PRESIDENCY COLLEGE
(AUTONOMOUS)
AFFILIATED TO BENGALURU CITY UNIVERSITY, APPROVED BY AICTE, DELHI & RECOGNISED BY THE GOVT. OF KARNATAKA
RE-ACCREDITED BY NAAC WITH 'A+' GRADE

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

UNIT 1: Introduction to Database Management System

Introduction, Characteristics of the Database Approach, Advantages of Using DBMS Approach. Transaction management and Structure of a DBMS, Database System Concepts and Architecture: Data Models, Schemas, and Instances, The 3-level architecture of DBMS – Hierarchical, Network, and Relational Model and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client-Server Architectures, Classification of Database Management Systems, Queries in DBMS.

INTRODUCTION TO DBMS

Databases and database systems are very important part of real life of modern world. Most of the people are interact with database for their day to day activities. For example people are interacting with bank database whenever they are using ATM for money transactions. People are interacting with railway reservation database whenever they are booking and cancelling railway ticket. We interact with database system whenever we purchase something on online, airline reservation, hotel reservation, college admission etc.

Database system plays a great role on the growing use of computers. It play a major role in almost all the fields in which computers are used like business, e-commerce, medicine, engineering, law, education, library science, agriculture, aeronautics, arts etc. Let us discuss some of the common terms used in database.

BASICS OF DATABASE

Q. What is Data, Database and Database Management system?

Data: Data is known facts or any raw facts, figures, statistics that can be recorded. Data may be in the form of numbers, characters, multimedia pictures, sound messages etc. (e.g. 1, ABC, 19 etc).

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

Information: The processed data is called information. : if we organize the above data (e.g. 1, ABC, 19 etc). in the following way, then they collectively represent meaningful.

Example

Roll	Name	Age
1	ABC	19

Database: A database is a **collection of logically related data** that can be recorded and have an implicit meaning.

- It is a collection of interrelated data.
- These can be stored in the form of tables.
- A database can be of any size and varying complexity.
- A database may be generated and manipulated manually or it may be computerized.

Ex: Student database, employee database etc.

1. employee database : (only one table)
    a. EMP (EmpNo: int, name: string, dob: date, PhNo: int)

2. Student database: (Contains 4 Tables)
    a. Student (snum: integer, sname: string, major: string, level: string, age: integer)
    b. Class (name: string, meets at: string, room: string, fid: integer)
    c. Enrolled (snum: integer, cname: string)
    d. Faculty (fid: integer, fname: string, deptid: integer)

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

Data Base Management System (DBMS): It is a **collection of programs that enables user to create and maintain a database.** In other words it is general-purpose software that provides the users with the processes of **defining, constructing, maintaining and manipulating the database for various applications.** The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

EX: UNIVERSITY database for maintaining information concerning students, courses, and grades in a university environment.

Database Applications:

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

Q. Explain the applications of Database

- *Sales*: For customer, product, and purchase information.
- *Accounting*: For payments, receipts, account balances, assets and other accounting information.
- *Human resources*: For information about employees, salaries, payroll taxes, and benefits, and for generation of paychecks.
- *Manufacturing*: For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores, and orders for items.
- *Online retailers*: For sales data noted above plus online order tracking, generation of recommendation lists, and maintenance of online product evaluations.
- *Banking*: For customer information, accounts, loans, and banking transactions.

- *Credit card transactions*: For purchases on credit cards and generation of monthly statements.

- *Finance*: For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds; also for storing real-time market data to enable online trading by customers and automated trading by the firm.

- *Universities*: For student information, course registrations, and grades (in addition to standard enterprise information such as human resources and accounting).

- *Airlines*: For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner.

Q. Explain briefly the properties of the database?

Properties of the database

- A database represents some aspect of real world, sometimes called mini world or the Universe of Discourse (UoD).
- A database is a logically coherent collection of data with some inherent meaning.
- A database is designed, built and populated with data for a specific purpose. It has an intended group of users.
- A database can be of any size and of varying complexity
- A database may be generated and manipulated manually or it may be computerized

Q. Explain briefly the different types of database users or Actors?

Data Base Users (Different people behind DBMS): or Type of Actors and Workers behind the Scene

There are <u>two categories of people behind DBMS</u>

a) ***Actors on the Scene.***
b) ***Workers Behind the Scene*****.**

a)There are four **USERS or Actors** on the scene of Database Management System
Actor on the scene: who actually use and control the content of the database

- 1.Database Administrator (DBA)
- 2. Database designer
- 3. End users :
    a) Naïve user or parametric end users
    b) Sophisticated user
    c) Stand-alone users
- 4. Software Engineer
    - System Analyst
    - Application Programmer

1. **Database Administrator (DBA**): In a database environment, the primary resource is the database itself and the secondary resource is the DBMS and related software.

- Authorizing access to the database.
- Coordinating and monitoring its use.
- Acquiring software and hardware resources as needed.

2. **Database designer**:

- Identifying the data to be stored in the database
- Choosing appropriate structures to represent and store this data undertaken before the database is actually implemented and populated with data.
- communicate with all prospective database users, in order to understand their requirements
- Develop a view of the database that meets the data and processing requirements for each group of users.

- These views are then analyzed and integrated with the views of other user groups. The final Database design must be capable of supporting the requirements of all user groups.

3**. End users:**

- Access to the database for querying, updating, and generating reports.
- Casual end users.
- Occasionally access the database.
- Need different information each time.
- Learn only a few facilities that they may use repeatedly.

a. Naïve user or parametric end users :

- Constantly querying and updating the database, using standard types of queries and updates called canned transactions that have been carefully programmed and tested
- Need to learn very little about the facilities provided by the DBMS
- Bank tellers check account balances and post withdrawals and deposits
- Reservation clerks for airlines, hotels, and car rental companies check availability for a given
- Request and make reservations.
- Clerks at receiving stations for courier mail enter package identifications via bar codes and descriptive information through buttons to update a central database of received and in transit packages.

b. sophisticated user:

- Engineers, scientists, business analysts, and others who thoroughly familiarize themselves with the facilities of the DBMS so as to implement their applications to meet their complex requirements.
- Try to learn most of the DBMS facilities in order to achieve their complex requirements.

c. Stand-alone users

- Maintain personal databases by using ready-made program packages that provide easy-to-use menu- or graphics-based interfaces. An example is the user of a tax package that stores a variety of personal financial data for tax purposes.
- Typically become very proficient in using a specific software package.

4. Software Engineers:
    - System Analysts: Determine the requirements of end users, especially naive and parametric end users, and develop specifications for canned transactions that meet these requirements.
    - Application programmers: implement these specifications as programs; then they test, debug, document, and maintain these canned transactions.

System Analyst and Application programmers are generally referred to as **Software Engineers**.

Q.What are the three types of workers involved in the database?

b. Workers behind the scene: Persons who enable the database to be developed and the DBMS software to be designed and implemented.

- DBMS system designers and implementers
- Tool developers
- Operators and maintenance personnel.

DBMS system designers and implementers: Are persons who design and implement the DBMS modules and interfaces as a software package.

Tool developers: Include persons who design and implement tools. Tools are optional packages that are often purchased separately.

Operators and maintenance personnel: Are the system administration personnel, who are responsible for the actual running and maintenance of the hardware and software environment for the database system.

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

Q. Discuss the main characteristics of the database approach and how it is better from traditional file system?

Characteristics of Database Approach:

Self-describing nature of a database system

- A DBMS catalog stores the description of a particular database (e.g. data structures, types, and constraints)
- The description is called meta-data.
- This allows the DBMS software to work with different database applications.

Insulation between programs and data / Program-data independence:

- Program-data independence.
- Allows changing data structures and storage organization without having to change the DBMS access programs.

Data Abstraction:

- A data model is used to hide storage and implementation details and present the users with a conceptual view of the database.
- Programs refer to the data model constructs rather than data storage details

Support of multiple views of the data:

- Each user may see a different view of the database, which describes only the data of interest to that user.

Sharing of data and multi-user transaction processing:

- Allowing a set of concurrent users to retrieve from and to update the database.
- Concurrency control within the DBMS guarantees that each transaction is correctly executed or aborted
- Recovery subsystem ensures each completed transaction has its effect permanently recorded in the database

- OLTP (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second correctly and efficiently.

Security of data:

- In a centralized database system, the privilege of modifying the database is not given to everyone. This is given only to database administrator.
- The DBA has full control over the database and he can implement security by placing restrictions on the data. Based on the permissions granted to them, the user can add, modify or query data.

For example, in a banking system, payroll personnel need to see only that part of the database that has information about the various bank employees. They do not need access to information about the customer account.

The main disadvantages in the traditional file oriented approach are:

- **Data Redundancy:** The same data is stored in two or more files.
- **Program / data dependency:** Any changes to the structure of a file may require changing all programs that access this file.
- **Lack of flexibility:** The retrieval possibilities are limited.
- **Data isolation:** In computer **file**-based **system**, **data** is isolated in separate **files**. It is difficult to update and to access particular information from **data files**.

Q. Explain traditional file system and its limitations

File oriented approach/ Traditional Approach:

In traditional file processing approach,

- Data definition is part of the application program and work with only one specific application.

- Each user defines and implements the files needed for a specific application as part of programming the application.

The limitations of file systems are described below: Traditional file systems are design driven; they require design change when new kind of data occurs.
For Example, A traditional customer master file having customer name, address, city and state. If we want to add one more column [field] fax number then it requires a complete restructuring of the file or redesign of the application code.

- In traditional file processing, every user group maintains its own files for handling its data processing applications.
- For example, one user, at the grade reporting office, may keep a file on students and their grades. Programs to print a student transcript and to enter new grades into the file are implemented.
- A second user, at the accounting office, may keep track of student's fees and their payments. Although both users are interested in data about students, each user maintains separate files-programs to manipulate these files because each requires some data not available from the other user's files.
- This redundancy in defining and storing data results in wasted storage space and in redundant effort [man power] to maintain common data up-to date.

Q. Roles and responsibility of DBA

- The role of DBA is very important and is defined by the following functions.
  - *Defining the schema*: The DBA defines the schema which contains the structure of the data in the application. The DBA determines what data needs to be present in the system and how this data has to be presented and organized.
  - *Liaising with users*: The DBA needs to interact continuously with the users to understand the data in the system and its use.

- *Defining Security & Integrity checks*: The DBA finds about the access restrictions to be defined and defines security checks accordingly. Data Integrity checks are defined by the DBA.
- *Defining Backup/Recovery Procedures*: The DBA also defines procedures for backup and recovery. Defining backup procedure includes specifying what data is to be backed up, the periodicity of taking backups and the medium and storage place to backup data.
- *Monitoring performance*: The DBA has to continuously monitor the performance of the queries and take the measures to optimize all the queries in the application.

Q. What are the functions of DBMS?

Functions of DBMS

Data Definition: The DBMS provides functions to define and structure of the data in the applications.
Data Manipulation: Once the data structured, data needs to be inserted, modified or designed. The functions which perform these operations are also part of the DBMS.
Data security and integrity: the DBMS contains functions which handle the security and integrity of the data in the applications.
Data recovery and concurrency: Recovery of data after a system failure are handled by the DBMS concurrent access of records by the multiple users are also handled by the DMBS.
Data Dictionary maintenance: Maintenance of the data dictionary which contains the data definition of the application is also one of the functions of a DBMS.
Performance: optimizing the performance of the queries is one of the important functions of the DBMS.

Q. What are the implication of the database approach? / How do you conclude DBMS approach

Implication of the database approach:

Potential for enforcing standards: The database approach permits the DBA to define and enforce standards among database users in large organization.

- **Reduced application development time:** Developing new applications such as the retrieval of certain data from the database for printing a new report takes very little time.
- **Flexibility:** DBMS allow certain types of changes to the structure of the database without affecting the stored data and the existing application programs.
- **Availability of up-to-date information:** A DBMS makes the database available to all users. As soon as one user's update is applied to the database, all users can immediately see this update.
- **Economies of Scale:** The DBMS approach permits consolidation of data and applications, thus reducing the amount of wasteful overlap between activities of data-processing personnel in different projects or departments.

Q. What are the advantages of using DBMS briefly explain all the advantages?

Advantages or capabilities of using DBMS:

- **Data independence:** Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an *abstract view of the data to insulate application code from such details*.

- **Efficient data access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently.

- **Data integrity and security:** If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. For example, before inserting

- salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce *access controls* that govern what data is visible to different classes of users.

- **Restricting unauthorized access** : When several users share the data, centralizing the administration of data can offer significant improvements.
  - Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.

- **Concurrent access**: A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time.

- **Providing backup and recovery:** A DBMS must provide facilities for recovering from hardware and software failures. The backup recovery subsystem of the DBMS is responsible for recovery. For example, if the computer system fails in the middle of a complex update program, the recovery subsystem is responsible, for making sure that the database is restored to the state it was in before the program started executing. The recovery subsystem could ensure that the program is resumed from the point at which it was interrupted so that its full effect is recorded in the database.

- **Reduced application development time:** Clearly, the DBMS supports many important functions that are common to many applications accessing data stored.

- **Controlling redundancy:** Redundancy is storing the same data multiple times leads to many problems. In the database approach, entire data is stored in a single location in a single database. Data redundancy can be avoided by using different constraints on the data. This doesn't permit any inconsistency and it saves storage space and manpower.

- **Permitting inference and actions using rules:** Database systems provide capabilities for defining deduction rules for inference new information from the stored database facts. Such systems are called deductive database system.

- **Permitting Multiple user interface:** Many types of users with varying levels of technical knowledge use a database. Hence a database should provide a variety of user interfaces. These include
  -query language for casual user
  -programming language interfaces for application programmers
  -forms and command for parametric users
  -menu driven interfaces and natural language interfaces for standalone users

Q. When not to use a DBMS?

There are a few situations in which DBMS system may involve unnecessary overhead costs. The overhead costs of using a DBMS are due to the following:

a) **Main inhibitors (costs) of using a DBMS**:
   i) **High initial investment** and possible **need for additional hardware**.
   ii) **Overhead for providing** generality, security, concurrency control, recovery, and integrity functions.

b) **When a DBMS may be unnecessary**:
   i) If the database and applications are simple, well defined and not expected to change.
   ii) If there are stringent real-time requirements that may not be met because of DBMS overhead.
   iii)If access to data by multiple users is not required.

c) **When no DBMS may be sufficient**:
   i) If the database system is **not able to handle the complexity of data because of modeling limitations**
   ii) If the database users need special operations not supported by the DBMS.

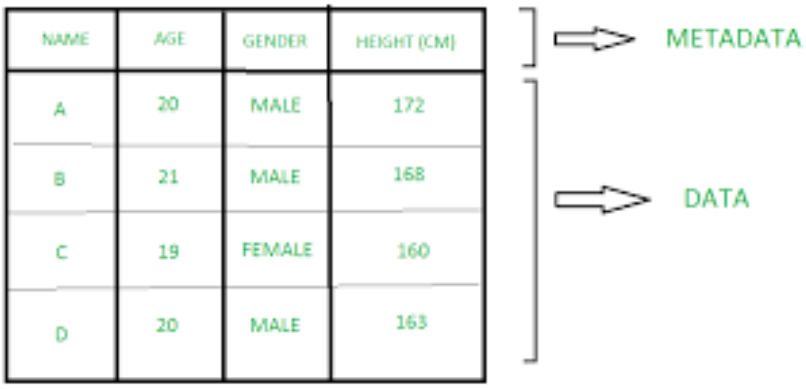
DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

DBMS Terminology:

Database system: Database and DBMS software together called as database system. If not only contain database but also a complete definition or description of the database structure and constraints. This definition stored in what is called DBMS catalog.

DBMS catalog: The catalog contains information such as the structure of each file, the type and storage format of each data item and various constraints on the data.

Metadata: The information such as structure of each file, the type and storage format of each data items and constraints on the data stored in DBMS catalos is called metadata. It describes the structure of the primary database. It contains **data about data**.



- Data dictionary:A **Data Dictionary** is a collection of names, definitions, and attributes about **data** elements that are being used or captured in a **database**,
  - A **Data Dictionary** also provides metadata about **data** elements.

Data Dictionary

Data Dictionary outlining a Database on Driver Details in NSW

Field Name	Data Type	Data Format	Field Size	Description	Example
License ID	Integer	NNNNNN	6	Unique number ID for all drivers	12345
Surname	Text		20	Surname for Driver	Jones
First Name	Text		20	First Name for Driver	Arnold
Address	Text		50	First Name for Driver	11 Rocky st Como 2233
Phone No.	Text		10	License holders contact number	0400111222
D.O.B	Date / Time	DD/MM/YYYY	10	Drivers Date of Birth	08/05/1956

A field is one item of information in a database record e.g. in an the below table, it is ROLL, NAME & AGE.

Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

A record consists of all the fields about an individual entry in a database e.g. all the details about one student in a school management database.

A file is the complete collection of all the records e.g. the complete collection of all the student records.

Table or **Relation**: Collection of related records. The columns of this relation are called **Fields**, **Attributes** or **Domains**. The rows are called **Tuples** or **Records**.

Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

Q. Database system concepts and architecture

Database Instances: Database change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an instance of the database.
Database Schema: The overall design of the database is called the database schema. A schema is a collection of named objects. Schemas provide a logical classification of objects in the database.
Data model: It is a set of concepts that can be used to describe the structure of a database that is to say data types, relationships and constraints.

DETAIL Database system concepts:

A **data model**: a collection of concepts that can be used to describe the structure of a database provides the necessary means to achieve this abstraction. By *structure of a database* we mean the data types, relationships, and constraints that apply to the data .Most data models also include a set of **basic operations** for specifying retrievals and updates on the database.

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

Schemas and Instances:

The description of the database is called the database schema, which is specified during the database design and is not expected change frequently. Most data models have certain conversions for displaying the schemas as diagrams. A displayed schema is called a schema diagram.

Consider an example for schema diagram:

Student

Regno	Sname	Class	Course	Address

Department

Dept_no	Dept_name	Dept_location

Course

Course_no	Course_name	Course_strength

Employee

Emp_name	Emp_no	Salary	Address	Dob

Instances: The collection of data stored in the database at a particular moment in time called a database instance or database state or snapshot. The actual data in a database may change frequently. For example, if we add a new student record or if we delete a record, then the database changes from one state to another state. When a new database is created its state is empty.

In a

Regno	Sname	Class	Course	Address
1001	Smith	II SEM	BCA	M.G.Road,Bangalore

figure, you see the relation student containing one student record. If a record is added to student table, then the state

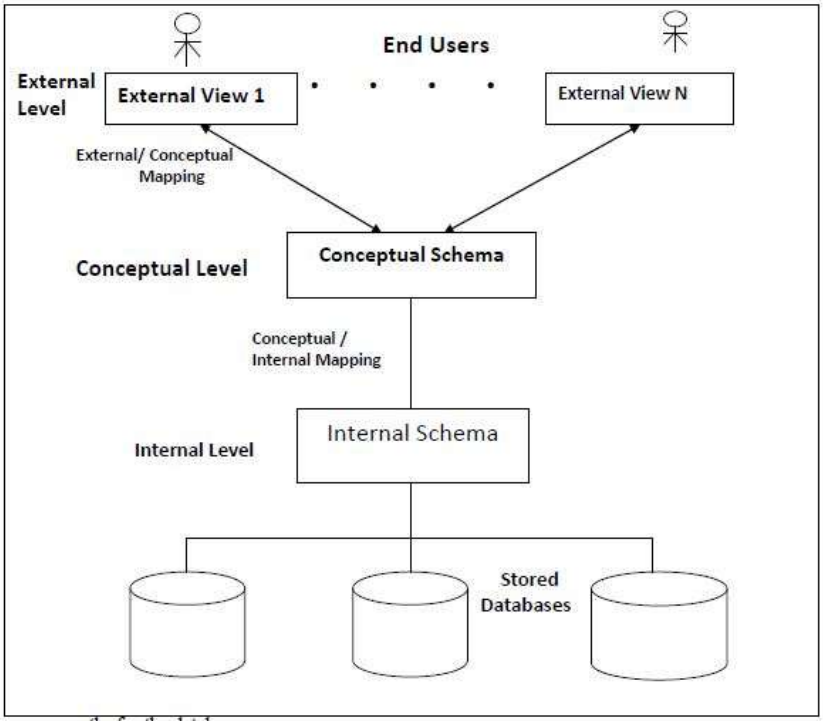
of the table changed from one to two records. This implies that the instance of the database may change time to time and it I also called as database state.

When a new database is created its state is empty. This means that there is no data in the database. However new records are inserted it's no longer empty.

Q. Explain the three schema architecture is also called ANSI/SPARC architecture or three-level architecture.

- A commonly used view of data approach is the three-level architecture suggested by the ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee). ANSI/SPARC proposed an architectural framework for databases.

The three schema architecture of database:



1. The **External Schema** is the view that the individual user of the database has. This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users.

A **view** is a relation that collects certain information from a set of base relations and separate view relation.

2. The **Conceptual schema** (sometimes called the logical schema) describes the stored data in terms of the data model of the DBMS. In a relational DBMS, the conceptual schema describes all relations that are stored in the database. It hides physical storage details, concentrating upon describing entities, data types, relationships, user operations, and constraints.
3. The **physical schema** specifies additional storage details. Essentially, the physical schema summarizes how the relations described in the conceptual schema are actually stored on secondary storage devices such as disks and tapes. It tells us what data is stored in the database and how.

The DBMS must transform a request specified by the external schema into a request against a conceptual schema and then into a request on the internal schema for processing over the stored database.

The process of transforming requests and result between the three levels of schema is known as mapping.

- **1. External / conceptual mapping:** The mapping between external and conceptual level gives the correspondence between records and relationships of external and conceptual levels.
- **2. Conceptual/ Internal mapping:** The mapping between conceptual and internal levels gives the correspondence between the structure [metadata] and size and position of the data in the stored database.

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

Data Independence:

There are two types of data independences:

1. Logical data independence
2. Physical data independence

Logical data independence: It is 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) or to reduce the database (by removing a record or data item) should not affect the external schema.

Example: Adding a phone number column to employee table. Only the view definition and the mappings are required to be changed in logical data independence. Application programs that reference this external schema construct must work as before, after changing the conceptual schema.

Physical data independence: It is capacity to change the internal schema without having to change conceptual or external schema. Some times to improve the performance of retrieval of data. We may need to change the internal storage structure or access path (Say some files need to be re-organized for efficient retrieval), than external or conceptual schema need not be changed.

Database languages and Interfaces

The DBMS provide appropriate language for each category of users. Some of the DBMS languages are,

Data definition language (DDL): This is used by the database Administrators [DBA's] and by the database designers to define schema objects. The DBMS will have a DDL compiler whose functions is to process DDL statements and generates schema objects [tables, indexes etc]. Which are stored in special file called data dictionary [system catalog]. The DDL is used to specify the conceptual schema only.

E.g. CREATE, DROP, TRUNCATE and ALTER Statements

- **Stored definition language (SDL):** It is used to specify internal schema.
- **View definition language (VDL):** It is used to specify views and their mappings to conceptual schema.

Data manipulation languages (DML): Once the database schemas are compiled, database can be manipulated using the language called DML.

Manipulation of database may include operations such as,

- Retrieval of data
- Insertion of data
- Deleting of data
- Modification of data

E.g. SELECT, DELETE, INSERT and UPDATE statements

Data Control Language (DCL): It is used to manage different transactions occurring within a database.
E.g. GRANT and REVOKE statements

Transaction Control Language (TCL): It is used to manage different transactions occurring within a database.
E.g. COMMIT ROLLBACK and SAV EPOINT statements.

DBMS Interfaces

Interfaces are the programs which enable the user to interact with the DBMS.

The different user-friendly interfaces provided by a DBMS may include the followings:

- **Menu based interfaces:** These interfaces presents the user with list of options called menus. Pull down menus are becoming very popular technique in windows based user interfaces; it provides the user to develop their own customized design.

- **Graphical user interfaces:** A GUI typically displays a schema to the user in diagrammatic form. The most GUI uses pointing device, such as mouse to pick certain parts of the displayed schema diagram.

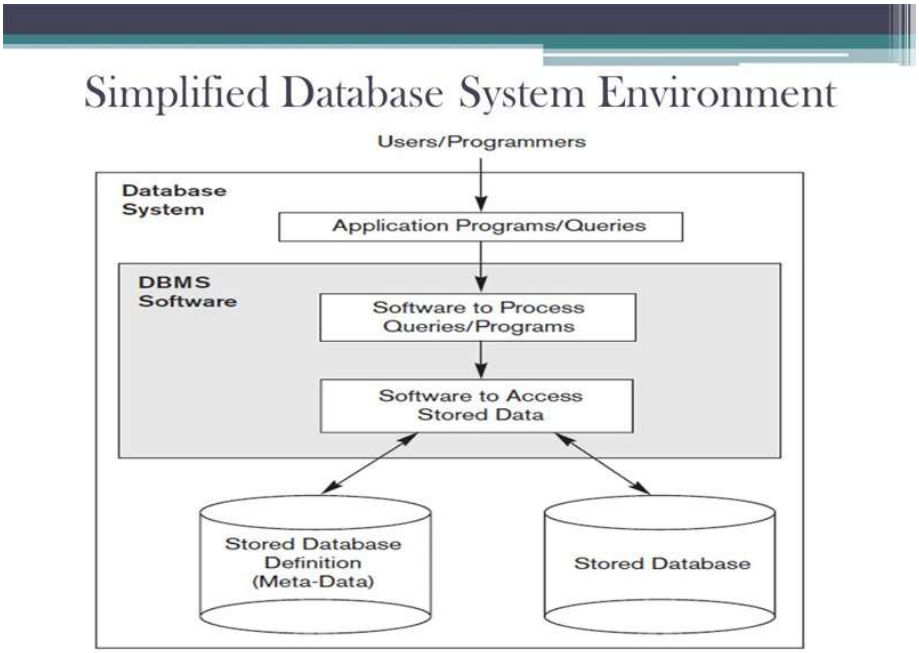
- **Natural language interfaces:** These interfaces accept request written in English or some other language and attempt to "**understand**" them. This interface generates a high-level query corresponding to the natural language request and submits it to the DBMS for processing.

- **Interfaces for parametric users:** Parametric users, such as bank tellers and reservation clerks, who are using only limited operation repeatedly, have usually some function keys to repeat the requests. System analysts design a special interface using function keys for specific class of users.

- **Interface for the Database administrator (DBA):** Most database systems contain privileged commands that can be used only by the staff of DBA. These include commands for creating schema objects, for creating new users, granting accounts authorization. **EX:** Grant, revoke etc.

Q. Explain the concept of Database system environment with a neat diagram?

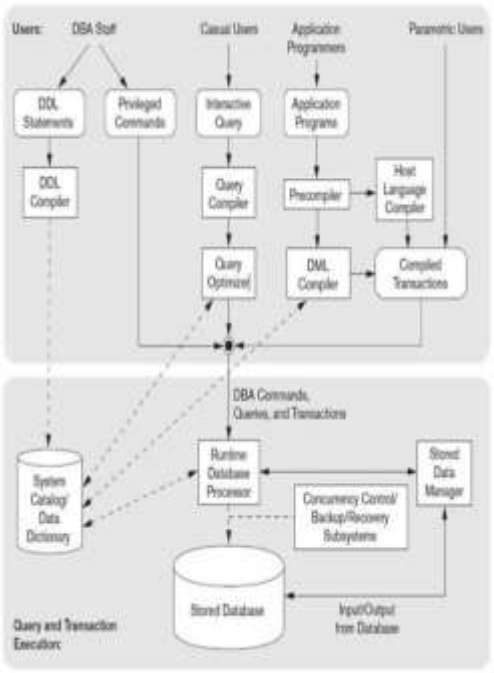
- We can divide the figure into Upper and Lower
  - Upper part – for Users
  - Lower part – For storage



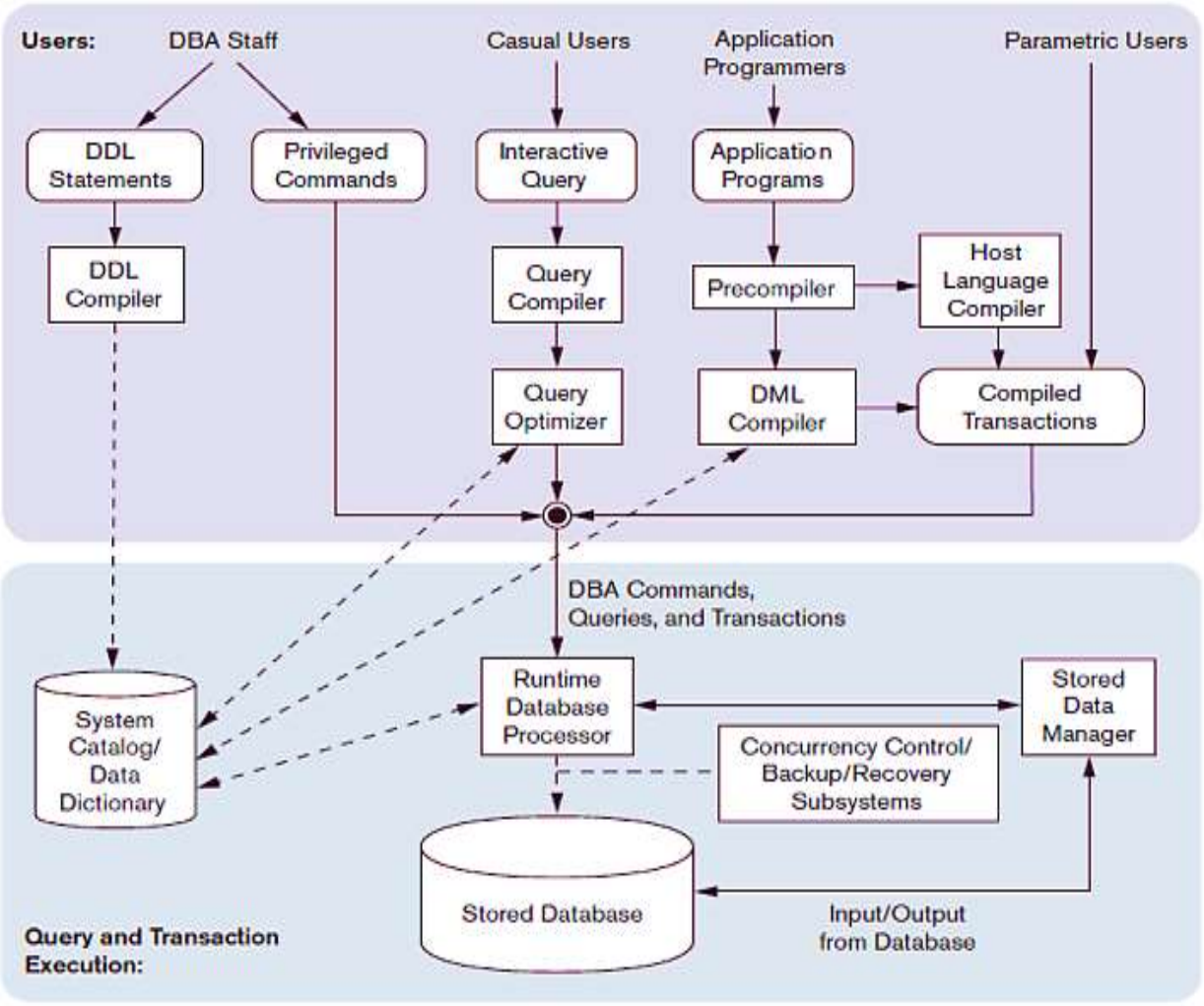
DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

The Database System Environment (typical)

- Upper part:
  - 4 kinds of usage modes
  - Database languages (SQL) and programming languages
- Lower part:
  - Core DBMS functionality
  - Runtime database processor has binary API which deals with actual data



DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021



- **Database system environment:**
- Upper : The top part of Figure,
- DBA STAFF:

It shows interfaces for **the DBA staff**, The DBA staff works on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands.

- **DDL COMPILER**→ The DDL compiler processes schema definitions, specified in the DDL and stores descriptions of the schemas (meta-data) in the DBMS catalog.
- The catalog includes information such as the names and sizes of files, names and data types of data items, storage details of each file, mapping information among schemas, and constraints.

- **casual users** who work with interactive interfaces to formulate queries,
  - **interactive query interface**: Casual users and persons with occasional need for information from the database interact using some form of interface
  - **QUERY COMPILER**: These queries are parsed and validated for correctness of the query syntax, the names of files and a query compiler that compiles them into an internal form.
  -

- **application programmers** who create programs using some host programming languages

  - The **Precompiler** extracts DML commands from an application program written in a host programming language. These commands are sent to the **DML compiler**
  - **DML compiler** for compilation into object code for database access. The rest of the program is sent to the host language compiler.
  - The object codes for the DML commands and the rest of the program are linked, forming a **canned transaction**

- **parametric users** who do data entry work by supplying parameters to predefined transactions.

- The object codes for the DML commands and the rest of the program are linked, forming a **canned transaction** whose executable code includes calls to the runtime database processor.

Lower part: **Database System Utilities and components**

- **RUNTIME DATABASE PROCESSOR:** handles database access at runtime
- **Stored data manager:** data transfer between disk and main memory

- In addition to possessing the software modules just described, most DBMSs have database utilities that help the DBA manage the database system.
- Common utilities have the following types of functions:
  - Loading. A loading utility is used to load existing data files—such as text files or sequential files—into the database. Usually, the current (source) format of the data file and the desired (target) database file structure are specified to the utility, which then automatically reformats the data and stores it in the database.
  - A backup utility creates a backup copy of the database, usually by dumping the entire database onto tape or other mass storage medium. The backup copy can be used to restore the database in case of disk failure.
  - Database storage reorganization. This utility can be used to reorganize a set of database files into different file organizations, and create new access paths to improve performance.
  - Performance monitoring. Such a utility monitors database usage and provides statistics to the DBA. The DBA uses the statistics in making decisions such as whether or not to reorganize files or whether to add or drop indexes to improve performance.

- Other utilities may be available for sorting files, handling data compression, monitoring access by users, interfacing with the network, and performing other functions.

Classification of Database Management Systems

The main classification - data model

- Relational data model (SQL systems)
- Object data model
- Document-based (JSON), graph-based, column-based, and key-value data models
- Legacy applications still work databases based on the hierarchical and network (COBOL) data models
- Experimental DBMSs are based on a tree-structured data model (XML)

Number of users:

- Single-user at a time
- Multiuser

Number of sites:

- Centralized
- Distributed (big data): homogeneous vs. heterogeneous (the same/different DBMS)

Cost:

- Free (MySQL, PostgreSQL)
- Commercial

Purpose – types of access path:

- General purpose
- Special purpose (online transaction processing (OLTP) systems)

Q. Define Data model and discuss the categories of data models?

Data model: It is a set of concepts that can be used to describe the structure of a database that is to say data types, relationships and constraints. The most popular high level model is the **entity relationship model.** It is easy to understand by the professionals and non technical users. It easily distinguishes the data, relationships between data and constraints. It also explains the way the organization uses and manages the information.

Categories of data model: Data model are categorized based on the types of concepts that they provide to describe the database structure.

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

1. **High level or conceptual data model**: This model provides the concepts that are close to user view. The concepts such as entity attribute and relationships.

An **entity** represents a real world object or concept such as employee or a project that is stored in a database.
An **attribute** represents of interest that further describes an entity, such as employee's name, age, salary and etc.
A **relationship** among two or more entities represents an association among two or more entities, for example **works on** relationship between an employee and project.

2. **Physical or low level data model**: This model provides the concept that describes the details of how data is stored in the computer. Concepts provided by low level data models are generally meant for computer specialist, not for typical end users.

3. **Representational or implementation data model:** this model is in between these two extremes of the physical model and the conceptual model. This does not hide all the storage details from the user and it can be implemented on a computer system directly. It represents data by using record structures and hence is sometimes called record-based data models.

These include three most popular data models:

1. Relational model : **Relational Model:** The Relational Model uses a collection of tables both data and the relationship among those data. Each table has multiple columns and each column has a unique name.

Relational database comprising of two tables.

Student_Table

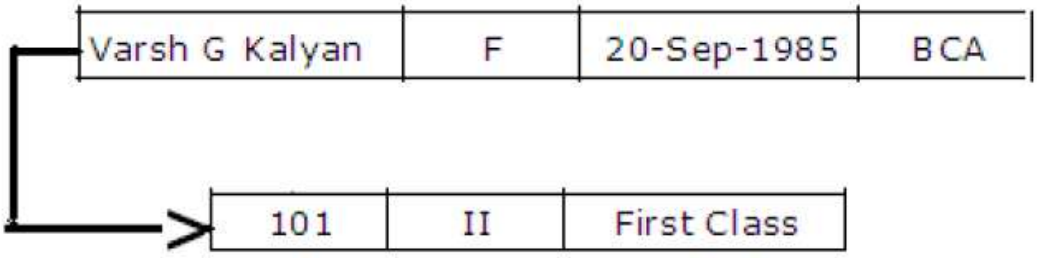
Regno	Name	Gender	DOB	Course
101	Varsh G Kalyan	F	20-Sep-1985	BCA
102	Mohith G Kalyan	M	20-Aug-1980	BBM
103	Nisarga	F	15-Jul-1983	BCom
104	Eenchara	F	04-Dec-1985	BCA

Result_Table

Regno	Sem	Result
101	II	First Class
102	II	First Class
103	II	Passes
104	II	Second Class

2. Network model :

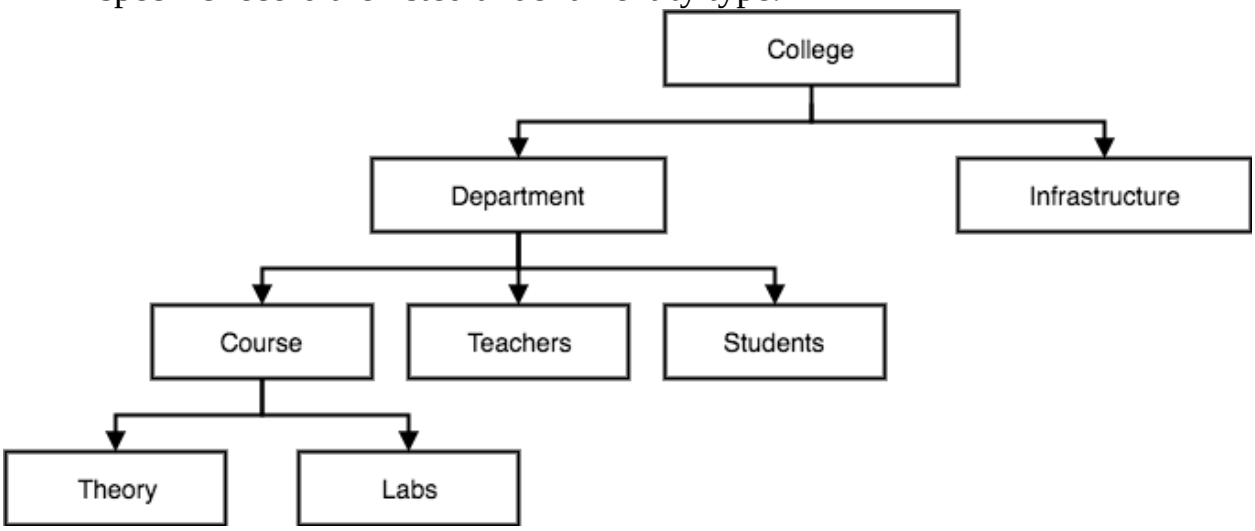
   The data in the network model are represented by collection of records and relationships among data are represented by links, which can be viewed as pointers



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 has one parent. All attributes of a specific record are listed under an entity type.



Q.What is Schema?

Schemas: The description of the database is called the database schema, which is specified during the database design and is not expected change frequently. Most data models have certain conversions for displaying the schemas as diagrams. A displayed schema is called a schema diagram.

Centralized Architecture:

The client/server architecture was developed to deal with computing environments in which a large number of PCs, workstations, file servers, printers, database servers

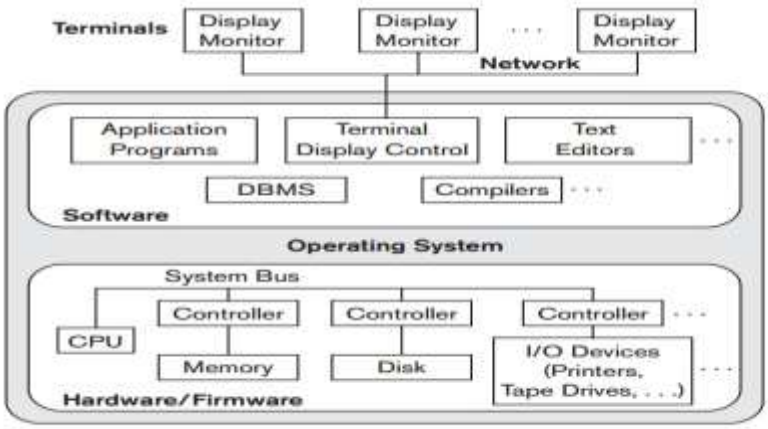


Figure 2.4
A physical centralized architecture.

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.

Client/Server Architecture:

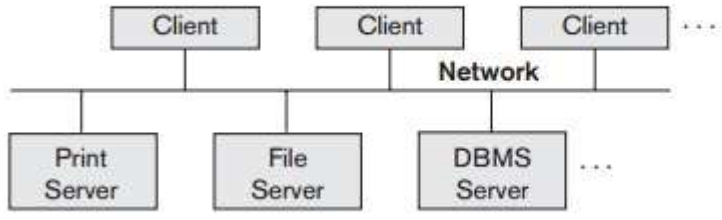


Figure 2.5
Logical two-tier client/server architecture.

Q. What is Database 3-TIER Architecture in DBMS?

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021

A **Database Architecture** is a representation of DBMS design. It helps to design, develop, implement, and maintain the database management system. A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, and altered. It also helps to understand the components of a database.

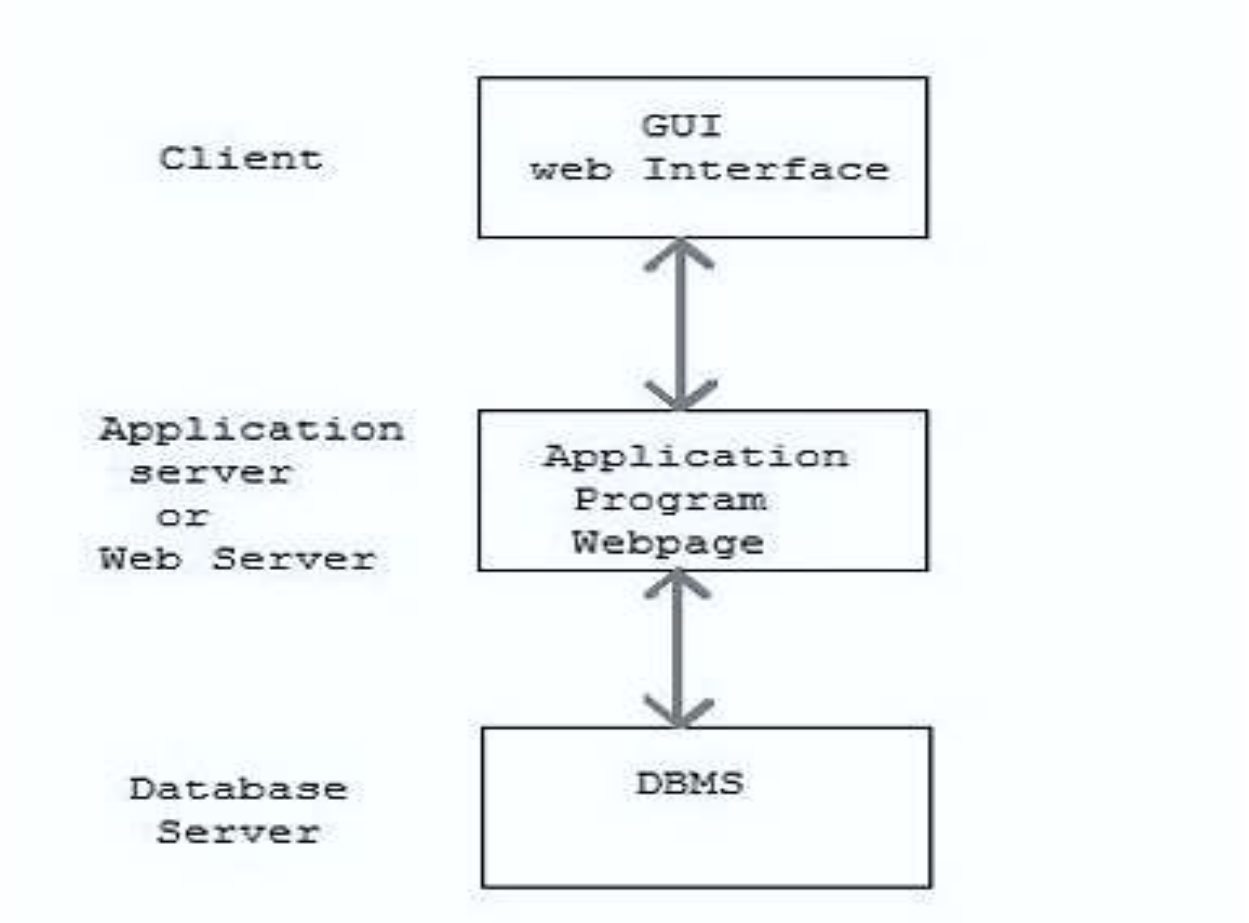
A Database stores critical information and helps access data quickly and securely. Therefore, selecting the correct Architecture of DBMS helps in easy and efficient data management.

There are mainly three types of DBMS architecture:

- One Tier Architecture (Single Tier Architecture)
- Two Tier Architecture
- Three Tier Architecture

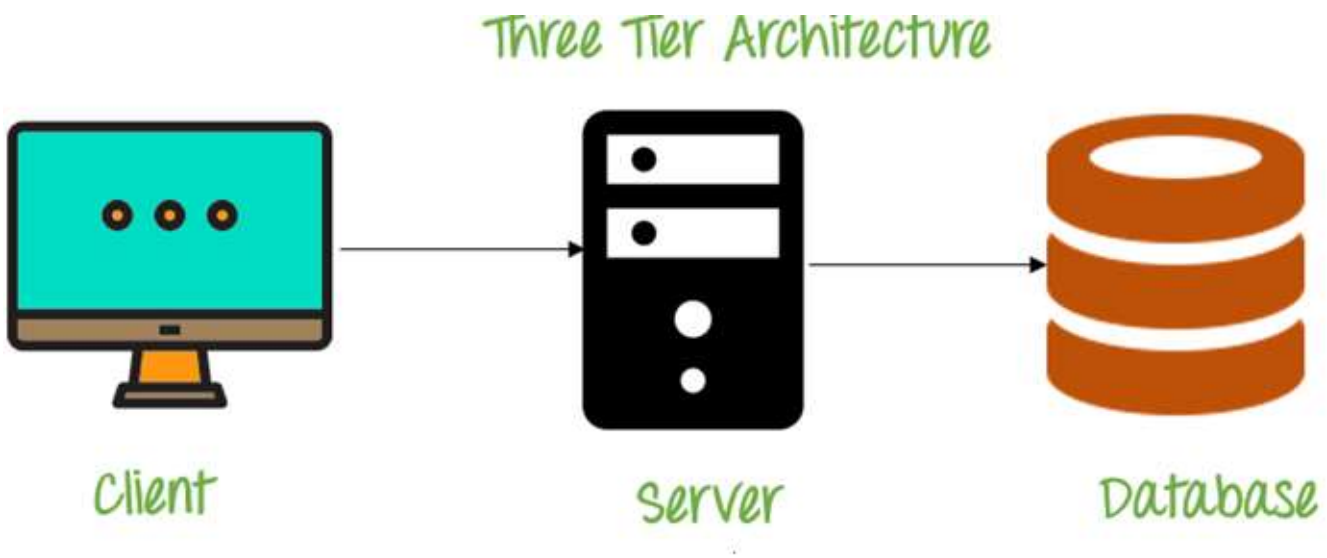
Tier Architecture in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, data access, data storage, and user interface is done independently as separate modules.

DATABASE MANAGEMENT SYSTEM UNIT-1 NOTES-2021



Three Tier architecture contains a presentation layer, an application layer, and a database server. 3-Tier database Architecture design is an extension of the 2-tier client-server architecture. A 3-tier architecture has the following layers:

1. Presentation layer (your PC, Tablet, Mobile, etc.)
2. Application layer (server)
3. Database Server



3 Tier Architecture Diagram

The Application layer resides between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user. The application layer(business logic layer) also processes functional logic, constraint, and rules before passing data to the user or down to the DBMS.

The goal of Three Tier client-server architecture is:

- To separate the user applications and physical database
- To support DBMS characteristics
- Program-data independence
- Supporting multiple views of the data

  Example: book your ticket by using IRCTC WEB PAGE.

Summary of Architecture

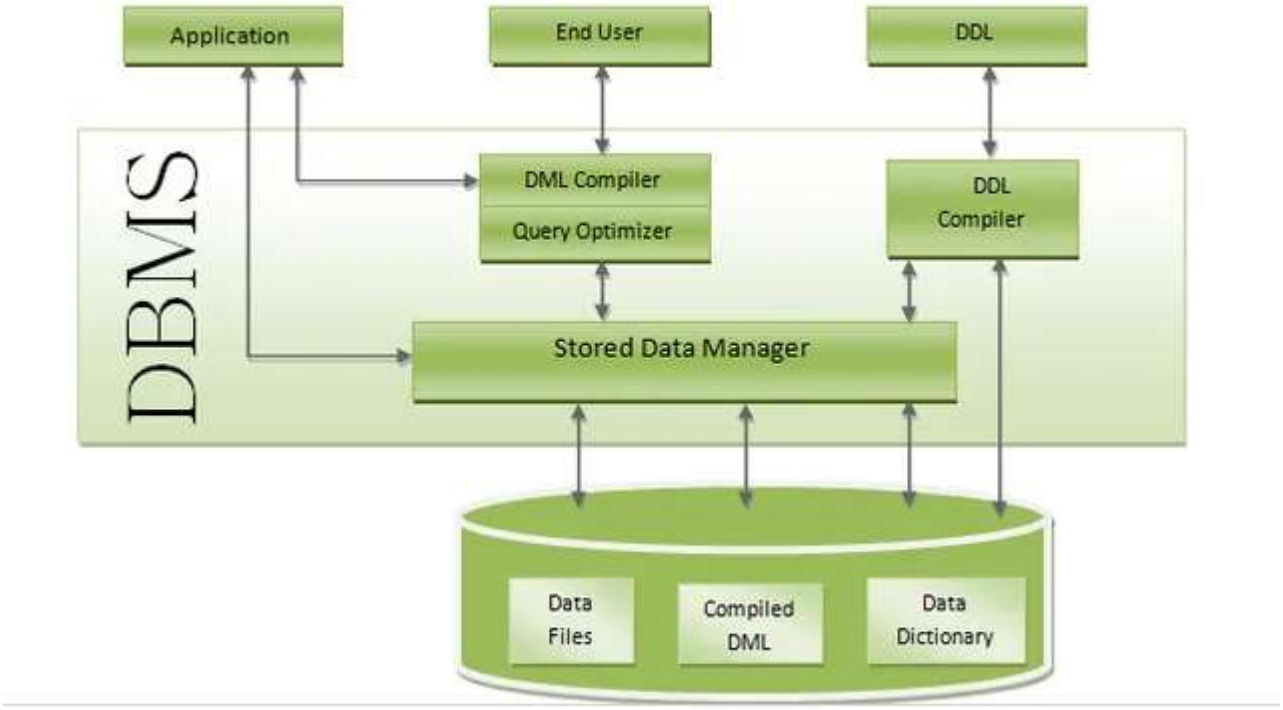
- An Architecture of DBMS helps in design, development, implementation, and maintenance of a database

• CENTRALISED	• The simplest database system architecture is 1 tier where the Client, Server, and Database all reside on the same machine • ALL in One machine
• 2- TIER CLIENT/ SERVER	• A two-tier architecture is a database architecture in DBMS where presentation layer(interface) runs on a client and data is stored on a server
• 3-TIER CLIENT/ SERVER	• Three-tier client-server architecture consists of the Presentation layer (PC, Tablet, Mobile, etc.), Application layer (server) and Database Server

Differentiate between centralized and distributed data base

Centralized	Distributed
Database is maintained at one site	Database is maintained at a number of different sites
If centralized system fails, entire system is halted.	If one system fails, system continues work with other sites
Less reliable	More reliable

Structure of DBMS



Structure of Database Management System (DBMS)

- **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, administrators, 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 objects like tables, indexes in the database for the first time. In other words, they create the structure of the database.

- **DDL Compiler: –** This part of the database is responsible for processing the DDL commands. That means this 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 requests 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: –** The query optimizer decides the best way to execute the user request which is received from the DML compiler.

- **Stored Data Manager: –** This is also known as Database Control System. It is one of the main central systems of the database. It is responsible for various tasks
  - ✓ It converts the requests received from query optimizer to machine-understandable form
  - ✓ It helps to maintain consistency and integrity by applying the constraints.
  - ✓ It controls concurrent access. If there are 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 back up the database and recovers data whenever required. Since it is a huge database and when there is any unexpected exploit of the

transaction, and reverting the changes is 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 are 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 a description of all the tables, view, materialized views, constraints, indexes, triggers, etc.

UNIT1 Important Questions

2 MARKS QUESTIONS

1. Define data, Information, database and DBMS.
2. What is Catalog and metadata?
3. Differentiate centralized-database and client-server database Architecture
4. Define Scheme and instance
5. Write all the Applications of DBMS.
6. Define Data dictionary and data model.

5 MARKS QUESTIONS

7. Explain the characteristics of database approach.
8. Explain briefly different types of database users./ List and explain the end users of database systems./ Explain different workers behind the screen.

9. Explain briefly data independence and its categories
10. Explain briefly the properties and functions of database
11. Explain roles and responsibilities of DBA
12. Explain the architecture of DBMS with a neat diagram/ Explain three schema architecture.
13. What are the advantages of DBMS and explain its applications?
14. Write a note on DBMS Environment?
15. Explain briefly the different type of database language
16. Define data model discuss the categories of data models
17. Write a note on interfaces of DBMS
18. Explain the main criteria for classifying DBMS
19. Explain the client-server architecture for Database in DBMS.
20. Justify how Database approach is better than traditional file system.

FAQ's:

1. Explain different functions of DBMS.
2. Explain the overall structure of a simplified database system environment with a diagram.
3. Explain the characteristics of database approach.
4. Explain the advantages of DBMS?
5. List and explain different database users.
6. Explain different types of database interfaces.
7. Explain three schema architecture with neat diagram.
8. Explain data schema and instances with example.
9. Explain DBMS architecture with neat diagram.
10. Explain centralized and client server architecture.
11. Explain the Structure of DBMS.