# PL/SQL Oracle Database Capstone Project

## Database Development with PL/SQL (INSY 8311)

**Academic Year:** 2025-2026 | **Semester:** I
**Lecturer:** Eric Maniraguha | eric.maniraguha@auca.ac.rw
**Institution:** Adventist University of Central Africa (AUCA)
**Project Completion Date:** December 7, 2025
**All Groups:** Monday, Tuesday, Wednesday, Thursday

## Project Overview

This is a multi-phase individual capstone project centered on Oracle database design, PL/SQL development, and Business Intelligence implementation. You will select a complex, real-world problem and build a production-ready database solution with analytical capabilities through 8 clearly defined phases.

** Important:** This project serves as your **FINAL EXAM** and significantly impacts your course grade.

## CRITICAL RULES (NON-NEGOTIABLE)

1. **STRICT DEADLINES** → Late submissions = ZERO POINTS
2. **INDIVIDUAL WORK** → No copying; must be original
3. **GITHUB REQUIRED** → All code tracked via GitHub
4. **QUALITY FIRST** → Production-ready code expected
5. **SCREENSHOTS PROOF** → Project name must be visible

## 8-PHASE PROJECT STRUCTURE

| Phase | Objective | Key Deliverable |
|---|---|---|
| **I** | Problem identification | PowerPoint (1-5 slides) |
| **II** | Business process modeling | UML/BPMN diagram + 1-page explanation |
| **III** | Logical database design | ER diagram + data dictionary |
| **IV** | Database creation | Oracle PDB + configuration scripts |
| **V** | Table implementation | CREATE/INSERT scripts + data validation |
| **VI** | PL/SQL development | Procedures, functions, packages, cursors |
| **VII** | Advanced programming | Triggers, auditing, security rules |
| **VIII** | Final documentation | GitHub repo + BI + 10-slide presentation |

## PHASE I: Problem Statement & Presentation

**Deadline:** November 17, 2025

### Objective

Identify a real-world problem requiring a PL/SQL Oracle database solution with BI potential.

### Requirements

**Problem must address:**

- Problem Definition: What specific issue are you solving?
- Context: Where/how will the system be used?
- Target Users: Who benefits?
- Project Goals: Expected outcomes?
- BI Potential: How could analytics enhance decision-making?

**Presentation Format:**

- PowerPoint with 1-5 slides (Helvetica font)
- Include diagrams if applicable
- Bullet points throughout
- Clear, professional layout

**File Naming:**

```
GrpName_StudentId_FirstName_ProjectName_DB.pptx
Example: mon_12121_eric_malariaMS_db.pptx
```

**Submit:** Topic selection form (link provided by lecturer)

---

## PHASE II: Business Process Modeling

**Objective:** Model business process relevant to MIS.

### Requirements

**Define Scope:**

- Outline business process clearly
- Ensure MIS relevance (order processing, inventory, HR, healthcare, etc.)
- Document objectives and outcomes

**Identify Key Entities:**

- Users (employees, customers, patients, etc.)
- Departments/Systems
- Data sources
- Explain roles and responsibilities

**Use Swimlanes:**

- Separate actors/departments
- Show responsibility allocation
- Clear handoff points

**Apply UML/BPMN Notations:**

- Proper symbols and conventions
- Start/end points marked
- Decision points with branches
- Data flows highlighted

**Ensure Logical Flow:**

- Start to end progression
- Decision points highlighted
- Dependencies correctly structured

**Documentation:**

- One-page explanation maximum
- Main components described
- MIS functions explained
- Organizational impact justified
- Analytics opportunities identified

**Tools:** Lucidchart (preferred), canva or draw.io

**Deliverable:** GitHub submission (diagram + explanation)

---

# PHASE III: Logical Model Design

**Objective:** Design detailed logical data model (3NF minimum).

## Requirements

**Entity-Relationship Model:**

- Identify all entities
- Specify attributes with data types
- Mark Primary Keys (PKs) and Foreign Keys (FKs)
- Clearly define cardinalities
- Document all constraints

**Normalization:**

- Eliminate repeating groups (1NF)
- Eliminate partial dependencies (2NF)
- Eliminate transitive dependencies (3NF)
- Justify normalization approach

**Data Dictionary:**

- Table names, columns, data types
- Constraints documented
- Assumptions listed

**BI Considerations:**

- Identify fact vs. dimension tables
- Plan for slowly changing dimensions
- Consider aggregation levels
- Design audit trails

**Deliverable:** GitHub submission (ER diagram + data dictionary + assumptions)

---

## PHASE IV: Database Creation

**Objective:** Create and configure Oracle pluggable database.

### Requirements

**Database Setup:**

```
Naming Format:
GrpName_StudentId_FirstName_ProjectName_DB
Example: mon_12121_eric_malariaMS_db

Admin User:
- Username: [Identifiable]
- Password: [Your first name]
- Privileges: Super admin
```

**Configuration:**

- Tablespaces for data and indexes
- Temporary tablespace
- Memory parameters (SGA, PGA)
- Archive logging enabled
- Autoextend parameters set

**GitHub Documentation:**

- Database creation scripts
- Tablespace configuration
- User setup documentation
- Project structure overview

**Deliverable:** GitHub submission (creation scripts + README + project overview)

---

## PHASE V: Table Implementation & Data Insertion

**Objective:** Build physical database structure with realistic test data.

### Requirements

**Table Creation:**

- All entities converted to tables
- Oracle data types used correctly
- PKs and FKs enforced
- Indexes created appropriately
- Constraints set (NOT NULL, UNIQUE, CHECK, DEFAULT)

**Data Insertion:**

- 100-500+ realistic rows per main table
- Represents actual use cases
- Includes edge cases and nulls
- Mix of demographic distributions
- Validates against business rules

**Data Integrity Verification:**

- SELECT queries verify data
- Constraints enforced properly
- Foreign key relationships tested
- Data completeness checked

**Testing Queries:**

- Basic retrieval (SELECT *)
- Joins (multi-table queries)
- Aggregations (GROUP BY)
- Subqueries

**Deliverable:** GitHub submission (CREATE/INSERT scripts + validation queries + test results)

---

## PHASE VI: Database Interaction & Transactions

**Objective:** Develop PL/SQL procedures, functions, and packages.

### Requirements

**Procedures (Minimum 3-5):**

- Parameterized with IN/OUT/IN OUT
- DML operations (INSERT, UPDATE, DELETE)
- Exception handling
- Well-documented

**Functions (Minimum 3-5):**

- Calculation functions
- Validation functions
- Lookup functions
- Proper return types

**Cursors:**

- Explicit cursors for multi-row processing
- Proper OPEN/FETCH/CLOSE
- Bulk operations for optimization

**Window Functions:**

- ROW_NUMBER(), RANK(), DENSE_RANK()
- LAG(), LEAD()
- PARTITION BY, ORDER BY
- Aggregates with OVER clause

**Packages:**

- Package specification (public interface)
- Package body (implementation)
- Related procedures grouped together

**Exception Handling:**

- Predefined exceptions caught
- Custom exceptions defined
- Error logging implemented
- Recovery mechanisms in place

**Testing:**

- Each procedure/function tested
- Edge cases validated
- Performance verified
- Test results documented

**Deliverable:** GitHub submission (PL/SQL scripts + test results)

---

# PHASE VII: Advanced Programming & Auditing

**Objective:** Implement triggers, business rules, and comprehensive auditing.

## Restriction Rule: CRITICAL REQUIREMENT

**Business Rule:**

```
Employees CANNOT INSERT/UPDATE/DELETE:
- On WEEKDAYS (Monday-Friday)
- On PUBLIC HOLIDAYS (upcoming month only)
```

**Implementation Steps:**

1. Holiday Management:
2. Audit Log Table:
3. Audit Logging Function:
4. Restriction Check Function:
5. Simple Triggers:
6. Compound Trigger

**Testing Requirements:**

- □ Trigger blocks INSERT on weekday (DENIED)
- □ Trigger allows INSERT on weekend (ALLOWED)
- □ Trigger blocks INSERT on holiday (DENIED)
- □ Audit log captures all attempts
- □ Error messages are clear
- □ User info properly recorded

**Deliverable:** GitHub submission (trigger code + audit queries + test results)

---

# PHASE VIII: Documentation, BI & Presentation

**Objective:** Finalize all documentation, implement BI, and deliver presentation.

## Part A: GitHub Repository

**Organize professionally:**

```
your-project-repo/
├── README.md (project overview)
├── database/
│   ├── scripts/ (all SQL: CREATE, INSERT, procedures, triggers)
│   └── documentation/
├── queries/
│   ├── data_retrieval.sql
│   ├── analytics_queries.sql
│   └── audit_queries.sql
├── business_intelligence/
│   ├── bi_requirements.md
│   ├── dashboards.md
│   └── kpi_definitions.md
├── screenshots/
│   ├── oem_monitoring/
│   ├── database_objects/
│   └── test_results/
└── documentation/
    ├── data_dictionary.md
    ├── architecture.md
    └── design_decisions.md
```

**README.md Must Include:**

- Project title and overview
- Student name and ID
- Problem statement (2-3 sentences)
- Key objectives
- Quick start instructions
- Links to documentation

**Screenshots Must Show:**

- ER diagram with all tables
- Database structure (SQL Developer)
- Sample data (5-10 rows)
- Procedures/triggers in editor
- Test results and execution
- Audit log entries

**Code Documentation:**

- All DDL (CREATE TABLE statements)
- All DML (INSERT scripts)
- All procedures with parameters
- All functions with return types
- All triggers with logic explanation
- All packages (specification & body)
- Analytics queries (window functions, aggregations)

**Data Dictionary:**

```
| Table | Column | Type | Constraints | Purpose |
|-------|--------|------|-------------|---------|
| PATIENTS | PAT_ID | NUMBER(10) | PK, NOT NULL | Unique identifier |
| ... | ... | ... | ... | ... |
```

## Part B: Business Intelligence (optional - 2 marks )

**BI Requirements:**

- Define KPIs that matter
- Identify decision support needs
- Document stakeholders

- Specify reporting frequency

**Dashboard Mockups (minimum):**

- Executive Summary (KPI cards, trends)
- Audit Dashboard (violations, denials)
- Performance Dashboard (resource usage)

**Deliverable:** BI dashboards/mockups + analytical queries + KPI definitions

## Part C: PowerPoint Presentation

**10 Slides Maximum (15-20 minutes total):**

| Slide | Content | Time |
|---|---|---|
| 1 | Title (project, name, date, AUCA) | 30 sec |
| 2 | Problem (issue, context, users) | 1 min |
| 3 | Solution (objectives, scope, features) | 1 min |
| 4 | Database (ER diagram, entities) | 1 min |
| 5 | Business Process (swimlanes, workflow) | 1 min |
| 6 | Technical (technology, PL/SQL) | 1 min |
| 7 | Advanced Features (triggers, auditing, security) | 2 min |
| 8 | BI & Analytics (dashboards, KPIs, reports) | 2 min |
| 9 | Results (data volume, tests, screenshots) | 2 min |
| 10 | Conclusion (achievements, lessons, Q&A) | 1 min |

**Formatting:**

- Font: Arial/Helvetica/Calibri (12-14pt body, 24-32pt titles)
- Max 4-5 bullet points per slide
- Include diagrams, screenshots, charts
- Professional appearance
- NO MORE THAN 10 SLIDES

**File Name:**

```
GrpName_StudentId_FirstName_ProjectTitle_Presentation.pptx
Example: mon_12121_eric_malariaMS_presentation.pptx
```

**Submission:**

- Email to eric.maniraguha@auca.ac.rw
- Google Drive shared link

- Include GitHub repository link

---

**BONUS:** +2 points for exceptional complexity/innovation

---

## SUBMISSION CHECKLIST

Before submitting:

- □ Code is ORIGINAL (no plagiarism)
- □ All code TESTED (no errors)
- □ GitHub ORGANIZED (meaningful commits)
- □ SCREENSHOTS include project name
- □ PowerPoint has MAX 10 slides
- □ All DOCUMENTATION complete
- □ BI IMPLEMENTATION included
- □ Submitted BEFORE DEADLINE

---

## SUPPORT

**Lecturer:** Eric Maniraguha
**Email:** eric.maniraguha@auca.ac.rw

---

## ✨ FINAL MESSAGE

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

This capstone represents your database development mastery. Focus on quality, test thoroughly, and submit on time.

**Good luck!** 🎓

---