**DATABASE MANAGEMENT SYSTEMS PROJECT:**

**PROJECT 9**

**HOTEL MANAGEMENT SYSTEM**

**BY**

**SIDDHARTH S SHAH**

**(2021A7PS2428P)**

**RIJUL BASSAMBOO**

**(2021A7PS0009P)**

**The Enitity-Relationship Model is as follows:**

We have listed a few assumptions as well for better understanding.

Diagram

Description automatically generated

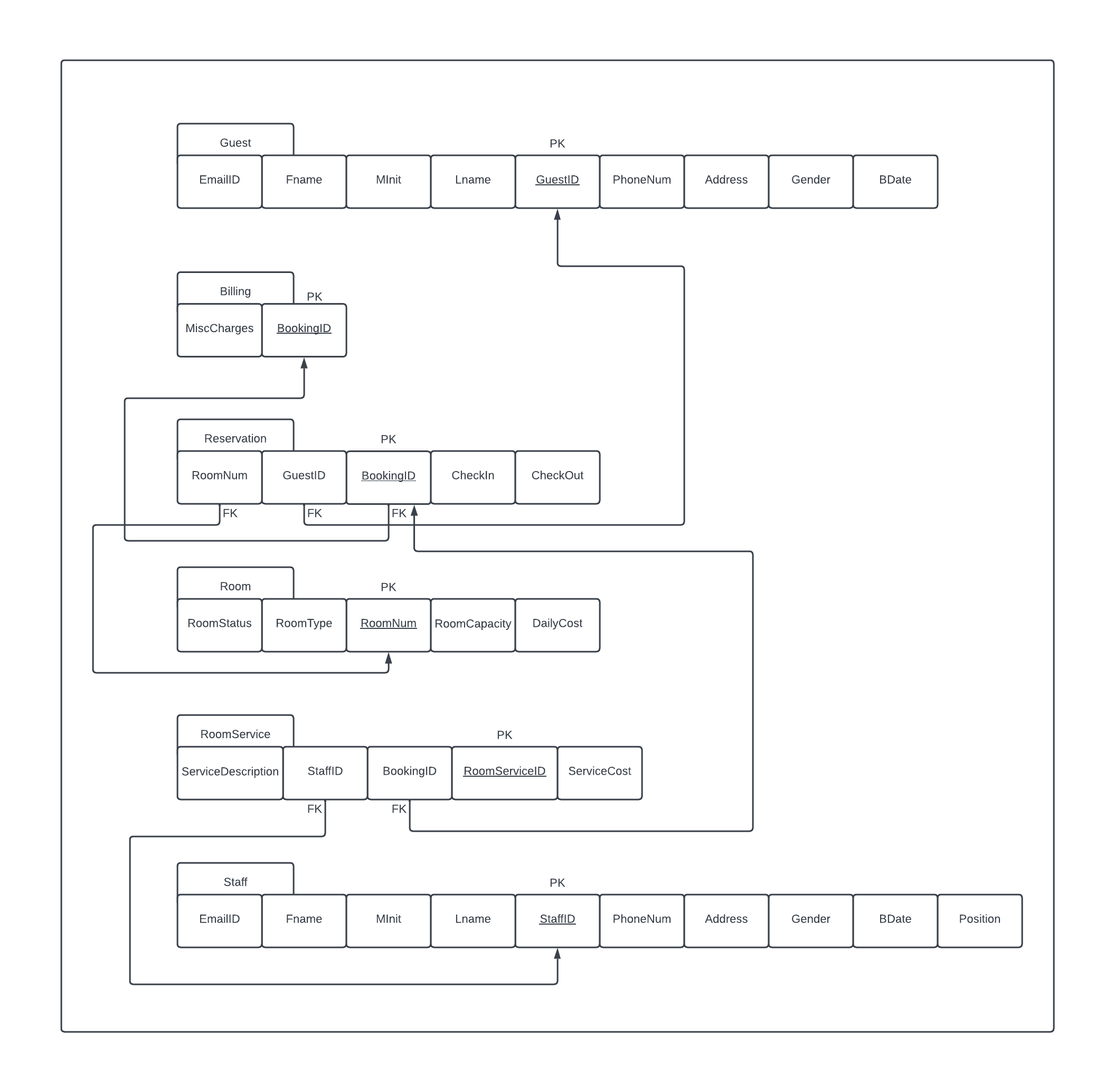
**Explanations and Assumptions:**

1. A reservation will be booked under a single guest, like in a typical hotel. If a guest needs to book multiple rooms simultaneously, he will have to do it under different reservations.
2. Room capacity is a derived attribute of Room type.
3. For the notation used to describe the cardinality of the ternary relationship Reservation, please refer to the link below:

<https://stackoverflow.com/questions/18365870/er-diagram-ternary-relationship-how-to-read-properly>

1. The bill will be generated only after checkout.
2. For allotting staff to a room, we have decided to go with a more realistic approach. We will be assigning random staff for a particular room service.
3. If the staff member leaves the hotel, then position becomes NULL.
4. For names of tables, views and columns, we have used First letter capitalised camel casing, however in mySQL , since it is case insensitive, we haven’t used any such casing.
5. For names of Procedures, Functions and Triggers we have used the special character ‘\_’, which separates the words inside the name. Eg: “Check\_out\_generate\_bill”
6. For revenue calculation, only the rooms which have been checked-out have been considered.

**The initial Relational Diagram is as follows:**



**The following are the constraints that need to be taken care of:**

* **Table Guest:**  
  EmailID varchar(50) NOT NULL,  
  Fname varchar(50) NOT NULL,  
  MInit char,  
  Lname varchar(50) NOT NULL,  
  GuestID int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,  
  PhoneNum numeric constraint max\_length CHECK (length(PhoneNum)=10) NOT NULL,  
  Gender char NOT NULL,  
  BDate date,  
  Address varchar(255) NOT NULL
* **Table Staff:**  
  EmailID varchar(50) NOT NULL,  
  Fname varchar(50) NOT NULL,  
  MInit char,  
  Lname varchar(50) NOT NULL,  
  StaffID int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,  
  PhoneNum numeric constraint max\_length CHECK (length(PhoneNum)=10) NOT NULL,  
  Gender char NOT NULL,  
  BDate date,  
  Address varchar(255) NOT NULL,  
  Position varchar(30) NOT NULL
* **Table Billing:**

MiscCharges int(20),

BookingID int unsigned NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (BookingID)

* **Table Room:**  
  RoomStatus bool default false,  
  RoomType varchar(20),  
  RoomNum int(5),

DailyCost int(10) NOT NULL,  
RoomCapacity int (2),  
RoomType varchar(20),  
PRIMARY KEY (RoomNum)

* **Table Reservation:**

RoomNum int(5),

GuestID int unsigned,

BookingID int unsigned,

CheckIn TIMESTAMP NOT NULL default current\_timestamp,

CheckOut TIMESTAMP default NULL,

PRIMARY KEY (BookingID),

FOREIGN KEY (GuestID) REFERENCES Guest(GuestID),

FOREIGN KEY (RoomNum) REFERENCES Room(RoomNum)

FOREIGN KEY (BookingID) REFERENCES Billing(BookingID)

* **Table RoomService:**

ServiceDescription varchar(255) NOT NULL,

StaffID int unsigned,

BookingID int unsigned,

RoomServiceID int unsigned NOT NULL AUTO\_INCREMENT,

ServiceCost float (10,2),

PRIMARY KEY (RoomServiceID),

FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),

FOREIGN KEY (BookingID) REFERENCES Reservation\_Billing(BookingID)

Next page

**Step-1: Removal of Redundancy**

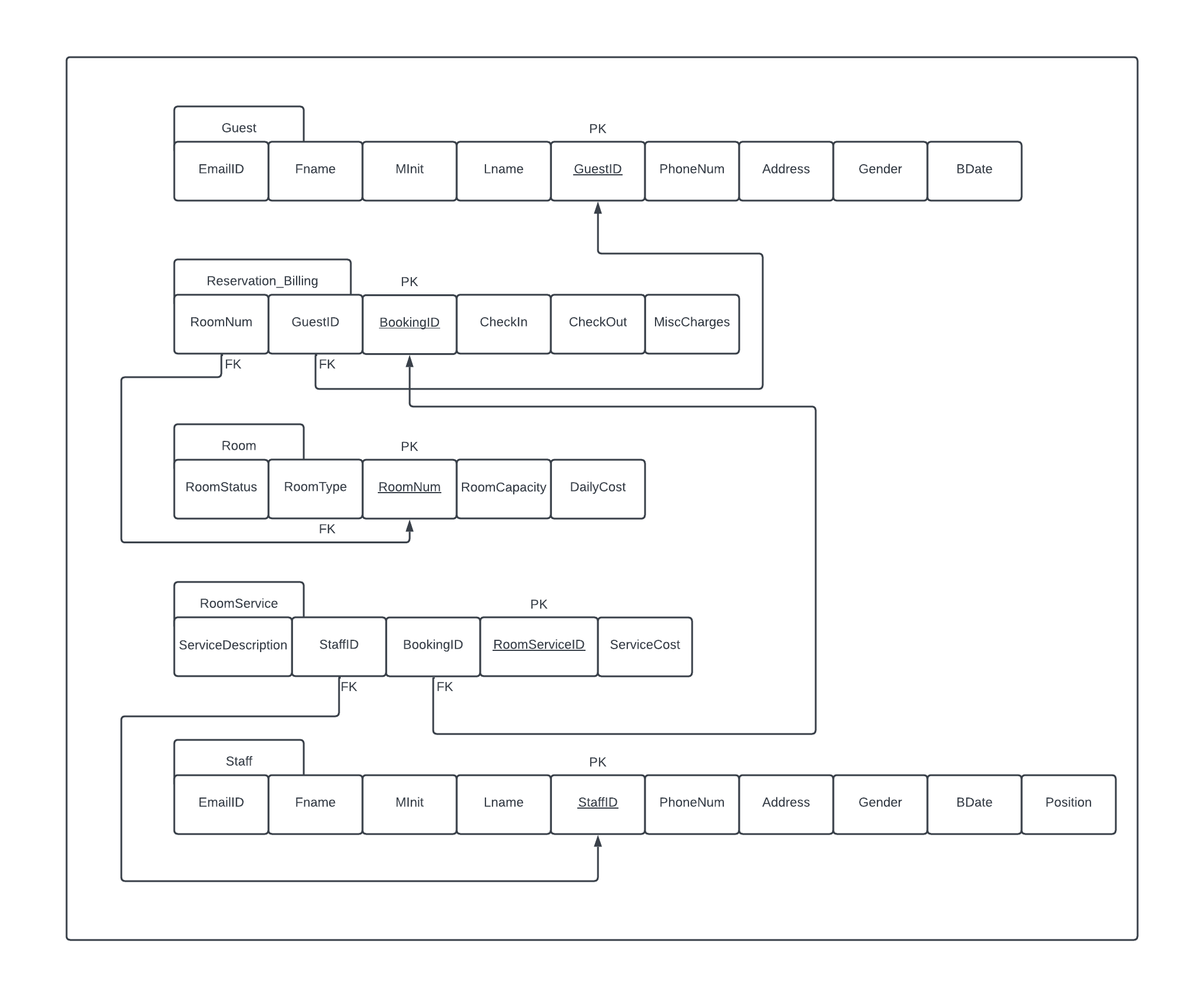
As the PK of both the Billing and Reservation Table is the same, we can merge the tables.

We will create separate views for them independently from this table.

The name of the new table will be Reservation\_Billing and the attributes will be RoomNum, GuestID, BookingID, CheckIn, CheckOut and MiscCharges.

Here is the updated Relational Diagram.

**Changes after Step-1:**



**Now, to check whether the Schema is in 1NF, 2NF and finally 3NF:**

**1NF:**

**Table GuestID:**

GuestID (PK) 🡪EmailID, Fname, Lname, MInit, PhoneNum, Address, Gender, BDate

**Table Reservation\_Billing:**

BookingID (PK) 🡪RoomNum(FK), GuestID(FK), CheckIn, CheckOut, MiscCharges

**Table Room:**

RoomNum (PK) 🡪RoomStatus, RoomType, RoomCapacity, DailyCost

**Table** **RoomService**:

RoomServiceID (PK) 🡪ServiceDescription, StaffID(FK), BookingID(FK), ServiceCost

**Table** **Staff**:

StaffID(PK) 🡪EmailID, Fname, Lname, MInit, PhoneNum, Address, Gender, BDate, Position

The given relational schema is already in 1NF as it does not have any multivalued and composite attributes.

Next Page

**2NF:** To check if the schema is in 2NF, let us first list down all the functional dependencies-

**Table GuestID:**

GuestID (PK) 🡪EmailID, Fname, Lname, MInit, PhoneNum, Address, Gender, BDate

**Table Reservation\_Billing:**

BookingID (PK) 🡪RoomNum(FK), GuestID(FK), CheckIn, CheckOut, MiscCharges

**Table Room:**

RoomNum (PK) 🡪RoomStatus, RoomType, RoomCapacity, DailyCost

RoomType 🡪RoomCapacity, DailyCost

**Table** **RoomService**:

RoomServiceID (PK) 🡪ServiceDescription, StaffID(FK), BookingID(FK), ServiceCost

**Table** **Staff**:

StaffID(PK) 🡪EmailID, Fname, Lname, MInit, PhoneNum, Address, Gender, BDate, Position

The given relational schema is already in 2NF as every non-prime attribute in the relations are fully functionally dependent on the primary key. Also, the primary key is composed of a single attribute.

**Next Page**

**3NF:** The relational schema is not in 3NF as the RoomType attribute, which is not a super key, is implying 2 non-prime attributes, namely RoomCapacity and DailyCost. Therefore, we need to decompose the table Room into two tables –

1. Room
2. RoomTypeDetails

The table Room will contain RoomNum (Primary Key), RoomType (which will be the Foreign Key for the table RoomTypeDetails) and RoomStatus.

The table RoomTypeDetails will contain RoomCapacity, RoomType (Primary Key) and DailyCost.

**The final functional dependencies are as follows:**

**Table GuestID:**

GuestID (PK) 🡪EmailID, Fname, Lname, MInit, PhoneNum, Address, Gender, BDate

**Table Reservation\_Billing:**

BookingID (PK) 🡪RoomNum(FK), GuestID(FK), CheckIn, CheckOut, MiscCharges

**Table Room:**

RoomNum (PK) 🡪RoomStatus, RoomType

**Table RoomTypeDetails:**

RoomType (PK) 🡪RoomCapacity, DailyCost

**Table** **RoomService**:

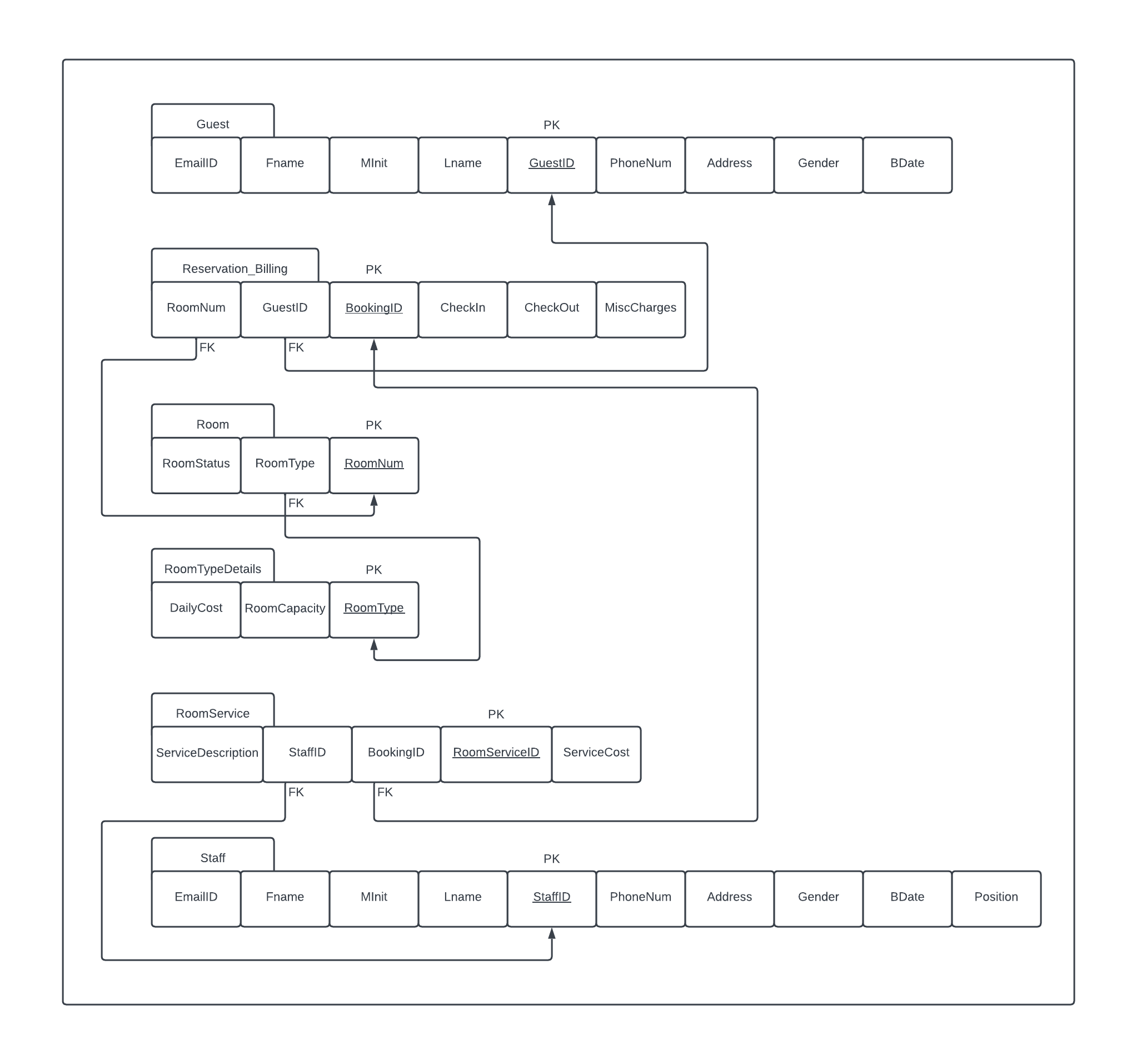
RoomServiceID (PK) 🡪ServiceDescription, StaffID(FK), BookingID(FK), ServiceCost

**Table** **Staff**:

StaffID(PK) 🡪EmailID, Fname, Lname, MInit, PhoneNum, Address, Gender, BDate, Position

Since, there are no transitive relations in the schema, hence the relational schema is in 3NF.

**Here is the final Relational Schema in 3NF.**



**The final constraints are as follows:**

* **Table Guest:**  
  EmailID varchar(50) NOT NULL,  
  Fname varchar(50) NOT NULL,  
  MInit char,  
  Lname varchar(50) NOT NULL,  
  GuestID int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,  
  PhoneNum numeric constraint max\_length CHECK (length(PhoneNum)=10) NOT NULL,  
  Gender char NOT NULL,  
  BDate date,  
  Address varchar(255) NOT NULL
* **Table Staff:**  
  EmailID varchar(50) NOT NULL,  
  Fname varchar(50) NOT NULL,  
  MInit char,  
  Lname varchar(50) NOT NULL,  
  StaffID int unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,  
  PhoneNum numeric constraint max\_length CHECK (length(PhoneNum)=10) NOT NULL,  
  Gender char NOT NULL,  
  BDate date,  
  Address varchar(255) NOT NULL,  
  Position varchar(30) NOT NULL
* **Table RoomTypeDetails:**DailyCost int(10) NOT NULL,  
  RoomCapacity int (2),  
  RoomType varchar(20) PRIMARY KEY
* **Table Room:**  
  RoomStatus bool default false,  
  RoomType varchar(20),  
  RoomNum int(5),  
  PRIMARY KEY (RoomNum),  
  FOREIGN KEY (RoomType) REFERENCES RoomTypeDetails(RoomType)
* **Table Reservation\_Billing:**

RoomNum int(5),

GuestID int unsigned,

BookingID int unsigned NOT NULL AUTO\_INCREMENT,

CheckIn TIMESTAMP NOT NULL default current\_timestamp,

CheckOut TIMESTAMP default NULL,

MiscCharges int(20),

PRIMARY KEY (BookingID),

FOREIGN KEY (GuestID) REFERENCES Guest(GuestID),

FOREIGN KEY (RoomNum) REFERENCES Room(RoomNum)

* **Table RoomService:**

ServiceDescription varchar(255) NOT NULL,

StaffID int unsigned,

BookingID int unsigned,

RoomServiceID int unsigned NOT NULL AUTO\_INCREMENT,

ServiceCost float(10,2),

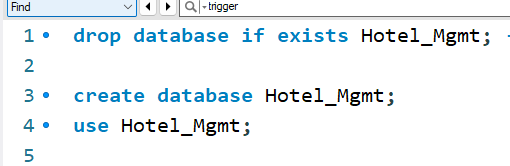
PRIMARY KEY (RoomServiceID),

FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),

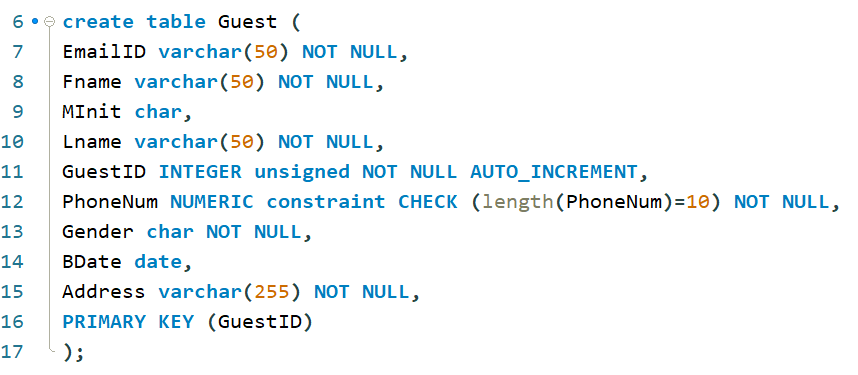
FOREIGN KEY (BookingID) REFERENCES Reservation\_Billing(BookingID)

**NOTE: Casing Convention:**

**SQL Queries and their Explanation:**



* drop command is used to drop any pre-existing database with the name “Hotel\_mgmt”
* create database and use db\_name are used to create and finally use the database “Hotel\_mgmt”

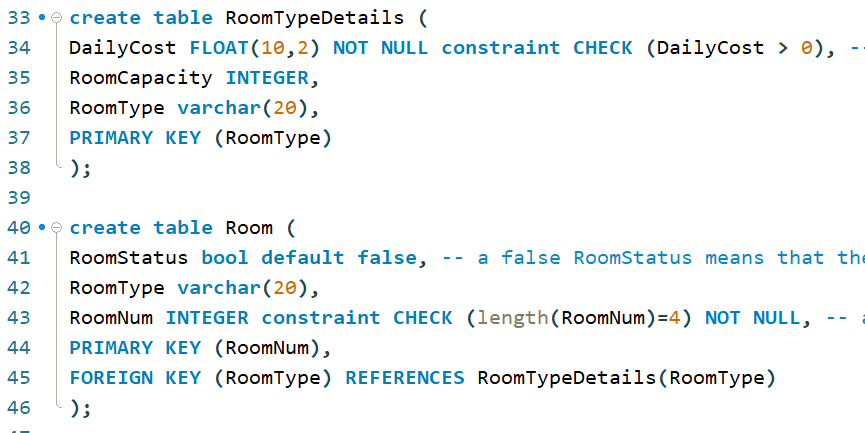


* create table command is used to make a table inside the database. Here the table with the tb\_name ‘guest’ is being made which has many attributes and their corresponding constraints.

Graphical user interface, text

Description automatically generated

* Here the table with the tb\_name ‘Staff’ is being made which has many attributes and their corresponding constraints.



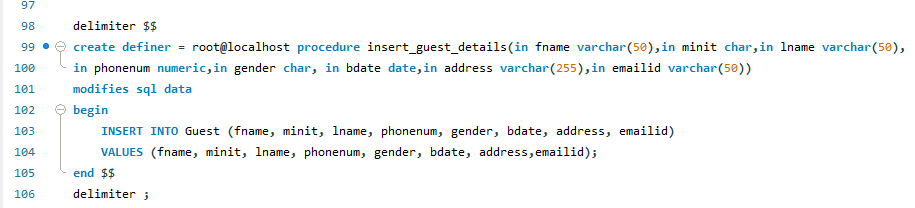
* Here the tables with the tb\_name ‘RoomTypeDetails’ and ‘Room’ are being made which have many attributes and their corresponding constraints.



* Here the tables with the tb\_name ‘Reservation\_Billing’ and ‘Room’ are being made which have many attributes and their corresponding constraints.



* Here we have made 4 views. Namely Reservation, Billing, EmptyRoomList and OccupiedRoomList.
* The Reservation view displays a few of the guest details, booking details and room details.
* The Billing view displays the bills generated per booking including room charges, room service charges, miscellaneous charges, and total charges.
* The EmptyRoomList view displays the list of the rooms which haven’t been occupied.
* The occupiedRoomList view displays the list of the rooms which have been occupied.



* The insert\_guest\_details procedure is used to enter the guest details into the Guest table.

Text

Description automatically generated

* The insert\_staff\_details procedure is used to enter the staff details into the Staff table.

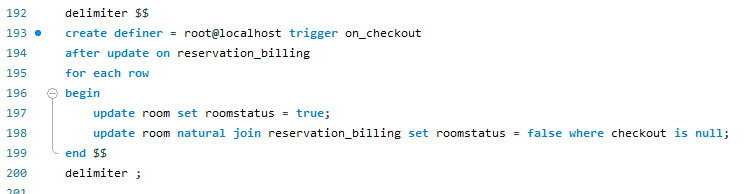


* The update\_guest\_details procedure is used to update the guest details. It checks if the passed parameter is null or not and then accordingly updates the passed parameters.

Graphical user interface, text, application, email

Description automatically generated

* The update\_staff\_details procedure is used to update the staff details. It checks if the passed parameter is null or not and then accordingly updates the passed parameters.



* The on\_checkout trigger updates the room status to false once the checkout is set to null. This happens after any updation to the reservation\_billing table.



* The check\_in\_reservation procedure does the check-in for the guests. It takes two parameters, the room number and the guest id.

Graphical user interface, text, application, email

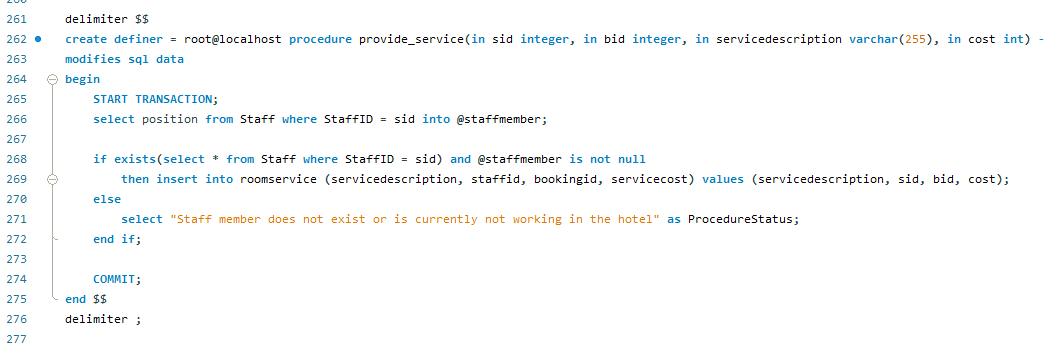
Description automatically generated

* The check\_out\_generate\_bill procedure is used to the generate the bill for the guest who has checked out of the hotel. It takes the booking id as a parameter.

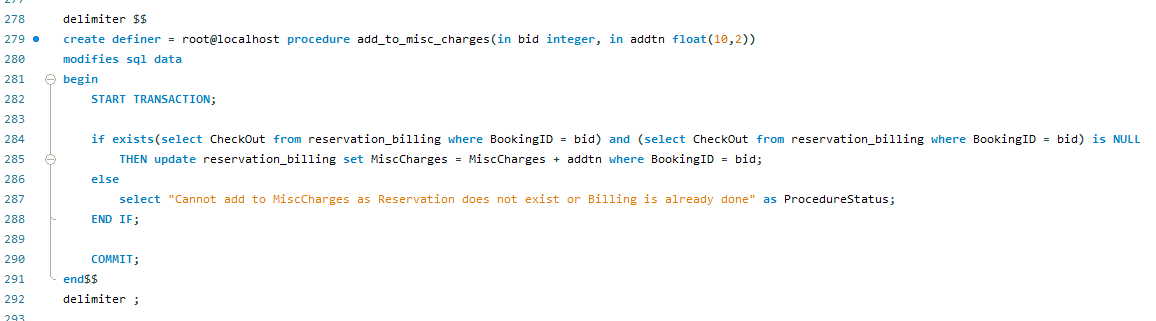
Graphical user interface, text, application

Description automatically generated

* The cancel\_booking procedure is used to cancel the booking of a guest. It takes booking id as a parameter.



* The provide\_service is a procedure which is used to add the records of any room service being provided to the guests. It takes the staff id, booking id, service description and the cost of the service as its parameters.

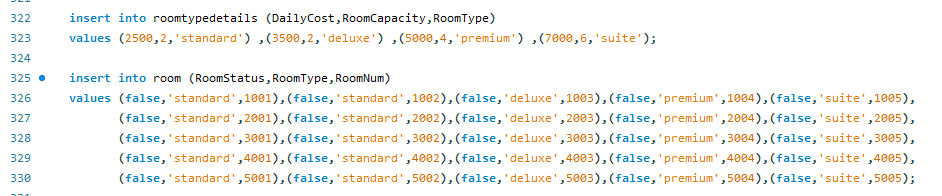


* The add\_to\_misc\_charges is the procedure used to add any kind of miscellaneous charge under the guest’s tab. It takes the booking id and the amount to be added (cost) as its parameters.

Graphical user interface, text, application

Description automatically generated

* The ADMIN\_viewemptyrooms procedure is used to view all the rooms which are empty.
* The ADMIN\_viewoccupiedrooms procedure is used to view all the rooms which are occupied.
* The ADMIN\_revenuefromtype procedure is used to derive the revenue generated from each room type.



* Inserting the room and roomtype details.

A picture containing table

Description automatically generated

* Inserting the guest details.

Table

Description automatically generated with medium confidence

* Inserting the staff details.

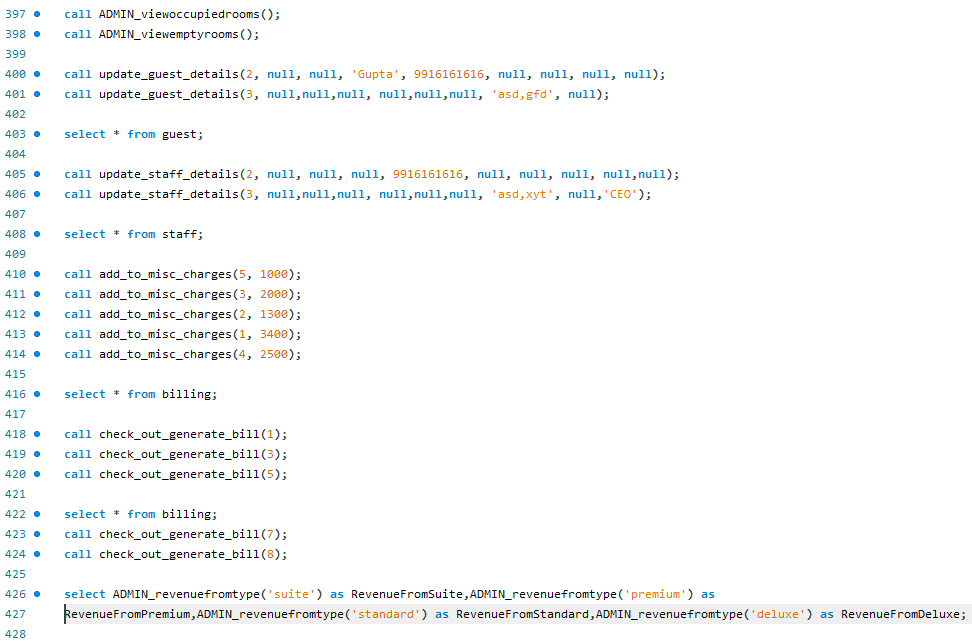
Text

Description automatically generated with medium confidence

* Calling the check\_in\_reservations procedure to check-in the guests.

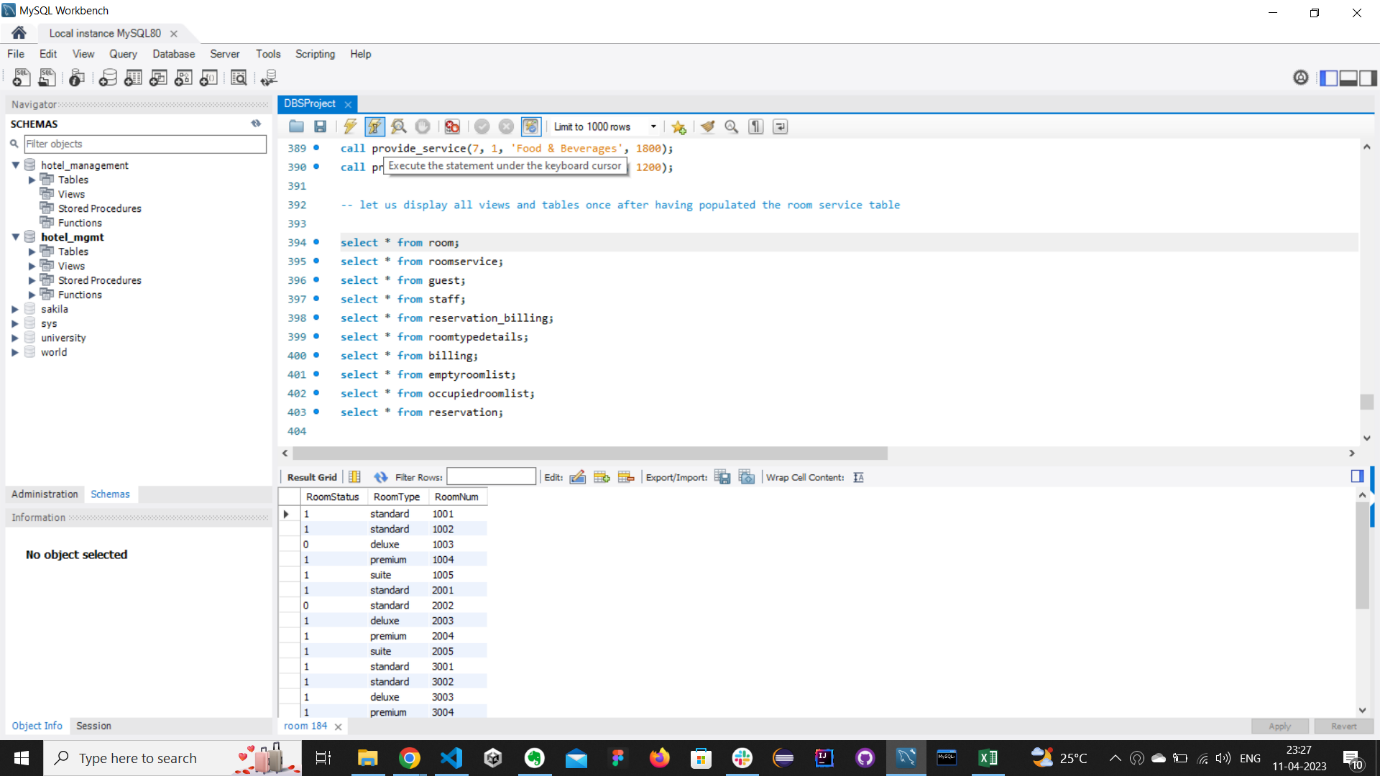
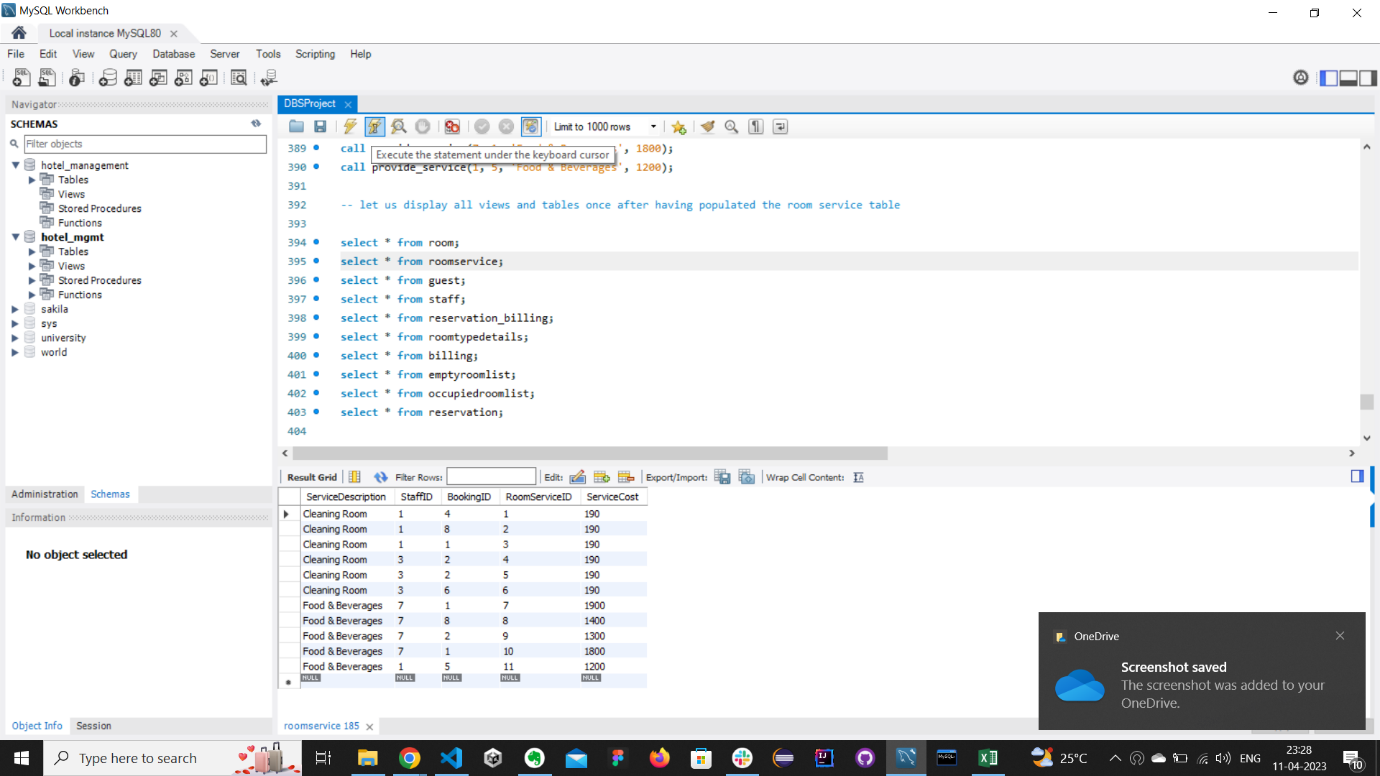
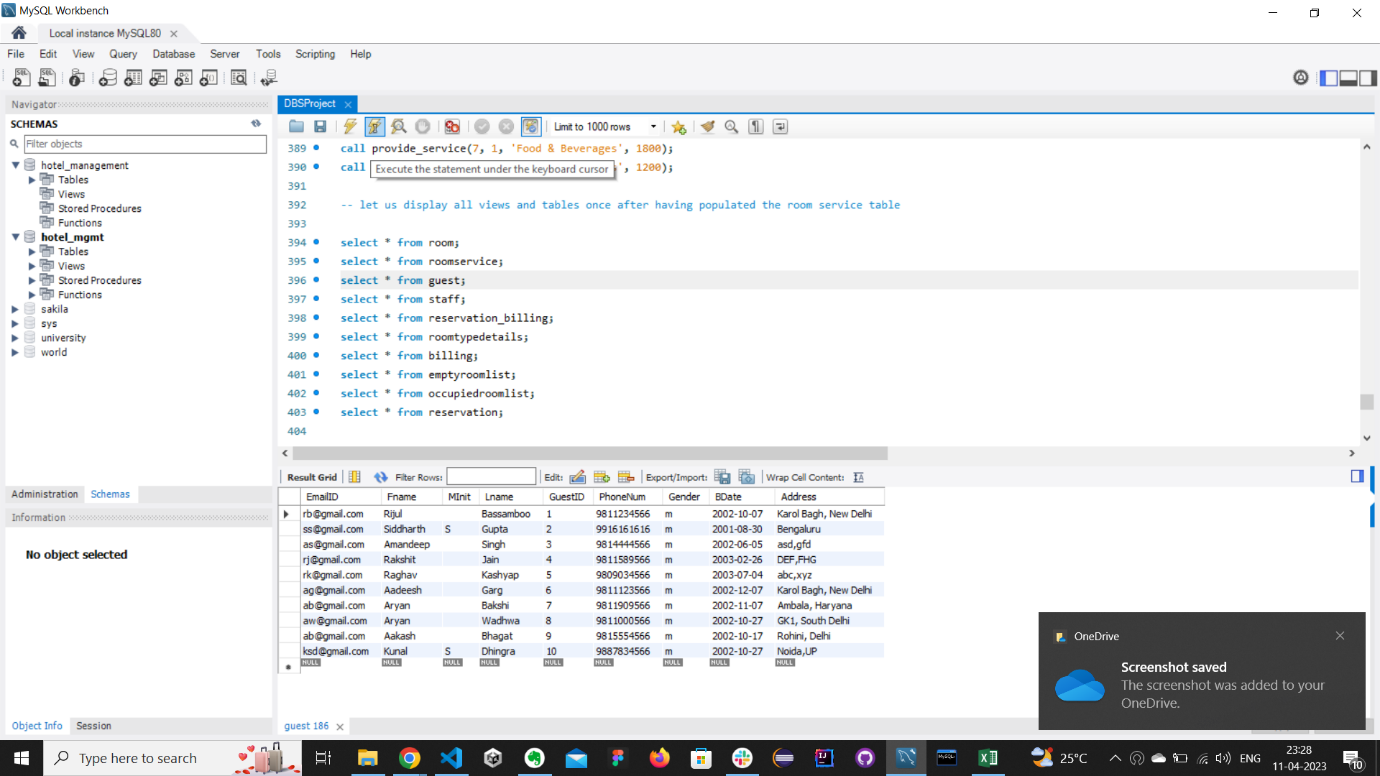
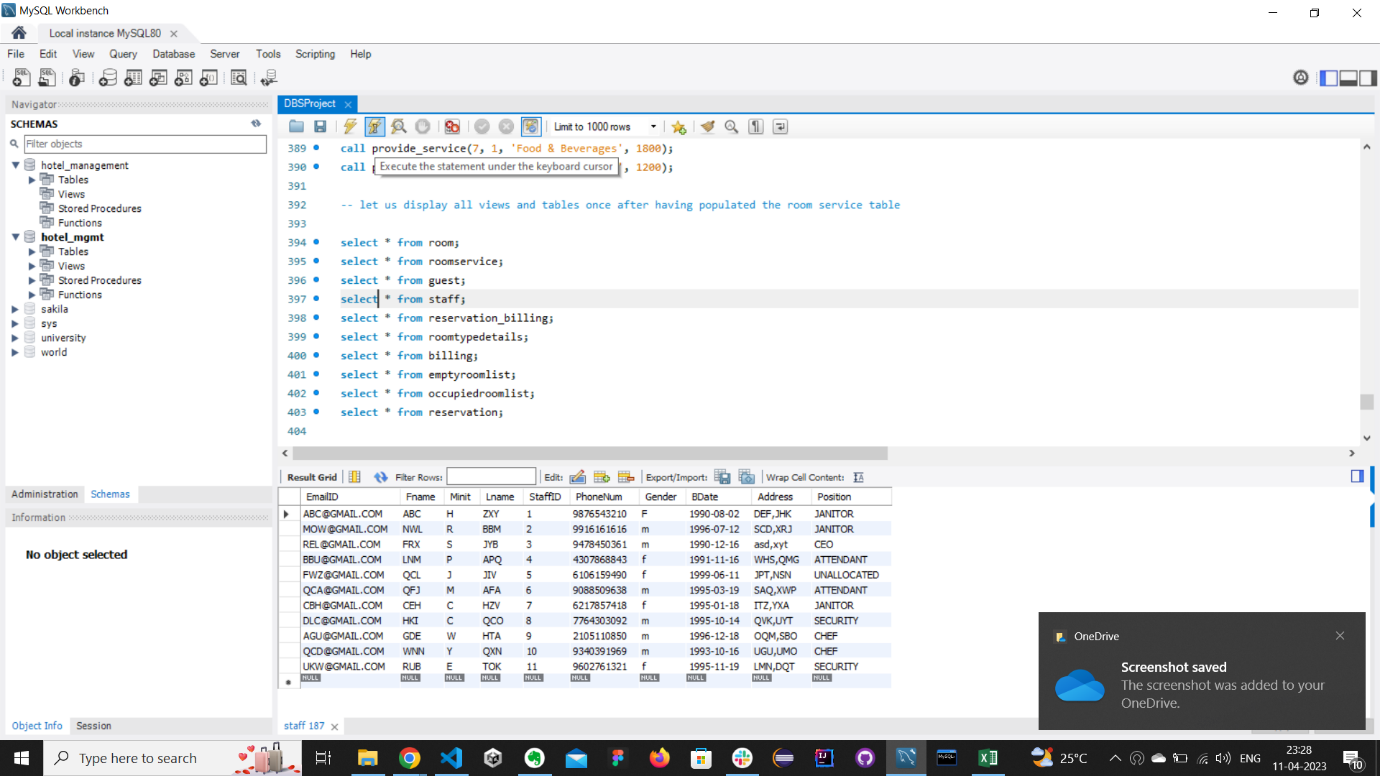
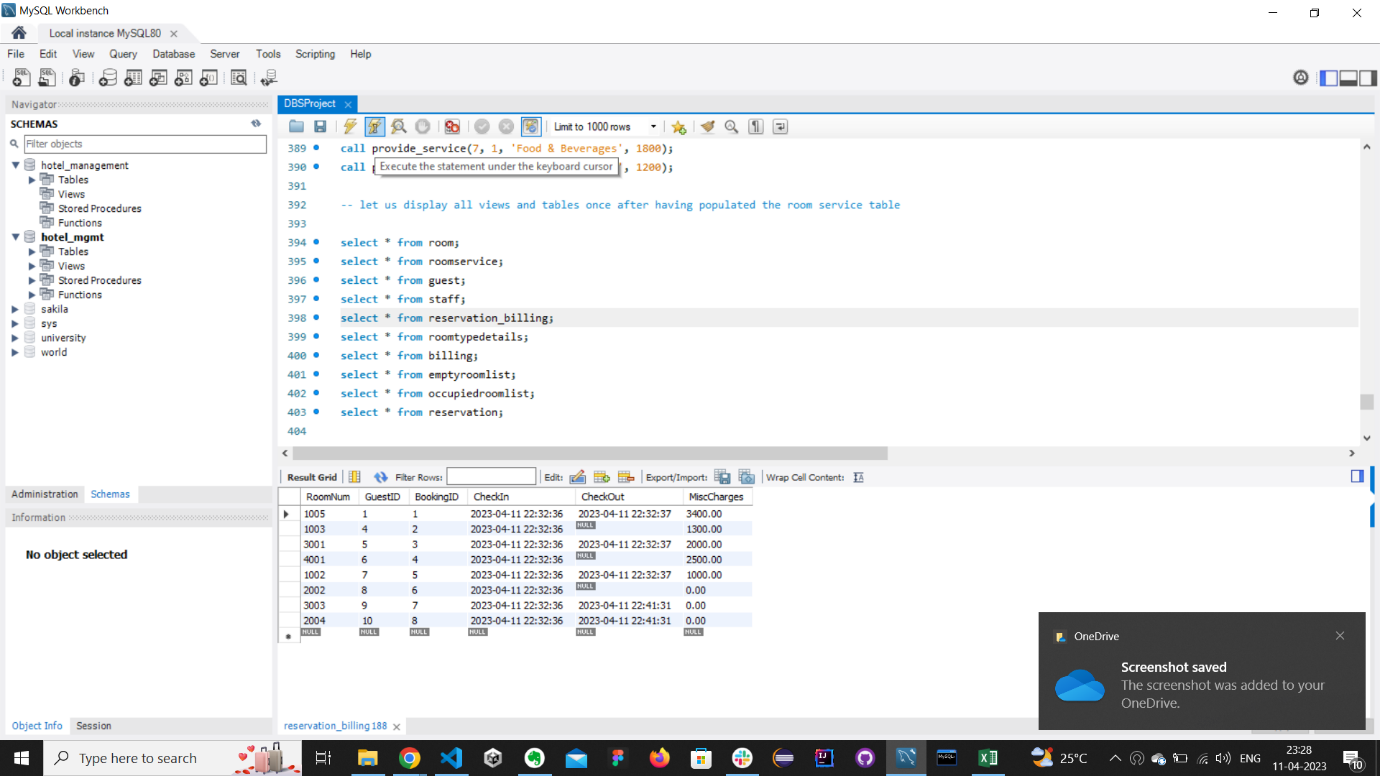
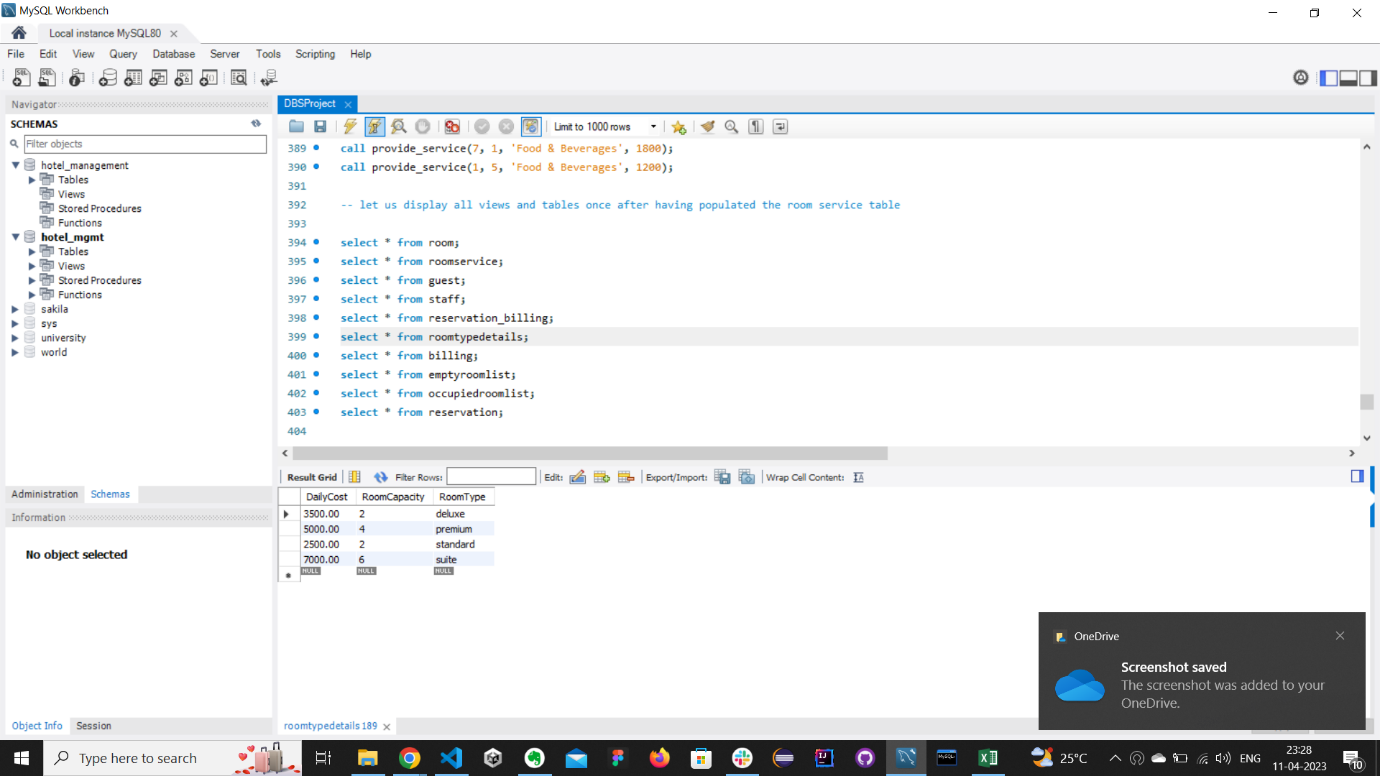
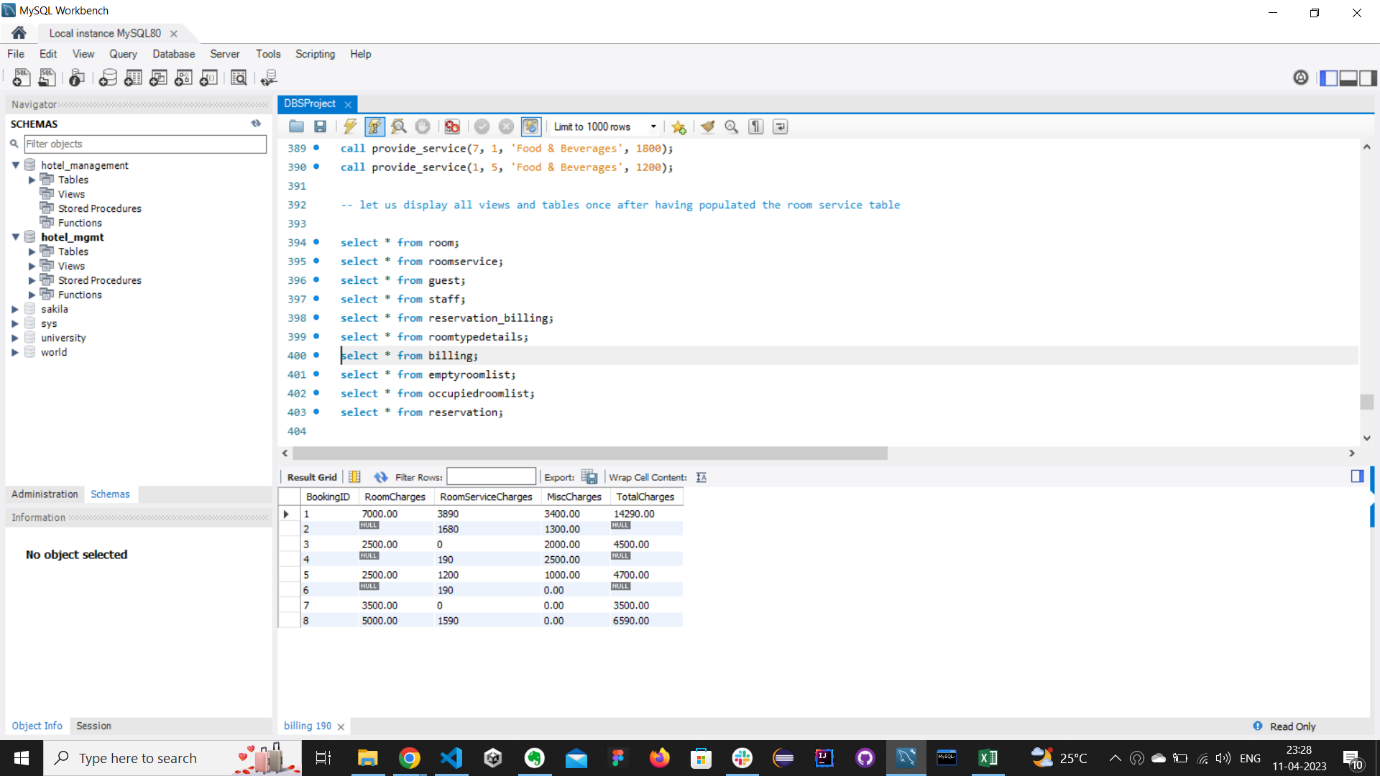
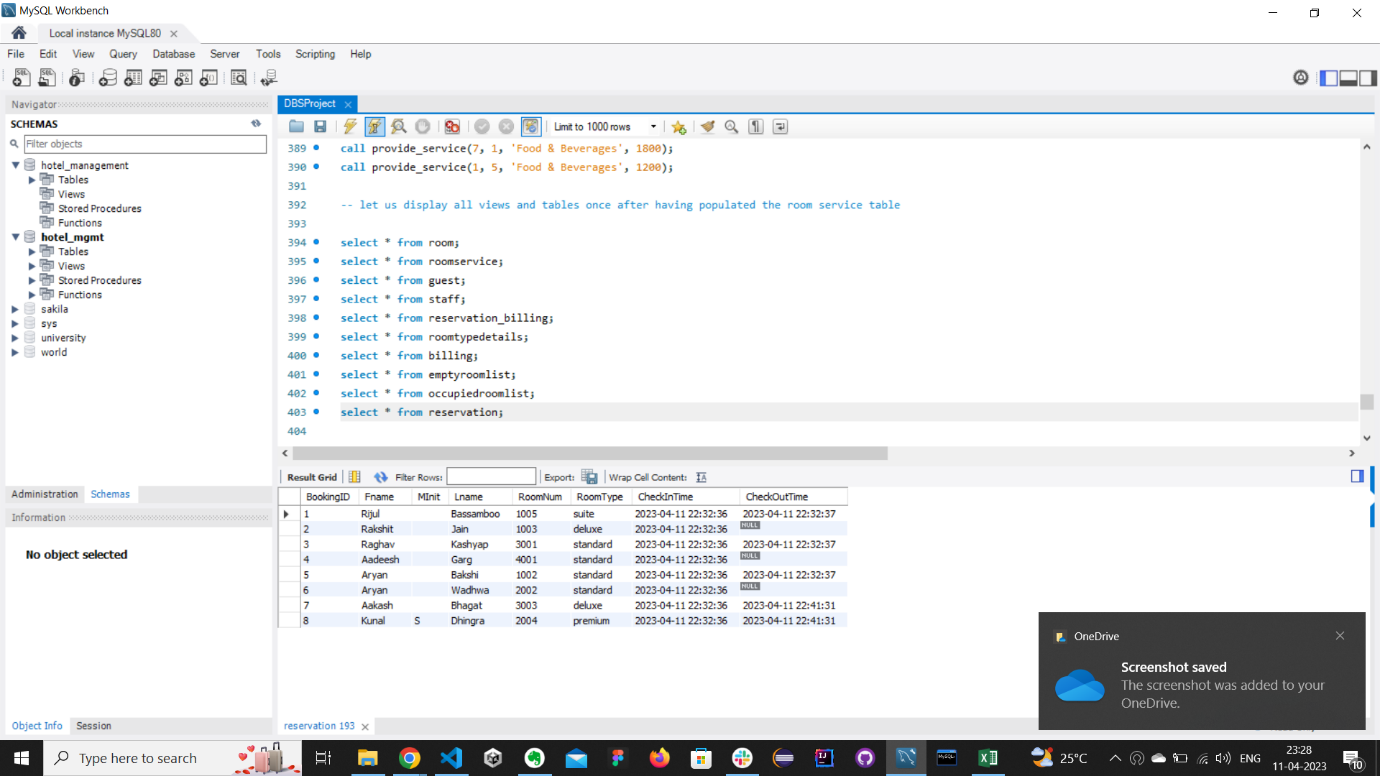
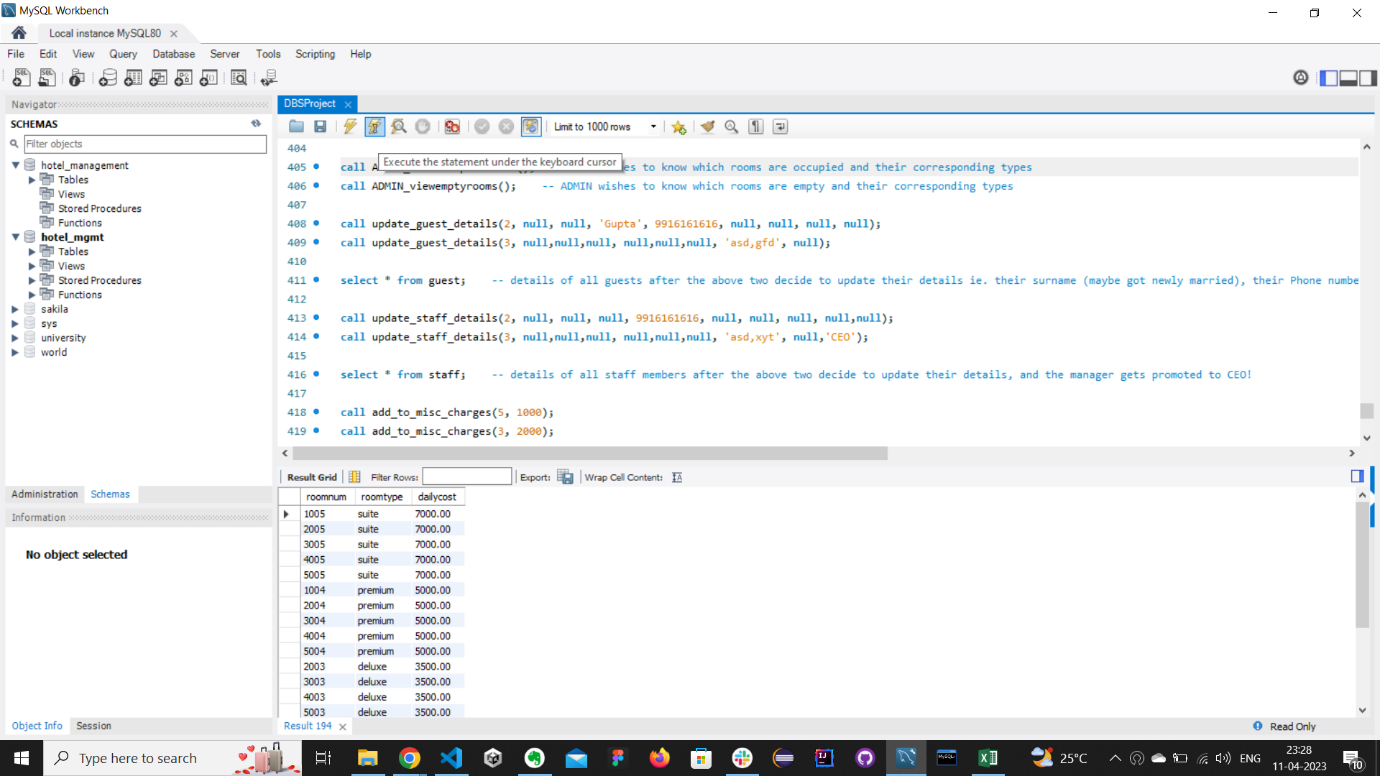
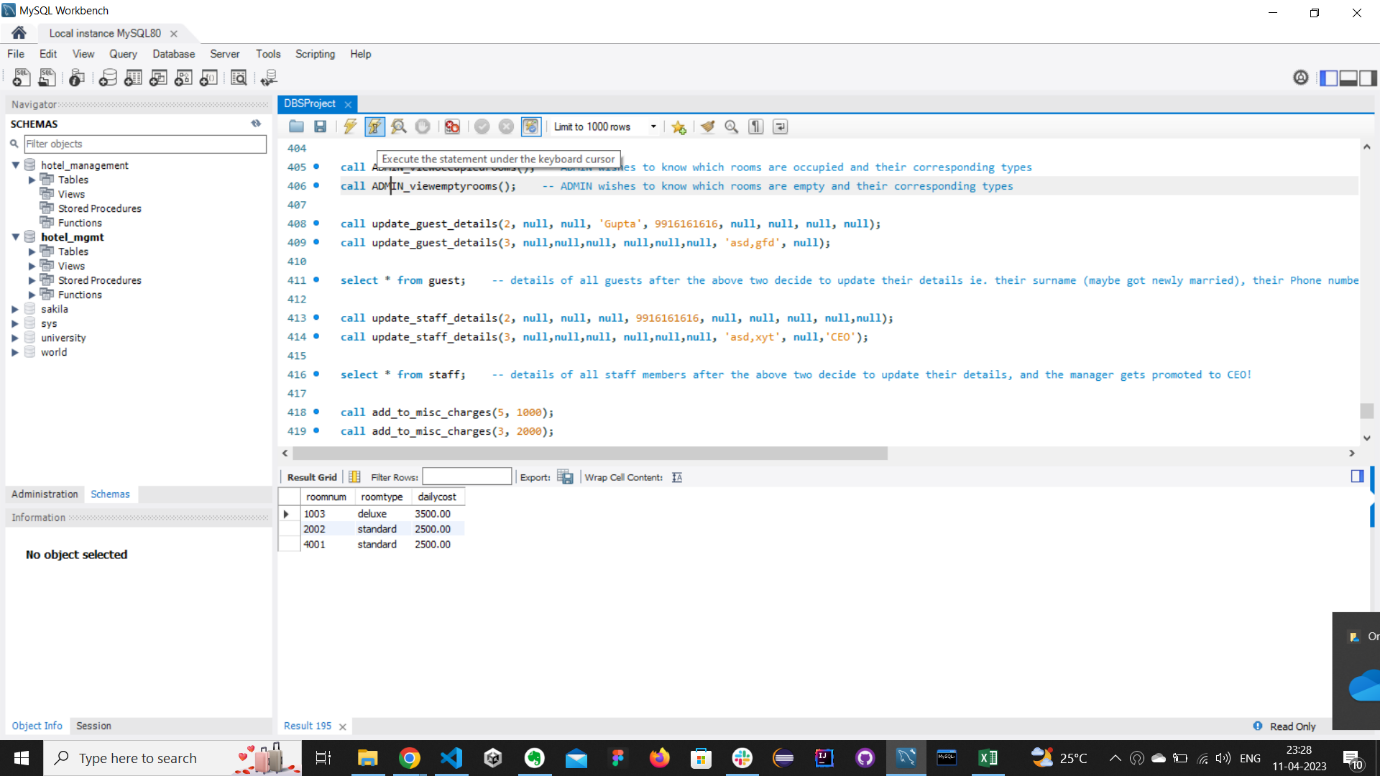
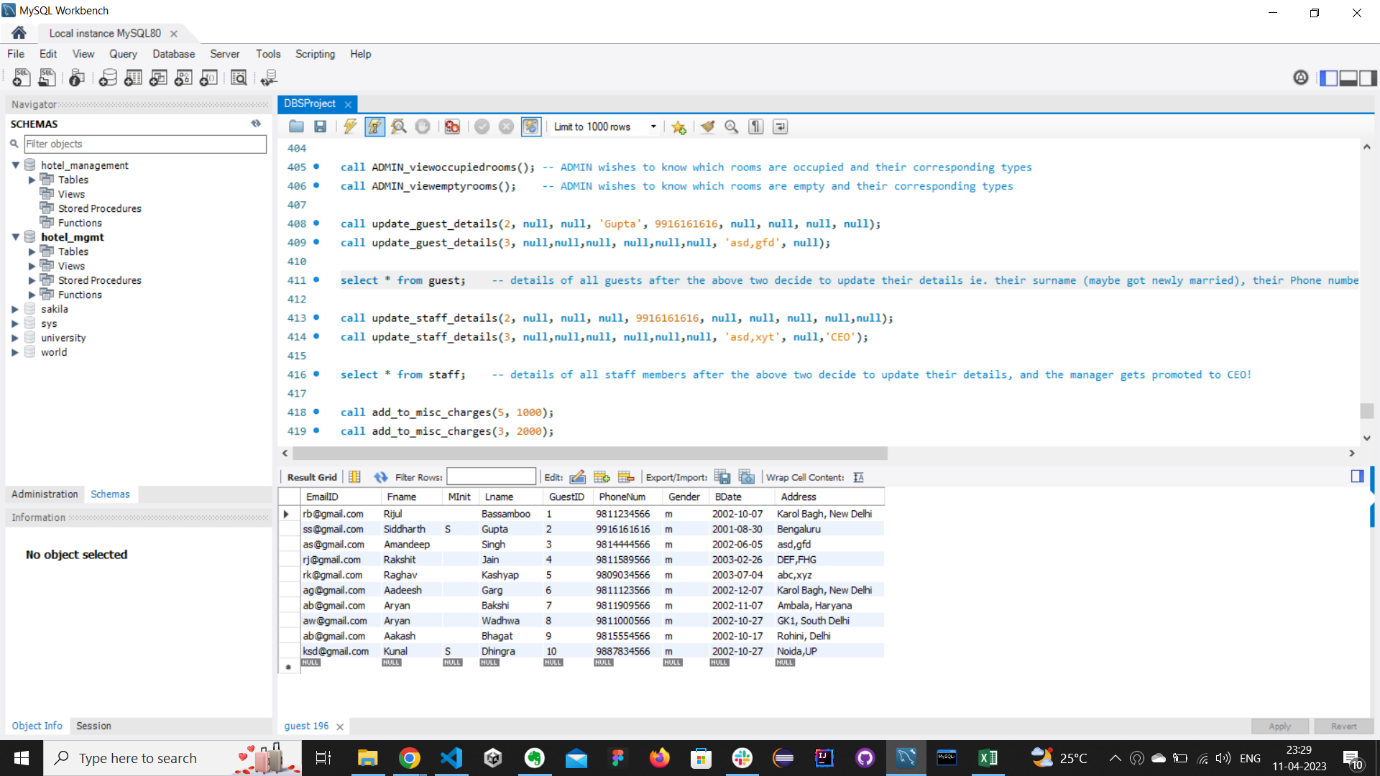
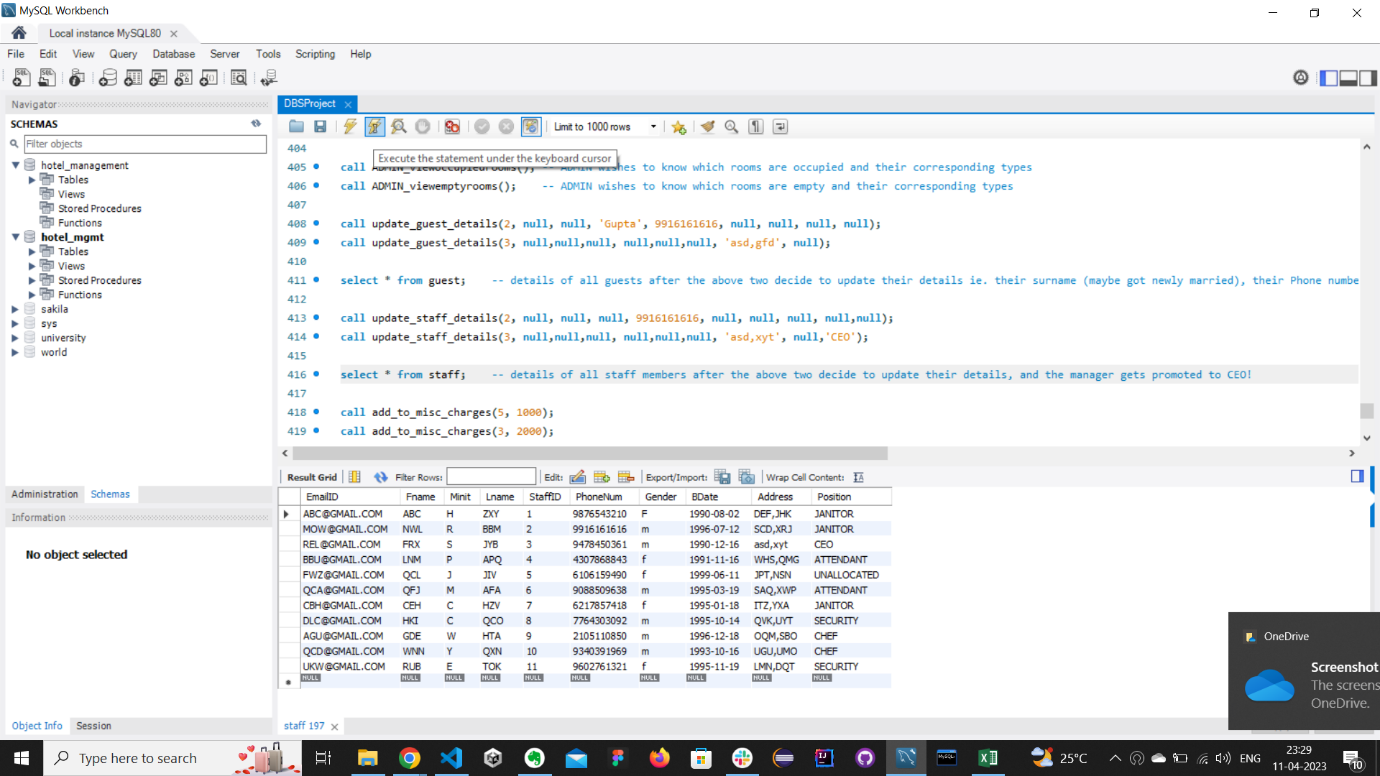
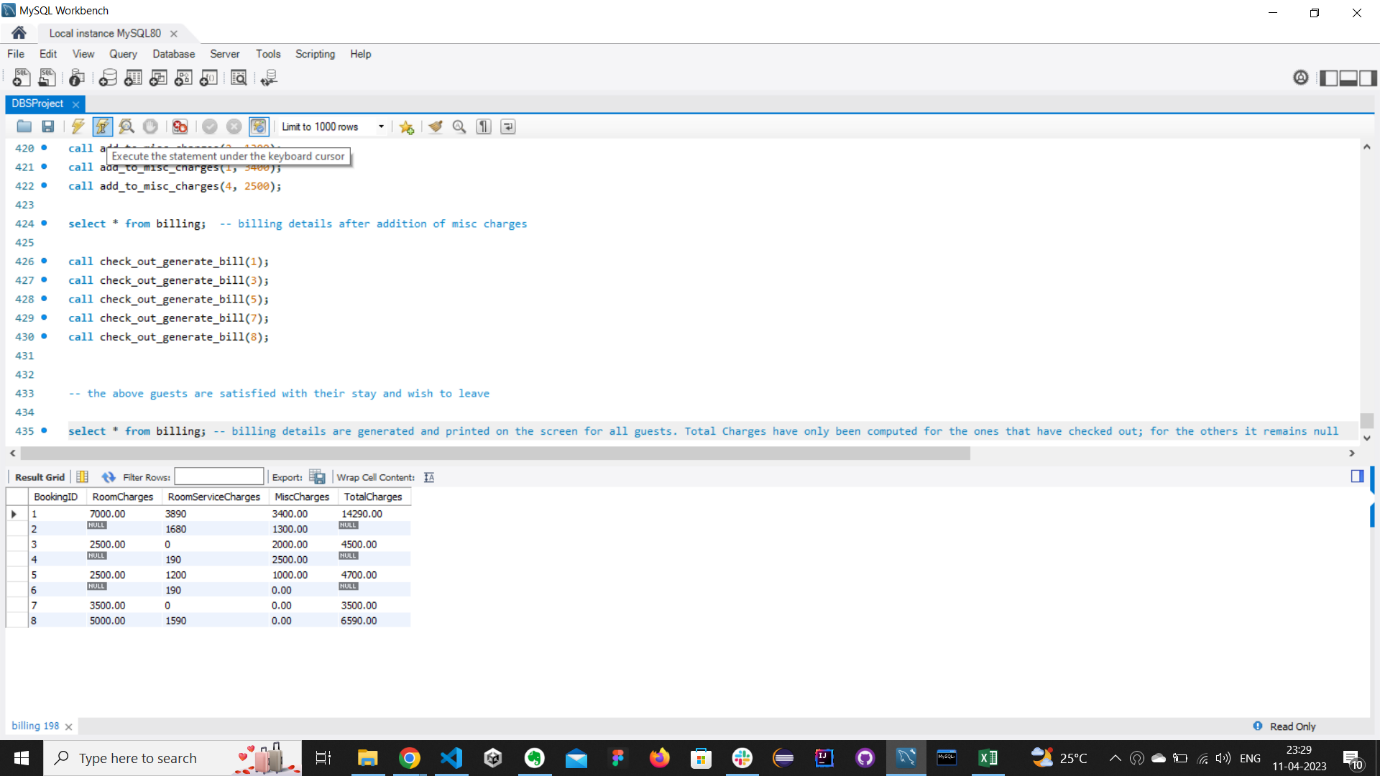
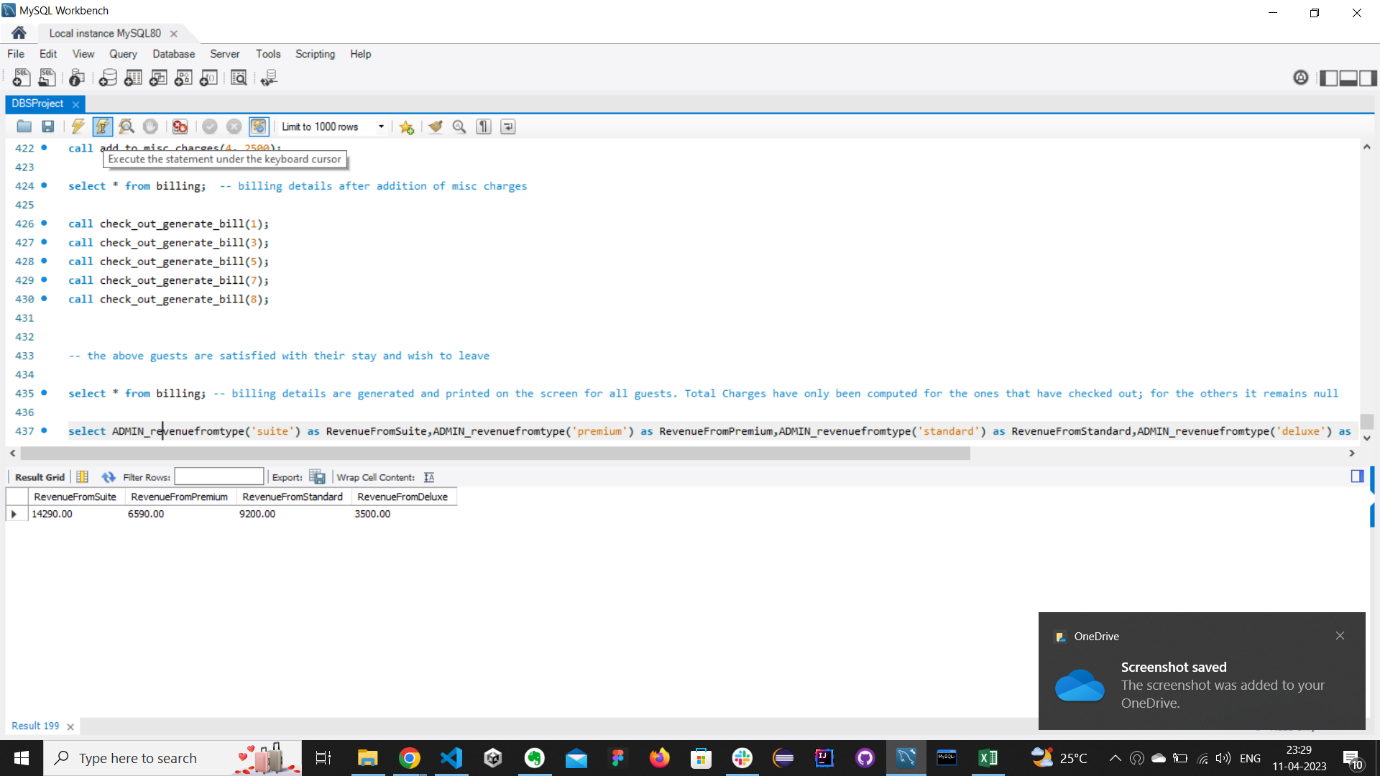
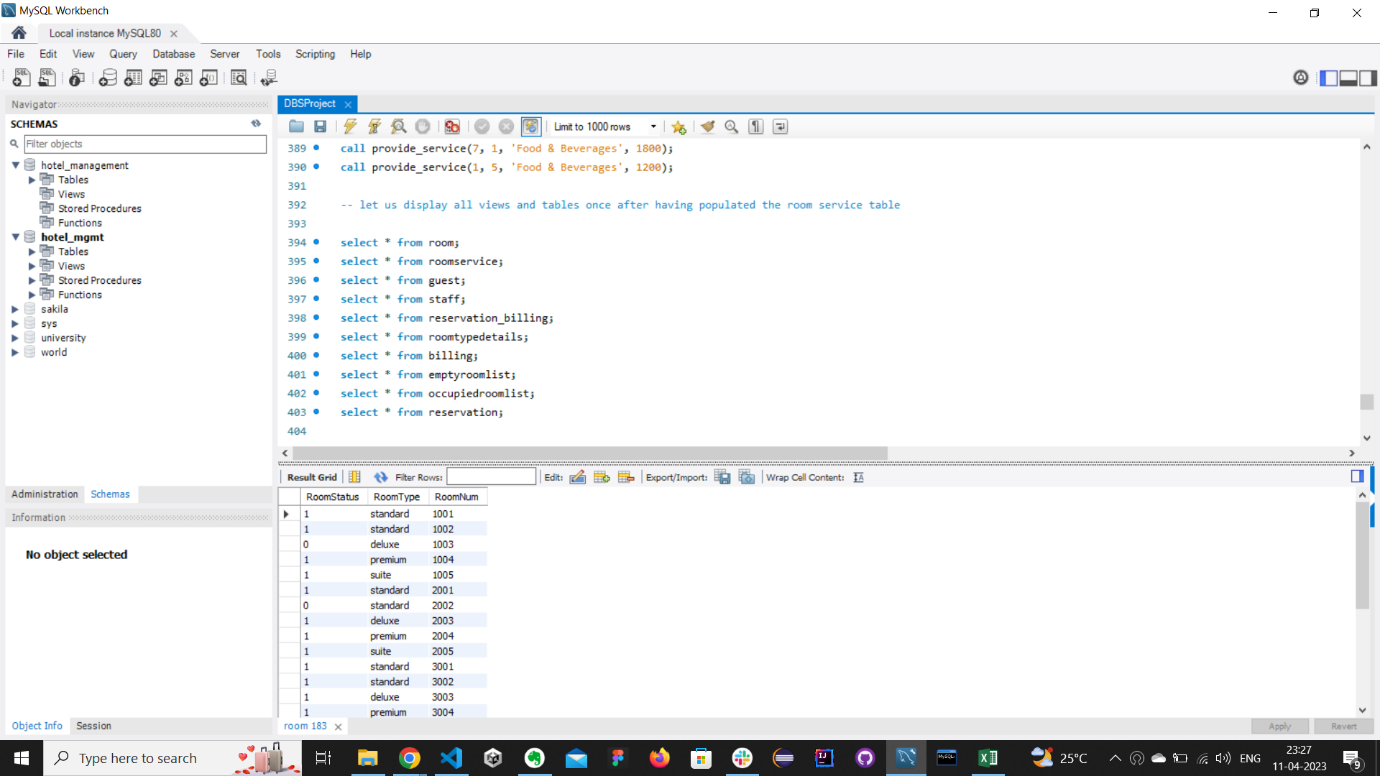


* Calling the cancel\_booking procedure.
* Calling the provide\_service procedure.
* Select commands for viewing the contents of the different tables.



* Calling ADMIN view empty and occupied room procedures.
* Calling the update guest details procedure.
* Calling the add to misc. charges procedure.
* Calling the check\_out\_generate\_bill procedure.
* Calling the ADMIN\_revenuefromtoye procedure.

**SCREENSHOT OF OUTPUTS:**



**Google Drive Link –** [**https://drive.google.com/drive/folders/1KTGc0iUG7BoXE7yj2xmGamtXb6wsm8nk**](https://drive.google.com/drive/folders/1KTGc0iUG7BoXE7yj2xmGamtXb6wsm8nk)

**The End.**

**Thank you.**