Data Analyst Project: SQL Analysis Using GitHub's DataCharmer Dataset

Project Overview:

In this project, I performed data analysis on a variety of business-related questions using SQL. The dataset used for this analysis was sourced from the DataCharmer GitHub repository. I used SQL Workbench to execute the queries and extract valuable insights from the database.

Tools and Technologies:

- **SQL Workbench**: To write and execute SQL queries for data analysis.
- **GitHub Data (DataCharmer Repository)**: The dataset contains information on employees, salaries, departments, and more.

Questions and SQL Queries:

1. List of employees by department

Include employee number, first name, last name, department number, and department name.

```
SELECT E.emp_no, E.first_name, E.last_name, D.dept_no, D.dept_name
FROM employees AS E
JOIN dept_emp AS DE ON E.emp_no = DE.emp_no
JOIN departments AS D ON DE.dept_no = D.dept_no;
```

2. Retrieve all the salary records of a given employee (by employee number)

Include employee number, salary, from date, and to date

```
SELECT *
FROM salaries
WHERE emp_no = '10001'
ORDER BY salary DESC
LIMIT 1;  -- Reflects the highest salary
```

3. Find all employees who have held a specific title (e.g., 'Engineer')

Include employee number, first name, last name, and title.

```
SELECT E.emp_no, E.first_name, E.last_name, T.title
FROM employees AS E
JOIN titles AS T ON E.emp_no = T.emp_no
WHERE T.title = 'Engineer'
ORDER BY 1;
```

4. List all departments along with their current managers

Include department number, department name, manager's employee number, first name, and last name.

```
SELECT DM.dept_no AS department_number, D.dept_name AS department_name,
    E.emp_no AS managers_employee_number, E.first_name, E.last_name
FROM dept_manager AS DM
JOIN departments AS D ON DM.dept_no = D.dept_no
JOIN employees AS E ON DM.emp_no = E.emp_no;
```

5. Count the number of employees in each department

Include department number, department name, and employee count.

```
SELECT D.dept_no AS department_number, D.dept_name AS department_name,
    COUNT(DE.emp_no) AS employee_count
FROM dept_emp AS DE
JOIN departments AS D ON DE.dept_no = D.dept_no
GROUP BY department_number;
```

6. Find all employees born in a specific year (e.g., 1954)

Include employee number, first name, last name, and birth date.

```
SELECT emp_no AS employee_number, first_name, last_name, birth_date
FROM employees
WHERE YEAR(birth_date) = 1954;
```

7. Find all employees hired in the last 50 years

Include employee number, first name, last name, and hire date.

```
SELECT emp_no AS employee_number, first_name, last_name, hire_date
FROM employees
WHERE hire_date >= DATE_SUB(CURDATE(), INTERVAL 50 YEAR);
```

8. Calculate the average salary for each department

Include department number, department name, and average salary.

```
SELECT D.dept_no AS department_number, D.dept_name AS department_name,
    AVG(S.salary) AS average_salary
FROM dept_emp AS DE
JOIN departments AS D ON DE.dept_no = D.dept_no
JOIN salaries AS S ON DE.emp_no = S.emp_no
GROUP BY department_name
ORDER BY department_number;
```

9. Find the gender distribution (number of males and females) in each department

Include department number, department name, count of males, and count of females.

```
SELECT D.dept_no AS department_number, D.dept_name AS department_name,
    SUM(E.gender = 'M') AS count_of_males, SUM(E.gender = 'F') AS count_of_females
FROM departments AS D
JOIN dept_emp AS DE ON D.dept_no = DE.dept_no
JOIN employees AS E ON DE.emp_no = E.emp_no
GROUP BY department_name
ORDER BY department_number;
```

10. Find the employees who have served the longest in the company

Include employee number, first name, last name, and number of years served.

```
SELECT emp_no AS employee_number, first_name, last_name,
    TIMESTAMPDIFF(YEAR, hire_date, CURDATE()) AS number_of_years_served
FROM employees
ORDER BY number_of_years_served DESC
LIMIT 5;
```

Analysis and Insights:

1. **Employee Distribution**: The SQL queries provided insights into employee distribution across departments, helping identify departments with more or fewer employees.
2. **Salary Analysis**: By retrieving salary records and calculating averages for departments, this project highlighted salary disparities and helped analyze trends in compensation across different departments.
3. **Gender Diversity**: The gender distribution query offered insights into the gender composition across departments, which could inform diversity initiatives.
4. **Employee Tenure**: The query that identifies the longest-serving employees provides key insights for workforce planning and recognizing employee loyalty.

Conclusion:

This project provided valuable insights into employee demographics, department structures, salary trends, and tenure. The findings can guide HR, management, and business leaders in making data-driven decisions regarding employee retention, department growth, and salary planning.