Adventist University of Central Africa

P.O. Box 2461 Kigali, Rwanda | www.auca.ac.rw | info@auca.ac.rw

Faculty of Information Technology

Capstone Project Academic Year: 2024-2025

Course Code & Name: Database Development with PL/SQL INSY 8311

Lecturer: Eric Maniraguha | eric.maniraguha@auca.ac.rw Date: April 20, 2025

All Groups (Mon, Tues, Wed, Thu)

Instructions: The project will be completed in phases, each with specific requirements. Please keep the following points in mind:

1. **Respect Deadlines:**
   - Deadlines are strict. Late submissions will not be accepted and will result in a score of zero for that phase.
   - Plan and manage your time effectively to ensure timely submissions.
2. **Proposal Submission:**
   - This is a personal project. Late submissions of the proposal will not be accepted and will result in zero points.
3. **Active Participation:**
   - You are expected to fully engage with the project and understand every aspect of it.
   - Ensure you are actively working and contributing to each phase.
4. **Focus on Quality:**
   - Aim for high-quality work in every aspect of the project. Quality will be a key factor in project acceptance.
   - Review your work carefully before submission to avoid last-minute delays.
5. **Exploration of Ideas:**
   - Feel free to explore different ideas during the implementation phase.
   - You may submit multiple ideas, and I will assist you in selecting the best one in the first phase.
6. **Project Acceptance:**
   - Not all projects will be accepted. If your proposal does not meet the required standards, you will be given two additional days to refine or rewrite it.
   - Make sure your proposal is well thought out and clearly defined to avoid this situation.
7. **Report Submission via GitHub:**
   - After the problem statement, you must submit the report for each subsequent phase via GitHub.
   - I will be monitoring all submissions through GitHub, so make sure to keep it updated regularly.

Note: This project is considered your final exam.

Project Outline: PL/SQL Oracle Database Design and Implementation

This project is a comprehensive, multi-phase individual assignment focused on Oracle database design and PL/SQL programming. You will select a complex, real-world problem and develop a working database solution through multiple clearly defined phases.

I. Phase: Problem Statement and Presentation

- **Deadline: March 25, 2025**, at **11:59 PM**

Objective

Identify a real-world problem that requires a PL/SQL-based Oracle database solution. The problem should be complex enough to involve **multiple entities, relationships, and business logic**.

Problem Statement Guidelines

Your submission should clearly outline:

- **Problem Definition:** What issue are you solving?
- **Context:** Where and how will the system be used? (e.g., hospitals, universities, businesses)
- **Target Users:** Who will benefit from your system?
- **Project Goals:** What outcomes are expected? (e.g., automate workflows, improve accuracy, enhance data security)

Presentation Requirements

Prepare a brief in-class presentation that explains:

- The **project objectives**
- The **main entities** in your database
- The **anticipated benefits** of your solution

Document Format

Please submit your work as a **PowerPoint presentation** with the following formatting:

- **Font:** Helvetica
- **Use bullet points** throughout the slides
- **Number of slides:** Maximum of **three slides** (you may submit one or two slides only)
- Ensure clarity and consistency in layout

File Naming Convention

- Name your file using the following format:
- **GrpName_StudentId_FirstName_ProjectName_DB.**
  - For example, *mon_12121_peter_studentMS_db*
- Make sure to use **underscores** (_) between each part.

II. Phase: Business Process Modeling (Related to Management Information Systems - MIS)

Objective

In this phase, you will model a business process relevant to Management Information Systems (MIS). The goal is to visualize how information flows within a system, how different entities interact, and how decision-making is supported through MIS.

Tasks and Deliverables

1. **Define the Scope**
   - Clearly outline the business process you are modeling.
   - Ensure the process is relevant to MIS, such as **order processing, inventory management, patient record handling, employee payroll, or customer service management**.
   - Define the **objectives and expected outcomes** of the process.

2. **Identify Key Entities**
   - List and describe all key **actors, departments, or systems** involved in the process.
   - Explain their **roles, responsibilities, and interactions** within the system.
   - Examples of entities: **users (employees/customers), databases, applications, managers, automated workflows**.

3. **Use Swimlanes for Clarity**
   - Organize your diagram using **swimlanes** to separate different actors or departments.
   - Ensure the swimlanes make it easy to understand who is responsible for each step of the process.

4. **Apply UML/BPMN Notations**
   - Use **UML (Unified Modeling Language) or BPMN (Business Process Model and Notation)** to represent your process visually.
   - Maintain consistency with proper symbols and conventions.
   - Include key elements such as **start and end points, tasks, decisions, and data flows**.

5. **Ensure a Logical Flow**
   - The process model should clearly **flow from start to end** with all interactions properly mapped.
   - Highlight **decision points, inputs, outputs, and system interactions**.
   - Ensure that **dependencies between steps** are correctly structured.

6. **Prepare an Explanation**
   - Write a brief **description of your diagram** (one page maximum).
   - Explain:

- **The main components and their interactions.**
- **How the process supports MIS functions** (e.g., improves decision-making, streamlines operations).
- **Why this process is important for organizational efficiency.**
- Tools Lucichart or draw.io

Additional Notes

- If you have **not taken the MIS course**, research how **information systems support business processes** to ensure your model aligns with MIS principles.
- Exceptional work may receive **bonus points** for **clarity, creativity, and technical accuracy**.
- This phase builds the foundation for later stages of database design, so ensure accuracy and completeness.

III. Phase: Logical Model Design

In this phase, you will design a **detailed logical data model** for your project, ensuring it aligns with the problem statement defined in **Phase 1** and the business process modeled in **Phase 2**.

Tasks and Deliverables

1. **Entity-Relationship (ER) Model:**
   - Identify and define all **entities** relevant to your project.
   - Specify **attributes** for each entity, including data types.
   - Clearly define **primary keys (PKs)** and **foreign keys (FKs)** to establish relationships.

2. **Relationships & Constraints:**
   - Establish and document all **relationships** between entities (e.g., one-to-one, one-to-many, many-to-many).
   - Apply **constraints** such as **NOT NULL**, **UNIQUE**, **CHECK**, and **DEFAULT** where necessary.

3. **Normalization:**
   - Ensure that your database design **eliminates redundancy** and adheres to normalization principles (**at least 3rd Normal Form - 3NF**).

4. **Handling Data Scenarios:**
   - Validate that your logical model can handle different **real-world data scenarios** effectively.

5. **Presentation & Feedback:**
   - Submit your logical model for review and feedback before proceeding to the next phase.
   - Your model should be **well-documented** and **clearly labeled** for readability.

IV. Phase: Database (Pluggable Database) Creation and Naming

This phase ensures that the database is created and managed effectively, with appropriate access controls and monitoring tools in place.

Tasks and Deliverables

1. **Database Creation:**
    - Create a PL/SQL Oracle database using the following naming format: GrpName_StudentId_FirstName_ProjectName_DB. For example, mon_12121_kaka_studentMS_db.
    - The **password** should be your **first name** (e.g., patrick).
    - Ensure the user has **super admin** privileges.

    Note: All activities and data will be stored inside the database you create.

2. **Oracle Enterprise Manager (OEM):**
    - Set up **Oracle Enterprise Manager (OEM)** to monitor and manage the database.
    - During the reporting period, you will be required to share **screenshots from OEM** to demonstrate your progress and work.

Note: You are required to report your progress via **GitHub**, including a screenshot that clearly shows your project names, to avoid plagiarism.

V. Phase: Table Implementation and Data Insertion

This phase emphasizes building the physical representation of the database and ensuring the data is both reliable and structured properly to support project operations and queries.

Tasks and Deliverables

1. **Table Creation:**
    - Implement the logical design by creating tables within the Oracle database. These tables should align with the requirements of your project and reflect the necessary data structure.
2. **Data Insertion:**
    - Insert **realistic** and **meaningful data** into the tables to support testing and demonstration of your project. This data should represent typical use cases and scenarios within the scope of the project.

3. **Data Integrity:**
   - Ensure **data integrity** by verifying that all data supports the necessary queries and operations specified in the problem statement. This includes ensuring accuracy, consistency, and validity of the data.
4. **Physical Database Structure:**
   - Convert the logical model into a **physical database structure**. This includes:
     - Creating tables with appropriate columns and data types.
     - Defining primary keys, foreign keys, and indexes where necessary.
     - Setting constraints such as **NOT NULL**, **UNIQUE**, and **CHECK** to ensure data validity and consistency.

VI. Phase: Database Interaction and Transactions

This phase focuses on database interaction, ensuring that operations such as fetching data, performing transformations, and handling errors are done effectively. It also encourages modular programming by using procedures, functions, and packages to organize database operations.

Tasks and Deliverables

1. **Database Operations**:
   - Perform **DML** (Data Manipulation Language) and **DDL** (Data Definition Language) operations to interact with the database, which include:
     - Insert, Update, Delete (DML)
     - Create, Alter, Drop (DDL)
2. **Task Requirements**:
   - **Define a Simple Problem Statement**:
     - Choose a simple problem or dataset to analyze using **Windows functions**. The problem should be relevant to your project and require analytics.
     - Look for opportunities to **group variables** during procedure implementation or function use.
   - **Procedures and Functions**:
     - Create **parameterized procedures** for fetching values from your database.
     - Use **cursors** to retrieve data from tables.
     - Ensure **exception handling** is applied to handle potential errors during database interactions.
   - **Testing**:
     - Perform thorough testing to verify that the queries and database operations are functioning as expected.
   - **Apply Packages**:
     - Create and implement **packages** that contain both procedures and functions for a specific purpose (such as data retrieval, manipulation, etc.).

VII. Phase: Advanced Database Programming and Auditing

In this phase, you will enhance your capstone project by applying advanced PL/SQL programming techniques and implementing auditing mechanisms to solve a real-world problem. The goal is to improve system functionality, automate operations, enforce security rules, and monitor user activity through database programming features such as **triggers**, **procedures**, **functions**, **packages**, and **auditing**.

Tasks and Deliverables

1. **Problem Statement Development**
   - Write a clear and concise problem statement that highlights the need for advanced database programming in your system.
   - Justify why triggers, packages, and auditing are necessary to solve a specific challenge or add value to your project.
   - Your system must include a **restriction rule** to:
     - **Prevent employees from performing any table manipulations (INSERT, UPDATE, DELETE)** during **weekdays** (Monday to Friday).
     - **Additionally, block activity during public holidays for the upcoming month only.**
     - You must use a **reference table** or **static list** to store the holiday dates for that month, which will be checked in your trigger logic.
2. **Trigger Implementation**
   - Implement the following:
     - **Simple Triggers** (e.g., BEFORE INSERT, AFTER DELETE) to enforce specific actions.
     - **Compound Triggers** to handle complex operations involving multiple rows or transactional consistency.
   - Your trigger(s) should:
     - Check the current **day of the week**
     - Refer to your **holiday table**
     - Prevent manipulation on **weekdays and holidays in the upcoming month**

3. **Auditing with Restrictions and Tracking**
   - Implement auditing features that:
     - Log changes to sensitive data
     - Track and record user actions for accountability
   - Apply restrictions using:
     - **Triggers** to block unauthorized access or manipulation
     - **Functions and packages** to automate audit tracking
   - Use the audit table to capture:
     - User ID

    - Date and time of action
    - Operation attempted
    - Status (allowed/denied)
- Explain how this approach enhances security and aligns with your system objectives.

VIII. Phase: Documentation and Demonstration – Reporting

This phase focuses on organizing and presenting your project outcomes through proper documentation and professional presentation. It includes two main deliverables: a GitHub report and a PowerPoint presentation.

1. GitHub Report

Your GitHub repository should include:

- **A short introduction with:**
    - Group members' names
    - Student IDs
    - Problem statement
- **Screenshots of results:**
    - Include only meaningful and relevant screenshots (avoid displaying all inserted data).
- **SQL Queries:**
    - Report all major queries used (e.g., DDL, DML, triggers, procedures).
    - Use a **clear naming convention** for queries and files to maintain structure and readability.

2. PowerPoint Presentation

- Prepare a well-structured PowerPoint presentation with the following content:
    - Introduction
    - Problem Definition
    - Methodology & Approach
    - Results
    - Conclusion, Recommendations, and Future Work

- **Presentation Guidelines:**
    - Maximum of **10 slides** (do not exceed this limit).
    - Summarize the content clearly – remember that detailed documentation is already on GitHub.
    - Use **bullet points** for text.
    - Include **visual aids** (diagrams, screenshots, charts) to support your explanation.
    - References should be included at the end.

- **Submission Instructions**
    - Before the exams, share your PowerPoint presentation with the instructor via email.
    - **Upload your presentation to Google Drive** and **share it** with the instructor's email address.

- This will assist in grading and allow easy access to your presentation materials.

Additional Notes

- **Independent Work:** Each student is expected to work independently and complete all project phases step by step.
- **Creativity & Complexity:** Choose a unique, meaningful problem. Projects that demonstrate originality and thoughtful design will be rewarded.
- **Progress Tracking:** Regular updates and presentations will be required to assess your progress.
- **Use of AI Tools:** AI tools may be used for research and paraphrasing. However, **plagiarism will not be tolerated**. Ensure all content is your original work.
- **No Excuses:** Deadlines are strict. No delays will be tolerated unless you're repeating the course. Stay disciplined and focused throughout.

Final Note:

This project serves as your **Final Exam**. Give it your best effort! Your commitment and creativity will define your success.

Whatever you do, work at it with all your heart, as working for the Lord, not for human masters." *— Colossians 3:23 (NIV)*

Good luck!