Report on the Hotel Management System (HMS)

Hotel Management System (HMS) - Database Report

This report outlines the structure, functionality, and key features of the Hotel Management System (HMS) database, which is designed to support efficient hotel operations through organized data handling and retrieval.

---

1. Database and Table Structure

The system operates on a database named **hms** and comprises multiple interrelated tables that manage different aspects of hotel operations:

• login

- Stores user credentials.
- **Fields:** username, password

• room

- Manages hotel room inventory.
- **Primary Key:** room_number
- **Fields:** room_number, availability, clean_status, price, bed_type

• customer

- Records guest booking information.
- **Primary Key:** id
- **Fields:** id, name, gender, country, status, deposit, room_number *(Foreign Key referencing room)*

• employee

- Stores details of hotel staff.
- **Fields:** name, age, job, salary, phone, aadhar, email

• driver

- Contains information about available drivers.
- **Fields:** name, age, company, brand, available

• department

- Lists various departments within the hotel and their budgets.
- **Fields:** department_name, budget

---

2. System Functionality and Queries

The HMS supports a wide range of SQL queries for operational, managerial, and analytical tasks.

Room Management

- Retrieve all room details or filter based on:
    - **Availability** (e.g., 'Available' or 'Occupied')
    - **Room number** or **price range**
    - **Clean status** (e.g., rooms needing cleaning)

Customer Management

- Display a list of all customers.
- Filter based on:
    - **Booking status** ('Checked-in' or 'Checked-out')
    - **Deposit amount**
- Retrieve details of a specific customer using **customer ID**.

Employee and Department Management

- Retrieve employees based on:
    - **Job title** (e.g., 'Housekeeper', 'Receptionist')
    - **Salary threshold**
- Count the total number of employees.
- List all departments along with their budgets.

Data Relationships

- **JOIN queries** are used to combine data from multiple tables.
    - For example: Display a customer's name, room number, bed type, and room price in a single view using a join between **customer** and **room** tables.

---

3. Data Manipulation Operations (DML)

The system enables standard data operations to update and maintain hotel records.

UPDATE

- Modify customer status to 'Checked-out'
- Update room **availability**
- Adjust employee **salary**

INSERT

- Add new entries such as:
  - New employee record
  - New room, customer, or department

DELETE

- Remove specific entries from the database:
  - Delete a room using **room_number**
  - Delete a customer using **id**

Conclusion

The **Hotel Management System (HMS)** database is a robust and structured platform designed to streamline hotel operations. It facilitates seamless data management for rooms, customers, employees, drivers, and departments through well-defined tables and efficient query handling. By supporting comprehensive DML operations and logical relationships among data entities, the system ensures accuracy, consistency, and flexibility in hotel administration.

1. Database Creation

```
CREATE DATABASE hms;
USE hms;
```

2. Creating Tables

a. Login Table

```
CREATE TABLE login (
    username VARCHAR(40),
    password VARCHAR(40)
);
INSERT INTO login VALUES ('admin', '12345');
```

b. Room Table

```
CREATE TABLE room (
    room_number VARCHAR(20) NOT NULL,
    availability VARCHAR(20) NOT NULL,
    clean_status VARCHAR(20) NOT NULL,
    price VARCHAR(20) NOT NULL,
    bed_type VARCHAR(30) NOT NULL,
    PRIMARY KEY (room_number)
);
INSERT INTO room VALUES
('101', 'Available', 'Clean', '2000', 'Single'),
('102', 'Available', 'Clean', '2500', 'Double'),
('103', 'Occupied', 'Clean', '3000', 'Single'),
('104', 'Available', 'Clean', '3500', 'Double'),
('105', 'Occupied', 'Dirty', '2200', 'Single'),
('106', 'Available', 'Clean', '2700', 'Single'),
('107', 'Occupied', 'Clean', '2300', 'Double'),
('108', 'Available', 'Dirty', '2000', 'Single'),
('109', 'Occupied', 'Clean', '4000', 'Double'),
('110', 'Available', 'Clean', '3000', 'Single'),
('111', 'Occupied', 'Clean', '2500', 'Single'),
('112', 'Available', 'Dirty', '3200', 'Double'),
('113', 'Available', 'Clean', '2100', 'Single'),
('114', 'Occupied', 'Dirty', '3500', 'Double'),
('115', 'Available', 'Clean', '2700', 'Single'),
('116', 'Occupied', 'Clean', '3000', 'Double'),
('117', 'Available', 'Dirty', '2600', 'Single'),
('118', 'Occupied', 'Clean', '2200', 'Double'),
('119', 'Available', 'Clean', '2800', 'Single'),
('120', 'Occupied', 'Dirty', '3300', 'Double');
```

c. Customer Table

```
CREATE TABLE customer (
    id VARCHAR(30) NOT NULL,
    number VARCHAR(30) NOT NULL,
    name VARCHAR(30) NOT NULL,
    gender VARCHAR(30) NOT NULL,
    country VARCHAR(30) NOT NULL,
    room_number VARCHAR(30) NOT NULL,
    status VARCHAR(30) NOT NULL,
    deposit VARCHAR(30) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY(room_number) REFERENCES room(room_number)
);
INSERT INTO customer VALUES
('C001', '9876543210', 'John Doe', 'Male', 'USA', '101', 'Checked-
in', '1000'),
('C002', '9876543211', 'Jane Doe', 'Female', 'Canada', '102',
'Checked-in', '1500'),
('C003', '9876543212', 'Robert Smith', 'Male', 'UK', '103',
'Checked-out', '500'),
('C004', '9876543213', 'Emily Davis', 'Female', 'Australia', '104',
'Checked-in', '2000'),
('C005', '9876543214', 'William Brown', 'Male', 'India', '105',
'Checked-out', '1000'),
('C006', '9876543215', 'Jessica Wilson', 'Female', 'USA', '106',
'Checked-in', '1200'),
('C007', '9876543216', 'Michael Taylor', 'Male', 'USA', '107',
'Checked-out', '800'),
('C008', '9876543217', 'Sarah Johnson', 'Female', 'Canada', '108',
'Checked-in', '1400'),
('C009', '9876543218', 'James Lee', 'Male', 'China', '109',
'Checked-in', '1600'),
('C010', '9876543219', 'Patricia Harris', 'Female', 'Mexico', '110',
'Checked-out', '600'),
('C011', '9876543220', 'Charles Clark', 'Male', 'India', '111',
'Checked-in', '1800'),
('C012', '9876543221', 'Linda Allen', 'Female', 'Australia', '112',
'Checked-out', '1500'),
('C013', '9876543222', 'David Hall', 'Male', 'USA', '113', 'Checked-
in', '900'),
('C014', '9876543223', 'Maria Scott', 'Female', 'Canada', '114',
'Checked-out', '700'),
('C015', '9876543224', 'Thomas Young', 'Male', 'UK', '115',
'Checked-in', '1100'),
('C016', '9876543225', 'Susan King', 'Female', 'USA', '116',
'Checked-in', '1300'),
('C017', '9876543226', 'Daniel Wright', 'Male', 'India', '117',
'Checked-out', '400'),
('C018', '9876543227', 'Karen Adams', 'Female', 'Mexico', '118',
'Checked-in', '1700'),
('C019', '9876543228', 'Steven Martinez', 'Male', 'USA', '119',
'Checked-in', '2000'),
('C020', '9876543229', 'Laura Perez', 'Female', 'Canada', '120',
'Checked-out', '1200');
```

d. Employee Table

```
CREATE TABLE employee (
    name VARCHAR(30) NOT NULL,
    age VARCHAR(10) NOT NULL,
    gender VARCHAR(30) NOT NULL,
    job VARCHAR(30) NOT NULL,
    salary VARCHAR(30) NOT NULL,
    phone VARCHAR(30) NOT NULL,
    aadhar VARCHAR(30) NOT NULL,
    email VARCHAR(40) NOT NULL
);
INSERT INTO employee VALUES
('John Miller', '30', 'Male', 'Manager', '50000', '9876543210',
'1234567890', 'john.miller@example.com'),
('Sara Parker', '28', 'Female', 'Receptionist', '30000',
'9876543211', '1234567891', 'sara.parker@example.com'),
('Tom Scott', '35', 'Male', 'Chef', '40000', '9876543212',
'1234567892', 'tom.scott@example.com'),
('Emily Taylor', '29', 'Female', 'Housekeeper', '25000',
'9876543213', '1234567893', 'emily.taylor@example.com'),
('Robert Brown', '45', 'Male', 'Security', '35000', '9876543214',
'1234567894', 'robert.brown@example.com'),
('Anna Wilson', '32', 'Female', 'Manager', '55000', '9876543215',
'1234567895', 'anna.wilson@example.com'),
('Michael Clark', '40', 'Male', 'Driver', '30000', '9876543216',
'1234567896', 'michael.clark@example.com'),
('Sarah Lewis', '38', 'Female', 'Housekeeper', '22000',
'9876543217', '1234567897', 'sarah.lewis@example.com'),
('David Walker', '50', 'Male', 'Security', '36000', '9876543218',
'1234567898', 'david.walker@example.com'),
('Jessica Harris', '27', 'Female', 'Receptionist', '28000',
'9876543219', '1234567899', 'jessica.harris@example.com');
```

e. Driver Table

```
CREATE TABLE driver (
    name VARCHAR(30) NOT NULL,
    age VARCHAR(10) NOT NULL,
    gender VARCHAR(20) NOT NULL,
    company VARCHAR(30) NOT NULL,
    brand VARCHAR(30) NOT NULL,
    available VARCHAR(10) NOT NULL,
    location VARCHAR(50) NOT NULL
);
INSERT INTO driver VALUES
('John Smith', '40', 'Male', 'ABC Transport', 'Toyota', 'Yes', 'New
York'),
('Amy Johnson', '28', 'Female', 'XYZ Cabs', 'Honda', 'No',
'Chicago'),
('David Green', '33', 'Male', 'Speed Cabs', 'Ford', 'Yes', 'San
Francisco'),
('Laura Adams', '36', 'Female', 'Sunshine Rides', 'Chevrolet',
'Yes', 'Los Angeles'),
('Paul Brown', '45', 'Male', 'Quick Transport', 'Mercedes', 'No',
'Houston'),
('Alice Clark', '38', 'Female', 'ABC Transport', 'BMW', 'Yes',
'Miami'),
('George White', '50', 'Male', 'XYZ Cabs', 'Audi', 'Yes', 'Dallas'),
('Emma Davis', '29', 'Female', 'Sunshine Rides', 'Nissan', 'No',
'Austin'),
('Robert Martinez', '42', 'Male', 'Speed Cabs', 'Toyota', 'Yes',
'Seattle'),
('Sophia Wilson', '31', 'Female', 'Quick Transport', 'Hyundai',
'No', 'Denver');
```

f. Department Table

```
CREATE TABLE department (
    department VARCHAR(30) NOT NULL,
    budget VARCHAR(30) NOT NULL
);
INSERT INTO department VALUES
('Housekeeping', '100000'),
('Front Desk', '150000'),
('Security', '120000'),
('Maintenance', '80000'),
('Restaurant', '200000'),
('Laundry', '50000'),
('Transportation', '70000'),
('Administration', '250000'),
('Marketing', '150000'),
('Events', '30000');
```

3.Queries

1. Get All Rooms Information

```
SELECT * FROM room;
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 101         | Available    | Clean        | 2000  | Single   |
| 102         | Available    | Clean        | 2500  | Double   |
| 103         | Occupied     | Clean        | 3000  | Single   |
| 104         | Available    | Clean        | 3500  | Double   |
| 105         | Occupied     | Dirty        | 2200  | Single   |
| 106         | Available    | Clean        | 2700  | Single   |
| 107         | Occupied     | Clean        | 2300  | Double   |
| 108         | Available    | Dirty        | 2000  | Single   |
| 109         | Occupied     | Clean        | 4000  | Double   |
| 110         | Available    | Clean        | 3000  | Single   |
| 111         | Occupied     | Clean        | 2500  | Single   |
| 112         | Available    | Dirty        | 3200  | Double   |
| 113         | Available    | Clean        | 2100  | Single   |
| 114         | Occupied     | Dirty        | 3500  | Double   |
| 115         | Available    | Clean        | 2700  | Single   |
| 116         | Occupied     | Clean        | 3000  | Double   |
| 117         | Available    | Dirty        | 2600  | Single   |
| 118         | Occupied     | Clean        | 2200  | Double   |
| 119         | Available    | Clean        | 2800  | Single   |
| 120         | Occupied     | Dirty        | 3300  | Double   |
+-------------+--------------+--------------+-------+----------+
```

2. Get Available Rooms

```
SELECT * FROM room WHERE availability = 'Available';
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 101         | Available    | Clean        | 2000  | Single   |
| 102         | Available    | Clean        | 2500  | Double   |
| 104         | Available    | Clean        | 3500  | Double   |
| 106         | Available    | Clean        | 2700  | Single   |
| 108         | Available    | Dirty        | 2000  | Single   |
| 110         | Available    | Clean        | 3000  | Single   |
| 112         | Available    | Dirty        | 3200  | Double   |
| 113         | Available    | Clean        | 2100  | Single   |
| 115         | Available    | Clean        | 2700  | Single   |
| 117         | Available    | Dirty        | 2600  | Single   |
| 119         | Available    | Clean        | 2800  | Single   |
+-------------+--------------+--------------+-------+----------+
```

3. Get Occupied Rooms

```
SELECT * FROM room WHERE availability = 'Occupied';
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 103         | Occupied     | Clean        | 3000  | Single   |
| 105         | Occupied     | Dirty        | 2200  | Single   |
| 107         | Occupied     | Clean        | 2300  | Double   |
| 109         | Occupied     | Clean        | 4000  | Double   |
| 111         | Occupied     | Clean        | 2500  | Single   |
| 114         | Occupied     | Dirty        | 3500  | Double   |
| 116         | Occupied     | Clean        | 3000  | Double   |
| 118         | Occupied     | Clean        | 2200  | Double   |
| 120         | Occupied     | Dirty        | 3300  | Double   |
+-------------+--------------+--------------+-------+----------+
```

4. Get Room Details by Room Number

```
SELECT * FROM room WHERE room_number = '101';
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 101         | Available    | Clean        | 2000  | Single   |
+-------------+--------------+--------------+-------+----------+
```

5. Get All Customers

```
SELECT * FROM customer;
+------+------------+----------------+--------+-----------+-------------+-------------+---------+
| id   | number     | name           | gender | country   | room_number | status      | deposit |
+------+------------+----------------+--------+-----------+-------------+-------------+---------+
| C001 | 9876543210 | John Doe       | Male   | USA       | 101         | Checked-in  | 1000    |
| C002 | 9876543211 | Jane Doe       | Female | Canada    | 102         | Checked-in  | 1500    |
| C003 | 9876543212 | Robert Smith   | Male   | UK        | 103         | Checked-out | 500     |
| C004 | 9876543213 | Emily Davis    | Female | Australia | 104         | Checked-in  | 2000    |
| C005 | 9876543214 | William Brown  | Male   | India     | 105         | Checked-out | 1000    |
| C006 | 9876543215 | Jessica Wilson  | Female | USA       | 106         | Checked-in  | 1200    |
| C007 | 9876543216 | Michael Taylor  | Male   | USA       | 107         | Checked-out | 800     |
| C008 | 9876543217 | Sarah Johnson   | Female | Canada    | 108         | Checked-in  | 1400    |
| C009 | 9876543218 | James Lee       | Male   | China     | 109         | Checked-in  | 1600    |
| C010 | 9876543219 | Patricia Harris | Female | Mexico    | 110         | Checked-out | 600     |
| C011 | 9876543220 | Charles Clark   | Male   | India     | 111         | Checked-in  | 1800    |
| C012 | 9876543221 | Linda Allen     | Female | Australia | 112         | Checked-out | 1500    |
| C013 | 9876543222 | David Hall      | Male   | USA       | 113         | Checked-in  | 900     |
| C014 | 9876543223 | Maria Scott     | Female | Canada    | 114         | Checked-out | 700     |
| C015 | 9876543224 | Thomas Young    | Male   | UK        | 115         | Checked-in  | 1100    |
| C016 | 9876543225 | Susan King      | Female | USA       | 116         | Checked-in  | 1300    |
| C017 | 9876543226 | Daniel Wright   | Male   | India     | 117         | Checked-out | 400     |
| C018 | 9876543227 | Karen Adams     | Female | Mexico    | 118         | Checked-in  | 1700    |
| C019 | 9876543228 | Steven Martinez | Male   | USA       | 119         | Checked-in  | 2000    |
| C020 | 9876543229 | Laura Perez     | Female | Canada    | 120         | Checked-out | 1200    |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
```

6. Get Customers with Status 'Checked-in'

```
SELECT * FROM customer WHERE status = 'Checked-in';
```

```
+------+------------+-----------------+--------+-----------+-------------+------------+---------+
| id   | number     | name            | gender | country   | room_number | status     | deposit |
+------+------------+-----------------+--------+-----------+-------------+------------+---------+
| C001 | 9876543210 | John Doe        | Male   | USA       | 101         | Checked-in | 1000    |
| C002 | 9876543211 | Jane Doe        | Female | Canada    | 102         | Checked-in | 1500    |
| C004 | 9876543213 | Emily Davis     | Female | Australia | 104         | Checked-in | 2000    |
| C006 | 9876543215 | Jessica Wilson  | Female | USA       | 106         | Checked-in | 1200    |
| C008 | 9876543217 | Sarah Johnson   | Female | Canada    | 108         | Checked-in | 1400    |
| C009 | 9876543218 | James Lee       | Male   | China     | 109         | Checked-in | 1600    |
| C011 | 9876543220 | Charles Clark   | Male   | India     | 111         | Checked-in | 1800    |
| C013 | 9876543222 | David Hall      | Male   | USA       | 113         | Checked-in | 900     |
| C015 | 9876543224 | Thomas Young    | Male   | UK        | 115         | Checked-in | 1100    |
| C016 | 9876543225 | Susan King      | Female | USA       | 116         | Checked-in | 1300    |
| C018 | 9876543227 | Karen Adams     | Female | Mexico    | 118         | Checked-in | 1700    |
| C019 | 9876543228 | Steven Martinez | Male   | USA       | 119         | Checked-in | 2000    |
+------+------------+-----------------+--------+-----------+-------------+------------+---------+
```

7. Get Customers with Deposit Greater Than 1000

```
SELECT * FROM customer WHERE deposit > 1000;
```

```
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
| id   | number     | name            | gender | country   | room_number | status      | deposit |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
| C002 | 9876543211 | Jane Doe        | Female | Canada    | 102         | Checked-in  | 1500    |
| C004 | 9876543213 | Emily Davis     | Female | Australia | 104         | Checked-in  | 2000    |
| C006 | 9876543215 | Jessica Wilson  | Female | USA       | 106         | Checked-in  | 1200    |
| C008 | 9876543217 | Sarah Johnson   | Female | Canada    | 108         | Checked-in  | 1400    |
| C009 | 9876543218 | James Lee       | Male   | China     | 109         | Checked-in  | 1600    |
| C011 | 9876543220 | Charles Clark   | Male   | India     | 111         | Checked-in  | 1800    |
| C012 | 9876543221 | Linda Allen     | Female | Australia | 112         | Checked-out | 1500    |
| C015 | 9876543224 | Thomas Young    | Male   | UK        | 115         | Checked-in  | 1100    |
| C016 | 9876543225 | Susan King      | Female | USA       | 116         | Checked-in  | 1300    |
| C018 | 9876543227 | Karen Adams     | Female | Mexico    | 118         | Checked-in  | 1700    |
| C019 | 9876543228 | Steven Martinez | Male   | USA       | 119         | Checked-in  | 2000    |
| C020 | 9876543229 | Laura Perez     | Female | Canada    | 120         | Checked-out | 1200    |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
```

8. Get Customer Information and Room Details

```
SELECT c.name, c.room_number, r.bed_type, r.price
FROM customer c
JOIN room r ON c.room_number = r.room_number;
```

```
+-----------------+-------------+----------+-------+
| name            | room_number | bed_type | price |
+-----------------+-------------+----------+-------+
| John Doe        | 101         | Single   | 2000  |
| Jane Doe        | 102         | Double   | 2500  |
| Robert Smith    | 103         | Single   | 3000  |
| Emily Davis     | 104         | Double   | 3500  |
| William Brown   | 105         | Single   | 2200  |
| Jessica Wilson   | 106          | Single    | 2700  |
| Michael Taylor   | 107          | Double    | 2300  |
| Sarah Johnson    | 108          | Single    | 2000  |
| James Lee        | 109          | Double    | 4000  |
| Patricia Harris  | 110          | Single    | 3000  |
| Charles Clark    | 111          | Single    | 2500  |
| Linda Allen      | 112          | Double    | 3200  |
| David Hall       | 113          | Single    | 2100  |
| Maria Scott      | 114          | Double    | 3500  |
| Thomas Young     | 115          | Single    | 2700  |
| Susan King       | 116          | Double    | 3000  |
| Daniel Wright    | 117          | Single    | 2600  |
| Karen Adams      | 118          | Double    | 2200  |
| Steven Martinez  | 119          | Single    | 2800  |
| Laura Perez      | 120          | Double    | 3300  |
+-----------------+-------------+----------+-------+
```

9. Get Customer Details by Customer ID

```
SELECT * FROM customer WHERE id = 'C001';
+------+------------+----------+--------+---------+-------------+------------+---------+
| id   | number     | name     | gender | country | room_number | status     | deposit |
+------+------------+----------+--------+---------+-------------+------------+---------+
| C001 | 9876543210 | John Doe | Male   | USA     | 101         | Checked-in | 1000    |
+------+------------+----------+--------+---------+-------------+------------+---------+
```

10. Get Employees Working in Housekeeping Department

```
SELECT * FROM employee WHERE job = 'Housekeeper';
+--------------+-----+--------+-------------+--------+------------+------------+---------------------------+
| name         | age | gender | job         | salary | phone      | aadhar     | email                     |
+--------------+-----+--------+-------------+--------+------------+------------+---------------------------+
| Emily Taylor | 29  | Female | Housekeeper | 25000  | 9876543213 | 1234567893 | emily.taylor@example.com |
| Sarah Lewis  | 38  | Female | Housekeeper | 22000  | 9876543217 | 1234567897 | sarah.lewis@example.com  |
+--------------+-----+--------+-------------+--------+------------+------------+---------------------------+
```

11. Get Employees with Salary Above 30000

```
SELECT * FROM employee WHERE salary > 30000;
+--------------+-----+--------+----------+--------+------------+------------+---------------------------+
| name         | age | gender | job      | salary | phone      | aadhar     | email                     |
+--------------+-----+--------+----------+--------+------------+------------+---------------------------+
| John Miller  | 30  | Male   | Manager  | 50000  | 9876543210 | 1234567890 | john.miller@example.com   |
| Tom Scott    | 35  | Male   | Chef     | 40000  | 9876543212 | 1234567892 | tom.scott@example.com     |
| Robert Brown | 45  | Male   | Security | 35000  | 9876543214 | 1234567894 | robert.brown@example.com  |
| Anna Wilson  | 32  | Female | Manager  | 55000  | 9876543215 | 1234567895 | anna.wilson@example.com   |
| David Walker | 50  | Male   | Security | 36000  | 9876543218 | 1234567898 | david.walker@example.com  |
+--------------+-----+--------+----------+--------+------------+------------+---------------------------+
```

12. Get Driver Information for Available Drivers

```
SELECT * FROM driver WHERE available = 'Yes';
+-----------------+-----+--------+-----------------+-----------+-----------+---------------+
| name            | age | gender | company         | brand     | available | location      |
+-----------------+-----+--------+-----------------+-----------+-----------+---------------+
| John Smith      | 40  | Male   | ABC Transport   | Toyota    | Yes       | New York      |
| David Green     | 33  | Male   | Speed Cabs      | Ford      | Yes       | San Francisco |
| Laura Adams     | 36  | Female | Sunshine Rides  | Chevrolet | Yes       | Los Angeles   |
| Alice Clark     | 38  | Female | ABC Transport   | BMW       | Yes       | Miami         |
| George White    | 50  | Male   | XYZ Cabs        | Audi      | Yes       | Dallas        |
| Robert Martinez | 42  | Male   | Speed Cabs      | Toyota    | Yes       | Seattle       |
+-----------------+-----+--------+-----------------+-----------+-----------+---------------+
```

13. Get All Departments with Their Budget

```
SELECT * FROM department;
+----------------+--------+
| department     | budget |
+----------------+--------+
| Housekeeping   | 100000 |
| Front Desk     | 150000 |
| Security       | 120000 |
| Maintenance    | 80000  |
| Restaurant     | 200000 |
| Laundry        | 50000  |
| Transportation | 70000  |
| Administration | 250000 |
| Marketing      | 150000 |
| Events         | 30000  |
+----------------+--------+
```

14. Get Employees from the 'Front Desk' Department

```
SELECT * FROM employee WHERE job = 'Receptionist';
```

```
+----------------+-----+--------+--------------+--------+------------+------------+----------------------------+
| name           | age | gender | job          | salary | phone      | aadhar     | email                      |
+----------------+-----+--------+--------------+--------+------------+------------+----------------------------+
| Sara Parker    | 28  | Female | Receptionist | 30000  | 9876543211 | 1234567891 | sara.parker@example.com    |
| Jessica Harris | 27  | Female | Receptionist | 28000  | 9876543219 | 1234567899 | jessica.harris@example.com |
+----------------+-----+--------+--------------+--------+------------+------------+----------------------------+
```

15. Get Rooms by Price Range

```
SELECT * FROM room WHERE price BETWEEN 2000 AND 3000;
```

```
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 101         | Available    | Clean        | 2000  | Single   |
| 102         | Available    | Clean        | 2500  | Double   |
| 103         | Occupied     | Clean        | 3000  | Single   |
| 105         | Occupied     | Dirty        | 2200  | Single   |
| 106         | Available    | Clean        | 2700  | Single   |
| 107         | Occupied     | Clean        | 2300  | Double   |
| 108         | Available    | Dirty        | 2000  | Single   |
| 110         | Available    | Clean        | 3000  | Single   |
| 111         | Occupied     | Clean        | 2500  | Single   |
| 113         | Available    | Clean        | 2100  | Single   |
| 115         | Available    | Clean        | 2700  | Single   |
| 116         | Occupied     | Clean        | 3000  | Double   |
| 117         | Available    | Dirty        | 2600  | Single   |
| 118         | Occupied     | Clean        | 2200  | Double   |
| 119         | Available    | Clean        | 2800  | Single   |
+-------------+--------------+--------------+-------+----------+
```

16. Update Customer Status to 'Checked-out'

```
UPDATE customer SET status = 'Checked-out' WHERE id = 'C001';
```

17. Update Room Availability to 'Occupied'

```
UPDATE room SET availability = 'Occupied' WHERE room_number = '101';
```

18. Insert New Employee

```
INSERT INTO employee (name, age, gender, job, salary, phone, aadhar,
email)
VALUES ('Mark Johnson', '30', 'Male', 'Manager', '50000',
'9876543220', '1234567899', 'mark.johnson@example.com');
```

19. Delete Room by Room Number

```
DELETE FROM room WHERE room_number = '101';
```

20. Delete Customer by Customer ID

```
DELETE FROM customer WHERE id = 'C002';
```

21. Get Total Revenue from All Occupied Rooms

```
SELECT SUM(price) AS total_revenue
FROM room
WHERE availability = 'Occupied';
+---------------+
| total_revenue |
+---------------+
|         28000 |
+---------------+
```

22. Get the Number of Rooms in Each Availability Status

```
SELECT availability, COUNT(*) AS num_rooms
FROM room
GROUP BY availability;
+--------------+-----------+
| availability | num_rooms |
+--------------+-----------+
| Occupied     |        10 |
| Available    |        10 |
+--------------+-----------+
```

23. Get All Employees and Their Job Titles

```
SELECT name, job FROM employee;
+----------------+--------------+
| name           | job          |
+----------------+--------------+
| John Miller    | Manager      |
| Sara Parker    | Receptionist |
| Tom Scott      | Chef         |
| Emily Taylor   | Housekeeper  |
| Robert Brown   | Security     |
| Anna Wilson    | Manager      |
| Michael Clark  | Driver       |
| Sarah Lewis    | Housekeeper  |
| David Walker   | Security     |
| Jessica Harris | Receptionist |
| Mark Johnson   | Manager      |
+----------------+--------------+
```

24. Get All Customers Who Stayed in a Specific Room

```
SELECT c.name, c.room_number, c.status
FROM customer c
WHERE c.room_number = '101';
+----------+-------------+-------------+
| name     | room_number | status      |
+----------+-------------+-------------+
| John Doe | 101         | Checked-out |
+----------+-------------+-------------+
```

25. Find All Rooms with 'Dirty' Cleaning Status

```
SELECT * FROM room WHERE clean_status = 'Dirty';
```

```
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 105         | Occupied     | Dirty        | 2200  | Single   |
| 108         | Available    | Dirty        | 2000  | Single   |
| 112         | Available    | Dirty        | 3200  | Double   |
| 114         | Occupied     | Dirty        | 3500  | Double   |
| 117         | Available    | Dirty        | 2600  | Single   |
| 120         | Occupied     | Dirty        | 3300  | Double   |
+-------------+--------------+--------------+-------+----------+
```

26. Find Employees Who Have Worked for More Than 5 Years (based on age)

```
SELECT * FROM employee WHERE age > 30;
+---------------+-----+--------+-------------+--------+------------+------------+---------------------------+
| name          | age | gender | job         | salary | phone      | aadhar     | email                     |
+---------------+-----+--------+-------------+--------+------------+------------+---------------------------+
| Tom Scott     | 35  | Male   | Chef        | 40000  | 9876543212 | 1234567892 | tom.scott@example.com     |
| Robert Brown  | 45  | Male   | Security    | 35000  | 9876543214 | 1234567894 | robert.brown@example.com  |
| Anna Wilson   | 32  | Female | Manager     | 55000  | 9876543215 | 1234567895 | anna.wilson@example.com   |
| Michael Clark | 40  | Male   | Driver      | 30000  | 9876543216 | 1234567896 | michael.clark@example.com |
| Sarah Lewis   | 38  | Female | Housekeeper | 22000  | 9876543217 | 1234567897 | sarah.lewis@example.com   |
| David Walker  | 50  | Male   | Security    | 36000  | 9876543218 | 1234567898 | david.walker@example.com  |
+---------------+-----+--------+-------------+--------+------------+------------+---------------------------+
```

27. Get Customers by Country

```
SELECT * FROM customer WHERE country = 'USA';
+------+------------+-----------------+--------+---------+-------------+-------------+---------+
| id   | number     | name            | gender | country | room_number | status      | deposit |
+------+------------+-----------------+--------+---------+-------------+-------------+---------+
| C001 | 9876543210 | John Doe        | Male   | USA     | 101         | Checked-out | 1000    |
| C006 | 9876543215 | Jessica Wilson  | Female | USA     | 106         | Checked-in  | 1200    |
| C007 | 9876543216 | Michael Taylor  | Male   | USA     | 107         | Checked-out | 800     |
| C013 | 9876543222 | David Hall      | Male   | USA     | 113         | Checked-in  | 900     |
| C016 | 9876543225 | Susan King      | Female | USA     | 116         | Checked-in  | 1300    |
| C019 | 9876543228 | Steven Martinez | Male   | USA     | 119         | Checked-in  | 2000    |
+------+------------+-----------------+--------+---------+-------------+-------------+---------+
```

28. Get the Room with the Highest Price

```
SELECT * FROM room WHERE price = (SELECT MAX(price) FROM room);
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 109         | Occupied     | Clean        | 4000  | Double   |
+-------------+--------------+--------------+-------+----------+
```

29. Get the Total Number of Employees

```
SELECT COUNT(*) AS total_employees FROM employee;
+-----------------+
| total_employees |
+-----------------+
|              11 |
+-----------------+
```

30. Get a List of All Available Rooms Sorted by Price

```
SELECT * FROM room WHERE availability = 'Available' ORDER BY price
ASC;
+-------------+--------------+--------------+-------+----------+
| room_number | availability | clean_status | price | bed_type |
+-------------+--------------+--------------+-------+----------+
| 108         | Available    | Dirty        | 2000  | Single   |
| 113         | Available    | Clean        | 2100  | Single   |
| 102         | Available   | Clean       | 2500 | Double  |
| 117         | Available   | Dirty       | 2600 | Single  |
| 106         | Available   | Clean       | 2700 | Single  |
| 115         | Available   | Clean       | 2700 | Single  |
| 119         | Available   | Clean       | 2800 | Single  |
| 110         | Available   | Clean       | 3000 | Single  |
| 112         | Available   | Dirty       | 3200 | Double  |
| 104         | Available   | Clean       | 3500 | Double  |
+-------------+-------------+-------------+------+---------+
```

31. Get List of Drivers Working for a Specific Company

```
SELECT * FROM driver WHERE company = 'ABC Transport';
```

```
+-------------+-----+--------+---------------+--------+-----------+----------+
| name        | age | gender | company       | brand  | available | location |
+-------------+-----+--------+---------------+--------+-----------+----------+
| John Smith  | 40  | Male   | ABC Transport | Toyota | Yes       | New York |
| Alice Clark | 38  | Female | ABC Transport | BMW    | Yes       | Miami    |
+-------------+-----+--------+---------------+--------+-----------+----------+
```

32. Get the Average Salary of Employees

```
SELECT AVG(salary) AS average_salary FROM employee;
```

```
+--------------------+
| average_salary     |
+--------------------+
| 36454.545454545456 |
+--------------------+
```

33. Get List of Customers Who Have Deposited More Than 1000

```
SELECT * FROM customer WHERE deposit > 1000;
```

```
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
| id   | number     | name            | gender | country   | room_number | status      | deposit |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
| C004 | 9876543213 | Emily Davis     | Female | Australia | 104         | Checked-in  | 2000    |
| C006 | 9876543215 | Jessica Wilson  | Female | USA       | 106         | Checked-in  | 1200    |
| C008 | 9876543217 | Sarah Johnson   | Female | Canada    | 108         | Checked-in  | 1400    |
| C009 | 9876543218 | James Lee       | Male   | China     | 109         | Checked-in  | 1600    |
| C011 | 9876543220 | Charles Clark   | Male   | India     | 111         | Checked-in  | 1800    |
| C012 | 9876543221 | Linda Allen     | Female | Australia | 112         | Checked-out | 1500    |
| C015 | 9876543224 | Thomas Young    | Male   | UK        | 115         | Checked-in  | 1100    |
| C016 | 9876543225 | Susan King      | Female | USA       | 116         | Checked-in  | 1300    |
| C018 | 9876543227 | Karen Adams     | Female | Mexico    | 118         | Checked-in  | 1700    |
| C019 | 9876543228 | Steven Martinez | Male   | USA       | 119         | Checked-in  | 2000    |
| C020 | 9876543229 | Laura Perez     | Female | Canada    | 120         | Checked-out | 1200    |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
```

34. Get List of Customers Who Have Checked-out

```
SELECT * FROM customer WHERE status = 'Checked-out';
```

```
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
| id   | number     | name            | gender | country   | room_number | status      | deposit |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
| C001 | 9876543210 | John Doe        | Male   | USA       | 101         | Checked-out | 1000    |
| C003 | 9876543212 | Robert Smith    | Male   | UK        | 103         | Checked-out | 500     |
| C005 | 9876543214 | William Brown   | Male   | India     | 105         | Checked-out | 1000    |
| C007 | 9876543216 | Michael Taylor  | Male   | USA       | 107         | Checked-out | 800     |
| C010 | 9876543219 | Patricia Harris | Female | Mexico    | 110         | Checked-out | 600     |
| C012 | 9876543221 | Linda Allen     | Female | Australia | 112         | Checked-out | 1500    |
| C014 | 9876543223 | Maria Scott     | Female | Canada    | 114         | Checked-out | 700     |
| C017 | 9876543226 | Daniel Wright   | Male   | India     | 117         | Checked-out | 400     |
| C020 | 9876543229 | Laura Perez     | Female | Canada    | 120         | Checked-out | 1200    |
+------+------------+-----------------+--------+-----------+-------------+-------------+---------+
```

35. Update Employee Salary

```
UPDATE employee SET salary = '60000' WHERE name = 'Sara Parker';
```