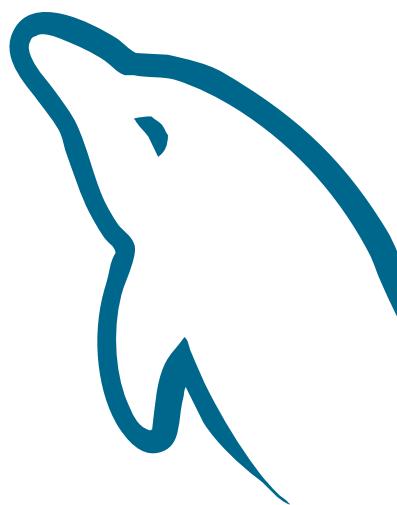
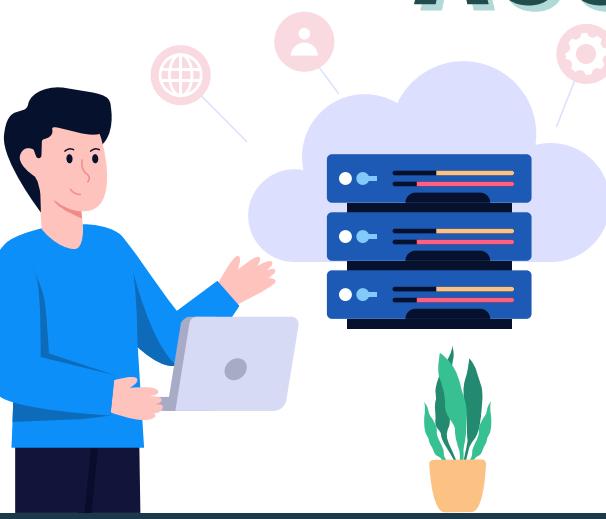




ASSIGNMENT





Multiple Joins + Subqueries + Group By + Ranking Functions

1. Find the total salary paid per department and rank them in descending order of total salary.
2. List employees who have worked on more than one project.
3. Find employees who are working on ongoing projects and order them by salary (highest first).
4. Find the most experienced employee in each department.
5. Find employees whose salary is above the average salary of their department.
6. Rank departments based on total salary + bonus, and rank employees within departments based on total compensation
7. Rank employees based on number of projects + average project duration and rank departments based on average project duration
8. Rank project managers based on number of employees under them and rank employees within project based on salary
9. Rank departments by total bonus distributed, and within each department, rank employees based on bonus received
10. Rank employees based on years of experience and project count, and rank departments based on average experience



CTEs Basic Problems

1. Write a CTE that retrieves employees along with their department and project details.
2. Use a CTE to find employees who have worked on more than one project.
3. Create a CTE to find employees earning more than the average salary of their department.
4. Use a CTE and JOINs to fetch employees who joined in the last two years along with their project names.
5. Create a CTE to calculate department-wise salary statistics (sum, avg, max).
6. Use a CTE with RANK() to find the top 5 highest-paid employees.
7. Write a CTE to find employees who have the longest tenure in their department.
8. Use a CTE with GROUP BY to count employees by department and classify them as Small, Medium, or Large.

9. Create a recursive CTE to find employees with a reporting hierarchy.
10. Write a query that uses a CTE, JOINs, and RANK() to find the second-highest-paid employee in each department.



Advanced CTE

1. Find departments with total compensation (salary + bonus) > 300,000 and rank employees within those departments by compensation.
2. Find departments where average experience > 3 years and within those departments, rank employees by project count.
3. Identify project managers (department heads) whose department's total bonus exceeds 50,000, rank departments and rank employees in those departments by bonus.
4. Find top 2 departments based on avg project duration and rank employees within departments based on joining date (experience).
5. Find employees who worked on more than one completed project, belong to departments with avg salary > 55k, and rank them by salary and project count.
6. Find departments where total number of employees > 5 and rank employees by total compensation (salary + bonus) & experience.
7. Identify employees who worked in more than 2 projects, belong to departments where the dept head name starts with 'M', and rank by salary & number of projects.
8. Find departments where total project count > 5, calculate average project duration, rank departments by duration, and rank employees within departments by number of completed projects.
9. Find employees who received bonuses greater than department average bonus, rank departments by total bonus, and rank employees by salary + bonus.
10. Find departments where the department head's name contains 'a', average employee experience > 4 years, and rank employees within those departments by project count and total compensation (salary + bonus).





Stored Procedure Tasks

1. Employee Experience Categorization & Bonus Status

Create a stored procedure that fetches each employee's full name, their years of experience (calculated using the joining date), and categorizes their experience level as 'Senior' (more than 5 years), 'Mid-Level' (2 to 4 years), or 'Junior' (less than 2 years). Additionally, include a column to check whether the employee has received a bonus or not. Use CASE statements to implement categorization and status display. This procedure should involve a LEFT JOIN between emp_table and bonus.



2. Department-wise Salary and Bonus Summary

Develop a stored procedure that generates a department-level summary report. The output should include the department name, total salary paid to employees, total bonuses distributed, and the number of employees in each department. Implement GROUP BY and aggregation functions, and use appropriate JOINS between emp_table and bonus tables.

3. Employee Project Completion Status

Design a procedure to retrieve the list of all employees along with their respective project names and current status. Use a CASE statement to convert the project status into more readable values: 'Done' for completed projects, 'In Progress' for ongoing projects, and 'Unknown' if no project is assigned. This procedure should involve JOINS between emp_table and projects.



4. Above-Average Salary Employees

Create a stored procedure that identifies employees earning above the average salary in their department. You need to implement a correlated subquery to calculate the department-wise average salary and compare each employee's salary accordingly.

5.Formatted Employee Names and Joining Month

Construct a procedure to display employees' names formatted properly (first letter uppercase, rest lowercase). Additionally, extract and display the month name from the employee's joining date using DATE functions. This will require using String Functions like UPPER(), LOWER(), CONCAT(), and date extraction functions.



6. Employees under Department Heads Starting with 'M'

Write a procedure to list all employees who are part of departments headed by department heads whose names start with the letter 'M'. This requires JOINing the emp_table and department_head tables and applying a LIKE filter.

7. Department-wise Average Employee Experience Summary

Develop a stored procedure to summarize the average experience of employees in each department. You should calculate the difference in years between the current date and the joining date. The procedure should display the department name, average experience in years, and employee count. GROUP BY and DATE functions will be used here.

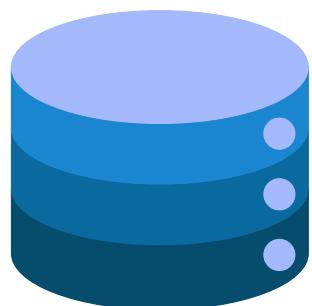


8. Employee Compensation Ranking Using CTE

Design a stored procedure that calculates each employee's total compensation by adding salary and bonus. Use a Common Table Expression (CTE) to prepare the data, then apply a Ranking Function to rank employees within their department based on total compensation. Involve JOINs between emp_table and bonus.

9. Employees with More Than One Project

Create a procedure to list employees who have been assigned to more than one project. Include their names and the count of projects they have worked on. This will require JOINing the emp_table and projects tables, applying GROUP BY, and using the HAVING clause to filter employees with more than one project.



10. Yearly Salary Growth Simulation

Create a stored procedure that projects the salary for the next year based on the following criteria:

- If the current salary is less than 50,000, increase by 10%.
- If the salary is between 50,000 and 70,000, increase by 7%.
- If the salary is above 70,000, increase by 5%.



View Creation Tasks

1. Employee Full Name with Department Head

Create a view that combines employee details and department head information. The view should display each employee's full name (first + last name), department, and the name of the department head. Implement JOINs and string concatenation.

2. Department-wise Bonus Summary View

Develop a view that summarizes the total bonuses distributed in each department. Utilize JOINs between emp_table and bonus and GROUP BY department.

3. Formatted Employee Names and Experience View

Create a view that displays employees' names in a properly formatted style (capitalized first letters) along with their years of experience calculated from their joining date.

4. Employee Compensation Summary View

Design a view to display each employee's salary, bonus, and total compensation. JOIN emp_table and bonus, and use COALESCE to handle cases where no bonus is assigned.

5. Project Details Summary View

Create a view combining project and employee data. Display project name, employee name, employee department, and project status. Use JOINs between projects and emp_table.



Thank You

To know more about our courses

+91 89517 96123 | +91 9916961234 | +91 89517 85123



SCAN NOW