--Hotel Management System

```
--The Hotel Management Database System streamlines hotel
--operations by managing room availability, guest
--information, bookings, and revenue.
--It features tables for hotels, rooms, guests, and bookings,
--with relationships ensuring data integrity. Key
--functionalities include listing hotel details, managing room
--types and prices, tracking guest stays, and calculating
--revenue.
--The system supports essential SQL operations, including
--joins for complex queries and backup and restore
--commands to ensure data security.
--This centralized, efficient database system enhances hotel
--management by providing robust data handling and
--comprehensive reporting capabilities

-- Create the database
CREATE DATABASE HOTELE_MANAGEMENT;

-- Switch to the database
USE HOTELE_MANAGEMENT;

-- Create the HOTEL table
CREATE TABLE HOTEL (
   hotel_No VARCHAR(12) PRIMARY KEY NOT NULL,
   Name VARCHAR(50) NOT NULL,
   H_Address VARCHAR(100) NOT NULL
);

-- Create the Room table
CREATE TABLE Room (
   Room_No VARCHAR(22) PRIMARY KEY NOT NULL,
    R_type VARCHAR(20) NOT NULL,
    price FLOAT NOT NULL,
    hotel_No VARCHAR(12) NOT NULL,
    FOREIGN KEY (hotel_No) REFERENCES HOTEL(hotel_No)
);

-- Create the Guest table
CREATE TABLE Guest (
    Guest_no VARCHAR(21) PRIMARY KEY NOT NULL,
    Name VARCHAR(50) NOT NULL,
    Guest_address VARCHAR(255) NOT NULL
);

-- Create the Booking table
CREATE TABLE Booking (
    hotel_No VARCHAR(12) NOT NULL,
    Guest_no VARCHAR(21) NOT NULL,
    Date_From DATE NOT NULL,
    Date_To DATE NOT NULL,
    Room_No VARCHAR(22) NOT NULL,
    PRIMARY KEY (hotel_No, Guest_no, Date_From, Room_No),
    FOREIGN KEY (hotel_No) REFERENCES HOTEL(hotel_No),
    FOREIGN KEY (Guest_no) REFERENCES Guest(Guest_no),
    FOREIGN KEY (Room_No) REFERENCES Room(Room_No)
);

-- Insert data into HOTEL table
INSERT INTO HOTEL (hotel_No, Name, H_Address) VALUES
('H1', 'Sheraton Addis', 'Addis Ababa'),
('H2', 'Hilton', 'Addis Ababa'),
('H3', 'Jupiter', 'Addis Ababa'),
('H4', 'Haile Resort', 'Awasa'),
('H5', 'Kuriftu', 'Deberezit');

-- Insert data into Room table
```

```
INSERT INTO Room (Room_No, R_type, price, hotel_No) VALUES
('R1', 'single', 1000, 'H1'),
('R10', 'VIP family', 6000, 'H1'),
('R11', 'VIP single', 4000, 'H1'),
('R13', 'single', 1000, 'H3'),
('R2', 'family', 2000, 'H1'),
('R3', 'family', 3000, 'H1'),
('R4', 'family', 2000, 'H2'),
('R5', 'family', 2500, 'H2'),
('R6', 'family', 2500, 'H4'),
('R7', 'family', 3000, 'H4'),
('R8', 'single', 1500, 'H4'),
('R9', 'single', 1500, 'H5');

-- Insert data into Guest table
INSERT INTO Guest (Guest_no, Name, Guest_address) VALUES
('G1', 'Abel Efrem', 'A.A'),
('G2', 'Hana Zewde', 'A.A'),
('G3', 'Mulualem Awel', 'Jima'),
('G4', 'Cheru Lemma', 'A.A'),
('G5', 'Sintayehu Geremew', 'A.A'),
('G6', 'Sileshi Tsegaye', 'Bahirdar'),
('G7', 'Teklay Nigus', 'Bahirdar'),
('G8', 'Eskadimas Melaku', 'A.A');

-- Insert data into Booking table
INSERT INTO Booking (hotel_No, Guest_no, Date_From, Date_To,
Room_No) VALUES
('H1', 'G3', '2024-06-02', '2024-06-08', 'R1'),
('H1', 'G6', '2024-06-02', '2024-06-07', 'R2'),
('H1', 'G1', '2024-06-02', '2024-06-06', 'R3'),
('H2', 'G2', '2024-06-02', '2024-06-08', 'R4'),
('H2', 'G4', '2024-06-02', '2024-06-08', 'R5'),
('H4', 'G7', '2024-06-02', '2024-06-08', 'R6'),
('H4', 'G5', '2024-06-02', '2024-06-08', 'R7'),
('H5', 'G1', '2024-06-02', '2024-06-08', 'R8'),
('H1', 'G1', '2024-06-02', '2024-06-15', 'R9'),
('H1', 'G2', '2024-06-02', '2024-06-15', 'R10');


-- Add foreign key to the Room table to reference the HOTEL table
ALTER TABLE Room
ADD CONSTRAINT FK_Hotel_Room
FOREIGN KEY (hotel_No) REFERENCES HOTEL(hotel_No);

-- Add foreign keys to the Booking table to reference the HOTEL, Guest, and Room tables
ALTER TABLE Booking
ADD CONSTRAINT FK_Booking_Hotel
FOREIGN KEY (hotel_No) REFERENCES HOTEL(hotel_No);

ALTER TABLE Booking
ADD CONSTRAINT FK_Booking_Guest
FOREIGN KEY (Guest_no) REFERENCES Guest(Guest_no);

ALTER TABLE Booking
ADD CONSTRAINT FK_Booking_Room
FOREIGN KEY (Room_No) REFERENCES Room(Room_No);


-- List full Detail of all Hotel
		--To list the full details of all hotels, you can use a simple SELECT statement.
		--Here is the SQL query that will retrieve all columns from the HOTEL table:


SELECT * FROM HOTEL;

--If you want to see a more detailed view, for example
```

```sql
--including related information from other tables,
--such as the number of rooms each hotel has, you could use a JOIN query.

SELECT
    h.hotel_No,
    h.Name,
    h.H_Address,
    COUNT(r.Room_No) AS NumberOfRooms
FROM
    HOTEL h
LEFT JOIN
    Room r ON h.hotel_No = r.hotel_No
GROUP BY
    h.hotel_No, h.Name, h.H_Address;

--This query will give you the hotel details along with the number of rooms each hotel has.

--List full detail of all Hotel in Addis Ababa

SELECT *
FROM HOTEL
WHERE H_Address = 'Addis Ababa';

--This query selects all columns from the HOTEL table where the H_Address is 'Addis Ababa'.

SELECT hotel_No, Name, H_Address
FROM HOTEL
WHERE H_Address = 'Addis Ababa';

--This query achieves the same result by explicitly listing the columns to retrieve.
```

```
--If you want to include additional information, such as the number of rooms
-- each hotel has in Addis Ababa, you can use a JOIN with a GROUP BY:

SELECT
    h.hotel_No,
    h.Name,
    h.H_Address,
    COUNT(r.Room_No) AS NumberOfRooms
FROM
    HOTEL h
LEFT JOIN
    Room r ON h.hotel_No = r.hotel_No
WHERE
    h.H_Address = 'Addis Ababa'
GROUP BY
    h.hotel_No, h.Name, h.H_Address;

--This query lists the details of all hotels in Addis Ababa and includes the number of rooms each hotel has.


--List all familly Rooms with price below 3000?

SELECT *
FROM Room
WHERE R_type = 'family' AND price < 3000;

--This query selects all columns from the Room table
--where the room type is 'family' and the price is below 3000


--List the price and type of all Rooms at the Sheraton Addis
```

```sql
SELECT
   r.price,
   r.R_type
FROM
   Room r
JOIN
   HOTEL h ON r.hotel_No = h.hotel_No
WHERE
   h.Name = 'Sheraton Addis';



--SELECT r.price, r.R_type: Selects the price and room type from the Room table.
--FROM Room r: Specifies the Room table with an alias r.
--JOIN HOTEL h ON r.hotel_No = h.hotel_No: Joins the Room table with the HOTEL table based on the hotel_No column.
--WHERE h.Name = 'Sheraton Addis': Filters the results to include only the rooms in the "Sheraton Addis" hotel.


--List the name and address of all Guest in Addis Ababa, Alphabetically Order by Name?
SELECT
   Name,
   Guest_address
FROM
   Guest
WHERE
   Guest_address = 'A.A'
ORDER BY
   Name ASC;
```

```sql
        --This query will provide the name and address of all guests in Addis Ababa,
        --ordered alphabetically by their names.

--What is the total revenue per night from all familly Rooms in Sheraton Addis

SELECT
   SUM(r.price) AS TotalRevenuePerNight
FROM
   Room r
JOIN
   HOTEL h ON r.hotel_No = h.hotel_No
WHERE
   h.Name = 'Sheraton Addis' AND r.R_type = 'family';

--This query will give you the total revenue per night from all family rooms in the Sheraton Addis.

--List all guests Currently staying at the Sheraton Addis?

SELECT
   g.Guest_no,
   g.Name,
   g.Guest_address
FROM
   Guest g
JOIN
   Booking b ON g.Guest_no = b.Guest_no
JOIN
   HOTEL h ON b.hotel_No = h.hotel_No
WHERE
   h.Name = 'Sheraton Addis'
   AND GETDATE() BETWEEN b.Date_From AND b.Date_To;
```

```sql
--SELECT g.Guest_no, g.Name, g.Guest_address: Selects the guest
number, name, and address from the Guest table.
--FROM Guest g: Specifies the Guest table with an alias g.
--JOIN Booking b ON g.Guest_no = b.Guest_no: Joins the Guest table
with the Booking table based on the Guest_no column.
--JOIN HOTEL h ON b.hotel_No = h.hotel_No: Joins the Booking table
with the HOTEL table based on the hotel_No column.
--WHERE h.Name = 'Sheraton Addis' AND GETDATE() BETWEEN
b.Date_From AND b.Date_To:
--Filters the results to include only guests staying at the "Sheraton
Addis" where the current date is between the booking start and end
dates.

SELECT GETDATE() AS CurrentDateTime;
SELECT *
FROM Booking
WHERE hotel_No IN (SELECT hotel_No FROM HOTEL WHERE
Name = 'Sheraton Addis');

--List the Name and Address of unreserved Hotel?
SELECT
    h.Name,
    h.H_Address
FROM
    HOTEL h
LEFT JOIN
    Booking b ON h.hotel_No = b.hotel_No
WHERE
    b.hotel_No IS NULL;

         --This query will provide the names and addresses of all hotels that
have no current reservations in your database
```

```
--Count the number of both familly and single Rooms in each Hotel?

SELECT
    h.hotel_No,
    h.Name,
    COUNT(CASE WHEN r.R_type = 'familly' THEN 1 END) AS
FamilyRoomCount,
    COUNT(CASE WHEN r.R_type = 'single' THEN 1 END) AS
SingleRoomCount
FROM
    HOTEL h
LEFT JOIN
    Room r ON h.hotel_No = r.hotel_No
GROUP BY
    h.hotel_No, h.Name;
--Explantion--
--SELECT h.hotel_No, h.Name: Selects the hotel number and name
from the HOTEL table.
--COUNT(CASE WHEN r.R_type = 'familly' THEN 1 END) AS
FamilyRoomCount: Counts the number of family rooms in each hotel.
--COUNT(CASE WHEN r.R_type = 'single' THEN 1 END) AS
SingleRoomCount: Counts the number of single rooms in each hotel.
--FROM HOTEL h: Specifies the HOTEL table with an alias h.
--LEFT JOIN Room r ON h.hotel_No = r.hotel_No: Performs a left join
between the HOTEL table and the Room table based on the hotel_No
column. This join includes all hotels, even those without corresponding
rooms.
--GROUP BY h.hotel_No, h.Name: Groups the results by hotel number
and name to ensure that the counts are calculated per hotel.

--List all Rooms inaccurately assign for a Hotel in booking table
SELECT
    b.hotel_No AS BookingHotel_No,
    b.Room_No,
    r.hotel_No AS RoomHotel_No
```

```sql
FROM
    Booking b
JOIN
    Room r ON b.Room_No = r.Room_No
WHERE
    b.hotel_No <> r.hotel_No;


--SELECT b.hotel_No AS BookingHotel_No, b.Room_No, r.hotel_No AS RoomHotel_No: Selects the hotel_No from the Booking table (aliased as BookingHotel_No), the room number, and the hotel_No from the Room table (aliased as RoomHotel_No).
--FROM Booking b: Specifies the Booking table with an alias b.
--JOIN Room r ON b.Room_No = r.Room_No: Joins the Booking table with the Room table based on the Room_No column.
--WHERE b.hotel_No <> r.hotel_No: Filters the results to include only those rows where the hotel_No in the Booking table does not match the hotel_No in the Room table.
--This query will help you identify all rooms that are inaccurately assigned to a hotel in the Booking table by checking for mismatches in the hotel_No between the Booking and Room tables.

--list for how many day Guest stay in a Hotel, room number and room type?

SELECT
    b.Guest_no,
    b.hotel_No,
    b.Room_No,
    r.R_type,
    DATEDIFF(DAY, b.Date_From, b.Date_To) AS NumberOfDays
FROM
    Booking b
JOIN
    Room r ON b.Room_No = r.Room_No;
```

```sql
	--This query will provide the number of days each guest stays in a hotel, along with the room number and room type.


--Join all table by INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN and describe their deference

		--INNER JOIN
		--An INNER JOIN returns only the rows that have matching values in both tables

SELECT
    h.hotel_No, h.Name AS HotelName, h.H_Address,
    r.Room_No, r.R_type, r.price,
    g.Guest_no, g.Name AS GuestName, g.Guest_address,
    b.Date_From, b.Date_To
FROM
    HOTEL h
INNER JOIN
    Room r ON h.hotel_No = r.hotel_No
INNER JOIN
    Booking b ON h.hotel_No = b.hotel_No AND r.Room_No = b.Room_No
INNER JOIN
    Guest g ON b.Guest_no = g.Guest_no;



		--LEFT JOIN
		--A LEFT JOIN returns all rows from the left table and the matched rows from the right table.
		-- If no match is found, NULL values are returned for columns from the right table
```

```
SELECT
   h.hotel_No, h.Name AS HotelName, h.H_Address,
   r.Room_No, r.R_type, r.price,
   g.Guest_no, g.Name AS GuestName, g.Guest_address,
   b.Date_From, b.Date_To
FROM
   HOTEL h
LEFT JOIN
   Room r ON h.hotel_No = r.hotel_No
LEFT JOIN
   Booking b ON h.hotel_No = b.hotel_No AND r.Room_No =
b.Room_No
LEFT JOIN
   Guest g ON b.Guest_no = g.Guest_no;

              --RIGHT JOIN
              --A RIGHT JOIN returns all rows from the right table and the
matched rows from the left table.
              --If no match is found, NULL values are returned for
columns from the left table

SELECT
   h.hotel_No, h.Name AS HotelName, h.H_Address,
   r.Room_No, r.R_type, r.price,
   g.Guest_no, g.Name AS GuestName, g.Guest_address,
   b.Date_From, b.Date_To
FROM
   HOTEL h
RIGHT JOIN
   Room r ON h.hotel_No = r.hotel_No
RIGHT JOIN
   Booking b ON r.Room_No = b.Room_No AND h.hotel_No =
b.hotel_No
RIGHT JOIN
   Guest g ON b.Guest_no = g.Guest_no;
```

```
		--FULL JOIN
		--A FULL JOIN returns all rows when there is a match in
either the left or right table.
		-- If there is no match, NULL values are returned for
columns from the table that lacks a matching row.

SELECT
    h.hotel_No, h.Name AS HotelName, h.H_Address,
    r.Room_No, r.R_type, r.price,
    g.Guest_no, g.Name AS GuestName, g.Guest_address,
    b.Date_From, b.Date_To
FROM
    HOTEL h
FULL JOIN
    Room r ON h.hotel_No = r.hotel_No
FULL JOIN
    Booking b ON h.hotel_No = b.hotel_No AND r.Room_No =
b.Room_No
FULL JOIN
    Guest g ON b.Guest_no = g.Guest_no;


--Use INNER JOIN when you only want to return rows that have
matching values in both tables.
--Use LEFT JOIN when you want to return all rows from the left table,
and the matched rows from the right table.
--Use RIGHT JOIN when you want to return all rows from the right
table, and the matched rows from the left table.
--Use FULL JOIN when you want to return all rows from both tables,
regardless of whether there is a match

BACKUP DATABASE HOTELE_MANAGEMENT
TO DISK = '\\DESKTOP-P6853S8\backup\HotelManagement.bak'
```

```
WITH FORMAT,
    MEDIANAME = 'SQLServerBackups',
    NAME = 'Full Backup of HotelManagement';

        RESTORE FILELISTONLY
FROM DISK = '\\DESKTOP-P6853S8\backup\HotelManagement.bak';
```