20.12.2023

CENG315

Hotel Management Database Project Design Report

**Prepared by**

Group 17

Serhat Eren Taş – 280201020

Ünal Dalkılıç – 280201068

Sıddık Can Boru – 280201073

Tarık Emre Gündüz – 280201075

Hüseyin Özdemir – 300201106

Table of Contents

1. [Project Description 2](#Project_Description)
2. [List all entities and relationships 3](#Entities_and_relationships)
3. [List all users of the system 4](#Users_of_system)
4. [ER Diagram 5](#ER_Diagram)
5. [Relational Schema 6](#Relational_schema)

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Name** | **Reason for Changes** | **Date** |
| Sıddık Can Boru | Initial report | 8.11.2023 |
| Sıddık Can Boru | Rewrite | 20.12.2023 |

**1. Project Description**

Project Title: *Hotel Management Database*

Project Description:

The Hotel Management Database is designed to address the data storage needs of a hotel by efficiently managing various aspects of its operations. This comprehensive database system aims to streamline guest management, room allocation, reservations, staff records, billing, supply inventory, and dining services. The system will enhance the hotel's operational efficiency, improve guest satisfaction, and facilitate the seamless running of the hotel.

*1.Guest Records:*

* Keeping thorough records of every visitor, including their contact information, preferences, and personal data.
* Recording check-in and check-out dates for each guest

*2.Room Management:*

* Keeping records of all available rooms, including room type, capacity, and pricing.
* Tracking room availability and occupancy status in real-time.
* Assigning rooms to guests during check-in, ensuring no double bookings occur during overlapping reservation periods.
* Implementing a system to handle room maintenance requests and cleaning schedules.

*3.Reservation Management:*

* Managing reservations efficiently to prevent double booking.
* Allow guests to reserve rooms online or through the front desk.

*4.Staff Records:*

* Keeping records of all hotel employees, including their responsibilities within the hotel, contact information, and personal data.

*5.Payment and Billing:*

* Manage guest payments, including room charges, additional services, and dining expenses.
* Generate and store itemized bills for each guest.
* Accept multiple payment methods, including cash, credit cards, and online transfers.

*6.Supply Management:*

* Tracking the inventory of hotel supplies, such as toiletries, linens, and cleaning materials.
* Keeping track of the ingredients in the hotel's restaurant.

*7. Dining Reservations and Orders:*

* Recording guest dining reservations, including the date, time, and number of guests.
* Managing restaurant table assignments.
* Taka and process guest orders.
* Integrate with the billing system to include dining expenses in the final bill.

**2. List all entities and relationships**

**Entity Sets and their attributes:**

1. Guest (TC/Passport no, name, surname, tel\_no, nationality, address, e-mail)
2. Booking (booking\_id, check\_in\_date, check\_out\_date, booking\_method, booking\_status, guest\_count)
3. Room (room\_id, floor, room\_type, availability, room\_status)
4. Room\_Type (room\_type\_id, price, person\_capacity, size, bed\_type, number\_of\_toilets balcony\_count)
5. Bill (bill\_id, price, date, time, bill\_status)
6. Payment (payment\_id, date, time)
7. Dinner\_Reservation (reservation\_id, name, total\_number\_of\_people, date, time)
8. Dinner\_Table (table\_id, capacity, availability)
9. Order (order\_number, order\_date, quantity, price)
10. Product (product\_id, name, unit\_price)
11. Restaurant\_customer (customer\_id, name, surname)
12. Staff (TC/YK\_no,salary, name, surname, nationality, gender, department\_name, position, tel\_no, address, e-mail)
13. Hotel (hotel\_name, district\_name, city\_name, address, rating)
14. Supply (supply\_id, brand, name, expiration\_date, quantity)

**Relationship Sets:**

1. books (Guest – Booking) (One-to-Many)

* books relation represents the owner for each booking.
* Each book has at least one guest.

1. booked\_room (Booking – Room) (Many-to-One)

* booked\_room relation represents the reserved room for each booking.

1. type\_of\_room (Room – Room\_Type) (Many-to-One)

* type\_of\_room relation represents the type of each room.

1. booking\_bill (Booking – Bill ) (One-to-One)

* booking\_bill relation represents each related booking and bill pair.

1. room\_guests (Guest – Room) (Many-to-One)

* room\_guests relation represents the room for each guest.

1. payment\_bill (Bill – Payment) (One-to-One)

* payment\_bill relation represents each related payment and bill pair.

1. guest\_payment (Guest – Payment) (One-to-Many)

* guest\_payment relation represents the guest that paid for each payment.

1. guest\_reservation (Guest – Dinner\_Reservation) (One-to-Many)

* guest\_reservation relation represents the guest that made reservation for each dinner reservation.

1. customer\_reservation (Restaurant\_Customer – Dinner\_Reservation) (One-to-Many)

* customer\_reservation relation represents the restaurant customer for each dinner reservation.

1. customer\_payment (Payment – Restaurant\_Customer) (One-to-Many)

* customer\_payment relation represents the payment of each restaurant customer.

1. dinner\_reservation\_bill (Bill – Dinner\_Reservation) (One-to-One)

* dinner\_reservation\_bill relation represents each dinner reservation and its bill.

1. dinner\_table\_reservation (Dinner\_Reservation – Dinner\_Table) (Many-to-One)

* dinner\_table\_reservation relation represents the table that reserved for each dinner reservation.

1. dinner\_table\_order (Dinner\_Table – Order) (One-to-Many)

* dinner\_table\_order relation represents the dinner table for each order made at this table.

1. product\_order (Order – Product) (Many-to-Many)

* product\_order relation represents every product for every orders.

1. hotel\_staff (Hotel – Staff) (One-to-Many)

* hotel\_staff relation represents the hotel for each staff. That specifies who works where.

1. hotel\_supply (Hotel – Supply) (One-to-Many)

* hotel\_supply relation represents the hotel for each supply. That indicates which supply is in which hotel.

1. room\_staff (Room - Staff)(Many-to-Many)

* room\_staff relation represents every staff that responsible for every room.

1. guests (Hotel - Guest) (Many-to-Many)

* guests relation represents hotel information for each guests.

**3. List all users of the system**

**Users of the System:**

1. **Admin:** Can access and modify all aspects of the database.
2. **Front Desk Staff:** Can view and manage guest and room-related information, reservations, and payments.
3. **Housekeeping Staff:** Can access room-related information and update room status.
4. **Restaurant Staff:** Can manage dining reservations and orders.
5. **Management:** Can generate reports and perform high-level management tasks.

**Assumptions and Business Rules:**

**BUSINESS RULES**

1. Guests must provide valid personal information when making a reservation, including name, contact number, and email address, etc.
2. Guests are allowed to cancel their reservation (for both dinner and room), but cancellations must be made within a specified time to avoid charges.
3. Guests can make special requests during the reservation process (extra pillows, specific room floor, connecting rooms, etc.).
4. A booking must contain only one guest identification information and count of the other guests because for a group of guests, there is only one person that needs to book.

**ASSUMPTIONS**

1. It is assumed that the availability of rooms is accurately reflected in the system and is updated in real-time.
2. The hotel assumes that the payment information provided by guests is valid, and the payment processing system is secure and reliable.
3. The accuracy of guest information, including contact details and special requests, is assumed to be the responsibility of the guests during the reservation process.
4. metin, diyagram, plan, şematik içeren bir resim

   Açıklama otomatik olarak oluşturuldu**ER Diagram**

1. **Relational Schema**

**Red type of words indicate they are added after reduction of relations**

1.Guest (TC/Passport no, name, surname, tel\_no, nationality, address, e-mail, room\_id)

2.Booking (booking\_id, check\_in\_date, check\_out\_date, booking\_method, booking\_status, guest\_count, TC/Passport\_no, room\_id, bill\_id)

3.Room (room\_id, floor, room\_type, availability, room\_status, roome\_type)

4.Room\_Type (room\_type, price, person\_capacity, size, bed\_type, number\_of\_toilets balcony\_count)

5.Bill (bill\_id, price, date, time, bill\_status)

6. Payment (payment\_id, date, time)

7.Dinner\_Reservation (reservation\_id, name, total\_number\_of\_people, date, time, bill\_id, table\_id)

1. Dinner\_Table (table\_id, capacity, availability)
2. Order (order\_number, order\_date, quantity, price, table\_id)
3. Product (product\_id, name, unit\_price)
4. Restaurant\_customer (customer\_id, name, surname)

12.Staff (TC/YK\_no,salary, name, surname, nationality, gender, department\_name, position, tel\_no, address, e-mail, hotel\_name, district\_name, city\_name)

13.Hotel (hotel\_name, district\_name, city\_name, address, rating)

14.Supply (supply\_id, brand, name, expiration\_date, quantity, hotel\_name, district\_name, city\_name)

**// below tables are coming from reduced relations**

15- payment\_bill (bill\_id ,payment\_id) (many-to-many)

remains

1. guest\_payment (TC/Passport no, payment\_ID) (One-to-Many)

remains because relation is not total

1. guest\_reservation (TC/Passport no, reservation\_ID) (One-to-Many)

remains because relation is not total

1. customer\_reservation (customerId , reservationId) (One-to-Many)

remains because relation is not total

1. customer\_payment (paymentId, CustomerId) (One-to-Many)

remains because relation is not total

20- product\_order (order\_name, order\_date, product\_id) (Many-to-Many)

remains

21- room\_staff (room\_id, TC/Passport no)(Many-to-Many)

remains

22-guests (hotel\_name , TC/Passport no) (Many-to-Many)

remains

**Relationship Sets:**

~~1.books (TC/Passport no - BookingId) (One-to-Many)~~

TC/Passport(guest table)added into booking table

~~2.booked\_room (bookingId – roomId) (Many-to-one)~~

room\_id(room table) is added into booking table

~~3.type\_of\_room (room\_id, room\_type) (many-to-one)~~

room\_type(Room\_Type table) added into room table

~~4.booking\_bill (booking\_,dd , bill\_id ) (One-to-One)~~

bill\_id(bill) added into booking table

~~5-room\_guests (TC/Passport no, room\_Id) (Many-to-One)~~

room\_id(room table) added into guest table

6- payment\_bill (bill\_id ,payment\_id) (many-to-many)

remains

1. guest\_payment (TC/Passport no, payment\_ID) (One-to-Many)

remains because relation is not total

1. guest\_reservation (TC/Passport no, reservation\_ID) (One-to-Many)

remains because relation is not total

1. customer\_reservation (customerId , reservationId) (One-to-Many)

remains because relation is not total

1. customer\_payment (paymentId, CustomerId) (One-to-Many)

remains because relation is not total

1. ~~dinner\_reservation\_bill (bill\_Id , reservation\_Id) (One-to-One)~~

bill\_id(bill table) added into Dinner\_Reservation table

1. ~~dinner\_table\_reservation (reservation\_ID, Table\_ID) (Many-to-One)~~

table\_ID (Dinner\_Table table) added into Dinner\_Reservation table

1. ~~dinner\_table\_order (table\_id, order\_name, order\_date )(One-to-Many)~~

table\_ID (Dinner\_Table table) added into Order table

1. product\_order (order\_name, order\_date, product\_id) (Many-to-Many)

remains

1. ~~hotel\_staff (hotel\_name, district\_name, city\_name , TC/Passport no) (One-to-Many)~~

hotel\_name, district\_name, city\_name(Hotel table) is added into staff table

1. ~~hotel\_supply (hotel\_name , supply\_id) (One-to-Many)~~

hotel\_name, district\_name, city\_name (Hotel table)is added into supply table

1. room\_staff (room\_id, TC/Passport no)(Many-to-Many)

remains

1. guests (hotel\_name , TC/Passport no) (Many-to-Many)

remains