SQL Project – Hotel Booking System

# Step 1: Define Requirements

Objective: To design and implement a database system to manage hotel rooms, customers, bookings, payments, staff schedules, and room services.  
  
Requirements:  
- Manage hotel rooms (type, price, availability)  
- Manage customers (details, bookings, payments)  
- Manage staff and housekeeping schedules  
- Manage room service requests  
- Generate reports (occupancy, revenue, workload)

# Step 2: Plan Tables

Tables:  
- Hotel\_Rooms(RoomID, RoomType, PricePerNight, Availability)  
- Customers(CustomerID, Name, Contact, Email)  
- Bookings(BookingID, CustomerID, RoomID, CheckInDate, CheckOutDate, Status)  
- Payments(PaymentID, BookingID, Amount, PaymentDate, PaymentMethod)  
- Staff(StaffID, Name, Role, ShiftTiming)  
- Housekeeping\_Schedule(ScheduleID, RoomID, StaffID, CleaningDate, Status)  
- Room\_Service(RequestID, RoomID, StaffID, ServiceDetails, RequestDate, Status)

# Step 3: Set Up Database

CREATE DATABASE HotelDB;  
USE HotelDB;

# Step 4: Create Tables

Example DDL:  
  
CREATE TABLE Hotel\_Rooms (  
 RoomID INT AUTO\_INCREMENT PRIMARY KEY,  
 RoomType VARCHAR(50) NOT NULL,  
 PricePerNight DECIMAL(10,2) NOT NULL,  
 Availability CHAR(1) DEFAULT 'Y'  
);  
  
-- Similar CREATE TABLE for Customers, Bookings, Payments, Staff, Housekeeping\_Schedule, Room\_Service

# Step 5: Insert Sample Data

INSERT INTO Hotel\_Rooms (RoomType, PricePerNight, Availability) VALUES  
('Single', 1500.00, 'Y'),  
('Double', 2500.00, 'Y'),  
('Suite', 5000.00, 'Y');  
  
-- Similarly insert 10 rows for each table

# Step 6: Queries

-- List available rooms  
SELECT \* FROM Hotel\_Rooms WHERE Availability = 'Y';  
  
-- Bookings of a specific customer  
SELECT b.BookingID, r.RoomType, b.CheckInDate, b.CheckOutDate, b.Status  
FROM Bookings b  
JOIN Hotel\_Rooms r ON b.RoomID = r.RoomID  
WHERE b.CustomerID = 2;  
  
-- Revenue report  
SELECT SUM(Amount) AS TotalRevenue FROM Payments;  
  
-- Count bookings by status  
SELECT Status, COUNT(\*) FROM Bookings GROUP BY Status;

# Step 7: Advanced Features

Views:  
CREATE VIEW available\_rooms\_view AS  
SELECT \* FROM Hotel\_Rooms WHERE Availability = 'Y';  
  
Stored Procedures:  
DELIMITER //  
CREATE PROCEDURE MakeBooking(IN p\_CustomerID INT, IN p\_RoomID INT, IN p\_CheckIn DATE, IN p\_CheckOut DATE)  
BEGIN  
 INSERT INTO Bookings (CustomerID, RoomID, CheckInDate, CheckOutDate, Status)  
 VALUES (p\_CustomerID, p\_RoomID, p\_CheckIn, p\_CheckOut, 'Booked');  
 UPDATE Hotel\_Rooms SET Availability = 'N' WHERE RoomID = p\_RoomID;  
END //  
DELIMITER ;  
  
Triggers:  
DELIMITER //  
CREATE TRIGGER AfterBookingInsert  
AFTER INSERT ON Bookings  
FOR EACH ROW  
BEGIN  
 UPDATE Hotel\_Rooms SET Availability = 'N' WHERE RoomID = NEW.RoomID;  
END //  
DELIMITER ;

**Worked SQL text file:**

[**https://github.com/vigneshprakash817/SQL/blob/main/Hotel\_DB.sql**](https://github.com/vigneshprakash817/SQL/blob/main/Hotel_DB.sql)

# Step 8: Reports

- Occupancy rate report  
- Revenue report by month  
- Pending housekeeping tasks  
- Room service requests  
- Staff workload report

# Step 9: Testing

- Insert a booking → room availability should change to 'N'  
- Cancel a booking → room availability should change to 'Y'  
- Insert payment → date should auto-fill  
- Run reports to verify outputs

# Step 10: Documentation

Include:  
- Project description & objectives  
- Table structures with relationships  
- SQL scripts (DDL, DML, queries)  
- Stored procedures, triggers, views  
- Sample outputs (Attached SQL text file)