Advanced SQL Problem Set with Dataset

Dataset Schema

Assume the following tables:

1. employees

----

emp_id	INT (PK)
first_name	VARCHAR(50)
last_name	VARCHAR(50)
department	VARCHAR(50)
salary	INT
join_date	DATE
manager_id	INT (FK to employees.emp_id)
project_id	INT (FK to projects.project_id)

2. departments

----

department_id	INT (PK)
department_name	VARCHAR(50)

3. projects

----

project_id	INT (PK)
project_name	VARCHAR(50)
start_date	DATE
end_date	DATE

Advanced SQL Problem Set

String Functions:
1. Find employees whose last name starts with 'S'.
2. Display first_name and last_name concatenated as full_name in uppercase.
3. Show employees with a 5-character first name.

Date Functions:
4. List employees who joined in the last 2 years.
5. Show number of days since each employee joined.
6. Find the month name and year from each employee's join_date.

Math Functions:
7. Round off each employee's salary to the nearest thousand.
8. Find employees whose salary is above the average salary.
9. Show absolute difference from company average salary.

Aggregate Functions with HAVING:
10. Find departments with more than 3 employees.
11. Show total and average salary per department with avg salary > 60000.

Advanced SQL Problem Set with Dataset

Subqueries:
12. Find the employee(s) with the maximum salary.
13. List employees earning more than avg salary in their department.
14. Show employees who joined before the earliest join date in IT.

JOINS:
15. Show each employee's name and manager's name.
16. List employees with their department name.
17. List employees not assigned any project.

Window Functions:
18. Assign a row number to employees in each department based on salary.
19. Show running total salary within each department.
20. Show difference in salary between employee and previous by join date.

CTE (Common Table Expressions):
21. Use CTE to calculate total salary per department, filter total > 200000.
22. Create a recursive CTE to generate numbers 1 to 10.
23. Use a CTE to find employees with duplicate first names.

Case Statements:
24. Label employees as 'Junior', 'Mid', or 'Senior' based on salary.
25. Count employees in salary categories using CASE.

NULL Functions:
26. Replace NULL department values with 'Unknown'.
27. Show employees with no department.
28. Use COALESCE to provide default for missing projects.