Hexaware Foundation Training 2025

Sql / Java Assignments

Ticket Booking System

29/03/2025

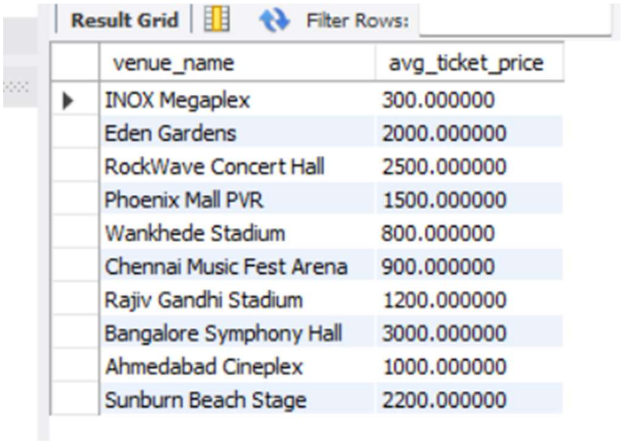
Riya Salesia P

Tasks 4.

Subquery and its types

1. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery.

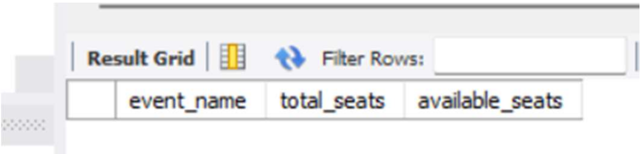
```
SELECT venue_name,
            (SELECT AVG(ticket_price)
            FROM Event WHERE Venue.venue_id = Event.venue_id)
            AS avg_ticket_price
FROM Venue;
```



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

2. Find Events with More Than 50% of Tickets Sold using subquery.

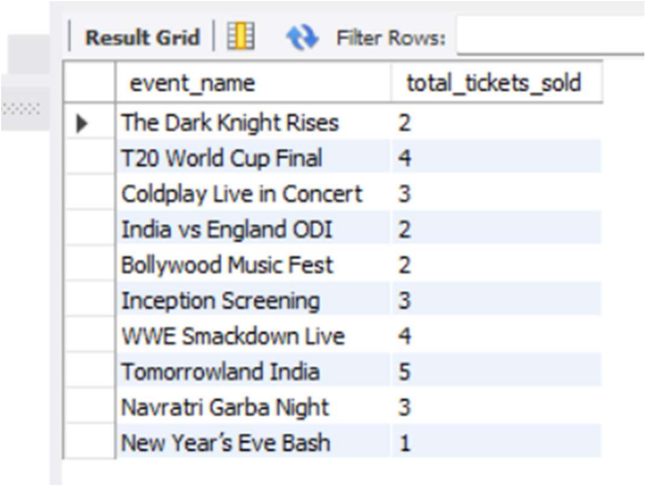
```
SELECT event_name, total_seats, available_seats
FROM Event WHERE (total_seats - available_seats) >
(SELECT total_seats * 0.5 FROM Event E
WHERE E.event_id = Event.event_id);
```



event_name	total_seats	available_seats

3. Calculate the Total Number of Tickets Sold for Each Event.

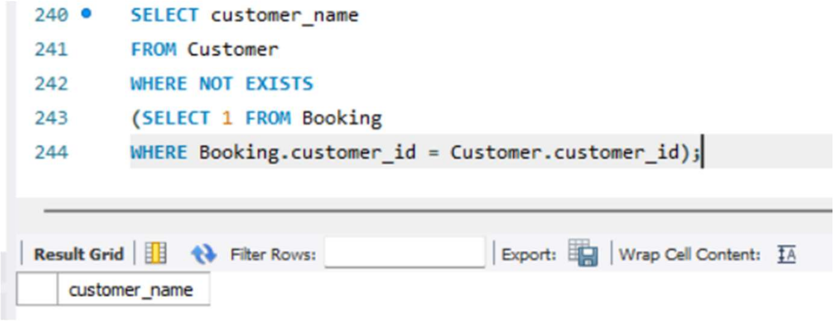
```
SELECT event_name,
       (SELECT SUM(num_tickets) FROM Booking
       WHERE Event.event_id = Booking.event_id) AS total_tickets_sold
FROM Event;
```



event_name	total_tickets_sold
The Dark Knight Rises	2
T20 World Cup Final	4
Coldplay Live in Concert	3
India vs England ODI	2
Bollywood Music Fest	2
Inception Screening	3
WWE Smackdown Live	4
Tomorrowland India	5
Navratri Garba Night	3
New Year's Eve Bash	1

4. Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery.

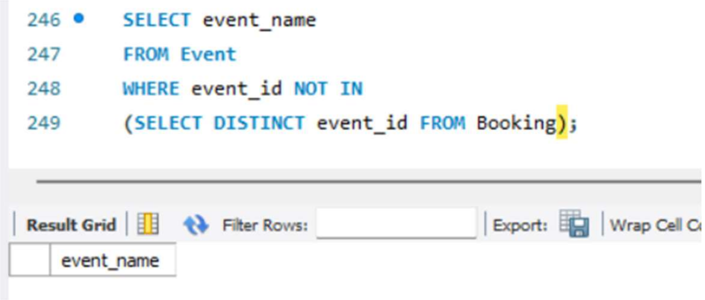
```
SELECT customer_name FROM Customer
WHERE NOT EXISTS
       (SELECT 1 FROM Booking WHERE Booking.customer_id = Customer.customer_id);
```



```
SELECT customer_name
FROM Customer
WHERE NOT EXISTS
(SELECT 1 FROM Booking
WHERE Booking.customer_id = Customer.customer_id);
```

customer_name

5. List Events with No Ticket Sales Using a NOT IN Subquery.

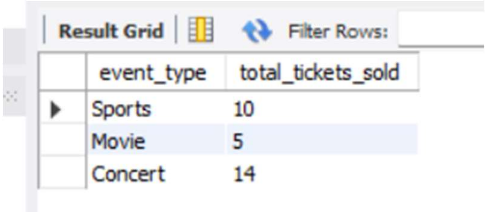


```
SELECT event_name
FROM Event
WHERE event_id NOT IN
(SELECT DISTINCT event_id FROM Booking);
```

event_name

6. Calculate the Total Number of Tickets Sold for Each Event Type Using a Subquery in the FROM Clause.

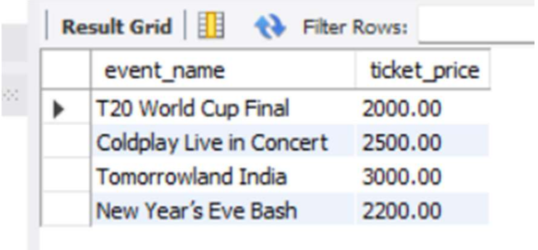
```
SELECT event_type, SUM(total_tickets) AS total_tickets_sold
FROM (SELECT event_type, num_tickets AS total_tickets FROM Booking
        JOIN Event where Booking.event_id = Event.event_id)
 AS TicketSales GROUP BY event_type;
```



event_type	total_tickets_sold
Sports	10
Movie	5
Concert	14

7. Find Events with Ticket Prices Higher Than the Average Ticket Price Using a Subquery in the WHERE Clause.

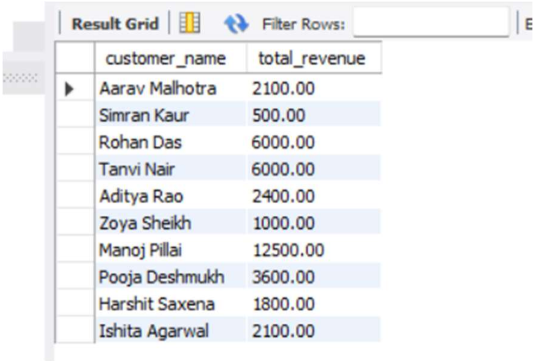
```
SELECT event_name, ticket_price
FROM Event
WHERE ticket_price > (SELECT AVG(ticket_price) FROM Event);
```



event_name	ticket_price
T20 World Cup Final	2000.00
Coldplay Live in Concert	2500.00
Tomorrowland India	3000.00
New Year's Eve Bash	2200.00

8. Calculate the Total Revenue Generated by Events for Each User Using a Correlated Subquery.

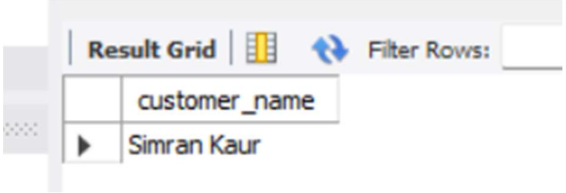
```
SELECT customer_name,
    (SELECT SUM(total_cost) FROM Booking
    WHERE Booking.customer_id = Customer.customer_id) AS total_revenue
FROM Customer;
```



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

9. List Users Who Have Booked Tickets for Events in a Given Venue Using a Subquery in the WHERE Clause.

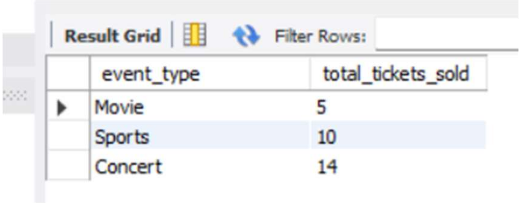
```
SELECT customer_name FROM Customer

WHERE customer_id IN

            (SELECT customer_id FROM Booking  WHERE event_id IN

                        (SELECT event_id FROM Event WHERE venue_id = 1));
```



customer_name
Simran Kaur

10. Calculate the Total Number of Tickets Sold for Each Event Category Using a Subquery with GROUP BY.

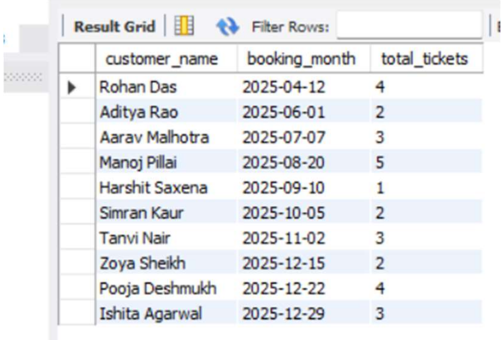
```
SELECT e.event_type, SUM(b.num_tickets)

AS total_tickets_sold FROM Event e

JOIN Booking b ON e.event_id = b.event_id

GROUP BY e.event_type;
```



event_type	total_tickets_sold
Movie	5
Sports	10
Concert	14

11. Find Users Who Have Booked Tickets for Events in each Month Using a Subquery with DATE_FORMAT.

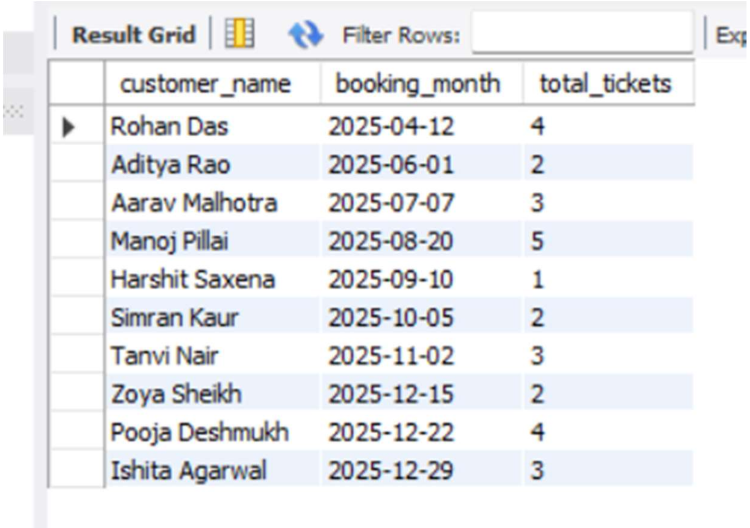
```
    SELECT customer_name, DATE_FORMAT(booking_date, '%Y-%m-%d') AS
booking_month,

    SUM(num_tickets) AS total_tickets FROM Booking

    JOIN Customer ON Booking.customer_id = Customer.customer_id

    GROUP BY customer_name, booking_month;
```



customer_name	booking_month	total_tickets
Rohan Das	2025-04-12	4
Aditya Rao	2025-06-01	2
Aarav Malhotra	2025-07-07	3
Manoj Pillai	2025-08-20	5
Harshit Saxena	2025-09-10	1
Simran Kaur	2025-10-05	2
Tanvi Nair	2025-11-02	3
Zoya Sheikh	2025-12-15	2
Pooja Deshmukh	2025-12-22	4
Ishita Agarwal	2025-12-29	3

12. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery

```
SELECT venue_name,
        (SELECT AVG(ticket_price)
        FROM Event WHERE Event.venue_id = Venue.venue_id) AS avg_ticket_price
FROM Venue;
```



customer_name	booking_month	total_tickets
Rohan Das	2025-04-12	4
Aditya Rao	2025-06-01	2
Aarav Malhotra	2025-07-07	3
Manoj Pillai	2025-08-20	5
Harshit Saxena	2025-09-10	1
Simran Kaur	2025-10-05	2
Tanvi Nair	2025-11-02	3
Zoya Sheikh	2025-12-15	2
Pooja Deshmukh	2025-12-22	4
Ishita Agarwal	2025-12-29	3