Hexaware Foundation Training 2025

SQL / Java Assignments

Ticket Booking System

28/03/2025

Riya Salesia P

Tasks 3:

Aggregate functions, Having, Order By, GroupBy and Joins:

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

   SELECT event_name, AVG(ticket_price) AS avg_ticket_price

   FROM Event GROUP BY event_name;

```
140 •  SELECT event_name, AVG(ticket_price) AS avg_ticket_price
141    FROM Event GROUP BY event_name;
142
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_name	avg_ticket_price
The Dark Knight Rises	300.000000
T20 World Cup Final	2000.000000
Coldplay Live in Concert	2500.000000
India vs England ODI	1500.000000
Bollywood Music Fest	800.000000
Inception Screening	900.000000
WWE Smackdown Live	1200.000000
Tomorrowland India	3000.000000
Navratri Garba Night	1000.000000
New Year's Eve Bash	2200.000000

Result 42

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

   SELECT event_name, SUM(total_cost) AS total_revenue

   FROM Booking B

   JOIN Event E where B.event_id = E.event_id

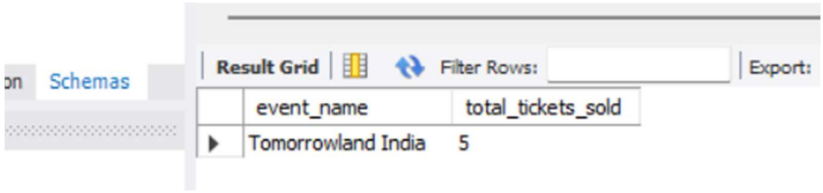
   GROUP BY event_name;

Result Grid | Filter Rows: | Export: | Wra

event_name	total_revenue
T20 World Cup Final	6000.00
India vs England ODI	2400.00
Inception Screening	2100.00
Tomorrowland India	12500.00
New Year's Eve Bash	1800.00
The Dark Knight Rises	500.00
Coldplay Live in Concert	6000.00
Bollywood Music Fest	1000.00
WWE Smackdown Live	3600.00
Navratri Garba Night	2100.00

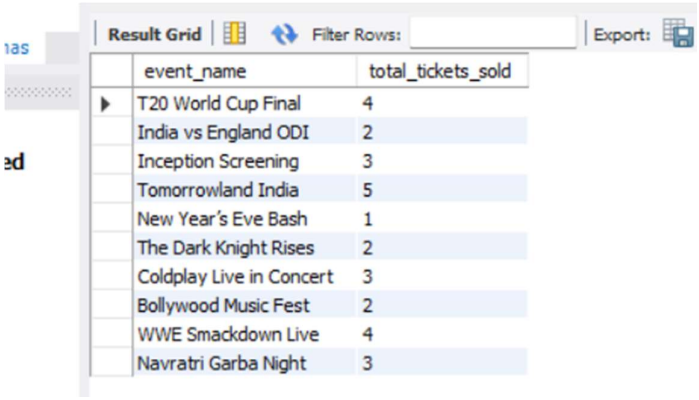
3. Write a SQL query to find the event with the highest ticket sales.

```
SELECT event_name, SUM(num_tickets) AS total_tickets_sold
FROM Booking B
JOIN Event E ON B.event_id = E.event_id
GROUP BY event_name
ORDER BY total_tickets_sold desc
LIMIT 1;
```



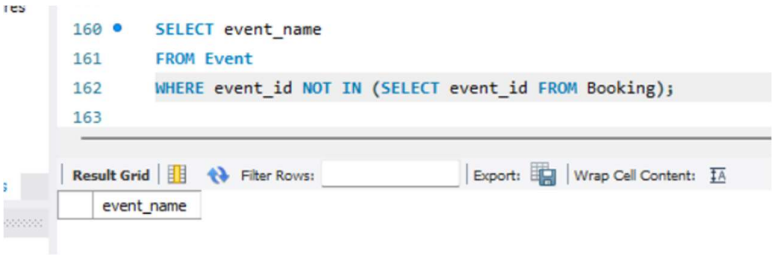
4. Write a SQL query to calculate the Total Number of Tickets Sold for Each Event.

```
SELECT event_name, SUM(num_tickets) AS total_tickets_sold
FROM Booking B
JOIN Event E where B.event_id = E.event_id
GROUP BY event_name;
```



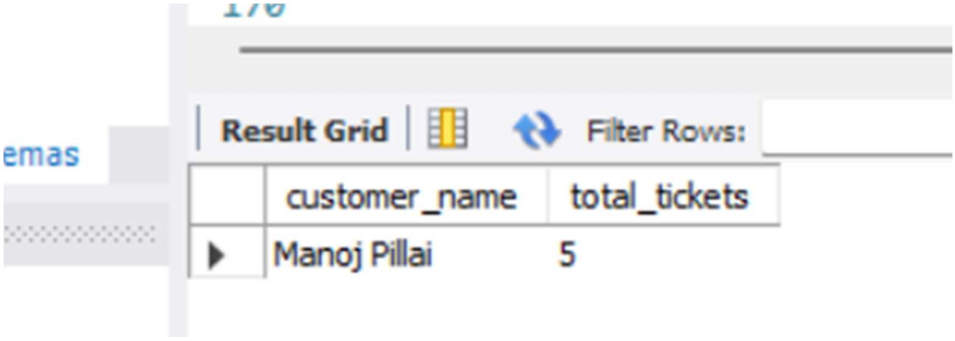
5. Write a SQL query to Find Events with No Ticket Sales.

```
SELECT event_name
FROM Event
WHERE event_id NOT IN (SELECT event_id FROM Booking);
```



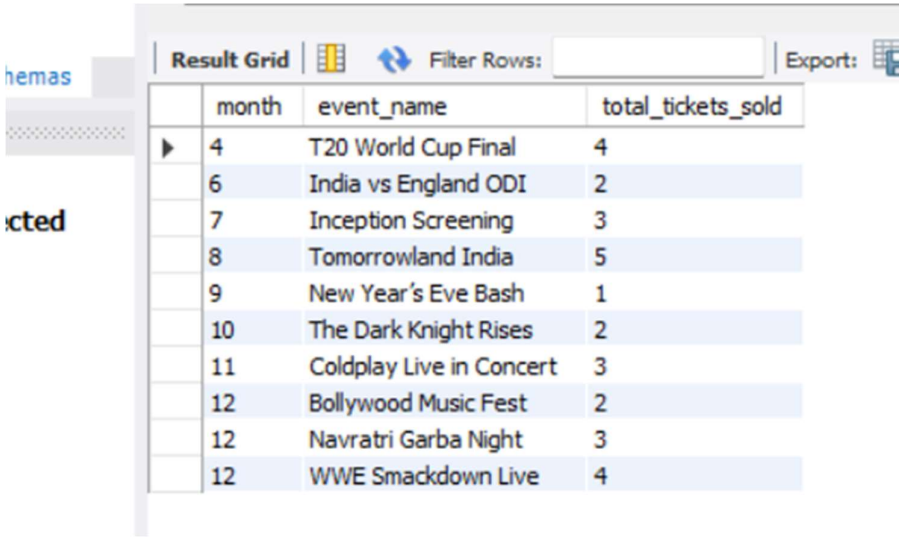
6. Write a SQL query to Find the User Who Has Booked the Most Tickets.

```
SELECT customer_name, SUM(num_tickets) AS total_tickets
FROM Booking B
JOIN Customer C where B.customer_id = C.customer_id
GROUP BY customer_name
ORDER BY total_tickets DESC
LIMIT 1;
```



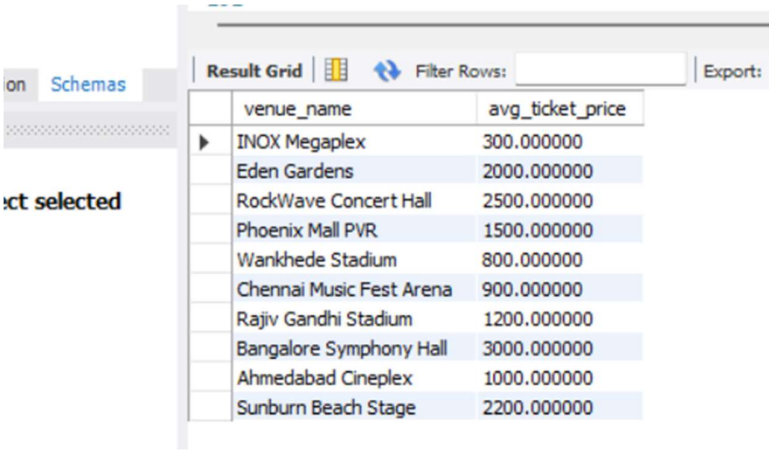
7. Write a SQL query to List Events and the total number of tickets sold for each month.

```
SELECT MONTH(booking_date) AS month, event_name, SUM(num_tickets)
AS total_tickets_sold
FROM Booking B
JOIN Event E ON B.event_id = E.event_id
GROUP BY MONTH(booking_date), event_name
ORDER BY month;
```



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

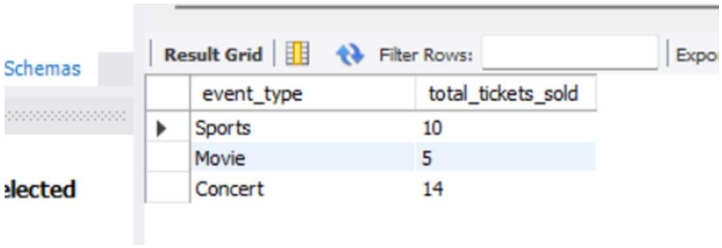
```
SELECT V.venue_name, AVG(E.ticket_price) AS avg_ticket_price
FROM Event E
JOIN Venue V ON E.venue_id = V.venue_id
GROUP BY V.venue_name;
```



venue_name	avg_ticket_price
INOX Megaplex	300.000000
Eden Gardens	2000.000000
RockWave Concert Hall	2500.000000
Phoenix Mall PVR	1500.000000
Wankhede Stadium	800.000000
Chennai Music Fest Arena	900.000000
Rajiv Gandhi Stadium	1200.000000
Bangalore Symphony Hall	3000.000000
Ahmedabad Cineplex	1000.000000
Sunburn Beach Stage	2200.000000

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

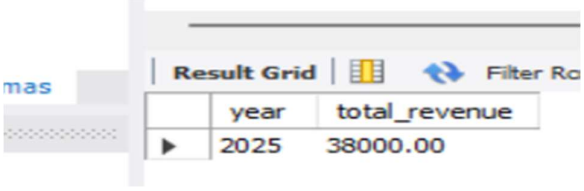
```
SELECT event_type, SUM(num_tickets) AS total_tickets_sold
FROM Booking B
JOIN Event E where B.event_id = E.event_id
GROUP BY event_type;
```



event_type	total_tickets_sold
Sports	10
Movie	5
Concert	14

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

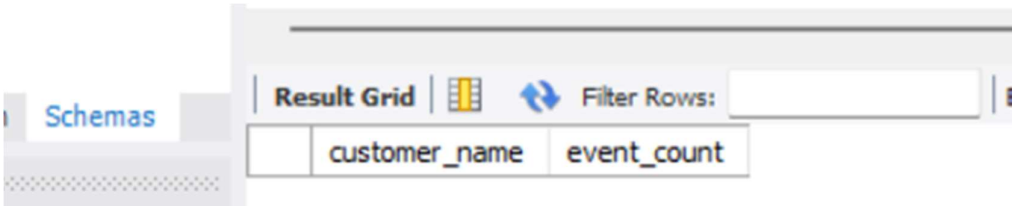
```
SELECT YEAR(booking_date) AS year, SUM(total_cost)
AS total_revenue FROM Booking B
GROUP BY YEAR(booking_date)
ORDER BY year;
```



year	total_revenue
2025	38000.00

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

```
SELECT customer_name, COUNT(distinct event_id) AS event_count
FROM Booking B
JOIN Customer C where B.customer_id = C.customer_id
GROUP BY customer_name
HAVING event_count > 1;
```



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

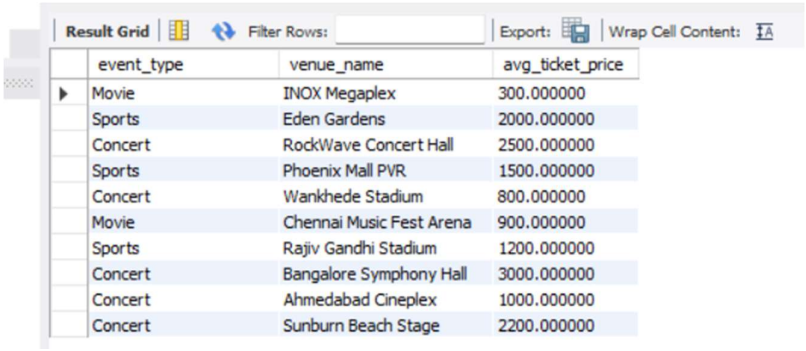
```
SELECT customer_name, SUM(total_cost) AS total_spent
FROM Booking B
JOIN Customer C ON B.customer_id = C.customer_id
GROUP BY customer_name;
```

Result Grid | Filter Rows:

customer_name	total_spent
Rohan Das	6000.00
Aditya Rao	2400.00
Aarav Malhotra	2100.00
Manoj Pillai	12500.00
Harshit Saxena	1800.00
Simran Kaur	500.00
Tanvi Nair	6000.00
Zoya Sheikh	1000.00
Pooja Deshmukh	3600.00
Ishita Agarwal	2100.00

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

```
SELECT event_type, V.venue_name, AVG(ticket_price) AS avg_ticket_price
FROM Event E
JOIN Venue V ON E.venue_id = V.venue_id
GROUP BY event_type, V.venue_name;
```

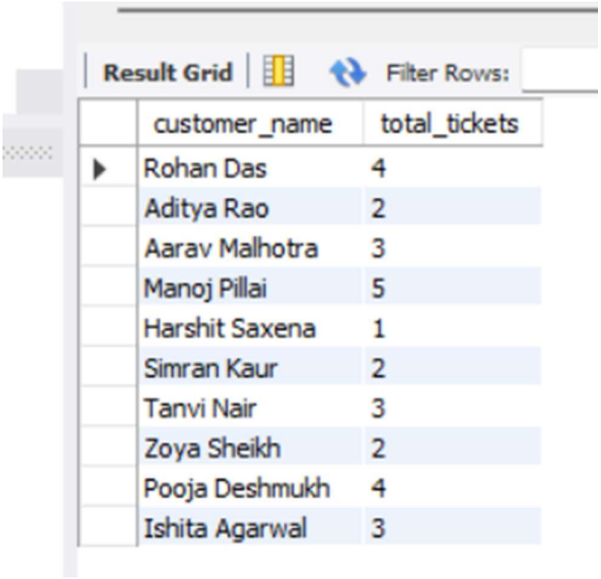


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_type	venue_name	avg_ticket_price
Movie	INOX Megaplex	300.000000
Sports	Eden Gardens	2000.000000
Concert	RockWave Concert Hall	2500.000000
Sports	Phoenix Mall PVR	1500.000000
Concert	Wankhede Stadium	800.000000
Movie	Chennai Music Fest Arena	900.000000
Sports	Rajiv Gandhi Stadium	1200.000000
Concert	Bangalore Symphony Hall	3000.000000
Concert	Ahmedabad Cineplex	1000.000000
Concert	Sunburn Beach Stage	2200.000000

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
FROM Booking B
JOIN Customer C ON B.customer_id = C.customer_id
WHERE booking_date >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)
GROUP BY customer_name;
```



Result Grid | Filter Rows:

customer_name	total_tickets
Rohan Das	4
Aditya Rao	2
Aarav Malhotra	3
Manoj Pillai	5
Harshit Saxena	1
Simran Kaur	2
Tanvi Nair	3
Zoya Sheikh	2
Pooja Deshmukh	4
Ishita Agarwal	3