**Que – 1. Problem Statement:**

Write a query to find all the details of those employees who earn the third-highest salary.

* Return **all** the columns from the employee's table.
* Use the **employees** table.

**Ans – 1.**

SELECT \*

FROM employees

WHERE salary =

(SELECT DISTINCT salary

FROM employees

ORDER BY salary DESC limit 1 offset 2)

ORDER BY employee\_id

**Que – 2. Problem Statement:**

Display the ‘full\_name’ (first and last name separated by space) of a manager who manages 4 or more employees.

* Return the column '**full\_name**'.
* Use the **employees** table.
* The column **manager\_id** in the employees table represents the employee\_id of the manager.

**Ans – 2.**

SELECT CONCAT(first\_name," ",last\_name) AS full\_name

FROM employees

WHERE employee\_id IN

(SELECT manager\_id

FROM employees

GROUP BY manager\_id

HAVING count(1)>4)

**Que – 3. Problem Statement:**

Display the details of those employees who have a **manager** working in the department that is **US** based. Also, display the output in ascending order of their ‘employee\_id’.

* Return the columns '**employee\_id**, '**first\_name**', '**last\_name**'.
* **manager\_id** column represents the **employee\_id** of the manager.

**Ans – 3.**

SELECT employee\_id,

first\_name,

last\_name

FROM employees

WHERE manager\_id IN

(SELECT e.manager\_id

FROM employees AS e

JOIN departments AS d

ON e.department\_id=d.department\_id

JOIN locations AS l

ON d.location\_id=l.location\_id

JOIN countries AS c

ON l.country\_id=c.country\_id

WHERE c.country\_id="US")

**Que – 4. Problem Statement:**

Display the details of all those departments that **don’t** have any working employees.

* Return the columns '**department\_id**', '**department\_name**

**Ans – 4.**

SELECT department\_id,

department\_name

FROM departments

WHERE department\_id NOT IN

(SELECT DISTINCT department\_id

FROM employees)

**Que – 5. Problem Statement:**

Display the details of the employees who belong to the **‘Europe’** region and order the output in descending order of salary and ascending order of employee\_id.

* Return the columns '**employee\_id**', '**full\_name**' (first\_name and last\_name separated by space), '**salary**', '**phone\_number**', '**department\_id**', '**department\_name**', '**street\_address**', '**city**', '**country\_name**', '**region\_id**', '**region\_name**'.

**Ans – 5.**

SELECT e.employee\_id,

CONCAT(e.first\_name," ",e.last\_name)AS full\_name

e.salary,

e.phone\_number,

e.department\_id,

d.department\_name,

l.street\_address,

l.city,

c.country\_name,

r.region\_id,

r.region\_name

FROM employees AS e

LEFT JOIN departments AS d

ON e.department\_id=d.department\_id

LEFT JOIN locations AS l

ON d.location\_id=l.location\_id

LEFT JOIN countries AS c

ON l.country\_id=c.country\_id

LEFT JOIN regions AS r

ON c.region\_id=r.region\_id

WHERE r.region\_name="Europe"

ORDER BY e.salary desc,e.employee\_id

**Que – 6. Problem Statement:**

Find the details of all those employees who work in the ‘**Human** **Resources**’ department.

* Return the columns '**employee\_id**', '**department\_id**', '**first\_name**', '**last\_name**', '**job\_id**', '**department\_name**'.
* Use the tables employees and departments.

**Ans – 6.**

SELECT e.employee\_id,

e.department\_id,

e.first\_name,

e.last\_name,

e.job\_id,

d.department\_name

FROM employees AS e

LEFT JOIN departments AS d

ON e.department\_id=d.department\_id

WHERE d.department\_name="Human Resources"

**Que – 7. Problem Statement:**

Display the details of the employees who joined the company before their managers joined the company.

* Return the columns '**employee\_id**', '**first\_name**', and '**last\_name**'.
* Return the result ordered by **employee\_id** in ascending order.

**Ans – 7.**

SELECT e.employee\_id,

e.first\_name,

e.last\_name

FROM employees AS e

LEFT JOIN

(SELECT employee\_id AS manager\_id,

hire\_date

FROM employees

WHERE employee\_id IN

(SELECT manager\_id

FROM employees

ORDER BY employee\_id)) AS manager

ON e.manager\_id=manager.manager\_id

WHERE e.hire\_date < manager.hire\_date

ORDER BY e.employee\_id

**Que – 8. Problem Statement:**

Find the details of the employees who earn less than the average salary in their respective departments.

* Return the columns '**employee\_id**', '**first\_name**', '**last\_name**', '**department\_id**', '**salary**'.
* Use the **employees** table.

**Ans – 8.**

SELECT e.employee\_id,

e.first\_name,

e.last\_name,

e.department\_id,

e.salary

FROM employees AS e

GROUP BY department\_id

HAVING salary<avg(salary)

**Que – 9. Problem Statement:**

Display the details i.e, employee id, first name, last name, salary, department name, and the city of those employees who earn the **same** **salary** as the employee that has the **minimum** **salary** when joined **between** the **dates** 1998-01-01 and 2003-12-31.

* Return the columns '**employee\_id**', '**first\_name**', '**last\_name**', '**salary**', '**department\_name**', and '**city**'.
* Use the tables employees, departments, and locations.

**Ans – 9.**

\*\*Not verified as Job\_history not available in database

SELECT e.employee\_id,

e.first\_name,

e.last\_name,

e.salary,

d.department\_name,

l.city

FROM employees AS e

LEFT JOIN job\_history AS jh

ON e.job\_id=jh.job\_id

LEFT JOIN departments AS d

ON jh.department\_id=d.department\_id

LEFT JOIN locations AS l

ON d.location\_id=l.location\_id

WHERE e.salary=min(e.salary)

AND jh.start\_date

BETWEEN "1998-01-01"

AND "2003-12-31"

**Que – 10. Problem Statement:**

Display the employee’s full name ( first name and last name separated by space) as ‘full\_name’ of all those employees whose salary is greater than 40% of their department’s total salary.

* Return the column '**full\_name**'.
* Use the **employees** table.

**Ans – 10.**

SELECT CONCAT(first\_name," ",last\_name) as full\_name

FROM employees

GROUP BY department\_id

HAVING salary > (40/100) \* sum(salary)

**Que – 11. Problem Statement:**

Based on the employee’s salary, divide the employees into three different classes.

1. Salary **greater than** 20,000 (i.e, excluding 20,000) as '**Class** **A**'
2. Salary **between** 10,000 to 20,000 (i.e, including both 10,000 and 20,000) as '**Class B**'
3. Salary **less than** 10,000 (i.e, excluding 10,000) as '**Class C**'. Return the new column as 'Salary\_bin'.

* Return the columns '**employee\_id**', '**salary**', '**Salary\_bin**'.
* Use the table employees.

**Ans – 11.**

SELECT employee\_id,

salary,

CASE

WHEN salary>20000 THEN

"Class A"

WHEN salary>=10000 THEN

"Class B"

WHEN salary>0 THEN

"Class C"

ELSE "Not Possible"

END AS Salary\_bin

FROM employees

**Que – 12. Problem Statement:**

Display the details of the employees who **had** job titles like ‘sales’ in the **past** and the **min\_salary** is greater than or equal to **6000**.

* Return the columns '**employee\_id**', '**department\_name**', '**job\_id**', '**job\_title**', and '**min\_salary**'.
* Return the employee's current information for the columns 'employee\_id*',*and*'*department\_name'.
* Return the employee's past information for the columns 'job\_id*', '*job\_title', and 'min\_salary'.

**NOTE:**

1. To get the **min\_salary** refer to the jobs table.
2. Refer to the **job\_history** table to get the details of past jobs.
3. An employee might have worked in multiple jobs in the past whose record will be available in job\_history.
4. If any employee hasn't worked in any jobs in the past, his record wouldn't be present in the job\_history table.

**Ans – 12.**

SELECT e.employee\_id,

d.department\_name,

jh.job\_id,

j.job\_title,

j.min\_salary

FROM employee AS e

LEFT JOIN job AS j

ON j.job\_id=e.job\_id

LEFT JOIN job\_history AS jh

ON j.job\_id=jh.job\_id

LEFT JOIN departments AS d

ON jh.department\_id=d.department\_id

WHERE j.job\_title LIKE "%sales%"

AND j.min\_salary>=6000