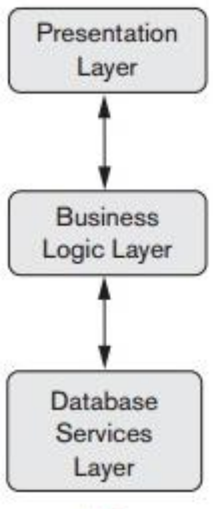
Many Web applications use an architecture called the **three-tier architecture**, which adds an intermediate layer between the client and the database server, as illustrated in Figure.


Client
GUI, Web Interface
Application Server or Web Server
Application Programs, Web Pages
Database Server
Database Management System

(a)

This intermediate layer or **middle tier** is called the **application server** or the **Web server**, depending on the application. This server plays an intermediary role by running application programs and storing business rules (procedures or constraints) that are used to access data from the database server.

It can also improve database security by checking a client's credentials before forwarding a request to the data-base server. Clients contain **GUI** interfaces and some additional application specific business rules. The intermediate server accepts requests from the client, processes the request and sends database queries and commands to the database server. Thus, the ***user interface***, ***application rules***, and ***data access*** act as the three tiers.



(b)

Figure (b) shows another architecture used by database and other application package vendors. The presentation layer displays information to the user and allows data entry. The business logic layer handles intermediate rules and constraints before data is passed up to the user or down to the DBMS. The bottom layer includes all data management services. The middle layer can also act as a **Web server**, which retrieves query results from the database server and formats them into dynamic Web pages that are viewed by the Web browser at the client side.

Database administrators (DBA)

The responsibility of **DBA** is to administrate the resource of databases. He is responsible for authorizing access the database. He will coordinate and monitor the database and acquire software and hardware resources needed. He is responsible for maintaining database security.

Responsibilities of DBA (Data Base Administrator): -

- Creates and maintain all databases required for development and testing.
- Performs ongoing tuning of the database instances.
- Install new versions of the oracle **DBMS** and its tools and any other tools that access the oracle database.
- Planes and implements backup and recovery the oracle database.
- Control’s migrations of programs, database changes, reference data changes through the development life cycle.

- Implement and enforces security for all the oracle databases.
- Performs databases re-organizations to assist performance.
- Evaluates releases of oracle and its tools.
- Provides technical support to application development team.
- Enforces and maintains database constraints to ensure integrity of the database.
- Administrators all database objects, including tables, views, indexes, clusters, packages, and procedures.
- Troubleshoots with problem regarding the databases, application and development tools.
- Create new databases users as required.
- Manage sharing of resources amongst applications.
- DBA has ultimate responsibility for the physical database design.

History of DBMS

Data is a collection of facts and figures. The data collection was increasing day to day and they needed to be stored in a device or a software which is safer.

Charles Bachman was the first person to develop the Integrated Data Store (**IDS**) which was based on network data model for which he was inaugurated with the Turing Award. It was developed in early **1960's**.

In the late **1960's**, **IBM** (International Business Machines Corporation) developed the Integrated Management Systems which is the standard database system used till date in many places. It was developed based on the **hierarchical** database model. It was during the year **1970** that the **relational** database model was developed by **Edgar Codd**. Many of the database models we use today are **relational based**. It was considered the standardized database model from then.

The relational model was still in use by many people in the market. Later during the same decade (**1980's**), **IBM** developed the **Structured Query Language (SQL)** as a part of **R** project. It was declared as a standard language for the **queries** by **ISO** and **ANSI**.

Further, there were many other models with rich features like complex queries, datatypes to insert images and many others. The Internet Age has perhaps influenced the data models much more. Data models were developed using object-oriented programming features, embedding with scripting languages like Hyper Text Markup Language (**HTML**) for **queries**.

Here, are the important **landmarks** from the history:

- 1960 - Charles Bachman designed first DBMS system
- 1970 - Codd introduced IBM'S Information Management System (IMS)
- 1976- Peter Chen coined and defined the Entity-relationship model also known as the ER model
- 1980 - Relational Model becomes a widely accepted database component
- 1985- Object-oriented DBMS develops.
- 1990s- Incorporation of object-orientation in relational DBMS.
- 1991- Microsoft ships MS access, a personal DBMS and that displaces all other personal DBMS products.
- 1995: First Internet database applications
- 1997: XML applied to database processing. Many vendors begin to integrate XML into DBMS products.

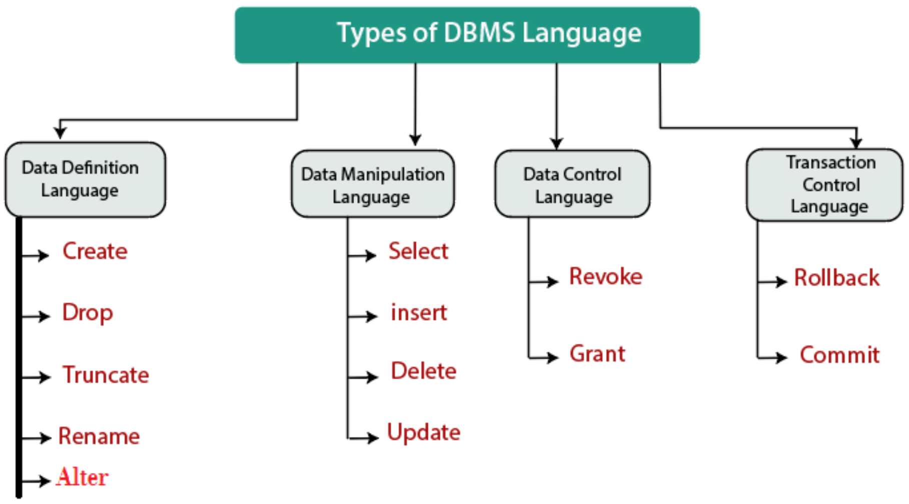
DBMS languages

Database languages are used to read, update and store data in a database. There are several such languages that can be used for this purpose; one of them is **SQL** (Structured Query Language).

Types of DBMS languages:

The following are the databases languages in the database management system:

- **Data Definition Language.**
- **Data Manipulation Language.**
- **Data Control Language.**
- **Transaction Control Language.**



Data Definition Language (DDL)

- **DDL** stands for Data Definition Language. 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.

There are following Data Definition Languages (DDL) Commands:

- **Create:** This command is used to create a new table or a new database.
- **Alter:** This command is used to alter or change the structure of the database table.
- **Drop:** This command is used to delete a table, index, or views from the database.
- **Truncate:** This command is used to delete the records or data from the table, but its structure remains as it is.
- **Rename:** This command is used to rename an object from the database.
- **Comment:** This command is used for adding comments to our table.

Data Manipulation Language (DML)

Data Manipulation Language is a language used to access or manipulate the data in the database. In simple words, this language is used to retrieve the data from the database, insert new data into the database, and delete the existing data from the database.

There are following Data Manipulation Language (DML) Commands:

- **Select:** This command is used to retrieve or access the data from the database table.
- **Insert:** This command is used to insert the records into the table.
- **Update:** This command is used to change/update the existing data in a table.
- **Delete:** This command is used to delete one or all the existing records from the table.

Data Control Language (DCL)

DCL is used to access the stored or saved data. It is mainly used for revoking and granting user access on a database. In the **Oracle** database, this language does not have the feature of rollback. It is a part of **SQL**.

There are following Data Control Language (DCL) Commands:

- **Grant:** This command allows user's access privileges to the database.
- **Revoke:** This command removes the accessibility of users from the database objects.

Transaction Control Languages (TCL)

Transaction Control language is a language which manages transactions within the database. It is used to execute the changes made by the data manipulation language (DML) statements.

There are following Transaction Control Language (TCL) Commands:

- **Commit:** This command is used to save the transactions in the database.
- **Rollback:** This command is used to restore the database to that state which was last committed.