
Due Date: May 04, 2025

Total Marks 160

Submission Instructions

1. Late assignments will not be accepted (marked zero).
2. Submit your soft copy on Google Classroom by 3rd May before 11:59pm.
3. There will be no credit if the given requirements are changed.
4. Only 2 members can form a group together and work should be equally divided.
5. Plagiarism will be strictly dealt.
8. Write a report for the assignment. Report must include a database schema including ERD and relational mapping. Report must also contain all SQL queries. Also attach output screenshot for each query.
9. Submit the report in pdf format, and the SQL file with the following naming convention.

Section-Roll#1-Roll#2. E.g. A-23i_1234-23i_5678.

CASE STUDY

In a modern hospital, maintaining data integrity, auditability, and automated workflows is crucial for managing patients, appointments, and medical prescriptions. This assignment focuses on building a SQL and PL/SQL-driven system that tracks sensitive patient-related operations and ensures transaction transparency via automated audit logs, triggers, and stored procedures.

You are tasked with designing and implementing core components of a simplified Hospital Management System with the following key requirements. A sample dataset is provided in the attached Excel file. Based on this data, complete the following tasks

A sample dataset is provided in the attached Excel file. Based on this data, complete the following tasks

Part#1 – Design (35 Marks)

1. Design SQL Schema for the given dataset. Identify the Entities/Tables and corresponding columns, constraints, and primary keys. Also, identify the relationships between different Entities and map them through foreign keys correctly. [15 Marks]
2. Using SQL queries perform following operations [20 Marks]
 - a. Create Database and Tables. Define Constraints, Primary Keys and Foreign Keys
 - b. Identify the relationships between different Entities and map them through foreign keys correctly.

[NOTE: Clearly Specify, any modification in the database tables performed due to successful implementation of the required triggers]

Part#2 – Insertions (25 Marks)

Download the provided dataset and load it into the created tables. You can use any utility for this purpose. Using individual insert queries for each row is not allowed (and possible) for the given dataset. You have to understand the type of data in a column while designing the schema. You can assume empty cells as NULL. Generate data for **Audit_Log** table while uploading your data using triggers.

[NOTE: Clearly Specify, any modification in the database tables performed due to successful implementation of the required triggers]

Part#3 – Queries & Triggers (100 Marks)

Write SQL Queries to Perform Following Operations

Creating Triggers**[100 Marks]****1. Insert Trigger on Patients Table [10 Marks]**

Create an AFTER INSERT trigger on the Patients table. When a new patient is registered, an entry should automatically be recorded in the Audit_Log table. The entry must include: patient_id, action_type = 'INSERT', table_name = 'PATIENTS', timestamp

2. Insert and Delete Triggers on Appointments Table [20 Marks]

Create two triggers on the Appointments table:

- INSERT Trigger: Logs when a new appointment is booked. Entry should include: appointment_id, patient_id, doctor_id, action_type = 'BOOK', and timestamp
- DELETE Trigger: Logs when an appointment is canceled. Entry should include: appointment_id, action_type = 'CANCEL', table_name = 'APPOINTMENTS', timestamp

Note: Ensure both actions are logged into the Audit_Log table.

3. Update Trigger on Doctors Table [10 Marks]

Create an AFTER UPDATE trigger on the Doctors table to monitor changes in availability. If the availability_status field is updated (e.g., from 'Available' to 'Unavailable'), log the following in Audit_Log: doctor_id, old_status and new_status , action_type = 'STATUS_UPDATE', timestamp and table_name = 'DOCTORS'

4. Conditional Trigger with Status Validation [10 Marks]

Create a trigger that prevents inserting an appointment if the doctor is unavailable or if the patient already has an active appointment on that day. Log any blocked attempt in the Audit_Log with action_type = 'REJECTED_INSERT'

5. Trigger for Automatic Prescription Issue [10 Marks]

Create a trigger that automatically generates a placeholder prescription entry (e.g., “Diagnosis Pending”) when an appointment is marked as 'Completed'. Log this operation in Audit_Log.

Database System**Assignment-3****6. Before/After Trigger Combo [10 Marks]**

Implement a BEFORE trigger to verify constraints (e.g., only one active appointment per patient at a time), and an AFTER trigger to record that appointment in Audit_Log once inserted. Include OLD and NEW values.

7. Cascading Triggers with Logic Conflict [30 Marks]

- Patients with more than 3 missed appointments should be flagged.
- Automatically mark a doctor as “Unavailable” after 3 missed appointments in a week.
- A doctor should not have more than 5 appointments per day.

Functional Testing Scenarios**1. Add a New Patient and a New Doctor**

- Insert a new patient and doctor record.
- Check Audit_Log for corresponding 'INSERT' entry.

2. Book an Appointment

- Book an appointment for the new patient with any doctor.
- Check Audit_Log for 'BOOK' action.

3. Cancel an Appointment

- Delete/cancel the appointment.
- Check Audit_Log for 'CANCEL' action.

4. Update Doctor Availability

- Change a doctor's availability_status.
- Confirm Audit_Log logs the change with both old and new values.

5. Issue a Prescription

- Insert a new prescription tied to a patient and appointment.
- Confirm a 'PRESCRIBE' entry is logged in Audit_Log.

Submission Instructions:

- You must populate the database with sufficient data to demonstrate the successful working of all implemented triggers.

Minimum requirements:

- At least 20 patients
 - At least 10 doctors
 - At least 20 different appointments covering various scenarios (e.g., completed, missed, canceled)
- Your report (PDF) must include screenshots for each trigger execution:
 - Insert Triggers: Demonstrate insertion into audit logs or related updates upon data addition.
 - Delete/Update Triggers: Show successful logging or relevant changes when data is updated or deleted.

Database System**Assignment-3**

- Highlight any flagging, automated calculations, or conditional logic triggered as a result of the operations.
- Ensure all data supports logical validation, such as:
 - Patients with multiple missed appointments being flagged.
 - Log entries being created on each significant operation.
 - Changes in availability or status being captured by audit triggers.

Submission Deliverables:**1. Code File (SQL & PL/SQL)**

Submit a single .sql file containing:

- Table creation commands with all constraints.
- Insert queries to populate the tables.
- All required triggers with logic implemented using PL/SQL (or equivalent for your DBMS).
- Any additional queries used for testing the logic

2. Excel File

Submit an .xlsx file containing:

- Sample data for each table: Patients, Doctors, Appointments, Audit_Log, etc.
- Data used for testing the triggers and other logic, including edge cases (e.g., multiple missed appointments).

3. Report (PDF Format)

- Your report must include the following:
- ERD Diagram showing entities and relationships.
- Relational Schema listing all tables with attributes, data types, primary/foreign keys.
- Schema Modifications (e.g., added columns for flagging, audit logging, or other enhancements).
- Screenshots of Trigger Testing
- Any additional observations or insights based on the test data.

Note:

1. In the sample database files, operations are performed by admin1 and admin 2. For **functional testing**, you need to create your member account, the account_id should be the last two digits of your roll number. "Performed_by" shows who enters the record. Both the group members must perform all operations. Marks will be deducted if you skip this part.