# RIGHT JOIN

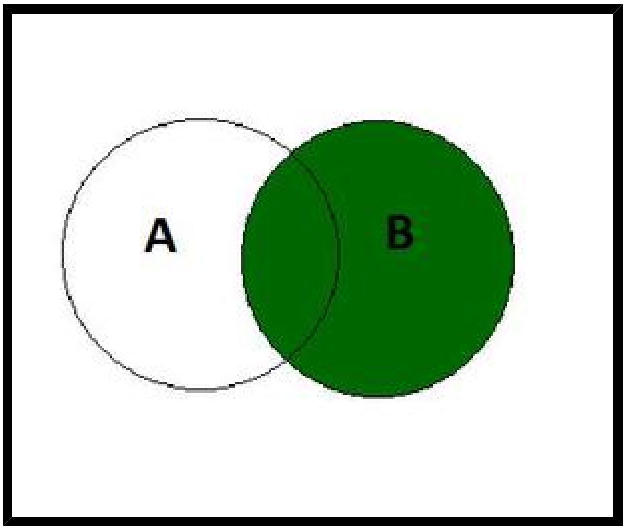
The **SQL RIGHT JOIN** Keyword is a powerful tool used to combine records from two tables. SQL RIGHT JOIN returns all records from the right table, and the matching records from the left table in the results set.

## SQL RIGHT JOIN Keyword

The **RIGHT JOIN in SQL** returns a table that contains all the records from the right table and only matching records from the left table.

In simpler terms, if a row is present in the right table but not in the left table, the result will include this row with NULL values for columns from the left table. Conversely, if a record from the left table is not in the right table, it will not be included in the result.

The **Visual Representation of RIGHT JOIN** is shown below in the **Venn Diagram**.



*RIGHT JOIN*

**Syntax**

```
SELECT column_name(s)
FROM tableA
RIGHT JOIN tableB
ON tableA.column_name = tableB.column_name;
```

## SQL RIGHT JOIN Examples

In this example, we will consider two tables employee table containing details of the **employees** working in the particular department the and **department** table containing the details of the department

### Employee Table

| emp_no | emp_name | dept_no |
|---|---|---|

| | | |
|---|---|---|
| E1 | Varun Singhal | D1 |
| E2 | Amrita Aggarwal | D2 |
| E3 | Ravi Anand | D3 |

## SQL Query to create employee table:

```
CREATE TABLE employee (
  emp_no CHAR(3) PRIMARY KEY,  -- Adjust length if needed
for employee numbers
  emp_name VARCHAR(50) NOT NULL,
  dept_no CHAR(2)
);

INSERT INTO employee (emp_no, emp_name, dept_no)
VALUES ('E1', 'Varun Singhal', 'D1'),
('E2', 'Amrita Aggarwal', 'D2'),
('E3', 'Ravi Anand', 'D3');
```

## Department Table

| dept_no | d_name | location |
|---|---|---|
| D1 | IT | Delhi |
| D2 | HR | Hyderabad |
| D3 | Finance | Pune |
| D4 | Testing | Noida |
| D5 | Marketing | Mathura |

## SQL Query to Create department table:

```
CREATE TABLE department (
  dept_no CHAR(2) PRIMARY KEY,
  d_name VARCHAR(20) NOT NULL,
  location VARCHAR(50)
);
INSERT INTO department (dept_no, d_name, location)
VALUES ('D1', 'IT', 'Delhi'),
       ('D2', 'HR', 'Hyderabad'),
       ('D3', 'Finance', 'Pune'),
       ('D4', 'Testing', 'Noida'),
       ('D5', 'Marketing', 'Mathura');
```

Now, we will perform SQL RIGHT JOIN on these two tables.

**Query**

```
SELECT emp_no, emp_name, d_name, location
FROM employee
RIGHT JOIN department
ON employee.dept_no = department.dept_no;
```

**Output**

| emp_no | emp_name | d_name | location |
|---|---|---|---|
| E1 | Varun Singhal | IT | Delhi |
| E2 | Amrita Aggarwal | HR | Hyderabad |
| E3 | Ravi Anand | Finance | Pune |
| | | Testing | Noida |
| | | Marketing | Mathura |

**Explanation**: As right join gives the matching rows and the rows that are present in the right table but not in the left table. Here in this example, we see that the department that contains no employee contains **[NULL] values of emp_no** and emp_name after performing the right join.

## Applications of SQL RIGHT JOIN

- **Merging Data**: Allows to merge data from different tables in database.
- **Ensuring Completeness**: A **RIGHT JOIN** ensures that all records from the **right table** are included in the result, even if there are no corresponding matches in the **left table**.
- **Handling Missing Values**: Allows to look for missing values in one of the table. For example, combining customer and orders table allows to look at customers and their orders.
- **Analyzing Relationships**: Useful in finding patterns and relations between data.

## Important Points About SQL RIGHT JOIN

- Right JOIN allows to join two table, keeping all the data or right table and only matching data of left table.
- Right JOIN is a type of outer join in SQL.
- It allows us to deal with missing values in database and also helps in analyzing relationships between data.
- Simplifies queries by eliminating the need for complex conditional logic using CASE statements.