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

-- # This query retrieves only different department names

-- # from the Worker table to avoid any repetitions

SELECT DISTINCT DEPARTMENT

FROM Worker;

2.Write an SQL query to print all Worker details

-- # sorts the results in ascending order by FIRST\_NAME

-- # and descending order by DEPARTMENT

SELECT \*

FROM Worker

ORDER BY FIRST\_NAME ASC, DEPARTMENT DESC;

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

-- # This query fetches all workers whose FIRST\_NAME contains the letter 'a'

SELECT \*

FROM Worker

WHERE FIRST\_NAME LIKE '%a%';

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

-- # Select all worker details

-- # Where FIRST\_NAME ends with 'h' AND has exactly 6 characters

SELECT \*

FROM Worker

WHERE FIRST\_NAME LIKE '\_h';

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

-- # Select all worker details

-- # Where SALARY is between 100000 and 500000 (inclusive)

SELECT \*

FROM Worker

WHERE SALARY BETWEEN 100000 AND 500000;

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

-- # Select all worker details

-- # Where JOINING\_DATE is in February 2014

SELECT \*

FROM Worker

WHERE MONTH(JOINING\_DATE) = 2 AND YEAR(JOINING\_DATE) = 2014;

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

-- # Count the number of workers

-- # Where the DEPARTMENT is 'Admin'

SELECT COUNT(\*) AS Admin\_Employee\_Count

FROM Worker

WHERE DEPARTMENT = 'Admin';

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

-- # Select FIRST\_NAME and LAST\_NAME of workers

-- # Where SALARY is between 50000 and 100000 inclusive

SELECT FIRST\_NAME, LAST\_NAME

FROM Worker

WHERE SALARY BETWEEN 50000 AND 100000;

9. SQL Query: Number of workers for each department in descending order

-- # Select DEPARTMENT and count of workers in each department

-- # Group results by DEPARTMENT

-- # Order the result by the worker count in descending order

SELECT DEPARTMENT, COUNT(\*) AS Num\_Workers

FROM Worker

GROUP BY DEPARTMENT

ORDER BY Num\_Workers DESC;

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

-- # Select all worker details

-- # Join Worker table with Title table using WORKER\_ID

-- # Filter only those where the TITLE is 'Manager'

SELECT W.\*

FROM Worker W

JOIN Title T ON W.WORKER\_ID = T.WORKER\_REF\_ID

WHERE T.WORKER\_TITLE = 'Manager';

WHEN WE INCLUDE ASSISTANT MANAGER AS WELL.

-- # Select all worker details

-- # Join Worker and Title tables using WORKER\_ID

-- # Filter for titles 'Manager' or 'Assistant Manager'

SELECT W.\*

FROM Worker W

JOIN Title T ON W.WORKER\_ID = T.WORKER\_REF\_ID

WHERE T.WORKER\_TITLE IN ('Manager', 'Assistant Manager');

11. Write an SQL query to determine the 2nd lowest salary without using TOP or limit method.

-- # Select the minimum salary

-- # That is greater than the minimum salary in the table

-- # This gives the second lowest salary without using TOP or LIMIT

SELECT MIN(SALARY) AS Second\_Lowest\_Salary

FROM Worker

WHERE SALARY > (

SELECT MIN(SALARY)

  FROM Worker

);

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

-- # Step 1: Find salaries that appear more than once

-- # Step 2: Fetch all workers whose salary is in that list

SELECT \*

FROM Worker

WHERE SALARY IN (

SELECT SALARY

FROM Worker

GROUP BY SALARY

HAVING COUNT(\*) > 1

);

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

-- # Step 1: Select the maximum salary

-- # That is less than the highest salary in the Worker table

-- # This ensures we get the second highest distinct salary value

SELECT MAX(SALARY) AS Second\_Highest\_Salary

FROM Worker

WHERE SALARY < (

-- # Get the highest salary in the table

SELECT MAX(SALARY)

FROM Worker

);

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

-- # Select a specific row (e.g., worker with ID = 1)

-- # Use UNION ALL to include it again (duplicating it)

SELECT \*

FROM Worker

WHERE WORKER\_ID = 1

UNION ALL

SELECT \*

FROM Worker

WHERE WORKER\_ID = 1;

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

-- # Step 1: Use a CTE or subquery to assign row numbers to each record

-- # Step 2: Count total rows and calculate 50% of them

-- # Step 3: Select only rows where row number <= 50% of total

WITH NumberedRows AS (

SELECT \*,

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

FROM Worker

),

TotalCount AS (

SELECT COUNT(\*) AS TotalRows FROM Worker

)

SELECT \*

FROM NumberedRows, TotalCount

WHERE RowNum <= TotalRows / 2;

16. Write an SQL query to fetch the departments that have less than three people in it.

-- # Step 1: Group workers by department

-- # Step 2: Count number of workers in each department

-- # Step 3: Filter those groups where the count is less than 3

SELECT DEPARTMENT, COUNT(\*) AS Num\_Workers

FROM Worker

GROUP BY DEPARTMENT

HAVING COUNT(\*) < 3;

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

-- # Step 1: Group the Worker table by department

-- # Step 2: Count the number of workers in each department using COUNT(\*)

-- # Step 3: Select the department name and the corresponding worker count

SELECT DEPARTMENT, COUNT(\*) AS Num\_Workers

FROM Worker

GROUP BY DEPARTMENT;

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

-- # Step 1: Order all records in descending order to get the latest entries

-- # Step 2: Limit the result to only the last 5 records

-- # Step 3: Wrap the result and order it ascendingly for natural display

SELECT \*

FROM (

SELECT \*

FROM Worker

ORDER BY WORKER\_ID DESC -- You can replace WORKER\_ID with JOINING\_DATE or CREATED\_AT

LIMIT 5

) AS LastFiveRecords

ORDER BY WORKER\_ID ASC; -- Display them in ascending order

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

-- # Step 1: For each worker, compare their salary to the maximum salary in their department

-- # Step 2: Use a correlated subquery to get the max salary for the current worker's department

-- # Step 3: Select those workers whose salary equals that max

SELECT FIRST\_NAME, DEPARTMENT, SALARY

FROM Worker w

WHERE SALARY = (

SELECT MAX(SALARY)

FROM Worker

WHERE DEPARTMENT = w.DEPARTMENT

);

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

-- # This query selects distinct salary values from the Worker table

-- # It filters out those salaries which are less than 3 distinct salaries greater than them

-- # In effect, this gives us the top 3 distinct salaries in descending order

SELECT DISTINCT SALARY

FROM Worker w1

WHERE 2 >= (

-- # This subquery counts how many distinct salaries are greater than the current one

SELECT COUNT(DISTINCT SALARY)

FROM Worker w2

WHERE w2.SALARY > w1.SALARY

)

ORDER BY SALARY DESC;

21. Write an SQL query to print the name of employees having the lowest salary in accunt and admin department

-- # This query selects FIRST\_NAME, DEPARTMENT, and SALARY of employees

-- # whose salary is the minimum salary in either the Account or Admin department

SELECT FIRST\_NAME, DEPARTMENT, SALARY

FROM Worker w

WHERE DEPARTMENT IN ('Account', 'Admin') -- # Step 1: Filter only Account and Admin departments

AND SALARY = (

-- # Step 2: Get the minimum salary in that specific department

SELECT MIN(SALARY)

FROM Worker

WHERE DEPARTMENT = w.DEPARTMENT

);