

Team: Postgres Wizards

Members: Sahong Song, Nithisha Sathishkumar, Andy Hoang

Project: Hotel Reservations

Business Problem

The hotel faces operational challenges in managing reservations, guest details, and store payments with the current process because of the increasing number of reservations and staff. Many reservations are requested at our hotel for various dates. The staff needs an easily accessible way to manage, view, schedule reservations, and track which staff member created each reservation or the reservations got online by guests. A solution is needed to replace the tedious process of manually picking up all calls and writing entries into Excel to schedule a reservation.

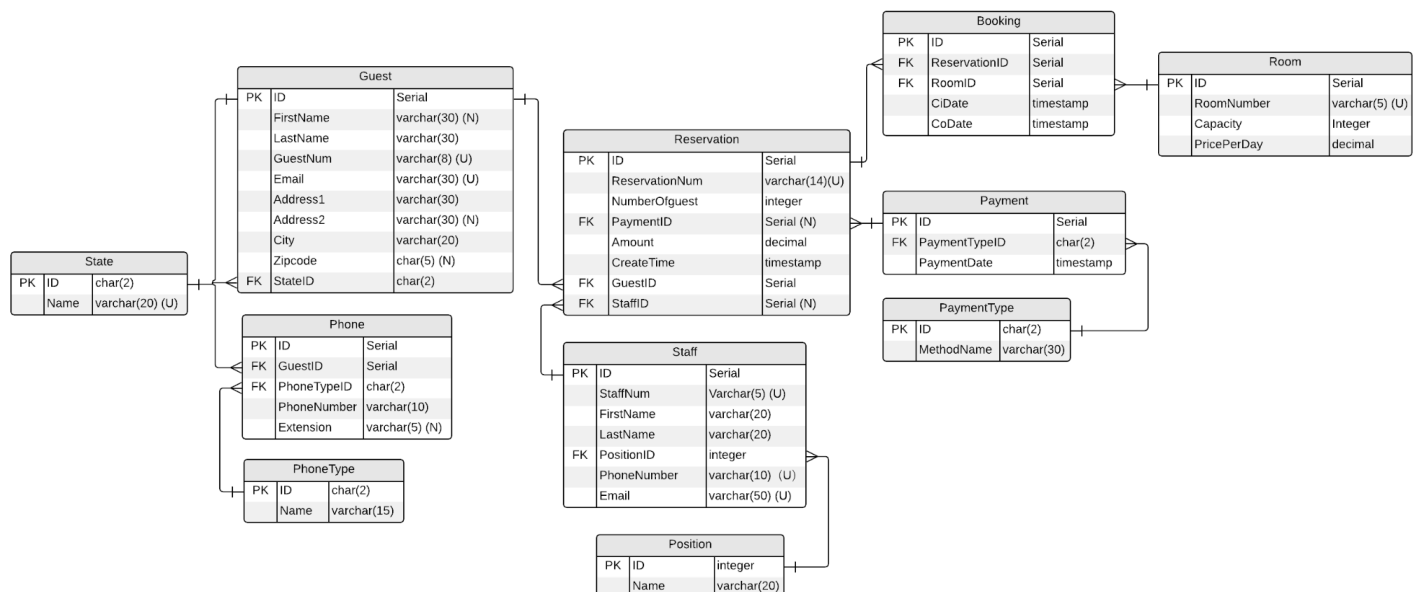
High-Level Interface:

- **Create Method:** These functions will create a record of the items below
 - ❖ Reservation
 - ❖ Booking
 - ❖ Guest
 - ❖ Staff
- **Update methods:** These functions will update any changes to the items
 - **Guest contact information**
 - Address
 - Email
 - Phone Number
 - **Staff**
 - First Name
 - Last Name
 - Email
 - Phone number
 - position
- **Booking Methods:**
 - Available rooms on specified Date
 - All Checkin and Check Out Times for a specified Room
- **List Methods:** Retrieves a list of items below
 - Staff
 - Guest
 - Reservation
 - Room
 - Payment
- **Get Methods:** Functions that retrieve specific information of items below
 - **Guest**
 - Address
 - Phone Number
 - Email
 - Guest's unique number

- **Staff**
 - Staff List
 - Job Position List
 - Staff Information by Staff Number
 - Staff Numbers
- **Reservation**
 - Creator's (guest) name
 - Creation time
 - Staff member's name that created it
- **Room**
 - Room Number
 - Price per day
 - Capacity

Schema Design :

https://lucid.app/lucidchart/2f79dea1-fbba-45f7-a2c1-0dc11ab092d3/edit?viewport_loc=482%2C-109%2C1423%2C1449%2C9Jhe~E8sqVUT&invitationId=inv_2f90cb2d-e907-46c8-a95b-939a26e3abbc



Method List

Create Methods: These functions will create a record of the items below:

- **Reservation and Booking**
 - public create_Reservation (Integer number_of_guest, integer number_of_room, String payment_type, String guest_email, string staff_number):
 - Create a new Reservation object and Booking object with the related information (e.g. payment type, GuestNum (customer who

ordered the reservation), staffNum (staff member that created the reservation for the guest). Overloaded/multiple methods would be created as to handle the nullable values (e.g. staffNum and paymentType)

- Return Reservation obj (Object with information in params)
- Return Booking obj (Object with information in params)

This API does not require users to give all information at one time instead it provides an interactive interface for users to input the necessary information one by one.

Each reservation may have one or more booking records, and each booking mentions one room. This Api supports that guest to order one or more rooms(bookings) at same time and then input cidate, codate for each room. Also the guest has to exist before creating the reservation

- **Guest**

- public createGuest (String firstName, String lastName, String email, String address1, String address2, String phoneType, String City, Char zipcode, String state):
 - Creates a new Guest object into the system with their personal data and contact information. PhoneType indicates whether they used their cell or work phone number. The nullable values are to be handled by overloaded methods capable of handling each case, but have essentially the same outcome.
 - Return Guest obj (Object with information in params)

- **Staff**

- public static boolean createStaff (FirstName, Lastname, email, phonenumber, positionID)
 - Insert createStaff into the system which allows for create staff information with their personal and contact information and will automatically set to a staffID
 - Return true bool is the operation is successful and Staff obj (Object with information in params)

Update Methods: These functions will update any changes made to the items below:

- **Reservation**

- Public static updatePayment(reservationNum, createTime, PaymentMethod, PaymentDate)
 - Returns an updated reservation with updated payment information

- **Guest information**

- public static boolean updateGuestPhoneNumber(GuestNum, phonenumber)

- update the Guest phone number using their GuestNum (Identification number).
 - Return true bool is the operation is successful and Guest obj (With newly updated information from parameters)
- public static boolean updateGuestEmail(guestNum, email)
 - update the Guest email using their GuestNum (Identification number).
 - Return true bool is the operation is successful and Guest obj (With newly updated information from parameters)
- public static boolean updateGuestAddress(guestNum, address)
 - update the Guest email using their GuestNum (Identification number).
 - Return true bool is the operation is successful and Guest obj (With newly updated information from parameters)
- **Staff information**
 - public static boolean updateStaffPhoneNumber(staffNum, phonenumber)
 - update the Staff phone number using their staffNum (Identification number).
 - Return true bool is the operation is successful and Staff obj (With newly updated information from parameters)
 - public static boolean updateStaffEmail(staffNum, email)
 - update the Staff email using their staffNum (Identification number).
 - Return true bool is the operation is successful and Staff obj (With newly updated information from parameters)
 - public static boolean updateStaffFirstName(staffNum, firstname)
 - update the Staff Firstname using their staffNum (Identification number).
 - Return true bool is the operation is successful and Staff obj (With newly updated information from parameters)
 - public static boolean updateStaffLastName(staffNum, lastname)
 - update the Staff lastname using their staffNum (Identification number).
 - Return true bool is the operation is successful and Staff obj (With newly updated information from parameters)
 - public static boolean updateStaffPosition(staffNum, PositionName)

- update the Staff PositionName using their staffNum (Identification number).
- Return true bool is the operation is successful and Staff obj (With newly updated information from parameters

Booking Methods: These functions provide lists based on specification/parameters:

- **date ranges for a room**
 - private getBookingsOnRoom(String roomNumber)
 - Retrieves information based on this room that has been checked in and checked out from the reservations.
 - Returning a list of all time periods (Timestamp List/Array) that were reserved history for this room.
- **Available room from date range**
 - public static void getAvailableRooms(Timestamp Date)
 - With the specified date, it retrieves all unavailable rooms that have not been checked out yet in the specified date. Then 'left join' with Room(left) table to get what rooms are available.
 - Returning an Array/List of what rooms with roomNumber are available for this time date range.

List Methods: These Functions will retrieve and list all the items below:

- Staff
 - public static void getStaffList()
 - Returns a list of all staff with their information including their staffnum in a ordered by their first name in alphabetical order
 - public static void getJobPositionList()
 - Returns list of job position names in a ordered by their first name in alphabetical order
- Guest
 - public static void getGuestList()
 - Returns a list of all guests with their information including their guestNum in a ordered by their first name in alphabetical order
- Reservation
 - Public getReservationList()
 - Return list of all reservations
- Booking
 - Public getBookingList()
 - Return list of all bookings

- Room
 - public static void getRoomList()
 - Returns a list of all Room with their information including their roomnumber in a ordered by their roomnumber
- Payment
 - public static void getPaymentList()
 - Returns list of payment type in a ordered by their name in alphabetical order

Get Methods: These functions will retrieve specific information of the items below:

- **Guest information such as an address, phone number, and email**
 - public static void getGuestInfoByGuestNum(GuestNum)
 - Returns firstname, Guestnum, lastname, phonenumber, email, address
 - public static void getGuestInfoWithoutGuestNum(LastName, Email)
 - Returns related guest information
- **Staff information**
 - public static void getStaffInfoByStaffNum(StaffNum)
 - Returns firstname, staffnum, lastname, phonenumber, email, position.name
 - public static void getStaff_Staffnum(FirstName, LastName, Email)
 - Returns a staffNum
 - public static void getStaffPhoneNumber(Staffnum)
 - Returns a Phonenumber
 - public static void getStaffEmail(Staffnum)
 - Returns a Email
- **Reservation**
 - Staff who created the reservation
 - private getStaffInfo(String reservationNum)
 - With the reservationNum, it would grab the staff member information that would be related to the creation of the reservation.
 - Returns a Staff obj with the staff information who created the reservation, if it returns null that means this reservation was booked online.
-

- **Existing reservation details ordered by a guest such as guest's name, create time, and which staff respond to this reservation.**
 - public getReservationInfo(String guestNum)
 - Input of guestNum will bring up the latest reservations reserved by the corresponding guestNum
 - Return Reservation obj that was reserved by this guest, with its information of the reservation.
- **Existing Bookings details in a certain time period.**
 - public getBookingInfo(String ReservationNum)
 - Input of ReservationNum will bring up the Bookings included by the Reservation.
Return Booking obj that was in the reservation, with its information of Cidate and CoDate)
- **Room information such as capacity, price per day**
 - public getRoomInfo(String roomNumber)
 - Input of roomNumber will bring up the information about this room.
 - Return the Room obj with the corresponding roomNumber, and get the information about this room.

Source Link:

https://github.com/NithishaSathishkumar/475Final_Project

Development tools and Environment.

- a. **Choose Database Software (DBMS): EG MySql, SQL Server, Postgres, SQLite**

We have opted to utilize Postgres as our database software for this project.

- b. **UI tools: Software you will use for developing your UI in Web, Forms, or console: EG c++, c# Java, Python etc. (Note – using an IDE for SQL databases is NOT an acceptable solution. You must use some higher level language.)**

For developing our user interface in the console, we have selected **Java** as our coding language. Additionally, we will be using Visual Studio Code as our integrated development environment (IDE) for Java coding.

JDBC will be an integral part of our project, facilitating the connectivity between our Java-based application and the selected database (Postgres).

- c. **Database Hosting: How are you going to host your database for your team, and allow users access. EG Cloud services, personal server)**

We have decided to host our database locally, avoiding reliance on external cloud services. This approach ensures greater control over access and management within our team.

d. Any other support software you will use in your project.

- Version Control: GitHub will serve as our platform for version control.
- Communication: Discord will be our chosen tool for team communication.
- Note-Taking: Google Docs will be employed for documenting important details.
- Diagram Creation: Lucidchart will be utilized for creating and managing diagrams in our project.

Demo steps:

1. check what rooms are available during the desired time
2. choose a room and check its info to ensure it is good for this guest
3. make a reservation for this guest
4. check if the booking successful
5. retrieve that room's viability

Show reservation data and
booking data in the our
database at this time



1. check what rooms are available during the desired time
2. choose a room and check its info ensure if it is good for this guest
3. make a reservation for this guest
4. check if the booking successful |



Show reservation data and
booking data in the our
database at this time since
they are changed now



Show available rooms at
this time since that room
will be not available during
the time we have ordered