Name: Tejaswini Kokane
Batch: 7670
Enrollment No.: EBEON0722630678

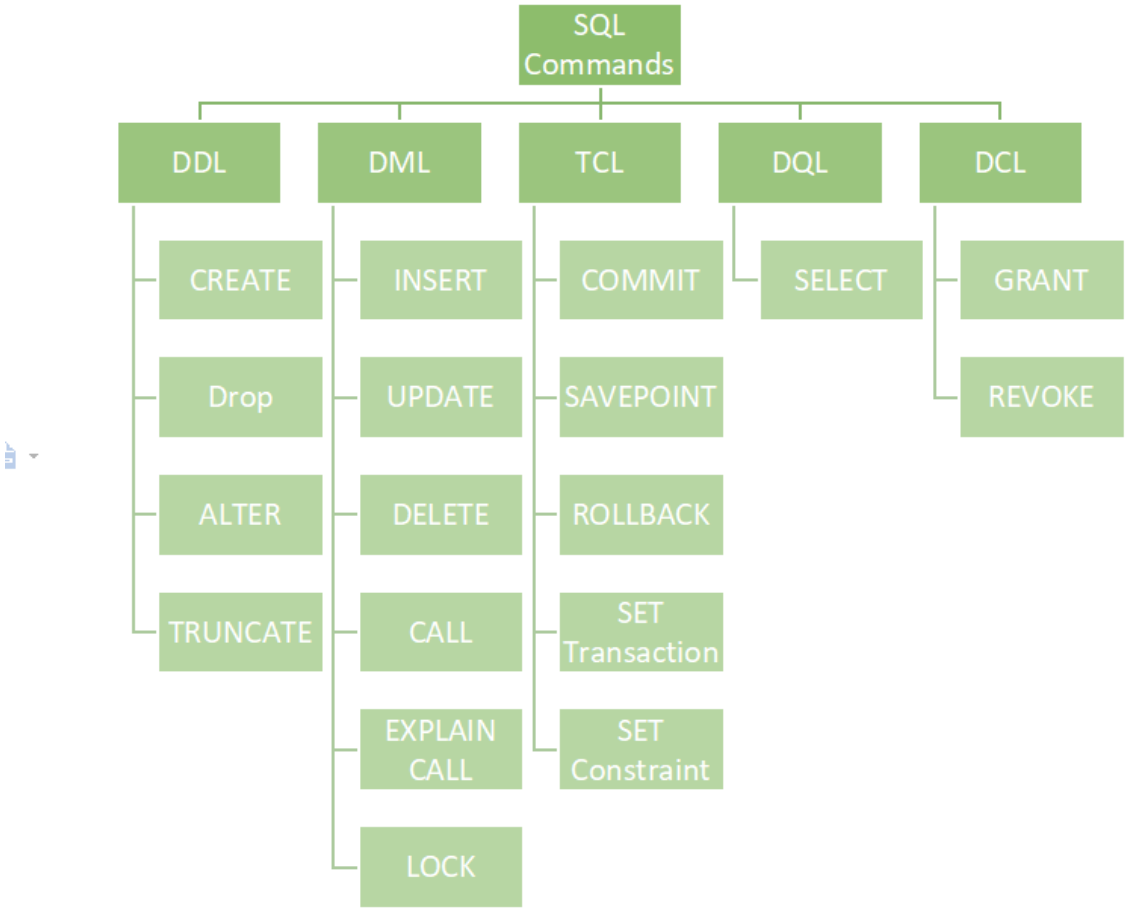
SQL (Structured Query Language) Commands

Structured Query Language (SQL) as we all know 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/ DRL Data Retrieval Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

Though many resources claim there to be another category of SQL clauses **TCL – Transaction Control Language**. So we will see in detail about TCL as well.



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. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.
List of DDL commands:

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

DQL (Data Query Language): Data Retrieval language.

DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement. This command allows getting the data out of the database to perform operations with it. When a SELECT is fired against a table or tables the result is compiled into a further temporary table, which is displayed or perhaps received by the program i.e. a front-end.

List of DQL:

SELECT: It is used to retrieve data from the database.

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. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands:

- **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 records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.

DCL (Data Control Language):

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

- **GRANT:** This command gives users access privileges to the database.
- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

Though many resources claim there to be another category of SQL clauses TCL – Transaction Control Language. So we will see in detail about TCL as well. TCL commands deal with the transaction within the database.

List of TCL commands:

- **COMMIT:** Commits a Transaction.
- **ROLLBACK:** Rollbacks a transaction in case of any error occurs.
- **SAVEPOINT:** Sets a save point within a transaction.
- **SET TRANSACTION:** Specify characteristics for the transaction.

https://www.oracle.com/database/technologies/xe-prior-release-downloads.html

https://youtu.be/seFRL1GAzLY

https://www.testingdocs.com/download-install-mysql-on-windows-11/

https://dev.mysql.com/downloads/installer/

https://youtu.be/eq-e_n7lm2M

https://www.youtube.com/watch?v=WuBcTJnIuzo

https://www.youtube.com/watch?v=wEHWYuzP7VE

C:\Windows\system32>mysql -u root -p

Enter password: *********

Mysql>create database student;

Mysql>use student;

Mysql>show databases;

DDL:

Create:

Syntax: CREATE TABLE *table_name* (*column1 datatype*,*column2 datatype*, *column3 datatype* ...);

Example:

mysql> create table Customers(CustomerID int(20), Name varchar(20), Address varchar(20))

```
  ->;
```

Query OK, 0 rows affected, 1 warning (0.22 sec)

```
mysql> Insert into Sales values(1, 'cars', 02, 2000);
```

Query OK, 1 row affected (0.13 sec)

Drop:

Syntax: `DROP TABLE table_name;`

Example:

```
mysql> show tables;
+-------------------+
| Tables_in_student |
+-------------------+
| customers         |
| emp               |
| people            |
| sales             |
+-------------------+
4 rows in set (0.10 sec)

mysql> Drop table Emp;
Query OK, 0 rows affected (0.56 sec)

mysql> Show tables;
+-------------------+
| Tables_in_student |
+-------------------+
| customers         |
| people            |
| sales             |
```

```
+-------------------+
3 rows in set (0.00 sec)
```

Alter:

Syntax: `ALTER TABLE table_name ADD column_name datatype;`

Example:

```
mysql> select * from stuedubridge;
+---------+--------+--------------------+
| roll_No | Name   | Subject            |
+---------+--------+--------------------+
|       1 | Anna   | SoftwareTesting    |
|       2 | Alice  | AptitudeSkills     |
|       3 | Emma   | CommunicationSkills |
|       4 | Sophia | Bootstrap          |
+---------+--------+--------------------+
4 rows in set (0.00 sec)

mysql> Alter table stuedubridge ADD City varchar(25);
Query OK, 0 rows affected (0.40 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from stuedubridge;
+---------+--------+--------------------+------+
| roll_No | Name   | Subject            | City |
+---------+--------+--------------------+------+
|       1 | Anna   | SoftwareTesting    | NULL |
|       2 | Alice  | AptitudeSkills     | NULL |
```

| 3 | Emma | CommunicationSkills | NULL |

| 4 | Sophia | Bootstrap | NULL |

+---------+--------+--------------------+------+

4 rows in set (0.00 sec)

DML:

Insert:

Syntax: `INSERT INTO table_name VALUES (value1, value2, value3, ...);`

Example:

mysql> create table Customers(CustomerID int(20), Name varchar(20), Address varchar(20))

-> ;

Query OK, 0 rows affected, 1 warning (0.22 sec)

mysql> Insert into Sales values(1, 'cars', 02, 2000);

Query OK, 1 row affected (0.13 sec)

Update:

Syntax: `UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;`

Example:

mysql> select * from stuedubridge;

+---------+--------+--------------------+

| roll_No | Name | Subject |

+---------+--------+--------------------+

| 1 | Anna | Testing |

| 2 | Alice | AptitudeSkills |

```
|      3 | Emma   | CommunicationSkills |
|      4 | Sophia | Bootstrap           |
+---------+--------+---------------------+
4 rows in set (0.00 sec)

mysql> update stuedubridge SET Subject='SoftwareTesting' WHERE roll_No=1;
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from stuedubridge;
+---------+--------+---------------------+
| roll_No | Name   | Subject             |
+---------+--------+---------------------+
|      1 | Anna   | SoftwareTesting     |
|      2 | Alice  | AptitudeSkills      |
|      3 | Emma   | CommunicationSkills |
|      4 | Sophia | Bootstrap           |
+---------+--------+---------------------+
4 rows in set (0.00 sec)
```

Delete*:*

Syntax: `DELETE FROM table_name WHERE condition;`

Example:

```
mysql> select * from stuedubridge;
+---------+----------+---------------------+
| roll_No | Name     | Subject             |
```

```
+---------+-----------+--------------------+
|       1 | Tejaswini | Software_testing   |
|       1 | Anna      | Testing            |
|       2 | Alice     | AptitudeSkills     |
|       3 | Emma      | CommunicationSkills |
|       4 | Sophia    | Bootstrap          |
+---------+-----------+--------------------+
5 rows in set (0.00 sec)

mysql> Delete from stuedubridge WHERE Name='Tejaswini';
Query OK, 1 row affected (0.10 sec)

mysql> select * from stuedubridge;
+---------+--------+--------------------+
| roll_No | Name   | Subject            |
+---------+--------+--------------------+
|       1 | Anna   | Testing            |
|       2 | Alice  | AptitudeSkills     |
|       3 | Emma   | CommunicationSkills |
|       4 | Sophia | Bootstrap          |
+---------+--------+--------------------+
4 rows in set (0.00 sec)
```

DQL:

Syntax: `SELECT * FROM table_name;`

Example:

mysql> select * from stuedubridge;

```
+---------+-----------+--------------------+
| roll_No | Name      | Subject            |
+---------+-----------+--------------------+
|       1 | Tejaswini | Software_testing   |
|       1 | Anna      | Testing            |
|       2 | Alice     | AptitudeSkills     |
|       3 | Emma      | CommunicationSkills |
|       4 | Sophia    | Bootstrap          |
+---------+-----------+--------------------+
```

5 rows in set (0.00 sec)

Select Distinct:

Syntax: `SELECT DISTINCT column1, column2, ...FROM table_name;`

Example:

mysql> select * from Stu_Edubridge;

```
+---------+--------+--------------------+
| roll_No | Name   | Subject            |
+---------+--------+--------------------+
|       1 | Anna   | SoftwareTesting    |
|       2 | Alice  | AptitudeSkills     |
|       3 | Emma   | CommunicationSkills |
|       4 | Sophia | Bootstrap          |
|       5 | Zoe    | SoftwareTesting    |
+---------+--------+--------------------+
```

5 rows in set (0.00 sec)

mysql> Select DISTINCT Subject from Stu_Edubridge;

```
+--------------------+
| Subject            |
+--------------------+
| SoftwareTesting    |
| AptitudeSkills     |
| CommunicationSkills |
| Bootstrap          |
+--------------------+
```

4 rows in set (0.08 sec)

Count():

Syntax: `SELECT COUNT(column_name) FROM table_name ORDER BY column_name(s);`

Example:

mysql> select * from Stu_Edubridge;

```
+---------+--------+--------------------+
| roll_No | Name   | Subject            |
+---------+--------+--------------------+
|       1 | Anna   | SoftwareTesting    |
|       2 | Alice  | AptitudeSkills     |
|       3 | Emma   | CommunicationSkills |
|       4 | Sophia | Bootstrap          |
|       5 | Zoe    | SoftwareTesting    |
+---------+--------+--------------------+
```

5 rows in set (0.00 sec)

mysql> Select Name, COUNT(*) from Stu_Edubridge GROUP BY subject;

```
+--------+----------+
| Name   | COUNT(*) |
+--------+----------+
| Anna   |        2 |
| Alice  |        1 |
| Emma   |        1 |
| Sophia |        1 |
+--------+----------+
```

4 rows in set (0.04 sec)

Order By:

Syntax: `SELECT column1, column2, ...FROM table_name ORDER BY column1, column2, ... ASC|DESC;`

Example:

mysql> select * from Stu_Edubridge;

```
+---------+--------+--------------------+
| roll_No | Name   | Subject            |
+---------+--------+--------------------+
|       1 | Anna   | SoftwareTesting    |
|       2 | Alice  | AptitudeSkills     |
|       3 | Emma   | CommunicationSkills |
|       4 | Sophia | Bootstrap          |
|       5 | Zoe    | SoftwareTesting    |
+---------+--------+--------------------+
```

5 rows in set (0.00 sec)

```
mysql> Select * from Stu_Edubridge WHERE Subject='SoftwareTesting' ORDER BY Name ASC;
+---------+------+-----------------+
| roll_No | Name | Subject         |
+---------+------+-----------------+
|       1 | Anna | SoftwareTesting |
|       5 | Zoe  | SoftwareTesting |
+---------+------+-----------------+
2 rows in set (0.18 sec)

mysql> Select * from Stu_Edubridge WHERE Subject='SoftwareTesting' ORDER BY Name DESC;
+---------+------+-----------------+
| roll_No | Name | Subject         |
+---------+------+-----------------+
|       5 | Zoe  | SoftwareTesting |
|       1 | Anna | SoftwareTesting |
+---------+------+-----------------+
2 rows in set (0.06 sec)

mysql> Select Name, Subject from Stu_Edubridge Where roll_No < 5
    -> ORDER BY Name DESC, Subject ASC;
+--------+--------------------+
| Name   | Subject            |
+--------+--------------------+
| Sophia | Bootstrap          |
```

| Emma | CommunicationSkills |

| Anna | SoftwareTesting |

| Alice | AptitudeSkills |

+--------+--------------------+

4 rows in set (0.05 sec)

Select Top (LIMIT Clause):

Syntax: `SELECT column_name(s) FROM table_name LIMIT number;`

Example:

mysql> Select * from Stu_Edubridge;

+---------+--------+--------------------+

| roll_No | Name | Subject |

+---------+--------+--------------------+

| 1 | Anna | SoftwareTesting |

| 2 | Alice | AptitudeSkills |

| 3 | Emma | CommunicationSkills |

| 4 | Sophia | Bootstrap |

| 5 | Zoe | SoftwareTesting |

+---------+--------+--------------------+

5 rows in set (0.00 sec)

mysql> Select Name from Stu_Edubridge LIMIT 1;

+------+

| Name |

+------+

| Anna |

+------+

1 row in set (0.05 sec)

Where Clause:

Syntax: `SELECT column1, column2, ...FROM table_name WHERE condition;`

Example:

```
mysql> Select * from Stu_Edubridge;
+---------+--------+--------------------+
| roll_No | Name   | Subject            |
+---------+--------+--------------------+
|       1 | Anna   | SoftwareTesting    |
|       2 | Alice  | AptitudeSkills     |
|       3 | Emma   | CommunicationSkills |
|       4 | Sophia | Bootstrap          |
|       5 | Zoe    | SoftwareTesting    |
|       6 | Anna   | SoftwareTesting    |
+---------+--------+--------------------+
6 rows in set (0.00 sec)
```

```
mysql> SELECT * from Stu_Edubridge WHERE Name='Anna' OR
Subject='SoftwareTesting';
+---------+------+-----------------+
| roll_No | Name | Subject         |
+---------+------+-----------------+
|       1 | Anna | SoftwareTesting |
|       5 | Zoe  | SoftwareTesting |
|       6 | Anna | SoftwareTesting |
+---------+------+-----------------+
```

3 rows in set (0.00 sec)

SQL Joins:

Inner Join:

Syntax: `SELECT column_name(s) FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;`

Example:

```
mysql> Select * from Sales;
+--------+-----------+------------+-------+
| saleID | Product   | CustomerID | Price |
+--------+-----------+------------+-------+
|      1 | cars      |          2 |  2000 |
|      2 | Furniture |          4 |  4000 |
+--------+-----------+------------+-------+
2 rows in set (0.04 sec)

mysql> Select * from Customers;
+------------+---------+----------+
| CustomerID | Name    | Address  |
+------------+---------+----------+
|          1 | Deepika | Delhi    |
|          2 | Reva    | Hariyana |
+------------+---------+----------+
2 rows in set (0.00 sec)
```

```
mysql> Select Sales.saleID, Sales.Product, Customers.Name
    -> from Sales
    -> INNER JOIN Customers ON Sales. CustomerID=Customers.CustomerID;
+--------+---------+------+
| saleID | Product | Name |
+--------+---------+------+
|      1 | cars    | Reva |
+--------+---------+------+
1 row in set (0.00 sec)
```

Left Join:

Syntax: `SELECT column_name(s)FROM table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;`

Example:

```
mysql> Select * from Sales;
+--------+-----------+------------+-------+
| saleID | Product   | CustomerID | Price |
+--------+-----------+------------+-------+
|      1 | cars      |          2 |  2000 |
|      2 | Furniture |          4 |  4000 |
+--------+-----------+------------+-------+
2 rows in set (0.00 sec)

mysql> Select * from Customers;
+------------+---------+----------+
| CustomerID | Name    | Address  |
```

```
+------------+---------+----------+
|          1 | Deepika | Delhi    |
|          2 | Reva    | Hariyana |
+------------+---------+----------+
2 rows in set (0.00 sec)
```

```
mysql> Select Sales.saleID,Sales.Product, Customers.Name
    -> from Sales
    -> LEFT JOIN Customers ON Sales.CustomerID=Customers.CustomerID;
+--------+-----------+------+
| saleID | Product   | Name |
+--------+-----------+------+
|      1 | cars      | Reva |
|      2 | Furniture | NULL |
+--------+-----------+------+
2 rows in set (0.00 sec)
```

Right Join:

Syntax: `SELECT column_name(s) FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;`

Example:

```
mysql> Select * from Sales;
+--------+-----------+------------+-------+
| saleID | Product   | CustomerID | Price |
+--------+-----------+------------+-------+
```

```
|      1 | cars      |         2 |  2000 |
|      2 | Furniture |         4 |  4000 |
+--------+-----------+-----------+-------+
2 rows in set (0.00 sec)

mysql> Select * from Customers;
+------------+---------+----------+
| CustomerID | Name    | Address  |
+------------+---------+----------+
|          1 | Deepika | Delhi    |
|          2 | Reva    | Hariyana |
+------------+---------+----------+
2 rows in set (0.00 sec)
mysql> Select Sales.saleID,Sales.Product, Customers.Name
    -> from Sales
    -> RIGHT JOIN Customers ON Sales.CustomerID=Customers.CustomerID;
+--------+---------+---------+
| saleID | Product | Name    |
+--------+---------+---------+
|   NULL | NULL    | Deepika |
|      1 | cars    | Reva    |
+--------+---------+---------+
2 rows in set (0.00 sec)
```

Union:

Syntax: `SELECT column_name(s) FROM table1 UNION SELECT column_name(s) FROM table2;`

Example:

```
mysql> Select * from Sales;
+--------+-----------+------------+-------+
| saleID | Product   | CustomerID | Price |
+--------+-----------+------------+-------+
|      1 | cars      |          2 |  2000 |
|      2 | Furniture |          4 |  4000 |
+--------+-----------+------------+-------+
2 rows in set (0.13 sec)

mysql> Select * from customers;
+------------+---------+----------+
| CustomerID | Name    | Address  |
+------------+---------+----------+
|          1 | Deepika | Delhi    |
|          2 | Reva    | Hariyana |
+------------+---------+----------+
2 rows in set (0.08 sec)
mysql> Select CustomerID from Sales
    -> UNION
    -> Select CustomerID from Customers;
+------------+
| CustomerID |
+------------+
```

| 2 |

| 4 |

| 1 |

+------------+

3 rows in set (0.03 sec)

Syntax: *Select * from table1 Left Join table2 ON (table1.field1, table2.field2, table3.field3)*

UNION

*Select * from table1 Right Join table2 ON (table1.field1, table2.field2, table3.field3)*

Example:

mysql> Select * from Sales;

+--------+-----------+------------+-------+

| saleID | Product | CustomerID | Price |

+--------+-----------+------------+-------+

| 1 | cars | 2 | 2000 |

| 2 | Furniture | 4 | 4000 |

+--------+-----------+------------+-------+

2 rows in set (0.13 sec)

mysql> Select * from customers;

+------------+---------+----------+

| CustomerID | Name | Address |

+------------+---------+----------+

| 1 | Deepika | Delhi |

```
|         2 | Reva    | Hariyana |
+-----------+---------+----------+
2 rows in set (0.08 sec)

mysql> Select * from Sales
   -> Left JOIN Customers
ON(Sales.SaleID,Sales.Product,Sales.Price)=(Customers.CustomerID,CUstomers
.Name,Customers.Address)
   -> UNION
   -> Select * From Sales
   -> RIGHT Join Customers
ON(Sales.SaleID,Sales.Product,Sales.Price)=(Customers.CUstomerID,Customers
.Name,Customers.Address);
+--------+-----------+------------+-------+------------+---------+----------+
| saleID | Product   | CustomerID | Price | CustomerID | Name    | Address  |
+--------+-----------+------------+-------+------------+---------+----------+
|      1 | cars      |          2 |  2000 |       NULL | NULL    | NULL     |
|      2 | Furniture |          4 |  4000 |       NULL | NULL    | NULL     |
|   NULL | NULL      |       NULL |  NULL |          1 | Deepika | Delhi    |
|   NULL | NULL      |       NULL |  NULL |          2 | Reva    | Hariyana |
+--------+-----------+------------+-------+------------+---------+----------+
4 rows in set, 4 warnings (0.17 sec)
```

TCL:

Commit:

Syntax: COMMIT;

Example:

```
mysql> Select * from customers;
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
+------------+----------+----------+
2 rows in set (0.08 sec)

mysql> Start Transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into customers
values(3,'Amit','Punjab'),(4,'Manish','India'),(5,'Ankit','Shimla');
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> Select * from Customers;
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
|          3 | Amit     | Punjab   |
|          4 | Manish   | India    |
|          5 | Ankit    | Shimla   |
+------------+----------+----------+
```

5 rows in set (0.00 sec)

mysql> Commit;

Query OK, 0 rows affected (0.04 sec)

mysql> Select * from Customers;

```
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
|          3 | Amit     | Punjab   |
|          4 | Manish   | India    |
|          5 | Ankit    | Shimla   |
+------------+----------+----------+
```

5 rows in set (0.00 sec)

RollBack:

1. Syntax: **ROLLBACK TO** savepoint_name;

Example:

mysql> select * from customers;

```
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
```

```
|          3 | Amit     | Punjab   |
|          4 | Manish   | India    |
|          5 | Ankit    | Shimla   |
+------------+----------+----------+
5 rows in set (0.05 sec)

mysql> Start Transaction;
Query OK, 0 rows affected (0.00 sec)


mysql> Update customers SET Name='Anshu' Where CustomerID=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Insert into customers values(6,'Deepak','WestIndies');
Query OK, 1 row affected (0.00 sec)


mysql> Delete from customers Where CustomerID=2;
Query OK, 1 row affected (0.03 sec)

mysql> Select * from Customers;
+------------+--------+------------+
| CustomerID | Name   | Address    |
+------------+--------+------------+
|          1 | Anshu  | Delhi      |
```

```
|         3 | Amit   | Punjab     |
|         4 | Manish | India      |
|         5 | Ankit  | Shimla     |
|         6 | Deepak | WestIndies |
+------------+--------+------------+
5 rows in set (0.00 sec)

mysql> Rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> Select * from Customers;
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|         1 | Sherlock | Delhi    |
|         2 | Reva     | Hariyana |
|         3 | Amit     | Punjab   |
|         4 | Manish   | India    |
|         5 | Ankit    | Shimla   |
+------------+----------+----------+
5 rows in set (0.00 sec)
```

Savepoint:

1. Syntax: SAVEPOINT savepoint_name;

Example:

```
mysql> Commit;
Query OK, 0 rows affected (0.00 sec)

mysql> Rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> Select * from Customers;
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
|          3 | Amit     | Punjab   |
|          4 | Manish   | India    |
|          5 | Ankit    | Shimla   |
+------------+----------+----------+
5 rows in set (0.00 sec)

mysql> Savepoint upd;
Query OK, 0 rows affected (0.00 sec)

mysql> Delete from customers where CustomerID=4;
Query OK, 1 row affected (0.06 sec)

mysql> Savepoint del;
Query OK, 0 rows affected (0.00 sec)
```

mysql> Select * from customers;

```
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
|          3 | Amit     | Punjab   |
|          5 | Ankit    | Shimla   |
+------------+----------+----------+
```

4 rows in set (0.00 sec)

Any Five SQL Queries that mam not taught:

MIN:

Syntax: `SELECT MIN(column_name) FROM table_name WHERE condition;`

Example:

mysql> select * from product;

```
+---------+-------+
| Name    | Price |
+---------+-------+
| Shivani |    15 |
| Appi    |    30 |
| Raja    |    45 |
| Dheeraj |    60 |
```

| kumar | 75 |

+---------+-------+

5 rows in set (0.06 sec)

mysql> Select MIN(Price) as smallestprice from product;

+---------------+

| smallestprice |

+---------------+

| 15 |

+---------------+

1 row in set (0.07 sec)

Max:

Syntax: `SELECT MAX(column_name) FROM table_name WHERE condition;`

Example:

mysql> select * from product;

+---------+-------+

| Name | Price |

+---------+-------+

| Shivani | 15 |

| Appi | 30 |

| Raja | 45 |

| Dheeraj | 60 |

| kumar | 75 |

+---------+-------+

5 rows in set (0.06 sec)

mysql> Select MAX(Price) as largestprice from product;

+--------------+

| largestprice |

+--------------+

| 75 |

+--------------+

1 row in set (0.00 sec)

CASE:

Syntax: `CASE WHEN condition1 THEN result1 WHEN condition2 THEN result2 WHEN conditionN THEN result ELSE result END;`

Example:

mysql> select * from product;

+---------+-------+

| Name | Price |

+---------+-------+

| Shivani | 15 |

| Appi | 30 |

| Raja | 45 |

| Dheeraj | 60 |

| kumar | 75 |

+---------+-------+

5 rows in set (0.06 sec)

mysql> Select Price , CASE When price>30 then 'Price is greater than 30' When price =30 then 'price equal to 30' else 'price is under 30' END as pricetext from product;

```
+-------+------------------------+
| Price | pricetext              |
+-------+------------------------+
|    15 | price is under 30      |
|    30 | price equal to 30      |
|    45 | Price is greater than 30 |
|    60 | Price is greater than 30 |
|    75 | Price is greater than 30 |
+-------+------------------------+
```

5 rows in set (0.08 sec)

Like:

Syntax: `SELECT column1, column2, ...FROM table_name WHERE columnN LIKE pattern;`

Example:

mysql> select * from customers;

```
+------------+----------+----------+
| CustomerID | Name     | Address  |
+------------+----------+----------+
|          1 | Sherlock | Delhi    |
|          2 | Reva     | Hariyana |
|          3 | Amit     | Punjab   |
```

```
|          5 | Ankit    | Shimla   |
+------------+----------+----------+
4 rows in set (0.10 sec)

mysql> Select * from customers where Name LIKE 'A%';
+------------+-------+---------+
| CustomerID | Name  | Address |
+------------+-------+---------+
|          3 | Amit  | Punjab  |
|          5 | Ankit | Shimla  |
+------------+-------+---------+
2 rows in set (0.00 sec)
```

Wildcard Character %:

Example:

```
mysql> select * from product;
+---------+-------+
| Name    | Price |
+---------+-------+
| Shivani |    15 |
| Appi    |    30 |
| Raja    |    45 |
| Dheeraj |    60 |
| kumar   |    75 |
+---------+-------+
5 rows in set (0.06 sec)
```

mysql> Select * from product where Name like 'ku%';

```
+-------+-------+
| Name  | Price |
+-------+-------+
| kumar |    75 |
+-------+-------+
```

1 row in set (0.48 sec)

Between operator:

Syntax: `SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2;`

Example:

mysql> select * from product;

```
+---------+-------+
| Name    | Price |
+---------+-------+
| Shivani |    15 |
| Appi    |    30 |
| Raja    |    45 |
| Dheeraj |    60 |
| kumar   |    75 |
+---------+-------+
```

5 rows in set (0.06 sec)

mysql> Select * from product where price between 45 AND 75;

```
+---------+-------+
| Name    | Price |
+---------+-------+
| Raja    |    45 |
| Dheeraj |    60 |
| kumar   |    75 |
+---------+-------+
3 rows in set (0.00 sec)
```