OOPS MINI PROJECT HOTEL MANAGEMENT SYSTEM

# AIM:

To construct a database for the Hotel management system and connect it with my SQL using java.

# Alogrithm:

**Start**

1. **Connect to Database**:
   * Establish a connection to the SQL database using JDBC with appropriate credentials.
2. **Display Menu**:
   * Show options:
     + Add Customer
     + Add Room
     + Make Reservation
     + Check-in
     + Make Payment
3. **Get User Input**:
   * Based on user choice, gather required input for the operation (e.g., customer details, room details, etc.).
4. **Execute Corresponding Operation**:
   * Perform the chosen operation (insert data into respective tables: Customers, Rooms, Reservations, CheckIns, Payments).
5. **Close Database Connection**:
   * After the operation, close the database connection.

**Stop**

# SQL QUERIES:

## Create the database

CREATE DATABASE HotelManagement;

## Use the created database

USE HotelManagement;

## Create the Customers table

CREATE TABLE Customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY, first\_name VARCHAR(50),

last\_name VARCHAR(50), email VARCHAR(100), phone VARCHAR(15),

address VARCHAR(255)

);

## Create the Rooms table

CREATE TABLE Rooms (

room\_id INT AUTO\_INCREMENT PRIMARY KEY,

room\_number VARCHAR(10), room\_type VARCHAR(50), price DECIMAL(10, 2),

status ENUM('Available', 'Occupied', 'Maintenance')

);

## Create the Reservations table

CREATE TABLE Reservations (

reservation\_id INT AUTO\_INCREMENT PRIMARY KEY, customer\_id INT,

room\_id INT, check\_in\_date DATE, check\_out\_date DATE,

status ENUM('Booked', 'Checked-in', 'Cancelled'),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id), FOREIGN KEY (room\_id) REFERENCES Rooms(room\_id)

);

## Create the Check-ins table

CREATE TABLE CheckIns (

check\_in\_id INT AUTO\_INCREMENT PRIMARY KEY, reservation\_id INT,

actual\_check\_in\_date DATE,

FOREIGN KEY (reservation\_id) REFERENCES Reservations(reservation\_id)

);

CREATE TABLE Payments (

payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

reservation\_id INT, payment\_date DATE, amount DECIMAL(10, 2),

payment\_method VARCHAR(50),

FOREIGN KEY (reservation\_id) REFERENCES Reservations(reservation\_id)

);

# JAVA PROGRAM :

import java.sql.\*; import java.util.Scanner;

public class HotelManagementSystem { public static void main(String[] args) {

Scanner sc = new Scanner(System.in); Connection conn = null;

// Database URL, username, and password

String dbURL = "jdbc:mysql://localhost:3306/HotelManagement"; String dbUsername = "root"; // Change as needed

String dbPassword = "password"; // Change as needed

try {

// Step 1: Connect to the Database

conn = DriverManager.getConnection(dbURL, dbUsername, dbPassword); System.out.println("Connected to the database!");

// Menu for User Input System.out.println("Enter 1 to Add Customer"); System.out.println("Enter 2 to Add Room");

System.out.println("Enter 3 to Make Reservation");

System.out.println("Enter 4 to Check-in"); System.out.println("Enter 5 to Make Payment");

int choice = sc.nextInt(); sc.nextLine(); // Consume newline

switch (choice) { case 1:

addCustomer(conn, sc); break;

case 2:

addRoom(conn, sc); break;

case 3:

makeReservation(conn, sc); break;

case 4:

checkIn(conn, sc); break;

case 5:

makePayment(conn, sc); break;

default:

System.out.println("Invalid option!");

}

} catch (SQLException e) {

System.out.println("Error connecting to database: " + e.getMessage());

} finally { try {

if (conn != null) conn.close();

} catch (SQLException e) {

System.out.println("Error closing connection: " + e.getMessage());

}

}

}

// Method to add a customer

public static void addCustomer(Connection conn, Scanner sc) { try {

System.out.println("Enter First Name:"); String firstName = sc.nextLine(); System.out.println("Enter Last Name:"); String lastName = sc.nextLine(); System.out.println("Enter Email:");

String email = sc.nextLine(); System.out.println("Enter Phone Number:"); String phone = sc.nextLine(); System.out.println("Enter Address:");

String address = sc.nextLine();

String sql = "INSERT INTO Customers (first\_name, last\_name, email, phone, address) VALUES (?,

?, ?, ?, ?)";

PreparedStatement pst = conn.prepareStatement(sql); pst.setString(1, firstName);

pst.setString(2, lastName); pst.setString(3, email); pst.setString(4, phone); pst.setString(5, address);

int rowsInserted = pst.executeUpdate(); if (rowsInserted > 0) {

System.out.println("Customer added successfully!");

}

} catch (SQLException e) {

System.out.println("Error adding customer: " + e.getMessage());

}

}

// Method to add a room

public static void addRoom(Connection conn, Scanner sc) { try {

System.out.println("Enter Room Number:"); String roomNumber = sc.nextLine(); System.out.println("Enter Room Type:"); String roomType = sc.nextLine(); System.out.println("Enter Price:");

double price = sc.nextDouble(); sc.nextLine(); // Consume newline

System.out.println("Enter Room Status (Available, Occupied, Maintenance):"); String status = sc.nextLine();

String sql = "INSERT INTO Rooms (room\_number, room\_type, price, status) VALUES (?, ?, ?, ?)"; PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, roomNumber); pst.setString(2, roomType); pst.setDouble(3, price); pst.setString(4, status);

int rowsInserted = pst.executeUpdate(); if (rowsInserted > 0) {

System.out.println("Room added successfully!");

}

} catch (SQLException e) {

System.out.println("Error adding room: " + e.getMessage());

}

}

// Method to make a reservation

public static void makeReservation(Connection conn, Scanner sc) { try {

System.out.println("Enter Customer ID:"); int customerId = sc.nextInt(); System.out.println("Enter Room ID:");

int roomId = sc.nextInt(); sc.nextLine(); // Consume newline

System.out.println("Enter Check-in Date (YYYY-MM-DD):"); String checkInDate = sc.nextLine(); System.out.println("Enter Check-out Date (YYYY-MM-DD):"); String checkOutDate = sc.nextLine();

String sql = "INSERT INTO Reservations (customer\_id, room\_id, check\_in\_date, check\_out\_date, status) VALUES (?, ?, ?, ?, 'Booked')";

PreparedStatement pst = conn.prepareStatement(sql); pst.setInt(1, customerId);

pst.setInt(2, roomId); pst.setString(3, checkInDate); pst.setString(4, checkOutDate);

int rowsInserted = pst.executeUpdate(); if (rowsInserted > 0) {

System.out.println("Reservation made successfully!");

}

} catch (SQLException e) {

System.out.println("Error making reservation: " + e.getMessage());

}

}

// Method to check in

public static void checkIn(Connection conn, Scanner sc) { try {

System.out.println("Enter Reservation ID:"); int reservationId = sc.nextInt();

sc.nextLine(); // Consume newline

System.out.println("Enter Actual Check-in Date (YYYY-MM-DD):"); String actualCheckInDate = sc.nextLine();

String sql = "INSERT INTO CheckIns (reservation\_id, actual\_check\_in\_date) VALUES (?, ?)"; PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, reservationId); pst.setString(2, actualCheckInDate);

int rowsInserted = pst.executeUpdate(); if (rowsInserted > 0) {

System.out.println("Check-in completed successfully!");

}

} catch (SQLException e) {

System.out.println("Error during check-in: " + e.getMessage());

}

}

// Method to make a payment

public static void makePayment(Connection conn, Scanner sc) { try {

System.out.println("Enter Reservation ID:"); int reservationId = sc.nextInt(); sc.nextLine(); // Consume newline

System.out.println("Enter Payment Date (YYYY-MM-DD):"); String paymentDate = sc.nextLine(); System.out.println("Enter Amount:");

double amount = sc.nextDouble(); sc.nextLine(); // Consume newline System.out.println("Enter Payment Method:"); String paymentMethod = sc.nextLine();

String sql = "INSERT INTO Payments (reservation\_id, payment\_date, amount, payment\_method) VALUES (?, ?, ?, ?)";

PreparedStatement pst = conn.prepareStatement(sql); pst.setInt(1, reservationId);

pst.setString(2, paymentDate); pst.setDouble(3, amount); pst.setString(4, paymentMethod);

int rowsInserted = pst.executeUpdate(); if (rowsInserted > 0) {

System.out.println("Payment processed successfully!");

}

} catch (SQLException e) {

System.out.println("Error making payment: " + e.getMessage());

}

}

# INPUT:

## Operation 1: Add Customer Input Prompt:

Choose an operation:

1. Add Customer
2. Add Room
3. Make Reservation
4. Check-in
5. Make Payment
6. Exit

## User Selects Option: 1

Enter First Name: John Enter Last Name: Doe

Enter Email: [john.doe@example.com](mailto:john.doe@example.com) Enter Phone Number: 123-456-7890

Enter Address: 123 Elm Street, Springfield, IL Customer added successfully!

## Operation 2: Add Room Input Prompt:

Choose an operation:

1. Add Customer
2. Add Room
3. Make Reservation
4. Check-in
5. Make Payment
6. Exit

## User Selects Option: 2

Enter Room Number: 101 Enter Room Type: Single Enter Price: 100.50

Enter Room Status: Available Room added successfully!

## Operation 3: Make Reservation Input Prompt:

Choose an operation:

1. Add Customer
2. Add Room
3. Make Reservation
4. Check-in
5. Make Payment
6. Exit

## User Selects Option: 3

Enter Customer ID: 1 Enter Room ID: 101

Enter Check-in Date (YYYY-MM-DD): 2024-11-25 Enter Check-out Date (YYYY-MM-DD): 2024-11-30 Reservation made successfully!

## Operation 4: Check-in Input Prompt:

Choose an operation:

1. Add Customer
2. Add Room
3. Make Reservation
4. Check-in
5. Make Payment
6. Exit

## User Selects Option: 4

Enter Reservation ID: 1

Enter Actual Check-in Date (YYYY-MM-DD): 2024-11-25 Check-in successful!

## Operation 5: Make Payment Input Prompt:

Choose an operation:

1. Add Customer
2. Add Room
3. Make Reservation
4. Check-in
5. Make Payment
6. Exit

## User Selects Option: 5

Enter Reservation ID: 1

Enter Payment Date (YYYY-MM-DD): 2024-11-25 Enter Amount: 500.00

Enter Payment Method: Credit Card Payment processed successfully!

## Exit Operation Input Prompt:

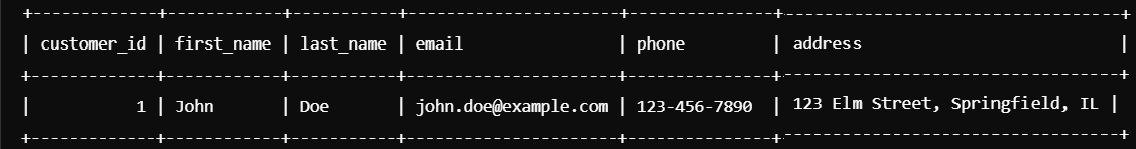
Choose an operation:

1. Add Customer
2. Add Room
3. Make Reservation
4. Check-in
5. Make Payment
6. Exit

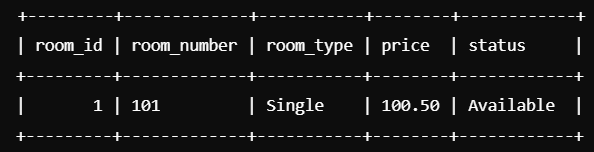
**User Selects Option**: 6 Exiting the system. Goodbye!

# OUTPUT:

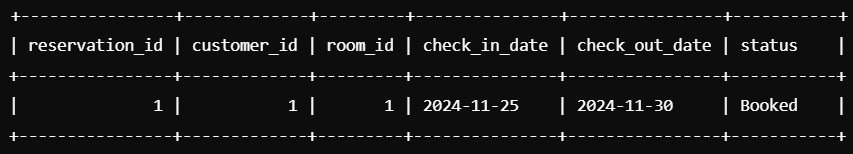
**Query 1: SELECT \* FROM Customers;**



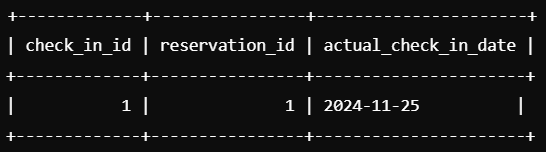
**Query 2: SELECT \* FROM Cases;**



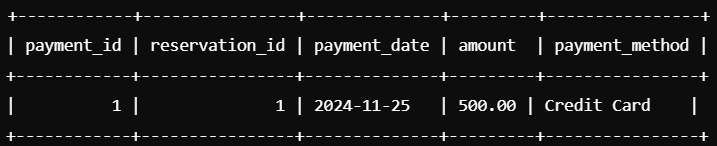
**Query 3: SELECT \* FROM Reservations;**



**Query 4: SELECT \* FROM CheckIns;**



**Query 5: SELECT \* FROM Payments;**



# RESULT:

The database construction for the Hotel management system has been successfully complected and connected with mySQL using java.

# MINI PROECT DONE BY:

S. RUPA DEVI