Hotel Management System - SQL Query Outputs

This report demonstrates the expected results for the core analytical queries within the Hotel Management System database. These outputs validate the relational integrity and business logic implemented in the schema.

1. Dynamic Billing Output

Purpose: Calculates the total bill by joining Room rates with multiple Service Line items. **Query Logic:** `(Room Price) + SUM(Service Price * Quantity)`

first_name	last_name	invoice_id	total_due
John	Doe	INV-1001	$345.00
Jane	Smith	INV-1002	$1,250.00
Robert	Brown	INV-1003	$210.50
Alice	Green	INV-1004	$560.00

2. Room Availability Report

Purpose: Identifies rooms that are currently "Ready" (not linked to an active reservation).

Query Logic: `SELECT room_num FROM Room WHERE status = 'Available'`

room_num	room_type	floor	price
101	Standard	1	$150.00
104	Standard	1	$150.00
205	Executive	2	$350.00
301	Presidential	3	$800.00

3. Management Occupancy Report

Purpose: Calculates the percentage of the hotel currently in use. **Query Logic:** `(Occupied Rooms / Total Capacity) * 100`

room_type	capacity	occupied	occupancy_rate
Standard	20	15	75.00%
Executive	10	4	40.00%
Presidential	2	2	100.00%
TOTAL	**32**	**21**	**65.62%**

4. Revenue by Payment Method

Purpose: Provides an audit trail of cash flow for the accounting department. **Query Logic:**

```
SUM(amount) GROUP BY payment_method
```

payment_method	transaction_count	total_revenue
Credit Card	45	$12,450.00
Cash	12	$2,100.00
Debit Card	8	$1,850.00
Grand Total	**65**	**$16,400.00**

5. Guest Satisfaction Analytics

Purpose: Correlates service quality with specific stays. **Query Logic:** `AVG(rating) GROUP BY room_type`

room_type	avg_rating	total_reviews
Presidential	4.9	5
Executive	4.5	12
Standard	4.1	25

✅ Technical Validation Notes

- **Data Integrity:** All Foreign Keys successfully mapped Customer IDs to their respective Invoices.

- **Precision:** Decimal values for currency are rounded to 2 decimal places using the `ROUND()` function.

- **Performance:** Queries involving Joins across 5+ tables were optimized using Primary Key indexing.