Project Title: SQL Querying for Real-World Database Management

Project Introduction:

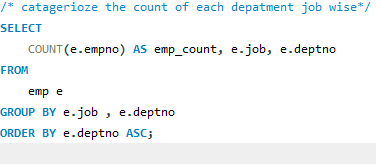
In this SQL project, I have developed a comprehensive set of queries to analyze and manage a relational database that simulates a company's employee and department structure. The project utilizes an SQL database with multiple tables — emp, dept, and salgrade — which represent employee records, department information, and salary grading, respectively. The project is designed to showcase a wide range of SQL querying techniques, from basic selection and joins to advanced window functions and analytical queries. By applying these queries, I demonstrate the ability to work with real-world data, perform in-depth analysis, and extract meaningful insights for business decision-making.

Key Project Details:

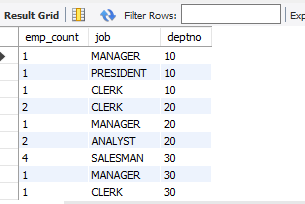
* Database Schema: The project uses three main tables:
  + emp: Contains employee information such as ID, name, job title, salary, commission, manager ID, and hire date.
  + dept: Includes details about the departments, such as department number, name, and location.
  + salgrade: Stores salary grade information, with associated salary ranges.
* SQL Queries: A series of 30+ complex queries that address various business needs, ranging from basic data retrieval to advanced analytics.
* Techniques Applied:
  + Aggregation (COUNT, SUM, AVG)
  + JOINs (INNER, LEFT, RIGHT)
  + Subqueries and Correlated Subqueries
  + Grouping and Sorting (GROUP BY, ORDER BY)
  + Window Functions (RANK, PARTITION BY, SUM() OVER)
  + Date Manipulation and Formatting
  + Conditional Logic (CASE WHEN)
* Objective: The goal of this project is to demonstrate proficiency in SQL, enhance data analysis skills, and solve practical database-related challenges efficiently.

Queries Used:

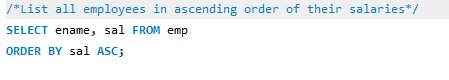
1. Categorize the count of each department job-wise.



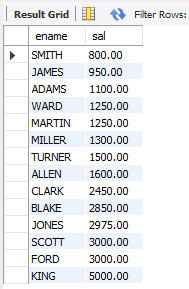
OUTPUT :



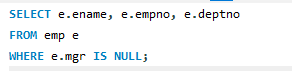
1. List all employees in ascending order of their salaries.



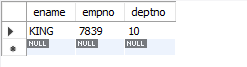
OUTPUT:



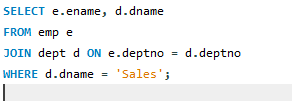
3.Give details of employees who have no manager.



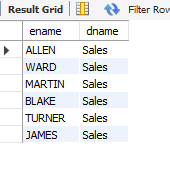
OUTPUT:



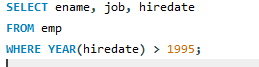
4.Fetch all employees working in the "Sales" department.



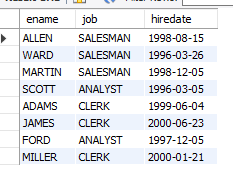
OUTPUT:



5.Find the name and job of employees whose hire date is after 1995.



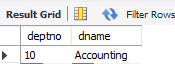
OUTPUT:



6.Retrieve details of all departments located in "New York."



OUTPUT:



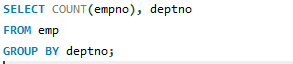
7..Fetch the names of all employees who earn a commission greater than 500.



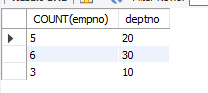
OUTPUT:



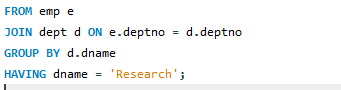
8.Find the total number of employees in each department.



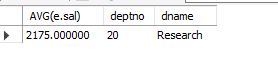
OUTPUT:



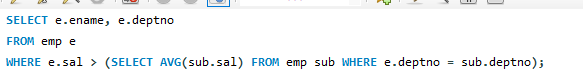
9.Calculate the average salary of employees in the "Research" department.



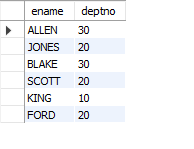
OUTPUT:



10.Find employees whose salary is higher than the average salary of their department.



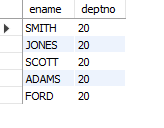
OUTPUT:



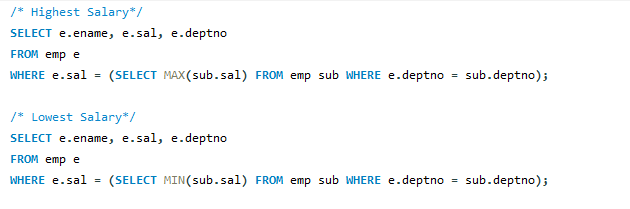
11.Identify employees who work in the same department as "SMITH."



OUTPUT:

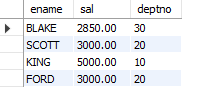


12.Calculate the highest and lowest salaries in each department.

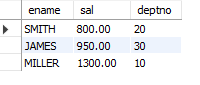


OUTPUT:

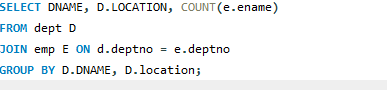
Highest Salary:



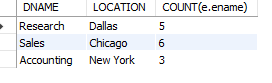
Lowest Salary:



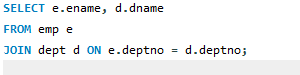
13.Display department name, location, and the total number of employees in each department.



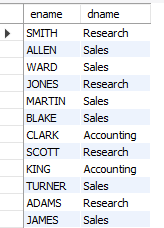
OUTPUT:



14.Retrieve the names of employees along with their department names.



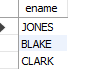
OUTPUT:



15.Find all employees working under a manager whose name is "KING."



OUTPUT:



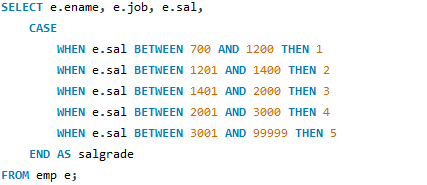
16.Format the hire date to display only the month and year.



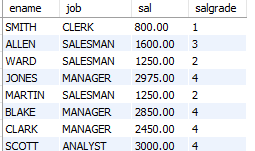
OUTPUT:



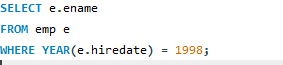
17.Display employee name, job, salary, and grade from the emp and salgrade tables.



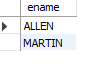
OUTPUT:



18.Retrieve employees who were hired in the year 1998.



OUTPUT:



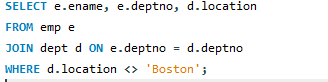
19.Find employees whose job is "SALESMAN" and have a commission greater than their salary.



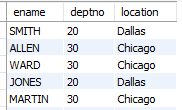
OUTPUT:



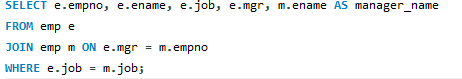
20.Find employees who belong to a department with a location other than "Boston."



OUTPUT:



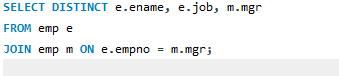
21.Find employees who have the same job as their manager.



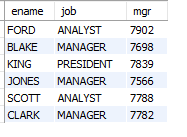
OUTPUT:



22.Get the empno, name, and job of all the managers.



OUTPUT:



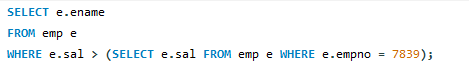
23.Find the names of employees whose salary is greater than the average salary of all employees.



OUTPUT:



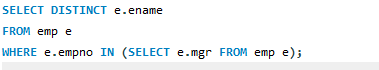
24.Find all employees whose salary is greater than the salary of the employee with empno = 7839.



OUTPUT:



25.Find all employees who are managers, i.e., those who are present in the mgr column of other employee rows.



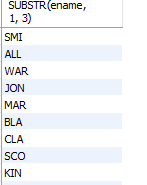
OUTPUT:



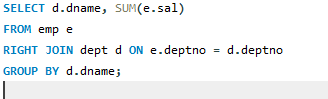
26.Find the first three characters of each employee's name from the emp table.



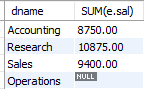
OUTPUT:



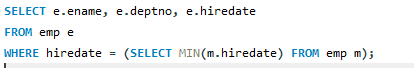
27.List all departments and the total salary of employees in each department (include departments with no employees).



OUTPUT:



28.Find the employee(s) who have been working the longest.



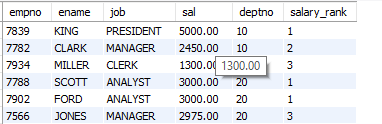
OUTPUT:



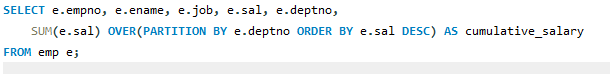
29.Assign a rank to employees based on their salaries within their departments.



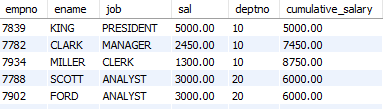
OUTPUT:



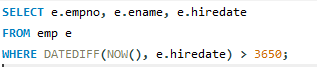
30.Calculate the cumulative salary of employees department-wise.



OUTPUT:



31.Identify employees who have been working in the company for more than 10 years.



OUTPUT:

