

Join in SQL

- Join is a query which is used to combine rows from two or more tables or views.
- It retrieves data from multiple table and creates a new table.
- There must be atleast one join condition either in the from clause or in the where clause joining two tables.

Types of join

- Join is of following types
 - (i) Cross join/ cross product join
 - (ii) Inner join
 - (a) Join
 - (b) Equi Join
 - (c) Self Join
 - (d) Natural Join
 - (iii) Outer Join
 - (a) Left outer Join
 - (b) Right outer Join
 - (c) Full outer Join

(i) Cross product Join / Cross Join

Old syntax:-

```
SELECT */COLUMNS FROM TABLE1, TABLE2;
```

New syntax:-

```
SELECT */COLUMNS FROM TABLE1 CROSS JOIN TABLE2;
```

(ii) Inner Join

- The datatype and size of joining attribute must be same among tables.

Old syntax

```
SELECT COLUMNS FROM TABLE1, TABLE2  
WHERE TABLE1.COLUMN OP TABLE2.COL;
```

Example

Find the eid, ename, designation, sal and dno of employees who are working under research department.

Syntax:-

```
SELECT E.EID, E.ENAME, E.DESIGN, E.SAL, D.DNO  
FROM EMP E, DEPT D  
WHERE E.DNO = D.DNO AND  
D.DNAME = 'RESEARCH';
```

New syntax

SELECT COLUMNS FROM TABLE1 T₁ JOIN/INNER JOIN
TABLE2 T₂ ON T₁.COL OP T₂.COL;

Example

Find eid, ename, designation, sal and dno of employees who are working under research department.

Syntax

SELECT E.EID, E.ENAME, E.DEGN, E.SAL, D.DNO
FROM EMP E JOIN DEPT D
ON E.DNO = D.DNO AND D.DNAME = 'RESEARCH';

(iii) Equi join

Syntax

SELECT COLUMNS FROM TABLE1 T₁ JOIN TABLE2 T₂
ON T₁.COLUMN = T₂.COLUMN;

Example

Retrieves the names of department where Mr. John works.

Syntax

SELECT D.DNAME
FROM EMP E JOIN DEPT D
ON E.DNO = D.DNO WHERE E.ENAME = 'JOHN';

(iv) Natural Join

Syntax

```
SELECT COLUMNS  
FROM TABLE1 T1 INNER JOIN TABLE T2;
```

(v) Outer Join

(a) Left outer join

The left outer join returns all rows from the left handside table or the first table specified in the on condition and only those rows from the right handside table where the join condition is met.

- For the non-matching rows of the first table NULL value will be supplied.

Syntax

* Old syntax

```
SELECT COLUMNS  
FROM TABLE1, TABLE2  
WHERE TABLE1.COLUMN = TABLE2.COLUMN(+);
```

* New syntax

```
SELECT COLUMNS  
FROM TABLE1 T1 LEFT OUTER JOIN TABLE2 T2  
ON T1.COLUMN = T2.COL;
```

(b) Right outer join

- It returns all rows from RHS table or the second table specified in the on condition and only those rows from LHS table or first table where join condition is met.
- For non-matching rows, NULL will be supplied to the first table or LHS table.

Syntax

* Old syntax

```
SELECT COLUMNS  
FROM TABLE1, TABLE2  
WHERE TABLE1.COLUMN(+) = TABLE2.COLUMN;
```

* New syntax

```
SELECT COLUMNS  
FROM TABLE1 T1 RIGHT OUTER JOIN TABLE2 T2  
ON T1.COLUMN = T2.COLUMN;
```

(c) Full outer join

- It returns all rows from LHS table and RHS table specified in the on condition.
- It places NULL value where join condition is not met.

Syntax

SELECT COLUMNS

FROM TABLE1 T₁ FULL OUTER JOIN TABLE2 T₂

ON T₁.COLUMN = T₂.COLUMN;

AGGREGATE FUNCTION / GROUP FUNCTION

- These functions are applied on specific groups. The groups are made by "GROUP BY" clause in select statement.
- If "GROUP BY" clause is not specified, then all the record in the table can be taken as group and the group functions are applied accordingly.
- The aggregate functions/group functions are
 - (a) AVG()
 - (b) MIN()
 - (c) MAX()
 - (d) SUM()
 - (e) COUNT()

(a) AVG(column/expression)

- It returns an average of n values by ignoring null values in a column.

Eg:- Find the average salary of employees

```
SELECT AVG(SAL) FROM EMP;
```

(b) MIN(column/expression)

- It returns a minimum value out of n values.

Eg:- Find the minimum salary of all employees

```
SELECT MIN(SAL) FROM EMP;
```

(c) SUM(column/expression)

- It returns the sum of a set of values

Eg:- Find the total expenditure incurred by the company towards salary on monthly basis.

```
SELECT SUM(SAL) FROM EMP;
```

(d) MAX (column/ Expression)

- It returns a maximum value out of n values.

· Eg.:- Find the maximum salary of all employees

SELECT MAX(SAL) FROM EMP;

(e) COUNT()

- It returns number of rows

* COUNT (Expression)

- It returns the no. of rows where expression is not null.

* COUNT(*)

- It returns number of rows in a table including duplicate and those with NULL.

Eg. Find the total number of employees in a company.

SELECT COUNT(EID) "Total employees"
FROM EMP;

OR

SELECT COUNT(*) FROM EMP;

Grouping of data

- Group by clause is used to make different group of rows out of total rows in a table or multiple tables.

"GROUP BY" clause

- "GROUP BY" clause is used with select statements to collect data from multiple records and to group the result by one or more columns.
- "GROUP BY" is used in a table to divide rows into different groups and the groups are treated separately.
- Group function can be applied into individual groups.
- "Having" clause is used withing the group by clause to restrict the groups.

Syntax SELECT EXPRESSION(S)/COLUMN(S),
AGGREGATE FUNCTION

FROM TABLE(S)

WHERE CONDITION

GROUP BY (COLUMN OR EXPRESSION)

HAVING CONDITION;

- The difference between where clause and having clause is that, where clause is used to restrict row, whereas, having clause is used to restrict groups in a group by clause within a select statement.

Eg. Make different groups of employee according to their department number.

⇒ SELECT DNO FROM EMP GROUP BY DNO;

Eg. Make different groups of employee according to their department name.

⇒ SELECT D.DNAME
FROM EMP E JOIN DEPT D
ON E.DNO = D.DNO
GROUP BY D.DNAME;

Eg. Group the employees according to their DNO and count number of employees in each department

⇒ SELECT DNO, COUNT(EID) "TOTAL"
FROM EMP
GROUP BY DNO;

Eg. Find maximum salary in each department

```
=> SELECT DNO, MAX(SAL)
      FROM EMP
      GROUP BY DNO;
```

Eg: Find the DNO where more than 3 employees are present

```
=> SELECT DNO, COUNT(EID)
      FROM EMP
      GROUP BY DNO
      HAVING COUNT(EID) > 3;
```

Eg:- Make different groups of employees according to their salary. The condition is that grouping is done above salary 7000 and each group should contain at least 3 employees.

```
=> SELECT SAL, COUNT(EID)
      FROM EMP
      WHERE SAL > 7000
      GROUP BY SAL
      HAVING COUNT(SID) > 2;
```

Set operation

- The set operations are used to join the results of two (or more) select queries.

Syntax

```
SELECT QUERY1  
SET OPERATION  
SELECT QUERY2;
```

- The select query 1 and select query 2 must be union compatible i.e. the number of columns and the datatype & size of the columns in both the select queries must be same to apply set operation.

(a) UNION

- It returns the combined result of the two select statement.
- It will use one record out of the duplicate records.

$$Q_1 \leftarrow \{t_1, t_2, t_3\}$$

$$Q_2 \leftarrow \{t_3, t_4\}$$

$$Q_1 \text{ UNION } Q_2 \leftarrow \{t_1, t_2, t_3, t_4\}$$

(b) UNION ALL

- It works like union operation but it writes duplicate records in the final result

$$q_1 \leftarrow \{t_1, t_2, t_3\}$$

$$q_2 \leftarrow \{t_3, t_4\}$$

$$q_1 \text{ UNION ALL } q_2 \leftarrow \{t_1, t_2, t_3, t_3, t_4\}$$

(c) INTERSECT

- It lists only the records that are common to the both queries.

$$q_1 \leftarrow \{t_1, t_2, t_3\}$$

$$q_2 \leftarrow \{t_3, t_4\}$$

$$q_1 \text{ INTERSECT } q_2 \leftarrow \{t_3\}$$

(d) MINUS

- It returns the records which are resulted only from the select query 1 but not from the select query 2 i.e. the minus operation remove the second query's result from the output if these are found in first query's result.

$$q_1 \leftarrow \{t_1, t_2, t_3\}$$

$$q_2 \leftarrow \{t_3, t_4\}$$

$$q_1 \text{ MINUS } q_2 \leftarrow \{t_1, t_2\}$$

sub-query or nested query

- A subquery is usually a select query in another select query.
- A sub query can be used with "WHERE", "HAVING" or "FROM" clause from another query.
- The general form of subquery is

```
SELECT QUERY  
OP (SELECT QUERY);
```

Rules:-

- (i) The subquery must be enclosed within a pair of parenthesis.
- (ii) A subquery may return more than one column in most of the cases. If nothing is returned, then the value is NULL.
- (iii) subquery is used in the RHS of conditional or relational operator.
- (iv) Order by clause can't be used in sub-query.

Note:-

Execution method

- In subquery the inner query is executed first and returns value(s) to the outer query.
- The outer query is executed next with the value of the inner query.

Sub-query types

(a) Single row subquery -

A subquery that returns only one row of data is known as single row subquery.

(b) Multiple row subquery -

A subquery that returns multiple rows of data is known as multiple row subquery.

Note:-

The operations "Op" used by outer query is one among the following operations

$=, \neq, >, >=, <, <=$

Eg:- Find EID and ENAME of employees who work under research department.

```
=> SELECT EID, ENAME  
FROM EMP  
WHERE DNO = (SELECT DNO  
FROM DEPT  
WHERE DNAME = 'RESEARCH');
```