**SQL IN OPERATOR:**

**Syntax #1:**

SELECT column\_name(s)

FROM table\_name

WHERE test\_expression IN (value1, value2, ...);

**Syntax #2**

SELECT column\_name(s)

FROM table\_name

WHERE test\_expression IN (SELECT subquery);

**1. Understanding the basic function of IN operator**

Let us see some basic functions:



**Example #1**

Find the details of employees having employee ids as 10023, 10030 and 10026.

**Code:**

SELECT \*

FROM employees

WHERE employeeid in (10023,10030 ,10026);

**Output:**



**Example #2**

Find the employeeid, department\_id, salary and city of employees from Oslo and New Delhi.

**Code:**

SELECT employeeid,departmentid, salary,city

FROM employees

WHERE CITY IN ('OSLO','NEW DELHI');

**Output:**



**Example 3:**

**Code:**

SELECT employeeid,departmentid, salary,city

FROM employees

WHERE CITY NOT IN ('OSLO','NEW DELHI');

**OUTPUT:**



**2. Using IN with BETWEEN expression operator**

IN expression operator is very frequently used along with BETWEEN operator in the WHERE clause. It is used to specify an additional condition in the query. IN filters only those rows/records which are present in the mentioned set.

**Example #1**

Find the names of the employees , who joined the company between 2003 and 2006 and are from Manhattan or New Delhi.

**Code:**

SELECT firstname,lastname, create\_at, city

FROM employees

WHERE create\_at BETWEEN '2003-01-01' AND '2006-12-31'

AND city IN ('Manhattan', 'New Delhi');

**Output:**



**3. Using IN operator with subqueries**

IN logical operator can be used to check if a value exists/matches with the results of a sub query or not.

**Example #1**

Find the names of the employees who are head of a department in the company.

**Code:**

SELECT employeeid, firstname, lastname, departmentid, city

FROM employees

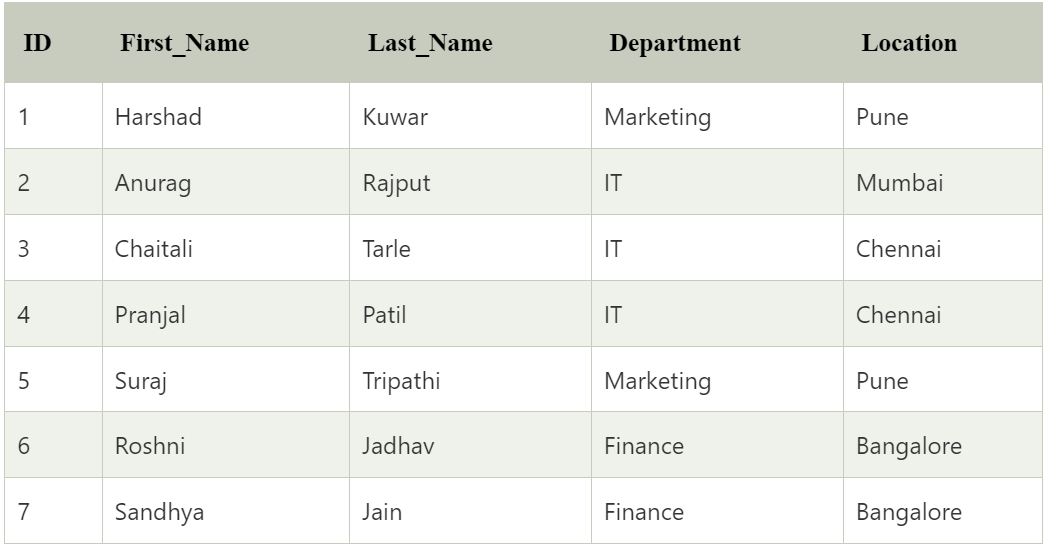
WHERE employeeid IN (SELECT head FROM department);

**Output:**



### SQL "AND" example with "SELECT" statement

This is how an SQL "****AND****" condition can be used in the SQL ****SELECT**** statement.

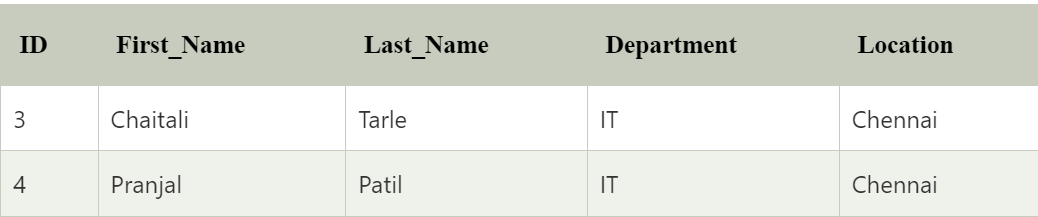


****Example 1:****

Write a query to get the records from emp tables in which department of the employee is IT and location is Chennai.

****Query:****

mysql> **SELECT** \***FROM** emp **WHERE** Department = "IT" AND Location = "Chennai";

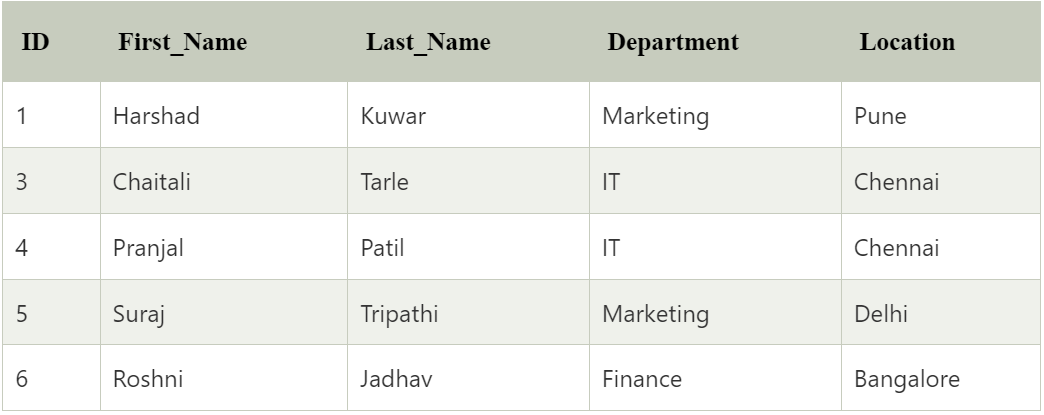


****SQL "AND" example with "UPDATE" statement:****

Write a query to update the records in the emp table in which department of the employee is Finance and ID is 7. For that particular employee, set the updated value of the department as HR.

****Query:****

mysql> **UPDATE** emp **SET** Department = "HR" **WHERE** Department = "Finance" AND ID = 7;

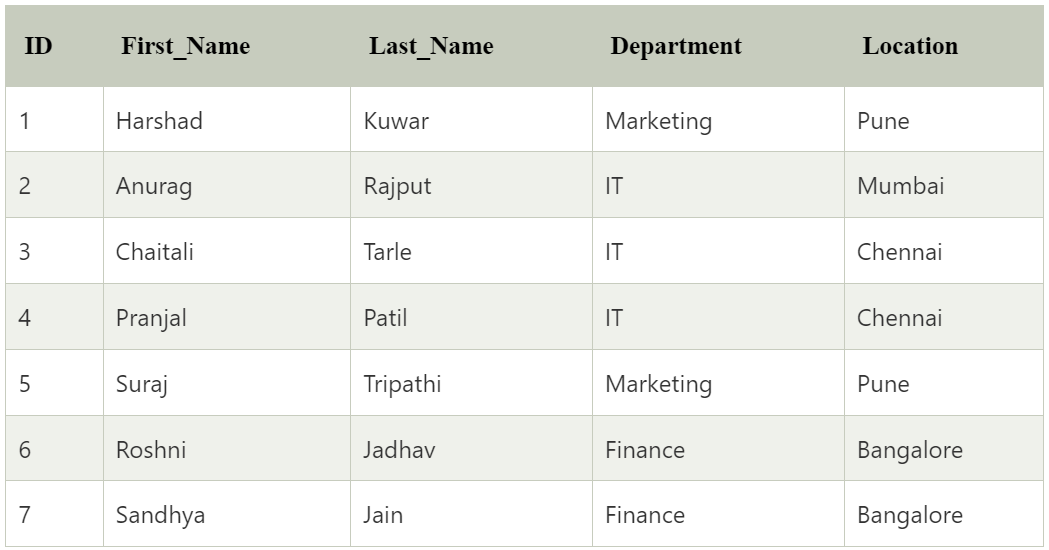


# SQL OR

The SQL ****OR**** condition is used in SQL query to create a SQL statement where records are returned when any one condition met. It can be used in a ****SELECT**** statement, ****INSERT**** statement, ****UPDATE**** statement or ****DELETE**** statement.

****Let's see the syntax for the OR condition:****

**SELECT** columns **FROM** tables **WHERE** condition 1 OR condition 2;

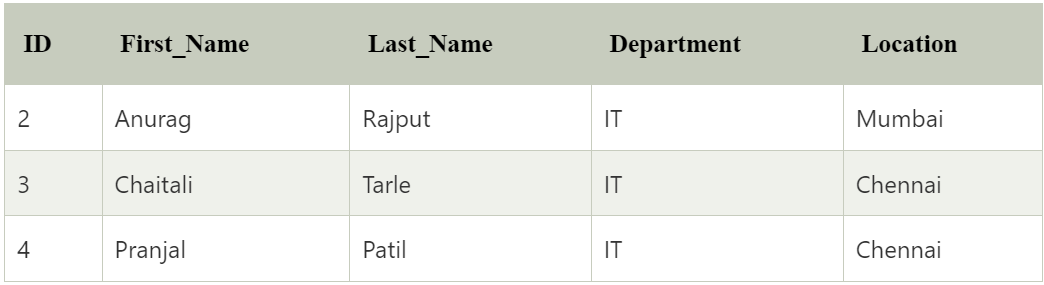


****Example 1:****

Write a query to get the records from emp tables in which department of the employee is IT or location is Chennai.

****Query:****

**SELECT** \***FROM** emp **WHERE** Department = "IT" OR Location = "Chennai";



# SQL SELECT AS :

* SQL '****AS'**** is used to assign a new name temporarily to a table column or even a table.

**SYNTAX:**

**SELECT** Column\_Name1 **AS** New\_Column\_Name, Column\_Name2  **As** New\_Column\_Name **FROM** Table\_Name.

**Example:**

****Assigning a temporary name to the column of a table:****

Let us take a table named orders, and it contains the following data:

