: **Task 3: Aggregate functions, Having, Order By, Group By, and Joins**

1. Write a SQL query to List Events and Their Average Ticket Prices

SELECT event\_name, AVG(ticket\_price) AS average\_ticket\_price

FROM Event

GROUP BY event\_name;

Output:

| event\_name | average\_ticket\_price |

|-----------------|----------------------|

| Movie Night | 500.00 |

| Cricket Match | 1500.00 |

| Music Evening | 300.00 |

| Bhopal Carnival | 700.00 |

2. Write a SQL query to Calculate the Total Revenue Generated by Events

SELECT SUM(total\_cost) AS total\_revenue

FROM Booking;

Output:

| total\_revenue |

|---------------|

| 10900.00 |

3. Write a SQL query to find the event with the highest ticket sales

SELECT event\_name, SUM(num\_tickets) AS total\_tickets\_sold

FROM Booking

JOIN Event ON Booking.event\_id = Event.event\_id

GROUP BY event\_name

ORDER BY total\_tickets\_sold DESC

LIMIT 1;

Output:

| event\_name | total\_tickets\_sold |

|-----------------|--------------------|

| Cricket Match | 5 |

4. Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event

SELECT event\_name, SUM(num\_tickets) AS total\_tickets\_sold

FROM Booking

JOIN Event ON Booking.event\_id = Event.event\_id

GROUP BY event\_name;

Output:

| event\_name | total\_tickets\_sold |

|-----------------|--------------------|

| Movie Night | 2 |

| Cricket Match | 5 |

| Music Evening | 1 |

| Bhopal Carnival | 3 |

5. Write a SQL query to Find Events with No Ticket Sales

SELECT event\_name

FROM Event

WHERE event\_id NOT IN (SELECT event\_id FROM Booking);

Output:

| event\_name |

|------------|

| \*No matching records\* |

6. Write a SQL query to Find the User Who Has Booked the Most Tickets

SELECT customer\_name, SUM(num\_tickets) AS total\_tickets\_booked

FROM Customer

JOIN Booking ON Customer.customer\_id = Booking.customer\_id

GROUP BY customer\_name

ORDER BY total\_tickets\_booked DESC

LIMIT 1;

Output:

| customer\_name | total\_tickets\_booked |

|----------------|----------------------|

| Rohan Verma | 5 |

7. Write a SQL query to List Events and the total number of tickets sold for each month

SELECT DATE\_FORMAT(booking\_date, '%Y-%m') AS month, event\_name, SUM(num\_tickets) AS total\_tickets\_sold

FROM Booking

JOIN Event ON Booking.event\_id = Event.event\_id

GROUP BY month, event\_name

ORDER BY month;

Output:

| month | event\_name | total\_tickets\_sold |

|-----------|-----------------|--------------------|

| 2024-10 | Movie Night | 2 |

| 2024-10 | Cricket Match | 5 |

| 2024-10 | Music Evening | 1 |

| 2024-10 | Bhopal Carnival | 3 |

8. Write a SQL query to calculate the average Ticket Price for Events in Each Venue

SELECT venue\_name, AVG(ticket\_price) AS average\_ticket\_price

FROM Event

JOIN Venue ON Event.venue\_id = Venue.venue\_id

GROUP BY venue\_name;

Output:

| venue\_name | average\_ticket\_price |

|--------------------|----------------------|

| Shree Ram Theatre | 500.00 |

| Bhopal Haat | 1500.00 |

| Vidisha Heritage | 300.00 |

| Kamal Palace | 700.00 |

9. Write a SQL query to calculate the total Number of Tickets Sold for Each Event Type

SELECT event\_type, SUM(num\_tickets) AS total\_tickets\_sold

FROM Booking

JOIN Event ON Booking.event\_id = Event.event\_id

GROUP BY event\_type;

Output:

| event\_type | total\_tickets\_sold |

|------------|--------------------|

| Movie | 2 |

| Sports | 5 |

| Concert | 4 |

10. Write a SQL query to calculate the total Revenue Generated by Events in Each Year

SELECT YEAR(booking\_date) AS year, SUM(total\_cost) AS total\_revenue

FROM Booking

GROUP BY year;

Output:

| year | total\_revenue |

|------|---------------|

| 2024 | 10900.00 |

11. Write a SQL query to list users who have booked tickets for multiple events

SELECT customer\_name, COUNT(DISTINCT event\_id) AS events\_booked

FROM Customer

JOIN Booking ON Customer.customer\_id = Booking.customer\_id

GROUP BY customer\_name

HAVING events\_booked > 1;

Output:

| customer\_name | events\_booked |

|---------------|---------------|

| \*No matching records\* |

12. Write a SQL query to calculate the Total Revenue Generated by Events for Each User

SELECT customer\_name, SUM(total\_cost) AS total\_revenue

FROM Customer

JOIN Booking ON Customer.customer\_id = Booking.customer\_id

GROUP BY customer\_name;

Output:

| customer\_name | total\_revenue |

|----------------|---------------|

| Aditi Mishra | 1000.00 |

| Rohan Verma | 7500.00 |

| Neha Pandey | 300.00 |

| Vikrant Shukla | 2100.00 |

13. Write a SQL query to calculate the Average Ticket Price for Events in Each Category and Venue

SELECT event\_type, venue\_name, AVG(ticket\_price) AS average\_ticket\_price

FROM Event

JOIN Venue ON Event.venue\_id = Venue.venue\_id

GROUP BY event\_type, venue\_name;

Output:

| event\_type | venue\_name | average\_ticket\_price |

|------------|--------------------|----------------------|

| Movie | Shree Ram Theatre | 500.00 |

| Sports | Bhopal Haat | 1500.00 |

| Concert | Vidisha Heritage | 300.00 |

| Concert | Kamal Palace | 700.00 |

14. Write a SQL query to List Users and the Total Number of Tickets They've Purchased in the Last 30 Days

SELECT customer\_name, SUM(num\_tickets) AS total\_tickets\_purchased

FROM Customer

JOIN Booking ON Customer.customer\_id = Booking.customer\_id

WHERE booking\_date >= DATE\_SUB(CURDATE(), INTERVAL 30 DAY)

GROUP BY customer\_name;

Output:

| customer\_name | total\_tickets\_purchased |

|----------------|-------------------------|

| \*No matching records\* |