



HOTEL RESERVATION DATABASE CASE STUDY

*Building a Scalable Reservation Database for
Data-Driven Hospitality Management*

KARLA KATHLEEN LEGARDE



CONTENT

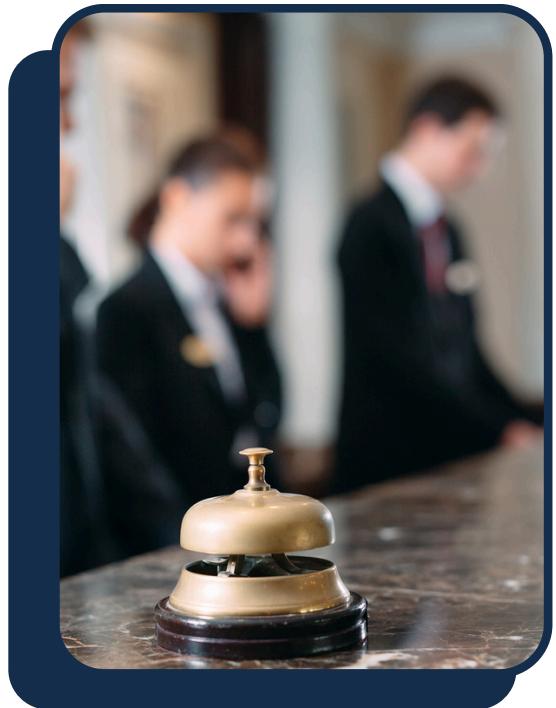
	TOPIC	PAGE NO.
1	Introduction	3
2	About the Company	3
3	Business Problem	4
4	Mission and Objectives	4
5	Reservation Process	5
6	Field Design	5
7	Preliminary and Final Field Lists	6
8	Data Dictionary	7-12
9	Entity Relationship Diagram	13
10	Relationships	13-14
11	SQL Queries	15-18
12	Views	19-23
13	Conclusion	24

INTRODUCTION

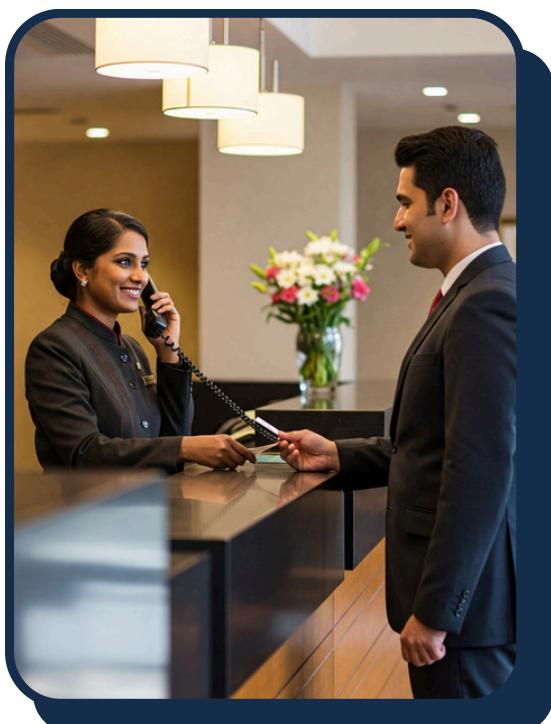
Managing hotel reservations is a complex process involving guests, rooms, payments, booking channels, and promotions. Without a structured database at the room-night grain, hotels struggle to answer basic business questions such as:

- Which channels bring in the most bookings?
- Who are our most loyal guests?
- What are the revenue trends by season?

This case study documents the design and analysis of a Hotel Reservation Database for SkyGate Hotel. The database was developed as part of our database design and analytics coursework, with the aim of supporting both daily operations and management reporting.



ABOUT THE COMPANY



Sky Gate Hotel is a newly established 3-star hospitality venture strategically located near the international airport, designed to cater to transit travelers, business professionals, and short-stay guests.

With four floors and a total of 20 well-appointed rooms, the hotel offers a comfortable and efficient lodging experience.

Its proximity to the airport makes it an ideal choice for layovers and early-morning departures, while its compact size allows for personalized guest service and streamlined management.

Currently in its soft opening phase, Sky Gate Hotel is focused on refining its operations, gathering guest feedback, and optimizing its service delivery.

BUSINESS PROBLEM

Operational teams need a single source of truth to:

1. Track every room-night reservation (who, when, which room type, which channel, which promo).
2. Tie reservations to payments reliably.
3. Report on revenue, cancellations, no-shows, and channel contribution.
4. Identify frequent guests for loyalty offers.

MISSION

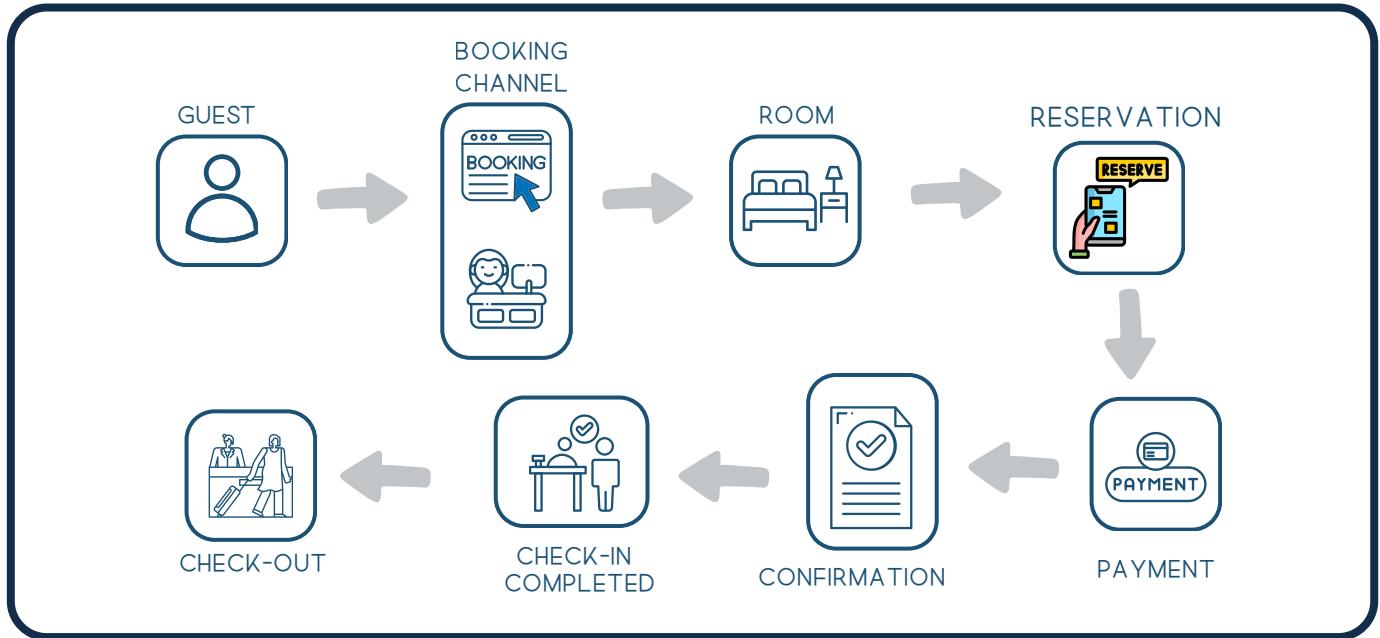
Implement a reservation database for SkyGate that is clean, constrained, and analytics-ready.



OBJECTIVES

- Optimize Operations: Streamline room and booking management.
- Elevate Guest Experience: Deliver smooth, personalized stays.
- Scale for Growth: Ensure systems support future expansion.
- Secure Data: Protect records across all departments.
- Track Performance: Monitor occupancy, revenue, and feedback in real time.

RESERVATION PROCESS



The reservation process at Sky Gate Hotel begins with booking and payment confirmation, continues through check-in, and is only completed once the guest checks out.

FIELD DESIGN (Pre-ERD Process for Reservations)

Goal

Design a relational model (with room-night granularity) + SQL analytics that produce consistent, repeatable insights.

1. List Explanatory Variables (EVs) that influence the KPIs: Room Type, Stay Dates/Season, Booking Channel, Lead Time, Promotion, Adults/Children mix, Reservation Status.
2. Map EVs and data needs to actual feeds (PMS, Payment Gateway, Promotions).
3. Draft a Preliminary Field Inventory (broad, inclusive).
4. Apply pruning rules: drop fields without a reliable source or out of reservations scope; remove redundancies and derivable fields; convert categories into lookups.
5. Normalize the surviving fields into tables and relationships; produce the ERD.

PRELIMINARY FIELD LIST

- Guest ID
- First Name
- Last Name
- Email
- Phone
- Address Line 1
- Address Line 2
- City
- State/Province
- Country
- Postal Code
- Birth Date
- VIP Flag
- Room ID
- Room Number
- Floor Number
- Room Type
- Room Status
- View Type
- Base Rate
- Season
- Max Adults
- Max Children
- Promo ID
- Promo Name
- Promo Description
- Promo Start Date
- Promo End Date
- Reservation ID
- Channel ID
- Channel Name
- Reservation Status
- Check-In Date
- Check-Out Date
- Adult Count
- Children Count
- Length of Stay
- Discount Type
- Payment ID
- Payment Method
- Payment Amount
- Payment Date
- Currency
- Gateway Reference
- Cancellation Reason
- Fees Amount
- Tax Amount
- Wi-Fi Speed
- Guest Feedback Score
- Discount Value
- IsActive (Promo)
- Max Occupancy
- Date
- Age

FINAL FIELD LIST

- Guest ID
- First Name
- Last Name
- Birth Date
- Email
- Phone
- Address Line
- City
- State/Province
- Country
- Postal Code
- Room ID
- Room Number
- Base Rate
- Room Type ID
- Room Type Name
- Max Occupancy
- Channel ID
- Channel Name
- Promo ID
- Promo Name
- Promo Description
- Discount Type
- Discount Value
- Start Date Key
- End Date Key
- Reservation Status ID
- Reservation Status Name
- Payment Date Key
- DateKey
- Year
- Month
- Month Name
- Day
- Reservation

DATA DICTIONARY

Guest

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
GuestID	INT	PRIMARY KEY IDENTITY(1,1)	Unique identifier assigned to each guest. Auto-incremented Primary Key
FirstName	VARCHAR(50)	NOT NULL	The guest's given/first name.
LastName	VARCHAR(50)	NOT NULL	The guest's family/last name or surname.
Email	VARCHAR(100)	NOT NULL UNIQUE CK_GuestEmailFormat CHECK (Email LIKE '_%(@_%.%)')	Unique email address used by the guest for communication and booking confirmations.
PhoneNumber	VARCHAR(15)	NOT NULL CK_GuestPhoneFormat CHECK (PhoneNumber LIKE '+[1-9][0-9]%' AND LEN(PhoneNumber) BETWEEN 8 AND 15)	Guest's contact number in E.164 international format, starting with a + followed by the country code and digits only (e.g., +14035551234). This format ensures compatibility with global telecom standards and avoids symbols like dashes, spaces, or parentheses.
AddressLine	VARCHAR(100)		Street-level portion of the guest's address. Can include apartment, suite, or unit number, street number, and street name (e.g., "Apt 205, 123 Main Street").
City	VARCHAR(50)		City where the guest resides.
StateProvince	VARCHAR(50)		State, province, or regional subdivision of the address
Country	VARCHAR(50)	NOT NULL	Full country name of the guest's residence (e.g., "Canada", "United States").
PostalCode	VARCHAR(12)		Postal or ZIP code of the guest's address (supports formats such as US ZIP+4, Canadian, or UK codes).

RoomType

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
RoomTypeID	INT	PRIMARY KEY IDENTITY(1,1)	Unique identifier for each room type. Auto-incremented Primary Key.
RoomTypeName	VARCHAR(15)	NOT NULL	Descriptive name of the room (e.g. Standard, Deluxe)
MaxOccupancy	SMALLINT	NOT NULL CK_MaxOccupancy CHECK (MaxOccupancy BETWEEN 1 AND 4)	Maximum number of adults permitted in the room. Must be specified and non-negative

Room

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
RoomID	INT	PRIMARY KEY IDENTITY(1,1)	Unique identifier for each room. Auto-Incremented Primary Key
RoomNumber	VARCHAR(5)	NOT NULL UNIQUE	Hotel-assigned room number (e.g., 101, 202). Unique across all rooms
RoomTypeID	INT	FOREIGN KEY NOT NULL	Foreign key linking to Room Type, defining the room's category (e.g. Standard, Deluxe) and occupancy
BasePrice	DECIMAL(6,2)	NOT NULL CHECK (BasePrice >= 0)	Base nightly rate for the room. Must be provided and non-negative. This is the standard rate before discounts or promotions are applied.

BookingChannel

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
ChannelID	INT	PRIMARY KEY IDENTITY(1,1)	Primary key that uniquely identifies each booking channel.
ChannelName	VARCHAR(50)	NOT NULL	Descriptive name of the booking channel (e.g., Website, Phone, Walk-In, Booking.com, Agoda, Expedia).

Promotion

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
PromoID	INT	PRIMARY KEY IDENTITY(1,1)	Unique identifier for each promotion. Auto-incremented Primary Key
PromoName	VARCHAR(100)	NOT NULL	Name of the promotion (e.g., "Winter Sale", "Early Bird").
PromoDescription	VARCHAR(255)	NOT NULL	Description of the offer. Can include terms, conditions, or highlights.
DiscountType	VARCHAR(20)	NOT NULL	Type of discount: either 'Percentage' or 'Fixed Amount'. Helps determine how DiscountValue is applied.
DiscountValue	DECIMAL(6,2)	NOT NULL	Type of discount: either 'Percentage' or 'Fixed Amount'. Helps determine how DiscountValue is applied.
StartDateKey	DATE	NOT NULL FOREIGN KEY	FK to DimDate(DateKey). Date when the promotion becomes active.
EndDateKey	DATE	NOT NULL FOREIGN KEY	FK to DimDate(DateKey). Date when the promotion expires.

DimDate

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
DateKey	DATE	PRIMARY KEY	Unique identifier for each calendar date (formatted YYYY-MM-DD). Used as the central reference for all date-related fields in the database.
Year	SMALLINT	NOT NULL	Numeric year (e.g., 2025).
Month	TINYINT	NOT NULL CHECK (Month BETWEEN 1 and 12)	Numeric month (1 = January ... 12 = December).
Day	TINYINT	NOT NULL CHECK (Day BETWEEN 1 and 31)	Numeric day of the month (1... 31)



Employee

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
EmployeeID	INT	PRIMARY KEY IDENTITY(1,1)	Unique identifier for each employee. Auto-incremented Primary Key
FirstName	VARCHAR(50)	NOT NULL	Employee's first name.
LastName	VARCHAR(50)	NOT NULL	Employee's last name.
Email	VARCHAR(50)	NOT NULL UNIQUE CK_EmployeeEmailFormat CHECK (Email LIKE '%[@%._%]')	Employee's email address. Must be unique and follow a basic email format.
PhoneNumber	VARCHAR(15)	NOT NULL CK_EmployeePhone CHECK (PhoneNumber LIKE '+[1-9][0-9]%' AND LEN(PhoneNumber) BETWEEN 8 AND 15)	Employee's contact number in E.164 international format.
AddressLine	VARCHAR(100)		Street-level portion of the employee's address. Can include apartment, suite, or unit number, street number, and street name (e.g., "Apt 205, 123 Main Street").
City	VARCHAR(50)		City where the employee resides.
StateProvince	VARCHAR(50)		State, province, or regional subdivision of the address
Country	VARCHAR(50)	NOT NULL	Full country name of the employee's residence
PostalCode	VARCHAR(12)		Postal or ZIP code of the employee's address

ReservationStatus

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
ReservationStatusID	INT	PRIMARY KEY IDENTITY(1,1)	Unique identifier for each reservation status. Auto-incremented Primary Key
Status	VARCHAR(30)	NOT NULL	Descriptive name of the status (e.g., Confirmed, Cancelled, Completed, No-Show)

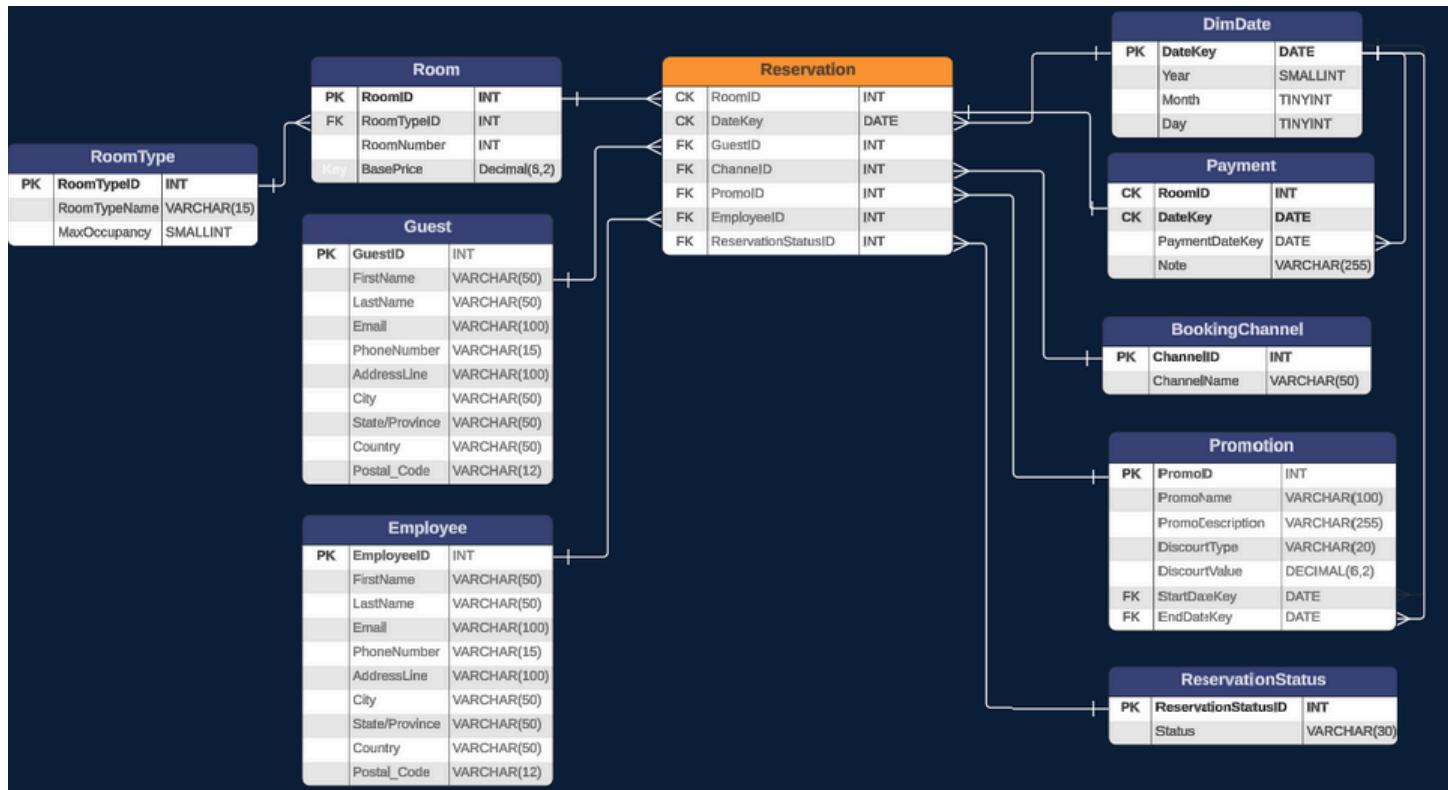
Payment

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
DateKey	INT	PRIMARY KEY IDENTITY(1,1)	Unique payment ID. Auto-incremented Primary Key
RoomID	INT	NOT NULL FOREIGN KEY	Foreign key referencing PaymentMethod.PaymentMethodID. Specifies how the payment was made
PaymentDateKey	DATE	NOT NULL FOREIGN KEY	FK to DimDate(DateKey). Date the payment was processed.
Note	VARCHAR(255)		Optional brief payment context. Informational only — no payment details/PII (no card numbers, auth codes, etc.)

Reservation

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
DateKey	DATE	COMPOSITE PRIMARY KEY, NOT NULL	FOREIGN KEY → DimDate.DateKey, part of Composite Primary Key with RoomID. The calendar date of the stay, links to the date dimension for consistent reporting.
RoomID	INT	COMPOSITE PRIMARY KEY, NOT NULL	FOREIGN KEY → Room.RoomID, part of Composite Primary Key with DateKey. Identifies the specific room.
GuestID	INT	NOT NULL FOREIGN KEY	References Guest.GuestID. Identifies who made the reservation
PromoID	INT	FOREIGN KEY	References Promotion.PromoID. Optional field that applies discounts or special offers to the reservation. Can be null if no promotion is used.
ChannelID	INT	NOT NULL FOREIGN KEY	Foreign key referencing BookingChannel.ChannelID. Indicates the booking source (e.g., website, mobile app, third-party OTA).
ReservationStatusID	INT	NOT NULL FOREIGN KEY	Foreign key referencing ReservationStatus.ReservationStatusID. Tracks the current state of the reservation (e.g., Confirmed, Cancelled, Completed).
EmployeeID	INT	FOREIGN KEY	Foreign key referencing Employee.EmployeeID. Optional field that identifies the staff member who created the reservation. Especially relevant for walk-in bookings or manual entries.

ENTITY RELATIONSHIP DIAGRAM



RELATIONSHIPS

RoomType → Room

1-to-Many: A single room type (Standard or Deluxe) can apply to many rooms

Guest → Reservation

1-to-Many: One guest can create many reservations,

BookingChannel → Reservation

1-to-Many: A booking channel (e.g., Website, OTA, Walk-in) can have many reservations.

Room → Reservation

1-to-Many: One room can have many reservations over time.

Promotion → Reservation

1-to-Many (Optional): A single promotion can be linked to many reservations, but each reservation may or may not have a promotion.

Employee → Reservation

1-to-Many: One employee may handle multiple walk-in/phone reservations, but each reservation is handled by one employee.

RELATIONSHIPS

ReservationStatus → Reservation

1-to-Many: A status (Completed, Cancelled, NoShow) can be applied to many reservations

DimDate → Reservation

1-to-Many: Each calendar date in DimDate can be linked to many reservation records

DimDate → Promotion

1-to-Many: Each calendar date in DimDate can be referenced as the start date or end date of many promotions.

DimDate → Payment

1-to-Many: Each calendar date in DimDate can be the payment date for many payment records.

Reservation → Payment

1-to-1: Each reservation record (one room-night) is allocated at most one payment record. Each payment record is allocated to exactly one reservation.



SQL Queries

This section showcases the SQL queries developed to directly answer key business questions for SkyGate Hotel. Each query is designed around practical scenarios, such as calculating occupancy rates, measuring revenue impact from promotions, or analyzing completed stays per room type. The queries demonstrate how raw transactional data is transformed into actionable insights that support decision-making for marketing, operations, and financial management.

SQL QUERY-1

MONTHLY OCCUPANCY RATE

Business Question

How has occupancy changed month-to-month since opening in June?

Purpose

- **Revenue:** tune base rates and promos by month.
- **Operations:** staff and schedule housekeeping around peaks/dips.
- **Marketing:** target softer months with airport-traveler offers.

Query and Result

```

SELECT d.MonthName,
       FORMAT(1.0 *
              SUM(CASE WHEN rs.Status = 'Completed'
                       THEN 1 ELSE 0 END) /
              COUNT(*), 'P2') AS OccupancyPct
  FROM Reservation r
  JOIN DimDate d
    ON r.DateKey = d.DateKey
  JOIN ReservationStatus rs
    ON r.ReservationStatusID = rs.ReservationStatusID
 GROUP BY d.MonthName, d.Month
 ORDER BY d.Month;

```

Results (1) Messages

	MonthName ↑↓	OccupancyPct ↑↓
1	June	100.00%
2	July	86.77%
3	August	72.31%
4	September	100.00%

SQL QUERY-2

REVENUE FROM PROMOTIONAL BOOKINGS

Business Question

How much revenue was generated from completed bookings with promotions?

Purpose

To measure the revenue generated from the promotion, evaluate its financial impact, and establish a baseline for future discount strategies and campaign planning.

Query and Result

```

SELECT
    p.PromoName,
    CAST(SUM(
        CASE
            WHEN p.DiscountType = 'Percentage' THEN rm.BasePrice * (1 - p.DiscountValue / 100.0)
            WHEN p.DiscountType = 'Amount' THEN rm.BasePrice - p.DiscountValue
            ELSE rm.BasePrice
        END
    ) AS DECIMAL(10,2)) AS PromoRevenue
FROM Reservation r
JOIN Rooms rm ON r.RoomID = rm.RoomID
JOIN ReservationStatus s ON r.ReservationStatusID = s.ReservationStatusID
JOIN Promotions p ON r.PromoID = p.PromoID
WHERE s.Status = 'Completed'
GROUP BY p.PromoName
ORDER BY PromoRevenue DESC;

```

Results (1) Messages

	PromoName	PromoRevenue
1	Opening 10%	55440.00

SQL QUERY-3

COMPLETED STAYS PER ROOM TYPE

Business Question

Which room type—Standard or Deluxe—has been booked most frequently since the hotel's soft opening, based on completed stays?

Purpose

- Plan housekeeping/turnover and maintenance windows by the heavier-used type.
- Design upgrade offers at check-in when Standard demand is high.

Query and Result

```

SELECT
    rt.RoomTypeName,
    COUNT(*) AS CompletedNights
FROM Reservation r
JOIN Rooms rm ON r.RoomID = rm.RoomID
JOIN RoomTypes rt ON rm.RoomTypeID = rt.RoomTypeID
JOIN ReservationStatus rs ON r.ReservationStatusID =
    rs.ReservationStatusID
WHERE rs.Status = 'Completed'
GROUP BY rt.RoomTypeName
ORDER BY CompletedNights DESC;

```

Results (1) Messages

	RoomTypeName	CompletedNights
1	Standard	750
2	Deluxe	722



SQL Views

This section highlights the SQL views created to simplify analysis and reporting for SkyGate Hotel. Each view consolidates data from multiple tables into a reusable structure, making it easier to track room availability per day, monitor monthly revenue, identify the top 10 frequent guests, and analyze confirmed rooms per night. Views serve as ready-to-use datasets for dashboards, reports, and stakeholder presentations, ensuring consistency and efficiency in accessing hotel performance metrics.

VIEW -1

View Name

View_RoomTypeAvailabilityPerDay

Scenario

The hotel needs to know how many rooms of each type are available on any given day.

Purpose

Help front desk, pricing, and operations teams manage inventory, optimize bookings, and plan staffing.

Tables Used

Reservation, DimDate, ReservationStatus

Query and Result

```

CREATE VIEW View_RoomTypeAvailability_ByDate AS
SELECT d.DateKey, rt.RoomTypeName,
       COUNT(rm.RoomID) AS AvailableRooms
  FROM DimDate d
  JOIN RoomTypes rt ON 1 = 1
  JOIN Rooms rm ON rm.RoomTypeID = rt.RoomTypeID
  LEFT JOIN Reservation r ON r.RoomID = rm.RoomID AND r.DateKey = d.DateKey
  LEFT JOIN ReservationStatus rs ON r.ReservationStatusID = rs.ReservationStatusID
 WHERE rs.Status <> 'Confirmed' OR rs.Status IS NULL
 GROUP BY d.DateKey, rt.RoomTypeName;

SELECT *
  FROM View_RoomTypeAvailability_ByDate
 WHERE DateKey = '2025-10-15';

```

Results (1) Messages

	DateKey ↑↓	RoomTypeName ↑↓	AvailableRooms ↑↓
1	2025-10-15	Deluxe	5
2	2025-10-15	Standard	4

VIEW -2

View Name

View_ConfirmedRoomNightsByMonth

Scenario

Management needs to track confirmed demand at the room-night level rather than just bookings.

Purpose

Summarize confirmed room-nights per month to monitor occupancy trends and support planning (staffing, pricing, promotions).

Tables Used

Reservation, DimDate, ReservationStatus

Query and Result

```

CREATE VIEW View_ConfirmedRoomNightsByMonth AS
SELECT
    d.[Year],
    d.[Month]    AS MonthNumber,
    d.MonthName,
    COUNT(*) AS ConfirmedRoomNights
FROM Reservation r
JOIN ReservationStatus rs ON rs.ReservationStatusID = r.ReservationStatusID
JOIN DimDate d ON d.DateKey = r.DateKey
WHERE rs.Status = 'Confirmed'
GROUP BY d.[Year], d.[Month], d.MonthName;

SELECT Year, MonthName, ConfirmedRoomNights
FROM View_ConfirmedRoomNightsByMonth
WHERE Year = 2025 AND MonthNumber IN (10,11,12)
ORDER BY MonthNumber;

```

Results (1) Messages

	Year	↑↓	MonthName	↑↓	ConfirmedRoomNights	↑↓
1	2025		October		413	
2	2025		November		454	
3	2025		December		535	

VIEW -3

View Name

View_MonthlyRevenue

Scenario

The Finance and Operations teams need visibility into gross revenue from bookings on a monthly basis.

Purpose

- Calculate monthly gross revenue from bookings (base rate minus promotions).
- Support trend analysis for dashboards and financial reporting.

Tables Used

Reservation, Room, ReservationStatus, Promotion

Query and Result

```

CREATE VIEW View_MonthlyRevenue AS
SELECT
    YEAR(r.DateKey) AS Yr,
    DATENAME(MONTH, r.DateKey) AS MonthName,
    CAST(SUM(
        CASE
            WHEN p.DiscountType = 'Percentage' THEN rm.BasePrice * (1 - p.DiscountValue/100.0)
            WHEN p.DiscountType = 'Amount' THEN rm.BasePrice - p.DiscountValue
            ELSE rm.BasePrice
        END
    ) AS DECIMAL (10,2)) AS TotalRevenue
FROM Reservation r
JOIN Rooms rm ON rm.RoomID = r.RoomID
JOIN ReservationStatus s ON s.ReservationStatusID = r.ReservationStatusID
LEFT JOIN Promotions p ON r.PromoID = p.PromoID
WHERE s.Status = 'Completed'
GROUP BY YEAR(r.DateKey), DATENAME(MONTH, r.DateKey), MONTH(r.DateKey);

SELECT *
FROM View_MonthlyRevenue
ORDER BY Yr, MonthNumber;

```

Results (1)		Messages	
	Yr ↑↓	MonthName ↑↓	TotalRevenue ↑↓
1	2025	June	55440.00
2	2025	July	94650.00
3	2025	August	46050.00
4	2025	September	54600.00

VIEW -4

View Name

View_Top10_FrequentGuests

Scenario

Operations team needs a quick view of which channels drive bookings.

Purpose

- Feed dashboards for channel mix and quality monitoring.
- Identify channels needing policy changes or targeted promos

Tables Used

Reservation, BookingChannel, ReservationStatus

Query and Result

```

CREATE VIEW _Top10_FrequentGuests AS
SELECT TOP (10)
    g.GuestID,
    CONCAT(g.FirstName, ' ', g.LastName) AS FullName, g.Email,
    g.City, g.Country, COUNT(*) AS Bookings
FROM Reservation r
JOIN Guests g ON r.GuestID = g.GuestID
JOIN ReservationStatus s ON s.ReservationStatusID =
r.ReservationStatusID
WHERE s.Status = 'Completed'
GROUP BY g.GuestID, g.FirstName, g.LastName, g.Email, g.City,
g.Country
ORDER BY COUNT(*) DESC;

SELECT *
FROM View_Top10_FrequentGuests;

```

	ChannelName	TotalBookings	Completed	Cancelled	NoShow
1	Booking.com	556	505	51	0
2	Expedia	444	444	0	0
3	Website	224	217	0	0
4	Agoda	210	114	60	0
5	Phone	149	118	31	0
6	Walk-In	74	74	0	0

CONCLUSION

By modeling at room-night granularity, enforcing keys & checks, and publishing reusable views, SkyGate now has:

- Trustworthy operational reporting (occupancy, channel outcomes)
- Actionable commercial insights (revenue trend, room type mix)
- A foundation for loyalty and promotion strategies (frequent guests, seasonal timing)

This project demonstrates how disciplined database design + approachable SQL can turn raw reservation records into repeatable business value.