Database Project Part 2

1. Indexing Requirements

- Hotel Table Indexes
  - ✓ Add a non-clustered index on the Name column to optimize queries that search for hotels by name.
  - ✓ Add a non-clustered index on the Rating column to speed up queries that filter hotels by rating.

- Room Table Indexes
  - ✓ Add a clustered index on the HotelID and RoomNumber columns to optimize room lookup within a hotel
  - ✓ Add a non-clustered index on the RoomType column to improve searches filtering by room type.

- Booking Table Indexes
  - ✓ Add a non-clustered index on GuestID to optimize guest-related booking searches.
  - ✓ Add a non-clustered index on the Status column to improve filtering of bookings by status.
  - ✓ Add a composite index on RoomID, CheckInDate, and CheckOutDate for efficient querying of booking schedules.

---

2. Views

- **View 1: ViewTopRatedHotels**
  - Create a view that displays the top-rated hotels (rating above 4.5) along with the total number of rooms and average room price for each hotel.
- **View 2: ViewGuestBookings**
  - Create a view that lists each guest along with their total number of bookings and the total amount spent on all bookings.
- **View 3: ViewAvailableRooms**
  - Create a view that lists available rooms for each hotel, grouped by room type and sorted by price in ascending order.
- **View 4: ViewBookingSummary**
  - Create a view that summarizes bookings by hotel, showing the total number of bookings, confirmed bookings, pending bookings, and canceled bookings.
- **View 5: ViewPaymentHistory**
  - Create a view that lists all payment records along with the guest name, hotel name, booking status, and total payment made by each guest for each booking.

3. Functions

- **Function 1: GetHotelAverageRating**
    - Create a function that takes HotelID as an input and returns the average rating of that hotel based on guest reviews.
- **Function 2: GetNextAvailableRoom**
    - Create a function that finds the next available room of a specific type within a given hotel.
- **Function 3: CalculateOccupancyRate**
    - Create a function that takes HotelID as input and returns the occupancy rate of that hotel based on bookings made within the last 30 days.

    ---

4. Stored Procedures

- **Stored Procedure 1: sp_MarkRoomUnavailable**
    - Create a stored procedure that updates the room's availability status to unavailable once a booking is confirmed.
- **Stored Procedure 2: sp_UpdateBookingStatus**
    - Create a stored procedure to change the status of a booking to 'Check-in', 'Check-out', or 'Canceled' based on the current date and booking details.
- **Stored Procedure 3: sp_RankGuestsBySpending**
    - Create a stored procedure that ranks guests by total spending across all bookings.

    ---

5. Triggers

- **Trigger 1: trg_UpdateRoomAvailability**
    - Create a trigger that automatically updates the room's availability to 'Unavailable' when a new booking is added.
- **Trigger 2: trg_CalculateTotalRevenue**
    - Create a trigger that calculates the total revenue for a hotel whenever a new payment is added.
- **Trigger 3: trg_CheckInDateValidation**
    - Create a trigger that prevents the insertion of bookings with a check-in date greater than the check-out date.