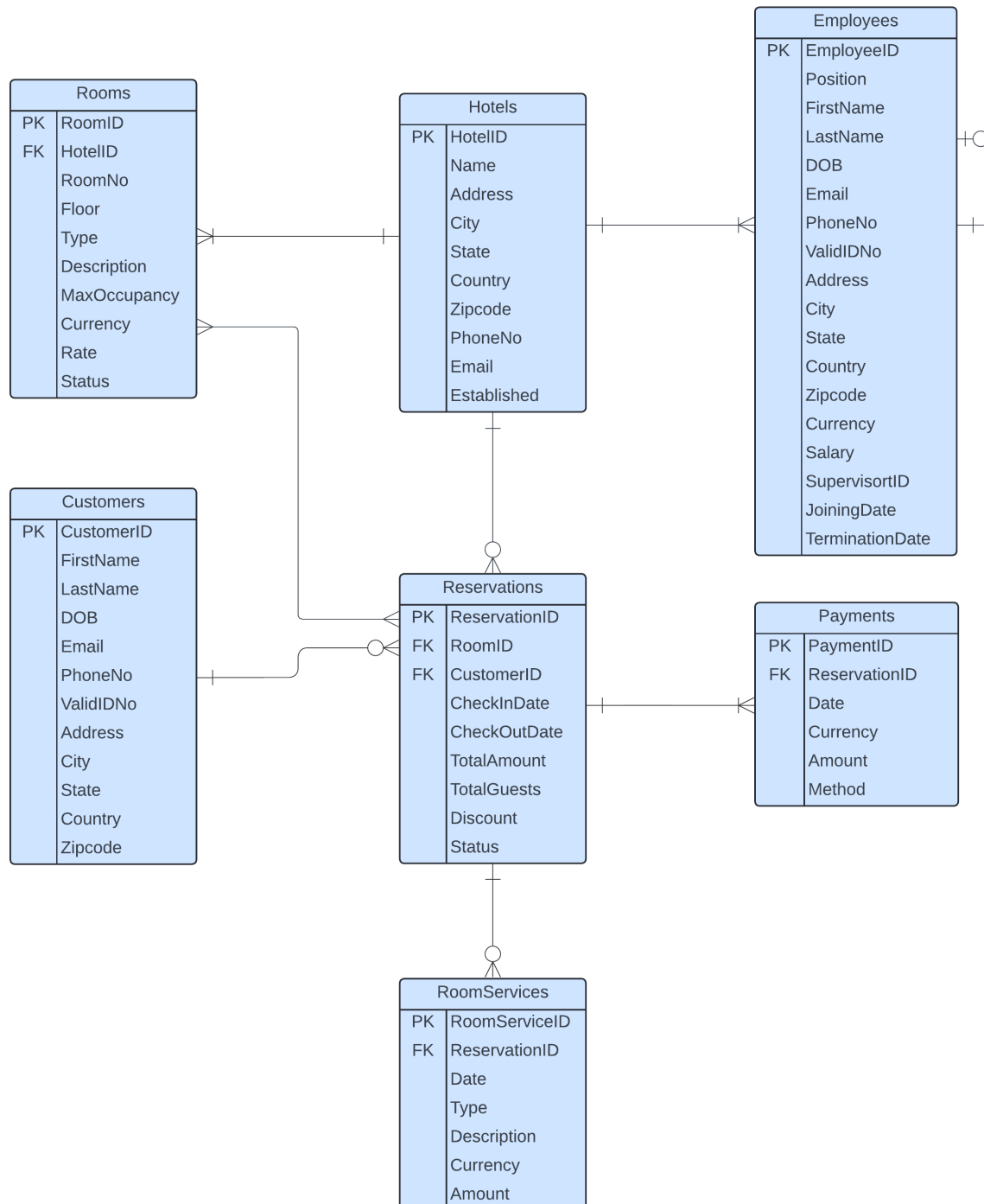


Question 1:

Following is the Entity-Relationship (ER) diagram for the room reservation system:

ER Diagram | Room Reservation System



Explanation and SQL queries:

Hotel Table:

```
CREATE TABLE Hotels (  
HotelID INT PRIMARY KEY,  
Name VARCHAR(255) NOT NULL,  
Address VARCHAR(255) NOT NULL,  
City VARCHAR(100) NOT NULL,  
State VARCHAR(100) NOT NULL,  
Country VARCHAR(100) NOT NULL,  
ZipCode VARCHAR(20) NOT NULL,  
PhoneNo VARCHAR(20) NOT NULL,  
Email VARCHAR(100),  
Established DATE  
);
```

This table would contain the hotel details: Unique hotel id - for a chain of hotels and information about the hotel.

```
CREATE TABLE Rooms (  
RoomID INT PRIMARY KEY,  
HotelID INT,  
RoomNo INT NOT NULL,  
Floor INT NOT NULL,  
Type VARCHAR(100) NOT NULL,  
Description VARCHAR(255),  
MaxOccupancy INT NOT NULL,  
Currency VARCHAR(20),  
Rate DECIMAL(10, 2) NOT NULL,  
Status VARCHAR(20),  
FOREIGN KEY (HotelID) REFERENCES Hotels(HotelID)  
);
```

This table would contain the room details:

RoomID would be the unique primary key for all rooms. Further description of the room – the room details would be stored in the table. The status of the room will store the room state if it is occupied, available, or under maintenance. The room table would reference the hotel id as a foreign key to refer to the hotel it is a part of.

```
CREATE TABLE Customers (  
CustomerID INT PRIMARY KEY,  
FirstName VARCHAR(100) NOT NULL,  
LastName VARCHAR(100) NOT NULL,  
DOB DATE NOT NULL,  
Email VARCHAR(100) NOT NULL UNIQUE,  
PhoneNo VARCHAR(20) NOT NULL,  
ValidIDNo VARCHAR(20) NOT NULL UNIQUE,  
Address VARCHAR(255) NOT NULL,  
City VARCHAR(100) NOT NULL,  
State VARCHAR(100) NOT NULL,
```

```
Country VARCHAR(100) NOT NULL,  
ZipCode VARCHAR(20) NOT NULL  
);
```

This table would contain the customer's details: Unique customer ID - for every customer and information about the customer.

```
CREATE TABLE Reservations (  
ReservationID INT PRIMARY KEY,  
RoomID INT,  
CustomerID INT,  
CheckInDate DATE NOT NULL,  
CheckOutDate DATE,  
Currency VARCHAR(20) NOT NULL,  
TotalAmount DECIMAL(10, 2) NOT NULL,  
TotalGuests INT NOT NULL,  
Discount DECIMAL(10, 2) NOT NULL,  
Status VARCHAR(50) NOT NULL,  
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),  
FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)  
);
```

This table contains all the information about the reservations. The reservation table has the reservation ID as its primary key and refers to the room ID and customer ID as the foreign key as every reservation is based on a particular room and for a particular customer. The reservations have a check-in date, a check-out date, and other details associated with the reservations.

```
CREATE TABLE Payments (  
PaymentID INT PRIMARY KEY,  
ReservationID INT,  
Date DATE,  
Currency VARCHAR(20) NOT NULL,  
Amount DECIMAL(10, 2) NOT NULL,  
Method VARCHAR(20) NOT NULL,  
FOREIGN KEY (ReservationID) REFERENCES Reservations(ReservationID) );
```

This table consists of all the payment details and information. It has a unique Payment ID and a reservation ID which is referenced from the reservations table as the foreign key. The table consists of other information such as the date, currency, amount and method of payment.

```
CREATE TABLE RoomServices (  
ServiceID INT PRIMARY KEY,  
ReservationID INT,  
Date DATE,  
Type VARCHAR(20),  
Description VARCHAR(255),  
Currency VARCHAR(20),  
Amount DECIMAL(10, 2),  
Method VARCHAR(20),
```

FOREIGN KEY (ReservationID) REFERENCES Reservations(ReservationID));

This table consists of all the room service details and information. It has a unique service ID and a reservation ID, which is referenced from the reservations table as the foreign key. The table contains other information such as the date, type, currency, amount and payment method.

```
CREATE TABLE Employees (  
  EmployeeID INT PRIMARY KEY,  
  Position VARCHAR(100) NOT NULL,  
  FirstName VARCHAR(100) NOT NULL,  
  LastName VARCHAR(100) NOT NULL,  
  DOB DATE NOT NULL,  
  Email VARCHAR(100) NOT NULL UNIQUE,  
  PhoneNo VARCHAR(20) NOT NULL,  
  ValidIDNo VARCHAR(20) NOT NULL UNIQUE,  
  Address VARCHAR(255) NOT NULL,  
  City VARCHAR(100) NOT NULL,  
  State VARCHAR(100) NOT NULL,  
  Country VARCHAR(100) NOT NULL,  
  ZipCode VARCHAR(20) NOT NULL,  
  Currency VARCHAR(20) NOT NULL,  
  Salary DECIMAL(10, 2) NOT NULL,  
  SupervisorID INT,  
  JoiningDate DATE NOT NULL,  
  TerminationDate DATE  
);
```

This table would contain the customer's details: Primary Key Employee ID - for every employee and information about the employee.

Relationships are denoted in the ER diagram.

Question 2:

The columns personId, name, email, dob, majorIds and bedId are not marked optional and hence would be considered to be mandatory.

The majors data file has multiple major id entries on same major name and description. The first occurrence of the major name has been used, and the rest of the duplicates have been dropped. I have used the id field instead of display id field in the majors data file for replacing the major names with ids.

The names have been concatenated to form the full name of the individual.

The address has then been split into address1, address2, city, state and zip. Since none of the rows have a complete address, I have kept them all as no zip code has been provided for any person.

The majors were converted to majorids.

To include the bedids, I first joined the occupancy data to inventory data and then joined it to the person data.

I finally applied the cleaning logic, where I checked every mandatory field to see if it was present and not null.

I removed the duplicate emails as the requirements of the assignment stated that the emails should be unique. The no of rows left was 100, which is the number of unique emails. The data loss is significant.

I formed the excluded data as well so that all the excluded data can be included in that CSV.

The code is in the Q2.ipynb file. The included cleaned data consists of only 100 rows, which satisfies all the requirements.