

Name: Dheeraj Vemula

SQL Assignment -5

1.Create the database named "TicketBookingSystem".

```
mysql> create database Ticket_booking_system;
Query OK, 1 row affected (0.01 sec)

mysql> use Ticket_booking_system
Database changed
```

2.Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

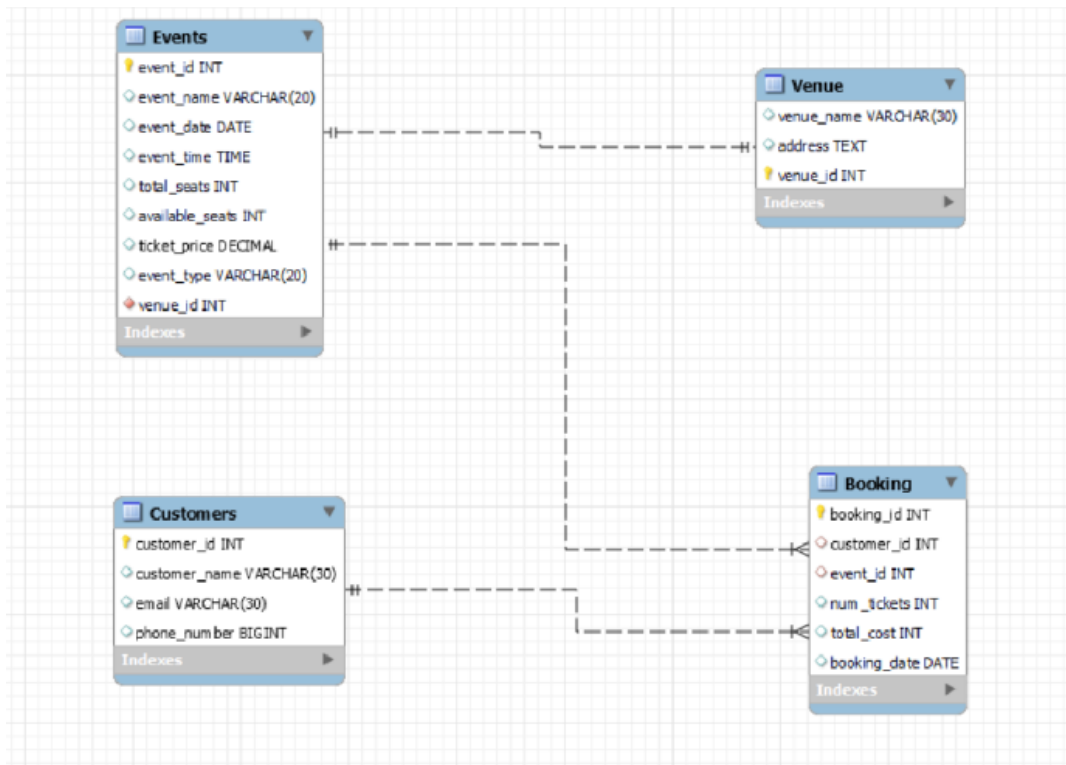
```
mysql> CREATE TABLE Venu (
->     venue_id INT PRIMARY KEY,
->     venue_name VARCHAR(255),
->     address VARCHAR(255)
-> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Event (
->     event_id INT PRIMARY KEY,
->     event_name VARCHAR(255),
->     event_date DATE,
->     event_time TIME,
->     venue_id INT,
->     total_seats INT,
->     available_seats INT,
->     ticket_price DECIMAL,
->     event_type VARCHAR(50),
->     booking_id INT,
->     FOREIGN KEY (venue_id) REFERENCES Venu(venue_id)
-> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Customer (
->     customer_id INT PRIMARY KEY,
->     customer_name VARCHAR(255),
->     email VARCHAR(255),
->     phone_number VARCHAR(15),
->     booking_id INT
-> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Booking (
->     booking_id INT PRIMARY KEY,
->     customer_id INT,
->     event_id INT,
->     num_tickets INT,
->     total_cost DECIMAL,
->     booking_date DATE,
->     FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
->     FOREIGN KEY (event_id) REFERENCES Event(event_id)
-> );
Query OK, 0 rows affected (0.06 sec)
```

3.Create an ERD (Entity Relationship Diagram) for the database.



4.Write a SQL query to insert at least 10 sample records into each table.

```
mysql> INSERT INTO Venu (venue_id, venue_name, address)
-> VALUES
-> (1, 'Chennai Kalyana Mandapam', 'Anna Nagar, Chennai'),
-> (2, 'Trivandrum Convention Center', 'MG Road, Trivandrum'),
-> (3, 'Mysuru Palace Banquet Hall', 'Palace Road, Mysuru'),
-> (4, 'Coimbatore Community Hall', 'Gandhipuram, Coimbatore'),
-> (5, 'Kochi Auditorium', 'Marine Drive, Kochi'),
-> (6, 'Madurai Convention Hall', 'Race Course Road, Madurai'),
-> (7, 'Bangalore Tech Park Auditorium', 'Whitefield, Bangalore'),
-> (8, 'Hyderabad Banjara Hills Banquet', 'Banjara Hills, Hyderabad'),
-> (9, 'Trichy Grand Plaza', 'Cantonment, Trichy'),
-> (10, 'Vizag Beach Resort', 'Beach Road, Vizag');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```

mysql> INSERT INTO Event (event_id, event_name, event_date, event_time, venue_id, total
_seats, available_seats, ticket_price, event_type, booking_id)
-> VALUES
-> (1, 'Kollywood Night', '2023-01-15', '19:30:00', 1, 100, 50, 25.00, 'Concert',
1),
-> (2, 'Onam Celebration', '2023-09-05', '13:00:00', 3, 200, 150, 30.00, 'Movie',
2),
-> (3, 'Mysuru Dasara Dance Festival', '2023-10-12', '18:00:00', 4, 150, 120, 22.
50, 'Concert', 3),
-> (4, 'Coimbatore Film Festival', '2023-05-20', '17:30:00', 5, 180, 100, 18.00,
'Movie', 4),
-> (5, 'Kochi Music Awards', '2023-07-28', '20:00:00', 2, 120, 80, 20.00, 'Concer
t', 5),
-> (6, 'Madurai Classical Dance Recital', '2023-04-08', '15:45:00', 6, 80, 60, 15
.00, 'Concert', 6),
-> (7, 'Bangalore Tech Conference', '2023-03-10', '10:30:00', 7, 300, 250, 35.00,
'Conference', 7),
-> (8, 'Hyderabad Cricket Tournament', '2023-06-18', '14:15:00', 8, 200, 180, 25.
00, 'Sports', 8),
-> (9, 'Trichy Science Exhibition', '2023-11-02', '11:00:00', 9, 120, 100, 12.00,
'Exhibition', 9),
-> (10, 'Vizag Beach Party', '2023-08-15', '21:30:00', 10, 150, 120, 30.00, 'Part
y', 10);
Query OK, 10 rows affected, 1 warning (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 1

```

```

mysql> INSERT INTO Customer (customer_id, customer_name, email, phone_number, booking_i
d)
-> VALUES
-> (1, 'Rajesh Kumar', 'rajesh.kumar@gmail.com', '9876543210', 1),
-> (2, 'Priya Suresh', 'priya.suresh@yahoo.com', '8765432109', 2),
-> (3, 'Ananya Menon', 'ananya.menon@hotmail.com', '7654321098', 3),
-> (4, 'Arjun Nair', 'arjun.nair@gmail.com', '6543210987', 4),
-> (5, 'Kavya Pillai', 'kavya.pillai@yahoo.com', '5432109876', 5),
-> (6, 'Manoj Sundaram', 'manoj.sundaram@gmail.com', '4321098765', 6),
-> (7, 'Neha Reddy', 'neha.reddy@yahoo.com', '3210987654', 7),
-> (8, 'Suresh Babu', 'suresh.babu@hotmail.com', '2109876543', 8),
-> (9, 'Vidya Shankar', 'vidya.shankar@gmail.com', '1098765432', 9),
-> (10, 'Aruna Kumar', 'aruna.kumar@yahoo.com', '9876543210', 10);
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0

```

```
mysql> INSERT INTO Booking (booking_id, customer_id, event_id, num_tickets, total_cost,
    booking_date)
    -> VALUES
    -> (1, 1, 1, 2, 50.00, '2023-01-10'),
    -> (2, 2, 2, 4, 120.00, '2023-08-02'),
    -> (3, 3, 3, 3, 67.50, '2023-09-25'),
    -> (4, 4, 4, 5, 90.00, '2023-04-15'),
    -> (5, 5, 5, 2, 40.00, '2023-07-20'),
    -> (6, 6, 6, 1, 15.00, '2023-03-28'),
    -> (7, 7, 7, 3, 105.00, '2023-02-12'),
    -> (8, 8, 8, 4, 100.00, '2023-05-28'),
    -> (9, 9, 9, 2, 24.00, '2023-11-05'),
    -> (10, 10, 10, 3, 90.00, '2023-08-20');
Query OK, 10 rows affected, 1 warning (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 1
```

Task 2:

1. Write a SQL query to list all Events.

```
mysql> SELECT * FROM Event;
```

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
1	Kollywood Night	2023-01-15	19:30:00	1	100	50	25	Concert	1
2	Onam Celebration	2023-09-05	13:00:00	3	200	150	30	Movie	2
3	Mysuru Dasara Dance Festival	2023-10-12	18:00:00	4	150	120	23	Concert	3
4	Coimbatore Film Festival	2023-05-20	17:30:00	5	180	100	18	Movie	4
5	Kochi Music Awards	2023-07-28	20:00:00	2	120	80	20	Concert	5
6	Madurai Classical Dance Recital	2023-04-08	15:45:00	6	80	60	15	Concert	6
7	Bangalore Tech Conference	2023-03-10	10:30:00	7	300	250	35	Conference	7
8	Hyderabad Cricket Tournament	2023-06-18	14:15:00	8	200	180	25	Sports	8
9	Trichy Science Exhibition	2023-11-02	11:00:00	9	120	100	12	Exhibition	9
10	Vizag Beach Party	2023-08-15	21:30:00	10	150	120	30	Party	10

```
10 rows in set (0.00 sec)
```

2. Write a SQL query to select events with available tickets

```
mysql> SELECT * FROM Event WHERE available_seats > 0;
```

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
1	Kollywood Night	2023-01-15	19:30:00	1	100	50	25	Concert	1
2	Onam Celebration	2023-09-05	13:00:00	3	200	150	30	Movie	2
3	Mysuru Dasara Dance Festival	2023-10-12	18:00:00	4	150	120	23	Concert	3
4	Coimbatore Film Festival	2023-05-20	17:30:00	5	180	100	18	Movie	4
5	Kochi Music Awards	2023-07-28	20:00:00	2	120	80	20	Concert	5
6	Madurai Classical Dance Recital	2023-04-08	15:45:00	6	80	60	15	Concert	6
7	Bangalore Tech Conference	2023-03-10	10:30:00	7	300	250	35	Conference	7
8	Hyderabad Cricket Tournament	2023-06-18	14:15:00	8	200	180	25	Sports	8
9	Trichy Science Exhibition	2023-11-02	11:00:00	9	120	100	12	Exhibition	9
10	Vizag Beach Party	2023-08-15	21:30:00	10	150	120	30	Party	10

```
10 rows in set (0.00 sec)
```

3. Write a SQL query to select events name partial match with 'cup'.

```
mysql> SELECT * FROM Event WHERE event_name LIKE '%cup%';
Empty set (0.00 sec)
```

4. Write a SQL query to select events with ticket price range is between 1000 to 2500.

```
mysql> SELECT * FROM Event WHERE ticket_price BETWEEN 1000 AND 2500;
Empty set (0.00 sec)
```

5. Write a SQL query to retrieve events with dates falling within a specific range.

```
mysql> SELECT * FROM Event WHERE event_date BETWEEN '2023-05-01' AND '2023-08-31';
```

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
4	Coimbatore Film Festival	2023-05-20	17:30:00	5	180	100	18	Movie	4
5	Kochi Music Awards	2023-07-28	20:00:00	2	120	80	20	Concert	5
8	Hyderabad Cricket Tournament	2023-06-18	14:15:00	8	200	180	25	Sports	8
10	Vizag Beach Party	2023-08-15	21:30:00	10	150	120	30	Party	10

```
4 rows in set (0.00 sec)
```

6. Write a SQL query to retrieve events with available tickets that also have "Concert" in their name.

```
mysql> SELECT * FROM Event WHERE available_seats > 0 AND event_type = 'Concert';
```

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
1	Kollywood Night	2023-01-15	19:30:00	1	100	50	25	Concert	1
3	Mysuru Dasara Dance Festival	2023-10-12	18:00:00	4	150	120	23	Concert	3
5	Kochi Music Awards	2023-07-28	20:00:00	2	120	80	20	Concert	5
6	Madurai Classical Dance Recital	2023-04-08	15:45:00	6	80	60	15	Concert	6

4 rows in set (0.00 sec)

7. Write a SQL query to retrieve users in batches of 5, starting from the 6th user.

```
mysql> SELECT * FROM Customer ORDER BY customer_id;
```

customer_id	customer_name	email	phone_number	booking_id
1	Rajesh Kumar	rajesh.kumar@gmail.com	9876543210	1
2	Priya Suresh	priya.suresh@yahoo.com	8765432109	2
3	Ananya Menon	ananya.menon@hotmail.com	7654321098	3
4	Arjun Nair	arjun.nair@gmail.com	6543210987	4
5	Kavya Pillai	kavya.pillai@yahoo.com	5432109876	5
6	Manoj Sundaram	manoj.sundaram@gmail.com	4321098765	6
7	Neha Reddy	neha.reddy@yahoo.com	3210987654	7
8	Suresh Babu	suresh.babu@hotmail.com	2109876543	8
9	Vidya Shankar	vidya.shankar@gmail.com	1098765432	9
10	Aruna Kumar	aruna.kumar@yahoo.com	9876543210	10

10 rows in set (0.00 sec)

8. Write a SQL query to retrieve bookings details contains booked no of ticket more than 4.

```
mysql> SELECT * FROM Booking WHERE num_tickets > 4;
```

booking_id	customer_id	event_id	num_tickets	total_cost	booking_date
4	4	4	5	90	2023-04-15

1 row in set (0.00 sec)

9. Write a SQL query to retrieve customer information whose phone number end with '0'.

```
mysql> SELECT * FROM Customer WHERE phone_number LIKE '%0';
```

customer_id	customer_name	email	phone_number	booking_id
1	Rajesh Kumar	rajesh.kumar@gmail.com	9876543210	1
10	Aruna Kumar	aruna.kumar@yahoo.com	9876543210	10

2 rows in set (0.00 sec)

10. Write a SQL query to retrieve the events in order whose seat capacity more than 150.

```
mysql> SELECT * FROM Event WHERE total_seats > 150 ORDER BY total_seats;
```

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
4	Coimbatore Film Festival	2023-05-20	17:30:00	5	180	100	18	Movie	4
2	Onam Celebration	2023-09-05	13:00:00	3	200	150	30	Movie	2
8	Hyderabad Cricket Tournament	2023-06-18	14:15:00	8	200	180	25	Sports	8
7	Bangalore Tech Conference	2023-03-10	10:30:00	7	300	250	35	Conference	7

4 rows in set (0.00 sec)

11. Write a SQL query to select events name not start with 'x', 'y', 'z'.

```
mysql> SELECT * FROM Event WHERE NOT (event_name LIKE 'x%' OR event_name LIKE 'y%' OR event_name LIKE 'z%');
```

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
1	Kollywood Night	2023-01-15	19:30:00	1	100	50	25	Concert	1
2	Onam Celebration	2023-09-05	13:00:00	3	200	150	30	Movie	2
3	Mysuru Dasara Dance Festival	2023-10-12	18:00:00	4	150	120	23	Concert	3
4	Coimbatore Film Festival	2023-05-20	17:30:00	5	180	100	18	Movie	4
5	Kochi Music Awards	2023-07-28	20:00:00	2	120	80	20	Concert	5
6	Madurai Classical Dance Recital	2023-04-08	15:45:00	6	80	60	15	Concert	6
7	Bangalore Tech Conference	2023-03-10	10:30:00	7	300	250	35	Conference	7
8	Hyderabad Cricket Tournament	2023-06-18	14:15:00	8	200	180	25	Sports	8
9	Trichy Science Exhibition	2023-11-02	11:00:00	9	120	100	12	Exhibition	9
10	Vizag Beach Party	2023-08-15	21:30:00	10	150	120	30	Party	10

10 rows in set (0.00 sec)

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

```
mysql> SELECT event_id, event_name, AVG(ticket_price) AS average_ticket_price
-> FROM Event
-> GROUP BY event_id, event_name;
```

event_id	event_name	average_ticket_price
1	Kollywood Night	25.0000
2	Onam Celebration	30.0000
3	Mysuru Dasara Dance Festival	23.0000
4	Coimbatore Film Festival	18.0000
5	Kochi Music Awards	20.0000
6	Madurai Classical Dance Recital	15.0000
7	Bangalore Tech Conference	35.0000
8	Hyderabad Cricket Tournament	25.0000
9	Trichy Science Exhibition	12.0000
10	Vizag Beach Party	30.0000

10 rows in set (0.00 sec)

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

```
mysql> SELECT SUM(total_cost) AS total_revenue
-> FROM Booking;
```

```
+-----+
| total_revenue |
+-----+
|          702 |
+-----+
```

1 row in set (0.00 sec)

Task 3:

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

```
mysql> SELECT event_id, event_name, SUM(num_tickets) AS total_tickets_sold
-> FROM Booking
-> GROUP BY event_id, event_name
-> ORDER BY total_tickets_sold DESC
-> LIMIT 1;
```

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

```
mysql> SELECT event_id, event_name, SUM(num_tickets) AS total_tickets_sold
-> FROM Booking
-> GROUP BY event_id, event_name;
```

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

```
mysql> SELECT event_id, event_name
-> FROM Event
-> WHERE event_id NOT IN (SELECT DISTINCT event_id FROM Booking);
Empty set (0.00 sec)
```

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

```
mysql> SELECT customer_id, customer_name, SUM(num_tickets) AS total_tickets_booked
-> FROM Booking
-> GROUP BY customer_id, customer_name
-> ORDER BY total_tickets_booked DESC
-> LIMIT 1;
```


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

```
mysql> SELECT MONTH(booking_date) AS month, event_id, event_name, SUM(num_tickets) AS total_tickets_sold
-> FROM Booking
-> GROUP BY month, event_id, event_name
-> ORDER BY month, event_id;
```

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

```
mysql> SELECT venue_id, venue_name, AVG(ticket_price) AS average_ticket_price
-> FROM Event
-> GROUP BY venue_id, venue_name;
```

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

```
mysql> SELECT event_type, SUM(num_tickets) AS total_tickets_sold
-> FROM Event
-> JOIN Booking ON Event.event_id = Booking.event_id
-> GROUP BY event_type;
+-----+-----+
| event_type | total_tickets_sold |
+-----+-----+
| Concert   | 8 |
| Movie     | 9 |
| Conference | 3 |
| Sports    | 4 |
| Exhibition | 2 |
| Party     | 3 |
+-----+-----+
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT YEAR(booking_date) AS year, SUM(total_cost) AS total_revenue
-> FROM Booking
-> GROUP BY year;
+-----+-----+
| year | total_revenue |
+-----+-----+
| 2023 | 702 |
+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT customer_id, customer_name, COUNT(DISTINCT event_id) AS events_booked
-> FROM Booking
-> GROUP BY customer_id, customer_name
-> HAVING events_booked > 1;
```

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

```
mysql> SELECT customer_id, customer_name, SUM(total_cost) AS total_revenue
-> FROM Booking
-> GROUP BY customer_id, customer_name;
```

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

```
mysql> SELECT venue_id, venue_name, event_type, AVG(ticket_price) AS average_ticket_price
-> FROM Event
-> GROUP BY venue_id, venue_name, event_type;
```

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

```
mysql> SELECT customer_id, customer_name, SUM(num_tickets) AS total_tickets_purchased
-> FROM Booking
-> WHERE booking_date >= CURDATE() - INTERVAL 30 DAY
-> GROUP BY customer_id, customer_name;
```

Task 4:

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

```
mysql> SELECT venue_id, venue_name,
-> (SELECT AVG(ticket_price) FROM Event WHERE Event.venue_id = Venu.venue_id
) AS average_ticket_price
-> FROM Venu;
```

venue_id	venue_name	average_ticket_price
1	Chennai Kalyana Mandapam	25.0000
2	Trivandrum Convention Center	20.0000
3	Mysuru Palace Banquet Hall	30.0000
4	Coimbatore Community Hall	23.0000
5	Kochi Auditorium	18.0000
6	Madurai Convention Hall	15.0000
7	Bangalore Tech Park Auditorium	35.0000
8	Hyderabad Banjara Hills Banquet	25.0000
9	Trichy Grand Plaza	12.0000
10	Vizag Beach Resort	30.0000

10 rows in set (0.00 sec)

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

```
mysql> SELECT event_id, event_name
-> FROM Event
-> WHERE (SELECT SUM(num_tickets) FROM Booking WHERE Booking.event_id = Event.event_id) > 0.5 * total_seats;
```

Empty set (0.00 sec)

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

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

event_id	event_name	total_tickets_sold
1	Kollywood Night	2
2	Onam Celebration	4
3	Mysuru Dasara Dance Festival	3
4	Coimbatore Film Festival	5
5	Kochi Music Awards	2
6	Madurai Classical Dance Recital	1
7	Bangalore Tech Conference	3
8	Hyderabad Cricket Tournament	4
9	Trichy Science Exhibition	2
10	Vizag Beach Party	3

10 rows in set (0.00 sec)

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

```
mysql> SELECT customer_id, customer_name
-> FROM Customer
-> WHERE NOT EXISTS (SELECT 1 FROM Booking WHERE Booking.customer_id = Customer.customer_id);
Empty set (0.00 sec)
```

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

```
mysql> SELECT event_id, event_name
-> FROM Event
-> WHERE event_id NOT IN (SELECT DISTINCT event_id FROM Booking);
Empty set (0.00 sec)
```

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

```
mysql> SELECT event_type, SUM(total_tickets_sold) AS total_tickets_sold
-> FROM (SELECT event_type, event_id,
->          (SELECT SUM(num_tickets) FROM Booking WHERE Booking.event_id = Event.event_id) AS total_tickets_sold
->        FROM Event) AS Subquery
-> GROUP BY event_type;
+-----+-----+
| event_type | total_tickets_sold |
+-----+-----+
| Concert   | 8 |
| Movie     | 9 |
| Conference | 3 |
| Sports    | 4 |
| Exhibition | 2 |
| Party     | 3 |
+-----+-----+
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT event_id, event_name, ticket_price
-> FROM Event
-> WHERE ticket_price > (SELECT AVG(ticket_price) FROM Event);
+-----+-----+-----+
| event_id | event_name                | ticket_price |
+-----+-----+-----+
| 1       | Kollywood Night           | 25 |
| 2       | Onam Celebration          | 30 |
| 7       | Bangalore Tech Conference | 35 |
| 8       | Hyderabad Cricket Tournament | 25 |
| 10      | Vizag Beach Party         | 30 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

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

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

customer_id	customer_name	total_revenue
1	Rajesh Kumar	50
2	Priya Suresh	120
3	Ananya Menon	68
4	Arjun Nair	90
5	Kavya Pillai	40
6	Manoj Sundaram	15
7	Neha Reddy	105
8	Suresh Babu	100
9	Vidya Shankar	24
10	Aruna Kumar	90

```
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT customer_id, customer_name
-> FROM Customer
-> WHERE EXISTS (SELECT 1 FROM Booking
->      JOIN Event ON Booking.event_id = Event.event_id
->      WHERE Event.venue_id = 1 AND Booking.customer_id = Customer.customer_id);
```

customer_id	customer_name
1	Rajesh Kumar

```
1 row in set (0.00 sec)
```

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

```
mysql> SELECT event_type, SUM(total_tickets_sold) AS total_tickets_sold
-> FROM (SELECT event_type,
->      (SELECT SUM(num_tickets) FROM Booking WHERE Booking.event_id = Event
.event_id) AS total_tickets_sold
->      FROM Event) AS Subquery
-> GROUP BY event_type;
```

event_type	total_tickets_sold
Concert	8
Movie	9
Conference	3
Sports	4
Exhibition	2
Party	3

```
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT customer_id, customer_name, DATE_FORMAT(booking_date, '%Y-%m') AS booking
_month
-> FROM Booking
-> JOIN Customer ON Booking.customer_id = Customer.customer_id;
```

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

```
mysql> SELECT venue_id, venue_name,  
->      (SELECT AVG(ticket_price) FROM Event WHERE Event.venue_id = Venu.venue_id  
) AS average_ticket_price  
-> FROM Venu;
```

venue_id	venue_name	average_ticket_price
1	Chennai Kalyana Mandapam	25.0000
2	Trivandrum Convention Center	20.0000
3	Mysuru Palace Banquet Hall	30.0000
4	Coimbatore Community Hall	23.0000
5	Kochi Auditorium	18.0000
6	Madurai Convention Hall	15.0000
7	Bangalore Tech Park Auditorium	35.0000
8	Hyderabad Banjara Hills Banquet	25.0000
9	Trichy Grand Plaza	12.0000
10	Vizag Beach Resort	30.0000

10 rows in set (0.00 sec)