Ranishree Anegundi
ID# 99795021

SWE 565 1 Advanced Database Systems
Hotel Management System (MySQL)

Table of Contents

Ranishree Anegundi
ID# 99795021

1. Introduction

The Hotel Management System (HMS) is a comprehensive software solution designed to streamline and automate various aspects of hotel operations. This document outlines the database design for the HMS, highlighting the structure, relationships, and rationale behind the chosen database system.

2. Database System

2.1 Database Management System (DBMS)

For the Hotel Management System, I have selected a relational database management system (RDBMS) due to its structured and efficient data organization. Specifically, I have chosen MySQL as the DBMS for its reliability, scalability, and widespread use in the industry. MySQL provides robust support for handling large datasets and ensures data integrity through its ACID (Atomicity, Consistency, Isolation, Durability) compliance.

3. Database Design

3.1 Entities and Relationships

The HMS database is designed to capture the following key entities:

Staff: Information about hotel staff, including roles, contact details, and others.
Rooms: Details about the hotel rooms, such as room type, availability, and pricing.
Guests: Information about the guests, including personal details and contact information.
Reservations: Records of guest reservations, including from-date and to-date.
Bookings: Records of guest bookings, including check-in and check-out dates.
Services: Various services offered by the hotel, such as catering, laundry, and room service.
Billing: Payments for BookingID.

The relationships between these entities are modeled to represent the interactions within the hotel management system. For example, a reservation is associated with a specific guest and a particular room, creating links between the Guests, Rooms, and Reservations tables.

3.2 Database Schema

The database schema consists of normalized tables to minimize redundancy and maintain data consistency. Below is a simplified representation of the schema:

Ranishree Anegundi
ID# 99795021

Staff Table:

StaffID (Primary Key)
FirstName
LastName
Designation
Email

Rooms Table:

RoomNum (Primary Key)
Type
Capacity
PricePerDay
Status

Guests Table:

GuestID (Primary Key)
FirstName
LastName
Address
Phone
Email

Reservations Table:

ReservationID (Primary Key)
GuestID (Foreign Key)
FromDate
ToDate
NoOfRooms

Bookings Table:

BookingID (Primary Key)
GuestID (Foreign Key)
RoomNum (Foreign Key)
StaffID (Foreign Key)
CheckInDate
CheckOutDate
NumOfGuests

Services Table:

ServiceID (Primary Key)

Ranishree Anegundi
ID# 99795021

BookingID (Foreign Key)
Note
Charges

Billing Table:
BillingID (Primary Key)
BookingID (Foreign Key)
PaymentDate
TotalAmount
Status

4. Rationale for Database Choice

The selection of MySQL as the database system for the Hotel Management System is driven by several key factors:

Open Source: MySQL is an open-source RDBMS, making it cost-effective and accessible for businesses of all sizes.

Scalability: MySQL is known for its scalability, allowing the hotel management system to handle an increasing volume of data and transactions as the business grows.

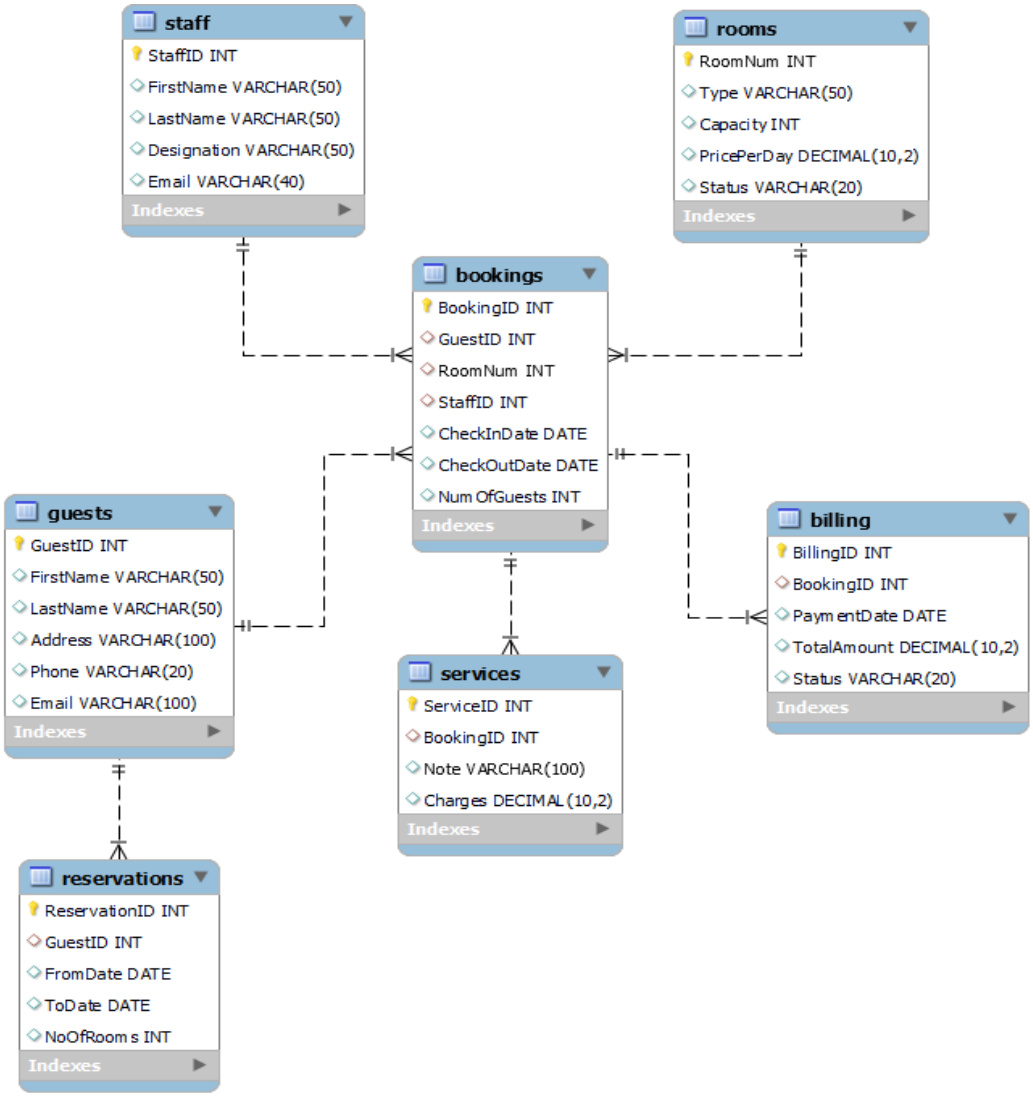
Community Support: MySQL has a vast and active community, providing extensive documentation, forums, and resources for troubleshooting and development.

Reliability: MySQL has a proven track record for reliability and stability, ensuring the secure storage and retrieval of crucial hotel management data.

Compatibility: MySQL is widely supported by various programming languages and frameworks, offering flexibility in system integration and development.

Ranishree Anegundi
ID# 99795021

5) Entity Relationship Diagram



Entity Relationship Diagram

Ranishree Anegundi
ID# 99795021

6) Queries

6.1) Query items under a specific category.

6.1.1) Query rooms available based on availability status.

```
SELECT
    r.RoomNum as RoomNumber,
    r.Type as RoomType,
    r.Capacity,
    r.PricePerDay
FROM Rooms r
WHERE r.Status = 'Available';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RoomNumber	RoomType	Capacity	PricePerDay
101	Standard	2	100.00
104	Mountain View	3	160.00
105	Executive	2	220.00
106	Suite	4	210.00
110	Ocean View	2	190.00
111	Mountain View	3	180.00
112	Standard	2	110.00
201	Deluxe	3	150.00
202	Standard	2	120.00
203	Ocean View	2	190.00
204	Mountain View	3	180.00
205	Standard	2	110.00
206	Suite	4	220.00
209	Deluxe	3	170.00

Ranishree Anegundi
ID# 99795021

6.1.2) Query managers in the Staff table.

```sql
-- Query managers in the Staff table
SELECT
    StaffID,
    CONCAT(FirstName, '', LastName) as ManagerName,
    Email
FROM Staff
WHERE Designation = 'Manager';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

StaffID	ManagerName	Email
1	JohnDoe	john.doe@examplehotel.com
5	MichaelBrown	michael.brown@downtownsuites.com

Ranishree Anegundi
ID# 99795021

6.1.3) Query to get the total number of guests, rooms booked on the specified date.

```sql
-- Query to get the total number of guests, rooms booked on the specified date
SELECT
    SUM(NumOfGuests) AS TotalGuests,
    COUNT(RoomNum) AS RoomsBooked
FROM Bookings
WHERE CheckInDate = '2023-01-10';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TotalGuests	RoomsBooked
4	2

6.2) Query data between different dates.

6.2.1) Display Guest Details and Booking Details based on CheckInDate between 2023-01-10 and 2023-01-20.

```sql
-- Query data between different dates
SELECT b.BookingID,
       b.RoomNum,
       CONCAT(g.FirstName, '', g.LastName) as GuestName,
       g.Address,
       g.Phone,
       g.Email,
       b.CheckInDate,
       b.CheckOutDate,
       b.NumOfGuests
FROM bookings b
JOIN Guests g ON g.GuestID = b.GuestID
WHERE b.CheckInDate BETWEEN '2023-01-10' AND '2023-01-20';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BookingID	RoomNum	GuestName	Address	Phone	Email	CheckInDate	CheckOutDate	NumOfGuests
1	101	AliceSmith	456 Oak St	987-654-3210	alice.smith@example.com	2023-01-10	NULL	2
2	201	EvaBrown	555 Seaside Dr	444-555-6666	eva.brown@seasideresort.com	2023-01-10	NULL	2

Ranishree Anegundi
ID# 99795021

6.3) Query data for a specific item.

6.3.1) Query Room Details based on Room Number 101.

```
-- Query Room Details based on Room Number 101
SELECT * FROM rooms
WHERE RoomNum = 101;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RoomNum	Type	Capacity	PricePerDay	Status
101	Standard	2	100.00	Available
NULL	NULL	NULL	NULL	NULL

6.3.2) Query managers for a particular hotel.

```
-- Query Booking and Guest Details based on GuestID and RoomNumber
SELECT b.BookingID,
       b.RoomNum,
       CONCAT(g.FirstName, '', g.LastName) as GuestName,
       g.Address,
       g.Phone,
       g.Email,
       b.CheckInDate,
       b.CheckOutDate,
       b.NumOfGuests
FROM bookings b
JOIN Guests g ON g.GuestID = b.GuestID
WHERE g.GuestID = 10 AND b.RoomNum = 103;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BookingID	RoomNum	GuestName	Address	Phone	Email	CheckInDate	CheckOutDate	NumOfGuests
4	103	LiamTaylor	123 Maple St	123-456-7890	liam.taylor@maplelodging.com	2023-03-01	2023-03-05	1

Ranishree Anegundi
ID# 99795021

6.3.3) Query to display Guest details and bookings based on room Occupied status.

```sql
SELECT
    G.GuestID,
    G.FirstName,
    G.LastName,
    B.BookingID,
    B.RoomNum,
    B.CheckInDate,
    B.CheckOutDate,
    B.NumOfGuests
FROM Bookings B
JOIN Guests G ON B.GuestID = G.GuestID
JOIN Rooms R ON B.RoomNum = R.RoomNum
WHERE R.Status = 'Occupied';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

GuestID	FirstName	LastName	BookingID	RoomNum	CheckInDate	CheckOutDate	NumOfGuests
1	Alice	Smith	1	101	2023-01-10	NULL	2
5	Isabella	Davis	3	102	2023-03-05	2023-03-10	1
10	Liam	Taylor	4	103	2023-03-01	2023-03-05	1
11	Emma	Brown	5	104	2023-05-12	NULL	1
12	Noah	Wilson	6	105	2023-05-20	NULL	2
15	Harper	Jackson	9	106	2023-09-02	NULL	2
3	Eva	Brown	2	201	2023-01-10	NULL	2

Ranishree Anegundi
ID# 99795021

6.3.4) Query Services by the guests in the specified Room.

```sql
-- Query services taken by the guest in the specified room
SELECT
    B.RoomNum,
    B.GuestID,
    S.ServiceID,
    S.Note,
    S.Charges
FROM Bookings B
JOIN Services S ON B.BookingID = S.BookingID
WHERE B.RoomNum = 102;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RoomNum	GuestID	ServiceID	Note	Charges
102	5	3	Room Service	10.00
102	5	14	Laundry	30.00

6.3.5) Query to get the number of staff members.

```sql
-- Query to get the number of staff members
SELECT
    COUNT(*) AS NumberOfStaff
FROM Staff;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

NumberOfStaff
7

Ranishree Anegundi
ID# 99795021

6.4) Calculate the Total number of items in a period of time.

6.4.1) Query total payment by room 102 with services and number of days stayed with room rent.

```sql
-- Query total payment by room 301 with services and number of days stayed with room rent
SELECT
    B.BookingID,
    R.RoomNum AS RoomNumber,
    DATEDIFF(B.CheckOutDate, B.CheckInDate) AS NumberOfDaysStayed,
    (R.PricePerDay * DATEDIFF(B.CheckOutDate, B.CheckInDate)) AS RoomRent,
    SUM(S.Charges) AS TotalServiceAmount,
    SUM(S.Charges) + (R.PricePerDay * DATEDIFF(B.CheckOutDate, B.CheckInDate)) AS TotalPayment
FROM Bookings B
JOIN Rooms R ON B.RoomNum = R.RoomNum
LEFT JOIN Services S ON B.BookingID = S.BookingID
WHERE R.RoomNum = 102
GROUP BY B.BookingID, R.RoomNum, R.PricePerDay;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

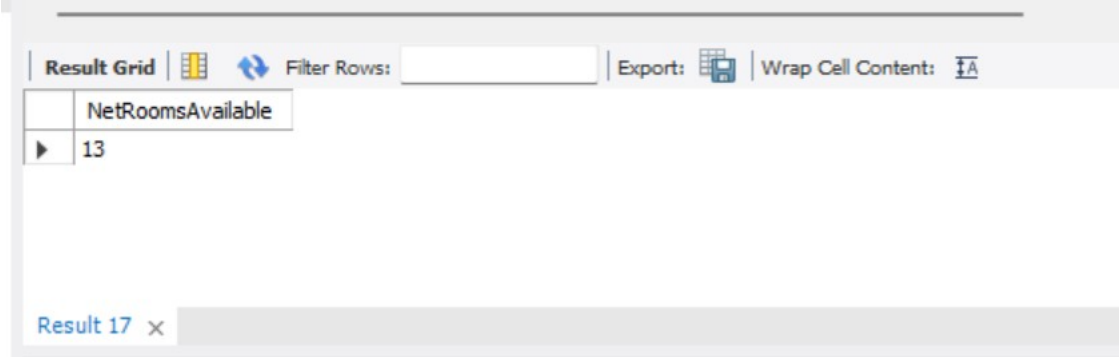
BookingID	RoomNumber	NumberOfDaysStayed	RoomRent	TotalServiceAmount	TotalPayment
3	102	5	1000.00	40.00	1040.00

6.4.2) Query net rooms available on a particular date.

```sql
-- Query net rooms available on a particular date
SELECT
    (RoomsAvailable - (NumberOfBookings + NumberOfReservations)) AS NetRoomsAvailable
FROM (
    SELECT COUNT(*) AS NumberOfReservations
    FROM Reservations
    WHERE '2023-04-01' BETWEEN FromDate AND ToDate
) AS ReservationsCount,
( SELECT COUNT(*) AS NumberOfBookings
  FROM Bookings
  WHERE '2023-04-01' BETWEEN CheckInDate AND CheckOutDate
) AS BookingsCount,
( SELECT COUNT(*) AS RoomsAvailable
  FROM Rooms r
  WHERE r.Status = 'Available'
) AS AvailableRoomsCount;
```

Ranishree Anegundi
ID# 99795021

Output:



7. Conclusion

The chosen MySQL database for the Hotel Management System is a robust and scalable solution that aligns with the system's requirements. Its relational nature, combined with features like data integrity and community support, makes it a suitable choice for efficiently managing hotel operations and enhancing overall customer experience.