Hotel Management System Case Study: Design and Implementation using DBMS

Satla Sree Varshith

VU21CSEN0100481

Bhuvan Lohit Dev Nakka

VU21CSEN0100463

Ratna Varshith Kolachala

VU21CSEN0100496

Introduction to use case

- The primary goal is to develop a robust and efficient database structure that supports various functionalities essential for efficient hotel operations.
- The system encompasses guest reservations, room management, billing, staff administration,feedback and reporting.
- The database design involves entity-relationship modeling, schema design, and query optimization to enhance performance and data integrity.
- The system also includes features for managing inventory, housekeeping, and guest feedback.

Problems faced in hotel management system

1. Integrating the hotel management system with various subsystems like point-of-sale (POS), accounting software, and third-party booking platforms can be complex.
2. Protecting guest and financial data is crucial. Data breaches can lead to reputation damage and legal issues.
3. Maintaining the system, including software updates and hardware maintenance, is a continuous task.
4. System downtime, software glitches, or network problems can disrupt hotel operations and inconvenience guests.
5. Adapting the system to accommodate growth, such as adding more rooms or properties, can be challenging.
6. Determining optimal room rates and managing pricing strategies to maximize revenue can be complex.
7. Handling guest complaints, feedback, and requests in a timely and satisfactory manner is essential for guest satisfaction.

Requirement analysis

Software Requirements:

Operating System:

Compatible with Windows, Linux, and macOS.

DBMS:

Use Oracle, MySQL, SQL Server, or SQLite.

Version Control:

Git for code management.

Documentation:

Tools like Word, LaTeX, or Markdown editors.

Security:

Encryption, access control, and vulnerability scanning.

Hardware Requirements:

Server :

Multi-core processor, ample RAM and storage, network connectivity.

Client Devices:

Standard devices (computers, mobile) with no strict requirements.

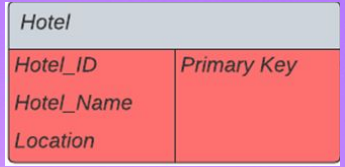
Networking:

Reliable network infrastructure.

Module Description

Hotel Table:

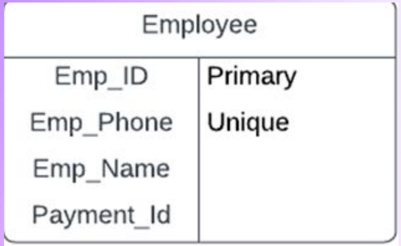
- Hotel_ID (Primary)
- Hotel_Name
- Location



Hotel	
Hotel_ID	Primary Key
Hotel_Name	
Location	

Employee Table:

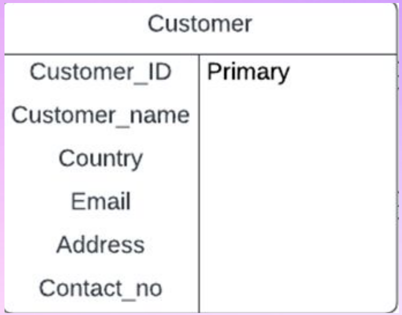
- Emp_ID (Primary)
- Emp_Name
- Emp_Phone(Unique)
- Payment_Id



Employee	
Emp_ID	Primary
Emp_Phone	Unique
Emp_Name	
Payment_Id	

Customer Table:

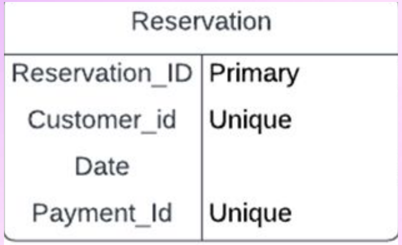
- Customer_ID (Primary)
- Name
- Country
- Email (Unique)
- Address



Customer	
Customer_ID	Primary
Customer_name	
Country	
Email	
Address	
Contact_no	

Reservation Table:

- Reservation_ID (Primary)
- Customer_id (Unique)
- Date
- Payment_Id (Unique)

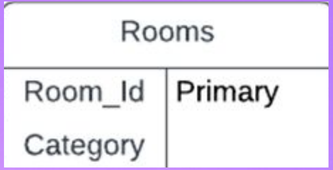


Reservation	
Reservation_ID	Primary
Customer_id	Unique
Date	
Payment_Id	Unique

Module Description

Rooms Table:

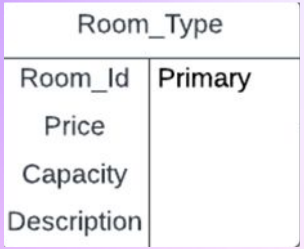
- Room_Id (Primary)
- Category



Rooms	
Room_Id	Primary
Category	

Room_Type Table:

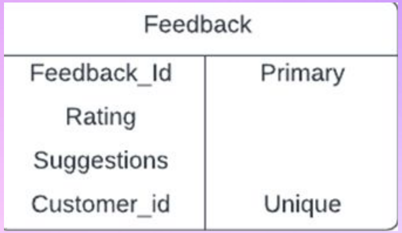
- Room_Id (Unique)
- Price
- Capacity
- Description



Room_Type	
Room_Id	Primary
Price	
Capacity	
Description	

Feedback Table:

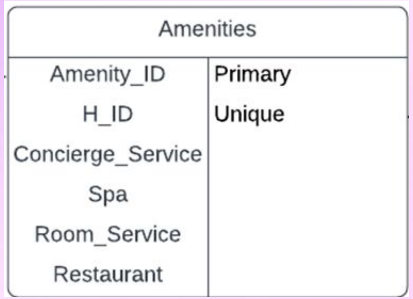
- Feedback_Id (Unique)
- Rating
- Suggestions
- Customer_id (Unique)



Feedback	
Feedback_Id	Primary
Rating	
Suggestions	
Customer_id	Unique

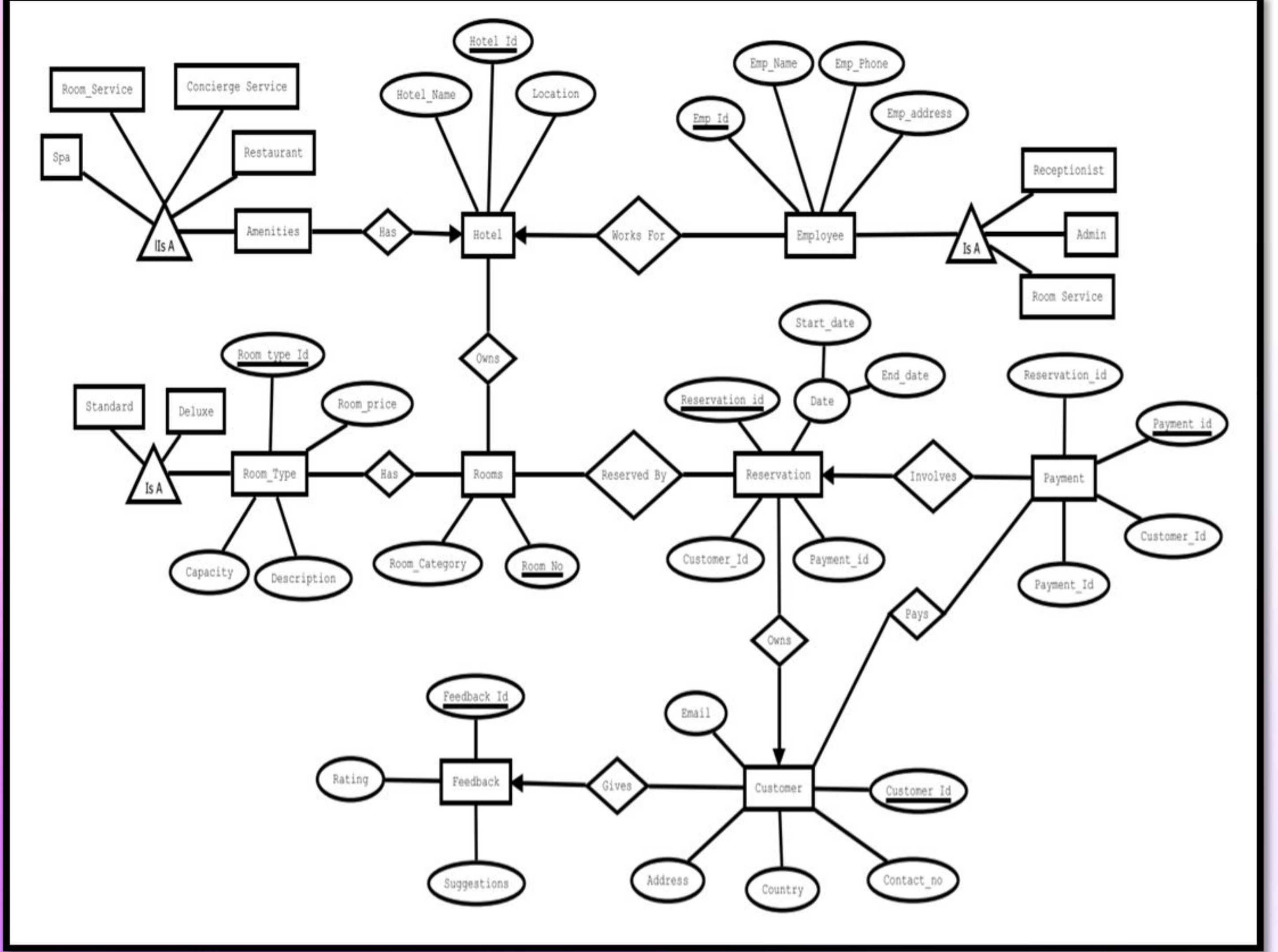
Amenities Table:

- Amenity_ID(Primary)
- H_ID (Unique)
- Concierge_Service
- Spa
- Room_Service

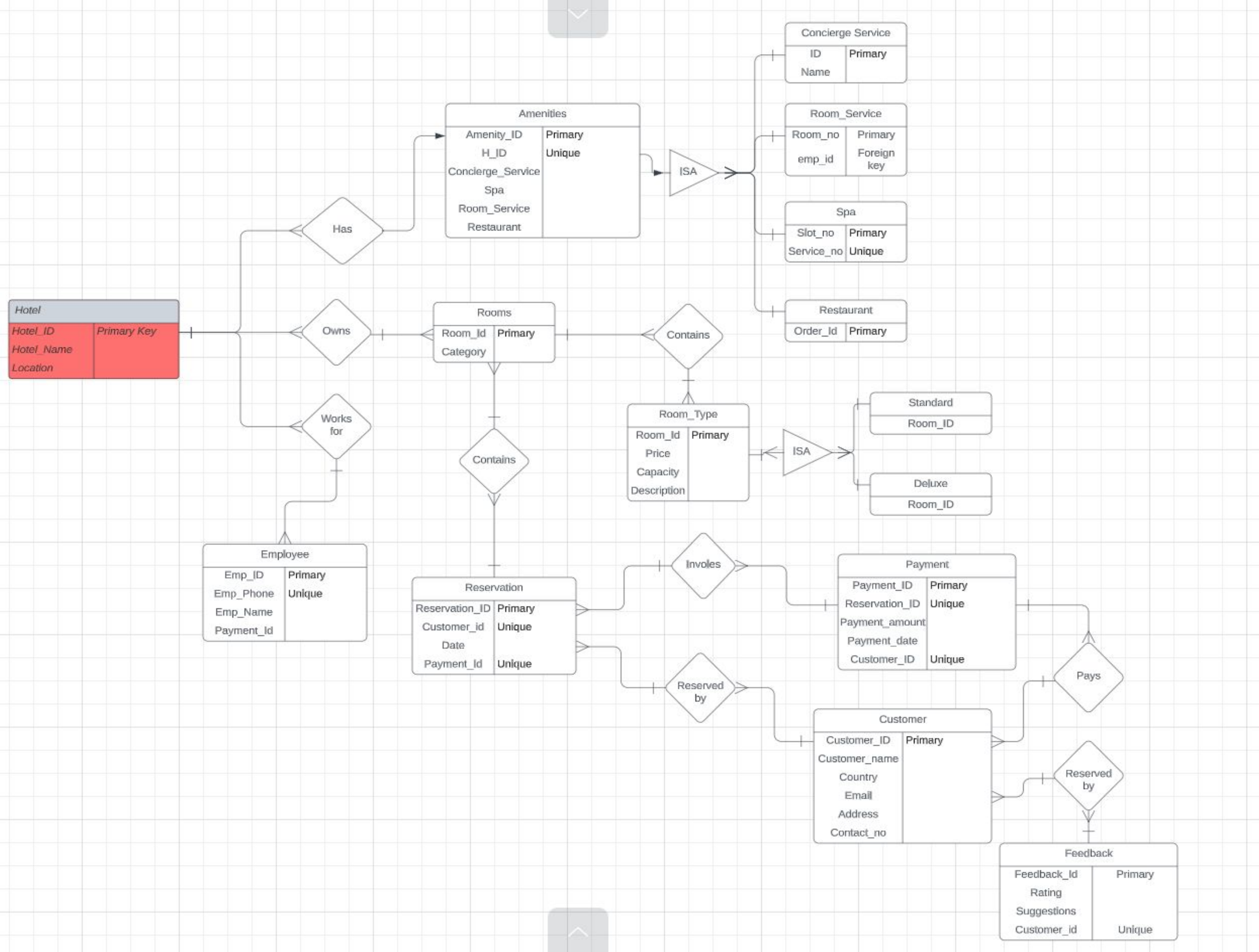


Amenities	
Amenity_ID	Primary
H_ID	Unique
Concierge_Service	
Spa	
Room_Service	
Restaurant	

ER Diagram


Hotel_Id
Room_Service
Concierge Service
Hotel_Name
Location
Emp_Name
Emp_Phone
Emp_Id
Emp_address
Spa
Restaurant
Receptionist
IIs A
Amenities
Has
Hotel
Works For
Employee
Is A
Admin
Room Service
Start_date
Owns
Room_type_Id
End_date
Standard
Deluxe
Room_price
Reservation_id
Date
Reservation_id
Payment_id
Is A
Room_Type
Has
Rooms
Reserved By
Reservation
Involves
Payment
Customer_Id
Capacity
Description
Room_Category
Room_No
Customer_Id
Payment_id
Payment_Id
Pays
Owns
Feedback_Id
Email
Rating
Feedback
Gives
Customer
Customer_Id
Suggestions
Address
Country
Contact_no

Lucid Chart



Hotel	
Hotel_ID	Primary Key
Hotel_Name	
Location	

Amenities	
Amenity_ID	Primary
H_ID	Unique
Concierge_Service	
Spa	
Room_Service	
Restaurant	

Concierge Service	
ID	Primary
Name	

Room_Service	
Room_no	Primary
emp_id	Foreign key

Spa	
Slot_no	Primary
Service_no	Unique

Restaurant	
Order_Id	Primary

Rooms	
Room_Id	Primary
Category	

Room_Type	
Room_Id	Primary
Price	
Capacity	
Description	

Standard
Room_ID

Deluxe
Room_ID

Employee	
Emp_ID	Primary
Emp_Phone	Unique
Emp_Name	
Payment_Id	

Reservation	
Reservation_ID	Primary
Customer_id	Unique
Date	
Payment_Id	Unique

Payment	
Payment_ID	Primary
Reservation_ID	Unique
Payment_amount	
Payment_date	
Customer_ID	Unique

Customer	
Customer_ID	Primary
Customer_name	
Country	
Email	
Address	
Contact_no	

Feedback	
Feedback_Id	Primary
Rating	
Suggestions	
Customer_id	Unique

Relationships in the Hotel Management System

- Concierge Service - Amenities
- Amenities - Spa
- Amenities - Room_Service
- Amenities - Restaurant
- Room_Service - Employee
- Spa - Amenities
- Rooms - Hotel
- Restaurant - Hotel
- Room_Type - Rooms
- Employee - Payment
- Reservation - Customer
- Reservation - Payment
- Payment - Customer
- Feedback - Customer

Cardinalities

One-to-One Relationships:

Concierge_Service to Amenities: 1-to-1

Amenities to Spa: 1-to-1

Amenities to Room_Service: 1-to-1

Amenities to Restaurant: 1-to-1

Room_Service to Employee: 1-to-1

Employee to Payment: 1-to-1

Reservation to Payment: 1-to-1

Feedback to Customer: 1-to-1

One-to-Many Relationships:

Rooms to Hotel: 1-to-Many (One Hotel can have many Rooms)

Restaurant to Hotel: 1-to-Many (One Hotel can have many Restaurants)

Reservation to Customer: 1-to-Many (One Customer can make many Reservations)

Many-to-Many Relationships:

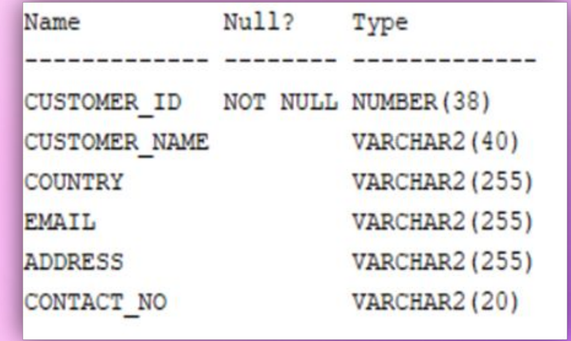
Room_Type to Rooms: Many-to-Many (Many Rooms can be of the same Room_Type, and one Room_Type can be associated with many Rooms).

LOGICAL DATABASE DESIGN

Table Creation

Customer Table:

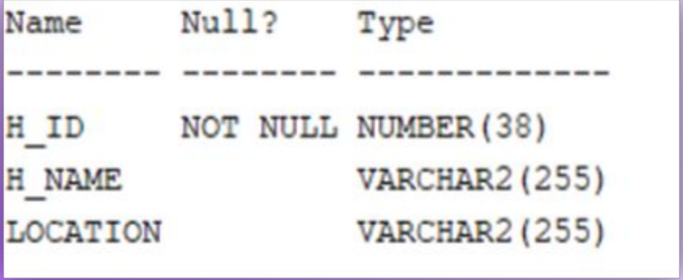
```
 CREATE TABLE Customer (customer_id INT PRIMARY KEY,

customer_name VARCHAR2(40),Country VARCHAR(255),

 email VARCHAR(255),Address VARCHAR(255),

Contact_no VARCHAR(20));
```



```
Name          Null?    Type
------------- -------- ------------
CUSTOMER_ID   NOT NULL NUMBER(38)
CUSTOMER_NAME          VARCHAR2(40)
COUNTRY                VARCHAR2(255)
EMAIL                  VARCHAR2(255)
ADDRESS                VARCHAR2(255)
CONTACT_NO             VARCHAR2(20)
```

Hotel Table:

```
CREATE TABLE Hotel (

            H_ID INT PRIMARY KEY,

            H_NAME VARCHAR(255),

            LOCATION VARCHAR(2550));
```



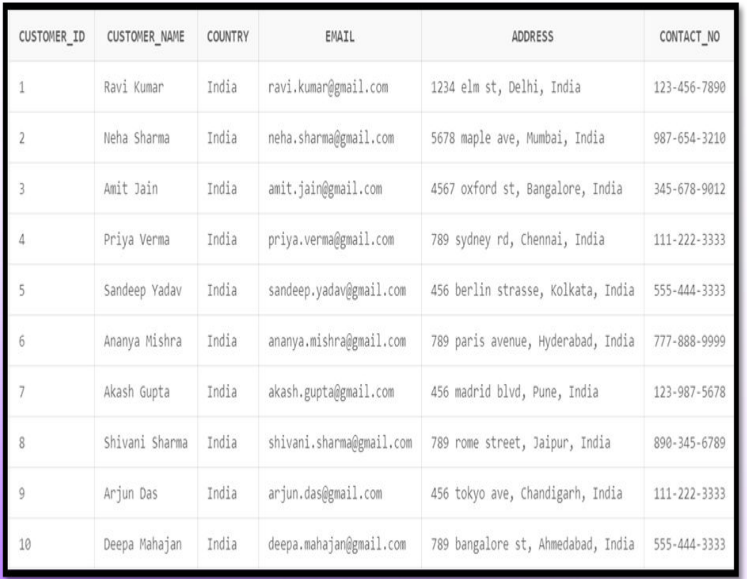
```
Name     Null?    Type
-------- -------- -------------
H_ID     NOT NULL NUMBER(38)
H_NAME            VARCHAR2(255)
LOCATION          VARCHAR2(255)
```

- We have created 17 other tables in SQL

INSERTION

Inserting values into Customer table:

```
INSERT INTO Customer values(1, 'Ravi Kumar', 'India', 'ravi.kumar@gmail.com', '1234 elm st, Delhi, India', '123-456-7890');

INSERT INTO Customer values(2, 'Neha Sharma', 'India', 'neha.sharma@gmail.com', '5678 maple ave, Mumbai, India',
'987-654-3210');

INSERT INTO Customer values(3, 'Amit Jain', 'India', 'amit.jain@gmail.com', '4567 oxford st, Bangalore, India',
'345-678-9012');

INSERT INTO Customer values(4, 'Priya Verma', 'India', 'priya.verma@gmail.com', '789 sydney rd, Chennai, India',
'111-222-3333');

INSERT INTO Customer values(5, 'Sandeep Yadav', 'India', 'sandeep.yadav@gmail.com', '456 berlin strasse, Kolkata, India',
'555-444-3333');

INSERT INTO Customer values(6, 'Ananya Mishra', 'India', 'ananya.mishra@gmail.com', '789 paris avenue, Hyderabad,
India', '777-888-9999');

INSERT INTO Customer values(7, 'Akash Gupta', 'India', 'akash.gupta@gmail.com', '456 madrid blvd, Pune, India',
'123-987-5678');

INSERT INTO Customer values(8, 'Shivani Sharma', 'India', 'shivani.sharma@gmail.com', '789 rome street, Jaipur, India',
'890-345-6789');

INSERT INTO Customer values(9, 'Arjun Das', 'India', 'arjun.das@gmail.com', '456 tokyo ave, Chandigarh, India',
'111-222-3333');

INSERT INTO Customer values(10, 'Deepa Mahajan', 'India', 'deepa.mahajan@gmail.com', '789 bangalore st, Ahmedabad,
India', '555-444-3333');
```



CUSTOMER_ID	CUSTOMER_NAME	COUNTRY	EMAIL	ADDRESS	CONTACT_NO
1	Ravi Kumar	India	ravi.kumar@gmail.com	1234 elm st, Delhi, India	123-456-7890
2	Neha Sharma	India	neha.sharma@gmail.com	5678 maple ave, Mumbai, India	987-654-3210
3	Amit Jain	India	amit.jain@gmail.com	4567 oxford st, Bangalore, India	345-678-9012
4	Priya Verma	India	priya.verma@gmail.com	789 sydney rd, Chennai, India	111-222-3333
5	Sandeep Yadav	India	sandeep.yadav@gmail.com	456 berlin strasse, Kolkata, India	555-444-3333
6	Ananya Mishra	India	ananya.mishra@gmail.com	789 paris avenue, Hyderabad, India	777-888-9999
7	Akash Gupta	India	akash.gupta@gmail.com	456 madrid blvd, Pune, India	123-987-5678
8	Shivani Sharma	India	shivani.sharma@gmail.com	789 rome street, Jaipur, India	890-345-6789
9	Arjun Das	India	arjun.das@gmail.com	456 tokyo ave, Chandigarh, India	111-222-3333
10	Deepa Mahajan	India	deepa.mahajan@gmail.com	789 bangalore st, Ahmedabad, India	555-444-3333

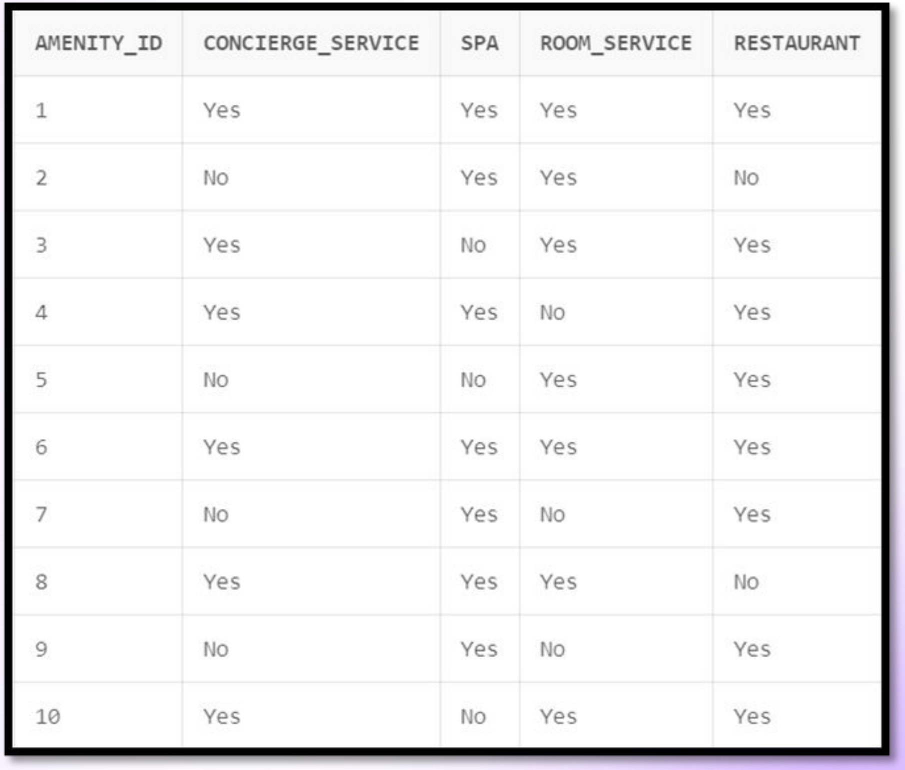
SQL Queries for Hotel Management System

SQL Query

List the amenities for the hotel:

- SELECT Amenity_ID, Concierge_Service, Spa, Room_Service, Restaurant

  FROM Amenities;

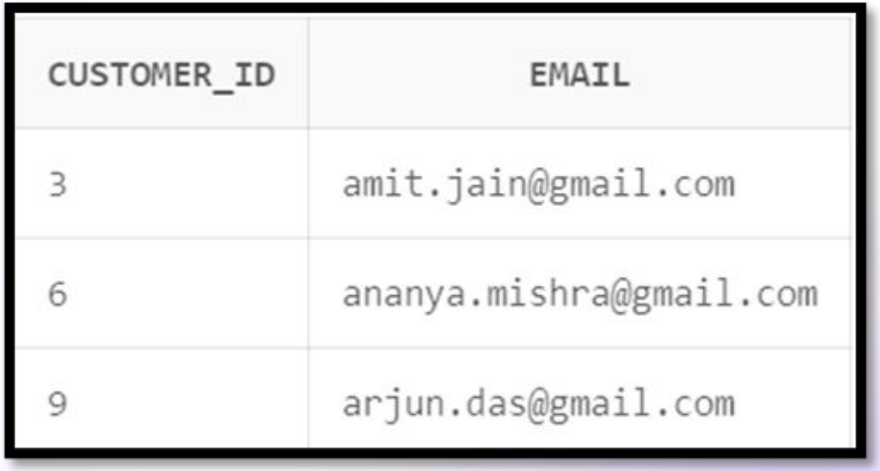


AMENITY_ID	CONCIERGE_SERVICE	SPA	ROOM_SERVICE	RESTAURANT
1	Yes	Yes	Yes	Yes
2	No	Yes	Yes	No
3	Yes	No	Yes	Yes
4	Yes	Yes	No	Yes
5	No	No	Yes	Yes
6	Yes	Yes	Yes	Yes
7	No	Yes	No	Yes
8	Yes	Yes	Yes	No
9	No	Yes	No	Yes
10	Yes	No	Yes	Yes

SQL Queries for Hotel Management System

List all the customers who made reservations for the Suite category.:

```
SELECT C.customer_id, C.email

FROM Customer C

JOIN Customer_Reservation CR ON C.customer_id =
CR.customer_id

JOIN Reservation R ON CR.reservation_id =
R.reservation_id

JOIN Room_Type RT ON R.occupancy_capacity =
RT.occupancy_capacity

WHERE RT.description = 'Suite';
```

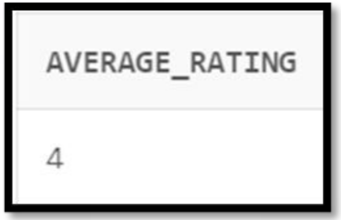


CUSTOMER_ID	EMAIL
3	amit.jain@gmail.com
6	ananya.mishra@gmail.com
9	arjun.das@gmail.com

SQL Queries for Hotel Management System

Find the average rating of feedback received from customers.

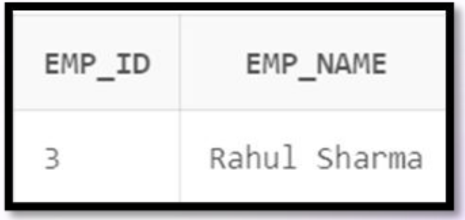
```
SELECT AVG(CF.Rating) AS average_rating

FROM Feedback CF;
```



AVERAGE_RATING
4

Retrieve the highest-paid staff member.

```
SELECT e.emp_id, e.emp_name

FROM Employee e

WHERE e.emp_salary = (SELECT MAX(emp_salary)
FROM Employee);
```



EMP_ID	EMP_NAME
3	Rahul Sharma

SQL Queries for Hotel Management System

SQL Query

Create a view that displays rooms with prices higher than 150.

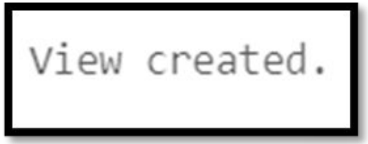
CREATE VIEW V1 AS

SELECT r.room_no, r.room_category, rt.description AS RoomType, rt.room_price

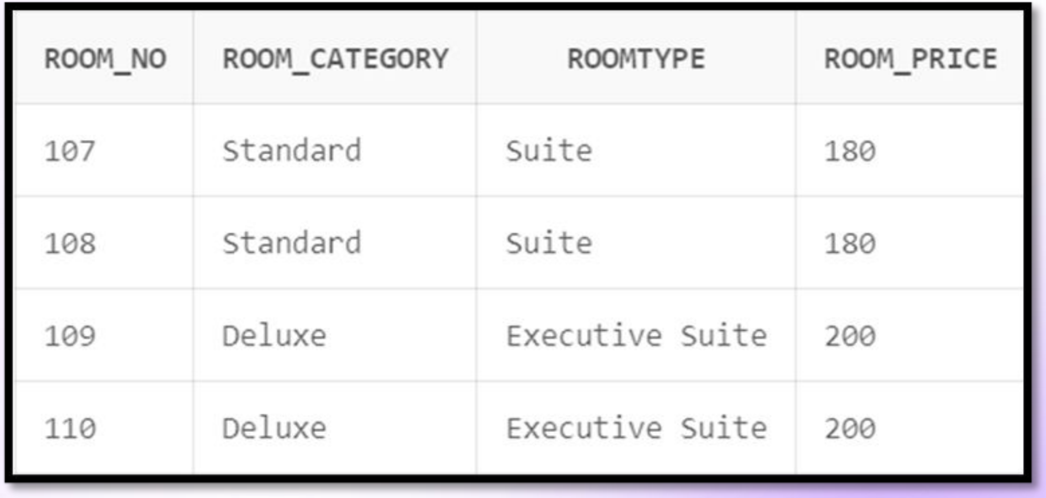
FROM Rooms r, Room_Type rt

WHERE r.room_type_id = rt.room_type_id AND rt.room_price > 150;

SELECT * FROM V1;



```
View created.
```



ROOM_NO	ROOM_CATEGORY	ROOMTYPE	ROOM_PRICE
107	Standard	Suite	180
108	Standard	Suite	180
109	Deluxe	Executive Suite	200
110	Deluxe	Executive Suite	200

Conclusion

1

The Hotel Management System is a crucial tool for streamlining operations and improving efficiency in the hotel industry.

2

The system also provides valuable insights through data analysis, allowing hotels to make informed decisions and enhance customer satisfaction.

3

By automating processes such as reservation management, check-in/check-out, and billing, the system reduces manual errors and saves time.