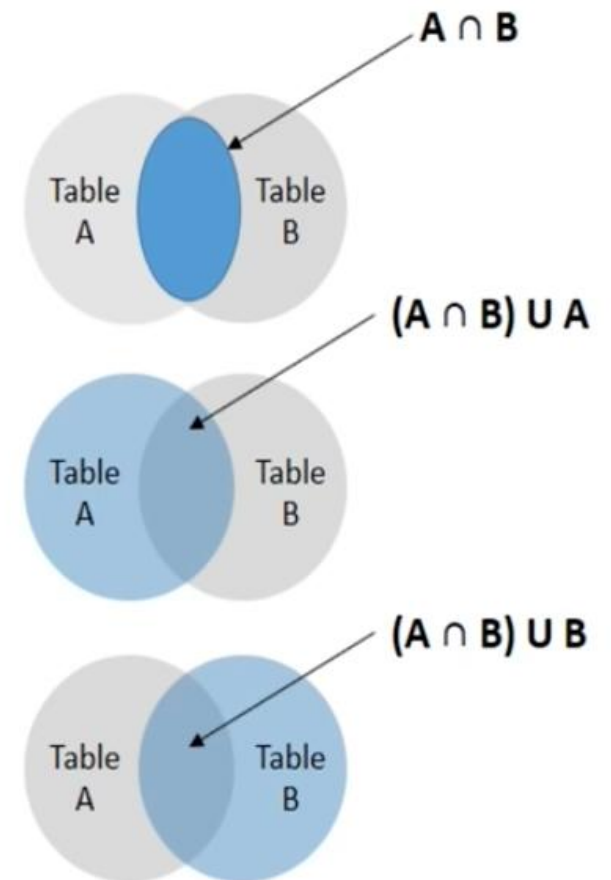


# SQL JOINS

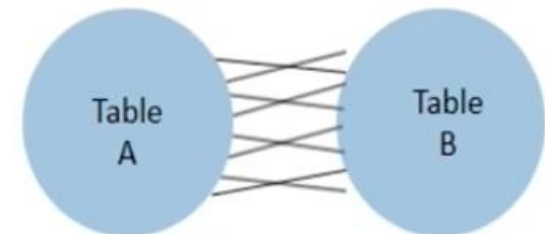
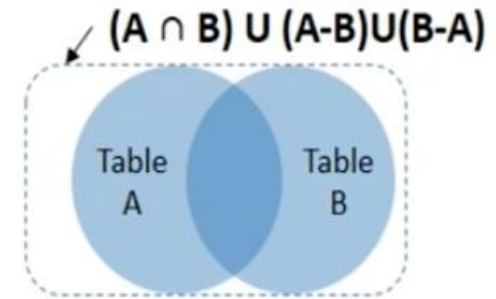
Combine rows/columns from two or more tables, based on a related column between them in a database

- **INNER JOIN** – Returns rows when there is a match in both tables.
- **LEFT JOIN** – Returns all rows from the left table, even if there are no matches in the right table.
- **RIGHT JOIN** – Returns all rows from the right table, even if there are no matches in the left table.



# SQL JOINS

- **FULL OUTER JOIN** – Returns rows when there is a match in one of the tables.
- **SELF JOIN** – Used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.
- **CARTESIAN JOIN (CROSS JOIN)** – Returns the Cartesian product of the sets of records from the two or more joined tables.



## SQL INNER JOIN

The INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. The query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate.

### SYNTAX :

```
SELECT table1.col1, table2.col2,..., table1.coln  
FROM table1  
INNER JOIN table2  
ON table1.commonfield = table2.commonfield;
```

## SQL INNER JOIN

emp_id	first_name e	last_name e	salary	dept_id
103	Harry	Potter	20000	12
102	Edwin	Thomas	15000	11
101	Steven	Cohen	10000	10
100	Erik	John	10000	12

dept_id	dept_name	manager_id	location_id
10	IT	200	1700
11	Marketing	201	1800
13	Resources	203	2400
14	Shipping	121	1500

```
SELECT e.emp_id, e.first_name,  
       e.last_name, d.dept_id,  
       d.dept_name  
FROM employees e  
INNER JOIN departments d  
ON e.dept_id=d.dept_id;
```

emp_id	first_name	last_name	dept_id	dept_name
101	Steven	Cohen	10	IT
102	Edwin	Thomas	11	Marketing

## SQL RIGHT JOIN

- The RIGHT JOIN returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.

SYNTAX :

```
SELECT table1.col1, table2.col2,..., table1.coln  
FROM table1  
RIGHT JOIN table2  
ON table1.commonfield = table2.commonfield;
```

## SQL FULL OUTER JOIN

The FULL OUTER JOIN combines the results of both left and right outer joins. The joined table will contain all records from both the tables and fill in NULLs for missing matches on either side.

### SYNTAX :

```
SELECT table1.col1, table2.col2,..., table1.coln  
FROM table1  
Left JOIN table2  
ON table1.commonfield = table2.commonfield;  
Union  
SELECT table1.col1, table2.col2,..., table1.coln  
FROM table1  
Right JOIN table2  
ON table1.commonfield = table2.commonfield;
```



## SQL FULL OUTER JOIN

emp_id	first_name	last_name	salary	dept_id
103	Harry	Potter	20000	12
102	Edwin	Thomas	15000	11
101	Steven	Cohen	10000	10
100	Erik	John	10000	12

dept_id	dept_name	manager_id	location_id
10	IT	200	1700
11	Marketing	201	1800
13	Resources	203	2400
14	Shipping	121	1500

```

SELECT e.emp_id, e.first_name, e.last_name, d.dept_id,
d.dept_name
FROM employees e
LEFT JOIN departments d
ON e.dept_id=d.dept_id
UNION
SELECT e.emp_id, e.first_name, e.last_name, d.dept_id,
d.dept_name
FROM employees e
RIGHT JOIN departments d
ON e.dept_id=d.dept_id;

```

emp_id	first_name	last_name	dept_id	dept_name
101	Steven	Cohen	10	IT
102	Edwin	Thomas	11	Marketing
103	Harry	Potter	Null	Null
100	Erik	John	Null	Null
Null	Null	Null	13	Resources

## SQL LEFT JOIN

The LEFT JOIN returns all the values from the left table, plus matched values from the right table or NULL in case of no matching join predicate.

### SYNTAX :

```
SELECT table1.col1, table2.col2,..., table1.coln  
FROM table1  
LEFT JOIN table2  
ON table1.commonfield = table2.commonfield;
```



## SQL LEFT JOIN

emp_id	first_name e	last_name e	salary	dept_id
103	Harry	Potter	20000	12
102	Edwin	Thomas	15000	11
101	Steven	Cohen	10000	10
100	Erik	John	10000	12

dept_id	dept_name	manager_id	location_id
10	IT	200	1700
11	Marketing	201	1800
13	Resources	203	2400
14	Shipping	121	1500

```
SELECT e.employee_id,
e.first_name, e.last_name,
d.dept_id, d.dept_name
FROM employees e
LEFT OUTER JOIN departments d
ON e.dept_id = d.dept_id;
```

emp_id	first_name	last_name	dept_id	dept_name e
101	Steven	Cohen	10	IT
102	Edwin	Thomas	11	Marketing
103	Harry	Potter	Null	Null
100	Erik	John	Null	Null



## SQL RIGHT JOIN

emp_id	first_name e	last_name e	salary	dept_id
103	Harry	Potter	20000	12
102	Edwin	Thomas	15000	11
101	Steven	Cohen	10000	10
100	Erik	John	10000	12

dept_id	dept_name	manager_id	location_id
10	IT	200	1700
11	Marketing	201	1800
13	Resources	203	2400
14	Shipping	121	1500

```
SELECT e.emp_id, e.first_name, e.last_name,
       d.dept_id, d.dept_name
FROM employees e
RIGHT JOIN departments d
ON e.dept_id=d.dept_id;
```

emp_id	first_name	last_name	dept_id	dept_name e
101	Steven	Cohen	10	IT
102	Edwin	Thomas	11	Marketing
Null	Null	Null	13	Resources
Null	Null	Null	14	Shipping



## SQL CROSS JOIN

- The CROSS JOIN produces a result set with the number of rows in the first table multiplied by the number of rows in the second.

### SYNTAX:

```
SELECT table1.col1, table2.col2,..., table1.coln  
FROM table1  
CROSS JOIN table2;
```

# SQL CROSS JOIN

- Assume there are 4 records in table1 and 3 records in table2

```
SELECT * FROM table1  
CROSS JOIN table2;
```

Table-1

alpha
A
B
C
D

Table-2

Num
1
2
3

A	1
A	2
A	3
B	1
B	2
B	3
C	1
C	2
C	3
D	1
D	2
D	3