### **### 1. Calculate the running total of salaries for each employee.**

```sql

SELECT emp\_no, salary, SUM(salary) OVER (PARTITION BY emp\_no ORDER BY from\_date) as running\_total

from salaries;

### **### 2. List employees' salaries with an additional column indicating whether they are above or below the average salary.**

```sql

SELECT emp\_no, salary ,

CASE

WHEN salary > ( SELECT AVG(salary) from salaries) THEN "Above Average"

ELSE "Below Average"

END as salary\_level

From salaries

WHERE to\_date = '9999-01-01';

### **### 3. Retrieve the highest salary for each department, along with the corresponding employee details.**

```sql

WITH DeptHighestSalaries AS (

SELECT de.dept\_no, MAX(s.salary) AS highest\_salary

FROM dept\_emp de

JOIN salaries s ON de.emp\_no = s.emp\_no

WHERE s.to\_date = '9999-01-01'

GROUP BY de.dept\_no

)

SELECT d.dept\_name, e.emp\_no, e.first\_name, e.last\_name, dhs.highest\_salary

FROM DeptHighestSalaries dhs

JOIN dept\_emp de ON dhs.dept\_no = de.dept\_no

JOIN salaries s ON de.emp\_no = s.emp\_no AND s.salary = dhs.highest\_salary

JOIN employees e ON de.emp\_no = e.emp\_no

JOIN departments d ON dhs.dept\_no = d.dept\_no

WHERE s.to\_date = '9999-01-01';

```

### Method-2

```sql

with aa as(

SELECT d.dept\_name, s.emp\_no, e.first\_name , e.last\_name, s.salary , de.dept\_no from salaries s

JOIN dept\_emp de on s.emp\_no = de.emp\_no

JOIN departments d on de.dept\_no = d.dept\_no

JOIN employees e on de.emp\_no = e.emp\_no

WHERE s.to\_date = '9999-01-01'

),

rank\_salary as(

SELECT aa.dept\_name, aa.emp\_no, aa.salary, aa.dept\_no, aa.first\_name, aa.last\_name

, ROW\_NUMBER() OVER (PARTITION BY aa.dept\_no ORDER BY aa.salary DESC) as rank\_salary from aa

)

SELECT rs.dept\_name, rs.emp\_no, rs.first\_name, rs.last\_name, rs.salary from rank\_salary rs

WHERE rank\_salary = 1;

```

### **### 4. Find the difference in days between each employee's hire date and the current date.**

```sql

SELECT e.emp\_no, e.first\_name, e.last\_name, e.hire\_date

, DATEDIFF(CURRENT\_DATE(), e.hire\_date) as days\_with\_company from employees e;

```

### **### 5. List employees who have had more than one job title at the same time using a self-join.**

```sql

SELECT t1.emp\_no, t1.title AS title1, t2.title AS title2, t1.from\_date, t1.to\_date

FROM titles t1

JOIN titles t2 ON t1.emp\_no = t2.emp\_no AND t1.title != t2.title

WHERE t1.from\_date <= t2.to\_date AND t2.from\_date <= t1.to\_date;

```

### **### 6. Retrieve employees who have worked in more than one department using a HAVING clause.**

```sql

SELECT e.emp\_no, e.first\_name, e.last\_name, COUNT(DISTINCT de.dept\_no) as total\_dept from employees e

JOIN dept\_emp de on e.emp\_no = de.emp\_no

GROUP BY e.emp\_no, e.first\_name, e.last\_name

HAVING total\_dept > 1;

```

### **### 7. Calculate the percentage of total salaries for each department.**

```sql

SELECT d.dept\_name, sum(s.salary) as dept\_salary,

sum(s.salary) \* 100 / (SELECT sum(ss.salary) from salaries ss WHERE ss.to\_date = '9999-01-01') as salary

from departments d

JOIN dept\_emp de on d.dept\_no = de.dept\_no

JOIN salaries s on de.emp\_no = s.emp\_no

WHERE s.to\_date = '9999-01-01' and de.to\_date = '9999-01-01'

GROUP BY d.dept\_name;

```

### **### 8. Find the second highest salary in the company.**

```sql

SELECT DISTINCT salary from salaries

ORDER BY salary DESC

LIMIT 1 OFFSET 1;

-- Method-2 ( Complex Query)

with ss as (

SELECT salary, ROW\_NUMBER() OVER(ORDER BY sub.salary DESC ) as Salary\_rank from (SELECT DISTINCT(salary) as salary from salaries) as sub

)

SELECT \* from ss

WHERE Salary\_rank = '2';

```

### **### 9. Retrieve the employee(s) with the longest duration in the same department.**

```sql

SELECT e.emp\_no, e.first\_name,de.from\_date, e.last\_name, d.dept\_name,

DATEDIFF(de.to\_date, de.from\_date) AS duration\_days

FROM employees e

JOIN dept\_emp de ON e.emp\_no = de.emp\_no

JOIN departments d ON de.dept\_no = d.dept\_no

WHERE de.to\_date = '9999-01-01'

ORDER BY duration\_days DESC

LIMIT 1;

```

### **### 10. Use a window function to rank employees by their salary within each department.**

```sql

SELECT e.emp\_no, e.first\_name, e.last\_name, d.dept\_name , s.salary ,

RANK() OVER (PARTITION BY de.dept\_no ORDER BY s.salary) as salary\_rank

from employees e

JOIN dept\_emp as de on e.emp\_no = de.emp\_no

JOIN departments d on de.dept\_no = d.dept\_no

JOIN salaries s ON e.emp\_no = s.emp\_no

WHERE s.to\_date = '9999-01-01'

```

### **### 11. Find the average salary for each department, but only include departments with more than 10 employees.**

```sql

SELECT d.dept\_no, d.dept\_name, AVG(s.salary) FROM departments d

JOIN dept\_emp de on d.dept\_no = de.dept\_no

JOIN salaries s on de.emp\_no = s.emp\_no

WHERE s.to\_date = '9999-01-01'

GROUP BY d.dept\_no, d.dept\_name

HAVING COUNT(de.emp\_no) > 10;

```

### **### 12. Identify employees whose salaries have increased over time.**

```sql

SELECT s.emp\_no, MIN(s.salary) as starting\_salary, MAX(s.salary) as latest\_salary FROM salaries s

GROUP BY s.emp\_no

HAVING starting\_salary < latest\_salary;

-- Method-2

SELECT s.emp\_no, MIN(s.salary) as starting\_salary, MAX(s.salary) as Highest\_salary,

(MAX(s.salary)-MIN(s.salary)) as Salary\_Increase from salaries s

GROUP BY s.emp\_no

HAVING salary\_increase <> 0

ORDER BY salary\_increase DESC;

```

### **### 13. Calculate the year-over-year salary growth for each employee.**

```sql

SELECT s.emp\_no,YEAR(s.from\_date) as salary\_year, s.salary

, LAG(s.salary) OVER(PARTITION BY emp\_no ORDER BY s.from\_date) as previous\_salary

, s.salary - (LAG(s.salary) OVER(PARTITION BY emp\_no ORDER BY s.from\_date)) as salary\_growth

from salaries s;

```

### **### 14. Retrieve the names of employees who are managers and have worked as non-managers as well.**

```sql

SELECT DISTINCT e.emp\_no, e.first\_name, e.last\_name

FROM employees e

JOIN dept\_manager dm ON e.emp\_no = dm.emp\_no

WHERE EXISTS (

SELECT 1

FROM dept\_emp de

WHERE de.emp\_no = e.emp\_no AND de.dept\_no NOT IN (SELECT dept\_no FROM dept\_manager WHERE emp\_no = e.emp\_no)

);

```

### **### 15. Find the department with the most employees who have not received a raise in the last 3 years.**

```sql

SELECT d.dept\_name, COUNT(e.emp\_no) AS no\_raise\_count

FROM employees e

JOIN dept\_emp de ON e.emp\_no = de.emp\_no

JOIN departments d ON de.dept\_no = d.dept\_no

JOIN salaries s ON e.emp\_no = s.emp\_no

WHERE s.to\_date = '9999-01-01'

AND NOT EXISTS (

SELECT 1

FROM salaries s2

WHERE s2.emp\_no = e.emp\_no AND s2.from\_date > s.from\_date AND s2.salary > s.salary AND s2.from\_date > DATE\_SUB(CURDATE(), INTERVAL 3 YEAR)

)

GROUP BY d.dept\_name

ORDER BY no\_raise\_count DESC

LIMIT 1;

```

### **### 16. Use a CASE statement to categorize employees by their length of service.**

```sql

SELECT e.emp\_no,e.first\_name,e.last\_name, e.hire\_date,

CASE

WHEN (DATEDIFF(CURRENT\_DATE, e.hire\_date)) < (25\*365) THEN '25 Years'

WHEN (DATEDIFF(CURRENT\_DATE, e.hire\_date)) < (35\*365) THEN '25 to 35 Years'

ELSE 'More than 35 years'

END as service\_length

from employees e;

```

### **### 17. Find the average duration an employee stays in a department.**

```sql

SELECT de.dept\_no, AVG(DATEDIFF(de.to\_date, de.from\_date)) AS avg\_duration\_days

FROM dept\_emp de GROUP BY de.dept\_no;

```

### **### 18. Retrieve employees who have been promoted in the last 30 years using a self-join.**

```sql

SELECT t1.emp\_no , t1.title as previous\_title, t2.title as current\_title from titles t1

JOIN titles t2 on t1.emp\_no = t2.emp\_no

WHERE t1.from\_date < t2.from\_date and t2.from\_date > SUBDATE(CURRENT\_DATE(), INTERVAL 30 YEAR);

```

### **### 19. List the departments where the average salary is greater than the overall average salary.**

```sql

SELECT d.dept\_name, AVG(s.salary) as avg\_salary from departments d

JOIN dept\_emp de on d.dept\_no = de.dept\_no

JOIN salaries s on de.emp\_no=s.emp\_no

WHERE s.to\_date = '9999-01-01'

GROUP BY d.dept\_name

HAVING AVG(s.salary) > (SELECT AVG(salary) from salaries WHERE to\_date = '9999-01-01');

```

### **### 20. Find the most common job title for employees who have been with the company for 15 years.**

```sql

SELECT t.title,COUNT(t.title) as total\_titles from titles t

WHERE DATEDIFF(t.to\_date,t.from\_date) > (15\*365)

GROUP BY t.title

ORDER BY COUNT(t.title) DESC

LIMIT 1;