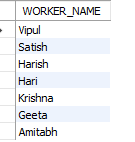
|  |
| --- |
| Create database Assignment;  CREATE TABLE Worker (  WORKER\_ID INT PRIMARY KEY,  FIRST\_NAME VARCHAR (50),  LAST\_NAME VARCHAR (50),  SALARY INT,  JOINING\_DATE DATE,  DEPARTMENT VARCHAR (50)  );  desc Workers;  show databases;  show tables;  INSERT INTO Worker (WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE, DEPARTMENT)  VALUES  (1, 'Vipul', 'Shah', 500000, '2014-02-15', 'Admin'),  (2, 'Satish', 'Tiwari', 600000, '2019-03-22', 'Finance'),  (3, 'Harish', 'Pandit', 55000, '2020-04-12', 'IT'),  (4, 'Hari', 'Kunwar', 70000, '2015-03-12', 'Marketing'),  (5, 'Krishna', 'Pandit', 545000, '2019-04-12', 'IT'),  (6, 'Geeta', 'Kuwari', 100000, '2014-02-12', 'Marketing'),  (7, 'Amitabh', 'Kuwari', 100000, '2015-03-12', 'Admin')  ; |

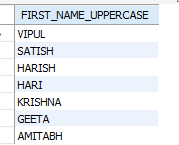
1. **Write an SQL query to fetch “FIRST\_NAME” from Worker table using the alias name as <WORKER\_NAME>.**

****

SELECT FIRST\_NAME AS WORKER\_NAME

FROM Worker;

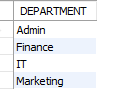
1. **Write an SQL query to fetch “FIRST\_NAME” from Worker table in upper case.**



SELECT UPPER(FIRST\_NAME) AS FIRST\_NAME\_UPPERCASE

FROM Worker;

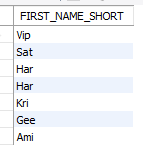
1. **Write an SQL query to fetch unique values of DEPARTMENT from Worker table.**

****

SELECT DISTINCT DEPARTMENT

FROM Worker;

1. **Write an SQL query to print the first three characters of FIRST NAME from Workers table.**



SELECT SUBSTRING (FIRST\_NAME, 1, 3) AS FIRST\_NAME\_SHORT

FROM Worker;

1. **Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Amitabh’ from Worker table.**

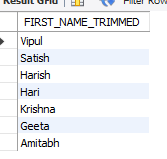


SELECT LOCATE ('a', FIRST\_NAME) AS POSITION\_A

FROM Worker

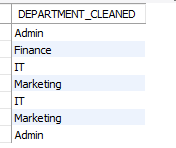
WHERE FIRST\_NAME = 'Amitabh';

1. **Write an SQL query to print the FIRST\_NAME from Worker table after removing white spaces from the right side.**



SELECT RTRIM(FIRST\_NAME) AS FIRST\_NAME\_TRIMMED

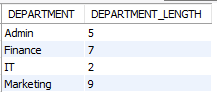
FROM Worker; WHERE FIRST\_NAME = 'Amitabh';

1. **Write an SQL query to print the DEPARTMENT from Worker table after removing white spaces from the left side.**

SELECT LTRIM(DEPARTMENT) AS DEPARTMENT\_CLEANED

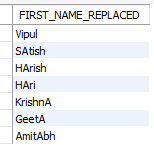
FROM Worker;

1. **Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.**



SELECT DISTINCT DEPARTMENT, LENGTH(DEPARTMENT) AS DEPARTMENT\_LENGTH

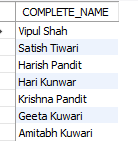
FROM Worker;

1. **Write an SQL query to print the FIRST\_NAME from Worker table after replacing ‘a’ with ‘A’.**

SELECT REPLACE (FIRST\_NAME, 'a', 'A') AS FIRST\_NAME\_REPLACED

FROM Worker;

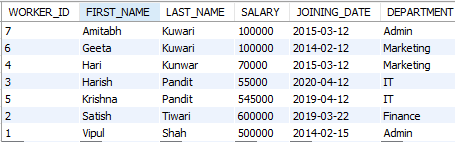
1. **Write an SQL query to print the FIRST\_NAME and LAST\_NAME from Worker table into a single column COMPLETE\_NAME. A space char should separate them.**



SELECT CONCAT (FIRST\_NAME, ' ', LAST\_NAME) AS COMPLETE\_NAME

FROM Worker;

1. **Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending.**

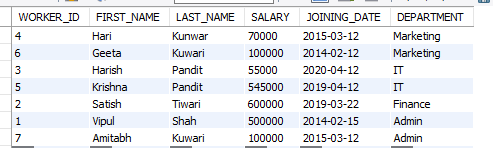


SELECT \*

FROM Worker

ORDER BY FIRST\_NAME ASC;

1. **Write an SQL query to print all Worker details from the Worker table order by DEPARTMENT Descending.**



SELECT \*

FROM Worker

ORDER BY DEPARTMENT DESC;

1. **Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from Worker table.**

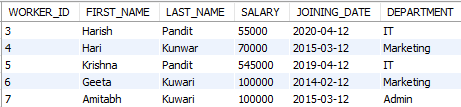
SELECT \*

FROM Worker

WHERE FIRST\_NAME = 'Vipul' OR FIRST\_NAME = 'Satish';



1. **Write an SQL query to print details of workers excluding first names, “Vipul” and “Satish” from Worker table.**



SELECT \*

FROM Worker

WHERE FIRST\_NAME = 'Vipul' OR FIRST\_NAME = 'Satish';

1. **Write** **an SQL query to print details of Workers with DEPARTMENT name as “Admin”.**

SELECT \*

FROM Worker

WHERE DEPARTMENT = 'Admin';

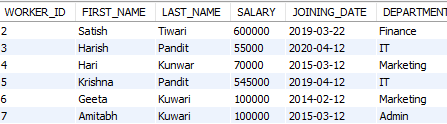


1. **Write an SQL query to print details of the Workers whose FIRST\_NAME contains ‘a’.**

SELECT \*

FROM Worker

WHERE FIRST\_NAME LIKE '%a%';



1. **Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘a’.**

SELECT \*

FROM Worker

WHERE FIRST\_NAME LIKE '%a%';

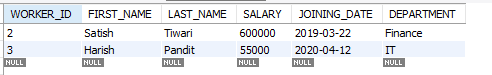


1. **Write** **an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘h’ and contains six alphabets.**

SELECT \*

FROM Worker

WHERE FIRST\_NAME LIKE '\_\_\_\_\_h' AND LENGTH(FIRST\_NAME) = 6;

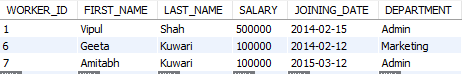


1. **Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.**

SELECT \*

FROM Worker

WHERE SALARY BETWEEN 100000 AND 500000;



1. **Write an SQL query to print details of the Workers who have joined in Feb’2014.**



SELECT \*

FROM Worker

WHERE JOINING\_DATE BETWEEN '2014-02-01' AND '2014-02-28';

1. **Write an SQL query to fetch the count of employees working in the department ‘Admin’.**

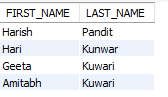


SELECT COUNT (\*) AS EmployeeCount

FROM Worker

WHERE DEPARTMENT = 'Admin';

1. **. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000**

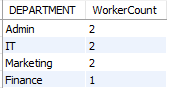


SELECT FIRST\_NAME, LAST\_NAME

FROM Worker

WHERE SALARY BETWEEN 50000 AND 100000;

1. **Write an SQL query to fetch the no. of workers for each department in the descending order.**



SELECT DEPARTMENT, COUNT (\*) AS WorkerCount

FROM Worker

GROUP BY DEPARTMENT

ORDER BY WorkerCount DESC;

1. **Write an SQL query to print details of the Workers who are also Managers.**



SELECT \*

FROM Worker

WHERE DEPARTMENT = 'Manager';

1. **Write an SQL query to fetch duplicate records having matching data in some fields of a table**



SELECT FIRST\_NAME, LAST\_NAME, COUNT(\*)

FROM Worker

GROUP BY FIRST\_NAME, LAST\_NAME

HAVING COUNT(\*) > 1;

1. **Write an SQL query to show only odd rows from a table.**

WITH NumberedRows AS (

SELECT \*, ROW\_NUMBER() OVER (ORDER BY WORKER\_ID) AS RowNum

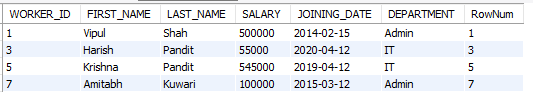
FROM Worker

)

SELECT \*

FROM NumberedRows

WHERE RowNum % 2 <> 0;



1. **Write an SQL query to show only even rows from a table.**

WITH NumberedRows AS (

SELECT \*, ROW\_NUMBER() OVER (ORDER BY WORKER\_ID) AS RowNum

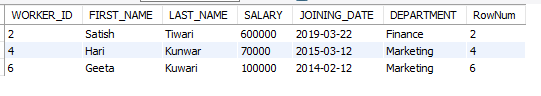
FROM Worker

)

SELECT \*

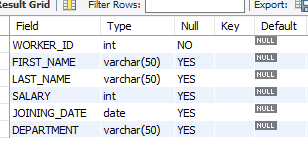
FROM NumberedRows

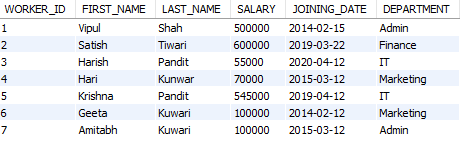
WHERE RowNum % 2 = 0;

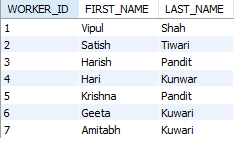


1. **Write an SQL query to clone a new table from another table.**

|  |
| --- |
| CREATE TABLE NewWorker AS  SELECT \*  FROM Worker; |





1. **Write an SQL query to fetch intersecting records of two tables.**

|  |
| --- |
| SELECT WORKER\_ID, FIRST\_NAME, LAST\_NAME  FROM Worker  INTERSECT  SELECT WORKER\_ID, FIRST\_NAME, LAST\_NAME  FROM Newworker; |

1. **Write an SQL query to show records from one table that another table does not have.**



SELECT \*

FROM Worker

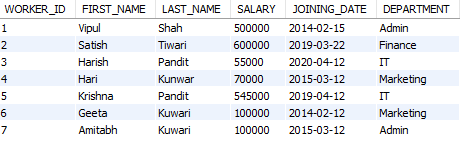
WHERE WORKER\_ID NOT IN (SELECT WORKER\_ID FROM newworker);

1. **Write an SQL query to show the current date and time.**



SELECT NOW() AS CurrentDateTime;

1. **Write an SQL query to show the top n (say 10) records of a table.**



SELECT \*

FROM Worker

ORDER BY WORKER\_ID

LIMIT 10;

1. **Write an SQL query to determine the nth (say n=5) highest salary from a table.**



WITH RankedSalaries AS (

SELECT

SALARY,

DENSE\_RANK() OVER (ORDER BY SALARY DESC) AS salary\_rank

FROM Worker

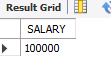
)

SELECT SALARY

FROM RankedSalaries

WHERE salary\_rank = 5;

1. **Write an SQL query to determine the 5th highest salary without using TOP or limit method.**



SELECT SALARY

FROM (

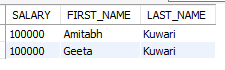
SELECT SALARY, ROW\_NUMBER() OVER (ORDER BY SALARY DESC) AS salary\_rank

FROM Worker

) AS RankedSalaries

WHERE salary\_rank = 5;

1. **Write an SQL query to fetch the list of employees with the same salary.**



SELECT SALARY, FIRST\_NAME, LAST\_NAME

FROM Worker

WHERE SALARY IN (

SELECT SALARY

FROM Worker

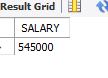
GROUP BY SALARY

HAVING COUNT(\*) > 1

)

ORDER BY SALARY, FIRST\_NAME, LAST\_NAME;

1. **Write an SQL query to show the second highest salary from a table.**



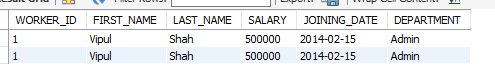
SELECT DISTINCT SALARY

FROM Worker

ORDER BY SALARY DESC

LIMIT 1 OFFSET 1;

1. **Write an SQL query to show one row twice in results from a table.**



SELECT \*

FROM worker

WHERE worker\_id = 1

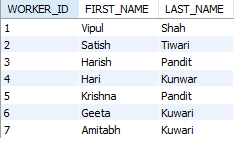
UNION ALL

SELECT \*

FROM worker

WHERE worker\_id = 1;

1. **Write an SQL query to fetch intersecting records of two tables.**



SELECT WORKER\_ID, FIRST\_NAME, LAST\_NAME

FROM Worker

INTERSECT

SELECT WORKER\_ID, FIRST\_NAME, LAST\_NAME

FROM Newworker;

1. **Write an SQL query to fetch the first 50% records from a table.**

WITH OrderedRecords AS (

SELECT \*,

ROW\_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum,

COUNT(\*) OVER () AS TotalCount

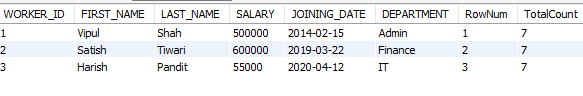
FROM worker

)

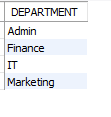
SELECT \*

FROM OrderedRecords

WHERE RowNum <= TotalCount / 2;



1. **Write an SQL query to fetch the departments that have less than five people in it.**



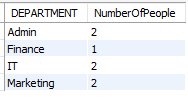
SELECT DEPARTMENT

FROM Worker

GROUP BY DEPARTMENT

HAVING COUNT(\*) < 5;

1. **Write an SQL query to show all departments along with the number of people in there.**



SELECT DEPARTMENT

FROM Worker

GROUP BY DEPARTMENT

HAVING COUNT(\*) < 5;

1. **Write an SQL query to show the last record from a table.**



SELECT \*

FROM Worker

ORDER BY WORKER\_ID DESC

LIMIT 1;

1. **Write an SQL query to fetch the first row of a table.**

SELECT \*

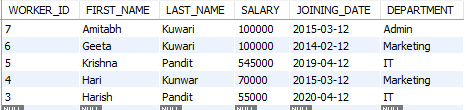
FROM Worker

ORDER BY WORKER\_ID

LIMIT 1;



1. **Write an SQL query to fetch the last five records from a table**



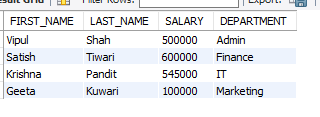
SELECT \*

FROM Worker

ORDER BY WORKER\_ID DESC

LIMIT 5;

1. **Write an SQL query to print the name of employees having the highest salary in each department.**



WITH RankedSalaries AS (

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT,

ROW\_NUMBER() OVER (PARTITION BY DEPARTMENT ORDER BY SALARY DESC) AS rn

FROM Worker

)

SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT

FROM RankedSalaries

WHERE rn = 1;

1. **Write an SQL query to fetch three max salaries from a table.**

SELECT salary

FROM worker

ORDER BY salary ASC

LIMIT 3;

1. **Write an SQL query to fetch three min salaries from a table.**

SELECT salary

FROM worker

ORDER BY salary DESC

LIMIT 3;



1. **Write an SQL query to fetch nth max salaries from a table.**



SELECT salary

FROM (

SELECT salary

FROM worker

ORDER BY salary DESC

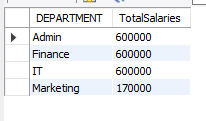
LIMIT 3

) AS subquery

ORDER BY salary

LIMIT 1;

1. **Write an SQL query to fetch departments along with the total salaries paid for each of them.**



SELECT DEPARTMENT, SUM(SALARY) AS TotalSalaries

FROM Worker

GROUP BY DEPARTMENT;

1. **Write an SQL query to fetch the names of workers who earn the highest salary.**



WITH MaxSalary AS (

SELECT MAX(SALARY) AS HighestSalary

FROM Worker

)

SELECT FIRST\_NAME, LAST\_NAME

FROM Worker

WHERE SALARY = (SELECT HighestSalary FROM MaxSalary);