DATABASE MANAGEMENT SYSTEM

Unit – I
INTRODUCTION

Database system:

It is a collection of interrelated data and collection of access programs to manage,utilize,store the data

DataBase: Collection of related data

Database Management System: It is a software used to assist in creating, defining, manipulation and maintenance of rhe database

Characterstics of Database system:

Self-Describing Nature of a Database System
A Database System contains not only the database itself but also the descriptions of data structure and constraints (meta-data). This information is used by the DBMS software or database users if needed. This separation makes a database system totally different from the traditional file-based system in which the data definition is a part of application programs.

Insulation between Program and Data
In the file based system, the structure of the data files is defined in the application programs so if a user wants to change the structure of a file, all the programs that access that file might need to be changed as well. On the other hand, in the database approach, the data structure is stored in the system catalog not in the programs. Therefore, one change is all that's needed.

Support multiple views of data
A view is a subset of the database which is defined and dedicated for particular users of the system. Multiple users in the system might have different views of the system. Each view might contain only the data of interest to a user or a group of users.

Sharing of data and Multiuser system
A multiuser database system must allow multiple users access to the database at the same time. As a result, the multiuser DBMS must have concurrency control strategies to ensure several users access to the same data item at the same time, and to do so in a manner that the data will always be correct – data integrity

Control Data Redundancy
In the Database approach, ideally each data item is stored in only one place in the database. In some cases redundancy still exists so as to improve system performance, but such redundancy is controlled and kept to minimum.

Data Sharing
The integration of the whole data in an organization leads to the ability to produce more information from a given amount of data.

Enforcing Integrity Constraints
DBMSs should provide capabilities to define and enforce certain constraints such as data type, data uniqueness, etc.

Restricting Unauthorised Access
Not all users of the system have the same accessing privileges. DBMSs should provide a security subsystem to create and control the user accounts.

Data Independence
System data (Meta Data) descriptions are separated from the application programs. Changes to the data structure is handled by the DBMS and not embedded in the program.

Transaction Processing
The DBMS must include concurrency control subsystems to ensure that several users trying to update the same data do so in a controlled manner. The results of any updates to the database must maintain consistency and validity.

Providing multiple views of data
A view may be a subset of the database. Various users may have different views of the database itself. Users may not need to be aware of how and where the data they refer to is stored.

Providing backup and recovery facilities
If the computer system fails in the middle of a complex update process, the recovery subsystem is responsible for making sure that the database is restored to the stage it was in before the process started executing.

Managing information
Managing information means taking care of it so that it works for us, and is useful for the work we are doing. The information we collect is no longer subject to "accidental disorganization" and becomes more easily accessible and integrated with the rest of our work. Managing information using a database allows us to become strategic users of the data we have.

DBMS vs File system

S.No	Difference factor	File System	DBMS
1	Definition	**Database management system (DBMS)** is a collection of interrelated data and a set of programs to access those data. Some of the very well known DBMS are Microsoft Access, Microsoft SQL Server, Oracle, SAP, dBASE, FoxPro, IBM dB2, SQLite etc.	**A file management system** is an abstraction to store, retrieve, management and update a set of files. A File Management System keep track on the files and also manage them.
2	Data Redundancy	In file system approach, each user defines and implements the needed files for a specific application to run. For example in sales department of an enterprise, One user will be maintaining the details of how many sales personnel are there in the sales department and their grades. Another user will be maintaining the sales person salary details.	Although the database approach does not remove redundancy completely, it controls the amount of redundancy in the database because in database approach, a single repository of data is maintained that is defined once and then accessed by many users. The fundamental characteristic of database approach is that the database system not only contains data's but it contains complete definition or description of the database structure and constraints.
3	Sharing of data	File system doesn't allow sharing of data or data sharing is very complex.	In DBMS data can be shared very easily due to centralized system
4	Data Consistency	When data is redundant, it is difficult to update. For e.g. if we want to change or update employee's address, then we have to make changes at all the places where data of that employee is stored. If by mistake, we forgot to change or update the address at one or more place then data inconsistency will occur i.e. the appearance of same data will differ from each other.	In DBMS, as there is no or less data redundancy, data remains consistent.

Database Users

Actors on Scene

Actors on the Scene These apply to "large" databases, not "personal" databases that are defined, constructed, and used by a single person via, say, Microsoft Access. Users may be divided into Those who actually use and control the database content, and those who design, develop and maintain database applications (called "Actors on the Scene"), and Those who design and develop the DBMS software and related tools, and the computer systems operators (called "Workers behind the Scene").

1. Database Administrator (DBA): This is the chief administrator, who oversees and manages the database system (including the data and software). Duties include authorizing

users to access the database, coordinating/monitoring its use, acquiring hardware/software for upgrades, etc. In large organizations, the DBA might have a support staff.

2. Database Designers: They are responsible for identifying the data to be stored and for choosing an appropriate way to organize it. They also define views for different categories of users. The final design must be able to support the requirements of all the user sub-groups.

3. End Users: These are persons who access the database for querying, updating, and report generation. They are main reason for database's existence!

Casual end users: use database occasionally, needing different information each time; use query language to specify their requests; typically middle- or high-level managers.

Naive/Parametric end users: Typically the biggest group of users; frequently query/update the database using standard canned transactions that have been carefully programmed and tested in advance. Examples: Bank tellers check account balances, post withdrawals/deposits Reservation clerks for airlines, hotels, etc., check availability of seats/rooms and make reservations. Shipping clerks (e.g., at UPS) who use buttons, bar code scanners, etc., to update status of intransit packages. **Sophisticated end users:** engineers, scientists, business analysts who implement their own applications to meet their complex needs.

Stand-alone users: Use "personal" databases, possibly employing a special purpose (e.g., financial) software package. Mostly maintain personal databases using ready-to-use packaged applications

4. System Analysts, Application Programmers, Software Engineers: System Analysts: determine needs of end users, especially naive and parametric users, and develop specifications for canned transactions that meet these needs. Database Management Systems (10CS54) 2 Application Programmers: Implement, test, document, and maintain programs that satisfy the specifications mentioned above.

Workers behind the Scene

DBMS system designers/implementors: provide the DBMS software that is at the foundation of all this.

Tool developers: design and implement software tools facilitating database system Design, performance monitoring, creation of graphical user interfaces, prototyping, etc.

Operators and maintenance personnel: responsible for the day-to-day operation of the system.

Advantages of Data base systems

The database management system has a number of advantages as compared to traditional computer file-based processing approach.

1.Controlling Data Redundancy

In non-database systems each application program has its own private files. In this case, the duplicated copies of the same data is created in many places. In DBMS, all data of an organization is integrated into a single database file. The data is recorded in only one place in the database and it is not duplicated.

2.Sharing of Data

In DBMS, data can be shared by authorized users of the organization. The database administrator manages the data and gives rights to users to access the data. Many users can be authorized to access the same piece of information simultaneously. The remote users can also share same data. Similarly, the data of same database can be shared between different application programs.

3.Data Consistency

By controlling the data redundancy, the data consistency is obtained. If a data item appears only once, any update to its value has to be performed only once and the updated value is immediately available to all users. If the DBMS has controlled redundancy, the database system enforces consistency.

4.Integration of Data

In Database management system, data in database is stored in tables. A single database contains multiple tables and relationships can be created between tables (or associated data entities). This makes easy to retrieve and update data.

5.Integration Constraints

Integrity constraints or consistency rules can be applied to database so that the correct data can be entered into database. The constraints may be applied to data item within a single record or the may be applied to relationships between records.

6.Data Security

Form is very important object of DBMS. You can create forms very easily and quickly in DBMS. Once a form is created, it can be used many times and it can be modified very easily. The created forms are also saved along with database and behave like a software component. A form provides very easy way (user-friendly) to enter data into database, edit data and display data from database. The non-technical users can also perform various operations on database through forms without going into technical details of a fatabase.

7.Report Writers

Most of the DBMSs provide the report writer tools used to create reports. The users can create very easily and quickly. Once a report is created, it can be used may times and it can be modified very easily. The created reports are also saved along with database and behave like a software component.

8.Control Over Concurrency

In a computer file-based system, if two users are allowed to access data simultaneously, it is possible that they will interfere with each other. For example, if both users attempt to perform update operation on the same record, then one may overwrite the values recorded by the other. Most database

management systems have sub-systems to control the concurrency so that transactions are always recorded with accuracy.

9.Backup and Recovery Procedures

In a computer file-based system, the user creates the backup of data regularly to protect the valuable data from damage due to failures to the computer system or application program. It is very time consuming method, if amount of data is large. Most of the DBMSs provide the 'backup and recovery' sub-systems that automatically create the backup of data and restore data if required.

10.Data Independence

The separation of data structure of database from the application program that uses the data is called data independence. In DBMS, you can easily change the structure of database without modifying the application program.

Database applications

Applications where we use Database Management Systems are:

- **Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry**: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System**: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
- **Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online shopping**: You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

Brief introduction of different Data Models

High-level Conceptual Data Models

High-level conceptual data models provide concepts for presenting data in ways that are close to the way people perceive data. A typical example is the entity relationship model, which uses main concepts like entities, attributes and relationships. An entity represents a real-world object such as an employee or a project. The entity has attributes that represent properties such as an employee's name, address and birthdate. A relationship represents an association among entities; for example, an employee works on many projects. A relationship exists between the employee and each project.

Physical data model :represent the model where it describes how data are stored in computer memory, how they are scattered and ordered in the memory, and how they would be retrieved from memory. Basically physical data model represents the data at data layer or internal layer. It represents each table, their columns and specifications, constraints like primary key, foreign key etc. It basically represents how each tables are built and related to each other in DB.

Representation data model: It is between High level & Low level data model Which provides concepts that may be understood by end-user but that are not too far removed from the way data is organized by within the computer.

The most common data models are

1. **Relational Model :**The Relational Model uses a collection of tables both data and the relationship among those data.

Customer-Name	Security Number	Address	City	AccountNumber
Preethi	111-222-3456	Yelhanka	Bangalore	A-101
Sharan	111-222-3457	Hebbal	Bangalore	A-125
Preethi	112-123-9878	Jaynagar	Bangalore	A-456
Arun	123-987-9909	MG road	Bangalore	A-987
Preethi	111-222-3456	Yelhanka	Bangalore	A-111
Rocky	222-232-0987	Sanjay Nagar	Bangalore	A-111

2.Account –Table

Account-Number	Balance
A-101	1000.00
A-125	1200.00
A-456	5000.00
A-987	1234.00
A-111	3000.00

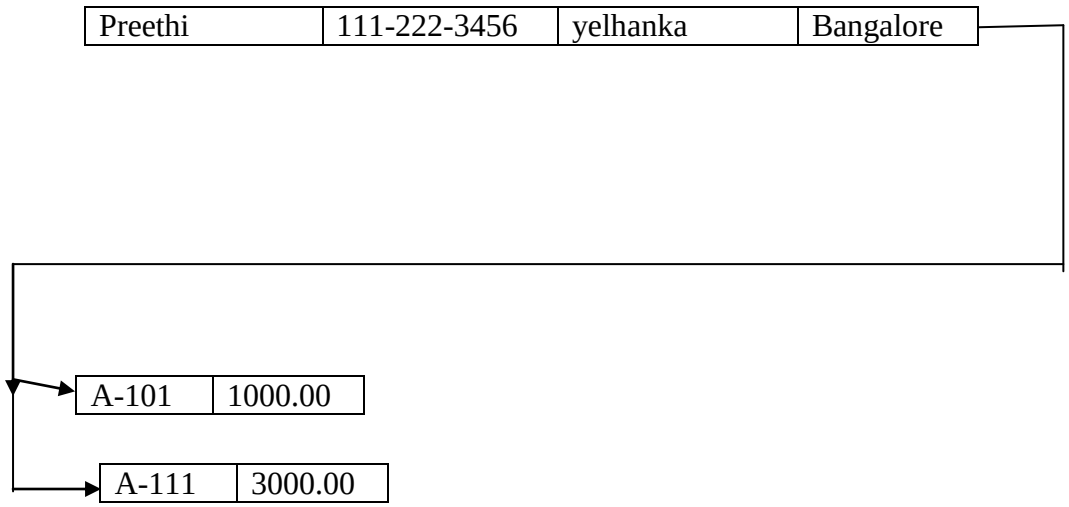
Customer Preethi and Rocky share the same account number A-111

Advantages:

1. The main advantage of this model is its ability to represent data in a simplified format.
2. The process of manipulating record is simplified with the use of certain key attributes used to retrieve data.
3. Representation of different types of relationship is possible with this model.

2. Network Model

The data in the network model are represented by collection of records and relationships among data are represented by links, which can be viewed as pointers.



The records in the database are organized as collection of arbitrary groups.

Advantages:

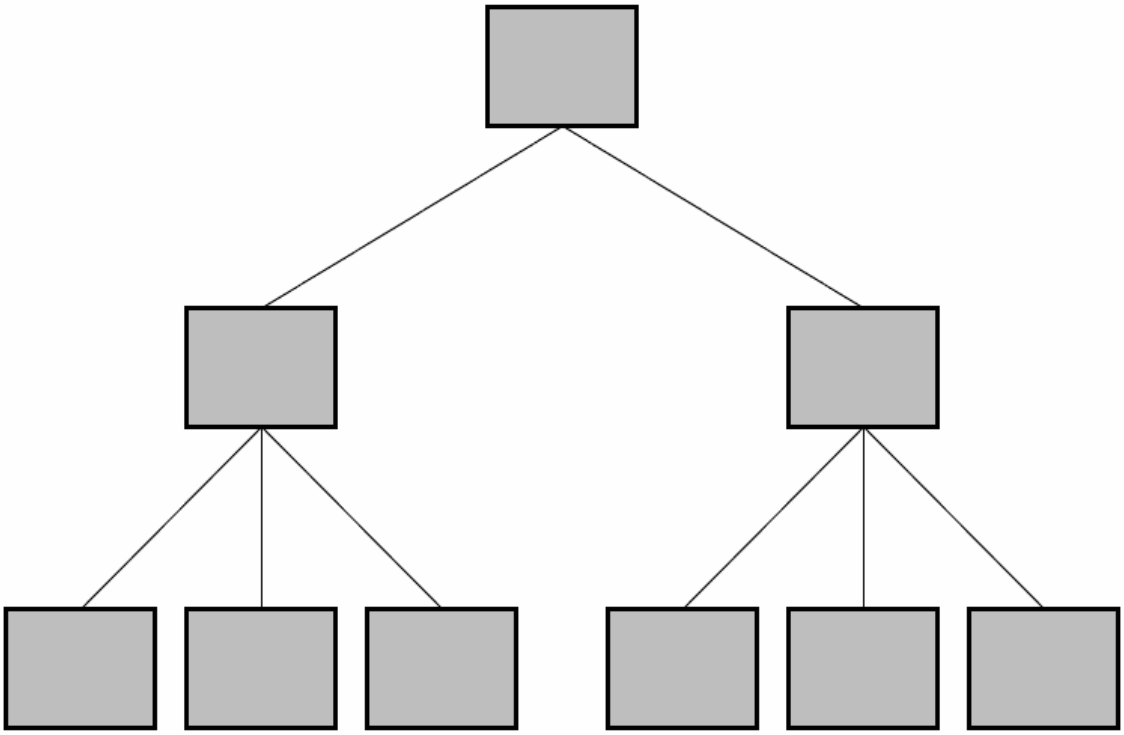
1. Representation of relationship between entities is implemented using pointers which allows the representation of arbitrary relationship
2. Unlike the hierarchical model it is easy.
3. data manipulation can be done easily with this model.

3.Hierarchical Model

A hierarchical data model is a data model which the data is organized into a tree like structure. The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent. All attributes of a specific record are listed under an entity type.

Advantages:

1. The representation of records is done using an ordered tree, which is natural method of implementation of one–to-many relationships.
2. Proper ordering of the tree results in easier and faster retrieval of records.
3. Allows the use of virtual records. This result in a stable database especially when modification of the data base is made.



4 Object-oriented Data Models

• Several models have been proposed for implementing in a database system.
• One set comprises models of persistent O-O Programming Languages such as C++ (e.g., in OBJECTSTORE or VERSANT), and Smalltalk (e.g., in GEMSTONE).
• Additionally, systems like O2, ORION (at MCC – then ITASCA), IRIS (at H.P.-used in Open OODB).

5 Object-Relational Models

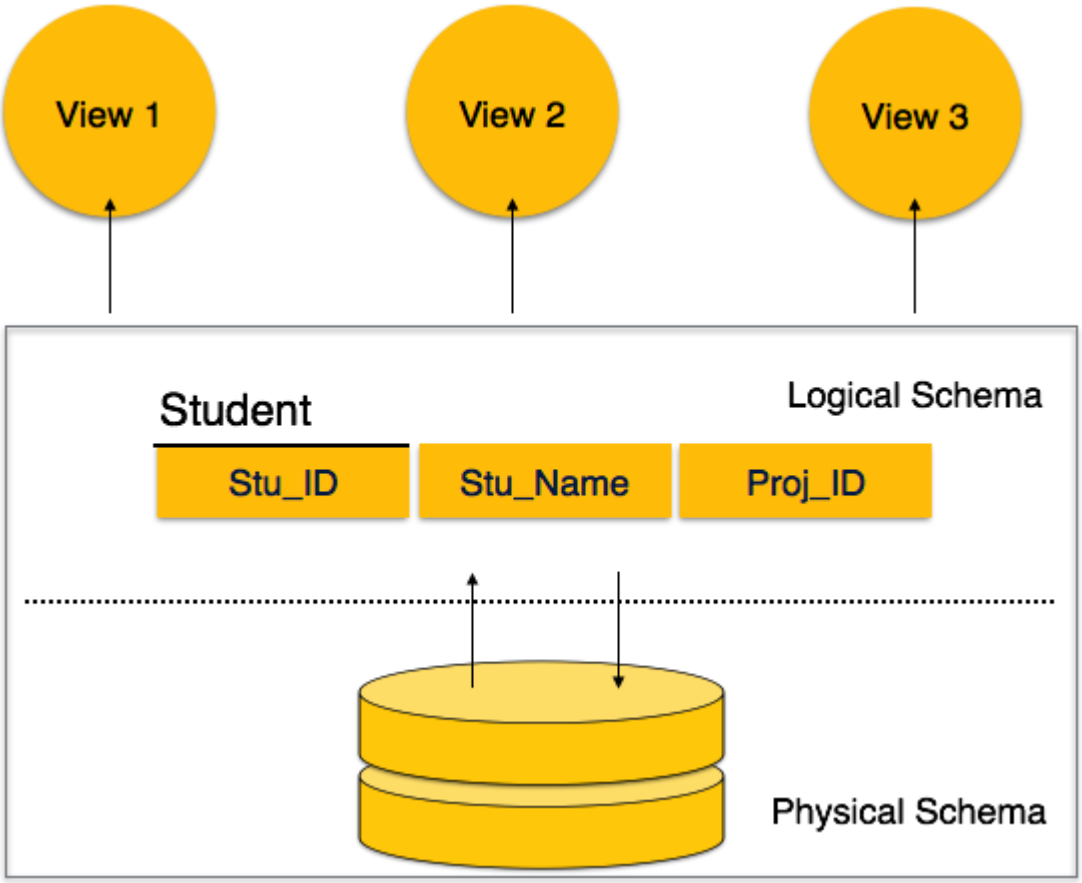
• Most Recent Trend. Started with Informix
• Universal Server.
• Relational systems incorporate concepts from object databases leading to objectrelational.
• Object Database Standard: ODMG-93, ODMG-version 2.0,ODMG-version 3.0.
• Exemplified in the latest versions of Oracle-10i,DB2, and SQL Server and other DBMSs.
• Standards included in SQL-99 and expected to be enhanced in future SQL standards.

Concepts of Schema, Instance and data independence

Database Schema

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.



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

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

Data independence is ability to modify a schema definition in one level without affecting a schema definition in the next higher level.

There are two levels of data independence:

1. Physical Data Independence
2. Logical Data Independence

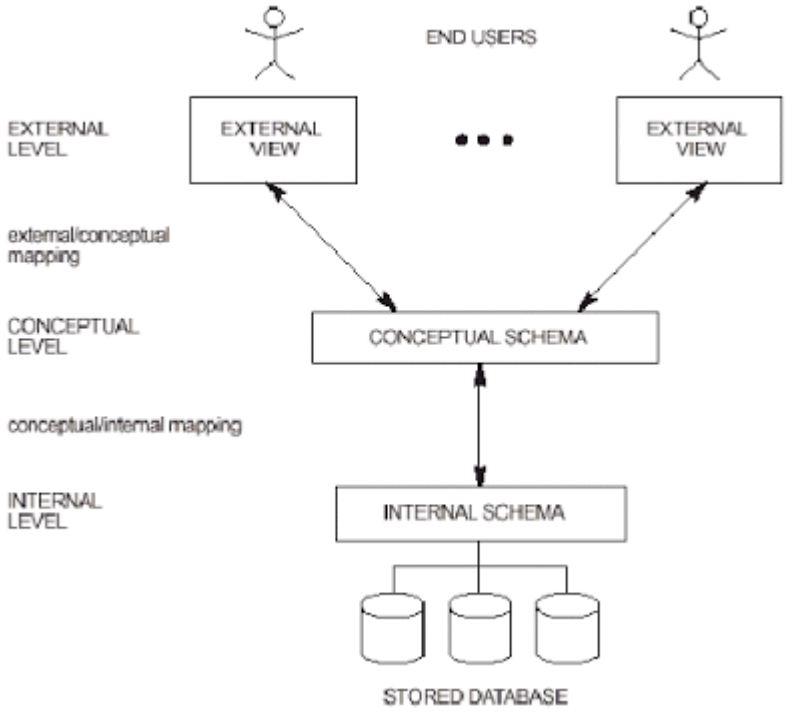
1. Physical Data Independence:

- Physical Data Independence is the ability to modify the physical schema without requiring any change in application programs.
- Modifications at the internal levels are occasionally necessary to improve performance. possible modifications at internal levels are change in file structures, compression techniques, hashing algorithms, storage devices, etc.
- Physical data independence separates conceptual levels from the internal levels.
- This allows to provide a logical description of the database without the need to specify physical structures.
- Comparatively, it is easy to achieve physical data independence.

2. Logical Data Independence:

- Logical data independence is ability to modify the conceptual schema without requiring any change in application programs.
- Modification at the logical levels are necessary whenever the logical structures of the database is altered.
- Logical data independence separates external level from the conceptual view.
- Comparatively it is difficult to achieve logical data independence.
- Application programs are heavily dependent on logical structures of the data they access.so any change in logical structure also requires programs to change.

Three tier schema architecture for data independence



EXTERNAL LEVEL (highest level)

• The user's view of the database.

• Consists of a number of different external views of the DB.

• Describes part of the DB for particular group of users.

• Provides a powerful and flexible security mechanism by hiding parts of the DB from certain users. The user is not aware of the existence of any attributes that are missing from the view

. • It permits users to access data in a way that is customized to their needs, so that the same data can be seen by different users in different ways, at the same time.

CONCEPTUAL LEVEL

The logical structure of the entire database as seen by DBA.

• What data is stored in the database

• The relationships among the data.

• Complete view of the data requirements of the organization, independent of any storage consideration.

• Represents: - ∀entities, attributes, relations - constraints on data - semantic information on data - security, integrity information Supports each external view: any data available to a user must be contained in, or derivable from the conceptual level.

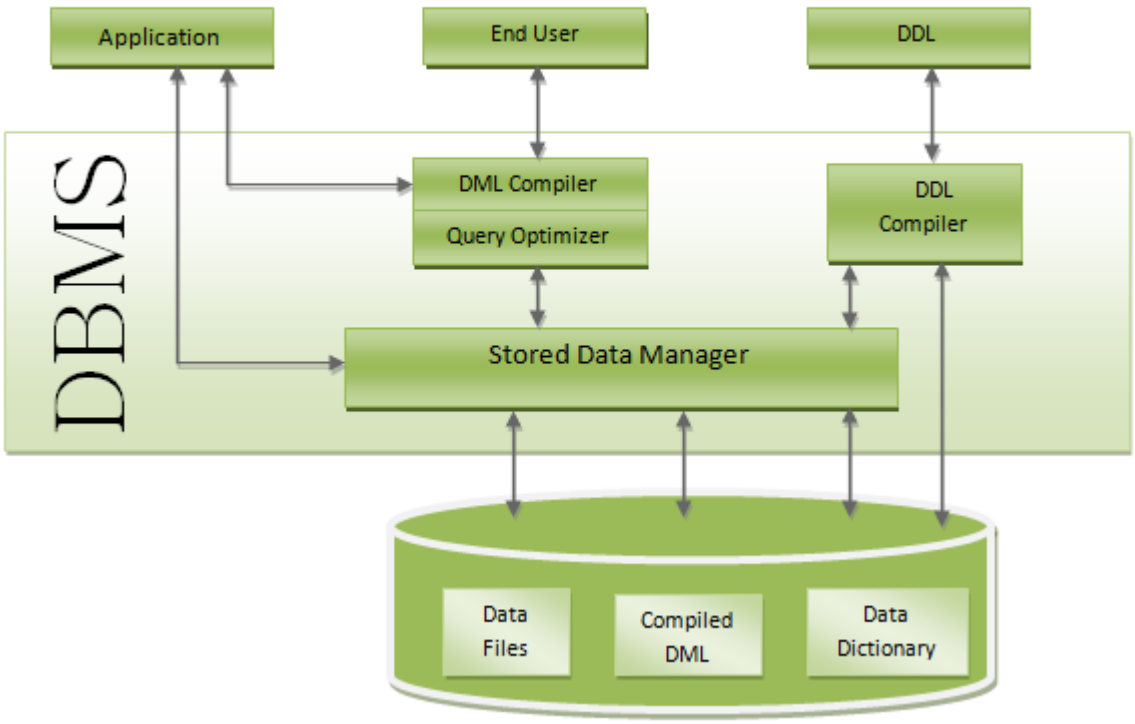
INTERNAL LEVEL

• Physical representation of the DB on the computer.

• How the data is stored in the database.

• Physical implementation of the DB to achieve optimal run– time performance and storage space utilization. - Storage space allocation for data and indexes - Record description for storage - Record placement - Data compression, encryption

Database System Structure



- **Applications: -** It can be considered as a user friendly web page where the user enters the requests. Here he simply enters the details that he needs and presses buttons to get the data.

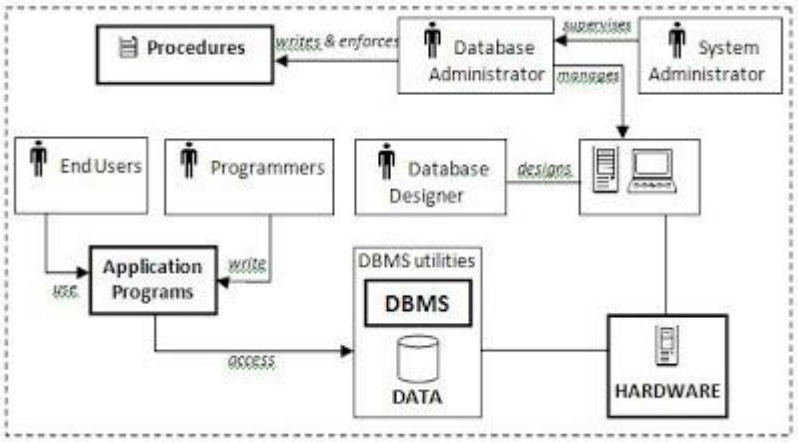
- **End User: -** They are the real users of the database. They can be developers, designers, administrator or the actual users of the database.
- **DDL: -** Data Definition Language (DDL) is a query fired to create database, schema, tables, mappings etc in the database. These are the commands used to create the objects like tables, indexes in the database for the first time. In other words, they create structure of the database.
- **DDL Compiler: -** This part of database is responsible for processing the DDL commands. That means these compiler actually breaks down the command into machine understandable codes. It is also responsible for storing the metadata information like table name, space used by it, number of columns in it, mapping information etc.
- **DML Compiler: -** When the user inserts, deletes, updates or retrieves the record from the database, he will be sending request which he understands by pressing some buttons. But for the database to work/understand the request, it should be broken down to object code. This is done by this compiler. One can imagine this as when a person is asked some question, how this is broken down into waves to reach the brain!
- **Query Optimizer: -** When user fires some request, he is least bothered how it will be fired on the database. He is not all aware of database or its way of performance. But whatever be the request, it should be efficient enough to fetch, insert, update or delete the data from the database. The query optimizer decides the best way to execute the user request which is received from the DML compiler. It is similar to selecting the best nerve to carry the waves to brain!
- **Stored Data Manager: -** This is also known as Database Control System. It is one the main central system of the database. It is responsible for various tasks
    - It converts the requests received from query optimizer to machine understandable form. It makes actual request inside the database. It is like fetching the exact part of the brain to answer.
    - It helps to maintain consistency and integrity by applying the constraints. That means, it does not allow inserting / updating / deleting any data if it has child entry. Similarly it does not allow entering any duplicate value into database tables.
    - It controls concurrent access. If there is multiple users accessing the database at the same time, it makes sure, all of them see correct data. It guarantees that there is no data loss or data mismatch happens between the transactions of multiple users.
    - It helps to backup the database and recover data whenever required. Since it is a huge database and when there is any unexpected exploit of transaction, and reverting the changes are not easy. It maintains the backup of all data, so that it can be recovered.

Data Files: - It has the real data stored in it. It can be stored as magnetic tapes, magnetic disks or optical disks.

Compiled DML: - Some of the processed DML statements (insert, update, delete) are stored in it so that if there is similar requests, it will be re-used.

Data Dictionary: - It contains all the information about the database. As the name suggests, it is the dictionary of all the data items. It contains description of all the tables, view, materialized views, constraints, indexes, triggers etc.

Database system environment



1. **Hardware:** Hardware refers to all of the system's physical devices; for example, computers storage devices, printers, network devices and etc.

2. **Software:** To make the database system work properly, three types of software are needed: operating system, DBMS software, and application programs.

a) ***Operating system****:* It manages all hardware components and allows other software to run on the computers. Examples of operating system software include Windows, Linux and etc.

b) ***DBMS software:*** It manages the database within the database system. Some examples of DBMS software include Oracle, Access, MySql and etc.

c) ***Application programs:*** These are used to access and manipulate data in the DBMS and to manage the computer environment in which data access and manipulation take place. Application programs are most commonly used to access data to generate reports. Most of the application programs provide GUI.

3. **People:** This component includes all users of the database system. According to the job nature, five types of users can be identified: systems administrators, database administrators, database designers, systems analysts and programmers, and end users.

a) ***System administrators:*** They supervise the database system's general operations.

b) ***Database administrators****:* They are also known as DBAs. They manage the DBMS and ensure that the database is functioning properly.

c) ***Database designers:*** They design the database structure. They are the database architects. As this is very critical, the designer's job responsibilities are increased.

d) ***Systems analysts and programmers:*** They design and implement the application programs. They design and create the data entry screens, reports, and procedures through which end users can access and manipulate the data.

e) ***End users:*** They are the people who use the application programs to run the organization's daily operations. For example, sales-clerks, supervisors, managers are classified as end users.

4. **Procedures:** Procedures are the instructions and rules that supervise the design and use of the database system. Procedures are a critical component of the system. Procedures play an important role in a company because they enforce the standards by which business is conducted in an organization

5. **Data:** Data refers the collection of facts stored in the database. Because data are the raw material from which information is generated, no database can exist without database.

Centralized Architecture of DBMS

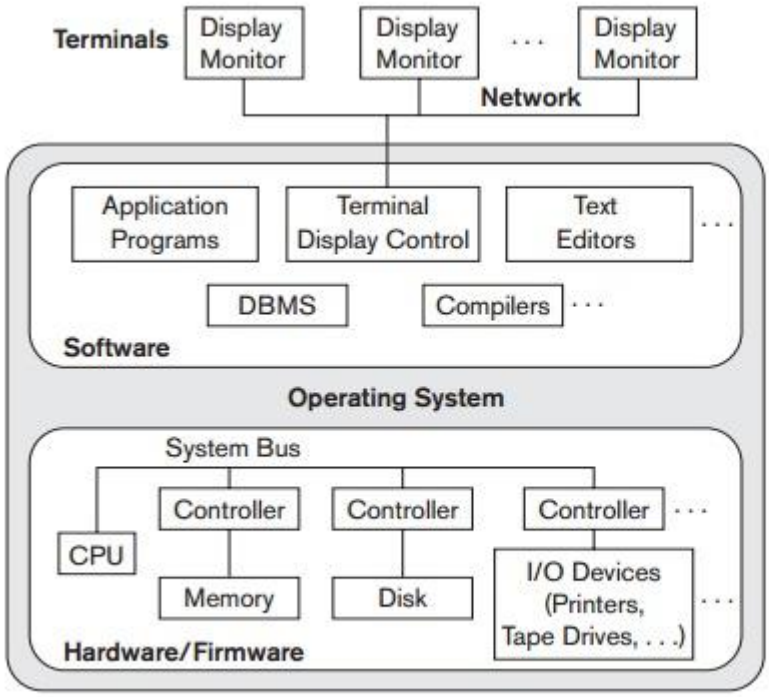


Figure 2.4
A physical centralized architecture.

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. 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

2.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 resources provided by specialized servers can be accessed by many client machines. 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.

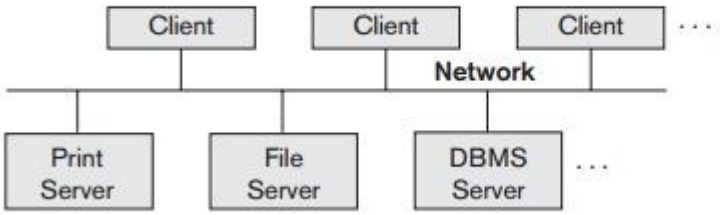


Figure 2.5
Logical two-tier client/server architecture.

Three-Tier Client Server Architecture

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 below

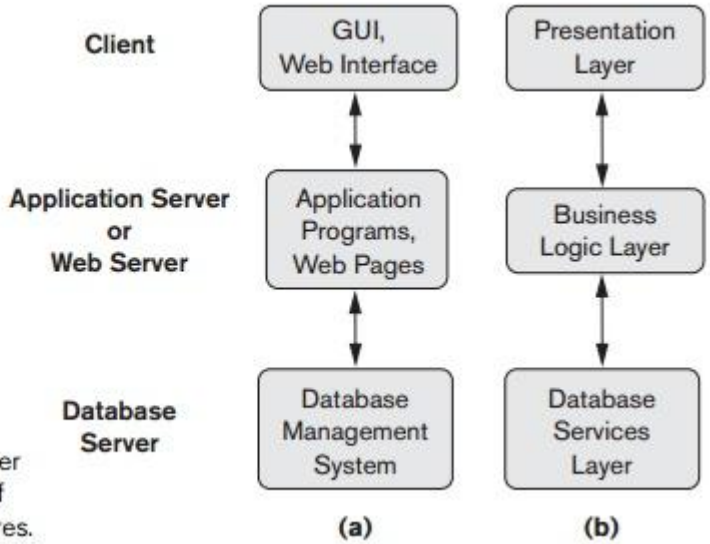


Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

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 run-ning 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, and then acts as a conduit for passing (partially) processed data from the database server to the clients, where it may be processed further and filtered to be presented to users in GUI format. Thus, the *user interface*, *application rules*, and *data access* act as the three tiers.