|  |
| --- |
| **SQL Joins and Operators** |
| LAB MANUAL 05 |

## SQL JOIN:

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

## Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

* **(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

      

## [SQL INNER JOIN Keyword](https://www.w3schools.com/sql/sql_join_inner.asp)

The INNER JOIN keyword selects records that have matching values in both tables.

### **INNER JOIN Syntax**

|  |
| --- |
| SELECT column\_name(s) FROM table1 INNER JOIN table2ON table1.column\_name = table2.column\_name; |

**Example:**

|  |
| --- |
| SELECT      first\_name,      last\_name,      employees.department\_id,      departments.department\_id,      department\_name  FROM      employees          INNER JOIN      departments ON departments.department\_id = employees.department\_id |

## [SQL LEFT JOIN Keyword](https://www.w3schools.com/sql/sql_join_inner.asp)

## A LEFT JOIN statement returns all rows from the left table along with the rows from the right table for which the join condition is met.

|  |
| --- |
| 1. SELECT t1.emp\_id, t1.emp\_name, t1.hire\_date, t2.dept\_name 2. FROM employees AS t1 LEFT JOIN departments AS t2 3. ON t1.dept\_id = t2.dept\_id ORDER BY emp\_id; |

## [SQL RIGHT JOIN Keyword](https://www.w3schools.com/sql/sql_join_inner.asp)

## The RIGHT JOIN is the exact opposite of the [LEFT JOIN](https://www.tutorialrepublic.com/sql-tutorial/sql-left-join-operation.php). It returns all rows from the right table along with the rows from the left table for which the join condition is met.

|  |
| --- |
| 1. SELECT t1.emp\_id, t1.emp\_name, t1.hire\_date, t2.dept\_name 2. FROM employees AS t1 RIGHT JOIN departments AS t2 3. ON t1.dept\_id = t2.dept\_id ORDER BY dept\_name; |

## Full Joins

A FULL JOIN returns all the rows from the joined tables, whether they are matched or not i.e. you can say a full join combines the functions of a [LEFT JOIN](https://www.tutorialrepublic.com/sql-tutorial/sql-left-join-operation.php) and a [RIGHT JOIN](https://www.tutorialrepublic.com/sql-tutorial/sql-right-join-operation.php). Full join is a type of [outer join](https://www.tutorialrepublic.com/sql-tutorial/sql-joining-tables.php#outer-join)that's why it is also referred as full outer join.

|  |
| --- |
| 1. SELECT t1.emp\_id, t1.emp\_name, t1.hire\_date, t2.dept\_name 2. FROM employees AS t1 FULL JOIN departments AS t2 3. ON t1.dept\_id = t2.dept\_id ORDER BY emp\_name; |

The following example updates the **stock** table by increasing the **unit\_price** value by 10% for a subset of prices. The WHERE clause specifies which prices to increase by applying the IN operator to the rows returned by a subquery that selects only the rows of the **stock** table where the **unit\_price** value is less than 75.

UPDATE stock SET unit\_price = unit\_price \* 1.1

WHERE unit\_price IN

(SELECT unit\_price FROM stock WHERE unit\_price < 75);