# DBMS Unit-01

By:
Mani Butwall
Asst. Prof. (CSE)

# Contents

- Introduction
- Difference between File Oriented and DBMS
- Data Abstraction
- Data Independence
- Data Definition Language (DDL)
- Data Manipulation Language (DML)

# Introduction

- **Database:** Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

# What is DBMS?

- DBMS or Database Management System is a software application used to access, create, and manage databases. With the help of DBMS, you can easily create, retrieve and update data in databases. A DBMS consists of a group of commands to manipulate the database and acts as an interface between the end-users and the database.
- Database Management Systems also aims to facilitate an overview of the databases, by providing a variety of administrative operations such as tuning, performance monitoring, and backup recovery.

Alteration

- Database Management Systems allows users to do the following:

1. **Define Data** – Allows the users to create, modify and delete the definitions which define the organization of the database.
2. **Update Data** – Provides access to the users to insert, modify and delete data from the database.
3. **Retrieve Data** – Allows the users to retrieve data from the database based on the requirement.
4. **Administration of users** – Registers the users and monitors their action, enforces data security, maintains data integrity, monitors performance and deals with concurrency control.

# Characteristics of DBMS

- The following are a few characteristics of DBMS:

1. To limit the permissions of the users
2. Provide multiple views of the single database schema
3. Facilitates security and removes data redundancy
4. Allows multi-user transaction processing and sharing of data
5. Follows the ACID property
6. Offers both physical and logical data independence

# Advantages of DBMS

Few of the advantages of the database management system are as follows:

1. It offers a variety of methods to store and retrieve various formats of data using the query language.
2. It can be *easily maintained* because of its nature of a centralized database system.
3. Facilitates *multiple applications* using the same data with less development and maintenance time.
4. Provides data security and integrity with minimal data duplicity and redundancy.
5. It allows seamless integration into the application programming languages like Java and Python to enable the users to connect a database with any application or website.
6. Has *automatic backup and recovery* systems to create an automatic backup of data.
7. Authorizes users who can view, share and access data.

# Disadvantages of DBMS

1. Databases Management Systems are often complex systems.
2. Few of the DBMS available in the market are licensed. So, you have to pay to us that DBMS in your organization.
3. Most leading companies store their data in a single database. Hence, if that database is damaged due to any reason, the complete data would be lost.
4. DBMS that you wish to use might not be compatible with an organization’s operational requirements.
5. DBMS are large in size and need time to setup.

# Difference between File oriented and DBMS

- **Redundancy of data**: Data is said to be redundant if same data is copied at many places. If a student wants to change Phone number, he has to get it updated at various sections. Similarly, old records must be deleted from all sections representing that student.
- **Inconsistency of Data:** Data is said to be inconsistent if multiple copies of same data does not match with each other. If Phone number is different in Accounts Section and Academics Section, it will be inconsistent. Inconsistency may be because of typing errors or not updating all copies of same data.
- **Difficult Data Access**: A user should know the exact location of file to access data, so the process is very cumbersome and tedious. If user wants to search student hostel allotment number of a student from 10000 unsorted students' records, how difficult it can be.

- **Unauthorized Access:** File System may lead to unauthorized access to data. If a student gets access to file having his marks, he can change it in unauthorized way.
- **No Concurrent Access**: The access of same data by multiple users at same time is known as concurrency. File system does not allow concurrency as data can be accessed by only one user at a time.
- **No Backup and Recovery:** File system does not incorporate any backup and recovery of data if a file is lost or corrupted.

# Popular DBMS Software

- Here, is the list of some popular DBMS system:

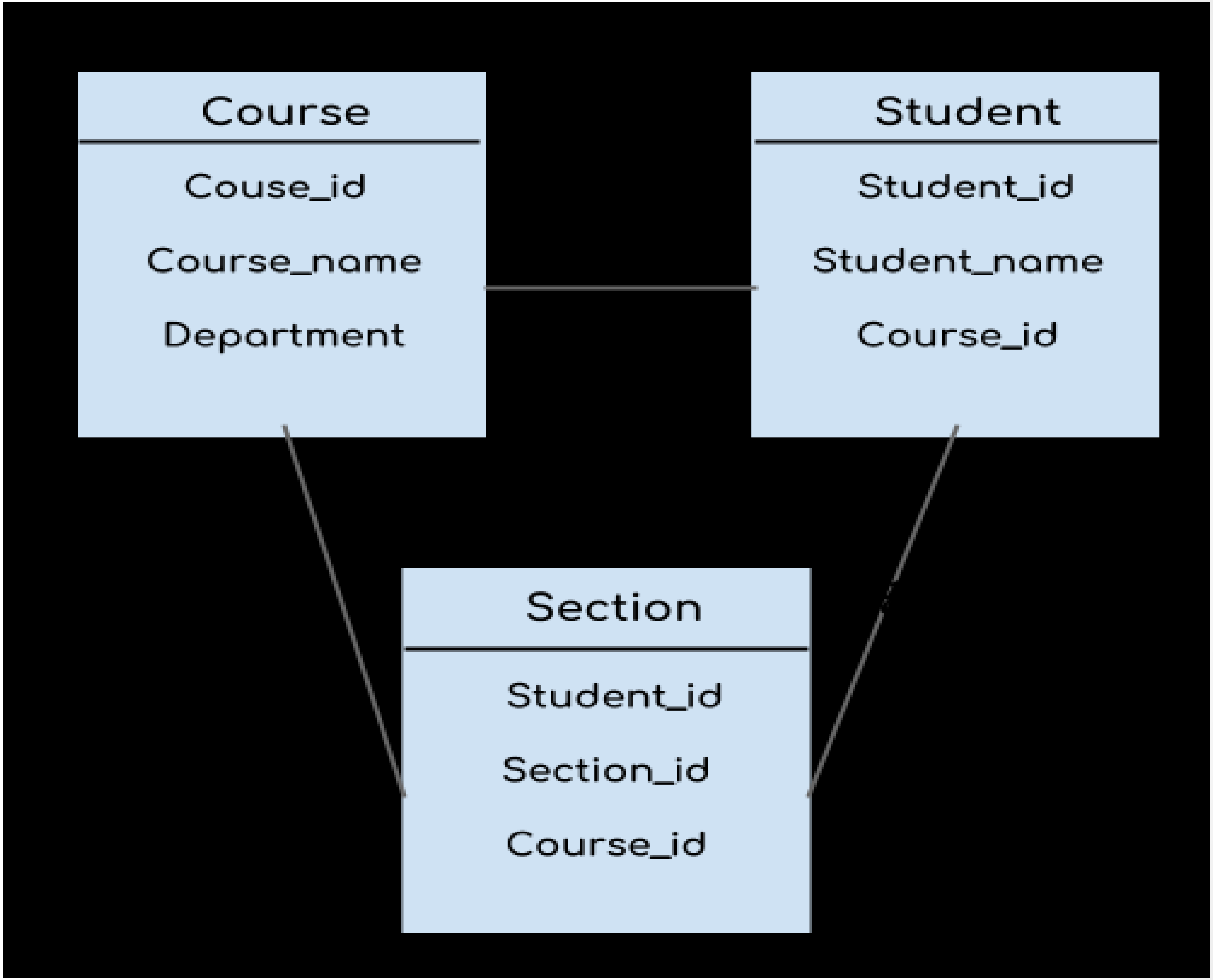
1. MySQL
2. Microsoft Access
3. Oracle
4. PostgreSQL
5. dBASE
6. FoxPro
7. SQLite
8. IBM DB2
9. LibreOffice Base
10. MariaDB
11. Microsoft SQL Server etc.

# Application of DBMS

| Sector | Use of DBMS |
|---|---|
| Banking | For customer information, account activities, payments, deposits, loans, etc. |
| Airlines | For reservations and schedule information. |
| Universities | For student information, course registrations, colleges and grades. |
| Telecommunication | It helps to keep call records, monthly bills, maintaining balances, etc. |
| Finance | For storing information about stock, sales, and purchases of financial instruments like stocks and bonds. |
| Sales | Use for storing customer, product & sales information. |
| Manufacturing | It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses. |
| HR Management | For information about employees, salaries, payroll, deduction, generation of paychecks, |

# Instance and Scheme

- **Definition of schema**: Design of a database is called the schema. Schema is of three types:
  1. Physical schema
  2. logical schema
  3. view schema
- For example: In the following diagram, we have a schema that shows the relationship between three tables: Course, Student and Section. The diagram only shows the design of the database, it doesn’t show the data present in those tables. Schema is only a structural view(design) of a database as shown in the diagram below.



Course

Couse_id

Course_name

Department

Student

Student_id

Student_name

Course_id

Section

Student_id

Section_id

Course_id

- The design of a database at physical level is called **physical schema**, how the data stored in blocks of storage is described at this level.
- Design of database at logical level is called **logical schema**, programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).
- Design of database at view level is called **view schema**. This generally describes end user interaction with database systems.

# DBMS Instance

- **Definition of instance**: The data stored in database at a particular moment of time is called instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.
- For example, lets say we have a single table student in the database, today the table has 100 records, so today the instance of the database has 100 records. Lets say we are going to add another 100 records in this table by tomorrow so the instance of database tomorrow will have 200 records in table. In short, at a particular moment the data stored in database is called the instance, that changes over time when we add or delete data from the database.

# Key Points

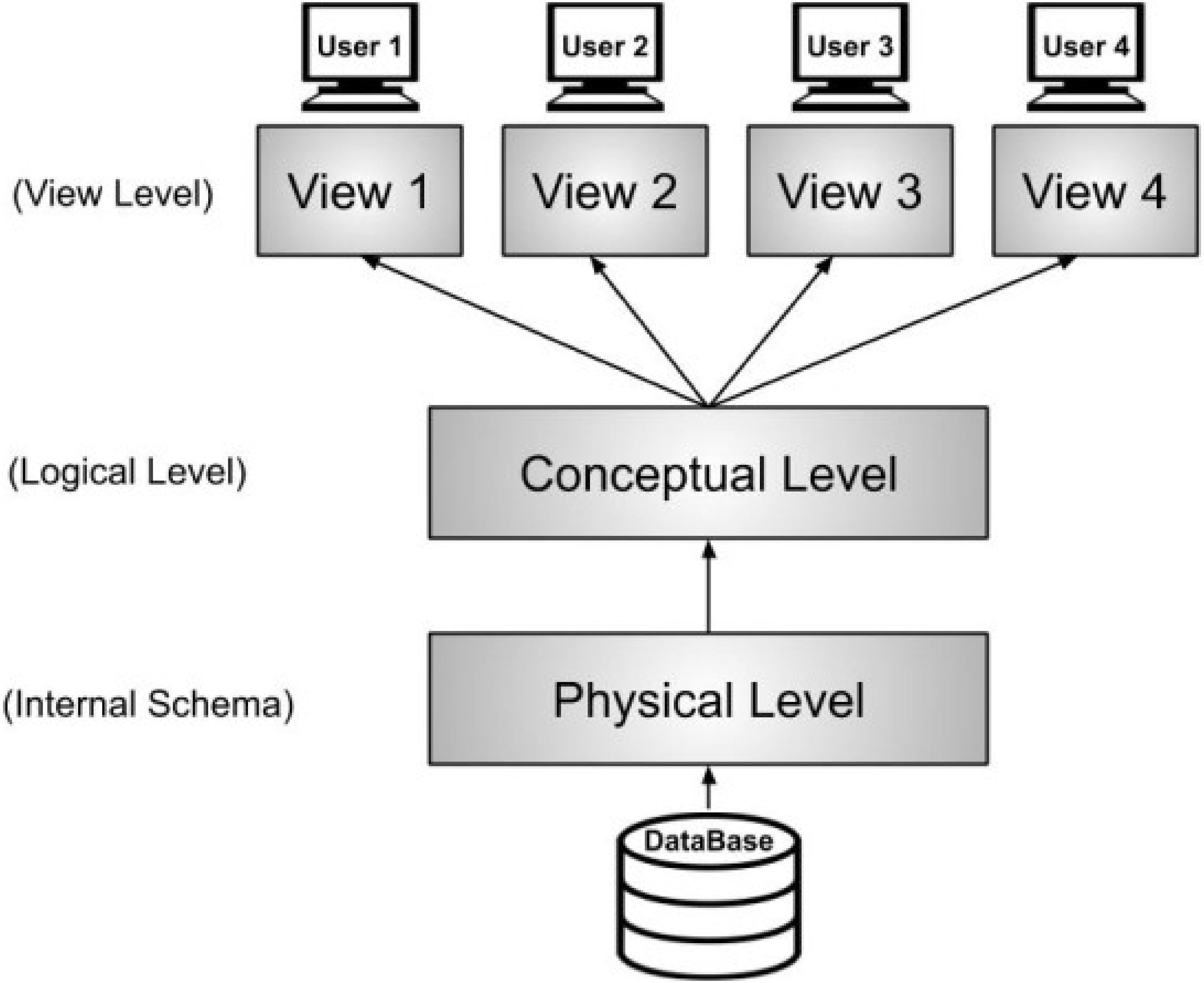
- The data which is stored in the database at a particular moment of time is called an instance of the database.
- The overall design of a database is called schema.
- A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
- A schema contains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

# Data Abstraction

- Data Abstraction refers to the process of hiding irrelevant details from the user. So, what is the meaning of irrelevant details? Let's understand this with one example.
- *Example:* If we want to access any mail from our Gmail then we don't know where that data is physically stored i.e is the data present in India or USA or what data model has been used to store that data? We are not concerned about these things. We are only concerned with our email. So, information like these i.e. location of data and data models are irrelevant to us and in data abstraction, we do this only. Apart from the location of data and data models, there are other factors that we don't care of.
- We hide the unnecessary data from the user and this process of hiding unwanted data is called Data Abstraction.

- There are mainly three levels of data abstraction and we divide it into three levels in order to achieve *Data Independence*. Data Independence means users and data should not directly interact with each other. The user should be at a different level and the data should be present at some other level. By doing so, Data Independence can be achieved. So, let's see in details what are these three levels of data abstraction:

1. View Level
2. Conceptual Level
3. Physical Level



**Levels of Data Abstraction**

# View Level or External Schema

- This level tells the application about how the data should be shown to the user.
- *Example:* If we have a login-id and password in a university system, then as a student, we can view our marks, attendance, fee structure, etc. But the faculty of the university will have a different view. He will have options like salary, edit marks of a student, enter attendance of the students, etc. So, both the student and the faculty have a different view. By doing so, the security of the system also increases. In this example, the student can't edit his marks but the faculty who is authorized to edit the marks can edit the student's marks. Similarly, the dean of the college or university will have some more authorization and accordingly, he will has his view. So, different users will have a different view according to the authorization they have.

# Conceptual Level or Logical Level

- This level tells how the data is actually stored and structured. We have different data models by which we can store the  data.
- *Example*: Let us take an example where we use the relational model for storing the data. We have to store the data of a student, the columns in the student table will be student_name, age, mail_id, roll_no etc. We have to define all these at this level while we are creating the database. Though the data is stored in the database but the structure of the tables like the student table, teacher table, books table, etc are defined here in the conceptual level or logical level. Also, how the tables are related to each other are defined here.
- **Overall, we can say that we are creating a blueprint of the data at the conceptual level.**

# Physical Level or Internal Schema

- As the name suggests, the Physical level tells us that where the data is actually stored i.e. it tells the actual location of the data that is being stored by the user.
- **The Database Administrators(DBA)** decide that which data should be kept at which particular disk drive, how the data has to be fragmented, where it has to be stored etc. They decide if the data has to be centralized or distributed. Though we see the data in the form of tables at view level the data here is actually stored in the form of files only. It totally depends on the DBA, how he/she manages the database at the physical level.

# Physical level data independence

- It refers to the characteristic of being able to modify the physical schema without any alterations to the conceptual or logical schema, done for optimization purposes, e.g., Conceptual structure of the database would not be affected by any change in storage size of the database system server. Changing from sequential to random access files is one such example. These alterations or modifications to the physical structure may include:
- Utilizing new storage devices.
- Modifying data structures used for storage.
- Altering indexes or using alternative file organization techniques etc.

# Logical level data independence

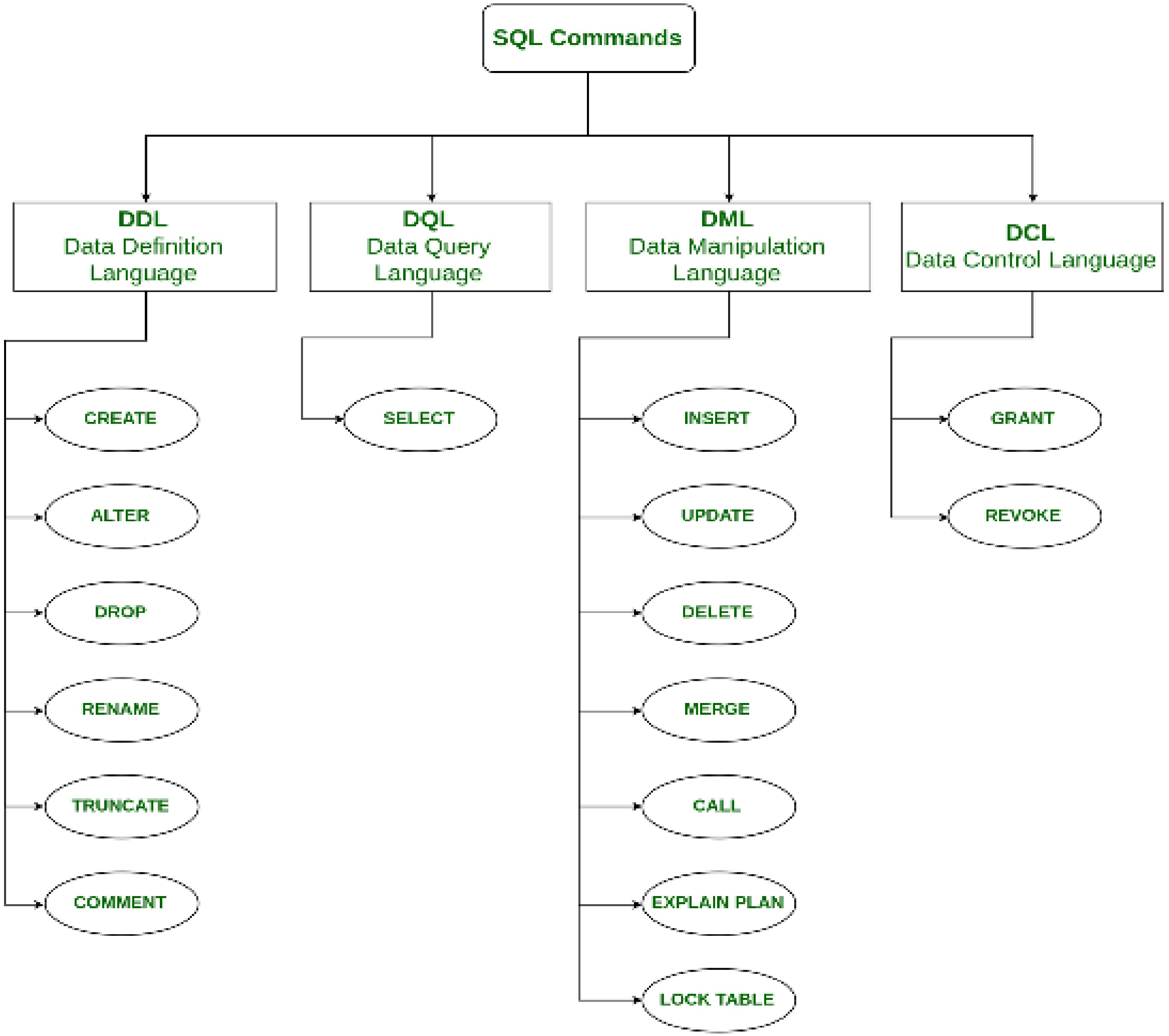
- It refers characteristic of being able to modify the logical schema without affecting the external schema or application program. The user view of the data would not be affected by any changes to the conceptual view of the data. These changes may include insertion or deletion of attributes, altering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

# Structured Query Language(SQL)

- Structured Query Language(SQL) is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert etc. to carry out the required tasks.
- These SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

# Types of SQL Commands



# DDL(Data Definition Language)

- DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
- **Examples of DDL commands:**

1. **CREATE** – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
2. **DROP** – is used to delete objects from the database.
3. **ALTER**-is used to alter the structure of the database.
4. **TRUNCATE**–is used to remove all records from a table, including all spaces allocated for the records are removed.
5. **COMMENT** –is used to add comments to the data dictionary.
6. **RENAME** –is used to rename an object existing in the database.

# 1. CREATE

**Syntax:**

CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

**Example:**

```
CREATE TABLE EMPLOYEE(Name VARCHAR2(20),
 Email VARCHAR2(100), DOB DATE);
```

**2. DROP:** It is used to delete both the structure and record stored in the table.

**Syntax**

DROP TABLE ;

**Example**

```
DROP TABLE EMPLOYEE;
```

# 3. ALTER

It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

**Syntax:**

- To add a new column in the table

  ALTER TABLE table_name ADD column_name COLUMN-definition;

- To modify existing column in the table
  ALTER TABLE MODIFY(COLUMN DEFINITION....);

**EXAMPLE**

ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));

ALTER TABLE STU_DETAILS MODIFY  column NAME VARCHAR2(20);

# 4. TRUNCATE

It is used to delete all the rows from the table and free the space containing the table.

**Syntax:**

```
TRUNCATE TABLE table_name;
```

**Example:**

```
TRUNCATE TABLE EMPLOYEE;
```

# DQL (Data Query Language)

- **DQL (Data Query Language)** :
- DML statements are used for performing queries on the data within schema objects. The purpose of DQL Command is to get some schema relation based on the query passed to it.
- **Example of DQL:**
- **SELECT**– is used to retrieve data from the a database.

**Syntax:**

SELECT expressions    FROM TABLES    WHERE conditions;

**Example:**

SELECT emp_name  FROM employee  WHERE age > 20;

# DML(Data Manipulation Language)

- The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.
- **Examples of DML:**

1. **INSERT** – is used to insert data into a table.
2. **UPDATE** – is used to update existing data within a table.
3. **DELETE**– is used to delete records from a database table.

# 1. INSERT

**Syntax:**

```
INSERT INTO TABLE_NAME  (col1, col2, col3,.... col N)
          VALUES (value1, value2, value3, .... valueN);
        Or
INSERT INTO TABLE_NAME  VALUES (value1, value2,
value3, .... valueN);
```

**For example:**

```
INSERT INTO student(Name, Subject)
VALUES (“Mani", "DBMS");
```

# 2. UPDATE:

This command is used to update or modify the value of a column in the table.

**Syntax:**

```
   UPDATE table_name SET [column_name1= value1,...colu
mn_nameN = valueN] [WHERE CONDITION]
```

**For example:**

```
   UPDATE students   SET User_Name = ‘Mani'
   WHERE Student_Id = '3'
```

# 3. DELETE

It is used to remove one or more row from a table.

**Syntax:**

```
   DELETE FROM table_name [WHERE condition];
```

**For example:**

```
   DELETE FROM student WHERE Name=“Mani";
```

# DCL(Data Control Language)

- DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.
- **Examples of DCL commands:**
  1. **GRANT**-gives user's access privileges to database.
  2. **REVOKE**-withdraw user's access privileges given by using the GRANT command.

# TCL(transaction Control Language)

- TCL commands deals with the transaction within the database.
- **Examples of TCL commands:**
  1. **COMMIT**– commits a Transaction.
  2. **ROLLBACK–** rollbacks a transaction in case of any error occurs.
  3. **SAVEPOINT**–sets a savepoint within a transaction.
  4. **SET TRANSACTION**–specify characteristics for the transaction.