**📘 Assignment: Exploring SQL Window Functions**

**📌 Total Marks: 10**

**👥 Teamwork Required: Two students per team**

**🔔 Important Instructions**

1. Collaborate with **one classmate** and form a group.
2. **Create a GitHub repository** for your group project.
3. Choose a **funny group name** for your repository (e.g., *ThePrimaryKeys*, *Commit\_the\_Query*, etc.).
4. **Add your instructor** as a collaborator on GitHub.
   * GitHub Username: **ericmaniraguha**
5. Each student **must contribute** code to the GitHub repository.
   * **Zero contribution = Zero marks**
6. Collaboration and equal participation are key. Contributions will be **tracked via GitHub**.

**🎯 Objective**

Students will:

* Work with **SQL Window Functions** on a chosen dataset.
* Perform **analytical queries** using:
  + LAG(), LEAD()
  + RANK(), DENSE\_RANK(), ROW\_NUMBER()
  + Aggregate window functions
* Provide **clear explanations** and **real-life applications** of each query.
* **Document** and **push** all work to GitHub.

**🧠 Tasks & Queries (10 Points Total)**

**Choose or create a dataset** from any domain: employees, sales, students, orders, products, etc.

**1. 📊 Compare Values with Previous or Next Records**

* Use LAG() and LEAD() to compare a chosen column (e.g., salary, sales, scores, prices).
* Indicate whether the value is **HIGHER**, **LOWER**, or **EQUAL** compared to the previous record.

**2. 🏅 Ranking Data within a Category**

* Use RANK() and DENSE\_RANK() to rank items within a category (e.g., employees by department, students by class).
* **Explain** the difference between RANK() and DENSE\_RANK() with **examples**.

**3. 🥇 Identifying Top Records**

* Fetch the **top 3 records** from each category (e.g., top 3 salaries per department).
* Handle duplicate values properly using **ranking functions**.

**4. ⏳ Finding the Earliest Records**

* Retrieve the **first 2 records** from each category based on a **date column** (e.g., hire date, registration date).
* Clearly **explain** the logic behind your query.

**5. 📈 Aggregation with Window Functions**

* Select all records and calculate:
  + The **maximum value per category** using PARTITION BY.
  + The **overall maximum value** across all records.
* Show the difference between **category-level** and **global** aggregation.

**📝 Submission Guidelines**

1. **GitHub Repository Requirements:**
   * Include:
     + SQL scripts for:
       - Table creation
       - Data insertion
       - Window function queries
     + README.md with:
       - Explanations
       - Screenshots
       - Real-life use cases
2. Add the instructor (**ericmaniraguha**) as a **collaborator**.
3. Submit the **GitHub repository link** before the deadline.

**⏰ Deadline:**

**April 17, 2025 – 11:59 PM**