Name: Pratik Sambhaji Pawar

Day:24

**Group By Clause:**

**-**Group rows that have the same values into summary rows.

-It collects data from multiple records and groups the results by one ore more columns

-Generally, we use group-by-clause with aggregation functions

**Count the number of students in each city**

Eg.

SELECT city , count(name)// create new columns

FROM student

GROUP BY city;

**Having Clause:**

**-**Similar to Where i.e. applies some conditions on rows.

-Used when we want to apply any condition after grouping

**Count the number of students in each city where max marks cross 90:**

SELECT count(name),city

FROM student

GROUP BY city

HAVING max(marks)>90;

**General Order :**

***SELECT columns*(s)**

**FROM *table \_name***

**WHERE *conditions***

**GROUP BY *columns***

**HAVING *conditions***

**ORDER BY *columns(s)* ASE;**

**Joins In SQL:**

**-**join is used to combine rows from two or more tables, based on the related column between them.

**Typed of JOINS:**

**Inner Join:**

**-**Return records that have matching values in both tables

Syntax:

SELECT *columns(s)*

FROM *table \_A*

INNER JOIN *table \_B*

ON *table \_A. col \_name=table \_B. col \_name;*

Eg.

SELECT \*

FROM student

INNER JOIN course

ON student. Student \_id=course \_id;

**Left Join:**

**-**Return all records from the left table, and the matched record from the right table.

Syntax:

SELECT *columns(s)*

FROM *table \_A AS a*

LEFT JOIN *table \_B AS b*

ON *a. col \_name=b. col \_name;*

Eg.

SELECT \*

FROM student AS s

LEFT JOIN course AS c

ON s. student \_id=c. cource \_id;

**Right Join:**

**-**Return all records from the right table, and the matched record from the left table.

Syntax:

SELECT *columns(s)*

FROM *table \_A AS a*

RIGHT JOIN *table \_B AS b*

ON *a. col \_name=b. col \_name;*

Eg.

SELECT \*

FROM student AS s

RIGHT JOIN course AS c

ON s. student \_id=c. course \_id;

**Full Join:**

**-**Return all record when there is math either left or right table

Syntax:

SELECT *columns(s)* FROM *table \_A AS a*

LEFT JOIN *table \_B AS b*

ON *a. col \_name=b. col \_name*

UNION

SELECT *columns(s)* FROM *table \_A AS a*

RIGHT JOIN *table \_B AS b*

ON *a. col \_name=b. col \_name;*

Eg.

SELECT \* FROM student AS s

LEFT JOIN course AS c

ON s. student \_id=c. course \_id;

UNIOIN

SELECT \*ROM student AS s

RIGHT JOIN course AS c

ON s. student \_id=c. course \_id;

**Left Exclusive join:**

Syntax:

SELECT *columns(s)* FROM *table \_A AS a*

LEFT JOIN *table \_B AS b*

ON *a. col \_name=b. col \_name*

*WHERE b. col* IS NULL*;*

Eg.

SELECT \*

FROM student AS s

LEFT JOIN course AS c

ON s. student \_id=c. course \_id

WHERE c.id IS NULL ;

**Right Exclusive join:**

Syntax:

SELECT *columns(s)* FROM *table \_A AS a*

RIGHT JOIN *table \_B AS b*

ON *a. col \_name=b. col \_name*

*WHERE b. col* IS NULL*;*

Eg.

SELECT \*

FROM student AS s

RIGHT JOIN course AS c

ON s. student \_id=c. course \_id

WHERE c.id IS NULL ;

**Self Join:-**

-it is a regular join but the table join with itself.

- Syntax:

SELECT *columns(s)*

FROM *table \_A AS a*

JOIN *table AS b*

ON *a. col \_name=b. col \_name;*

Eg.

SELECT a.name as manager \_name ,b. name

FROM employee AS a

JOIN employee AS c

ON a. student \_id=b. course \_id;

**Union:**

**-**it is used to combine the result –set of two or more SELECT sataments

-Gives the UNIQUE records.

Syntax:

SELECT *columns*  FROM *table \_A*

UNION

SELECT *columns* FROM *table \_B*