# Individual Assignment I: PL/SQL Window Functions Mastery Project

**Course:** Database Development with PL/SQL (INSY 8311) **Instructor:** Eric Maniraguha | eric.maniraguha@auca.ac.rw **Assignment Date:** September 19, 2025 **Groups:** A, B, C, D **Deadline:** September 29, 2025 (11:59 PM) – **No late submissions accepted**

## Message to Students

This assignment is critical for your learning and future assessments. The concepts and implementations you master here will appear in upcoming quizzes and the midterm.

## To Maximize Your Grade

1. **Read and follow all instructions carefully.**
2. **Use the required formats exactly.**
3. **Submit a complete project on time** (late = zero).
4. **Provide clear, specific analysis and professional documentation.**

## Academic Integrity Guidelines

- **Read the instructions carefully.**
- **Prepare your documents clearly and professionally.**
- **Avoid copy-paste** from online sources or other students' work — this will count as plagiarism.
- **Do not use ChatGPT or any other AI tool** to generate your answers. If detected, you will receive **zero marks for the entire assignment**.
- **Learn to paraphrase** and express ideas in your own words. This is essential for both your integrity and learning.

## Assignment Overview

**Objective:** Demonstrate mastery of PL/SQL window functions by solving a realistic business problem, implementing analytical queries, and documenting results in a professional GitHub repository.

**Weight:** 10 raw points (aggregated to 5 points in final grade).

## Step-by-Step Requirements

### Step 1: Problem Definition (2 pts)

Define a **specific, measurable** business scenario.

**Format:**

- **Business Context** (company type, department, industry)
- **Data Challenge** (2–3 sentences explaining the problem)
- **Expected Outcome** (decision/insight from analysis)

*High scoring:* "Identify top products per region, analyze customer frequency, and segment customers for marketing." *Low scoring:* "Analyze sales data to find patterns."

---

## Step 2: Success Criteria (part of 2 pts)

Define **exactly 5 measurable goals**:

1. Top 5 products per region/quarter → `RANK()`
2. Running monthly sales totals → `SUM() OVER()`
3. Month-over-month growth → `LAG()/LEAD()`
4. Customer quartiles → `NTILE(4)`
5. 3-month moving averages → `AVG() OVER()`

---

## Step 3: Database Schema (part of 2 pts)

Design **minimum 3 related tables** with foreign keys.

| Table | Purpose | Key Columns | Example Row |
|---|---|---|---|
| customers | Customer info | customer_id (PK), name, region | 1001, John Doe, Kigali |
| products | Product catalog | product_id (PK), name, category | 2001, Coffee Beans, Beverages |
| transactions | Sales records | transaction_id (PK), customer_id (FK), product_id (FK), sale_date, amount | 3001, 1001, 2001, 2024-01-15, 25000 |

**Include an ER diagram.**

---

## Step 4: Window Functions Implementation (4 pts)

Implement **all 4 categories** (1 point each):

1. **Ranking**: `ROW_NUMBER()`, `RANK()`, `DENSE_RANK()`, `PERCENT_RANK()` *Use case: Top N customers by revenue*
2. **Aggregate**: `SUM()`, `AVG()`, `MIN()`, `MAX()` with frame comparisons (`ROWS` vs `RANGE`) *Use case: Running totals & trends*
3. **Navigation**: `LAG()`, `LEAD()`, growth % calculations *Use case: Period-to-period analysis*
4. **Distribution**: `NTILE(4)`, `CUME_DIST()` *Use case: Customer segmentation*

**Format (for each function):**

- Query (with comments)
- Screenshot (results clearly visible)

- Interpretation (2–3 sentences)

---

### Step 5: GitHub Repository (1.5 pts)

- Repo name: `plsql-window-functions-[lastname]-[firstname]`
- Visibility: Public
- Must include:
  - SQL scripts (error-free)
  - Screenshots (organized, clear)
  - Professional README (problem, schema, queries, insights, references)

---

### Step 6: Results Analysis (2 pts)

Write insights in 3 layers:

1. **Descriptive** – What happened? (patterns, trends, outliers)
2. **Diagnostic** – Why? (causes, comparisons, influencing factors)
3. **Prescriptive** – What next? (recommendations, business actions)

---

### Step 7: References (0.5 pts)

- Minimum 10 sources (Any style)
- Include docs, tutorials, academic papers, business resources
- Add in README:

  > “All sources were properly cited. Implementations and analysis represent original work. No AI-generated content was copied without attribution or adaptation.”

---

## Academic Integrity

**Not allowed:** plagiarism, uncredited AI output, sharing solutions, late submission. **Allowed:** official docs, tutorials (with citation), adapting examples, concept discussions.

---

## Grading Breakdown (10 pts → 5 pts final)

- **Problem Definition & Criteria**
- **Window Functions Implementation**
- **Results Analysis**
- **Technical Quality & GitHub**
- **References & Integrity Statement**

---

## Submission

**To:** eric.maniraguha@auca.ac.rw **Subject:** ☑ – Repository Link

**Body:**

```
Repository Link: [GitHub URL]
Business Problem: [1 sentence]
Key Findings: [2 insights]
Sources Consulted: [# references]
```

---

## Final Checklist

- □ Repo is public & accessible
- □ SQL runs without errors
- □ 20+ screenshots included
- □ README is professional
- □ Problem + schema + functions + analysis + references complete
- □ Integrity statement included

---

## Professional & Ethical Note

> *"Whoever is faithful in very little is also faithful in much."* – Luke 16:10

As database professionals, uphold accuracy, confidentiality, and integrity. Your reputation is built on consistent honesty, quality, and responsibility.

---