1. Write a query to calculate the average salary for each department and rank the departments based on their average salary.

Query: WITH AvgSalaryPerDept AS (

SELECT

department,

AVG(salary) AS avg\_salary,

RANK() OVER (ORDER BY AVG(salary) DESC) AS dept\_rank

FROM

employees

GROUP BY

department

)

SELECT

department,

avg\_salary,

dept\_rank

FROM

AvgSalaryPerDept;

1. Write a query to calculate the tenure of each employee in years and rank employees by their tenure within each department.

Query:

WITH EmployeeTenure AS (

SELECT

employee\_id,

department,

hire\_date,

DATEDIFF(MONTH, hire\_date, GETDATE()) / 12.0 AS tenure\_years,

RANK() OVER (PARTITION BY department ORDER BY hire\_date ASC) AS tenure\_rank

FROM

employees

)

SELECT

employee\_id,

department,

hire\_date,

tenure\_years,

tenure\_rank

FROM

EmployeeTenure;

1. Write a query to calculate the difference in salary between each employee and the next hired employee.

Query: WITH EmployeeSalaryDiff AS (

SELECT

employee\_id,

hire\_date,

salary,

LEAD(employee\_id) OVER (ORDER BY hire\_date ASC) AS next\_employee\_id,

LEAD(salary) OVER (ORDER BY hire\_date ASC) AS next\_employee\_salary,

LEAD(hire\_date) OVER (ORDER BY hire\_date ASC) AS next\_employee\_hire\_date

FROM

employees

)

SELECT

employee\_id,

hire\_date,

salary,

next\_employee\_id AS next\_hired\_employee\_id,

next\_employee\_hire\_date AS next\_hired\_employee\_hire\_date,

next\_employee\_salary AS next\_hired\_employee\_salary,

next\_employee\_salary - salary AS salary\_difference

FROM

EmployeeSalaryDiff;

1. Write a query to find the largest salary increase between consecutive hires.

Query:

WITH EmployeeSalaryDiff AS (

SELECT

employee\_id,

hire\_date,

salary,

LEAD(employee\_id) OVER (ORDER BY hire\_date ASC) AS next\_employee\_id,

LEAD(salary) OVER (ORDER BY hire\_date ASC) AS next\_employee\_salary,

LEAD(hire\_date) OVER (ORDER BY hire\_date ASC) AS next\_employee\_hire\_date

FROM

employees

)

SELECT

employee\_id,

hire\_date,

salary,

next\_employee\_id AS next\_hired\_employee\_id,

next\_employee\_hire\_date AS next\_hired\_employee\_hire\_date,

next\_employee\_salary AS next\_hired\_employee\_salary,

(next\_employee\_salary - salary) AS salary\_increase

FROM

EmployeeSalaryDiff

WHERE

next\_employee\_salary IS NOT NULL -- Ensure there is a next employee to compare with

ORDER BY

salary\_increase DESC

LIMIT 1;

1. Write a query to track the salary changes over time for each employee, showing the current and previous salary with the change in percentage.

Query:

WITH SalaryChanges AS (

SELECT

employee\_id,

hire\_date,

salary,

LAG(salary) OVER (PARTITION BY employee\_id ORDER BY hire\_date) AS previous\_salary,

LAG(salary) OVER (PARTITION BY employee\_id ORDER BY hire\_date) / NULLIF(salary, 0) AS salary\_change\_percentage

FROM

employees

)

SELECT

employee\_id,

hire\_date,

salary,

COALESCE(previous\_salary, 0) AS previous\_salary,

CASE

WHEN previous\_salary IS NULL THEN NULL

ELSE ROUND((salary\_change\_percentage - 1) \* 100, 2)

END AS change\_percentage

FROM

SalaryChanges

ORDER BY

employee\_id, hire\_date;