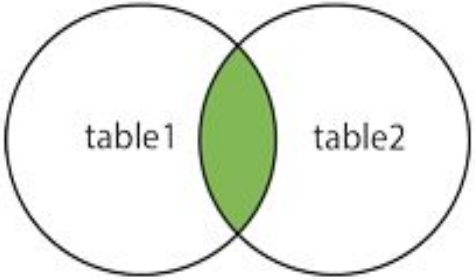
SQL JOINs

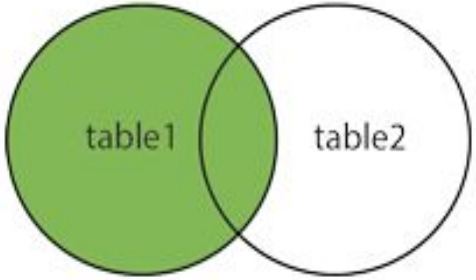
Different Types of SQL JOINs

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table

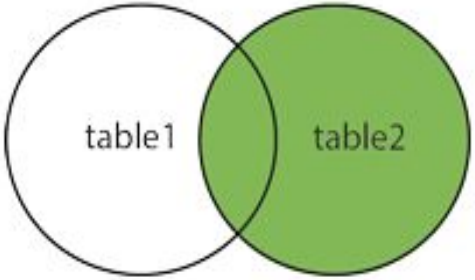
INNER JOIN



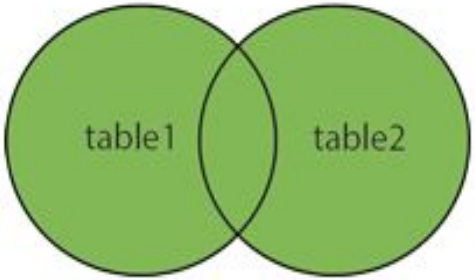
LEFT JOIN

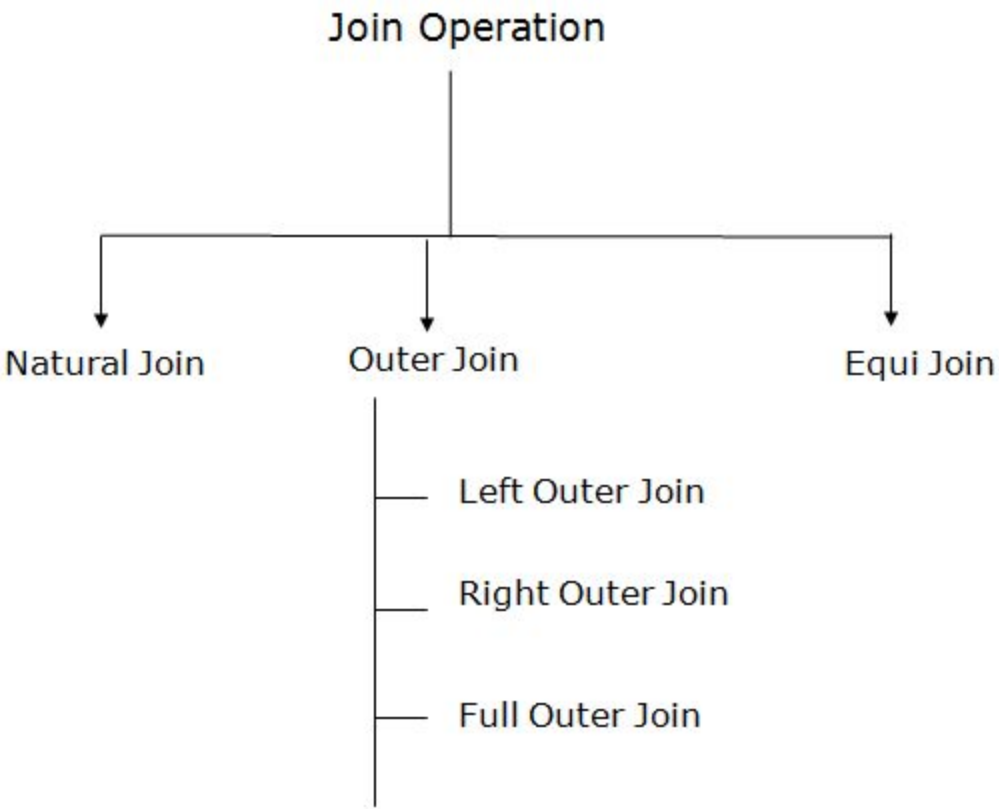


RIGHT JOIN



FULL OUTER JOIN




Join Operation
Natural Join
Outer Join
Equi Join
Left Outer Join
Right Outer Join
Full Outer Join

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

PROJECT

PROJECT_NO	EMP_ID	DEPARTMENT
101	1	Testing
102	2	Development
103	3	Designing
104	4	Development

INNER JOIN

In SQL, INNER JOIN selects records that have matching values in both tables as long as the condition is satisfied. It returns the combination of all rows from both the tables where the condition satisfies.

Syntax:

```
SELECT table1.column1, table1.column2, table2.column1,....
 FROM table1
 INNER JOIN table2
 ON table1.matching_column = table2.matching_column;
```

Example:

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT
 FROM EMPLOYEE
 INNER JOIN PROJECT
 ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development

Natural Join:

- A natural join is the set of tuples of all combinations in R and S that are equal on their common attribute names.
- Select colnames
- From table1 natural join table2;

Equi join:

It is also known as an inner join. It is the most common join. It is based on matched data as per the equality condition. The equi join uses the comparison operator(=).

Syntax :

```
SELECT column_list

FROM table1, table2....

WHERE table1.column_name =

table2.column_name;
```

Example –

```
SELECT student.name, student.id, record.class, record.city

FROM student, record

  WHERE student.city = record.city;
```

THETA JOIN allows you to merge two tables based on the condition represented by theta. Theta joins work for all comparison operators.

Syntax:

```
SELECT *

FROM table_name1, table_name2

WHERE table_name1.column [> | = | < |  >= | <= ] table_name2.column;
```

Example –

```
SELECT student.name, record.id, record.city

FROM student, record

  WHERE Student.id < Record.id
```

Outer Join:

The outer join operation is an extension of the join operation. It is used to deal with missing information.

Left outer join:

- **LEFT JOIN** returns all the rows from the table on the left even if no matching rows have been found in the table on the right. When no matching record found in the table on the right, NULL is returned.

Example: Let's use the above EMPLOYEE table and SALARY table:

```
SELECT table1.column1, table1.column2, table2.column1,....

FROM table1

LEFT JOIN table2

ON table1.matching_column = table2.matching_column;
```

Example

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT

FROM EMPLOYEE

LEFT JOIN PROJECT

ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development
Russell	NULL
Marry	NULL

```
SELECT
  order_id,
  status,
  first_name,
  last_name
FROM
  orders
LEFT JOIN employees ON employee_id = salesman_id
ORDER BY
  order_date DESC;
```

Example 2:

```
SELECT  order_id, status, first_name, last_name
FROM orders
LEFT JOIN employees ON employee_id = salesman_id
ORDER BY order_date DESC;
```

The following picture illustrates the result:

ORDER_ID	STATUS	FIRST_NAME	LAST_NAME
88	Shipped	Daisy	Ortiz
94	Shipped	Freya	Gomez
1	Pending	Evie	Harrison
14	Shipped	(null)	(null)
17	Shipped	(null)	(null)
15	Shipped	(null)	(null)
36	Shipped	(null)	(null)
57	Shipped	Scarlett	Gibson
28	Canceled	Scarlett	Gibson
29	Shipped	(null)	(null)
30	Shipped	(null)	(null)
31	Canceled	(null)	(null)
60	Shipped	Freya	Gomez

Right outer join:

- **RIGHT JOIN** returns all the columns from the table on the right even if no matching rows have been found in the table on the left. Where no matches have been found in the table on the left, NULL is returned.

Syntax:

```
SELECT table1.column1, table1.column2, table2.column1,....

FROM table1

RIGHT JOIN table2

ON table1.matching_column = table2.matching_column;
```

Example:

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT

FROM EMPLOYEE

RIGHT JOIN PROJECT

ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development

```
SELECT first_name, last_name,order_id, status
FROM orders
RIGHT JOIN employees ON
    employee_id = salesman_id
WHERE job_title = 'Sales Representative'
ORDER BY first_name, last_name;
```

FIRST_NAME	LAST_NAME	ORDER_ID	STATUS
Alice	Wells	(null)	(null)
Charlotte	Webb	(null)	(null)
Chloe	Cruz	51	Shipped
Chloe	Cruz	55	Pending
Chloe	Cruz	92	Shipped
Chloe	Cruz	95	Shipped
Chloe	Cruz	41	Shipped
Chloe	Cruz	4	Shipped
Chloe	Cruz	59	Shipped
Daisy	Ortiz	102	Shipped

The result includes all employees whose job title is **Sales Representative** and their orders.

If a salesman is not in charge of any sales order such as Alice Wells, Charlotte Webb, the order_id and status columns are filled with NULL values.

Full outer join:

- Full outer join is like a left or right join except that it contains all rows from both tables.
- In a **FULL OUTER JOIN** , all tuples from both relations are included in the result, irrespective of the matching condition.

Syntax:

```
SELECT table1.column1, table1.column2, table2.column1,....

FROM table1

FULL JOIN table2

ON table1.matching_column = table2.matching_column;
```

Example:

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT

FROM EMPLOYEE

FULL JOIN PROJECT

ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development
Russell	NULL
Marry	NULL