

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 :

insert into <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;
```