

Hotel Booking and Management System

By Group (4)

Aram Mohammad Alhejaili (4451279)

Atheer Abdullah Mutaq Alharbi (4450980)

Razan Hamdan Almuzaini (4451389)

Supervised By

DR. LIYAKATHUNISA ANSARI

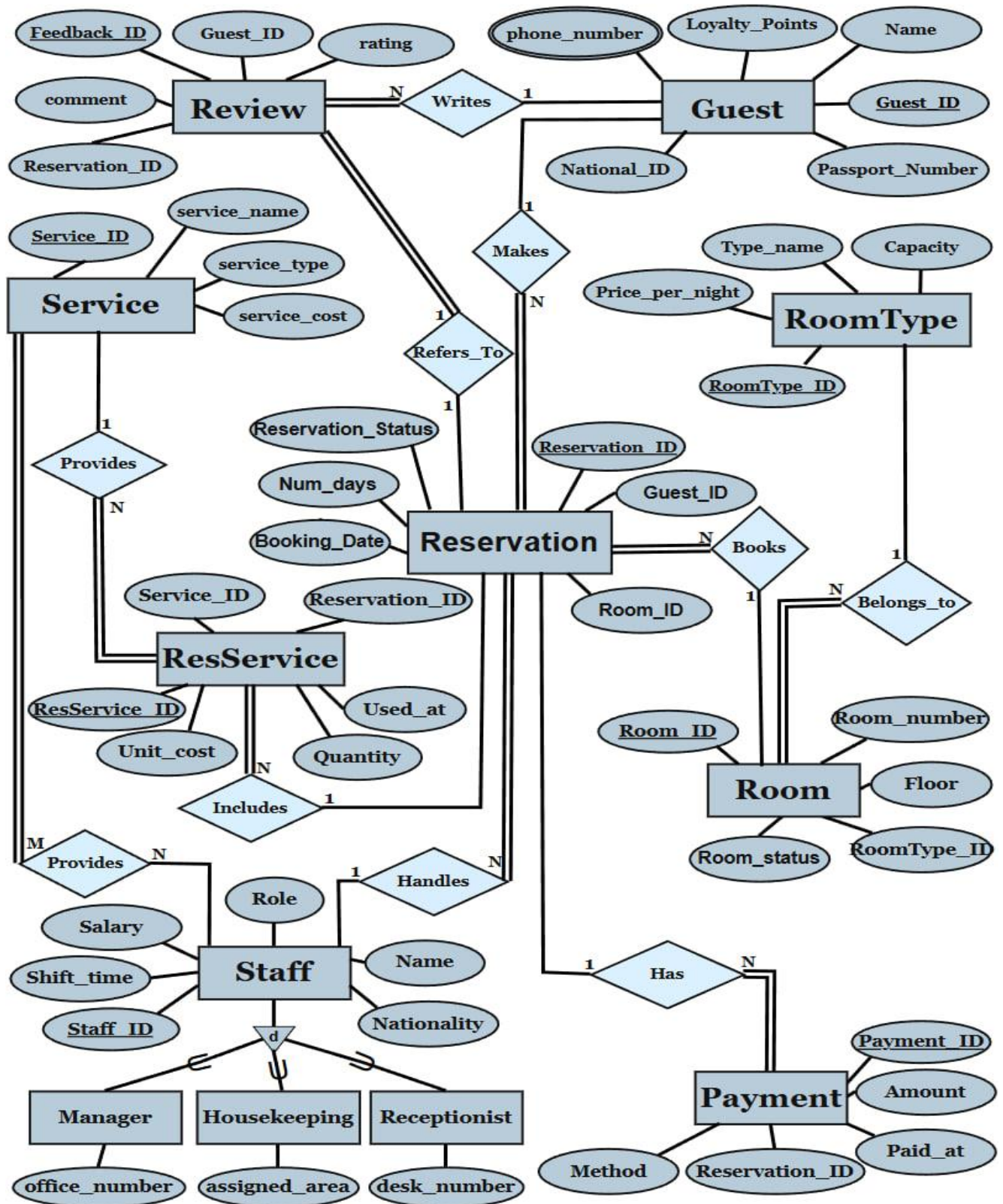
1. Introduction

Modern hotels rely on well-structured database systems to manage core operations such as reservations, guest information, room availability, services, staff, and payment transactions efficiently. Hotels deal with large volumes of data on a daily basis, and using manual methods or poorly designed systems to manage this data can result in redundancy, inconsistencies, and operational inefficiencies that negatively affect service quality and overall performance.

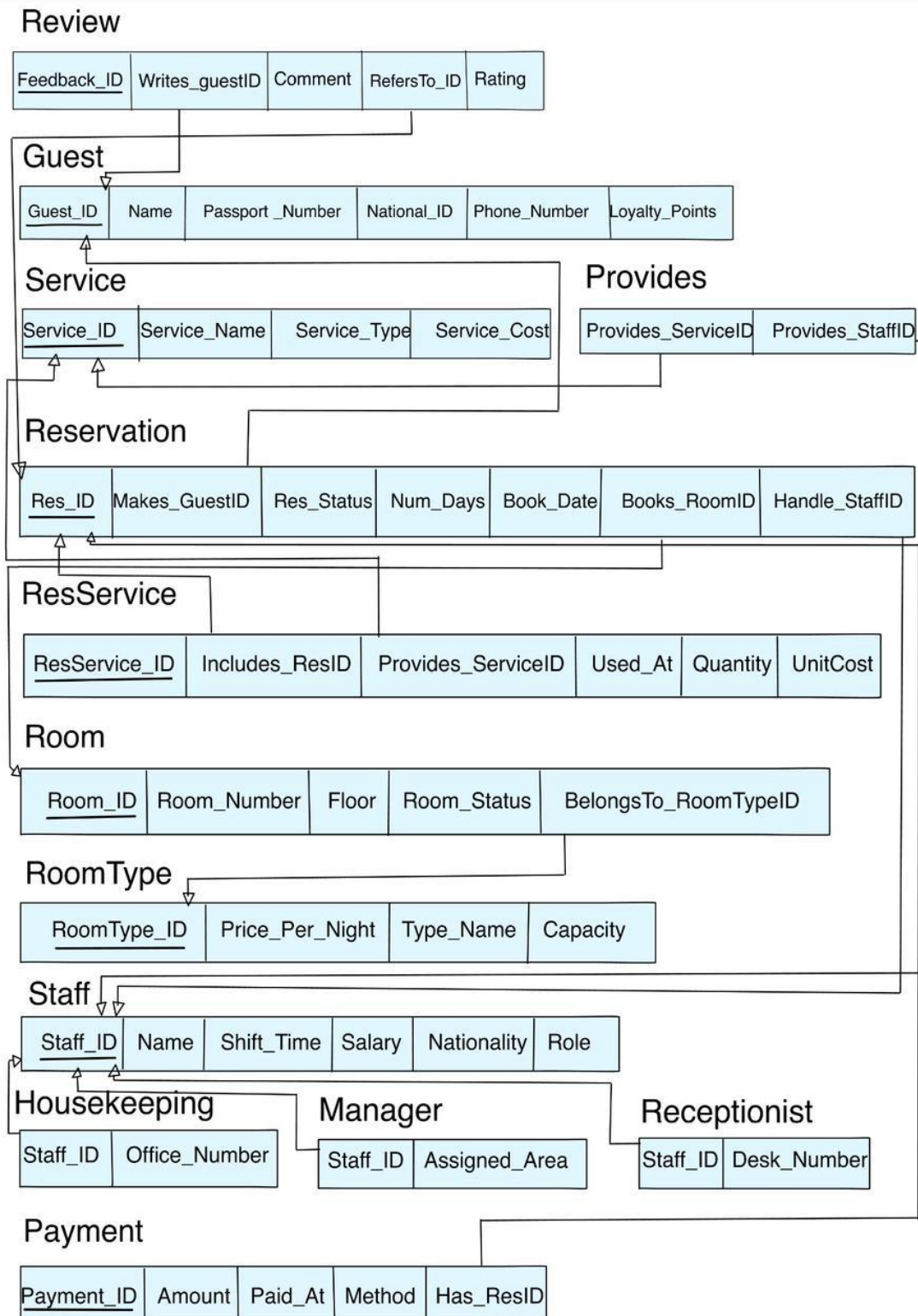
This project focuses on the problem domain of hotel operations by proposing a Hotel Booking and Management System designed to organize and manage hotel data in a structured and integrated manner. The system supports key hotel activities including room reservations, check-in and check-out tracking, room availability management, service usage recording, and payment processing. By defining essential entities such as guests, rooms, reservations, services, staff, and payments, the system ensures accurate data storage and clear relationships between data components.

The main objective of this project is to design an effective data management solution that meets the operational needs of a hotel. The proposed system supports guests by enabling them to make reservations, select appropriate room types, use hotel services, and submit reviews. Additionally, it assists hotel staff in managing bookings, maintaining room status, and recording payments in an organized manner. Overall, the system improves data consistency, reduces redundancy, and provides reliable access to information, contributing to improved operational efficiency and better service quality.

2. Conceptual Model Design (EER Diagram)



3. Relational Model Design



ER-to-Relational Scheme (Mapping algorithm)

Step 1: Mapping of Regular Entity Types

- We create the relations REVIEW, GUEST, SERVICE, RESERVATION, RESSERVICR, ROOM, ROOMTYPE, STAFF, PAYMENT in the relational schema corresponding to the regular entities in the ER diagram.
- Feedback_ID, Guest_ID, Service_ID, Reservation_ID, RoomType_ID, Room_ID, Staff_ID, ResService_ID, Payment_ID are the primary keys for relations.

Step 2: Mapping of Binary 1:1 Relation Types

- 1:1 relation Refers_To is mapped by choosing the participating entity type REVIEW, because its participation in the Refers_To relationship type is total.

Step 3: Mapping of Binary 1:N Relationship Types

(Identify the relation that represent the participating entity type at the N-side of the relationship type)

- For WRITES we include the primary key guest_ID of the GUEST relation as foreign key in the REVIEW relation and call it Writes_guestID.
- For MAKES we include the primary key guest_ID of the GUEST relation as foreign key in the RESERVATION relation and call it Makes_guestID.
- For REFERS_TO we include the primary key Reservation_ID of the RESERVATION relation as foreign key in the REVIEW relation and call it RefersTo_ID.
- For PROVIDES we include the primary key Service_ID of the Service relation as foreign key in the ResService relation and call it Provides_ServiceID.
- For BOOKS we include the primary key ROOM_ID of the ROOM relation as foreign key in the Reservation relation and call it Books_RoomID
- For BELONG_To we include the primary key roomType_ID of the ROOMTYPE relation as foreign key in the ROOM relation and call it BelongTo_RoomTypeID.
- For INCLUDES we include the primary key reservation_ID of the RESERVATION relation as foreign key in the RESSERVICES relation and call it includes_ResID
- For HANDLES we include the primary key staff_ID of the STAFF relation as foreign key in the RESERVATION relation and call it Handle_StaffID

Step 4: Mapping of Binary M:N Relationship Types

- The M:N relation type Provides from ER diagram is mapped by creating a relation PROVIDES in the relation scheme, the primary keys of the STAFF and SERVICE relations are included as foreign keys in PROVIDES and renamed Provides_ServiceID and Provides_StaffID

Step 5: Mapping of Generalisation and Specialisation Hierarchies to a Relation Schema, option 8A: Multiple relations-Superclass and subclasses

- We create multiple relations of subclasses (housekeeping, Manager, Reception) and include primary key (Staff_ID) of superclass as foreign key in subclasses.

4. Database Implementation and SQL Queries

- Create the database using CREATE TABLE statements.

```
CREATE TABLE Guest(
    Guest_ID NUMBER(10) CONSTRAINT G_PK PRIMARY KEY ,
    National_ID VARCHAR(20),
    Guest_Name VARCHAR(25) CONSTRAINT Name_NN NOT NULL,
    Passport_Number VARCHAR(20),
    Phone_Number VARCHAR(10),
    Loyalty_Points NUMBER(10));

CREATE TABLE RoomType (
    RoomType_ID NUMBER(10) CONSTRAINT R_PK PRIMARY KEY ,
    Type_Name VARCHAR(50),
    Price_Per_Night NUMBER(10,2),
    Room_Capacity NUMBER(10));

CREATE TABLE Room (
    Room_ID NUMBER(10) CONSTRAINT RID_PK PRIMARY KEY,
    Room_Number NUMBER(10),
    Floor_Number NUMBER(10),
    Room_Status VARCHAR2(25) CHECK (Room_Status IN ('Available', 'Occupied', 'Maintenance')),
    RoomType_ID NUMBER(10)); ALTER TABLE Room ADD CONSTRAINT R_FK FOREIGN KEY (RoomType_ID) REFERENCES RoomType(RoomType_ID);

CREATE TABLE Staff (
    Staff_ID NUMBER(10) CONSTRAINT Staff_ID_PK PRIMARY KEY,
    Staff_Name VARCHAR(25),
    Staff_Nationality VARCHAR(50),
    Staff_Salary NUMBER(10,2),
    Staff_Role VARCHAR2(25) CHECK (Staff_Role IN ('Manager', 'Receptionist', 'Housekeeping')),
    Shift_Time VARCHAR2(25) CHECK (Shift_Time IN ('Morning', 'Evening', 'Night')));

CREATE TABLE Reservation (
    Reservation_ID NUMBER(10) CONSTRAINT Reservation_PK PRIMARY KEY,
    Guest_ID NUMBER(10),
    Room_ID NUMBER(10),
    Booking_Date DATE,
    Num_Days NUMBER(10),
    Reservation_Status VARCHAR2(25) CHECK ( Reservation_Status IN('confirmed', 'checked_in', 'checked_out', 'cancelled')),
    Handle_StaffID NUMBER(10));
ALTER TABLE Reservation ADD CONSTRAINT R2_FK FOREIGN KEY (Guest_ID) REFERENCES Guest(Guest_ID);
ALTER TABLE Reservation ADD CONSTRAINT R3_FK FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID);
ALTER TABLE Reservation ADD CONSTRAINT S1_FK FOREIGN KEY (Handle_StaffID) REFERENCES Staff(Staff_ID);

CREATE TABLE Payment (
    Payment_ID NUMBER(10) CONSTRAINT Payment_PK PRIMARY KEY,
    Amount NUMBER(10,2),
    Paid_At DATE,
    Payment_Method VARCHAR2(25) CHECK ( Payment_Method IN('Cash', 'Card', 'Online')),
    Has_ResID NUMBER(10)); ALTER TABLE Payment ADD CONSTRAINT H1_FK FOREIGN KEY (Has_ResID) REFERENCES Reservation(Reservation_ID);

CREATE TABLE Service_Hotel (Service_ID NUMBER(10) CONSTRAINT Service_pk PRIMARY KEY,
    Service_Name VARCHAR(50),
    Service_Type VARCHAR(50),
    Service_Cost NUMBER(10,2));

CREATE TABLE ResService ( ResService_ID NUMBER(10) CONSTRAINT ResService_PK PRIMARY KEY,
    Includes_ResID NUMBER(10),
    Provides_ServiceID NUMBER(10),
    Used_At DATE,
    Quantity NUMBER(10),
    UnitCost NUMBER(10,2)); ALTER TABLE ResService ADD CONSTRAINT I1_FK FOREIGN KEY (Includes_ResID) REFERENCES Reservation(Reservation_ID);
ALTER TABLE ResService ADD CONSTRAINT P1_FK FOREIGN KEY (Provides_ServiceID) REFERENCES Service_Hotel(Service_ID);

CREATE TABLE Manager_ ( Staff_ID NUMBER(10),
    Office_Number NUMBER(10)); ALTER TABLE Manager_ ADD CONSTRAINT M1_FK FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID);

CREATE TABLE Receptionist (Staff_ID NUMBER(10),
    Desk_Number NUMBER(10)); ALTER TABLE Receptionist ADD CONSTRAINT D1_FK FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)

CREATE TABLE Housekeeping (Staff_ID NUMBER(10),
    Assigned_Area VARCHAR(50)); ALTER TABLE Housekeeping ADD CONSTRAINT HK_FK FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID);

CREATE TABLE Review ( Feedback_ID NUMBER(10) CONSTRAINT F1_PK PRIMARY KEY,
    Guest_ID NUMBER(10),
    Reservation_ID NUMBER(10),
    Guest_Comment VARCHAR(150),
    Rating NUMBER(10) CHECK (Rating BETWEEN 1 AND 5));
ALTER TABLE Review ADD CONSTRAINT RG_FK FOREIGN KEY (Guest_ID) REFERENCES Guest(Guest_ID);
ALTER TABLE Review ADD CONSTRAINT RRI_FK FOREIGN KEY (Reservation_ID) REFERENCES Reservation(Reservation_ID);

CREATE TABLE Provides (
    Provides_ServiceID NUMBER(10),
    Provides_StaffID NUMBER(10));
ALTER TABLE Provides ADD CONSTRAINT Provides_Service_FK FOREIGN KEY (Provides_ServiceID) REFERENCES Service_Hotel(Service_ID);
ALTER TABLE Provides ADD CONSTRAINT Provides_Staff_FK FOREIGN KEY (Provides_StaffID) REFERENCES Staff(Staff_ID);
```

- Populate each table with a sufficient number of sample rows.

```

INSERT INTO Guest VALUES
(1,'112233098','Abdullah Alharbi','P12345','0502689144',45);
INSERT INTO Guest VALUES
(2,'1030987665','Fay Aldosari','P54321','0559988776',10);
INSERT INTO Guest VALUES
(3,'1123084445','Rayan Almutairi','P54344','0560622300',25);

INSERT INTO RoomType VALUES
(1,'Standard Single',200,2);
INSERT INTO RoomType VALUES
(2,'Deluxe Double',300,3);
INSERT INTO RoomType VALUES
(3,'Executive Suite',450,5);

INSERT INTO Room VALUES
(101,101,1,'Available',1);
INSERT INTO Room VALUES
(102,102,2,'Occupied',2);
INSERT INTO Room VALUES
(202,202,2,'Maintenance',1);
INSERT INTO Room VALUES
(301,301,3,'Occupied',3);
INSERT INTO Room VALUES
(302,302,3,'Available',2);

INSERT INTO Staff VALUES
(1,'Mohammed Alamri','Saudi',6500,'Manager','Morning');
INSERT INTO Staff VALUES
(2,'Huda Alharbi','Saudi',4500,'Receptionist','Evening');
INSERT INTO Staff VALUES
(3,'Mohammad Ali','India',3000,'Housekeeping','Morning');

INSERT INTO Manager_ VALUES (1,10);
INSERT INTO Receptionist VALUES (2,5);
INSERT INTO Housekeeping VALUES (3,'East Wing');
INSERT INTO Reservation VALUES
(1,1,101,TO_DATE('2025-10-10','YYYY-MM-DD'),3,'confirmed',2);
INSERT INTO Reservation VALUES
(2,2,102,TO_DATE('2025-01-15','YYYY-MM-DD'),2,'checked_in',1);
INSERT INTO Service_Hotel VALUES
(1,'Room Cleaning','Cleaning',50);
INSERT INTO Service_Hotel VALUES
(2,'Spa Session','Spa',200);
INSERT INTO ResService VALUES
(1,1,1,TO_DATE('2025-02-24 10:00','YYYY-MM-DD HH24:MI'),1,50);
INSERT INTO ResService VALUES
(2,2,2,TO_DATE('2025-01-16 15:00','YYYY-MM-DD HH24:MI'),1,200);
INSERT INTO Payment VALUES
(1,600,TO_DATE('2025-10-10 12:00','YYYY-MM-DD HH24:MI'),'Card',1);
INSERT INTO Payment VALUES
(2,900,TO_DATE('2025-01-16 18:00','YYYY-MM-DD HH24:MI'),'Cash',2);
INSERT INTO Review VALUES
(1,1,1,'Great stay, The room was clean and the staff were very friendly. I enjoyed the service a lot.',5);
INSERT INTO Review VALUES
(2,2,2,'Good but expensive. The room was nice and comfortable, but I think the price was a bit high for the services provided.',4);
INSERT INTO Provides VALUES (1,3);
INSERT INTO Provides VALUES (2,1);

```


- Basic retrievals using SELECT-FROM-WHERE

```
// Retrieve all available rooms
SELECT Room_Number, Room_Status
FROM Room
WHERE Room_Status = 'Available';
```

ROOM_NUMBER	ROOM_STATUS
1	101 Available
2	302 Available

```
//Retrieve staff members working the Morning shift
SELECT Staff_Name, Shift_Time
FROM Staff
WHERE Shift_Time = 'Morning';
```

STAFF_NAME	SHIFT_TIME
1 Mohammed Alamri	Morning
2 Mohammad Ali	Morning

- Aggregation and grouping (GROUP BY, HAVING)

```
//Count number of reservations per guest
SELECT Guest_ID, COUNT(*) AS Total_Reservations
FROM Reservation
GROUP BY Guest_ID;
```

GUEST_ID	TOTAL_RESERVATIONS
1	1
2	2

```
//Retrieve reservations where total payments exceed 500
SELECT Has_ResID AS Reservation_ID, SUM(Amount) AS Total_Paid
FROM Payment
GROUP BY Has_ResID
HAVING SUM(Amount) > 500;
```

RESERVATION_ID	TOTAL_PAID
1	600
2	900

- Sorting results in ascending/descending order

```
//Retrieve rooms ordered by floor (descending)
SELECT Room_Number, Floor_Number
FROM Room
ORDER BY Floor_Number DESC;
```

ROOM_NUMBER	FLOOR_NUMBER
1	302
2	301
3	102
4	202
5	101

```
//Retrieve staff ordered by salary (ascending)
SELECT Staff_Name, Staff_Salary
FROM Staff
ORDER BY Staff_Salary ASC;
```

STAFF_NAME	STAFF_SALARY
1 Mohammad Ali	3000
2 Huda Alharbi	4500
3 Mohammed Alamri	6500

- Joins between multiple tables (INNER JOIN, LEFT JOIN)

```
//Retrieve reservations with guest names
SELECT Reservation.Reservation_ID, Guest.Guest_Name
FROM Reservation
JOIN Guest ON Reservation.Guest_ID = Guest.Guest_ID;
```

RESERVATION_ID	GUEST_NAME
1	1 Abdullah Alharbi
2	2 Fay Aldosari

```
//Retrieve all rooms including those with no reservations (LEFT JOIN)
SELECT Room.Room_Number, Reservation.Reservation_ID
FROM Room
LEFT JOIN Reservation ON Room.Room_ID = Reservation.Room_ID;
```

ROOM_NUMBER	RESERVATION_ID
1	101
2	102
3	302 (null)
4	202 (null)
5	301 (null)

- Subqueries and nested queries

```
//Retrieve employees who handle reservations
SELECT Staff_Name
FROM Staff
WHERE Staff_ID IN (
    SELECT Handle_StaffID
    FROM Reservation);
```

STAFF_NAME
1 Mohammed Alamri
2 Huda Alharbi

5. Conclusion

In this project, We designed and implemented a database system for a Hotel Booking and Management System. We began by analyzing the requirements of real hotel operations, identifying the essential entities needed in the system. We then translated these requirements into an EER diagram, followed by a well-structured relational schema.

During the SQL implementation phase, we created and populated the database using DDL and DML commands, applied integrity constraints, and executed a variety of SQL queries including joins, aggregations, sorting, and conditional retrieval. This process helped us confirm that our design works correctly and showed that the system can handle real tasks.

Team Meetings & Task Allocation

Topics Discussed	Meeting date
Choose project topic. Understanding project requirements. Discussing main entities and system scope	4 Nov
Reviewing EER Diagram	10 Nov
Reviewing Relational Schema	17 Nov
Reviewing and ensuring consistency between EER, schema, and SQL. Writing the conclusion. Reviewing final report	27 Nov

Member	Tasks Completed
Aram	Introduction + EER Diagram
Razan	Relational Schema + Conclusion
Atheer	Implementation SQL (DDL, DML, queries, constraints)