DATABASE

DDL->Data definition language->create,alter,drop,truncate

DML->Insert ,update,delete

DCL-> Grant,reworke

TCL->commit,rollback,savepoint

DQL->

\l->all database

\c->connecting database

\dt ->tables inside the database

Create database -> CREATE DATABASE database\_name;

Create table -> CREATE TABLE table\_name(var datatype);

Print table->SELECT \* FROM table\_name;

Alter table name-> ALTER TABLE old\_table\_name RENAME TO new\_name;

Add colum-> ALTER TABLE table\_name ADD column\_name datatype;

Drop column -> ALTER TABLE table\_name drop column\_name;

Drop table ->DROP TABLE table\_name;

Tuncate table->TRUNCATE TABLE table\_name;

Insert data into table -> INSERT INTO table\_name VALUES(values);

Update the value -> UPDATE table\_name SET column\_name=insert\_value where unick\_colum = unick\_value;

Delete row ->DELETE FROM table\_name WHERE unick\_colum=unick\_value;

Order data->SELECT \* FROM student\_data ORDER BY age (asc/desc);

Printing the limited row -> SELECT \* FROM table\_name LIMIT 3;

Find the null valu -> SELECT \* FROM table\_name WHERE colum\_name ISNULL;

LIKE -> S%,%S,

* SELECT \* FROM table\_name WHERE column\_name LIKE ‘d%’;
* SELECT \* FROM table\_name WHERE column\_name LIKE ‘%d’;

BETWEEN ->RANGE

IN

* SELECT \* FROM table\_name WHERE colum\_name IN('VIJAY','karan');

WHERE ->CONDITION

Sum()

found(Avg(),3)->for decimal value

Min()

Max()

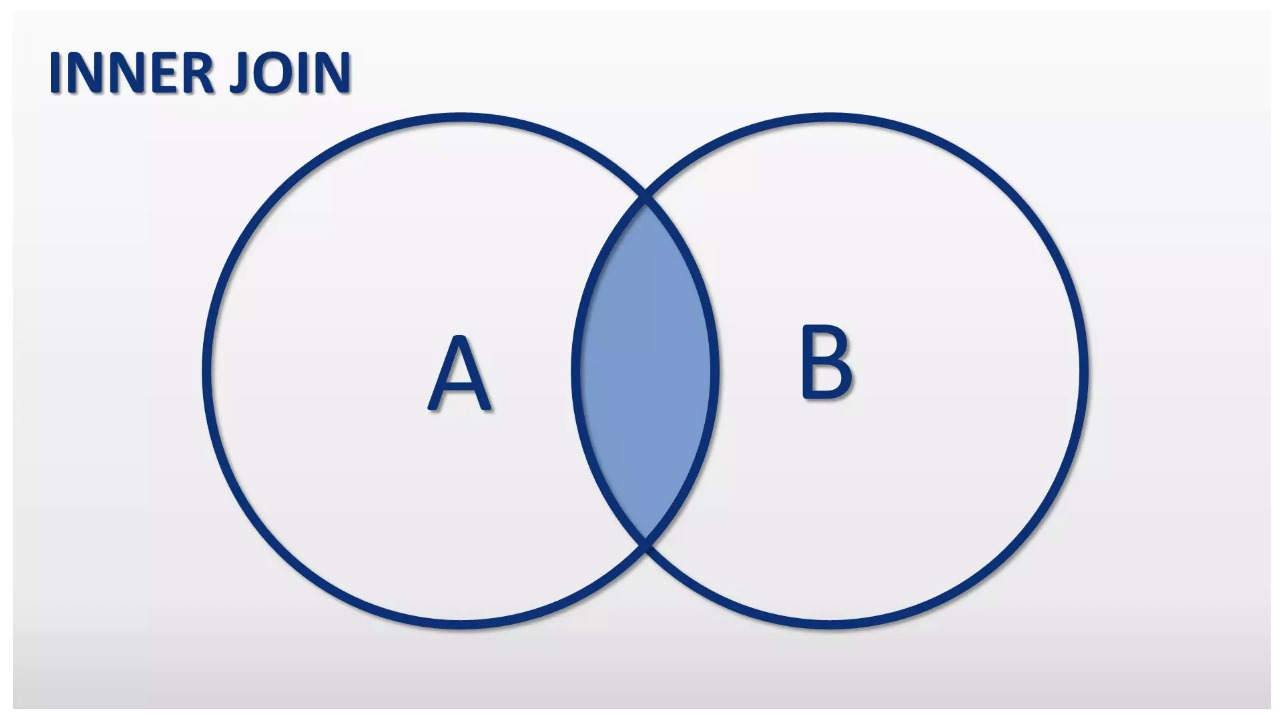
Count()

Group by()->having Is important when using the group by()

**JOINS:**

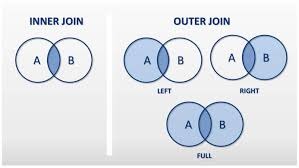
Joins are used to connect the two tables .

Types:

Inner join -> 

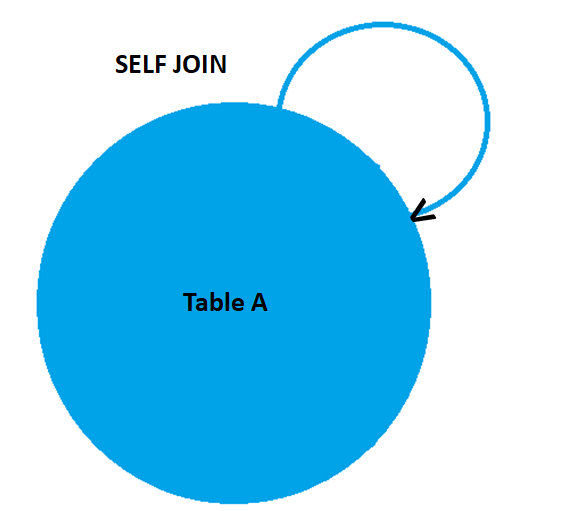
takes the commom fields.

Outer join - >



left outer join , right outer join

Self join->



**Inner join:**

SELECT \* FROM 1sttable\_name JOIN 2ndtable\_name on 1sttable\_name.colum\_name = 2ndtable\_name.colum\_name;

(same syntex for all the joins )

**Ranking :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sal id | Dep | sal | Rank | | |
| 1 | It | 7000 | 1 | 1 | 1 |
| 2 | It | 7000 | 1 | 2 | 1 |
| 3 | It | 5500 | 3 | 3 | 2 |
| 4 | Hr | 6000 | 1 | 1 | 1 |
| 5 | Hr | 5000 | 2 | 2 | 2 |

**Sysntex:**

select (emp\_id,dep,sal),RANK() OVER(PARTITION BY dep ORDER BY sal DESC) AS rank

FROM aids; ->for the type 1

select (emp\_id,dep,sal),DENSE\_RANK() OVER(PARTITION BY dep ORDER BY sal DESC) AS denserank FROM aids; ->for the type 3

select (emp\_id,dep,sal),row\_number() OVER(PARTITION BY dep ORDER BY sal DESC) AS rowmum FROM aids; ->for the type 2