**DATA DIVE SERIES SQL**

Here’s a comprehensive SQL assessment designed to help you practice and reinforce your SQL skills across basic, easy, medium, and advanced levels. This set of 50 questions includes essential SQL topics and tasks that cover creating tables, inserting data, and performing various operations on that data.  
  
1. Setting Up the Dataset  
Before you start the questions, you’ll need to create a sample database with a few tables that we’ll reference throughout the assessment. Here’s a sample database schema to get started:  
  
Database: Company  
Tables  
Employees: Stores information about employees in the company.  
Columns: EmployeeID (INT, Primary Key), FirstName (VARCHAR), LastName (VARCHAR), DepartmentID (INT), Salary (DECIMAL), JoiningDate (DATE), Position (VARCHAR)  
Departments: Stores information about departments in the company.  
Columns: DepartmentID (INT, Primary Key), DepartmentName (VARCHAR), ManagerID (INT, Foreign Key referencing Employees)  
Projects: Stores information about projects within the company.  
Columns: ProjectID (INT, Primary Key), ProjectName (VARCHAR), StartDate (DATE), EndDate (DATE), Budget (DECIMAL)  
EmployeeProjects: A many-to-many relationship table that connects employees with projects.  
Columns: EmployeeID (INT, Foreign Key), ProjectID (INT, Foreign Key), HoursAllocated (INT)

**Basic Level**

1. Write a query to retrieve all columns from the Employees table.
2. Retrieve only the FirstName and LastName of all employees in the Employees table.
3. Find all employees with a salary greater than 60000.
4. Write a query to find employees who joined the company after January 1, 2020.
5. Use ORDER BY to sort employees by Salary in descending order.
6. Display distinct departments from the Departments table.
7. Find the count of employees in the Employees table.
8. Use LIKE to find employees whose LastName starts with 'S'.
9. Retrieve employees who don’t belong to any department.
10. Calculate the average salary of all employees.

**Easy Level**

1. Retrieve all departments where the number of employees is more than 5.
2. Write a query to retrieve employees who earn between 40000 and 70000.
3. Show the total number of employees working in each department.
4. Use JOIN to display employees along with their department names.
5. Write a query to find employees who work in the 'IT' department.
6. Find all projects where the budget is less than 400000.
7. Display employees and the number of projects they are working on.
8. Retrieve a list of employees who work on multiple projects.
9. Write a query to list employees, their departments, and their projects.
10. Find employees with salaries higher than the average salary.
11. Write a query to calculate the total salary by department.
12. Display employees and their joining dates, showing the most recent hires first.
13. Write a query to find the maximum, minimum, and average budget of projects.
14. Create a view to show employees and their department names.
15. Retrieve the top 2 highest-paid employees.
16. Display the number of projects per department.
17. Write a query to get the employee who spends the most hours on a project.
18. Identify departments where the average employee salary is above 60000.
19. Write a query to get employees who work on all projects.

30.Find departments that have the highest budget allocation for their projects.

1. Write a query to retrieve employees who work on the highest number of projects.
2. Display a list of managers who have more than 10 employees under them.
3. Find projects where the total allocated hours exceed 500.
4. Write a recursive query to retrieve all managers and their reporting employees.
5. Calculate the percentage of employees in each department.
6. Retrieve employees who have been in the company for more than 5 years.
7. Write a query to find overlapping projects (where start and end dates overlap).

**Advanced Level**

39.Write a query to list the department with the most employees.

40.Use a subquery to find employees whose salaries are above the average salary for their department.

1. Find all departments with budgets that exceed the average department budget.
2. Create a stored procedure to calculate the total salary expense by department.
3. Write a trigger to prevent any updates to the EmployeeProjects table if the HoursAllocated exceeds 40.
4. Write a query to retrieve the average salary, excluding the top 10% of earners.
5. Write a query to retrieve departments with more than 1 employees, sorted by the average salary of the department.
6. Create a function that returns the tenure (in years) of an employee based on their joining date.
7. Write a CTE (Common Table Expression) to list employees with their cumulative hours on projects.
8. Write a query to calculate the difference between the maximum and minimum salaries in each department.

Sambhav Chathly

LinkedIn- linkedin.com/in/sambhavchathly/