

## SQL JOINS

### What is SQL Join?

- ❖ JOIN clause combines rows from two or more tables.
- ❖ creates a set of rows in a temporary table.

### Types of SQL JOIN

- EQUI JOIN
  - EQUI JOIN is a simple SQL join.
  - Uses the equal sign(=) as the comparison operator for the condition
- NON EQUI JOIN
  - NON EQUI JOIN uses comparison operator other than the equal sign.
  - The operators uses like >, <, >=, <= with the condition.

### Types of SQL EQUI JOIN

- INNER JOIN
  - Returns only matched rows from the participating tables.
  - Match happened only at the key record of participating tables.
- OUTER JOIN
  - Returns all rows from one table and
  - Matching rows from the secondary table and
  - Comparison columns should be equal in both the tables.

### List of SQL JOINS

- INNER JOIN
- LEFT JOIN OR LEFT OUTER JOIN
- RIGHT JOIN OR RIGHT OUTER JOIN
- FULL OUTER JOIN
- NATURAL JOIN
- CROSS JOIN
- SELF JOIN

### INNER JOIN

The INNER JOIN selects all rows from both participating tables as long as there is a match between the columns.

An SQL INNER JOIN is same as JOIN clause, combining rows from two or more tables.

Example: INNER JOIN

```
SELECT * FROM table_A
```

INNER JOIN table\_B

ON table\_A.A=table\_B.A;

### **LEFT JOIN or LEFT OUTER JOIN**

The SQL LEFT JOIN, joins two tables and fetches rows based on a condition, which are matching in both the tables.

The unmatched rows will also be available from the table before the JOIN clause.

Example: LEFT JOIN or LEFT OUTER JOIN

SELECT \* FROM table\_A

LEFT JOIN table\_B

ON table\_A.A=table\_B.A;

### **RIGHT JOIN or RIGHT OUTER JOIN**

The SQL RIGHT JOIN, joins two tables and fetches rows based on a condition, which are matching in both the tables.

The unmatched rows will also be available from the table written after the JOIN clause.

Example : RIGHT JOIN or RIGHT OUTER JOIN

SELECT \* FROM table\_A

RIGHT JOIN table\_B

ON table\_A.A=table\_B.A;

### **FULL OUTER JOIN**

Combines the results of both left and right outer joins.

Returns all matched or unmatched rows.

Includes tables on both sides of the join clause.

Example: FULL OUTER JOIN

SELECT \* FROM table\_A

FULL OUTER JOIN table\_B

ON table\_A.A=table\_B.A;

### **NATURAL JOIN**

The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with same name of associate tables will appear once only.

The associated tables have one or more pairs of identically named columns.

The columns must be the same data type.

Don't use ON clause in a natural join.

Example: NATURAL JOIN

SELECT \*

```
FROM table_A  
NATURAL JOIN table_B;
```

### **CROSS JOIN**

The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table, if no WHERE clause is used along with CROSS JOIN.

This kind of result is called as Cartesian Product.

If, WHERE clause is used with CROSS JOIN, it functions like an INNER JOIN.

Example : CROSS JOIN

```
SELECT *  
FROM table_A  
CROSS JOIN table_B;
```

### **SELF JOIN**

A self join is a join in which a table is joined with itself (Unary relationships), specially when the table has a FOREIGN KEY which references its own PRIMARY KEY.

To join a table itself means that each row of the table is combined with itself and with every other row of the table.

The self join can be viewed as a join of two copies of the same table.

Example : SELF JOIN

```
SELECT *  
FROM table_A X, table_A Y  
WHERE X.A=Y.A;
```

Example : INNER JOIN

```
SLECT * FROM table_A  
INNER JOIN table_B  
ON table_A.A=table_B.A;
```

### **LEFT JOIN or LEFT OUTER JOIN**

The SQL LEFT JOIN, joins two tables and fetches rows based on a condition, which are matching in both the tables.

The unmatched rows will also be available from the table before the JOIN clause.

Example : LEFT JOIN or LEFT OUTER JOIN

```
SELECT * FROM table_A  
LEFT JOIN table_B  
ON table_A.A=table_B.A;
```

## **RIGHT JOIN or RIGHT OUTER JOIN**

The SQL RIGHT JOIN, joins two tables and fetches rows based on a condition, which are matching in both the tables.

The unmatched rows will also be available from the table written after the JOIN clause.

Example : RIGHT JOIN or RIGHT OUTER JOIN

```
SELECT * FROM table_A  
RIGHT JOIN table_B  
ON table_A.A=table_B.A;
```

## **FULL OUTER JOIN**

In SQL the FULL OUTER JOIN combines the results of both left and right outer joins and returns all (matched or unmatched) rows from the tables on both sides of the join clause.

Example : FULL OUTER JOIN

```
SELECT * FROM table_A  
FULL OUTER JOIN table_B  
ON table_A.A=table_B.A;
```

## **NATURAL JOIN**

The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with same name of associate tables will appear once only.

The associated tables have one or more pairs of identically named columns. The columns must be the same data type. Don't use ON clause in a natural join.

Example : NATURAL JOIN

```
SELECT *  
FROM table_A  
NATURAL JOIN table_B;
```

## **CROSS JOIN**

The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table, if no WHERE clause is used along with CROSS JOIN. This kind of result is called as Cartesian Product.

If, WHERE clause is used with CROSS JOIN, it functions like an INNER JOIN.

Example: CROSS JOIN

```
SELECT *  
FROM table_A  
CROSS JOIN table_B;
```

## **SELF JOIN**

A self join is a join in which a table is joined with itself (Unary relationships), specially when the table has a FOREIGN KEY which references its own PRIMARY KEY.

To join a table itself means that each row of the table is combined with itself and with every other row of the table. The self join can be viewed as a join of two copies of the same table.

Example : SELF JOIN

```
SELECT *
```

```
FROM table_A X, table_A Y
```

```
WHERE X.A=Y.A;
```