



**Eastern
Mediterranean
University**

CMPE353/CMSE351

Database Management Systems

Labs 2 – 7

Design of a Hotel Room Booking System

Fall 2025-2026

Task Overview

You are asked to design a relational database according to the given schema below using Oracle Live SQL software. The task has several subtasks, and each is described below. Each task will be carried out during weekly lab hours (Labs 2-6). Therefore, each subtask must be completed weekly by the set lab dates.

Problem Statement

The objective is to design a database schema for a **Hotel Room Booking System**. The schema below is designed to manage data for a hotel reservation platform. It includes information about guests, rooms, reservations, payments, room amenities, and reservation services.

Each guest has contact information stored in the system, including address and phone number, which is used for contact and emergency purposes. The platform stores essential details to facilitate communication and service delivery. Payments are recorded for each reservation, including information about the payment method and status. Each room can have multiple amenities, and each reservation can include multiple additional services. This structure supports the hotel's operations in managing guest interactions, room inventory, amenities, reservation processing, service orders, and payment handling.

In this system:

There are **10 tables** in total: "**guests**," "**rooms**," "**amenities**," "**room_amenities**," "**reservations**," "**reservation_rooms**," "**services**," "**reservation_services**," "**payments**," and "**reviews**."

- Guest information, including their unique ID, name, email, address, phone number, and ID number, is stored in the "**guests**" table.
- Information about rooms, such as their unique ID, room number, room type, price per night, and capacity, is stored in the "**rooms**" table.
- Information about available amenities, such as amenity ID, name, and description, is stored in the "**amenities**" table.
- The relationship between rooms and their amenities is managed through the "**room_amenities**" table, which tracks which amenities are available in which rooms.
- Reservation records, including details such as guest ID, check-in date, check-out date, total amount, status ('Confirmed', 'CheckedIn', 'CheckedOut', 'Cancelled'), and description, are stored in the "**reservations**" table. The description field allows hotel staff to add notes before, during, and after the guest's stay (e.g., special requests, issues, observations).
- Reservation room assignments, including details such as their unique ID, reservation ID, room ID, and number of nights, are stored in the "**reservation_rooms**" table.
- Information about available services, such as service ID, service name, description, and price, is stored in the "**services**" table (e.g., "SPA A-class," "SPA B-class," "Airport Transfer," "Room Service").
- The relationship between reservations and ordered services is managed through the "**reservation_services**" table, which tracks which services were ordered for which reservations, along with the quantity.
- Payment information, including details such as payment ID, reservation ID, payment method (e.g., "CreditCard," "Cash," "BankTransfer"), status (e.g., 'Completed', 'Pending', 'Refunded'), and payment date, is stored in the "**payments**" table, linked to specific reservations.
- Guest reviews, including review ID, guest ID, reservation ID, rating (1-5), comment, and review date, are stored in the "**reviews**" table. Each review is linked to a specific guest and their reservation.

- Each reservation can have multiple rooms, managed through the "**reservation_rooms**" table, which tracks the relationship between rooms and reservations.
- Each room in a reservation can be booked for a different number of nights.
- Each room can have multiple amenities, managed through the "**room_amenities**" table.
- Each reservation can have multiple services, managed through the "**reservation_services**" table.
- Each payment entry corresponds to a unique reservation, ensuring that payment information is directly associated with the correct booking.

Database Schema

1. guests

- **guest_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **name** (VARCHAR2, Not Null)
- **email** (VARCHAR2, Unique, Not Null)
- **address** (VARCHAR2, Not Null)
- **phone_number** (VARCHAR2, Not Null)
- **id_number** (VARCHAR2, Not Null) - Government ID or passport number

2. rooms

- **room_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **room_number** (VARCHAR2, Unique, Not Null)
- **room_type** (VARCHAR2(20) DEFAULT 'Single' CHECK (room_type IN ('Single','Double','Suite','Deluxe')) NOT NULL)
- **price_per_night** (NUMBER(10, 2), Not Null)
- **capacity** (NUMBER, Not Null) - Maximum number of guests

3. amenities

- **amenity_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **amenity_name** (VARCHAR2, Not Null)
- **description** (VARCHAR2, Nullable) - Description can be optional

4. room_amenities

- **room_amenity_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **room_id** (Foreign Key, References rooms.room_id, Not Null)
- **amenity_id** (Foreign Key, References amenities.amenity_id, Not Null)

5. reservations

- **reservation_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **guest_id** (Foreign Key, References guests.guest_id, Not Null)
- **check_in_date** (DATE, Not Null)
- **check_out_date** (DATE, Nullable) - Can be null for open-ended or ongoing stays
- **total_amount** (NUMBER(10, 2), Nullable) - Can be null until final calculation
- **reservation_status** (VARCHAR2(20) DEFAULT 'Confirmed' CHECK (reservation_status IN ('Confirmed','CheckedIn','CheckedOut','Cancelled'))) NOT NULL)
- **description** (VARCHAR2, Nullable) - Notes about the reservation (special requests, issues, observations)

6. reservation_rooms

- **reservation_room_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **reservation_id** (Foreign Key, References reservations.reservation_id, Not Null)
- **room_id** (Foreign Key, References rooms.room_id, Not Null)
- **number_of_nights** (NUMBER, Not Null) - Duration of stay for this room

7. services

- **service_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **service_name** (VARCHAR2, Not Null)
- **description** (VARCHAR2, Not Null)
- **price** (NUMBER(10, 2), Not Null)

8. reservation_services

- **reservation_service_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **reservation_id** (Foreign Key, References reservations.reservation_id, Not Null)
- **service_id** (Foreign Key, References services.service_id, Not Null)
- **quantity** (NUMBER, Not Null, Default 1) - Number of times the service was ordered, defaults to 1

9. payments

- **payment_id** (Primary Key, NUMBER, GENERATED ALWAYS AS IDENTITY)
- **reservation_id** (Foreign Key, References reservations.reservation_id, Not Null)
- **payment_method** (VARCHAR2(20) CHECK (payment_method IN ('CreditCard','Cash','BankTransfer'))) NOT NULL)
- **payment_status** (VARCHAR2(20) DEFAULT 'Pending' CHECK (payment_status IN ('Completed','Pending','Refunded'))) NOT NULL)
- **payment_date** (DATE, Not Null, Default SYSDATE)

10. reviews

LAB 2: Database Creation and Setup

Tasks:

- Build the tables in Oracle Live SQL with your account according to the schema
- Build the relations among the tables
- Delete, modify or update the tables
- Delete, modify or update the constraints
- Insert sample data into tables

LAB 3: Data Population

Tasks:

- Insert sample data into the tables (**AT LEAST 20 RECORDS FOR EACH TABLE**)
- Modify, update, or delete entries

LAB 4, 5: SQL Queries

Complete the following queries:

1. Retrieve the total number of guests in the system.
2. List the room numbers and prices of "Suite" type rooms that are priced above \$200 per night.
3. What are the details of the reservation for the guest "John Smith"?
4. How many reservations include the room with room number "101"?
5. Retrieve the status of the reservation made with check-in date '2025-10-05'.
6. Find all reservations with a total amount greater than \$1,000.
7. Retrieve the most expensive "Deluxe" room.
8. Retrieve the reservation ID and the total price of reservations that contain more than one room.
9. List the room IDs, room numbers, room types, and prices for all rooms included in the reservation with the reservation ID 2.
10. Find the guest who placed the highest number of reservations.
11. List all rooms that have the amenity "WiFi".
12. Find all services ordered for reservation ID 5 along with their quantities and prices.
13. List all rooms that have not been reserved.
14. Find the most reserved room.
15. Calculate the total revenue generated from all "SPA" services (services with "SPA" in their name).

16. List all reservations that included both room service and airport transfer services.
 17. Find guests who have made multiple reservations.
 18. Insert a new guest into the "guests" table and a new reservation with multiple services into the "reservation_services" table using a PL/SQL block.
-

LAB 6: Triggers

Task:

Create a trigger that automatically updates the status of a reservation to 'CheckedIn' when the payment status is updated to 'Completed' in the payments table. After creating the trigger, test it by updating a payment status to 'Completed' and verify if the corresponding reservation's status has been updated.

LAB 7: Final Assessment

Quiz and presentation of all works done throughout the semester.

Lab Grading Policy

1. **Lab Attendance: 5%** of the total grade will be based on attendance at lab sessions [2-6]. (1 point each)
 2. **Assignment Submission: 10%**
Students must submit their completed work within two days following the lab session. The submission should include all assigned tasks. Throughout the labs (2-6), there will be a total of five assignments, each worth 2 points. These assignments will be submitted on Teams. Late submissions will lose points, so be sure to submit your work on time.
 3. **Lab Final Exam: 5%** of the total grade will be attributed to the lab's final exam.
-

Important Note

There will be **no make-up sessions** for missed labs or the lab final exam. Please ensure you attend and submit your work on time.