

DATA BASE MANAGEMENT SYSTEM

UNIT-III

**Relational Model (RM)** represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

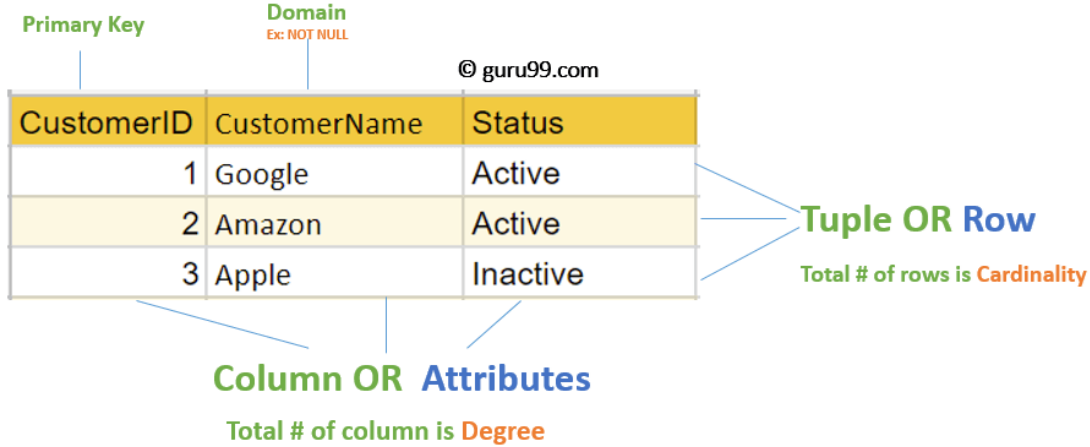
**Some popular Relational Database management systems are:**

- DB2 and Informix Dynamic Server – IBM
- Oracle and RDB – Oracle
- SQL Server and Access – Microsoft

**Relational Model Concepts in DBMS**

1. Attribute: Each column in a Table. Attributes are the properties which define a relation. e.g., Student\_Rollno, Name, etc.
2. Tables – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
3. Tuple – It is nothing but a single row of a table, which contains a single record.
4. Relation Schema: A relation schema represents the name of the relation with its attributes.
5. Degree: The total number of attributes which in the relation is called the degree of the relation.
6. Cardinality: Total number of rows present in the Table.
7. Column: The column represents the set of values for a specific attribute.
8. Relation instance – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
9. Relation key – Every row has one, two or multiple attributes, which is called relation key.
10. Attribute domain – Every attribute has some pre-defined value and scope which is known as attribute domain.

## Table also called Relation



### Relational Integrity Constraints

Relational Integrity constraints in DBMS are referred to conditions which must be present for a valid relation. These Relational constraints in DBMS are derived from the rules in the mini-world that the database represents.

There are many types of Integrity Constraints in DBMS. Constraints on the Relational database management system is mostly divided into three main categories are:

1. Domain Constraints
2. Key Constraints
3. Referential Integrity Constraints

### Domain Constraints

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.

Domain constraints specify that within each tuple, and the value of each attribute must be unique. This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

### Key Constraints

An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

Example:

CustomerID	CustomerName	Status
------------	--------------	--------

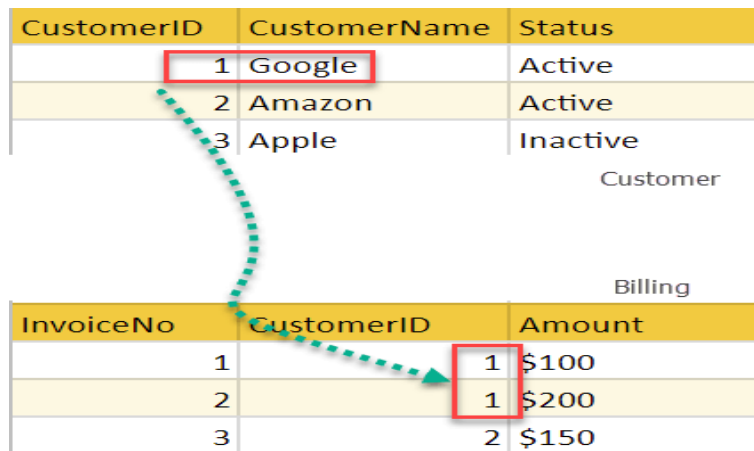
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =” Google”.

### Referential Integrity Constraints

Referential Integrity constraints in DBMS are based on the concept of Foreign Keys. A foreign key is an important attribute of a relation which should be referred to in other relationships. Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.

**Example:**



In the above example, we have 2 relations, Customer and Billing.

Tuple for CustomerID =1 is referenced twice in the relation Billing. So we know CustomerName=Google has billing amount \$300

### Operations in Relational Model

Four basic update operations performed on relational database model are

Insert, update, delete and select.


- Insert is used to insert data into the relation
- Delete is used to delete tuples from the table.
- Modify allows you to change the values of some attributes in existing tuples.
- Select allows you to choose a specific range of data.

Whenever one of these operations are applied, integrity constraints specified on the relational database schema must never be violated.

### Insert Operation

The insert operation gives values of the attribute for a new tuple which should be inserted into a relation.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

### Update Operation

You can see that in the below-given relation table CustomerName= 'Apple' is updated from Inactive to Active.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

### Delete Operation

To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

In the above-given example, CustomerName= "Apple" is deleted from the table.

The Delete operation could violate referential integrity if the tuple which is deleted is referenced by foreign keys from other tuples in the same database.

### Select Operation

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active



CustomerID	CustomerName	Status
2	Amazon	Active

In the above-given example, CustomerName="Amazon" is selected

## Advantages of Relational Database Model

- **Simplicity:** A Relational data model in DBMS is simpler than the hierarchical and network model.
- **Structural Independence:** The relational database is only concerned with data and not with a structure. This can improve the performance of the model.
- **Easy to use:** The Relational model in DBMS is easy as tables consisting of rows and columns are quite natural and simple to understand
- **Query capability:** It makes possible for a high-level query language like SQL to avoid complex database navigation.
- **Data independence:** The Structure of Relational database can be changed without having to change any application.
- **Scalable:** Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.

## Disadvantages of Relational Model

- Few relational databases have limits on field lengths which can't be exceeded.
- Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.

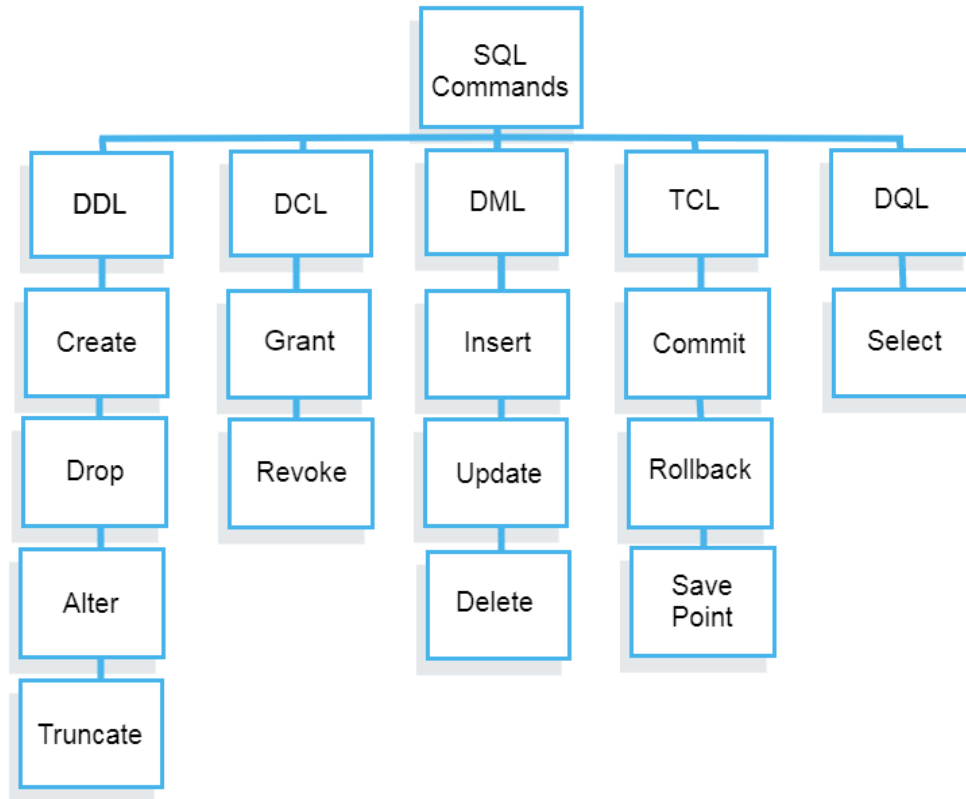
## SQL (Structured Query Language)

SQL is a database language designed for the retrieval and management of data in a relational database. SQL is the standard language for database management. All the RDBMS systems like MySQL, MS Access, Oracle, Sybase, Postgres, and SQL Server use SQL as their standard database language. SQL programming language uses various commands for different operations. We will learn about the like DCL, TCL, DQL, DDL and DML commands in SQL with examples.

## Types of SQL

Here are five types of widely used SQL queries.

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language(DCL)
- Transaction Control Language(TCL)
- Data Query Language (DQL)



Types of SQL

## DDL

Data Definition Language helps you to define the database structure or schema. Five types of DDL commands in SQL are:

### 1.CREATE

CREATE statements is used to define the database structure schema:

#### Syntax:

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

#### For example:

```
Create database university;
```

```
Create table students;
```

```
Create view for_students;
```

### 2.DROP

Drops commands remove tables and databases from RDBMS.

#### Syntax

```
DROP TABLE ;
```

#### For example:

Drop object\_type object\_name;  
Drop database university;  
Drop table student;

### 3. ALTER

Alters command allows you to alter the structure of the database.

**Syntax:**

To add a new column in the table

ALTER TABLE table\_name ADD column\_name COLUMN-definition;

To modify an existing column in the table:

ALTER TABLE MODIFY(COLUMN DEFINITION....);

**For example:**

Alter table guru99 add subject varchar;

### 4. TRUNCATE:

This command used to delete all the rows from the table and free the space containing the table.

**Syntax:**

TRUNCATE TABLE table\_name;

**Example:**

TRUNCATE table students;

## Data Manipulation Language

Data Manipulation Language (DML) allows you to modify the database instance by inserting, modifying, and deleting its data. It is responsible for performing all types of data modification in a database.

There are three basic constructs which allow database program and user to enter data and information are:

Here are some important DML commands in SQL:

- INSERT
- UPDATE
- DELETE

### 1. INSERT:

This is a statement is a SQL query. This command is used to insert data into the row of a table.

**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 students (RollNo, FirstName, LastName) VALUES ('60', 'Tom', Erichsen');

## **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,...column_nameN = valueN] [WHERE  
CONDITION]
```

### **For example:**

```
UPDATE students  
SET FirstName = 'Jhon', LastName= 'Wick'  
WHERE StudID = 3;
```

## **3.DELETE:**

This command is used to remove one or more rows from a table.

### **Syntax:**

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

### **For example:**

```
DELETE FROM students  
WHERE FirstName = 'Jhon';
```

## **DCL**

DCL (Data Control Language) includes commands like GRANT and REVOKE, which are useful to give “rights & permissions.” Other permission controls parameters of the database system.

### **Examples of DCL commands:**

Commands that come under DCL:

- Grant
- Revoke

### **1.Grant:**

This command is use to give user access privileges to a database.

#### **Syntax:**

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

#### **For example:**

```
GRANT SELECT ON Users TO'Tom'@'localhost';
```

### **2.Revoke:**

It is useful to back permissions from the user.

#### **Syntax:**

```
REVOKE privilege_name ON object_name FROM {user_name |PUBLIC |role_name}
```

#### **For example:**

```
REVOKE SELECT, UPDATE ON student FROM BCA, MCA;
```



## TCL

Transaction control language or TCL commands deal with the transaction within the database.

### 1.Commit

This command is used to save all the transactions to the database.

**Syntax:**

Commit;

**For example:**

```
DELETE FROM Students  
WHERE RollNo =25;  
COMMIT;
```

### 2.Rollback

Rollback command allows you to undo transactions that have not already been saved to the database.

**Syntax:**

ROLLBACK;

**Example:**

```
DELETE FROM Students  
WHERE RollNo =25;
```

### 3.SAVEPOINT

This command helps you to sets a savepoint within a transaction.

**Syntax:**

SAVEPOINT SAVEPOINT\_NAME;

**Example:**

```
SAVEPOINT RollNo;
```

## DQL

Data Query Language (DQL) is used to fetch the data from the database. It uses only one command:

**SELECT:**

This command helps you to select the attribute based on the condition described by the WHERE clause.

**Syntax:**

```
SELECT expressions  
FROM TABLES  
WHERE conditions;
```

**For example:**

```
SELECT FirstName  
FROM Student  
WHERE RollNo > 15;
```