**🧩 Capstone Project: Multi‑Branch Healthcare Diagnostic System**

**Sprint 1: ERD & Schema Design (1 week)**

**✅ Tasks:**

* Analyze requirements and identify entities (e.g., Branch, Department, Patient, Visit, Doctor, Test, Test\_Package, Test\_Report, Technician, Bill, Inventory).
* Create **ER Diagram** with proper relationships:
  + 1:M (e.g., Branch → Departments, Department → Doctors)
  + M:N (e.g., Test\_Package ↔ Test via associative table)
  + Weak entity relationships (e.g., Visit → Test\_Report)
* Normalize schema to **3NF**:
  + Remove repeating groups
  + Ensure no partial or transitive dependencies
* **Deliverables**:
  + ER Diagram (UML/Crow’s Foot notation)
  + Schema creation scripts (CREATE TABLE with PK/FK, UNIQUE, NOT NULL, and CHECK constraints)
  + Documentation of normalization steps, assumptions, and business rules

**🛠️ Sprint 2: Data Model & PL/SQL Development\*\* (2 weeks)**

**✅ Tasks:**

* Populate schema with **test data** (via INSERT scripts or small CSV imports).
* Develop **PL/SQL packages and procedures**:
  + **Branch\_PKG**: Add/edit branch info
  + **Visit\_PKG**: Register patient visits
  + **Billing\_PKG**: Generate bills, apply discounts/taxes
  + **Report\_PKG**: Upload and fetch diagnostic test reports
  + **Inventory\_PKG**: Check stock, update item quantities
* Use %TYPE and %ROWTYPE for schema independence
* Introduce **exception handling** and **autonomous transactions** for audit logging
* **Deliverables**:
  + PL/SQL packages with CRUD operations
  + Exception-safe error logging mechanisms
  + Test scripts invoking procedures

**🚀 Sprint 3: Advanced Logic & Analytics (2 weeks)**

**✅ Tasks:**

* Develop **cursor-based reporting** and parameterized views:
  + Sales summary per branch/department
  + Inventory consumption trends
  + Patient visit frequency
  + Alert package: tests near expiry
* Implement **interactive PL/SQL screens** (console or APEX snippets)
* Design **final diagnostics report generation**
* Introduce **automated alerts** using autonomous transactions
* **Deliverables**:
  + Advanced queries, packages and reports
  + Documentation detailing logic + SQL examples
  + Presentation/demo of system functionality

**🎯 Capstone Outcome Summary**

| **Sprint** | **Outcome** |
| --- | --- |
| **Sprint 1** | Robust 3NF schema and ER Diagram |
| **Sprint 2** | Data population and core package logic |
| **Sprint 3** | Advanced queries, reporting, and alerts |

**📄 Additional Suggestions:**

* Write business rules clearly in each package
* Save all scripts in a structured Git repo (DDL/, DML/, PLSQL/)
* Provide a **Sprint Status Tracker** (functions, coverage, blockers)
* Prepare a **retrospective**: what worked, what could improve, lessons learned