

UNIT 1- Database System Concepts & Architecture

Database

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Using the database, you can easily retrieve, insert, and delete the information.

Database Management System

- Database management system is software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

DBMS allows users the following tasks:

- **Data Definition:** It is used for creation, modification, and removal of definition that defines the organization of data in the database.
- **Data Updation:** It is used for the insertion, modification, and deletion of the actual data in the database.
- **Data Retrieval:** It is used to retrieve the data from the database which can be used by applications for various purposes.
- **User Administration:** It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

Characteristics of DBMS

- It uses a digital repository established on a server to store and manage the information.
- It can provide a clear and logical view of the process that manipulates data.

- o DBMS contains automatic backup and recovery procedures.
- o It contains ACID properties which maintain data in a healthy state in case of failure.
- o It can reduce the complex relationship between data.
- o It is used to support manipulation and processing of data.
- o It is used to provide security of data.
- o It can view the database from different viewpoints according to the requirements of the user.

Advantages of DBMS

- o **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
- o **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
- o **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- o **Reduce time:** It reduces development time and maintenance need.
- o **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- o **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS

- o **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
- o **Size:** It occupies a large space of disks and large memory to run them efficiently.
- o **Complexity:** Database system creates additional complexity and requirements.
- o **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

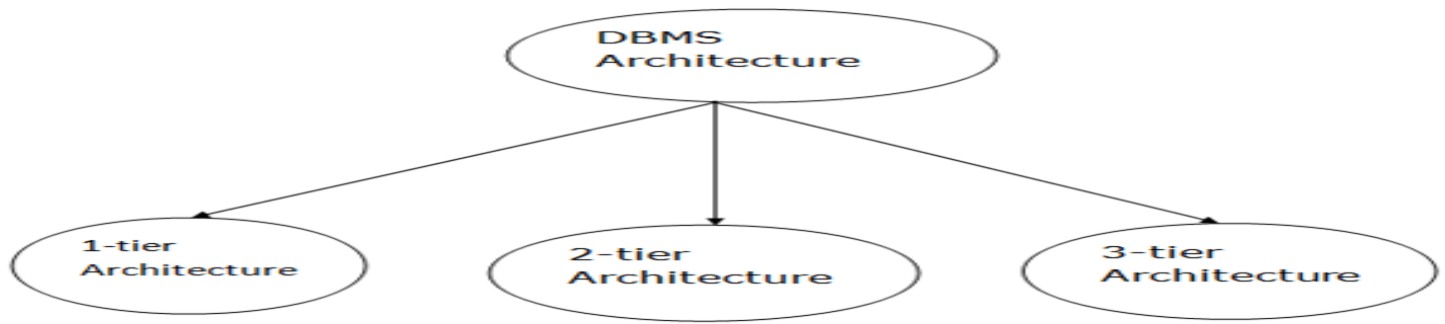
DBMS vs. File System

There are following differences between DBMS and File system:

DBMS	File System
DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	File system is a collection of data. In this system, the user has to write the procedures for managing the database.
DBMS gives an abstract view of data that hides the details.	File system provides the detail of the data representation and storage of data.
DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from the system failure.	File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will lost.
DBMS provides a good protection mechanism.	It is very difficult to protect a file under the file system.
DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	File system can't efficiently store and retrieve the data.
DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while other deleting some information or updating some information.

DBMS Architecture

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

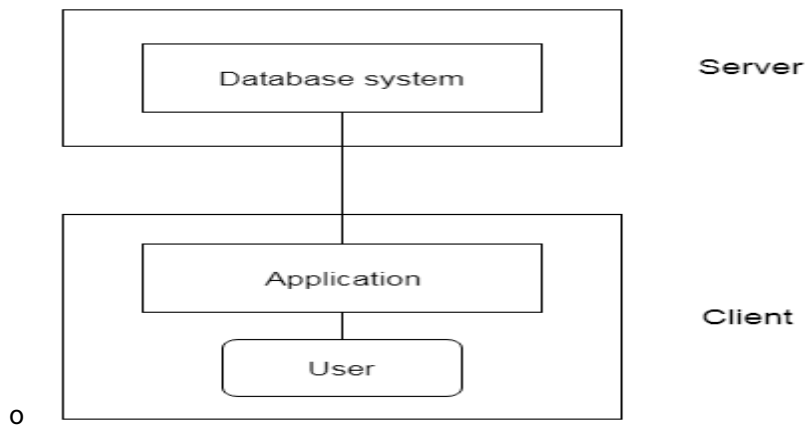


1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

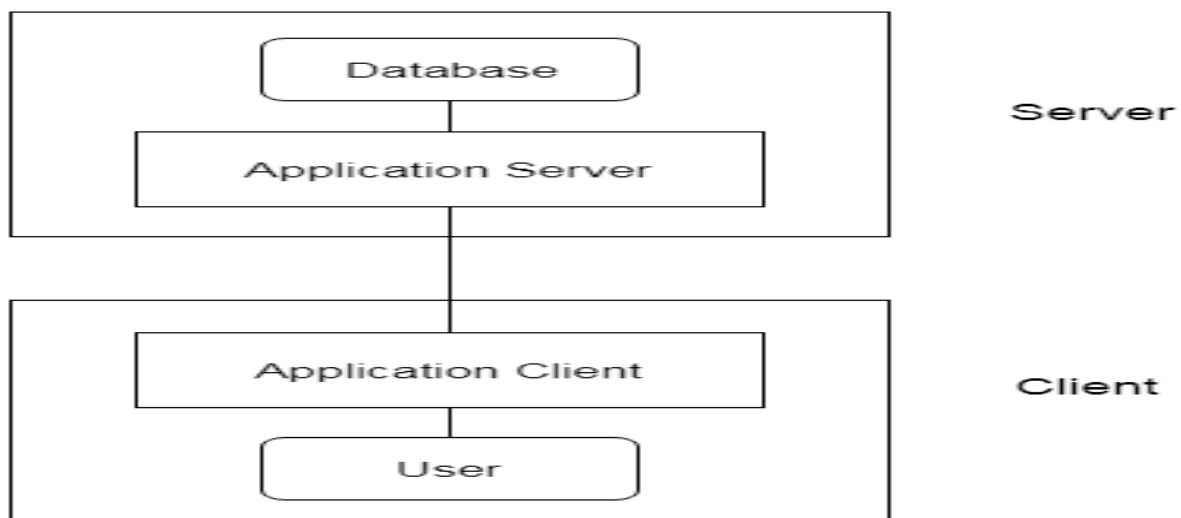
2-Tier Architecture

- o The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
- o The user interfaces and application programs are run on the client-side.
- o The server side is responsible to provide the functionalities like: query processing and transaction management.
- o To communicate with the DBMS, client-side application establishes a connection with the server side.



3-Tier Architecture

- o The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- o The application on the client-end interacts with an application server which further communicates with the database system.
- o End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- o The 3-Tier architecture is used in case of large web application.

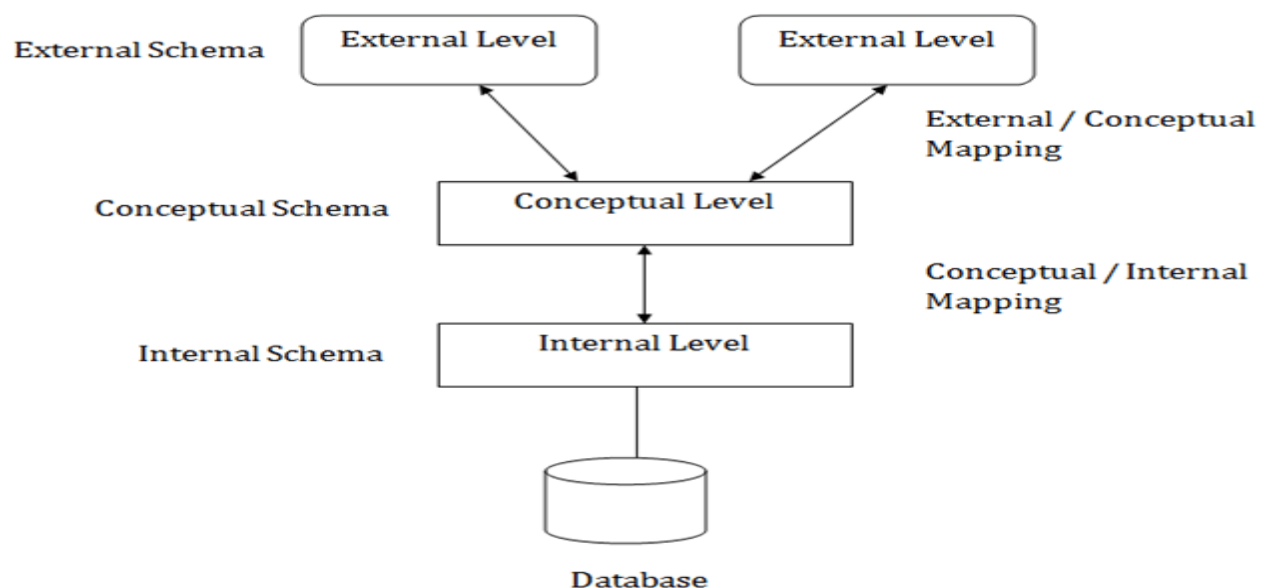


Three schema Architecture

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

In the above diagram:

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



In the above diagram:

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

1. Internal Level

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

2. Conceptual Level

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

3. External Level

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

and hides the remaining database from that user group.

- o The view schema describes the end user interaction with database systems.

Data Independence

- o Data independence can be explained using the three-schema architecture.
- o Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

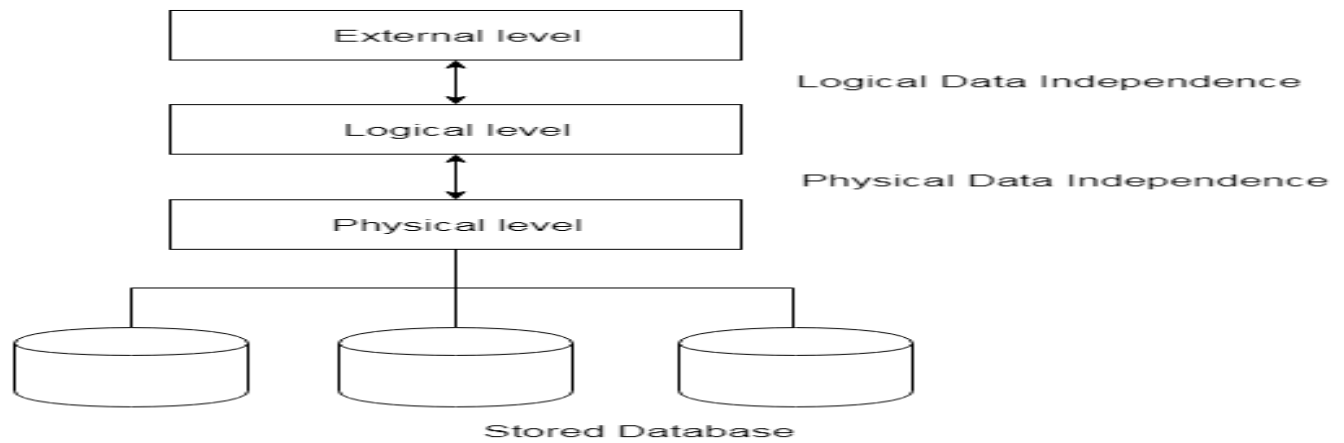
There are two types of data independence:

1. Logical Data Independence

- o Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- o Logical data independence is used to separate the external level from the conceptual view.
- o If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- o Logical data independence occurs at the user interface level.

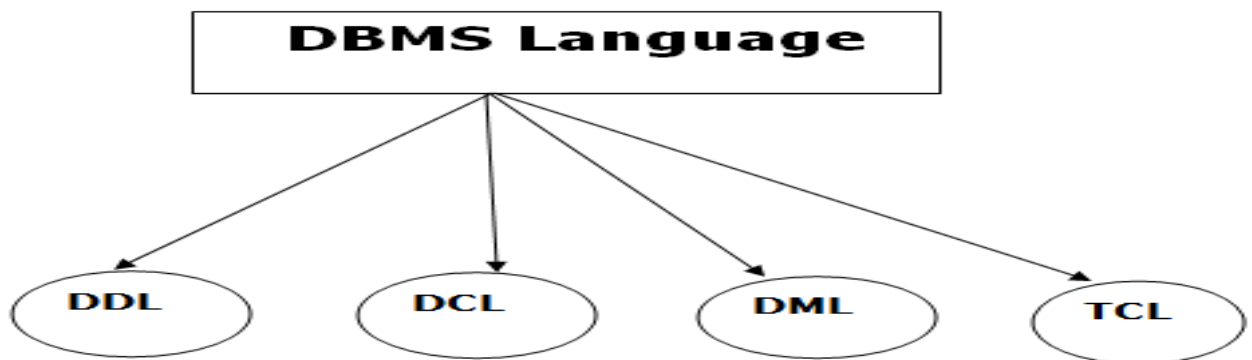
2. Physical Data Independence

- o Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- o If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- o Physical data independence is used to separate conceptual levels from the internal levels.
- o Physical data independence occurs at the logical interface level.



Database Language

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



1. Data Definition Language

- o **DDL** stands for **Data Definition Language**. It is used to define database structure or pattern.
- o It is used to create schema, tables, indexes, constraints, etc. in the database.
- o Using the DDL statements, you can create the skeleton of the database.
- o 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.

DDL commands are:-

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

2. Data Manipulation Language

DML stands for **Data Manipulation Language**. It is used for accessing and manipulating data in a database. It handles user requests.

DML commands are:-

- **Select:** It is used to retrieve data from a database.
- **Insert:** It is used to insert data into a table.
- **Update:** It is used to update existing data within a table.
- **Delete:** It is used to delete all records from a table.
- **Merge:** It performs UPSERT operation, i.e., insert or update operations.
- **Call:** It is used to call a structured query language or a Java subprogram.
- **Explain Plan:** It has the parameter of explaining data.
- **Lock Table:** It controls concurrency.

3. Data Control Language

- **DCL** stands for **Data Control Language**. It is used to retrieve the stored or saved data.
- The DCL execution is transactional. It also has rollback parameters.

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

Here are some tasks that come under DCL:

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

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

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

4. Transaction Control Language

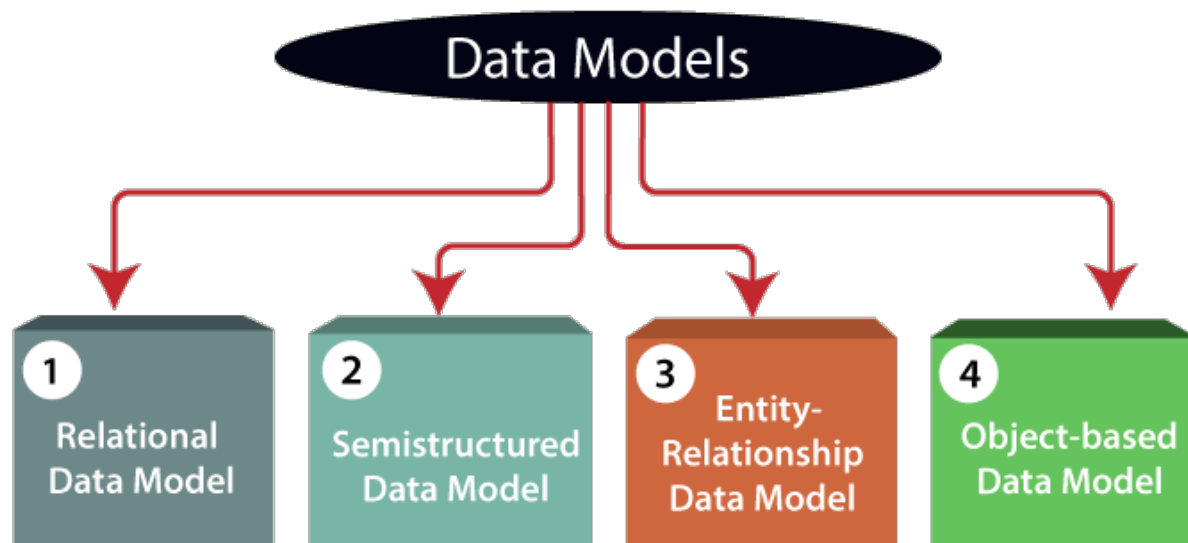
TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

Here are some tasks that come under TCL:

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

Data Models

Data Model is the modeling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction. Therefore, there are following four data models used for understanding the structure of the database:



1) Relational Data Model: This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations. This model was initially described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications.

2) Entity-Relationship Data Model: An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and

relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.

3) Object-based Data Model: An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in 1980s, various database systems following the object-oriented approach were developed.

4) Semistructured Data Model: This type of data model is different from the other three data models (explained above). The semistructured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets. The Extensible Markup Language, also known as XML, is widely used for representing the semistructured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

Data model Schema and Instance

- o The data which is stored in the database at a particular moment of time is called an instance of the database.
- o The overall design of a database is called schema.
- o A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
- o A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
- o A database schema can be represented by using the visual diagram. That diagram shows the database objects and relationship with each other.
- o A database schema is designed by the database designers to help programmers whose software will interact with the database. The process of database creation is called data modeling.

A schema diagram can display only some aspects of a schema like the name of record type, data type, and constraints. Other aspects can't be specified through the schema diagram. For example, the given figure neither show the data type of each data item nor the relationship among various files.

In the database, actual data changes quite frequently. For example, in the given figure, the database changes whenever we add a new grade or add a student. The data at a particular moment of time is called the instance of the database.

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------