Hotel Reservation System

Furkan Ünüvar

Furkan Öztürk

Pınar Erçin

Entity-Relationship Diagram

The ER Diagram for the Hotel Reservation System is provided below:

Entities

- Customers
- Rooms
- Reservations
- Payments
- Employees
- Room Services
- Feedback (Complaints)
- Events

Relationships

- Customers and Reservations (1:M)
- Reservations and Payments (1:1)
- Rooms and Reservations (1:M)
- Rooms and Room Services (1:M)
- Customers and Feedback (1:M)
- Customers and Events (M:M)
- Employees and Feedback (M:M)

Data Model Design

Customers Table

Purpose: Store customer details.

- `customer_id` (Primary Key, INTEGER, NOT NULL)
- `name` (VARCHAR(255), NOT NULL)
- `phone` (VARCHAR(15), UNIQUE, NOT NULL)
- `e_mail` (VARCHAR(255), UNIQUE, NOT NULL)

Rooms Table

Purpose: Manage room details.

- `room_id` (Primary Key, INTEGER, NOT NULL)
- `type` (VARCHAR(50), NOT NULL)
- `pricing` (DECIMAL(10,2), NOT NULL)
- `capacity` (INTEGER, NOT NULL)

Reservations Table

Purpose: Track customer reservations.

- `reservation_id` (Primary Key, INTEGER, NOT NULL)
- `customer_id` (Foreign Key to `Customers.customer_id`, NOT NULL)
- `room_id` (Foreign Key to `Rooms.room_id`, NOT NULL)
- `check_in_date` (DATE, NOT NULL)
- `check_out_date` (DATE, NOT NULL)

Payments Table

Purpose: Record reservation payments.

- `payment_id` (Primary Key, INTEGER, NOT NULL)
- `reservation_id` (Foreign Key to `Reservations.reservation_id`, UNIQUE, NOT NULL)
- `amount` (DECIMAL(10,2), NOT NULL)
- `payment_date` (DATE, NOT NULL)

Employees Table

Purpose: Store employee information.

- `employee_id` (Primary Key, INTEGER, NOT NULL)
- `name` (VARCHAR(255), NOT NULL)
- `position` (VARCHAR(100), NOT NULL)
- `contact` (VARCHAR(50), NOT NULL)

Room Services Table

Purpose: Manage additional services offered in rooms.

- `service_id` (Primary Key, INTEGER, NOT NULL)
- `room_id` (Foreign Key to `Rooms.room_id`, NOT NULL)
- `service_type` (VARCHAR(100), NOT NULL)
- `cost` (DECIMAL(10,2), NOT NULL)

Feedback (Complaints) Table

Purpose: Record customer feedback.

- `feedback_id` (Primary Key, INTEGER, NOT NULL)
- `customer_id` (Foreign Key to `Customers.customer_id`, NOT NULL)
- `feedback_details` (TEXT, NOT NULL)
- `feedback_date` (DATE, NOT NULL)

Events Table

Purpose: Store hotel event information.

- `event_id` (Primary Key, INTEGER, NOT NULL)
- `event_name` (VARCHAR(255), NOT NULL)
- `date` (DATE, NOT NULL)
- `participation_fee` (DECIMAL(10,2), NULL)

Customer-Event Table (For M:M Relationship)

Purpose: Link customers and events.

- `customer_id` (Foreign Key to `Customers.customer_id`, NOT NULL)
- `event_id` (Foreign Key to `Events.event_id`, NOT NULL)
- PRIMARY KEY (`customer_id, event_id`)

Listing 1: Query that lists how many reservations each customer has made

```sql
SELECT
    C.name AS Customer_Name,
    COUNT(Res.reservation_id) AS Total_Reservations
FROM
    Customers C
LEFT JOIN
    Reservations Res ON C.customer_id = Res.customer_id
GROUP BY
    C.name
ORDER BY
    Total_Reservations DESC;
```

Listing 2: Query that lists the number of employees based on their positions

```sql
SELECT
    E.position,
    COUNT(E.employee_id) AS Total_Employees
FROM
    Employees E
GROUP BY
    E.position
ORDER BY
    Total_Employees DESC;
```

Listing 3: List of all reservations made in the last 3 months

```sql
SELECT
    C.name AS Customer_Name,
    R.room_type AS Room_Type,
    Res.check_in_date,
    Res.check_out_date
FROM
    Reservations Res
JOIN
    Customers C ON Res.customer_id = C.customer_id
JOIN
    Rooms R ON Res.room_id = R.room_id
WHERE
    Res.check_in_date >= DATEADD(MONTH, -3, GETDATE())
ORDER BY
    Res.check_in_date DESC;
```