**SQL**

DDL :

create new objects(table, view, synonym, procedure, function, trigger) and modify the structure of the objects&Drop the objects from database.

Create: create a new database and tables/objects(We mention size for character data type)

Syntax to create database : create database<DB name>;

EX: Create database Myclass;

Syntax to create table : create table <table name>(<column name1><data type>(size),<column name2><data type(size));

EX : Create table employee(EID int,EName Varchar(10),Sal int);

Alter : alter the structure of data base(adding or removing attributes)

Syntax to add : alter table name\_of\_table add column\_name column\_definition;

EX : alter table employee add father\_name varchar(60);

Syntax to remove :

alter table name\_of\_table drop Column\_Name\_1 , column\_Name\_2

EX : alter table employee drop EID,Sal;

Syntax to modify :

alter table table\_name modify ( column\_name column\_datatype(size));

EX : alter table employee modify (Ename Varchar(10));

Drop : delete the table(total table) but not the data, we can easily remove the entire table, view, or index from the database.

Syntax to drop database : drop database <database\_name>;

EX : Drop database myclass;

Truncate : used to remove all the records of database

Syntax to remove table : truncate <table\_name>;

EX : truncate employee;

To check need to use select command select \* from employee;

SP\_Rename : used to change the name of the database.

Syntax to change : rename table <old\_table\_name to new\_table\_name>;

Ex : rename table employee to employee\_details;

DML(Data Manipulation Language)

Select : Select is the most important data manipulation command in SQL.it shows the records of the specified table and also shows the particular record of a particular column by using the Where clause.

Syntax : Select \* from <table\_name>;

EX : select \* from Employee;

Syntax : Select Emp\_Id, Emp\_Salary from Employee; (it displays the all the values of emp id,emp sal from employee table)

Syntax : select \* from Student where emp\_sal = 80000;

(The where clause displays the values of emp\_sal who are having 80000 from the table)

Insert : which allows users to insert data in database tables.

Syntax : insertinto <table\_name> ( column\_Name1 , column\_Name2 , column\_Name3 )  values (value\_1, value\_2, value\_3) ;

EX :

insert into employee (E\_ID, Emp\_Name, Emp\_Sal) VALUES (107, ‘Nandu’, 80000);

Update : is used to update or modify the existing data in database tables.

Syntax : update <Table\_name> Set [column\_name1= value\_1, ….., column\_nameN = value\_N] where condition;

(Here, 'UPDATE', 'SET', and 'WHERE' are the SQL keywords)

EX : update employee set E\_Id = 1234 where E\_salary =25000;

Syntax to update multiple fields :

EX : update employee set E\_ID = 1234,E\_sal = 25000 where E\_Name = ‘Nandu’ And E\_loc = ‘hyd’;

Delete : used to remove single or multiple existing records from the database tables.

this command of DML does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

Syntax : delete from <tablename> where condition ;

EX : delete from employee where e\_id = 1234;

Syntax to delete multiple records/rows from table:

EX : delete from employee where E\_Id<2345;(delete the employee Id whose ID is less than 2345);

* INNER JOIN:
  + An INNER JOIN returns only the rows that have matching values in both tables.
  + Example:

SELECT customers.name, orders.order\_id

FROM customers

INNER JOIN orders ON customers.customer\_id = orders.customer\_id;

* LEFT JOIN (or LEFT OUTER JOIN):
  + A LEFT JOIN returns all rows from the left table and the matching rows from the right table. If there is no match, NULL values are returned from the right table.
  + Example:

SELECT customers.name, orders.order\_id

FROM customers

LEFT JOIN orders ON customers.customer\_id = orders.customer\_id;

* RIGHT JOIN (or RIGHT OUTER JOIN):
  + A RIGHT JOIN returns all rows from the right table and the matching rows from the left table. If there is no match, NULL values are returned from the left table.
  + Example:

SELECT customers.name, orders.order\_id

FROM customers

RIGHT JOIN orders ON customers.customer\_id = orders.customer\_id;

* FULL JOIN (or FULL OUTER JOIN):
  + A FULL JOIN returns all rows when there is a match in either the left or right table. If there is no match, NULL values are returned from the non-matching side.
  + Example:

SELECT customers.name, orders.order\_id

FROM customers

FULL JOIN orders ON customers.customer\_id = orders.customer\_id;

* SELF JOIN:
  + A SELF JOIN is used to join a table with itself. It is typically used to represent hierarchical data, such as an organizational structure.
  + Example:

SELECT e1.name AS employee, e2.name AS manager

FROM employees AS e1

LEFT JOIN employees AS e2 ON e1.manager\_id = e2.employee\_id;

CROSS JOIN:

* + A CROSS JOIN returns the Cartesian product of two tables, resulting in all possible combinations of rows from both tables.
  + Example:

SELECT customers.name, products.name

FROM customers

CROSS JOIN products;