***DDL - Data Definition Language***

* **CREATE :**
* How to create Database:

SYNTAX: create database database\_name;

* How to create Table :

SYNTAX: create table table\_name;

* **DROP:**
* How to drop Database:

SYNTAX: drop database database\_name;

* How to drop table:

SYNTAX: drop table table\_name;

* **ALTER**: to change the definition or to change something in table

Syntax :alter table table\_name {add/drop/modify} col\_name data\_type;

* To add new column:

Syntax : alter table table\_name add col\_name data\_type;

* to delete any column :

Syntax : alter table table\_name drop col\_name;

* To change column:

Syntax : alter table table\_name modify col\_name new\_type;

* How to change column name in table ?

Syntax : alter table table\_name rename column col\_name to new\_name;

* How to change table\_name ?

Syntax : alter table table\_name rename to new\_name;

***DQL – Data Query Language***

* **Select :** to retrieve data from table
* How to show table:

SYNTAX: select \* from table\_name;

* to select particular column value :

SYNTAX: select col1,col2,..,coln from table\_name;

* Show any column:

SYNTAX: select colname from table\_name;

EX : select city from cust\_info;

* not show duplicate values:

SYNTAX: select distinct colname from table\_name

EX: select distinct city from cust\_info;

* Show particular data:

SYNTAX: select colname from table\_name where col\_name like ‘a%’;

EX: select c\_name from cust\_info where c\_name like'a%';

* Orderby:

SYNTAX: select \* from table\_name order by col\_name;

EX: select \* from cust\_info order by cid;

* descending order or ascending order:

SYNTAX: select \* from table\_name order by col\_name DESC;

EX: select \* from cust\_info order by cid DESC;

SYNTAX: select \* from table\_name order by ASC;

EX : select \* from cust\_info order by cid ASC;

* condition :

SYNTAX: select \* from table\_name where col\_name>50000 order by col\_name DESC

EX: select \* from cust\_info where total>50000 order by cid DESC;

***DML – Data Manipulation Language***

* **INSERT :** insert value into table
* How to insert values into table

SYNTAX: insert into table\_name(col1,col2,..,coln) values(val1,val2,..,valn);

* **UPDATE :** update values into table
* How to update :

SYNTAX : update table\_name set col=val where condition;

* **DELETE :** delete any values from the table
* How to delete:

SYNTAX: delete from table\_name where condition;

* **TRUNCTAE :** to remove all the records from the table

Syntax: truncate table table\_name;

***DCL – Data Control Language***

* **GRANT :**
* HOW TO CREATE USER :

SYNTAX: create user 'rutu'@'localhost' identified by 'password';

* TAKE A PERMISSION :

SYNTAX: grant SELECT on abc.customer to 'rutu'@'localhost';

grant INSERT,UPDATE on abc.customer to 'rutu'@'localhost';

grant ALL on abc.customer to 'rutu'@'localhost';

* HOW TO SHOW THE GRANT:

SYNTAX: show grants for 'rutu'@'localhost';

* REVOKE :

SYNTAX: revoke select on database\_name.table\_name from ‘rutu’@’localhost’;

EX: revoke select on abc.customer from 'rutu'@'localhost';

***KEY COSTRAIN***

* **PRIMARY KEY:**
* **How to apply primary key at a time of table creation:**

**SYNTAX: create table movie(id int,name varchar(20),email varchar(20) NOT NULL,DOR date,primary key(id));**

* **How to apply primary key after creation of table ?**

**SYNTAX: alter table table\_name add key\_name key(col\_name);**

* **How to remove primary key after apply to the table ?**

**SYNTAX: alter table table\_name drop key\_name key;**

* **UNIQUE KEY:**
* **How to apply unique key after creation of table ?**

**SYNTAX: alter table table\_name add key\_name key(col1,col2,..,coln);**

* **How to remove primary key after apply to the table ?**

**SYNTAX: alter table table\_name drop index uniquekey column;**

* **FOREIGN KEY:**
* **How to apply foreign key at a time of table creation. (table1 custmer , tabe2 product)**

**SYNTAX:** CREATE TABLE product (  
    productID int NOT NULL PRIMARY KEY,  
    productNumber int NOT NULL,  
    customerID int FOREIGN KEY REFERENCES customer(customerID)  
);

* **How to apply unique key after creation of table ?**

**SYNTAX:** ALTER TABLE product  
ADD FOREIGN KEY (customerID) REFERENCES customer(customerID);

***GROUP BY:***

* **select count(cid) from customer group by city;**
* **select count(cid) from customer where city IN('abd',''goa') group by city;**
* **select city, count(cid),sum(total) AS REVENUE from customer where city IN('abd',''goa') group by city;**
* **select city,count(cid),sum(total) AS REVENUE from customer where SUM(total)>=50000 group by city (FALSE)**
* **select city,count(cid),sum(total) AS REVENUE from customer group by city HAVING(SUM(total)>=50000);**

***FUNCTIONS:***

* **IN :**

**SYNTAX :select \* from table name where city IN(abd,surat, goa);**

* **COUNT():**

**syntax : select count(id) from table name;**

**select count(id) AS total\_customer from customer;**

* **SUM():**

**syntax : select SUM( total) AS REVENUE from customer;**

* **AVG():**

**syntax: select AVG( total) AS REVENUE from customer;**

* **MAX():**

**syntax: select MAX( total) AS REVENUE from customer;**

* **COMMIT**
* **ROLLBACK**
* **SAVEPOINT**
* first we need to set sys's autocommit to zero

Syntax : set autocommit=0;

* COMMIT: to save the changes made on data table Syntax : COMMIT;

* ROLLBACK: to remove changes made on data table

Syntax : ROLLBACK;

* How to rollback upto savepoint only?

Syntax : rollback to savepoint\_name;

* SAVEPOINT: to save changes made on perticular point
* How to create savepoint

Syntax : savepoint savepoint\_name;

**JOINS**

* **Inner Join:**

**Syntax :** SELECT table1.column1, table2.column2...FROM table1INNER JOIN table2ON table1.common\_filed = table2.common\_field;

* **Left Join:**

SYNTAX: SELECT table1.column1, table2.column2...FROM table1LEFT JOIN table2ON table1.common\_filed = table2.common\_field;

* **Right Join:**

SYNTAX: SELECT table1.column1, table2.column2...FROM table1RIGHT JOIN table2Omon\_filed = table2.common\_field;

* **Full Join:**

SYNTAX: SELECT table1.column1, table2.column2...FROM table1FULL JOIN table2 ON table1.common\_filed = table2.common\_field;