**Hotel Database Management System**

**Overview**

This project involves the management of hotel guest and booking information using a SQL database. The primary goal is to create a database schema that efficiently stores and retrieves data related to guests and their bookings, and to execute SQL queries to extract useful insights and perform necessary operations.

**Description**

The project consists of two main tables: Guests and Bookings. The Guests table stores personal information about the guests, and the Bookings table stores information about the bookings made by these guests. Several SQL queries have been developed to perform various operations, such as inserting records, finding guest statistics, calculating financial metrics, and generating reports.

**Scope**

1. **Database Schema Design**:
   * Create tables for guests and bookings.
   * Establish relationships between the tables.
2. **Data Insertion**:
   * Insert sample data into the Guests and Bookings tables.
3. **Data Retrieval and Analysis**:
   * Perform SQL queries to retrieve and analyze data.
   * Generate reports based on the data.
4. **Financial Calculations**:
   * Calculate total revenue.
   * Determine the average stay duration.
   * Identify top-spending guests.
5. **Guest and Booking Management**:
   * Identify guests with multiple bookings.
   * List guests based on various criteria.
   * Analyze guest stay patterns.

**Database Schema**

The database used in this project is Hotel.

**Table: Guests**

This table stores information about the guests.

**CREATE TABLE Guests (**

**guest\_id INT PRIMARY KEY AUTO\_INCREMENT,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**email VARCHAR(100),**

**phone\_number VARCHAR(15),**

**address VARCHAR(200),**

**city VARCHAR(50)**

**);**

**Table: Bookings**

This table stores information about the bookings made by guests.

**CREATE TABLE Bookings (**

**booking\_id INT PRIMARY KEY AUTO\_INCREMENT,**

**guest\_id INT,**

**room\_number INT,**

**check\_in\_date DATE,**

**check\_out\_date DATE,**

**amount DECIMAL(10,2),**

**FOREIGN KEY (guest\_id) REFERENCES Guests(guest\_id)**

**);**

**Data Insertion**

The following SQL statements insert sample data into the Guests and Bookings tables.

Insert Guest Records

**INSERT INTO Guests (first\_name, last\_name, email, phone\_number, address, city) VALUES**

**('Varun', 'Kumar', 'varun.kumar@gmail.com', '9876543210', '123 MG Road', 'Bangalore'),**

**('Harsh', 'Sharma', 'harsh.sharma@gmail.com', '8765432109', '456 Park Street', 'Kolkata'),**

**('Sakshi', 'Mehta', 'sakshi.mehta@gmail.com', '7654321098', '789 Nehru Place', 'Delhi'),**

**('Pooja', 'Singh', 'pooja.singh@gmail.com', '6543210987', '101 Marine Drive', 'Mumbai'),**

**('Rahul', 'gupta', 'rahul.gupta@gmail.com', '5432109876', '202 Ring Road', 'Ahmedabad'),**

**('Binu', 'Nair', 'binu.nair@gmail.com', '4321098765', '303 MG Road', 'Bangalore'),**

**('Vikas', 'chauhan', 'vikas.chauhan@gmail.com', '3210987654', '404 Jubilee Hills', 'Mumbai'),**

**('Amit', 'Verma', 'amit.verma@gmail.com', '2109876543', '505 Civil Lines', 'Lucknow'),**

**('Kavita', 'Joshi', 'kavita.joshi@gmail.com', '1098765432', '606 Mall Road', 'Mumbai'),**

**('Neeraj', 'Chopra', 'neeraj.chopra@gmail.com', '1987654321', '707 Residency Road', 'Mumbai');**

Insert Booking Records

**INSERT INTO Bookings (guest\_id, room\_number, check\_in\_date, check\_out\_date, amount) VALUES**

**(1, 101, '2024-05-20', '2024-05-21', 900.00),**

**(2, 102, '2024-05-22', '2024-05-27', 900.00),**

**(2, 104, '2024-05-25', '2024-05-26', 900.00),**

**(3, 103, '2024-05-23', '2024-05-28', 1000.00),**

**(4, 104, '2024-05-24', '2024-05-24', 1000.00),**

**(5, 105, '2024-05-25', '2024-05-30', 1000.00),**

**(6, 106, '2024-05-26', '2024-05-26', 1000.00),**

**(7, 107, '2024-05-27', '2024-06-01', 1200.00),**

**(8, 108, '2024-05-28', '2024-06-02', 1200.00),**

**(9, 109, '2024-05-29', '2024-06-03', 1200.00),**

**(10, 110, '2024-05-30', '2024-06-04', 1200.00);**

**SQL Queries**

Several SQL queries are used to extract useful information from the database.

1. **Find the guest who has made the most bookings.**

WITH GuestBooking AS (

SELECT g.guest\_id ,

CONCAT(g.first\_name , ' ',g.last\_name) AS full\_name,

COUNT(b.booking\_id) AS booking\_count

FROM Guests g

JOIN Bookings b

ON g.guest\_id = b.guest\_id

GROUP BY g.guest\_id, g.first\_name , g.last\_name

),

RankedGuests AS (

SELECT guest\_id ,

full\_name ,

booking\_count,

DENSE\_RANK() OVER(ORDER BY booking\_count DESC ) rnk

FROM GuestBooking

)

SELECT guest\_id , full\_name , booking\_count

FROM RankedGuests

WHERE rnk =1;

1. **List the guests who have bookings from 25-May to 1-June.**

SELECT b.guest\_id,

CONCAT(g.first\_name ,' ', g.last\_name ) AS Guest\_name

FROM Guests g

JOIN Bookings b

ON g.guest\_id = b.guest\_id

WHERE check\_in\_date BETWEEN '2024-05-25' AND '2024-06-01' ;

1. **Find the total revenue generated from all bookings.**

SELECT

SUM(

CASE

WHEN DATEDIFF(check\_out\_date, check\_in\_date) = 0 THEN 1

ELSE DATEDIFF(check\_out\_date, check\_in\_date)

END \* amount

) AS total\_revenue

FROM Bookings;

1. **Find the average stay duration of guests.**

WITH Stay AS (

SELECT

CASE

WHEN DATEDIFF(check\_out\_date , check\_in\_date) = 0

THEN 1

ELSE DATEDIFF(check\_out\_date , check\_in\_date)

END AS stay\_duration

FROM

Bookings

)

SELECT

ROUND(AVG(stay\_duration), 2) AS average\_stay\_duration

FROM

Stay;

1. **Find the guest who booked the same room multiple times.**

SELECT CONCAT(g.first\_name, ' ' , g.last\_name) AS full\_name ,

b.guest\_id , b.room\_number,

COUNT(\*) AS Booking\_count

FROM Bookings b

JOIN Guests g

ON b.guest\_id = g.guest\_id

GROUP BY

g.first\_name , g.last\_name , b.guest\_id , b.room\_number

HAVING COUNT(\*) > 1;

1. **List the top 2 guests by total amount spent.**

SELECT G.first\_name, G.last\_name, SUM(B.amount) AS total\_amount\_spent

FROM Guests G

JOIN Bookings B ON G.guest\_id = B.guest\_id

GROUP BY G.guest\_id, G.first\_name, G.last\_name

ORDER BY SUM(B.amount) DESC

LIMIT 2;

1. **Find the average total amount spent by guests who stayed more than 3 days.**

SELECT ROUND(AVG(total\_amount\_spent),2) AS average\_amount\_spent

FROM (

SELECT G.guest\_id, SUM(B.amount) AS total\_amount\_spent

FROM Guests G

JOIN Bookings B ON G.guest\_id = B.guest\_id

WHERE DATEDIFF(B.check\_out\_date, B.check\_in\_date) > 3

GROUP BY G.guest\_id

) AS subquery;

1. **List all guests along with their total stay duration and amount across all bookings.**

SELECT G.guest\_id,CONCAT(G.first\_name, ' ',G.last\_name) AS full\_name,

SUM(DATEDIFF(B.check\_out\_date, B.check\_in\_date)) AS total\_stay\_duration,

SUM(B.amount) AS total\_amount\_spent

FROM Guests G

JOIN Bookings B ON G.guest\_id = B.guest\_id

GROUP BY G.guest\_id, full\_name;

1. **Find the city from where the most guests have stayed.**

SELECT G.city, COUNT(\*) AS number\_of\_guests

FROM Guests G

JOIN Bookings B ON G.guest\_id = B.guest\_id

GROUP BY G.city

ORDER BY COUNT(\*) DESC

LIMIT 1;