SQL Project: Hotel Infrastructure Sample Database

This project demonstrates intermediate SQL concepts including:

- Table Creation
- Data Insertion
- Joins
- Sub-queries

The project aims to depict the utilization of joins, sub-queries, and other intermediate SQL concepts in the context of a hotel's infrastructure.

------------- Table Definitions ---------------------------

```
-- Create the Guests table
CREATE TABLE Guests (
    GuestID INT PRIMARY KEY,
    FirstName TEXT,
    LastName TEXT,
    Email TEXT
);
-- Create the Rooms table
CREATE TABLE Rooms (
    RoomID INT PRIMARY KEY,
    RoomType TEXT,
    Price FLOAT
);
-- Create the Bookings table
CREATE TABLE Bookings (
    BookingID INT PRIMARY KEY,
    GuestID INT,
    RoomID INT,
    BookingDate DATE,
    NumberOfNights INT,
    FOREIGN KEY (GuestID) REFERENCES Guests(GuestID),
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)
);
```

-------------------Sample Data Insertion ---------------------

```
-- Insert sample data into the Guests table
INSERT INTO Guests (GuestID, FirstName, LastName, Email) VALUES
(1, 'Alice', 'Smith', 'alice@email.com'),
(2, 'Bob', 'Johnson', 'bob@email.com'),
(3, 'Cindy', 'Williams', 'cindy@email.com');
```

-- Insert sample data into the Rooms table ------------------------

```
INSERT INTO Rooms (RoomID, RoomType, Price) VALUES
(1, 'Single', 100.00),
(2, 'Double', 150.00),
(3, 'Suite', 200.00);
```

-- Insert sample data into the Bookings table ---------------

```
INSERT INTO Bookings (BookingID, GuestID, RoomID, BookingDate, NumberOfNights) VALUES
(1, 1, 1, '2023-01-01', 2),
(2, 2, 2, '2023-01-10', 3),
(3, 3, 3, '2023-01-15', 1),
(4, 1, 3, '2023-02-01', 2),
(5, 2, 1, '2023-02-05', 1);
```

----------------------------SQL Queries---------------------------

-- Query 1: Join all tables and select useful information

```
-- This query shows how guests, their bookings, and the corresponding rooms are related.
```

```
SELECT
    g.GuestID,
    g.FirstName,
    g.LastName,
    r.RoomType,
    r.Price,
    b.BookingDate,
    b.NumberOfNights
FROM
    Guests g
JOIN Bookings b ON g.GuestID = b.GuestID
JOIN Rooms r ON b.RoomID = r.RoomID;
```

-- Query 2: Find the total expenditure for each guest using a sub-query

-- *This query calculates the total expenditure for each guest by summing up the room prices for their bookings.*

```
SELECT
    g.GuestID,
    g.FirstName,
    g.LastName,
    (SELECT SUM(r.Price * b.NumberOfNights)
     FROM Bookings b
     JOIN Rooms r ON b.RoomID = r.RoomID
     WHERE g.GuestID = b.GuestID) AS TotalExpenditure
FROM
    Guests g;
```

-- Query 3: Find guests who have spent more than $300

-- *This query identifies guests whose total expenditure across all bookings is more than $300.*

```
SELECT
```

```
    g.GuestID,
    g.FirstName,
    g.LastName,
    SUM(r.Price * b.NumberOfNights) AS TotalExpenditure
FROM
    Guests g
JOIN Bookings b ON g.GuestID = b.GuestID
JOIN Rooms r ON b.RoomID = r.RoomID
GROUP BY
    g.GuestID, g.FirstName, g.LastName
HAVING
    TotalExpenditure > 300;
```