```
/* NextGen SQL Capstone Project: Workforce Strategic Analysis
   Author: Loveth Ilen
   Institution: 10Alytics
   Date: January 22, 2026
*/


-- Quick data check
select * from attendance;
select * from department;
select * from employee;
select * from performance;
select * from salary;
select * from turnover;



-- Employee Retention Analysis

-- Question 1. Who are the top 5 highest serving employees?
SELECT
   (e.first_name || ' ' || e.last_name) AS full_name,
   CASE
      WHEN t.turnover_date IS NOT NULL THEN (t.turnover_date::date - 
e.hire_date::date)
      ELSE (CURRENT_DATE - e.hire_date::date)
   END AS days_served
FROM employee e
LEFT JOIN turnover t ON e.employee_id = t.employee_id
```

```sql
ORDER BY 2 DESC

LIMIT 5;


-- Question 2. What is the turnover rate for each department?

SELECT d.department_name,

       ROUND((COUNT(t.employee_id) * 100.0 / NULLIF(COUNT(e.employee_id), 0)), 2)
AS turnover_rate

FROM department d

LEFT JOIN employee e ON d.department_id = e.department_id

LEFT JOIN turnover t ON e.employee_id = t.employee_id

GROUP BY d.department_name

ORDER BY turnover_rate DESC;


-- Question 3. Which employees are at risk of leaving based on their performance?


SELECT

    (e.first_name || ' ' || e.last_name) AS full_name,

    d.department_name,

    p.performance_score

FROM employee e

JOIN performance p ON e.employee_id = p.employee_id

JOIN department d ON e.department_id = d.department_id

LEFT JOIN turnover t ON e.employee_id = t.employee_id

WHERE t.turnover_date IS NULL
```

```
AND p.performance_date = (SELECT MAX(performance_date) FROM performance
WHERE employee_id = e.employee_id)

AND p.performance_score < 3.5;


-- Question 4. What are the main reasons employees are leaving the company?

SELECT reason_for_leaving, COUNT(*) AS total_left

FROM turnover

GROUP BY reason_for_leaving

ORDER BY total_left DESC;




-- PERFORMANCE ANALYSIS




-- Question 1. How many employees have left the company?

SELECT COUNT(*) AS total_departures FROM turnover;


-- Question 2. How many employees have a performance score of 5.0 / below 3.5?

WITH LatestPerf AS (

   SELECT performance_score, ROW_NUMBER() OVER(PARTITION BY employee_id
ORDER BY performance_date DESC) as rank

   FROM performance

)

SELECT

COUNT(CASE WHEN performance_score = 5.0 THEN 1 END) AS score_5_count,
```

```sql
    COUNT(CASE WHEN performance_score < 3.5 THEN 1 END) AS below_3_5_count
FROM LatestPerf WHERE rank = 1;

-- Question 3. Which department has the most employees with a performance of 5.0 / below 3.5?

WITH LatestPerf AS (
    SELECT employee_id, performance_score, ROW_NUMBER() OVER(PARTITION BY employee_id ORDER BY performance_date DESC) as rank
    FROM performance
)
SELECT d.department_name,
    COUNT(CASE WHEN lp.performance_score = 5.0 THEN 1 END) AS score_5,
    COUNT(CASE WHEN lp.performance_score < 3.5 THEN 1 END) AS score_low
FROM department d
JOIN employee e ON d.department_id = e.department_id
JOIN LatestPerf lp ON e.employee_id = lp.employee_id
WHERE lp.rank = 1
GROUP BY d.department_name
ORDER BY score_low DESC;

-- Question 4. What is the average performance score by department?

SELECT d.department_name, ROUND(AVG(p.performance_score), 2) AS avg_perf
FROM department d
```

```sql
JOIN employee e ON d.department_id = e.department_id

JOIN performance p ON e.employee_id = p.employee_id

WHERE p.performance_date = (SELECT MAX(performance_date) FROM performance
WHERE employee_id = e.employee_id)

GROUP BY d.department_name;



-- SALARY ANALYSIS



-- Question 1. What is the total salary expense for the company?


SELECT SUM(salary_amount) AS total_active_salary

FROM salary s

WHERE s.salary_date = (SELECT MAX(salary_date) FROM salary WHERE employee_id
= s.employee_id)

AND s.employee_id NOT IN (SELECT employee_id FROM turnover);


-- Question 2. What is the average salary by job title?

SELECT e.job_title, ROUND(AVG(s.salary_amount), 2) AS avg_salary

FROM employee e

JOIN salary s ON e.employee_id = s.employee_id

WHERE s.salary_date = (SELECT MAX(salary_date) FROM salary WHERE employee_id
= s.employee_id)

GROUP BY e.job_title
```

```
ORDER BY avg_salary DESC;


-- Question 3. How many employees earn above 80,000?


SELECT COUNT(DISTINCT employee_id) AS high_earners
FROM salary
WHERE salary_amount > 80000
AND salary_date = (SELECT MAX(salary_date) FROM salary WHERE employee_id =
salary.employee_id);


-- Question 4. How does performance correlate with salary across departments?
SELECT
    d.department_name,
    ROUND(AVG(s.salary_amount), 2) AS avg_salary,
    ROUND(AVG(p.performance_score), 2) AS avg_perf
FROM department d
JOIN employee e ON d.department_id = e.department_id
JOIN salary s ON e.employee_id = s.employee_id
JOIN performance p ON e.employee_id = p.employee_id
WHERE s.salary_date = (SELECT MAX(salary_date) FROM salary WHERE employee_id
= e.employee_id)
AND p.performance_date = (SELECT MAX(performance_date) FROM performance
WHERE employee_id = e.employee_id)
GROUP BY d.department_name;
```

```
-- FINAL ADDED VALUE ANALYSE


-- 1. TENURE RISK: Identifying high-performing LONG TERM EMPLOYEES

-- This helps us see which long-term employees we need to keep.

SELECT

    e.first_name || ' ' || e.last_name AS employee,

    (CURRENT_DATE - e.hire_date) / 365 AS years_served,

    p.performance_score

FROM employee e

JOIN performance p ON e.employee_id = p.employee_id

WHERE (CURRENT_DATE - e.hire_date) / 365 > 8

ORDER BY 3 DESC;


-- 2. FINANCIAL RISK: Identifying High-Salary Underperformers

-- Finding where the $3M budget isn't being used well (e.g., Bob Lee).

SELECT DISTINCT ON (e.employee_id)

    e.first_name || ' ' || e.last_name AS employee,

    d.department_name,

    s.salary_amount,

    p.performance_score

FROM employee e

JOIN salary s ON e.employee_id = s.employee_id

JOIN performance p ON e.employee_id = p.employee_id
JOIN department d ON e.department_id = d.department_id
WHERE s.salary_amount > 80000
  AND p.performance_score < 3.5
ORDER BY e.employee_id, s.salary_date DESC, p.performance_date DESC;
```