PostgreSQL Practical Test – SQL Skills Assessment

This test is designed to assess your **SQL skills**, covering topics from **basic database operations to advanced queries**.

🔹 **Instructions:**

✓ Read each question carefully and enter your SQL query in the provided space.

✓ Ensure correct **syntax and formatting** for your queries.

✓ Queries should be optimized for **efficiency and best practices**.

✓ You may use **pgAdmin** or any PostgreSQL environment to test your queries before submission.

🔹 **Test Details:**

📌 **Total Marks:** 100

⏳ **Duration:** 60 minutes

📊 **Topics Covered:** Database Basics, DML Operations, Filtering, Joins, Aggregations, String & Date Functions, and Set Operations.

📢 **Note:** Plagiarism or direct copy-pasting from external sources will result in disqualification.

Best of luck! 🍀

Section A: Basic Concepts (20 Marks)

1. Write a command to create a database named company_db.
2. Create a table employees & departments with the following structure:
3. Difference between Drop, Truncate & Delete.
4. List all the **numeric** and **string** data types available in PostgreSQL.
5. Explain the difference between VARCHAR, TEXT, and CHAR data types.

Section B: Data Manipulation (20 Marks)

1. Insert the following records into the employee's table:
2. Write a query to **increase the salary** of all employees by **10%**.
3. Write a query to delete employees **hired before 2022**.
4. Retrieve all employees who earn between 50000 and 80000.
5. Select employees whose first name **starts with 'J'** using the LIKE operator.
6. Explain the RETURNING clause in PostgreSQL with an example.

Section C: Sorting & Aggregation (20 Marks)

1. Retrieve all employees and sort them **by salary in descending order**.
2. Retrieve the **top 3 highest-paid** employees.
3. Find the **total salary expense** of the company.
4. Find the **average salary** of employees and filter only those with an average salary greater than 70000 using the HAVING clause.

Section D: String & Date Functions (15 Marks)

1. Concatenate the first_name and last_name with a space in between.
2. Extract the **year** from the hire_date column.
3. Convert all first_name values to **uppercase**.
4. Find the difference in **years** between the current date and hire_date.
5. Use DATE_TRUNC to round off hire_date to the nearest **month**.

Section E: Advanced Filtering & Conditional Logic (10 Marks)

1. Use COALESCE to replace NULL salaries with 50000.
2. Find the **highest and lowest salaries** using GREATEST and LEAST.
3. Use NULLIF to prevent division by zero when calculating salary percentages.

Section F: Joins & Set Operations (15 Marks)

1. Given a departments table with dept_id and dept_name, write a query to **join employees and departments** on dept_id using an INNER JOIN.
2. Retrieve all employees, ensuring that those **without a department are also included** (use LEFT JOIN).
3. Find employees **who are not present in another table** using EXCEPT.
4. Explain the difference between UNION and UNION ALL with an example.