

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT**



**MIDTERM PROJECT**

**TOPIC: DESIGNING A DATABASE MODEL FOR A HOTEL  
RESERVATION BOOKING SYSTEM**

**Course:** Database

**Course ID:** 231IS9303

**Lecturer:** Lam Hong Thanh

*TP. Hồ Chí Minh, 12/2023*

## **LIST OF MEMBERS**

<b>Name</b>	<b>Student Code</b>	<b>Level of completion</b>
1. Le Thi Huong	K224141724	100%
2. Nguyen Thuong Ky Anh	K224141713	100%
3. Nguyen Thi Thuy Linh	K224141727	100%
4. Tran Phuong Nghi	K224141729	100%
5. Bui Dang Quynh Nhu	K224141733	100%

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT .....</b>	1
<b>PART I: OVERVIEW OF THE TOPIC .....</b>	2
1. Introduction .....	2
2. Functions Needed for Hotel Reserve Booking System.....	2
2.1. For Customers.....	2
2.2. For Hotel Managers .....	3
3. Business Process .....	3
4. Business Rules .....	5
5. DFD Diagram.....	7
<b>PART II: DESIGN OF ENTITY-RELATIONSHIP MODEL .....</b>	7
1. Entities participating in the model .....	7
2. Determining attributes of the entities.....	9
3. Relationship between entities.....	13
3.1. Relationship 1:1 .....	13
3.2. Relationship 1:n .....	13
3.3. Relationship n:n .....	14
4. ER (EER) Diagram .....	15
<b>PART III: LOGICAL DESIGN .....</b>	16
1. Transforming the EER diagram into Relations.....	16
1.1. Relations 1:1 .....	16
1.2. Relations 1:n .....	16
1.3. Relation n:n.....	17
1.4. Composite transformation .....	18
2. Data normalization .....	19
<b>PART IV: PHYSICAL DESIGN .....</b>	19
1. Choosing DBMS .....	19
2. Choosing data type, range value, constraints .....	19
3. Database Diagram .....	28
<b>PART V: IMPLEMENTATION .....</b>	28
1. Creating Database and Tables.....	28

2. Creating Function to be Used in a Computed Column .....	33
2.1. Total_Price (Reservation table) .....	33
2.2. Last_Price (Invoice table).....	34
3. Inserting Values into Tables.....	35
<b>PART VI: PERFORMING SOME QUERIES.....</b>	<b>45</b>

## **ACKNOWLEDGEMENT**

To complete this assignment, our team would like to sincerely thank Lecturer Lam Hong Thanh - the instructor of the Database course for her wholehearted help and facilitation for us in the learning process. She directly taught and guided us to do this exercise with great enthusiasm and care. We have invested a great deal of effort and seriousness in the process, hoping that our work will receive positive feedback.

In the process of writing this essay, due to limited understanding, it is difficult to avoid shortcomings. We look forward to receiving your comments to make the essay more and more complete.

## **PART I: OVERVIEW OF THE TOPIC**

### **1. Introduction**

The project examines the aspect of the hospitality industry which is Hotel management. In the 21st century the use of the internet, computers and other electronic devices have made handling different jobs and aspects of management very easy. This is the design and implementation of a database of a hotel reservation booking system of logical data and transactions in a centralized and organized manner and also provides a user-friendly interface with which the user can interact easily.

This is designed to create a platform that allows both the user and administrator to keep track of customer reservations, room category, services, payments, room rating and other relevant information involved in the hotel reservation booking system. The implementation is based on the requirements for a hotel reservation booking system.

The project accomplished the task of building a system that ensures accurate record maintenance which was done through proper identification of customers and the proper designation of user functions with most of the processes being done automatically. An electronic hotel management information system is required to assist management of data in the hospitality industry and also to make the entire hotel management process easier. The database system was created using SQL server (SQL).

Within the scope of this project, our main objective is to hone our skills in performing fundamental tasks such as identifying entities, relationships, drawing diagrams, and using basic SQL language. The most crucial aspect is to understand and successfully implement basic database problems.

### **2. Functions Needed for Hotel Reserve Booking System**

#### **2.1. For Customers**

##### **View Hotel Information:**

- Allow customers to choose room types, specify the number of guests, and select booking dates.
- Display detailed information about room types and prices.

#### View Schedule and Events:

- Display the schedule of events and activities at the hotel.
- Provide information about tours or special events.

#### Order Room Services:

- Allow customers to request room services such as food and beverages or housekeeping.

#### Payment and Promotion:

- Facilitate payment for pre-booked services and rooms, with the added benefit of promotions.

## **2.2. For Hotel Managers**

#### Reservation Management:

- Monitor, confirm, and manage room reservations.
- Perform functions like canceling reservations or modifying booking details.

#### Room and Schedule Management:

- Update information on room availability and the hotel's schedule.
- Adjust schedules and room allocations for special events or emergencies.

#### Service Management:

- Monitor and manage customer requests for room services.
- Ensure services are delivered as requested, maintaining quality.

#### Statistics and Reporting:

- Compile data and generate reports on business performance, room occupancy rates, and customer reviews.
- A hotel management system should provide a user-friendly interface for both customers and managers to optimize user experience and managerial efficiency.

## **3. Business Process**

**Step 1:** Customer provides the necessary information: Number of guests, Check-in date, Check-out date, and preferred Location. This allows the system to search and display hotels that meet the customer's requirements.

**Step 2:** The system uses the entered information to search for suitable hotels. Then, displaying a list of hotels with detailed information, photos, and prices to assist customers in choosing hotels based on their needs and personal preferences.

**Step 3:** The customer selects the desired hotel and views a list of available room types. Customer chooses the room type and specifies the number of rooms needed

**Step 4:** In addition, the hotel also provides a number of convenient services such as Transportation, Tour, Food & Beverage,...The customer selects additional services to add to the reservation. These services are optional, and if the customer does not require them, they can skip this step

**Step 5:** Customer provides personal information including: First name, Last name, Date of birth, Phone number, Email, ID card.

**Step 6:** Customer reviews the room selected, selected services and personal information. Then, entering payment method and confirm the reservation.

**Step 7:** System compiles the total price, including room charges, additional services, and transaction fees, while accounting for any applicable discounts. In cases where promotions are applied, customers who input a valid discount code, meet the specified conditions, and complete a reservation on the promotional day will be eligible for a discount. After compiling, the system provides customer an invoice code for payment.

**Step 8:** In case of choosing the online payment method, customer pays invoice based on the invoice code provided. If the customer chooses the option for cash payment, the customer will pay at the hotel during the stay. Simultaneously, it completes the final step in the booking process.

**Step 9:** System manager is responsible for monitoring the entire process, including storing essential hotel and customer information. Additionally, the manager handles the operation of the system and addresses any errors that customers encountered in the booking process.

**Step 10:** Once the customer completes the booking process , the system notifies the hotel manager with necessary details about the reservation.

**Step 11:** After checking the customer's necessary information, the hotel manager confirms the reservation. Simultaneously, the hotel manager updates the room status on the system to reflect the change from 'Available' to 'Not available.'

**Step 12:** Hotel manager sends an email about successful reservation notification to the customer. Email includes a reservation code and hotel contact information.

**Step 13:** After staying at hotel, customers rate their experience on a scale from excellent to very bad (excellent/good/average/bad/very bad) and provide a detailed description of their stay. Subsequently, the system receives and analyzes this feedback, utilizing customer input to calculate the ratings for different hotels.

#### **4. Business Rules**

- Customer is identified by Customer ID and has First Name, Last Name, Gender, Identified ID, Date of Birth, Contact No, Email. A customer can make no reservations or multiple reservations, but a reservation must belong to one customer.

A customer can choose multiple hotels or none at all, and each hotel can be chosen multiple by multiple customers or not chosen by any customer.

- Hotel is identified by Hotel ID and has attributes of Hotel Name, Address. A hotel has many management staff but one management staff belongs to only one hotel.

- Room is identified by Room No and has a floor and description. A hotel will have multiple rooms but each room belongs to only one hotel.

Customers can book multiple rooms for a minimum of one, and each room will be booked by multiple customers or not booked by any customer.

- Room Category is identified by Category Code and has Category\_Name, Category Description, Capacity, Room Price. A room has only one and only one room category while one room category is available in multiple rooms.

- Service has an identifier Service ID and has Service Price, Service Name. A customer can book zero, one or more services and one service can serve none or many customers.

A reservation can book zero or more services, and each service can be booked by one or more reservations.

- Reservation has an identifier Reservation ID and has attributes Check-in date, Check-out date, Total Price, Discount Percent, Reservation Date. A hotel can be booked by no or more reservations but a reservation can only be made in advance of one and only one hotel.

- Invoice is identified by Invoice ID, Issue Time, Last Price. A customer can have multiple invoices but one invoice belongs to only one customer.

A reservation can issue 1 and only 1 invoice, while a invoice is issued one and only one reservation

- Payment has identifier a Payment No and has Payment Method, Time Paid. One payment pays only one and one invoice and one invoice is also paid by only one payment.

A payment is made by one and only one customer but one customer makes one or more payments.

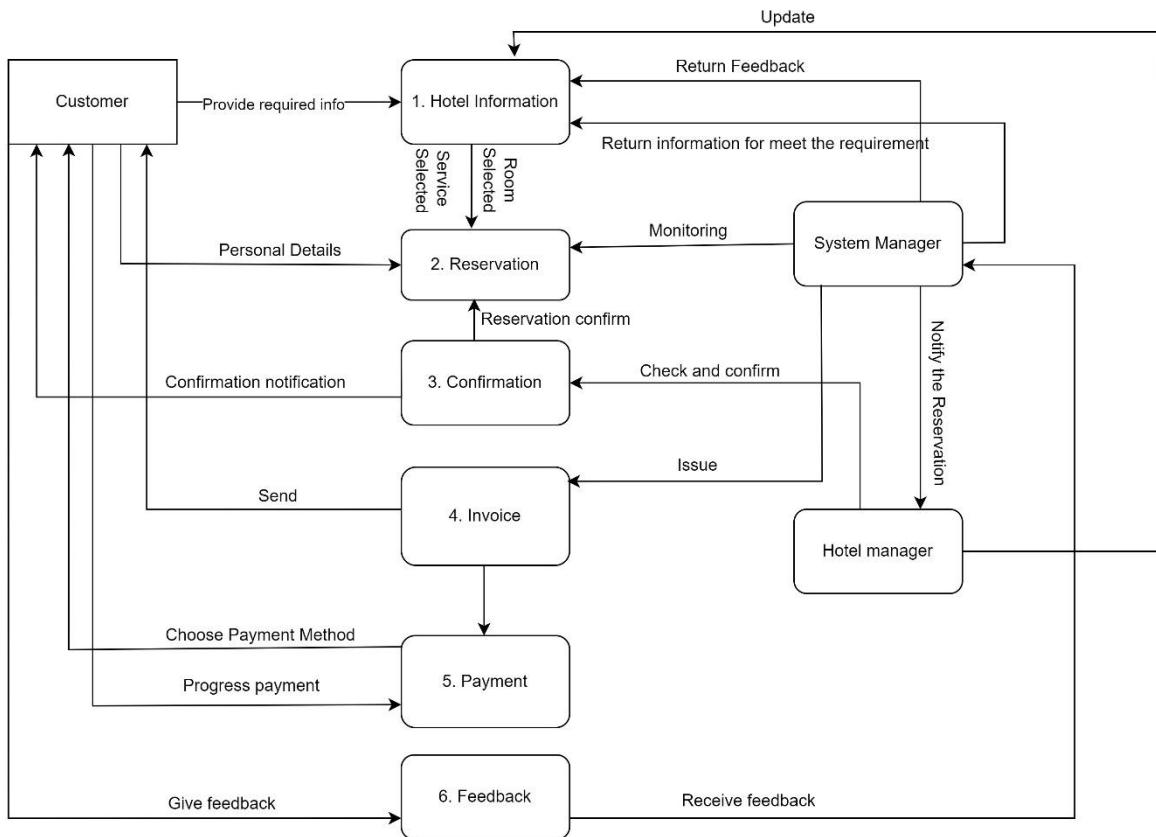
- Hotel Manager is identified by HEmployee ID and has First Name, Last Name. A Hotel Manager can approve one or more Reservations but one reservation is approved by only one manager.

- System Manager is identified by SEmployee ID and has First Name, Last Name. The system manager can monitor multiple hotels at the same time, one hotel is supervised by one system manager.

- A customer can rate one, many, or no rooms at all, and a room can be rated by one customer,no customers or multiple customers at different times.

- A room can be booked for none, one reservation or many reservations during one time slot and one reservation can book none, one or many rooms.

## 5. DFD Diagram



## PART II: DESIGN OF ENTITY-RELATIONSHIP MODEL

### 1. Entities participating in the model

No	Entity	Description
1	<b>Customer</b>	Represents a customer engaging in the process of reserving hotel rooms.
2	<b>Hotel</b>	Represents information about hotels, including hotel ID, name, and address provided for customers.
3	<b>Room Category</b>	Represents different types of rooms, such as double rooms, single rooms, etc.

<b>4</b>	<b>Reservation</b>	Represents an object involved in storing information about room reservations.
<b>5</b>	<b>Hotel Manager</b>	Represents managing information about individuals or departments directly involved in overseeing the operations of the hotel.
<b>6</b>	<b>Service</b>	Represents a service provided to the hotel guests, such as room service, dining, and entertainment.
<b>7</b>	<b>Room</b>	Represents a unit of accommodation within the hotel. This entity contains information related to the rooms that customers can reserve in the hotel.
<b>8</b>	<b>Payment</b>	Represents Tracking and managing the payment process for customers, and providing detailed information about payment transactions related to hotel room reservations.
<b>9</b>	<b>Invoice</b>	Represents an object involved in tracking payment information and invoices for room reservations and used services.
<b>10</b>	<b>System Manager</b>	Represents managing system administrators, defining permissions, and tracking administrative activities related to hotel room reservations and other services.
<b>11</b>	<b>Room Rating</b>	Represents customer reviews of hotel rooms, providing detailed information about service quality and user experience.

## 2. Determining attributes of the entities

<b>Entity</b>	<b>Attribute</b>	<b>Identifier</b>	<b>Required or Optional</b>	<b>Description</b>
<b>Customer</b>	Customer_ID	X	Required	A series of numbers used to uniquely identify a specific customer.
	First_Name		Required	First name of customer.
	Last_Name		Required	Last name of customer.
	Gender		Required	Gender of customer.
	Date_of_Birth		Optional	Customer's date of birth.
	Contact_No		Required	Customer's contact phone number.
	Email		Required	Customer's contact email.
<b>Hotel</b>	Identified_ID		Required	Customer's national identification number.
	Hotel_ID	X	Required	A series of numbers used to uniquely identify a specific hotel.
	Hotel_Name		Required	Name of hotel.
<b>Room Category</b>	Address (Composite)		Required	Location of hotel.
	Category_Code	X	Required	A string of characters used to uniquely identify a specific room type.

<b>Reservation</b>	Category_Name		Required	Name of the room type.
	Category_Description		Required	Detailed description of each room type.
	Room_Price		Required	Price of each room type.
	Capacity		Required	Maximum number of guests for each room type.
	Reservation_ID	X	Required	A string of characters and numbers used to uniquely identify each specific room reservation.
	Day_Checkin		Required	Check-in date for the booked room.
	Day_Checkout		Required	Check-out date for the booked room.
	Total_Price		Required	The total amount that the customer has to pay without applying any discount codes.
	Reservation_Date		Required	Booking date of customer.
	Discount_Percent		Optional	Discount percentage for booking.
<b>Hotel Manager</b>	HEmployee_ID	X	Required	A string of characters and numbers used to uniquely identify each specific hotel manager.

	First_Name		Required	First name of manager hotel.
	Last_Name		Required	Last name of manager hotel.
Service	Service_ID	X	Required	A string of characters and numbers used to uniquely identify each specific service of hotel.
	Service_Name		Required	Service name the customer intends to use.
	Service_Price		Required	Fee for the service utilized by the customer.
Room	Room_ID	X	Required	A series of numbers used to uniquely identify each specific room of each hotel on the system
	Floor		Required	Floor number of the room.
	Description		Required	Description of the unique features of each room.
	Room_No		Required	Room number
Invoice	Invoice_ID	X	Required	A string of characters and numbers used to uniquely identify each specific payment invoice.
	Issue_Time		Required	Invoice printing time.

	Last_Price		Required	Final amount the customer has to pay after the discount.
Payment	Payment_No	X	Required	A string of characters and numbers used to uniquely identify each transaction code of the customer's invoices.
	Payment_Method		Required	Customer's payment methods such as cash payment or bank transfer.
	Time_Paid		Required	Time of payment for the room by the customer.
	SEmployee_ID	X	Required	A string of characters and numbers used to uniquely identify each system manager.
System Manager	First_Name		Required	First name of system manager.
	Last_Name		Required	Last name of system manager.
Room Rating (associative entity)	Room_Rating		Required	Customer satisfaction rating for the hotel room.
	Rating_Description		Optional	Comments and feedback on the hotel room as well as the hotel services.

### **3. Relationship between entities**

#### **3.1 Relationship 1:1**

No	Relationship between entities	Type of relationship	Explanation
1	Invoice - Reservation	1:1	A reservation can issue 1 and only 1 invoice, while a invoice is issued from 1 and only 1 reservation.
2	Invoice - Payment	1:1	A Payment can pay for 1 and only 1 invoice, while 1 invoice is issued for 1 and only 1 payment.

#### **3.2 Relationship 1:n**

No	Relationship between entities	Type of relationship	Explanation
1	Customer - Reservation	1:n	A customer can have 1, many or no reservations, while a reservation belongs to 1 and only 1 customer.
2	Customer - Invoice	1:n	A customer can receive 1 or many invoices, while an invoice belongs to 1 and only 1 customer.
3	Customer - Payment	1:n	A customer can issue 1 or many payments, while a payment can be issued by 1 and only 1 customer.
4	Hotel Manager - Reservation	1:n	A hotel manager can manage 1 or many reservations, while a reservation can be managed by 1 and only 1 hotel manager.

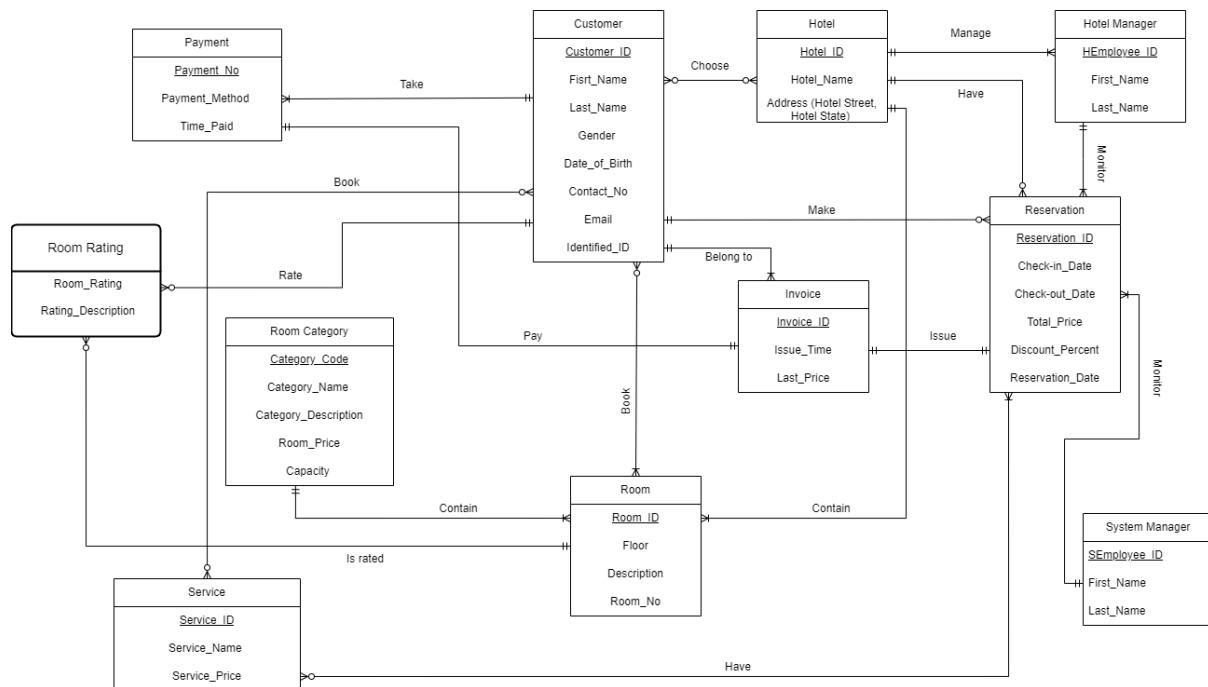
<b>5</b>	Hotel - Reservation	1:n	A hotel can have 1, many or no reservations, while 1 reservation belongs to 1 and only 1 hotel.
<b>6</b>	System Manager - Reservation	1:n	A system manager can manage 1 or many reservations, while a reservation can be managed by 1 and only 1 system manager.
<b>7</b>	Room Category - Room	1:n	A room category can contain 1 or many rooms, while a room belongs to 1 and only 1 room category.
<b>8</b>	Hotel - Room	1:n	A hotel can contain 1 or many rooms, while a room belongs to 1 and only 1 hotel.
<b>9</b>	Hotel - Hotel Manager	1:n	A hotel can be managed by one or many hotel managers, while 1 hotel manager can manage 1 and only 1 hotel.

### 3.3 Relationship n:n

No	Relationship between entities	Type of relationship	Explanation
<b>1</b>	Customer - Hotel	n:n	A customer can book one, many or no hotels, while 1 hotel can serve 1, many or no customers.
<b>2</b>	Customer - Room	n:n	A customer can book one or many rooms, while 1 room can serve 1, many or no customers.
<b>3</b>	Customer - Service	n:n	A customer can book one, many or no services, while 1 service can serve 1, many

			or no customers.
4	Customer - Room (Associative)	n:n	A customer can rate one, many or no rooms, while 1 room can be rated by 1, many or no customers.
5	Service - Reservation	n:n	A reservation can book one, many or no services, while 1 service can serve 1 or many reservation.
6	Room - Reservation	n:n	A reservation can be processed for one, many or no rooms, while 1 room can be reserved for 1, many or no rooms.

#### 4. ER (EER) Diagram



## PART III: LOGICAL DESIGN

### 1. Transforming the EER diagram into Relations

#### 1.1 Relations 1:1

No	Relationship between entities	Type of relationship	Relation transformation
1	Invoice - Reservation	1:1	Transfer primary key of Reservation (Reservation_ID) becomes the foreign key of Invoice.
2	Invoice - Payment	1:1	Transfer the primary key of Invoice (Invoice_ID) becomes the foreign key of Payment.

#### 1.2 Relations 1:n

No	Relationship between entities	Type of relationship	Relation transformation
1	Customer - Reservation	1:n	Transfer the primary key of Customer (Customer_ID) becomes the foreign key of Reservation.
2	Customer - Invoice	1:n	Transfer the primary key of Customer (Customer_ID) becomes the foreign key of Invoice.
3	Customer - Payment	1:n	Transfer the primary key of Customer (Customer_ID) becomes the foreign key of Payment.
4	Hotel Manager - Reservation	1:n	Transfer the primary key of Hotel Manager (HEmployee_ID) becomes the foreign key

			of Reservation.
5	Hotel - Reservation	1:n	Transfer the primary key of Hotel (Hotel_ID) becomes the foreign key of Reservation.
6	System Manager - Reservation	1:n	Transfer the primary key of System Manager (SEmployee_ID) becomes a foreign key of Reservation.
7	Room Category - Room	1:n	Transfer the primary key of Room Category (Category_Code) becomes a foreign key of Room.
8	Hotel - Room	1:n	Transfer the primary key of Hotel (Hotel_ID) becomes the foreign key of Room.
9	Hotel - Hotel Manager	1:n	Transfer the primary key of Hotel (Hotel_ID) becomes the foreign key of Hotel Manager.

### 1.3 Relation n:n

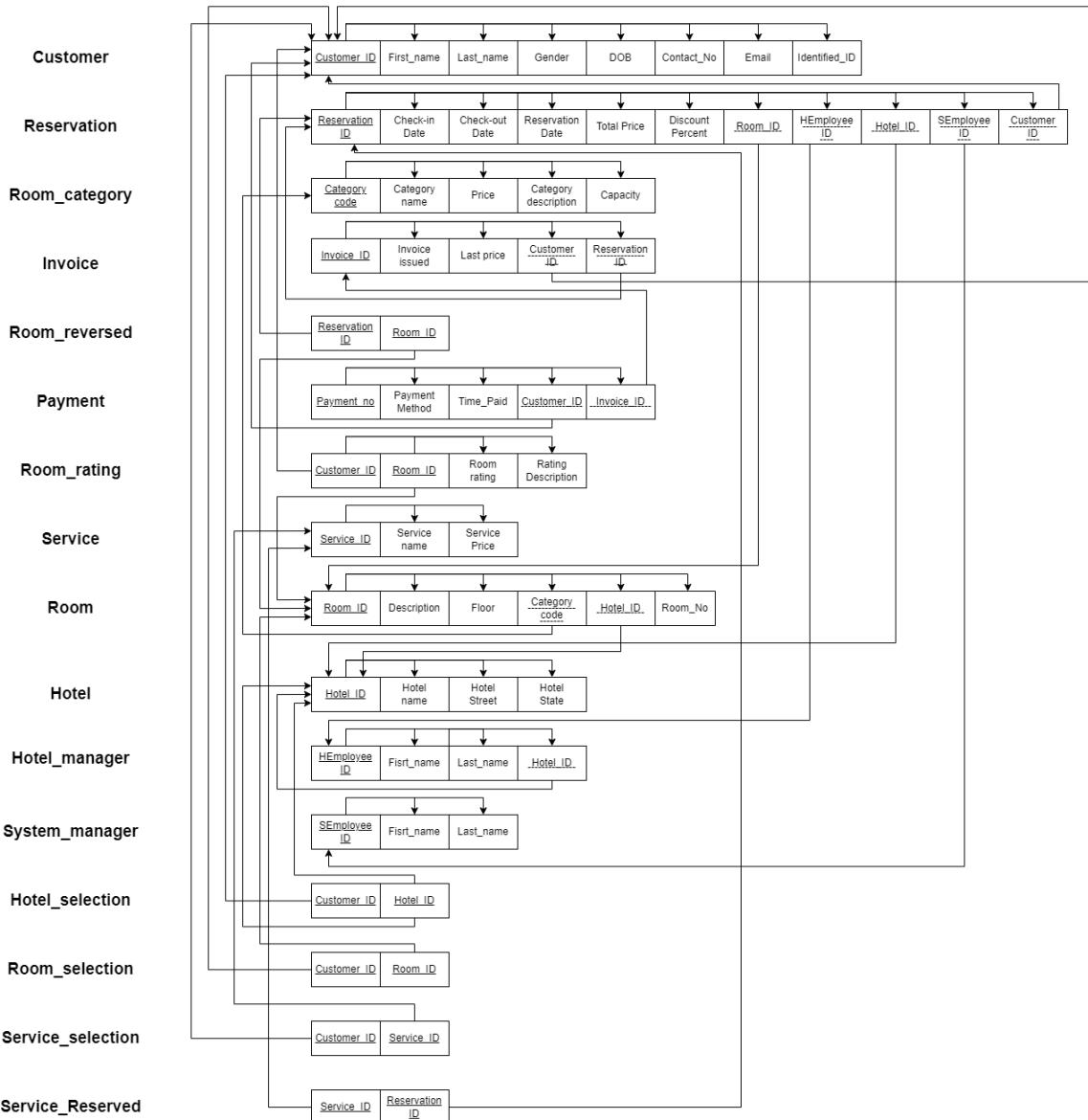
No	Relationship between entities	Type of relationship	Relation transformation
1	Customer - Hotel	n:n	Create a new entity called "Hotel_selection" containing 2 primary keys of Customer (Customer_ID) and Hotel (Hotel_ID).
2	Customer - Room	n:n	Create a new entity called "Room_selection" containing 2 main keys

			of Customer (Customer_ID) and Room (Room_No).
3	Customer - Service	n:n	Create a new entity called "Service_selection" containing 2 primary keys of Customer (Customer_ID) and Service (Service_ID).
4	Customer - Room (Associative)	n:n	Creates a new entity called "Room_rating" containing 2 primary keys of Customer (Customer_ID) and Room (Room_No).
5	Service - Reservation	n:n	Create a new entity named "Service_reserved" containing 2 primary keys of Service (Service_ID) and Reservation (Reservation_ID).
6	Room - Reservation	n:n	Create a new entity named "Room_reversed" containing 2 primary keys of Room (Room_No) and Reservation (Reservation_ID).

#### 1.4 Composite transformation

The Hotel entity has a multi-valued attribute called Address, so the values are separated into attributes including: Hotel City, Hotel District, Hotel Province.

## 2. Data normalization



## PART IV: PHYSICAL DESIGN

### 1. Choosing DBMS

Database Management System: SQL Server

### 2. Choosing data type, range value, constraints

#### Customer Table

<b>Field</b>	<b>Data type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Customer_ID	numeric(10,0)	0-10	Primary key
First_Name	varchar(20)	0-20	
Last_Name	varchar(20)	0-20	
Gender	varchar(6)	0-6	Check(Gender IN ('male', 'female'))
Date_of Birth	date		
Contact_No	nvarchar(20)	0-20	
Email	nvarchar(30)	0-30	
Identifier_ID	numeric(20,0)	0-20	

### **Hotel Table**

<b>Field</b>	<b>Data type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Hotel_ID	numeric(10,0)	0-10	Primary key
Hotel_Name	varchar(50)	0-50	
Hotel_Street	varchar(30)	0-30	
Hotel_State	varchar(30)	0-30	

### **Hotel\_Selection Table**

<b>Field</b>	<b>Data type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Hotel_ID	numeric(10,0)	0-10	Primary key, Foreign key from <b>Hotel_Selection</b> reference to Hotel(Hotel_ID)

Customer_ID	numeric(10,0)	0-10	Primary key, Foreign key from <b>Hotel_Selection</b> reference to Customer(Customer_ID)
-------------	---------------	------	---

### Room\_Category Table

Field	Data type	Domain Constraint	Another Constraints
Category_Code	varchar(8)	0-8	Primary key
Category_Name	varchar(20)	0-20	
Category_Description	varchar(50)	0-50	
Room_Price	decimal(10,2)		
Capacity	int		

### Room Table

Field	Data Type	Domain Constraint	Another Constraints
Room_ID	numeric(20,0)		Primary key
Room_No	int		
Floor	int		
Description	varchar(50)	0-50	
Category_Code	varchar(8)	0-8	Foreign key from <b>Room</b> reference to

			Room_Category (Category_Code)
Hotel_ID	numeric(10,0)	0-10	Foreign key from <b>Room</b> reference to Hotel (Hotel_ID)

### **System\_Manager Table**

Field	Data Type	Domain Constraint	Another Constraints
SEmployee_ID	char(4)	4	Primary key
First_Name	varchar(20)	0-20	
Last_Name	varchar(20)	0-20	

### **Hotel\_Manager Table**

Field	Data Type	Domain Constraint	Another Constraints
HEmployee_ID	varchar(6)	0-6	Primary key
First_Name	varchar(20)	0-20	
Last_Name	varchar(20)	0-20	
Hotel_ID	numeric(10,0)	0-10	Foreign key from <b>Hotel_Manager</b> reference to Hotel(Hotel_ID)

### **Room\_Selection Table**

<b>Field</b>	<b>Data type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Room_ID	numeric(20,0)		Primary key, Foreign key from <b>Room_Selection</b> reference to Room(Room_ID)
Customer_ID	numeric(10,0)	0-10	Primary key, Foreign key from <b>Room_Selection</b> reference to Customer (Customer_ID)

### Service Table

<b>Field</b>	<b>Data Type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Service_ID	varchar(6)	0-6	Primary key
Service_Name	varchar(30)	0-30	
Service_Price	decimal(10,2)		

### Service\_Selection Table

<b>Field</b>	<b>Data type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Customer_ID	numeric(10,0)	0-10	Primary key, Foreign key from <b>Service_Selection</b> reference to Customer (Customer_ID)
Service_ID	varchar(6)	0-6	Primary key, Foreign key from <b>Service_Selection</b> reference to

			Service (Service_ID)
--	--	--	----------------------

### Reservation Table

Field	Data type	Domain Constraint	Another Constraints
Reservation_ID	varchar(10)	0-10	Primary key
Check-in_Date	date		
Check-out_Date	date		
Reservation_Date	date		
Discount_Percent	decimal(5,2)		
Room_ID	numeric(20,0)		Foreign key from <b>Reservation</b> reference to Room(Room_ID)
HEmployee_ID	varchar(6)	0-6	Foreign key from <b>Reservation</b> reference to Hotel_Manager (HEmployee_ID)
Hotel_ID	numeric(10,0)	0-10	Foreign key from <b>Reservation</b> reference to Hotel(Hotel_ID)
SEmployee_ID	char(4)	4	Foreign key from <b>Reservation</b> reference to System_Manager (SEmployee_ID)
Customer_ID	numeric(10,0)	0-10	Foreign key from <b>Reservation</b> reference to Customer (Customer_ID)
Total_Price	decimal(10,2)		Use function, computed column

### Invoice Table

Field	Data Type	Domain Constraint	Another Constraints
Invoice_ID	varchar(10)	0-10	Primary key
Invoice_issued	datetime		
Customer_ID	numeric(10,0)	0-10	Foreign key from <b>Invoice</b> reference to Customer (Customer_ID)
Reservation_ID	varchar(10)	0-10	Foreign key from <b>Invoice</b> reference to Reservation (Reservation_ID)
Last_Price	decimal(10,2)		Use function, computed column

### Payment Table

Field	Data Type	Domain Constraint	Another Constraints
Payment_No	numeric(10,0)	0-10	Primary key
Payment_Method	varchar(20)	0-20	
Time_paid	datetime		
Customer_ID	numeric(10,0)	0-10	Foreign key from <b>Payment</b> reference to Customer(Customer_ID)
Invoice_ID	varchar(10)	0-10	Foreign key from <b>Payment</b> reference to Invoice(Invoice_ID)

### **Room\_Rating Table**

<b>Field</b>	<b>Data Type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Customer_ID	numeric(10,0)	0-10	Primary key, Foreign key from <b>Room_rating</b> reference to Customer (Customer_ID)
Room_ID	numeric(20,0)		Primary key, Foreign key from <b>Room_rating</b> reference to Room (Room_ID)
Room_Rating	int		CHECK(Room_Rating in(1,2,3,4,5))
Rating_Description	varchar(200)	0-200	

### **Service\_Reserved Table**

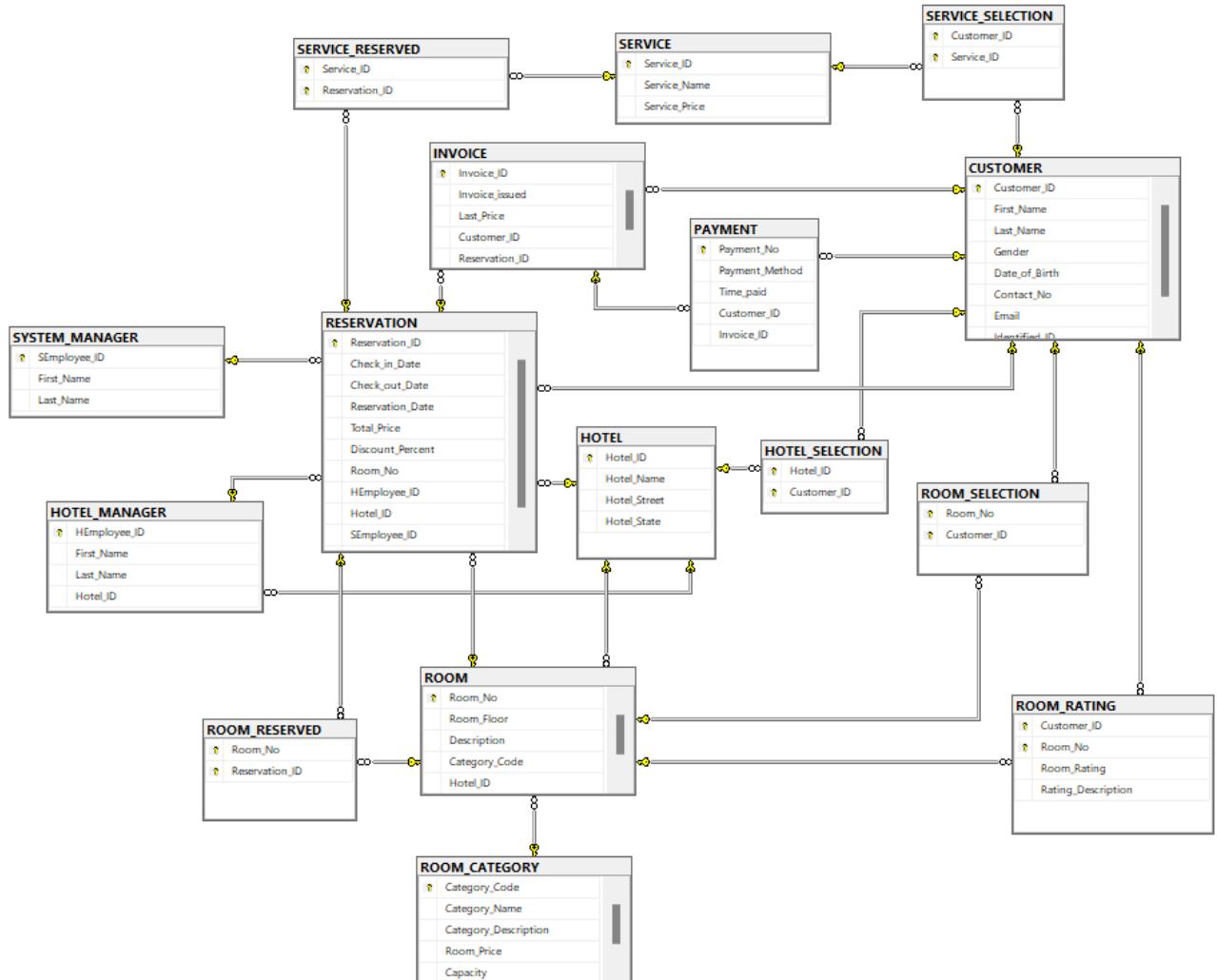
<b>Field</b>	<b>Data type</b>	<b>Domain Constraint</b>	<b>Another Constraints</b>
Service_ID	varchar(6)	0-6	Primary key, Foreign key from <b>Service_Reserved</b> reference to Service(Service_ID)

Reservation_ID	varchar(10)	0-10	Primary key, Foreign key from <b>Service_Reserved</b> reference to Reservation(Reservation_ID)
----------------	-------------	------	--

### **Room\_Reserved Table**

Field	Data type	Domain Constraint	Another Constraints
Room_ID	numeric(20,0)		Primary key, Foreign key from <b>Room_Reserved</b> reference to Room(Room_ID)
Reservation_ID	varchar(10)	0-10	Primary key, Foreign key from <b>Room_Reserved</b> reference to Reservation(Reservation_ID)

### 3. Database Diagram



## PART V: IMPLEMENTATION

### 1. Creating Database and Tables

CREATE DATABASE HRS

CREATE TABLE CUSTOMER(

```

Customer_ID NUMERIC(10,0) PRIMARY KEY ,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20) NOT NULL,
Gender VARCHAR(6) NOT NULL CHECK (Gender IN ('Male', 'Female')),
Date_of_Birth DATE,
Contact_No NVARCHAR(20) NOT NULL,
  
```

```
Email NVARCHAR(30) NOT NULL,  
Identified_ID NUMERIC(20,0) NOT NULL)
```

```
CREATE TABLE HOTEL(  
    Hotel_ID NUMERIC(10,0) PRIMARY KEY,  
    Hotel_Name VARCHAR(50) NOT NULL,  
    Hotel_Street VARCHAR(30) NOT NULL,  
    Hotel_State VARCHAR(30) NOT NULL);
```

```
CREATE TABLE HOTEL_SELECTION(  
    Hotel_ID NUMERIC(10,0),  
    Customer_ID NUMERIC(10,0),  
    PRIMARY KEY (Hotel_ID, Customer_ID),  
    FOREIGN KEY (Hotel_ID) REFERENCES Hotel(Hotel_ID),  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
);
```

```
CREATE TABLE ROOM_CATEGORY(  
    Category_Code VARCHAR(8) PRIMARY KEY,  
    Category_Name VARCHAR(20) not null,  
    Category_Description VARCHAR(50) not null,  
    Room_Price Decimal(10,2) not null,  
    Capacity int not null  
);
```

```
CREATE TABLE ROOM(  
    Room_ID NUMERIC(20,0) PRIMARY KEY,  
    Room_No INT,  
    Floor INT NOT NULL,  
    Description VARCHAR(50) not null,  
    Category_Code VARCHAR(8) not null ,
```

```
    Hotel_ID NUMERIC(10,0) not null,  
    FOREIGN KEY (Category_Code) REFERENCES  
Room_Category(Category_Code),  
    FOREIGN KEY (Hotel_ID) REFERENCES Hotel(Hotel_ID)  
);
```

```
CREATE TABLE SYSTEM_MANAGER(  
    SEmployee_ID CHAR(4) PRIMARY KEY,  
    First_Name VARCHAR(20) not null,  
    Last_Name VARCHAR(20) not null  
);
```

```
CREATE TABLE HOTEL_MANAGER(  
    HEmployee_ID VARCHAR(6) PRIMARY KEY,  
    First_Name VARCHAR(20) not null,  
    Last_Name VARCHAR(20) not null,  
    Hotel_ID NUMERIC(10,0) FOREIGN KEY REFERENCES Hotel(Hotel_ID)  
);
```

```
CREATE TABLE ROOM_SELECTION(  
    Room_ID NUMERIC(20,0),  
    Customer_ID NUMERIC(10,0),  
    PRIMARY KEY (Room_ID, Customer_ID),  
    FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID),  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
);
```

```
CREATE TABLE SERVICE(  
    Service_ID VARCHAR(6) PRIMARY KEY,  
    Service_Name VARCHAR(30) not null,  
    Service_Price Decimal(10,2) not null
```

);

```
CREATE TABLE SERVICE_SELECTION(
    Customer_ID NUMERIC(10,0),
    Service_ID VARCHAR(6),
    PRIMARY KEY (Customer_ID, Service_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Service_ID) REFERENCES Service(Service_ID)
);
```

```
CREATE TABLE RESERVATION(
    Reservation_ID VARCHAR(10) PRIMARY KEY,
    Check_in_Date DATE NOT NULL,
    Check_out_Date DATE NOT NULL,
    Reservation_Date DATE NOT NULL,
    Discount_Percent DECIMAL(5,2),
    Room_ID NUMERIC(20,0) FOREIGN KEY REFERENCES ROOM(Room_ID) ,
    HEmployee_ID VARCHAR(6) FOREIGN KEY REFERENCES
    Hotel_Manager(HEmployee_ID) ,
    Hotel_ID NUMERIC(10,0) FOREIGN KEY REFERENCES Hotel(Hotel_ID) ,
    SEmployee_ID CHAR(4) FOREIGN KEY REFERENCES
    System_Manager(SEmployee_ID),
    Customer_ID NUMERIC(10,0) FOREIGN KEY (Customer_ID) REFERENCES
    Customer(Customer_ID)
);
```

```
CREATE TABLE INVOICE(
    Invoice_ID VARCHAR(10) PRIMARY KEY,
    Invoice_issued DATETIME NOT NULL,
    Customer_ID NUMERIC(10,0) FOREIGN KEY REFERENCES
    Customer(Customer_ID) ,
```

```
Reservation_ID VARCHAR(10) FOREIGN KEY REFERENCES
Reservation(Reservation_ID)
);
```

```
CREATE TABLE PAYMENT(
    Payment_No NUMERIC(10,0) PRIMARY KEY,
    Payment_Method VARCHAR(20) not null CHECK(Payment_Method
IN('Cash','Online Transfer','Credit Card')),
    Time_paid DATETIME not null,
    Customer_ID NUMERIC(10,0) FOREIGN KEY REFERENCES
Customer(Customer_ID),
    Invoice_ID VARCHAR(10) FOREIGN KEY (Invoice_ID) REFERENCES
Invoice(Invoice_ID)
);
```

```
CREATE TABLE ROOM_RATING(
    Customer_ID NUMERIC(10,0),
    Room_ID NUMERIC(20,0),
    Room_Rating INT not null CHECK (Room_Rating IN (1,2,3,4,5)),
    Rating_Description VARCHAR(200),
    PRIMARY KEY (Customer_ID, Room_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID)
);
```

```
CREATE TABLE SERVICE_RESERVED(
    Service_ID VARCHAR(6),
    Reservation_ID VARCHAR(10),
    PRIMARY KEY (Service_ID, Reservation_ID),
    FOREIGN KEY (Service_ID) REFERENCES Service(Service_ID),
    FOREIGN KEY (Reservation_ID) REFERENCES Reservation(Reservation_ID)
```

);

```
CREATE TABLE ROOM_RESERVED(
    Room_ID NUMERIC(20,0),
    Reservation_ID VARCHAR(10),
    PRIMARY KEY (Room_ID, Reservation_ID),
    FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID),
    FOREIGN KEY (Reservation_ID) REFERENCES Reservation(Reservation_ID)
);
```

## **2. Creating Function to be Used in a Computed Column**

### **2.1 Total\_Price (Reservation table)**

```
CREATE FUNCTION dbo.TotalPrice(
    @Check_in_Date DATE,
    @Check_out_Date DATE,
    @Room_ID NUMERIC(20,0),
    @Reservation_ID VARCHAR(10)
)
RETURNS DECIMAL(20,2)
AS
BEGIN
    DECLARE @Room_Price DECIMAL(10,2);
    DECLARE @Service_Price DECIMAL(10,2);
    -- Get Room_Price values
    SELECT @Room_Price = RC.Room_Price
    FROM ROOM_CATEGORY RC
    JOIN ROOM R ON RC.Category_Code = R.Category_Code
    WHERE R.Room_ID = @Room_ID;
    -- Get Service_Price values
    SELECT @Service_Price = S.Service_Price
    FROM SERVICE S
```

```

JOIN SERVICE_RESERVED SR ON S.Service_ID = SR.Service_ID
WHERE SR.Reservation_ID= @Reservation_ID;
-- Calculate and return results
RETURN (CEILING(DATEDIFF(DAY, @Check_in_Date, @Check_out_Date)))
* @Room_Price + COALESCE(@Service_Price,0);
END;
-- Add column into table
ALTER TABLE RESERVATION
ADD Total_Price AS dbo.TotalPrice(Check_in_Date, Check_out_Date, Room_ID,
Reservation_ID);

```

## **2.2 Last\_Price (Invoice table)**

```

CREATE FUNCTION dbo.LastPrice(
    @RESERVATION_ID VARCHAR(10))
RETURNS DECIMAL(10,2)
AS
BEGIN
    DECLARE @Total_Price DECIMAL(10,2);
    DECLARE @Discount_Percent DECIMAL(10,2);
    -- Get Total_Price, Discount_Percent values
    SELECT @Total_Price = RS.Total_Price,
        @Discount_Percent = ISNULL(RS.Discount_Percent, 0)
    FROM RESERVATION RS
    WHERE RS.Reservation_ID = @RESERVATION_ID;
    RETURN @Total_Price * (1 - @Discount_Percent);
END;
-- Add column into table
ALTER TABLE INVOICE
ADD LAST_PRICE AS dbo.LastPrice(RESERVATION_ID);

```

### **3. Inserting Values into Tables**

#### **CUSTOMER**

```
INSERT INTO Customer (Customer_ID, First_Name, Last_Name, Gender,  
Date_of_Birth, Contact_No, Email, Identified_ID)
```

#### **VALUES**

```
(6111045842, 'John', 'Doe', 'Male', '1990-01-01', '123-456-7890',  
'john.doe@email.com', 518465),  
(6522547126, 'Jane', 'Smith', 'Female', NULL, '987-654-3210',  
'jane.smith@email.com', 456758),  
(6233146325, 'Bob', 'Johnson', 'Male', '1995-08-20', '552-575-5555',  
'bob.johnson@email.com', 784109),  
(6144798102, 'Alice', 'Williams', 'Female', NULL, '111-111-5147',  
'alice.williams@email.com', 147012),  
(6275246101, 'Charlie', 'Davis', 'Male', '1992-11-05', '945-999-2546',  
'charlie.davis@email.com', 780345),  
(6096421362, 'Eva', 'Anderson', 'Female', '1988-07-12', '561-333-1889',  
'eva.anderson@email.com', 676328),  
(6747101023, 'Daniel', 'Miller', 'Male', '1998-04-25', '3224-166-6656',  
'daniel.miller@email.com', 922101),  
(6568278431, 'Sophia', 'Brown', 'Female', '1983-09-30', '512-442-4241',  
'sophia.brown@email.com', 203234),  
(6449112235, 'Frank', 'Taylor', 'Male', '1993-06-15', '773-222-5622',  
'frank.taylor@email.com', 527067),  
(6320001268, 'Olivia', 'White', 'Female', NULL, '602-777-7527',  
'olivia.white@email.com', 899010),  
(2654321098, 'Liam', 'Johnson', 'Male', '1991-03-18', '987-654-3211',  
'liam.johnson@email.com', 369852),  
(5785432109, 'Emma', 'Anderson', 'Female', '1985-12-04', '123-456-7899',  
'emma.anderson@email.com', 753951),  
(9876543210, 'Mason', 'Williams', 'Male', '1994-07-27', '555-777-8888',  
'mason.williams@email.com', 456321),
```

(1967654321, 'Ava', 'Taylor', 'Female', NULL, '888-999-7777',  
'ava.taylor@email.com', 123789);

## **HOTEL**

INSERT INTO HOTEL (Hotel\_ID, Hotel\_Name, Hotel\_Street, Hotel\_State)  
VALUES

(21110, 'Grand Plaza Hotel', '123 Main Street', 'California'),  
(21211, 'Ocean View Resort', '456 Ocean Avenue', 'Washington'),  
(21312, 'Sunset Oasis Hotel', '789 Sunset Boulevard', 'Florida'),  
(21413, 'Riverside Lodge', '101 Riverside Drive', 'Arizona'),  
(21514, 'Urban Luxe Hotel', '512 Urban Street', 'Nevada'),  
(21615, 'Golden Sands Resort', '737 Beach Road', 'Colorado'),  
(21716, 'Skyline Suites', '888 Skyline Avenue', 'Georgia'),  
(21817, 'Central View Hotel', '456 Central Street', 'New York'),  
(21918, 'Desert Oasis Resort', '657 Oasis Road', 'Texas'),  
(22019, 'Tropical Paradise Hotel', '111 Palm Lane', 'Hawaii'),  
(22120, 'Forest Retreat Lodge', '30 Forest Lane', 'Oregon'),  
(22221, 'Cityscape Grand Hotel', '393 City View Road', 'Illinois'),  
(22322, 'Polar Lights Resort', '444 Aurora Avenue', 'Alaska'),  
(22423, 'Bourbon Street Inn', '777 Bourbon Street', 'Louisiana'),  
(22524, 'Mountain Serenity Lodge', '888 Mountain View Road', 'Montana'),  
(22625, 'Red Rock Resort', '974 Red Rock Lane', 'Utah'),  
(227, 'Harbor View Hotel', '365 Harbor Drive', 'Massachusetts');

## **HOTEL\_SELECTION**

INSERT INTO HOTEL\_SELECTION (Hotel\_ID, Customer\_ID)  
VALUES

(21110, 6111045842), -- John Doe at Grand Plaza Hotel  
(21110, 2654321098),  
(21110, 5785432109),  
(21413, 6522547126), -- Jane Smith at Riverside Lodge

(21514, 6233146325), -- Bob Johnson at Urban Luxe Hotel  
(21211, 6144798102), -- Alice Williams at Ocean View Resort  
(22019, 6275246101), -- Charlie Davis at Tropical Paradise Hotel  
(22524, 6096421362), -- Eva Anderson at Bourbon Street Inn  
(21312, 6747101023), -- Daniel Miller at Sunset Oasis Hotel  
(21312, 9876543210),  
(22221, 6568278431), -- Sophia Brown at Cityscape Grand Hotel  
(21615, 6449112235), -- Frank Taylor at Golden Sands Resort  
(22625, 6320001268), -- Olivia White at Red Rock Resort  
(22777, 1967654321);

## **ROOM CATEGORY**

INSERT INTO ROOM\_CATEGORY (Category\_Code, Category\_Name,  
Category\_Description, Room\_Price, Capacity)

VALUES

('A101', 'Standard Single', 'Basic single room with essential amenities', 100, 1),  
(‘A102’, ‘Standard Double’, ‘Standard room with a double bed’, 150.5, 2),  
(‘A103’, ‘Deluxe Suite’, ‘Spacious suite with additional amenities’, 250.75, 4),  
(‘A108’, ‘Penthouse’, ‘Luxurious penthouse with panoramic views’, 500.8, 6),  
(‘A110’, ‘Presidential Suite’, ‘The most luxurious suite in the hotel’, 1000, 6);

## **ROOM**

INSERT INTO Room (Room\_ID, Room\_No, Floor, Description, Category\_Code,  
Hotel\_ID)

VALUES

(587159533,102, 1, 'Oceanfront room with balcony', 'A101', 21110),  
(587159534,103, 1,'Coastal-themed room with balcony', 'A102', 21110),  
(587159598,1102,11,'Romantic suite with sunset view','A108',21110),  
(587159965,908,9,'Family-friendly room with play area','A103',21110),  
(369852147,202, 2, 'Luxury room with pool view', 'A108', 21413),  
(103245698,302, 3, 'Executive room with skyline view', 'A110', 21514),

(874198941,103, 1, 'Cozy room with city view', 'A102', 21211),  
(554698745,502, 5, 'Elegant suite with beach view', 'A110', 22019),  
(456789365,1012, 10, 'Executive suite with private balcony', 'A108', 22524),  
(789654153,610, 6,'Artistic room with mountain backdrop', 'A103', 21312),  
(789654123,201, 2, 'Spacious suite with mountain view', 'A103', 21312),  
(897456123,302, 3, 'Charming room with garden view', 'A101', 22221),  
(258741236,403, 4, 'Modern room with street view', 'A102', 21615),  
(987456313,401, 4, 'Classic room with courtyard view', 'A103', 22625),  
(312398745,9021, 9, 'Tranquil room with river view', 'A103', 22625),  
(227541258,1021, 10, 'Contemporary room with urban vibe', 'A110', 22777),  
(789684523,503, 5, 'Deluxe room with lake view', 'A108', 227);

## **SYSTEM MANAGER**

INSERT INTO System\_Manager (SEmployee\_ID, First\_Name, Last\_Name)  
VALUES  
(‘E011’, ‘John’, ‘Smith’),  
(‘E012’, ‘Alice’, ‘Johnson’),  
(‘E013’, ‘David’, ‘Lee’),  
(‘E014’, ‘Emily’, ‘Wang’),  
(‘E015’, ‘Michael’, ‘Brown’),  
(‘E016’, ‘Sophia’, ‘Davis’),  
(‘E017’, ‘Daniel’, ‘Nguyen’),  
(‘E018’, ‘Olivia’, ‘Miller’),  
(‘E019’, ‘James’, ‘Garcia’),  
(‘E020’, ‘Emma’, ‘Taylor’);

## **HOTEL MANAGER**

INSERT INTO Hotel\_Manager (HEmployee\_ID, First\_Name, Last\_Name, Hotel\_ID)  
VALUES  
(‘M1011’, ‘John’, ‘Smith’, 21110),  
(‘M1021’, ‘Alice’, ‘Johnson’, 21211),

('M1031', 'Bob', 'Miller', 21312),  
('M1041', 'Eva', 'Brown', 21413),  
('M1052', 'Charlie', 'Davis', 21514),  
('M1062', 'Sophia', 'Taylor', 21615),  
('M1074', 'Daniel', 'Anderson', 21716),  
('M1087', 'Olivia', 'White', 21817),  
('M1092', 'Frank', 'Jones', 21918),  
('M1100', 'Emma', 'Williams', 22019);  
('M1111', 'Liam', 'Johnson', 22221),  
('M1132', 'Emma', 'Anderson', 22322),  
('M1153', 'Mason', 'Williams', 22423),  
('M1194', 'Ava', 'Taylor', 22524),  
('M1165', 'Noah', 'Davis', 22625),  
('M1166', 'Sophia', 'Brown', 227);

## **ROOM\_SELECTION**

INSERT INTO ROOM\_SELECTION (Room\_ID, Customer\_ID)  
VALUES

(587159533, 6111045842),  
(587159534, 6111045842),  
(587159598, 2654321098),  
(587159965, 5785432109),  
(369852147, 6522547126),  
(103245698, 6233146325),  
(874198941, 6144798102),  
(554698745, 6275246101),  
(456789365, 6096421362),  
(789654153, 6747101023),  
(789654123, 9876543210),  
(897456123, 6568278431),  
(258741236, 6449112235),

(987456313, 6320001268),  
(312398745, 6320001268),  
(227541258, 1967654321),  
(789684523, 1967654321);

## **SERVICE**

```
INSERT INTO Service (Service_ID, Service_Name, Service_Price)
VALUES
('S51121', 'Room Cleaning', 30.50),
('S51241', 'Airport Shuttle', 40.00),
('S51332', 'Breakfast Buffet', 40.00),
('S51557', 'Laundry', 30.25),
('S51610', 'Fitness Center Access', 45.00),
('S51727', 'Spa', 50.00);
```

## **SERVICE\_SELECTION**

```
INSERT INTO SERVICE_SELECTION (Customer_ID, Service_ID)
VALUES
(6111045842, 'S51121'),
(2654321098, 'S51241'),
(5785432109, 'S51332'),
(6522547126, 'S51557'),
(6233146325, 'S51610'),
(6144798102, 'S51727'),
(6275246101, 'S51121'),
(9876543210, 'S51241'),
(6568278431, 'S51332'),
(6449112235, 'S51557');
```

## **RESERVATION**

```
INSERT INTO RESERVATION (Reservation_ID, Check_in_Date, Check_out_Date,
```

Reservation\_Date, Discount\_Percent, Room\_ID, HEmployee\_ID, Hotel\_ID,  
SEmployee\_ID,Customer\_ID)

VALUES

('R101', '2023-01-01', '2023-01-05', '2022-12-26', 0.1 , 587159533, 'M1011', 21110,  
'E011', 6111045842),  
(R113', '2023-01-01', '2023-01-05', '2022-12-26', 0.1 , 587159534, 'M1011', 21110,  
'E011', 6111045842),  
(R102', '2023-05-11', '2023-05-15', '2022-04-26', NULL , 587159598, 'M1011', 21110,  
'E011', 2654321098),  
(R103', '2023-08-01', '2023-08-02', '2022-07-28', NULL, 587159965, 'M1011', 21110,  
'E011', 5785432109),  
(R104', '2023-02-10', '2023-02-15', '2022-12-27', 0.15, 369852147, 'M1041', 21413,  
'E012', 6522547126),  
(R105', '2023-03-20', '2023-03-25', '2022-02-28', 0.05, 103245698, 'M1052', 21514,  
'E013', 6233146325),  
(R106', '2023-04-05', '2023-04-10', '2022-03-29', 0.1, 874198941,'M1021', 21211,  
'E014', 6144798102),  
(R107', '2023-05-15', '2023-05-20', '2022-04-30', 0.12, 554698745, 'M1100', 22019,  
'E015', 6275246101),  
(R118', '2023-06-01', '2023-06-05', '2023-04-05', 0.1, 456789365, 'M1194', 22524,  
'E016', 6096421362),  
(R117', '2023-07-10', '2023-07-15', '2023-05-10', 0.1, 789654153, 'M1031', 21312,  
'E017', 6747101023),  
(R108', '2023-08-20', '2023-08-25', '2023-05-15', 0.1, 789654123, 'M1031', 21312,  
'E017', 9876543210),  
(R109', '2023-09-05', '2023-09-10', '2023-07-20', 0.05, 897456123, 'M1111', 22221,  
'E019', 6568278431),  
(R110', '2023-10-15', '2023-10-20', '2023-08-25', 0.15, 258741236, 'M1062', 21615,  
'E020', 6449112235),  
(R111', '2023-10-17', '2023-10-20', '2023-09-25', 0.15, 987456313, 'M1165', 22625,  
'E020', 6320001268),

('R114', '2023-10-17', '2023-10-20', '2023-09-25', 0.15, 312398745, 'M1165', 22625, 'E020', 6320001268),  
('R112', '2023-10-20', '2023-10-23', '2023-08-30', 0.1, 227541258, 'M1166', 227, 'E019', 1967654321),  
('R115', '2023-10-20', '2023-10-23', '2023-08-30', 0.1, 789684523, 'M1166', 227, 'E019', 1967654321);

## **INVOICE**

INSERT INTO INVOICE (Invoice\_ID, Invoice\_Issued, Customer\_ID, Reservation\_ID)

VALUES

('INV821', '2022-12-26 10:30:00', 6111045842, 'R101'),  
('INV822', '2022-04-26 11:45:00', 2654321098, 'R102'),  
('INV823', '2022-07-28 09:15:00', 5785432109, 'R103'),  
('INV824', '2022-12-27 14:20:00', 6522547126, 'R104'),  
('INV825', '2022-02-28 12:00:00', 6233146325, 'R105'),  
('INV826', '2022-03-29 08:45:00', 6144798102, 'R106'),  
('INV827', '2022-04-30 16:30:00', 6275246101, 'R107'),  
('INV828', '2023-05-15 13:10:00', 9876543210, 'R108'),  
('INV829', '2023-07-20 10:00:00', 6568278431, 'R109'),  
('INV830', '2023-08-25 09:30:00', 6449112235, 'R110'),  
('INV831', '2023-09-25 07:30:00', 6320001268, 'R111'),  
('INV832', '2023-08-30 19:25:00', 1967654321, 'R112'),  
('INV833', '2022-12-26 21:52:00', 6111045842, 'R113'),  
('INV834', '2023-09-25 14:39:00', 6320001268, 'R114'),  
('INV835', '2023-08-30 18:10:00', 1967654321, 'R115'),  
('INV836', '2023-05-10 08:57:00', 6747101023, 'R117'),  
('INV837', '2023-04-05 11:23:00', 6096421362, 'R118');

## **PAYMENT**

INSERT INTO PAYMENT (Payment\_No, Payment\_Method, Time\_paid,

Customer\_ID, Invoice\_ID)

**VALUES**

(911, 'Cash', '2023-01-05 12:30:00', 6111045842, 'INV821'),  
(912, 'Credit Card', '2023-02-15 14:45:00', 2654321098, 'INV822'),  
(913, 'Online Transfer', '2023-03-25 10:15:00', 5785432109, 'INV823'),  
(914, 'Cash', '2023-04-10 15:20:00', 6522547126, 'INV824'),  
(915, 'Credit Card', '2023-05-20 13:00:00', 6233146325, 'INV825'),  
(916, 'Online Transfer', '2023-06-05 09:45:00', 6144798102, 'INV826'),  
(917, 'Cash', '2023-07-15 17:30:00', 6275246101, 'INV827'),  
(918, 'Credit Card', '2023-08-25 14:10:00', 9876543210, 'INV828'),  
(919, 'Online Transfer', '2023-09-10 11:00:00', 6568278431, 'INV829'),  
(920, 'Cash', '2023-10-20 10:30:00', 6449112235, 'INV830'),  
(921, 'Online Transfer', '2023-10-20 12:15:00', 6320001268, 'INV831'),  
(922, 'Cash', '2023-10-23 11:30:00', 1967654321, 'INV832'),  
(923, 'Cash', '2023-01-05 21:15:00', 6111045842, 'INV833'),  
(924, 'Online Transfer', '2023-10-20 20:30:00', 6320001268, 'INV834'),  
(925, 'Credit Card', '2023-10-23 10:45:00', 1967654321, 'INV835'),  
(926, 'Cash', '2023-07-15 19:30:00', 6747101023, 'INV836'),  
(927, 'Credit Card', '2023-06-05 12:15:00', 6096421362, 'INV837');

**ROOM RATING**

INSERT INTO ROOM\_RATING (Customer\_ID, Room\_ID, Room\_Rating,

Rating\_Description)

**VALUES**

(6111045842, 587159533, 5, 'Absolutely loved the oceanfront room!'),  
(6522547126, 369852147, 4, 'Great room, comfortable with an excellent city view'),  
(6233146325, 103245698, 5, 'The suite was fantastic, loved the mountain view'),  
(6144798102, 874198941, 3, 'The luxury room was good, but the pool view could be better'),  
(6275246101, 554698745, 4, 'Executive room offered an impressive skyline view'),  
(6096421362, 456789365, 2, 'Charming room, but the garden view was disappointing'),

(6747101023, 789654153, 3, 'Modern room, but not as expected'),  
(6568278431, 897456123, 5, 'Classic room with a serene courtyard view'),  
(6449112235, 258741236, 4, 'Deluxe room provided a stunning lake view'),  
(6320001268, 987456313, 5, 'Modern room but it is quite noisy'),  
(6320001268, 312398745, 5, 'Elegant suite with a mesmerizing beach view');

## **SERVICE RESERVED**

INSERT INTO SERVICE\_RESERVED (Service\_ID, Reservation\_ID)

VALUES

('S51121', 'R101'),  
('S51241', 'R102'),  
('S51332', 'R103'),  
('S51557', 'R104'),  
('S51610', 'R105'),  
('S51727', 'R106'),  
('S51121', 'R107'),  
('S51241', 'R108'),  
('S51332', 'R109'),  
('S51557', 'R110');

## **ROOM RESERVED**

INSERT INTO ROOM\_RESERVED (Room\_ID, Reservation\_ID)

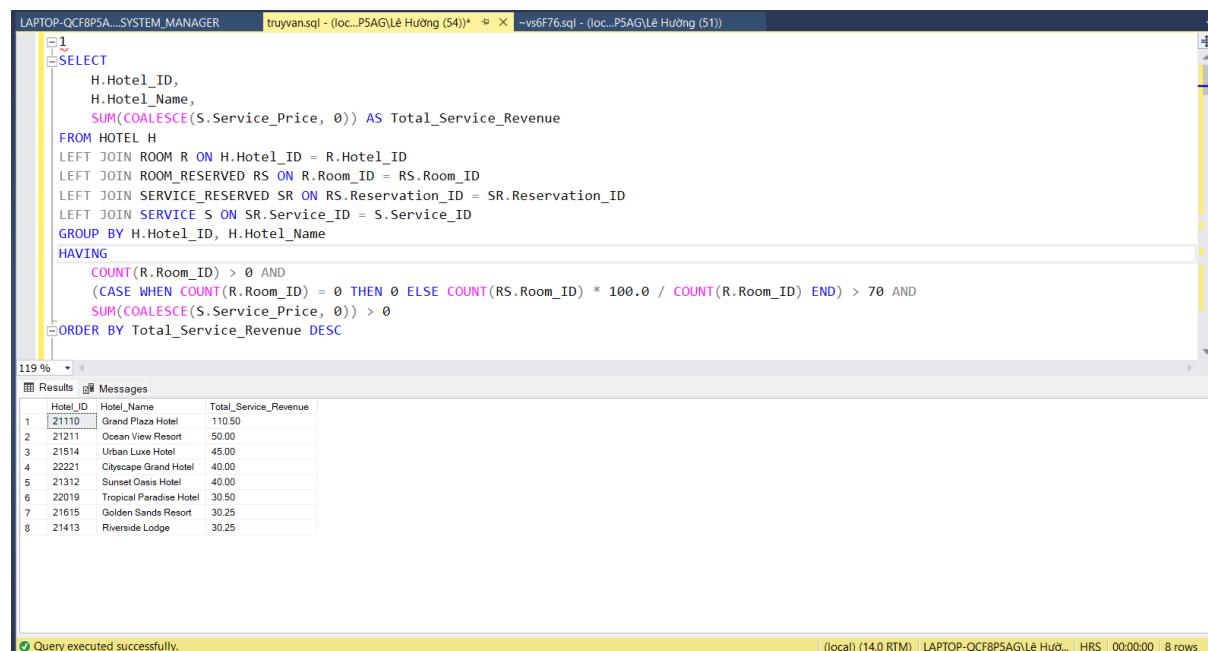
VALUES

(587159533, 'R101'),  
(587159534, 'R113'),  
(587159598, 'R102'),  
(587159965, 'R103'),  
(369852147, 'R104'),  
(103245698, 'R105'),  
(874198941, 'R106'),  
(554698745, 'R107'),

```
(456789365, 'R118'),
(789654153, 'R117'),
(789654123, 'R108'),
(897456123, 'R109'),
(258741236, 'R110'),
(987456313, 'R111'),
(312398745, 'R114'),
(227541258, 'R112'),
(789684523, 'R115');
```

## PART VI: PERFORMING SOME QUERIES

1. Write a query to display Hotel\_ID, Hotel\_Name and the column which calculates total service revenue for each hotel, counting only from hotels with a booked room rate higher than 70%, sorted by total service revenue in descending order



```
LAPTOP-QCF8P5A...SYSTEM_MANAGER [truyvan.sql - (loc..P5AG\Le Huong (54)) * X ~vs6F76.sql - (loc..P5AG\Le Huong (51))]

SELECT
    H.Hotel_ID,
    H.Hotel_Name,
    SUM(COALESCE(S.Service_Price, 0)) AS Total_Service_Revenue
FROM HOTEL H
LEFT JOIN ROOM R ON H.Hotel_ID = R.Hotel_ID
LEFT JOIN ROOM_RESERVED RS ON R.Room_ID = RS.Room_ID
LEFT JOIN SERVICE_RESERVED SR ON RS.Reservation_ID = SR.Reservation_ID
LEFT JOIN SERVICE S ON SR.Service_ID = S.Service_ID
GROUP BY H.Hotel_ID, H.Hotel_Name
HAVING
    COUNT(R.Room_ID) > 0 AND
    (CASE WHEN COUNT(R.Room_ID) = 0 THEN 0 ELSE COUNT(RS.Room_ID) * 100.0 / COUNT(R.Room_ID) END) > 70 AND
    SUM(COALESCE(S.Service_Price, 0)) > 0
ORDER BY Total_Service_Revenue DESC

119 % ▾ Results Messages
Hotel_ID Hotel_Name Total_Service_Revenue
1 21110 Grand Plaza Hotel 110.50
2 21211 Ocean View Resort 50.00
3 21514 Urban Luxe Hotel 45.00
4 22221 Cityscape Grand Hotel 40.00
5 21312 Sunset Oasis Hotel 40.00
6 22019 Tropical Paradise Hotel 30.50
7 21615 Golden Sands Resort 30.25
8 21413 Riverside Lodge 30.25

Query executed successfully. (local) (14.0 RTM) LAPTOP-QCF8P5AG\Le Hu... | HRS | 00:00:00 | 8 rows
```

2. Write a query to display all customers (Customer\_ID, First\_Name, Last\_Name) and the number of bookings made for each customer, sorted by number of bookings in descending order

LAPTOP-QCF8PSA...SYSTEM\_MANAGER truyvan.sql - (loc...P5AG\Lê Hường (54)\* -vs6F76.sql - (loc...P5AG\Lê Hường (51))

```

2
SELECT
    C.Customer_ID,
    C.First_Name, C.Last_Name,
    COUNT(R.Reservation_ID) AS Total_Reservations
FROM CUSTOMER C
LEFT JOIN RESERVATION R ON C.Customer_ID = R.Customer_ID
GROUP BY C.Customer_ID, C.First_Name, C.Last_Name
ORDER BY Total_Reservations DESC

```

Results Messages

Customer_ID	First_Name	Last_Name	Total_Reservations
1 1987654321	Ava	Taylor	2
2 6111045842	John	Doe	2
3 6320001268	Olivia	White	2
4 6449112235	Frank	Taylor	1
5 6522547126	Jane	Smith	1
6 6568278431	Sophia	Brown	1
7 6747101023	Daniel	Miller	1
8 9876543210	Mason	Williams	1
9 6144798102	Alice	Williams	1
10 6233146325	Bob	Johnson	1
11 6275246101	Charlie	Davis	1
12 2654321098	Liam	Johnson	1
13 5785432109	Emma	Anderson	1
14 6096421362	Eva	Anderson	1

3. Write a query to display the information of Hotel where Hotel\_Manager is working

LAPTOP-QCF8PSA...SYSTEM\_MANAGER truyvan.sql - (loc...P5AG\Lê Hường (54)\* -vs6F76.sql - (loc...P5AG\Lê Hường (51))

```

-- Hiển thị Hotel Manager của từng Hotel
SELECT
    H.Hotel_ID,
    H.Hotel_Name,
    HM.Employee_ID AS Hotel_Manager_ID,
    (HM.First_Name + ' ' + HM.Last_Name) AS Hotel_Manager_Name
FROM
    HOTEL H
JOIN
    HOTEL_MANAGER HM ON H.Hotel_ID = HM.Hotel_ID

```

Results Messages

Hotel_ID	Hotel_Name	Hotel_Manager_ID	Hotel_Manager_Name
1 21110	Grand Plaza Hotel	M1011	John Smith
2 21211	Ocean View Resort	M1021	Alice Johnson
3 21312	Sunset Oasis Hotel	M1031	Bob Miller
4 21413	Riverside Lodge	M1041	Eva Brown
5 21514	Urban Luxe Hotel	M1052	Charlie Davis
6 21615	Golden Sands Resort	M1062	Sophia Taylor
7 21716	Skyline Suites	M1074	Daniel Anderson
8 21817	Central View Hotel	M1087	Olivia White
9 21918	Desert Oasis Resort	M1092	Frank Jones
10 22019	Tropical Paradise Hotel	M1100	Emma Williams
11 22221	Cityscape Grand Hotel	M1111	Liam Johnson
12 22322	Polar Lights Resort	M1132	Emma Anderson
13 22423	Bourbon Street Inn	M1153	Mason Williams
14 22625	Red Rock Resort	M1165	Noah Davis

Query executed successfully.

4. Write a query to display Reservation\_ID, Reservation\_Date, Room\_ID, Hotel\_ID about Reservations monitored by System Manager with 'T' in the first name. Also display System Manager ID and their names.

```

LAPTOP-QCF8PSA\SYSTEM_MANAGER truyvan.sql - (loc...)PSAG\Lê Hường (54)* -vs6F76.sql - (loc...)PSAG\Lê Hường (51)
-- Hiển thị Reservation được System Manager có chữ 'T' trong first name giám sát
SELECT
    R.Reservation_ID, R.Reservation_Date,
    R.Room_ID,
    R.Hotel_ID,
    R.SEmployee_ID,
    R.Customer_ID,
    (SM.First_Name+' '+SM.Last_Name) as SystemManager_Name
FROM
    RESERVATION R
JOIN
    SYSTEM_MANAGER SM ON R.SEmployee_ID = SM.SEmployee_ID
WHERE
    SM.First_Name LIKE '%a%';

```

144 % Results Messages

	Reservation_ID	Reservation_Date	Room_ID	Hotel_ID	SEmployee_ID	Customer_ID	SystemManager_Name
1	R104	2022-12-27	369852147	21413	E012	6522547126	Alice Johnson
2	R105	2022-02-28	103245698	21514	E013	6233146325	David Lee
3	R107	2022-04-30	554698745	22019	E015	6275246101	Michael Brown
4	R108	2023-05-15	789654123	21312	E017	9876543210	Daniel Nguyen
5	R109	2023-07-20	897456123	22221	E019	6568278431	James Garcia
6	R110	2023-08-25	258741236	21615	E020	6449112235	Emma Taylor
7	R111	2023-09-25	987456313	22625	E020	6320001268	Emma Taylor
8	R112	2023-08-30	227541258	227	E019	1967654321	James Garcia
9	R114	2023-09-25	312398745	22625	E020	6320001268	Emma Taylor
10	R115	2023-08-30	789645453	227	E019	1967654321	James Garcia
11	R117	2023-05-10	789654153	21312	E017	6747101023	Daniel Nguyen
12	R118	2023-04-05	456789365	22524	E016	6096421362	Sophia Davis

Query executed successfully. (local) (14.0 RTM) | LAPTOP-QCF8PSAG\Lê Hường | HRS 00:00:00 | 12 rows

5. Write a query to display all rooms are higher than average priced and highly rated.

```

truyvan.sql - (loc...)PSAG\Lê Hường (54)* -vs6F76.sql - (loc...)PSAG\Lê Hường (51)
SELECT Room_ID FROM ROOM WHERE CATEGORY_CODE IN (
    SELECT Category_Code FROM ROOM_CATEGORY
    WHERE Room_Price > (SELECT AVG(Room_Price) FROM ROOM_CATEGORY))
INTERSECT
SELECT Room_ID FROM ROOM_RATING
WHERE Room_Rating > 4;

```

144 % Results Messages

	Room_ID
1	103245698

Query executed successfully. (local) (14.0 RTM) | LAPTOP-QCF8PSAG\Lê Hường | HRS 00:00:00 | 1 rows

6. Write a query to display the Customer\_ID, name, Room\_ID, Room\_No, Category\_Name for all customers who have booked rooms with a capacity of 4 or more.

truyvan.sql - (loc...P5AG\Lê Hướng (54))\* -vs6F76.sql - (loc...P5AG\Lê Hướng (51))

```

SELECT
    C.Customer_ID,
    (C.First_Name + ' ' + C.Last_Name) AS Customer_Name,
    R.Room_ID, R.Room_No,
    RC.Category_Name,
    RC.Capacity
FROM
    CUSTOMER C
    JOIN RESERVATION RS ON C.Customer_ID = RS.Customer_ID
    JOIN ROOM R ON RS.Room_ID = R.Room_ID
    JOIN ROOM_CATEGORY RC ON R.Category_Code = RC.Category_Code
WHERE
    RC.Capacity >= 4

```

144 %

Results Messages

	Customer_ID	Customer_Name	Room_ID	Room_No	Category_Name	Capacity
1	2654321098	Liam Johnson	587159598	1102	Penthouse	6
2	5785432109	Emma Anderson	587159965	908	Deluxe Suite	4
3	6522547126	Jas Smith	36985147	202	Penthouse	6
4	6233146326	Bob Johnson	103245698	302	Presidential Suite	6
5	6275246101	Charlie Davis	554699745	502	Presidential Suite	6
6	9876543210	Mason Williams	789654123	201	Deluxe Suite	4
7	6320001268	Olivia White	987456313	401	Deluxe Suite	4
8	1967654321	Ava Taylor	227541256	1021	Presidential Suite	6
9	6320001268	Olivia White	312398745	9021	Deluxe Suite	4
10	1967654321	Ava Taylor	789684523	503	Penthouse	6
11	6747101023	Daniel Miller	789654153	610	Deluxe Suite	4
12	6096421362	Eva Anderson	456789365	1012	Penthouse	6

Query executed successfully. (local) (14.0 RTM) LAPTOP-QCF8P5AG\Lê Huờ... HRS 00:00:00 12 rows

7. Write a query to display Customer\_ID, Customers' name, Gender, Reservation\_ID of customers who have booked rooms but have not used any services.

--Liệt kê các khách hàng đã đặt phòng nhưng chưa sử dụng dịch vụ nào.

```

SELECT C.Customer_ID,
    (C.First_Name+ ' '+ C.Last_Name) as Customer_Name,
    C.Gender, R.Reservation_ID
FROM CUSTOMER C
JOIN RESERVATION R ON C.Customer_ID = R.Customer_ID
LEFT JOIN SERVICE_SELECTION SS ON C.Customer_ID = SS.Customer_ID
WHERE SS.Service_ID IS NULL;

```

192 %

Results Messages

	Customer_ID	Customer_Name	Gender	Reservation_ID
1	6320001268	Olivia White	Female	R111
2	1967654321	Ava Taylor	Female	R112
3	6320001268	Olivia White	Female	R114
4	1967654321	Ava Taylor	Female	R115
5	6747101023	Daniel Miller	Male	R117
6	6096421362	Eva Anderson	Female	R118

Query executed successfully. (local) (14.0 RTM) LAPTOP-QCF8P5AG\Lê Huờ... HRS 00:00:00 6 rows

8. Write a query to display Customer\_ID, Customers' name, Reservation\_ID, Invoice\_ID, Last\_Price of customers who booked rooms in the first 3 months of the year and the total amount they had to pay, sorted by total amount in descending order.

--sắp xếp theo tổng số tiền giảm dần.

```

SELECT C.Customer_ID,
       (C.First_Name + ' ' + C.Last_Name) AS Customer_Name,
       R.Reservation_ID, I.Invoice_ID, R.CHECK_IN_DATE,
       SUM(I.LAST_PRICE) AS Total_Last_Price
  FROM CUSTOMER C
 JOIN INVOICE I ON C.Customer_ID = I.Customer_ID
 JOIN RESERVATION R ON I.Reservation_ID=R.Reservation_ID
 WHERE MONTH(CHECK_IN_DATE) BETWEEN 1 AND 3
 GROUP BY C.Customer_ID, C.First_Name,C.Last_Name,I.Invoice_ID,
          R.CHECK_IN_DATE, R.Reservation_ID
 ORDER BY Total_Last_Price DESC;

```

192 %

Results Messages

Customer_ID	Customer_Name	Reservation_ID	Invoice_ID	CHECK_IN_DATE	Total_Last_Price	
1	6233146325	Bob Johnson	R105	INV825	2023-03-20	4792.75
2	6522547126	Jane Smith	R104	INV824	2023-02-10	2154.11
3	6111045842	John Doe	R113	INV833	2023-01-01	541.80
4	6111045842	John Doe	R101	INV821	2023-01-01	387.45

Query executed successfully. (local) (14.0 RTM) | LAPTOP-QCF8P5AG\Le Huong | HRS | 00:00:00 | 4 rows

9. Write a query to display Customer\_ID, Customers' name, information about room and total number of days booked for each room, sorted by total number of days in descending order.

```

SELECT C.Customer_ID,
       CONCAT(C.First_Name, ' ', C.Last_Name) AS Customer_Name,
       R.Room_ID,R.Room_No,R.Description,
       DATEDIFF(DAY, RS.Check_in_Date, RS.Check_out_Date) AS Days_Booked
  FROM CUSTOMER C
 JOIN RESERVATION RS ON C.Customer_ID = RS.Customer_ID
 JOIN ROOM R ON RS.Room_ID=R.Room_ID
 ORDER BY Days_Booked DESC

```

192 %

Results Messages

Customer_ID	Customer_Name	Room_ID	Room_No	Description	Days_Booked	
1	6522547126	Jane Smith	369852147	202	Luxury room with pool view	5
2	6233146325	Bob Johnson	103245698	302	Executive room with skyline view	5
3	6144798102	Alice Williams	874198941	103	Cozy room with city view	5
4	6275246102	Charlie Davis	554698745	502	Elegant suite with beach view	5
5	9876543210	Mason Williams	789654123	201	Spacious suite with mountain view	5
6	6568278431	Sophia Brown	897456123	302	Charming room with garden view	5
7	6449112233	Frank Taylor	258741236	403	Modem room with street view	5
8	6747101023	Daniel Miller	789654153	610	Artistic room with mountain backdrop	5
9	6096421362	Eva Anderson	456789365	1012	Executive suite with private balcony	4
10	6111045842	John Doe	587159533	102	Oceanfront room with balcony	4
11	2654321098	Liam Johnson	587159598	1102	Romantic suite with sunset view	4

Query executed successfully. (local) (14.0 RTM) | LAPTOP-QCF8P5AG\Le Huong | HRS | 00:00:00 | 17 rows

10. Write a query to display the Customer\_ID, name, total amount paid, and the CustomerLevel column with titles VIP (>5000), GOLD(>2000), SILVER(>1000), and BRONZE respectively for those customers whose total amount price paid is. Sort the result by total amount paid in descending order.

```

truyvan.sql - (loc...P5AG\Le Huong (54))* | -vs6F76.sql - (loc...P5AG\Le Huong (51))

--WITH CustomerSpending AS (
SELECT Customer_ID, Last_Price AS TotalAmountPaid
FROM INVOICE
GROUP BY Customer_ID, Last_Price)

SELECT
C.Customer_ID,
(C.First_Name + ' ' + C.Last_Name) AS Customer_Name,
CS.TotalAmountPaid,
(SELECT CASE
WHEN CS.TotalAmountPaid > 5000 THEN 'VIP'
WHEN CS.TotalAmountPaid > 2000 THEN 'Gold'
WHEN CS.TotalAmountPaid > 1000 THEN 'Silver'
ELSE 'Bronze'
END) AS CustomerLevel
FROM
CUSTOMER C
JOIN CustomerSpending CS ON C.Customer_ID = CS.Customer_ID
ORDER BY CS.TotalAmountPaid DESC;

```

99 %

Results Messages

	Customer_ID	Customer_Name	TotalAmountPaid	CustomerLevel
1	6233146325	Bob Johnson	4792.75	Gold
2	6275246101	Charlie Davis	4426.84	Gold
3	1967654321	Ava Taylor	2700.00	Gold
4	6522547125	Jane Smith	2154.11	Gold
5	2654321098	Liam Johnson	2043.20	Gold
6	6096421383	Eva Anderson	1802.88	Silver
7	1967654321	Ava Taylor	1352.16	Silver
8	9876543210	Mason Williams	1164.38	Silver
9	6747101023	Daniel Miller	1128.38	Silver
10	6144798102	Alice Williams	722.25	Bronze
11	6449112235	Frank Taylor	665.34	Bronze
12	6320001268	Olivia White	639.41	Bronze
13	6111045842	John Doe	541.80	Bronze
14	6568278431	Sophia Brown	513.00	Bronze

11. Write a query to display the payment methods most chosen by customers are ranked from highest to lowest and the percentage of each method.

```

LAPTOP-QCF8P5A\SYSTEM_MANAGER truyvan.sql - (loc...P5AG\Le Huong (54))* | -vs6F76.sql - (loc...P5AG\Le Huong (51))

SELECT
Payment_Method,
COUNT(Payment_No) AS Total_Payment_Amount,
CAST(COUNT(Payment_No) * 100.0 / SUM(COUNT(Payment_No)) OVER () AS DECIMAL(5,2)) AS Payment_Percentage,
RANK() OVER (ORDER BY COUNT(Payment_No) DESC) AS Payment_Rank
FROM PAYMENT
GROUP BY Payment_Method
ORDER BY Payment_Rank;

```

131 %

Results Messages

	Payment_Method	Total_Payment_Amount	Payment_Percentage	Payment_Rank
1	Cash	7	41.18	1
2	Credit Card	5	29.41	2
3	Online Transfer	5	29.41	2

Query executed successfully. (local) (14.0 RTM) LAPTOP-QCF8P5A\Le Huo... HRS 00:00:00 3 rows