**DBMS PROJECT OVERVIEW**

**1)Title :**

**HOSTEL MANAGEMENT SYSTEM**

**2)Team Members:**

* **YASHIKA CHANANA PES1UG22CS714**
* **U SRIRAM PES2UG22CS635**

**3) Purpose :**

A **Hostel Management System** is a comprehensive software solution designed to streamline and automate the operations of a hostel, ensuring efficient room allocation, fee collection, and resident management. It simplifies tasks such as maintaining resident records, processing check-ins and check-outs, managing maintenance requests, and tracking inventory, while also enhancing communication between management and residents through a centralized platform. By automating repetitive processes, it reduces manual workload, improves resource utilization, and ensures compliance with safety and legal standards. Additionally, it provides detailed reports and analytics for better decision-making, ultimately improving operational efficiency and resident satisfaction.

**4) Scope of a Hostel Management System:**

1. **Resident and Room Management**
   * Efficient handling of resident data, including personal details, room assignments, and stay history, along with streamlined room allocation and occupancy tracking.
2. **Fee and Payment Management**
   * Automated calculation of fees, invoicing, and payment tracking with support for multiple payment modes and real-time financial reporting.
3. **Maintenance and Resource Management**
   * Tracking and managing maintenance requests, scheduling repairs, and overseeing inventory, including hostel furniture, appliances, and consumables.
4. **Security and Compliance**
   * Ensuring safety through visitor logging, access control, and compliance with legal requirements such as fire safety regulations and resident identification.
5. **Reporting and Analytics**
   * Generating detailed reports on occupancy, revenue, expenses, and operational efficiency, offering actionable insights for management and strategic planning.

**5) Description :**

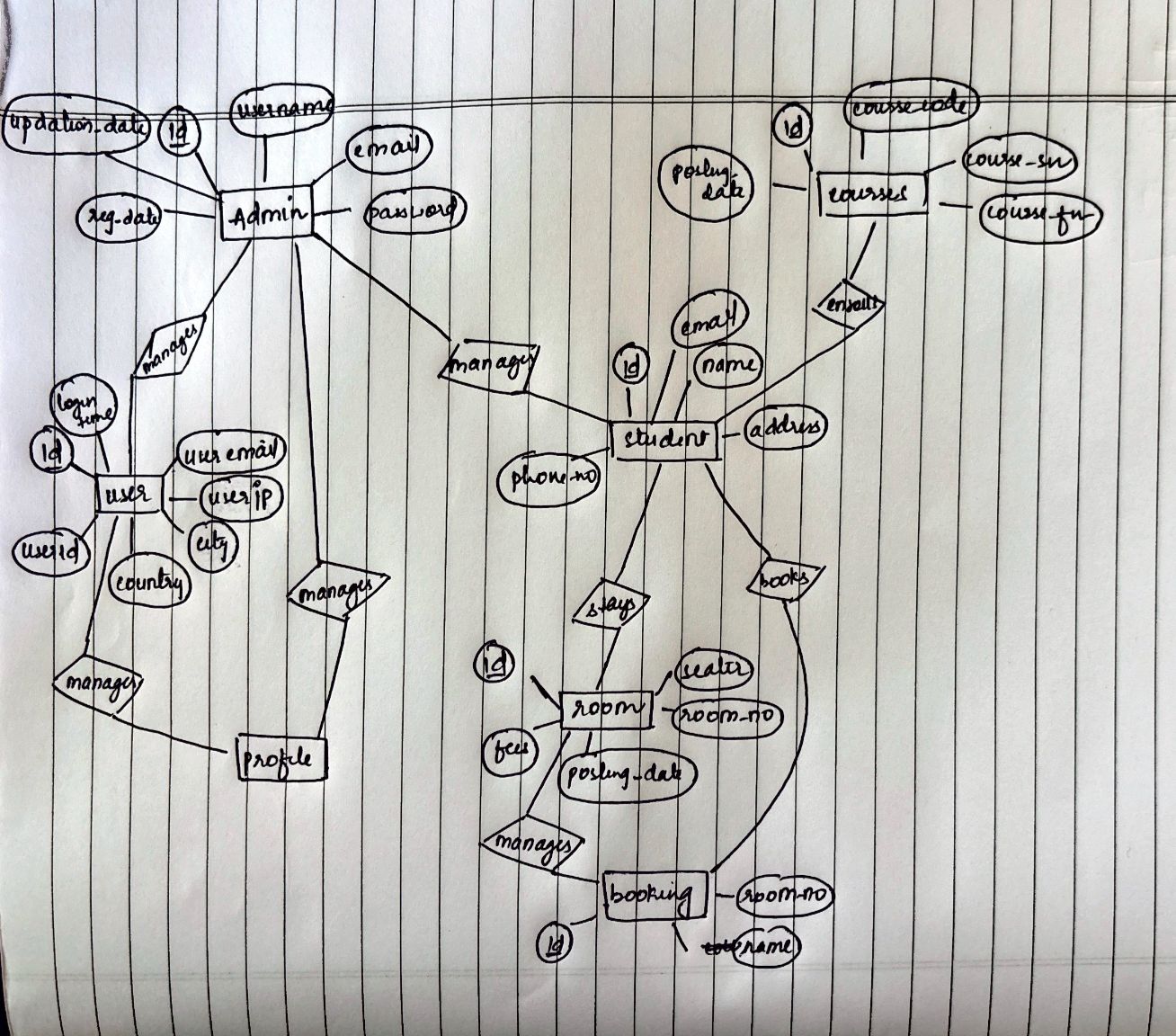
A **Hostel Management System Database Management System (DBMS)** is a structured framework that stores, organizes, and manages all the data related to hostel operations. It ensures efficient and secure handling of information, providing easy access, retrieval, and updates while maintaining data integrity. The DBMS is the backbone of the hostel management system, enabling smooth operation of functionalities such as room allocation, resident records, fee management, maintenance tracking, and inventory management.

**Key Features of a Hostel Management System DBMS:**

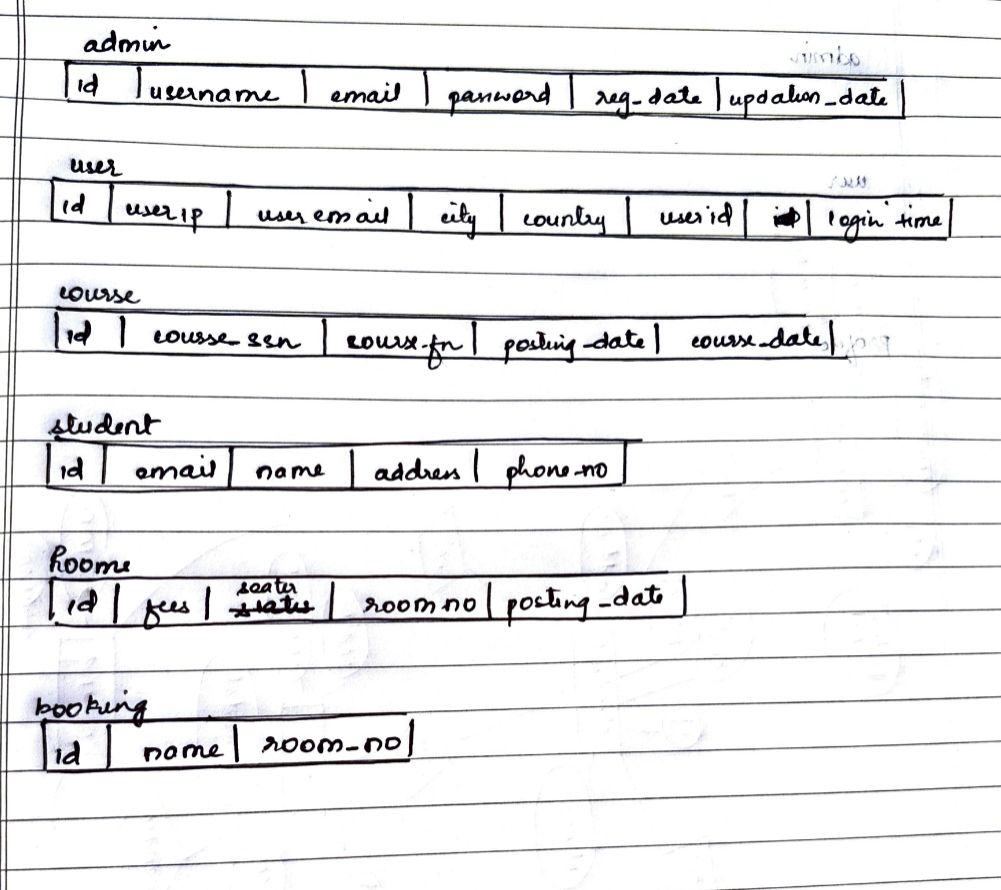
1. **Data Storage and Organization:**
   * Stores data in structured tables for residents, rooms, payments, maintenance requests, and staff.
   * Implements relationships between tables for seamless data connectivity (e.g., linking resident details to room assignments and payment records).
2. **Data Integrity and Security:**
   * Ensures the accuracy and consistency of data through constraints like primary keys, foreign keys, and validation rules.
   * Protects sensitive information with user authentication, role-based access control, and encryption.
3. **Automated Queries and Transactions:**
   * Supports SQL queries to fetch, update, or delete data as needed.
   * Handles complex transactions like fee payments, ensuring atomicity, consistency, isolation, and durability (ACID properties).
4. **Scalability and Flexibility:**
   * Capable of managing large volumes of data for hostels with numerous residents and rooms.
   * Easily adaptable to accommodate additional features, new hostels, or increased operational complexity.
5. **Backup and Recovery:**
   * Provides mechanisms for regular data backups to prevent loss.
   * Enables recovery of data in case of unexpected failures or corruption.
6. **Reporting and Analytics:**
   * Generates real-time reports on room occupancy, financial records, maintenance activities, and more.
   * Offers data insights for strategic decision-making.

By leveraging a robust DBMS, hostel management systems ensure efficient, secure, and scalable management of operations, enhancing overall functionality and user experience.

**6)ER Diagram :**

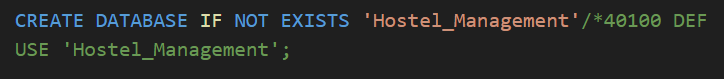


**7)Relation Schema :**

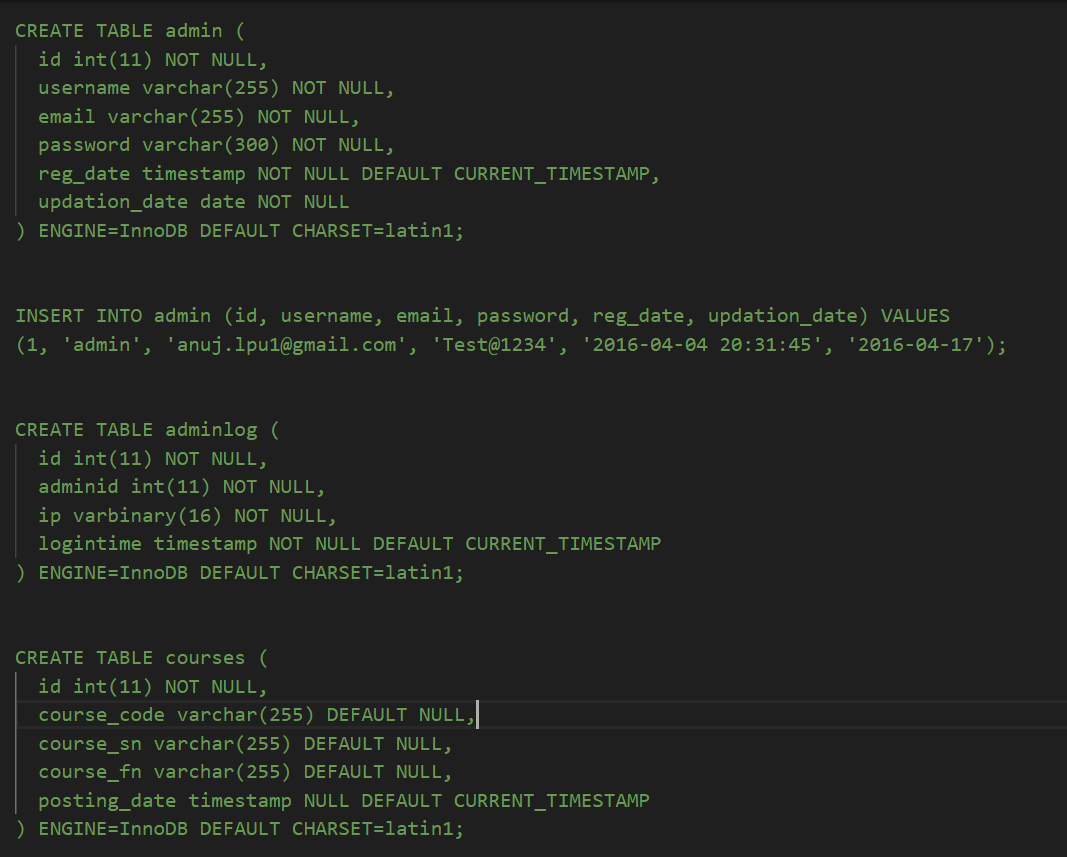


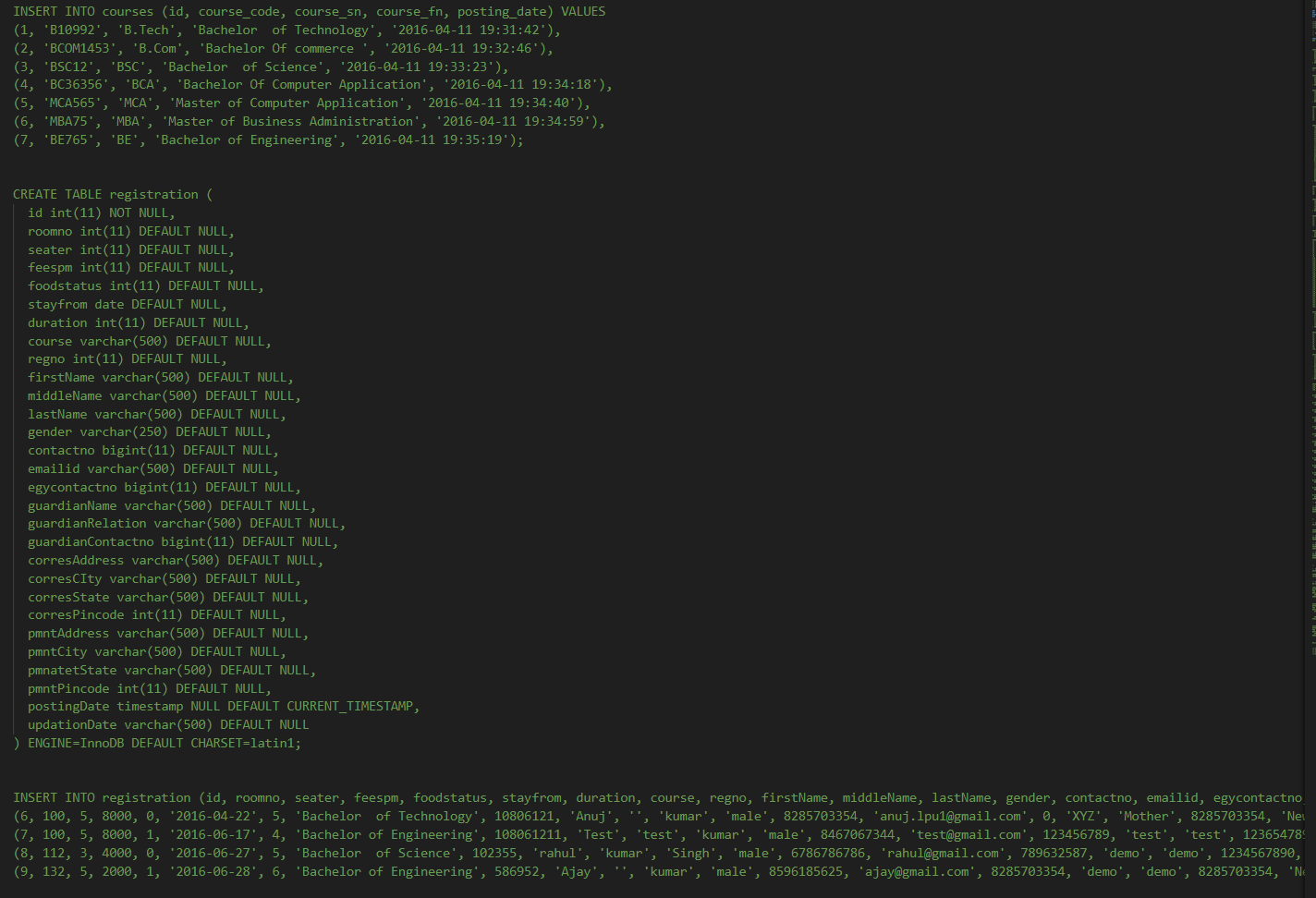
**8)Queries**

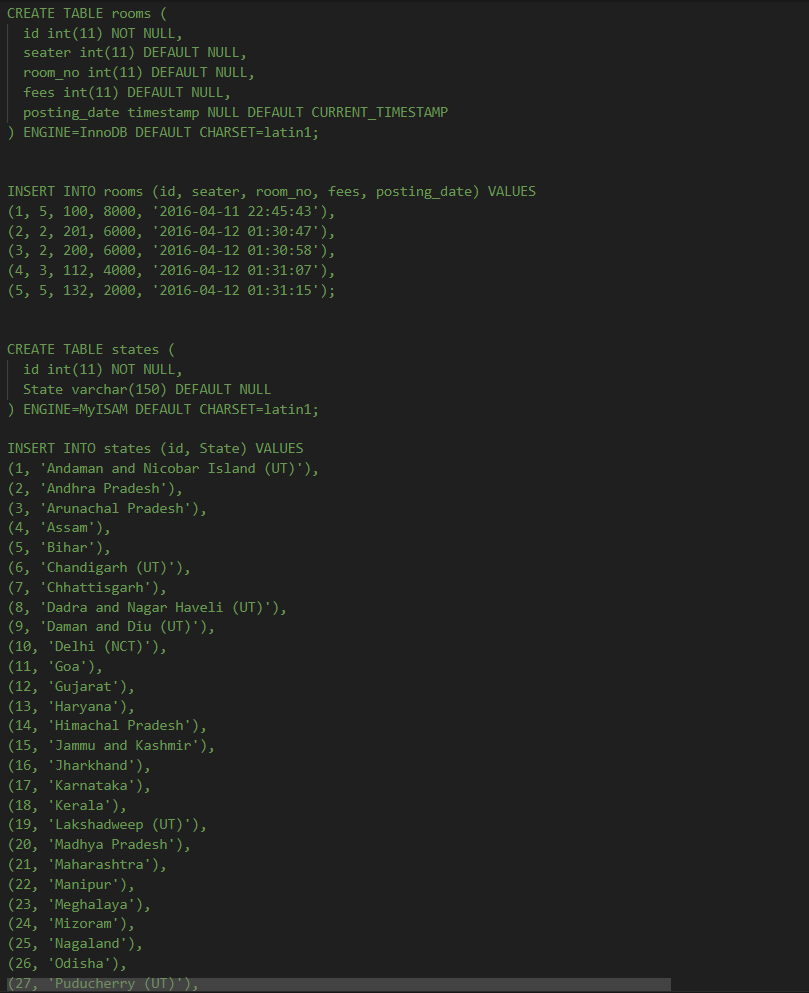
**Creating Database :**

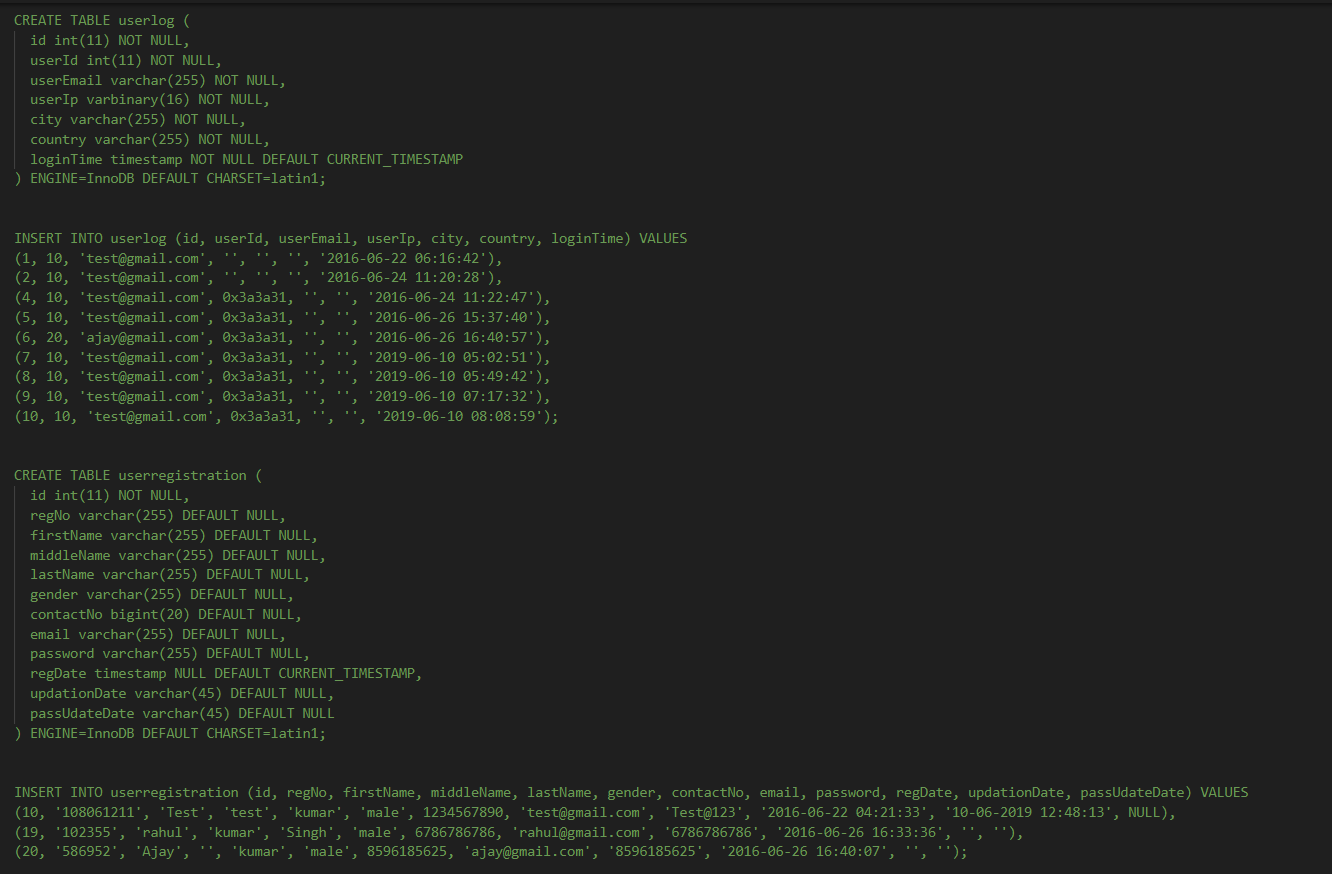
****

**Creating tables :**

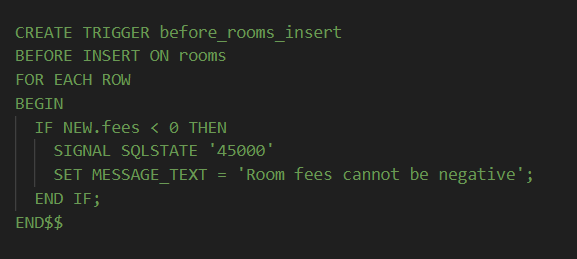
****

****

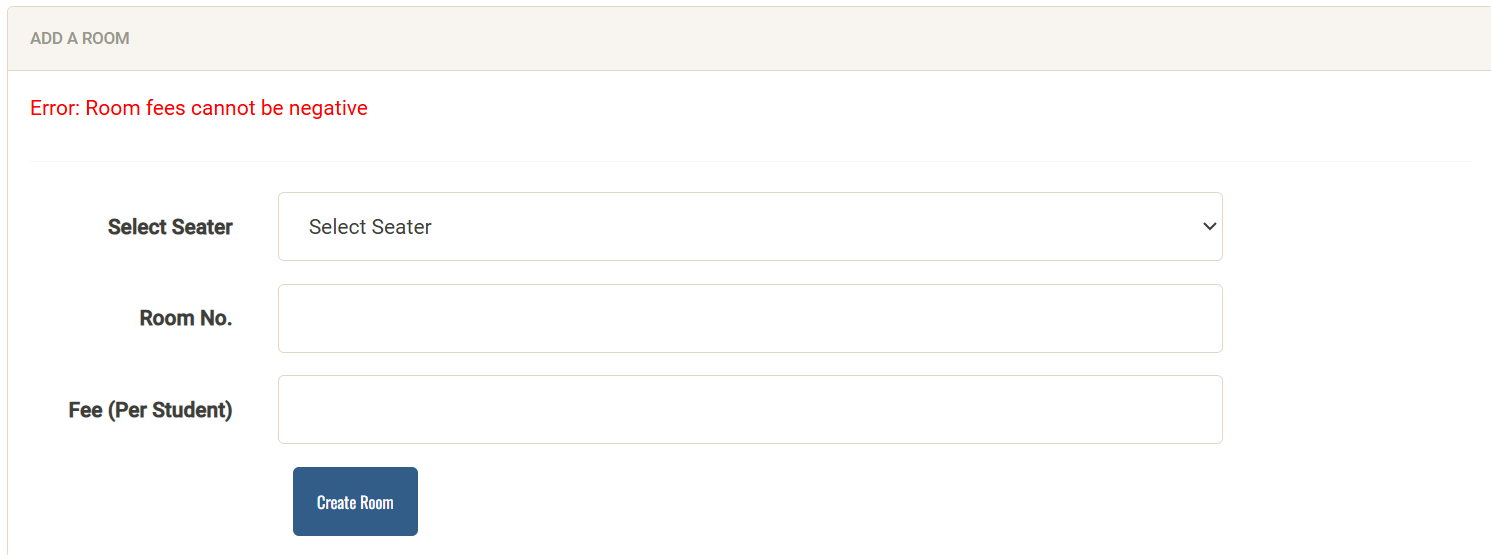
****

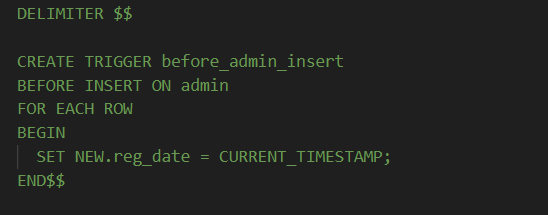
****

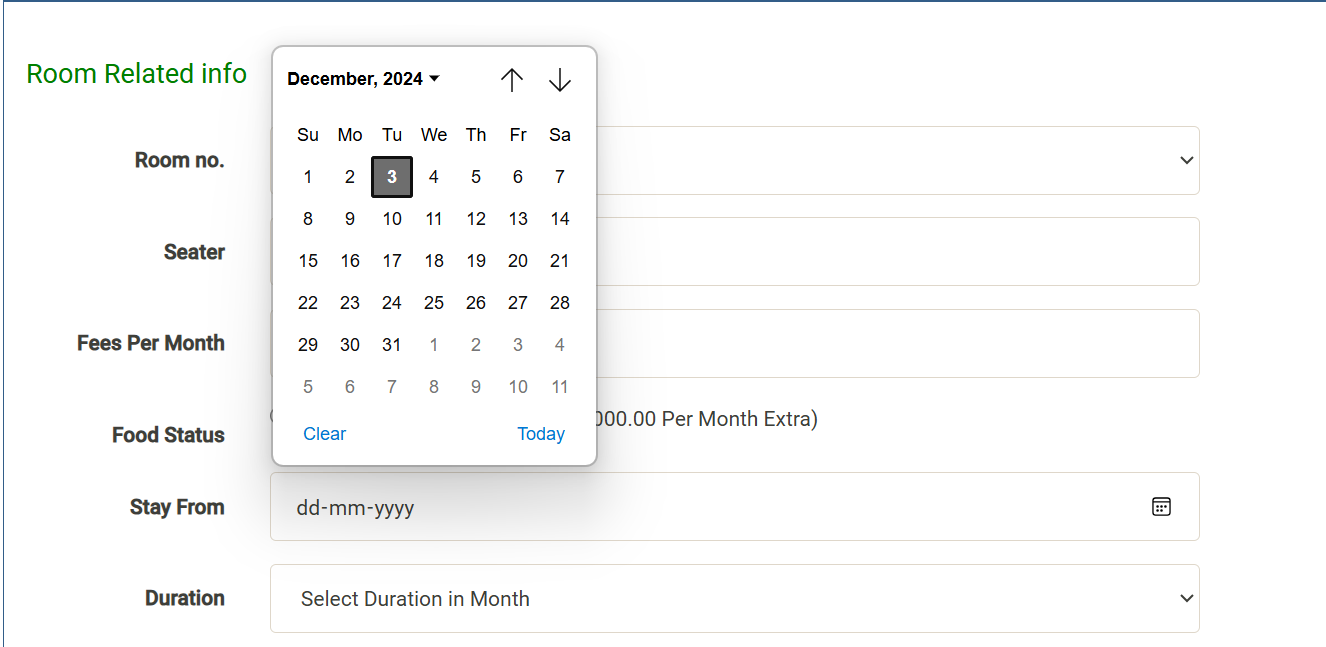
**Triggers:**

****

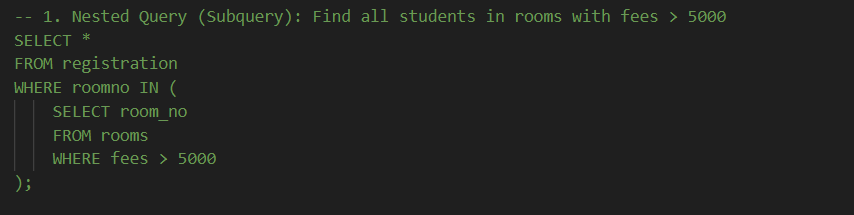
**Trigger Usage:**

****

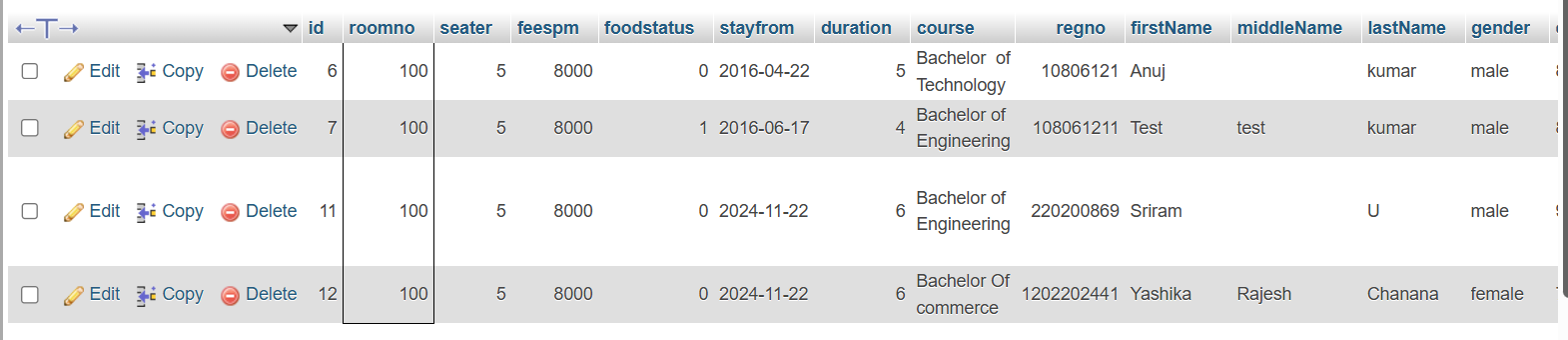
****

****

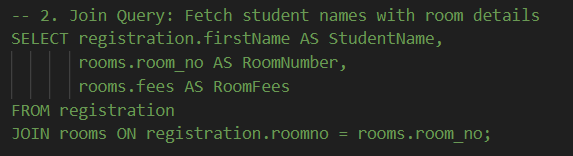
**Query-1:**

****

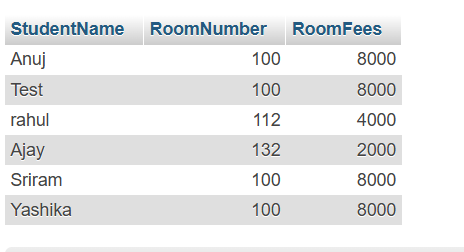
**Query-1 Usage:**

****

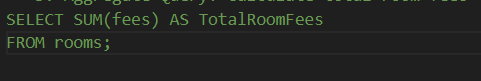
**Query-2:**

****

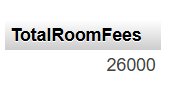
**Procedure-2 Usage:**

****

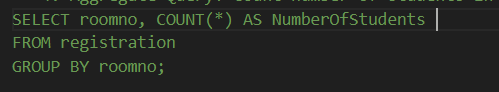
**Query-3:**

****

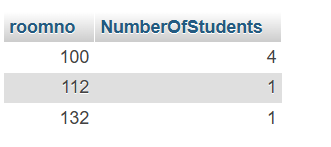
**Query-3 Usage:**

****

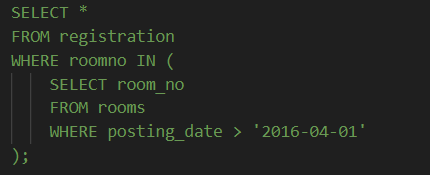
**Query-4:**

****

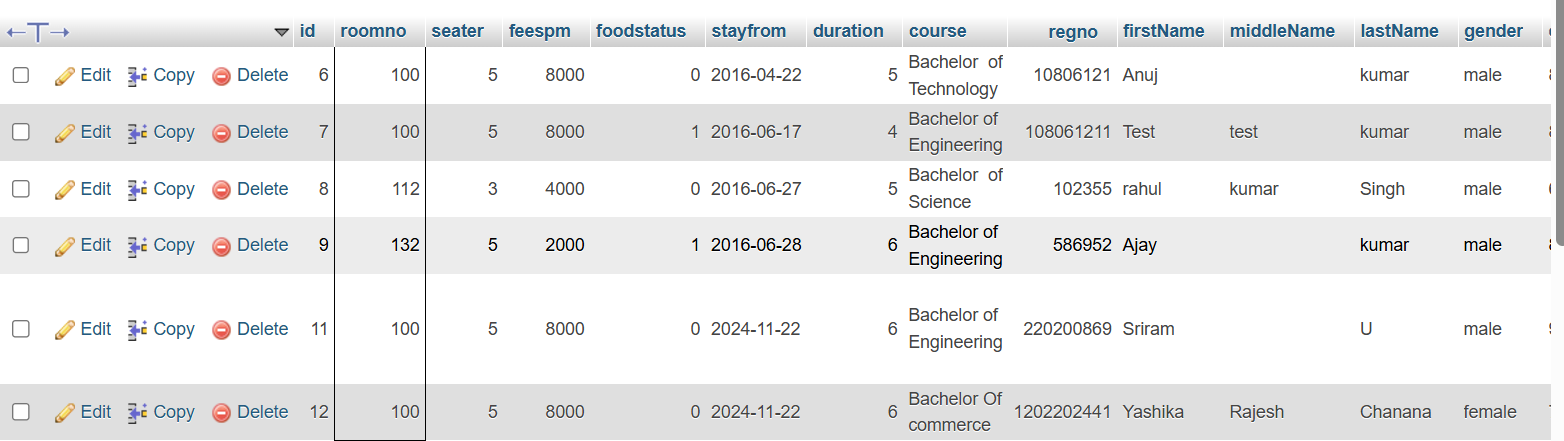
**Query-4 Usage:**

****

**Query 5:**

****

**Query 5 usage:**

****