* **Week 5: Advanced SQL Queries, Reporting, Views, and Optimization – Summary**

**🔍 1. Revenue Reports**

**a. Total Revenue per Hotel**

SELECT

h.hotel\_name,

SUM(i.total\_amount) AS total\_revenue

FROM

invoice i

JOIN bookings b ON i.booking\_id = b.booking\_id

JOIN hotel h ON b.hotel\_hotel\_id = h.hotel\_id

GROUP BY h.hotel\_id;

**b. Monthly Revenue per Hotel**

SELECT

h.hotel\_name,

MONTH(i.payment\_date) AS month,

YEAR(i.payment\_date) AS year,

SUM(i.total\_amount) AS monthly\_revenue

FROM

invoice i

JOIN bookings b ON i.booking\_id = b.booking\_id

JOIN hotel h ON b.hotel\_hotel\_id = h.hotel\_id

GROUP BY h.hotel\_id, YEAR(i.payment\_date), MONTH(i.payment\_date)

ORDER BY h.hotel\_name, year, month;

**📊 2. Occupancy Rate Analysis**

SELECT

h.hotel\_name,

COUNT(DISTINCT rb.rooms\_room\_id) AS rooms\_occupied,

h.room\_capacity,

ROUND(COUNT(DISTINCT rb.rooms\_room\_id) / h.room\_capacity \* 100, 2) AS occupancy\_rate\_percent

FROM

rooms\_booked rb

JOIN bookings b ON rb.bookings\_booking\_id = b.booking\_id

JOIN hotel h ON b.hotel\_hotel\_id = h.hotel\_id

GROUP BY h.hotel\_id;

**🧑‍💼 3. Guest Analytics and Service Usage**

**a. Top 10 Guests by Bookings**

SELECT

g.guest\_first\_name,

g.guest\_last\_name,

COUNT(b.booking\_id) AS total\_bookings

FROM

guests g

JOIN bookings b ON g.guest\_id = b.guests\_guest\_id

GROUP BY g.guest\_id

ORDER BY total\_bookings DESC

LIMIT 10;

**b. Top 10 Guests by Service Usage**

SELECT

g.guest\_first\_name,

g.guest\_last\_name,

COUNT(hs.service\_used\_id) AS services\_used

FROM

guests g

JOIN bookings b ON g.guest\_id = b.guests\_guest\_id

JOIN hotel\_services\_used\_by\_guests hs ON b.booking\_id = hs.bookings\_booking\_id

GROUP BY g.guest\_id

ORDER BY services\_used DESC

LIMIT 10;

**🧩 4. SQL Views for Reusability**

**a. Monthly Revenue View**

CREATE VIEW monthly\_revenue\_view AS

SELECT

h.hotel\_name,

MONTH(i.payment\_date) AS month,

YEAR(i.payment\_date) AS year,

SUM(i.total\_amount) AS monthly\_revenue

FROM

invoice i

JOIN bookings b ON i.booking\_id = b.booking\_id

JOIN hotel h ON b.hotel\_hotel\_id = h.hotel\_id

GROUP BY h.hotel\_id, YEAR(i.payment\_date), MONTH(i.payment\_date);

**b. Guest Booking Summary View**

CREATE VIEW guest\_booking\_summary AS

SELECT

g.guest\_id,

g.guest\_first\_name,

g.guest\_last\_name,

COUNT(b.booking\_id) AS total\_bookings,

SUM(i.total\_amount) AS total\_spent

FROM

guests g

JOIN bookings b ON g.guest\_id = b.guests\_guest\_id

LEFT JOIN invoice i ON b.booking\_id = i.booking\_id

GROUP BY g.guest\_id;

**⚙️ 5. Stored Procedure: Book Room + Invoice Automation**

DELIMITER //

CREATE PROCEDURE book\_room\_with\_invoice (

IN guestId INT,

IN hotelId INT,

IN empId INT,

IN checkIn DATETIME,

IN checkOut DATETIME,

IN totalRooms INT,

IN paymentType VARCHAR(45),

IN totalAmount DECIMAL(10,2)

)

BEGIN

DECLARE newBookingId INT;

-- Insert Booking

INSERT INTO bookings (

booking\_date, duration\_of\_stay, check\_in\_date, check\_out\_date,

booking\_payment\_type, total\_rooms\_booked, hotel\_hotel\_id,

guests\_guest\_id, employees\_emp\_id, total\_amount

)

VALUES (

NOW(), DATEDIFF(checkOut, checkIn), checkIn, checkOut,

paymentType, totalRooms, hotelId, guestId, empId, totalAmount

);

SET newBookingId = LAST\_INSERT\_ID();

-- Assign Rooms

INSERT INTO rooms\_booked (bookings\_booking\_id, rooms\_room\_id)

SELECT newBookingId, room\_id

FROM rooms

WHERE hotel\_hotel\_id = hotelId

LIMIT totalRooms;

-- Generate Invoice

INSERT INTO invoice (booking\_id, total\_amount, payment\_status, payment\_date)

VALUES (newBookingId, totalAmount, 'Paid', NOW());

END //

DELIMITER ;

**🚀 6. Index Optimization**

-- Indexes to improve performance on frequent filters & joins

CREATE INDEX idx\_booking\_date ON bookings(booking\_date);

CREATE INDEX idx\_check\_in\_date ON bookings(check\_in\_date);

CREATE INDEX idx\_guest\_id ON bookings(guests\_guest\_id);

CREATE INDEX idx\_room\_id ON rooms\_booked(rooms\_room\_id);

CREATE INDEX idx\_payment\_date ON invoice(payment\_date);

CREATE INDEX idx\_hotel\_id ON hotel(hotel\_id);

**✅ Week 5 Status: Fully Completed**

All planned enhancements and optional improvements for Week 5 have been implemented:

* Real-time reporting queries
* Reusable SQL views
* Automation using stored procedures
* Indexing for query performance