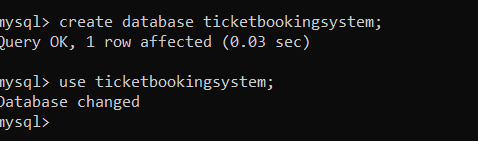
ASSIGNMENT-5

TICKET BOOKING SYSTEM

**Task 1**

**Database Design:**

1.create database



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

Venue

Event

Customers

Booking

1.

CREATE DATABASE TicketBookingSystem;

USE TicketBookingSystem;

2.

CREATE TABLE Venu(

venue\_id int NOT NULL PRIMARY KEY,

venue\_name VARCHAR(30),

address VARCHAR(50)

);

CREATE TABLE Event(

event\_id int NOT NULL PRIMARY KEY,

event\_name VARCHAR(255),

event\_date DATE,

event\_time TIME,

venue\_id int,

total\_seats int(10000),

available\_seats int(10000),

ticket\_price DECIMAL(10,2),

event\_type VARCHAR(40) CHECK (event\_type IN('Movie','Sports','COncert')),

booking\_id int,

FOREIGN KEY (venue\_id) REFERENCES Venue(venue\_id),

FOREIGN KEY ( booking\_id) REFERENCES Booking(booking\_id)

);

CREATE TABLE Customer(

customer\_id int NOT NULL PRIMARY KEY,

customer\_name VARCHAR(20),

email VARCHAR(30),

phone\_number INT(10),

booking\_id int ,

FOREIGN KEY (booking\_id) REFERENCES Booking(booking\_id)

);

CREATE TABLE Booking(

booking\_id int NOT NULL PRIMARY KEY,

customer\_id int,

event\_id int,

num\_tickets int,

total\_cost Decimal(10,2),

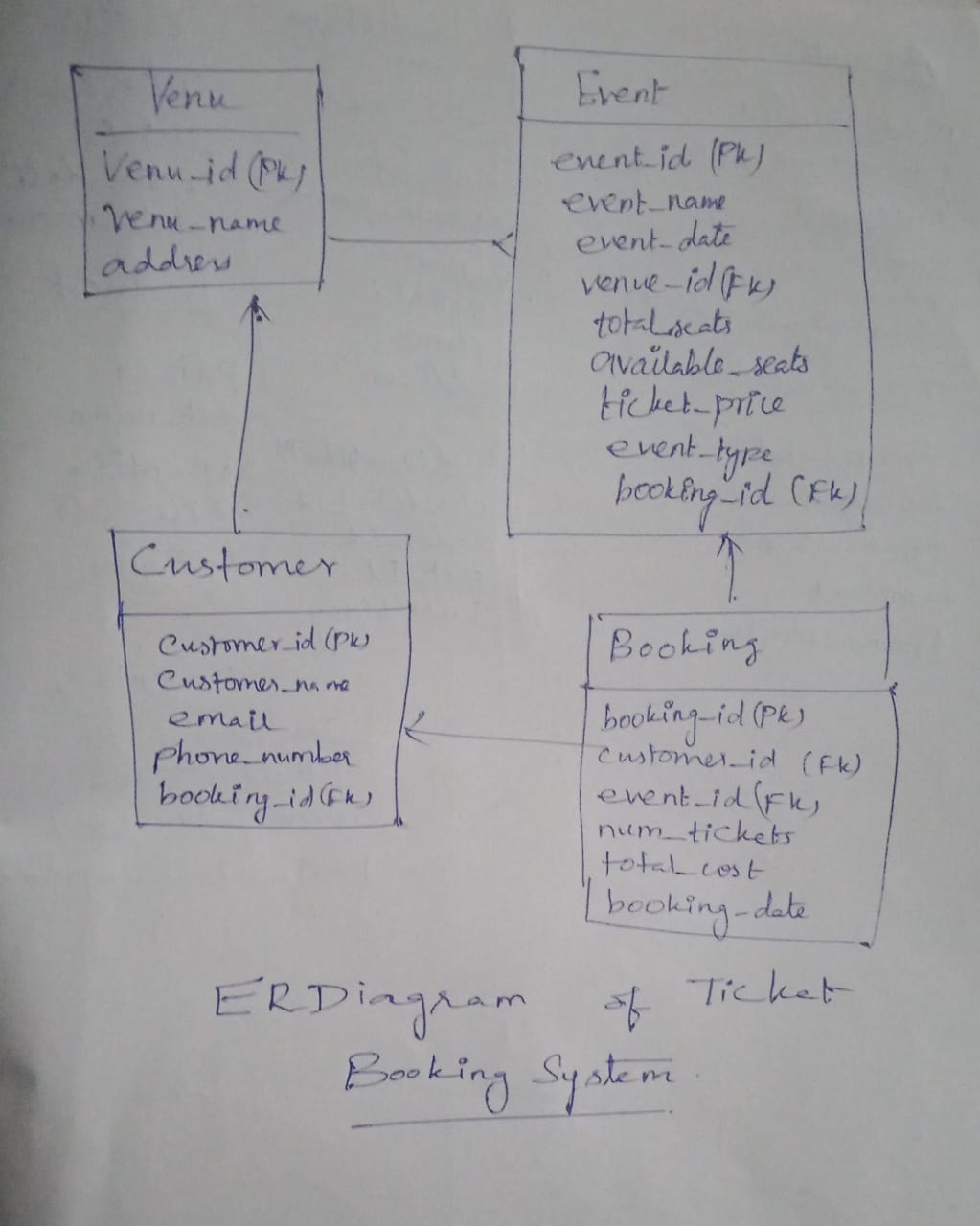
booking\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (event\_id) REFERENCES Event(event\_id)

);

3.ERD DIAGRAM



2. **Tasks 2: Select, Where, Between, AND, LIKE:**

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

2. Write a SQL query to list all Events.

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

4. Write a SQL query to select events name partial match with ‘cup’.

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

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

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

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

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

10. Write a SQL query to retrieve customer information whose phone number end with ‘000’

11. Write a SQL query to retrieve the events in order whose seat capacity more than 15000.

12. Write a SQL query to select events name not start with ‘x’, ‘y’, ‘z’

QUERIES

1. INSERT INTO Venu VALUES

(1, 'ABC Venue', 'Address AA'),

(2, 'BAC Venue', 'Address BB'),

(3, 'CAB Venue', 'Address CC'),

(4, 'MNC Venue', 'Address DD '),

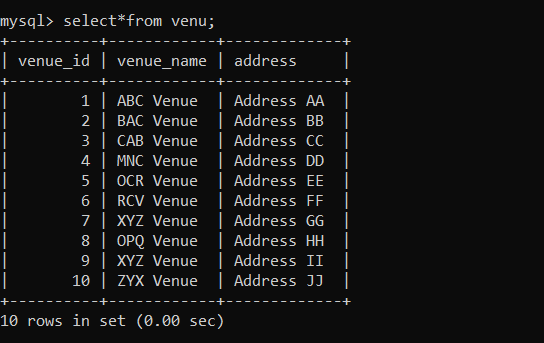
(5, 'OCR Venue', 'Address EE'),

(6, 'RCV Venue', 'Address FF'),

(7, 'XYZ Venue', 'Address GG'),

(8, 'OPQ Venue', 'Address HH'),

(9, 'XYZ Venue', 'Address II'),



INSERT INTO Customer VALUES

(1, 'Rani', 'rani@example.com','9123456780', 1),

(2, 'Ravi', 'ravi@example.com','9087654321', 2),

(3, 'Riya', 'riya@example.com', '9078654321', 3),

(4, 'Mani', 'Mani@example.com', '9085674321', 4),

(5, 'Chitra','Chitra@example.com','9807654321', 5),

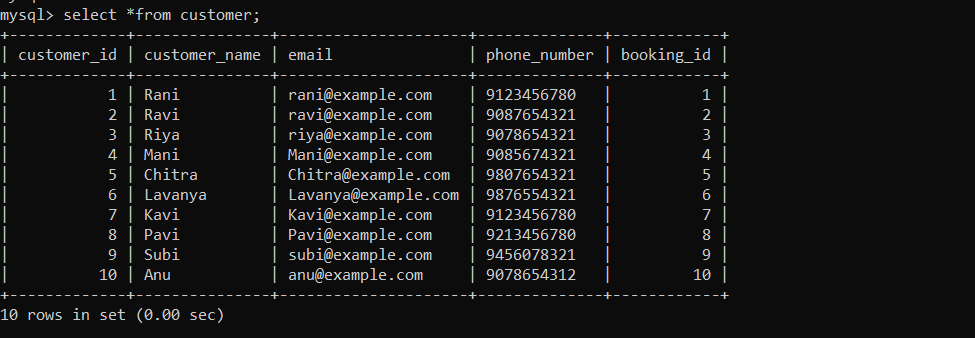
(6, 'Lavanya','Lavanya@example.com','9876554321', 6),

(7, 'Kavi', 'Kavi@example.com', '9123456780', 7),

(8, 'Pavi', 'Pavi@example.com', '9213456780', 8),

(9, 'Subi', 'subi@example.com', '9456078321', 9),

(10, 'Anu', 'anu@example.com', '9078654312', 10);



INSERT INTO Event VALUES

(1, 'Hollywood event', '2023-01-01', '12:00:00', 1, 2000, 1500, 2000.00, 'Movie', 1),

(2, 'Popstar event', '2023-02-01', '14:00:00', 2, 3000, 2000, 1500.00, 'Concert', 2),

(3, 'Bollywood Movie Night', '2023-03-15', '19:30:00', 3, 1000, 800, 500.00, 'Movie', 3),

(4, 'Cricket Championship', '2023-04-10', '15:00:00', 1, 20000, 15000, 1500.00, 'Sports', 4),

(5, 'Carnatic Music Concert', '2023-05-20', '18:00:00', 2, 500, 400, 1000.00, 'Concert', 5),

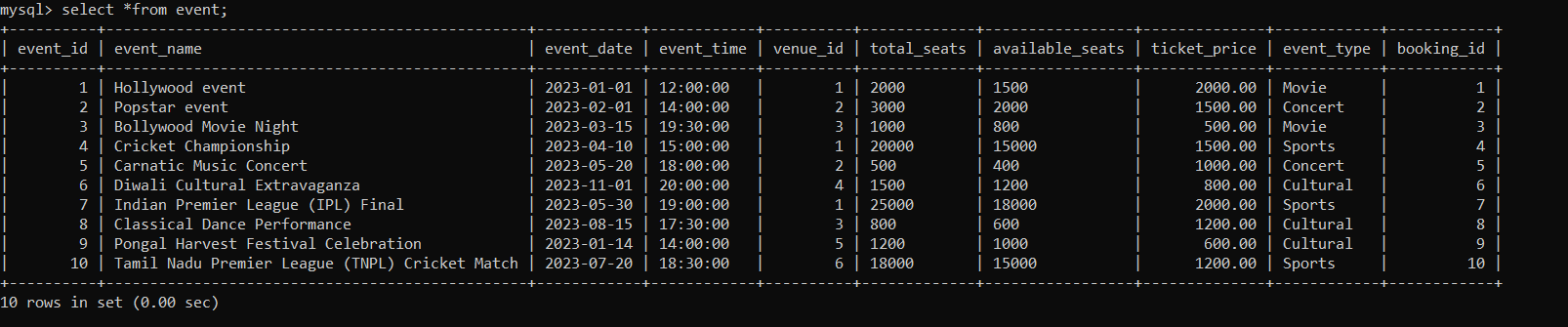
(6, 'Diwali Cultural Extravaganza', '2023-11-01', '20:00:00', 4, 1500, 1200, 800.00, 'Cultural', 6),

(7, 'Indian Premier League (IPL) Final', '2023-05-30', '19:00:00', 1, 25000, 18000, 2000.00, 'Sports', 7),

(8, 'Classical Dance Performance', '2023-08-15', '17:30:00', 3, 800, 600, 1200.00, 'Cultural', 8),

(9, 'Pongal Harvest Festival Celebration', '2023-01-14', '14:00:00', 5, 1200, 1000, 600.00, 'Cultural', 9),

(10, 'Tamil Nadu Premier League (TNPL) Cricket Match', '2023-07-20', '18:30:00', 6, 18000, 15000, 1200.00, 'Sports', 10);



INSERT INTO Booking VALUES

(1, 1, 1, 5, 1000.00, '2023-01-01'),

(2, 2, 2, 3, 600.00, '2023-02-01'),

(3, 3, 3, 9, 1800.00, '2023-11-03'),

(4, 4 ,4, 6, 1200.00, '2023-04-23'),

(5, 5, 5, 3, 600.00, '2023-05-18'),

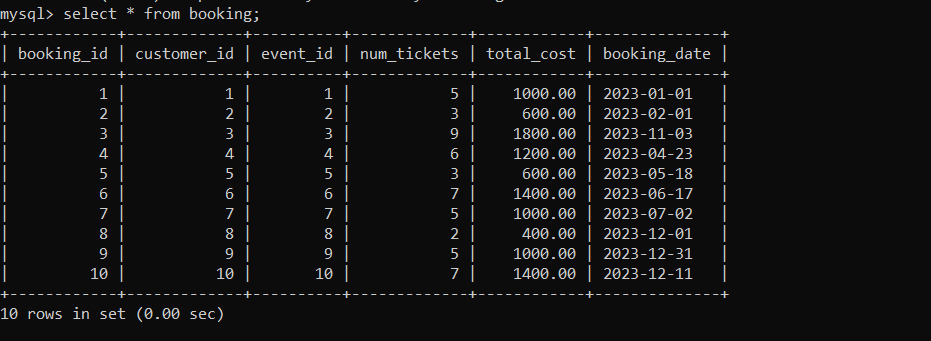
(6, 6, 6, 7, 1400.00, '2023-06-17'),

(7, 7, 7, 5, 1000.00, '2023-07-02'),

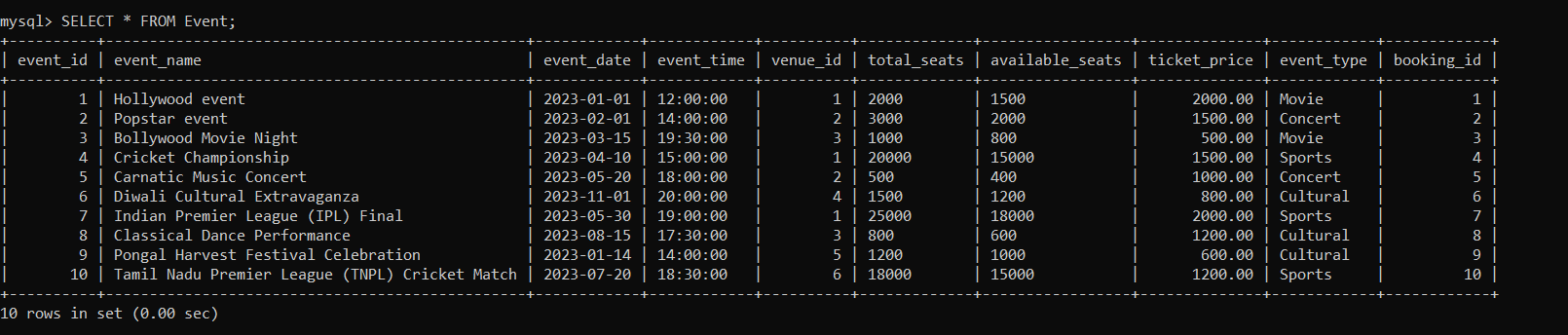
(8, 8, 8, 2, 400.00, '2023-12-01'),

(9, 9, 9, 5, 1000.00, '2023-12-31'),

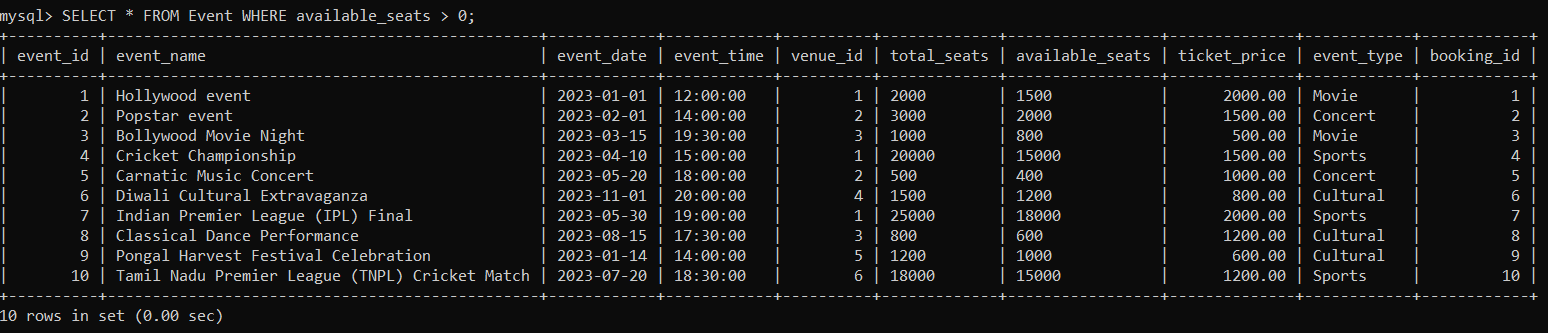
(10, 10, 10, 7, 1400.00, '2023-12-11');



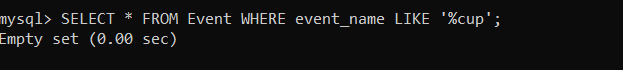
2.



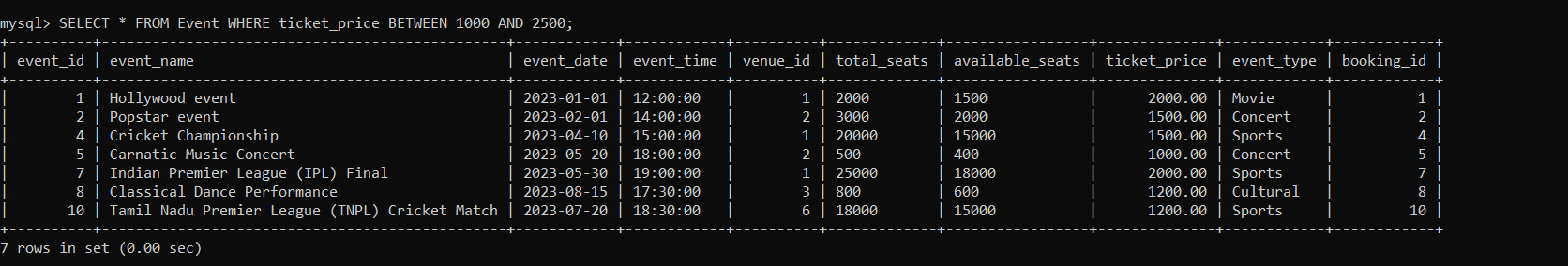
3.



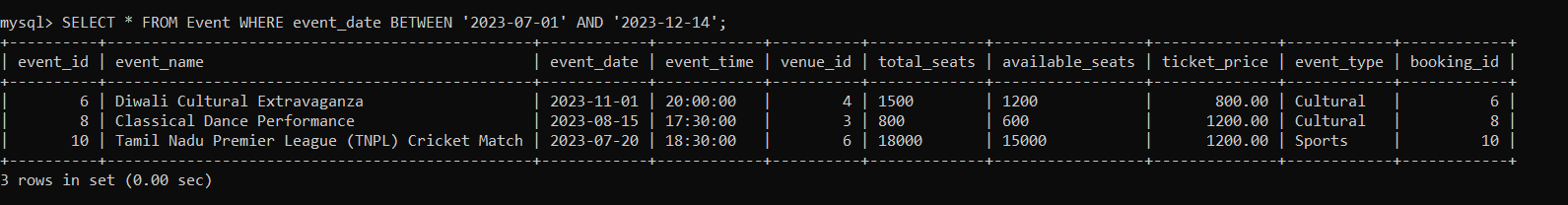
4.



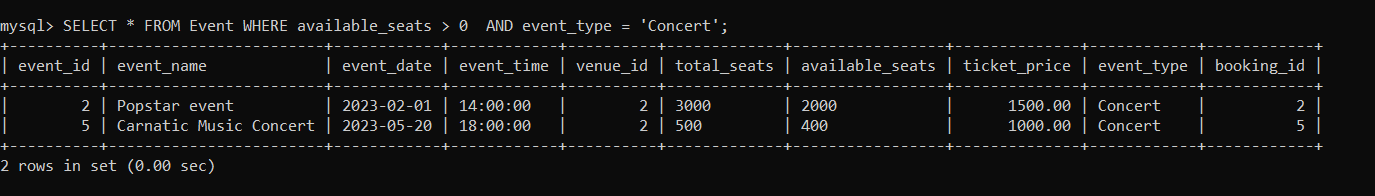
5.



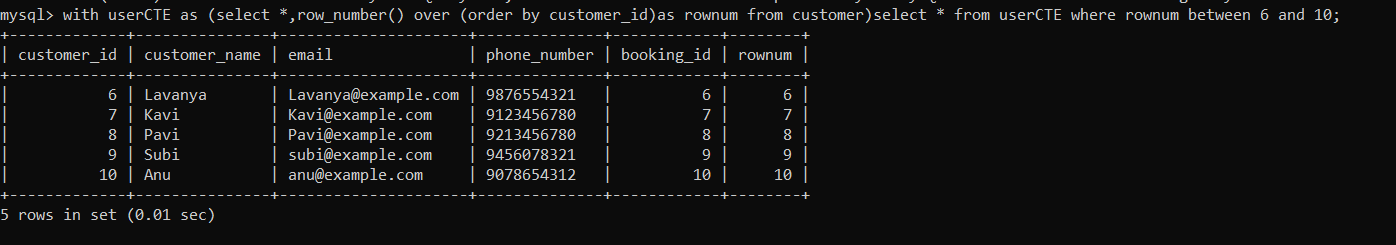
6.



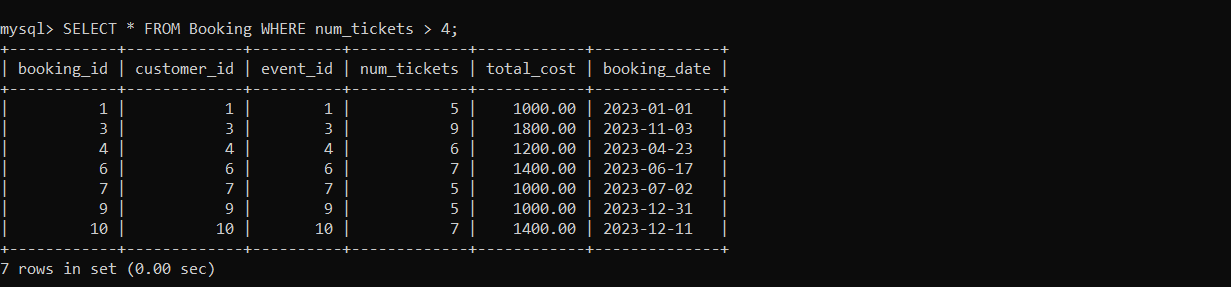
7.



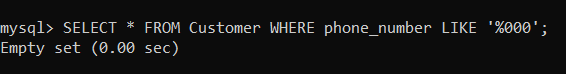
8.



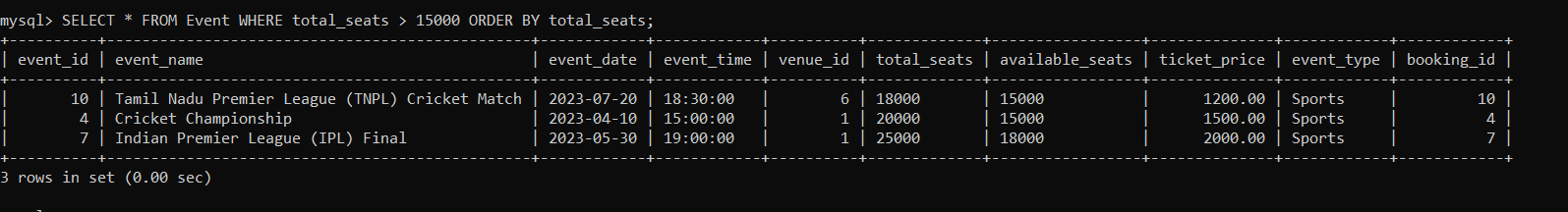
9.



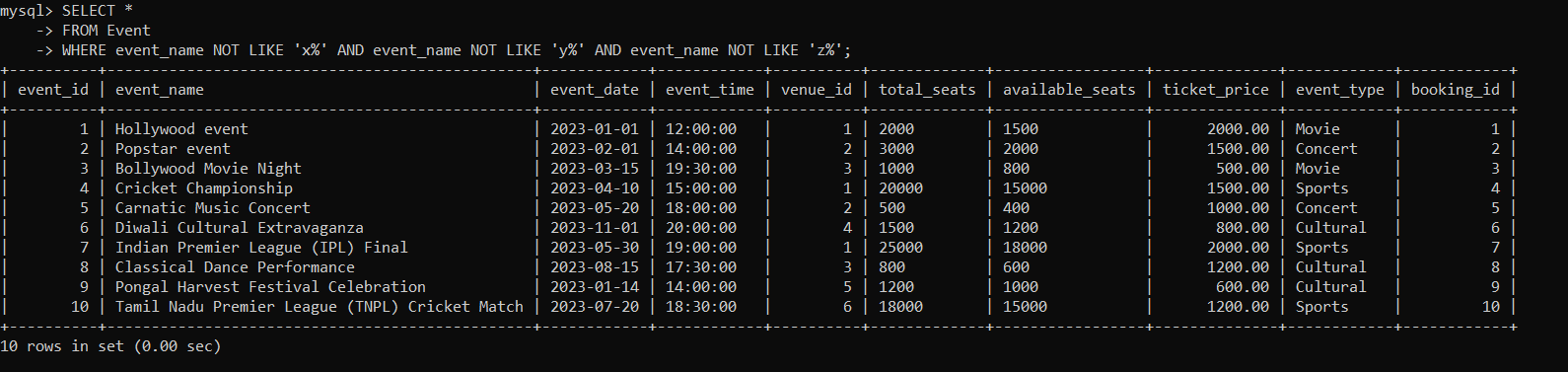
10.



11.



12.



Task 3

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

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

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

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

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

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

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

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

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

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

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

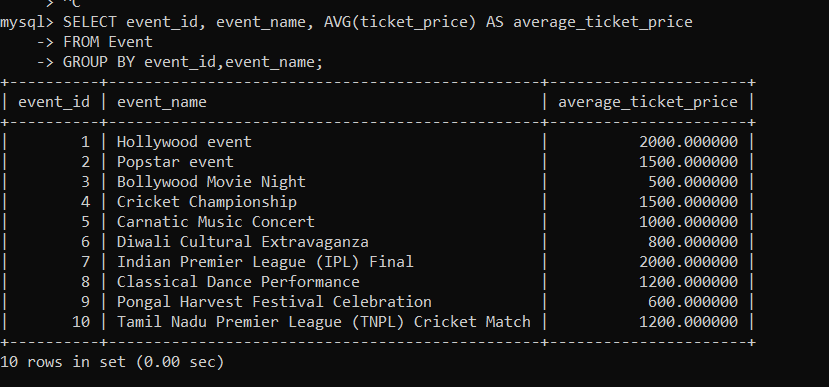
12. Write a SQL query to calculate the Total Revenue Generated by Events for Each User.

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

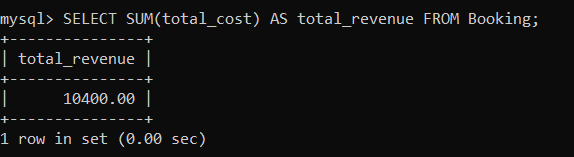
14. Write a SQL query to list Users and the Total Number of Tickets They've Purchased in the Last 30 Days.

QUERIES

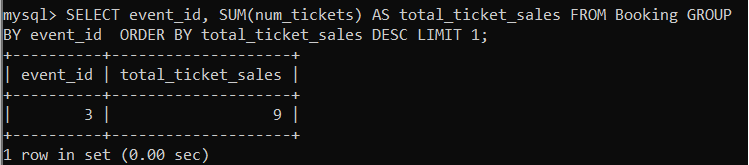
1.



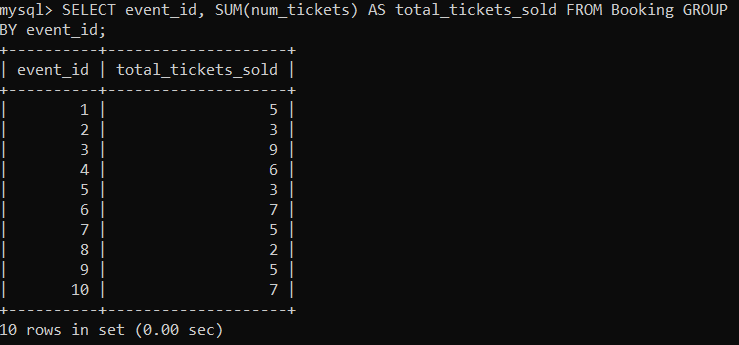
2.



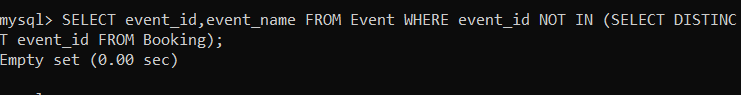
3.



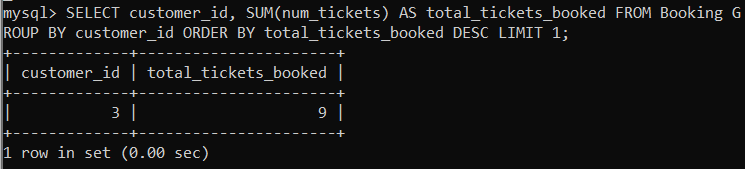
4.



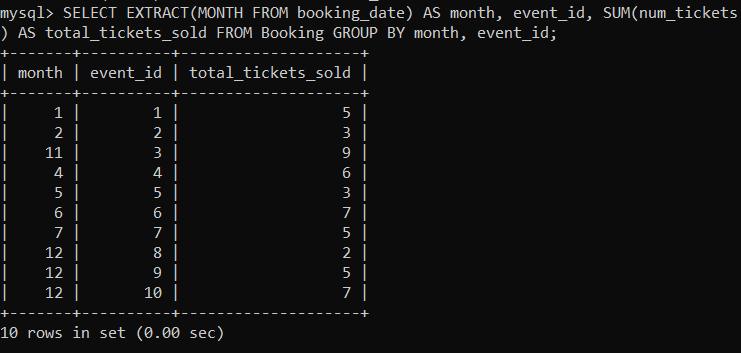
5.



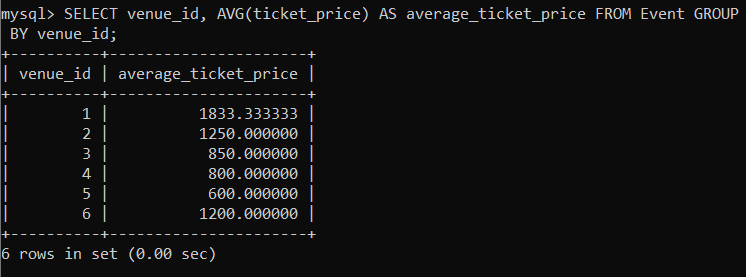
6.



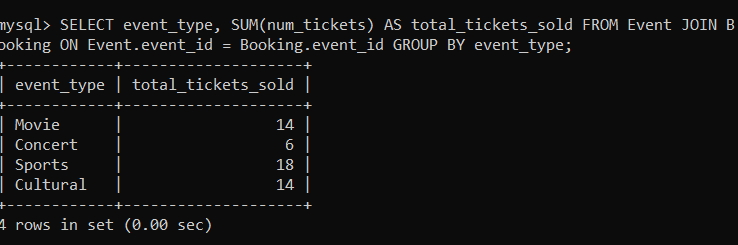
7.



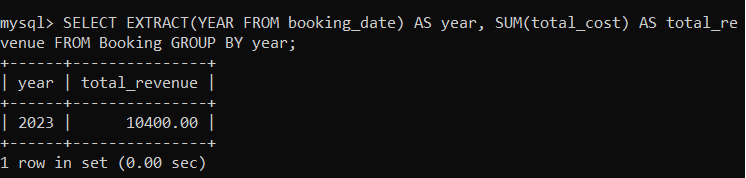
8.



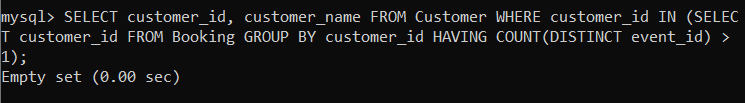
9.



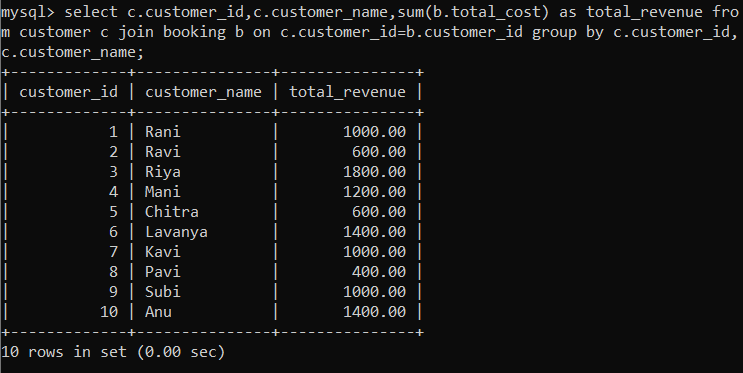
10.



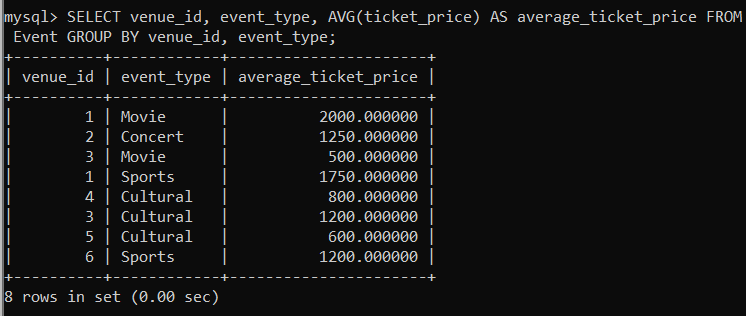
11.



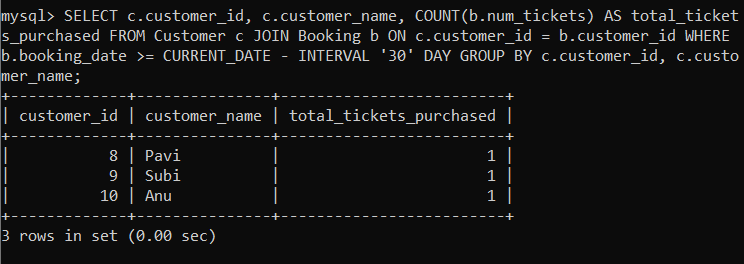
12.



13.



14.



**Tasks 4: Subquery and its types**

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

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

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

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

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

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

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

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

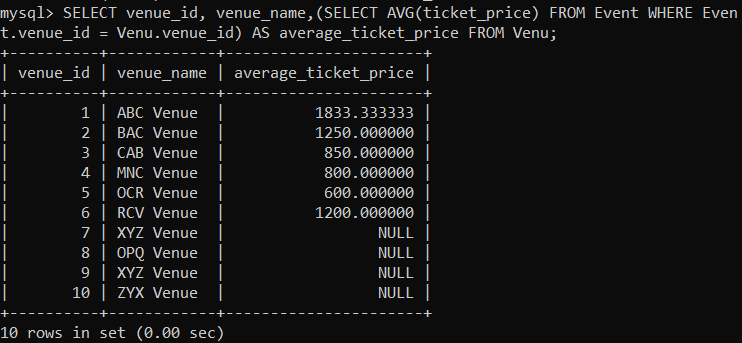
9. List Users Who Have Booked Tickets for Events in a Given Venue Using a Subquery in the WHERE Clause.

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

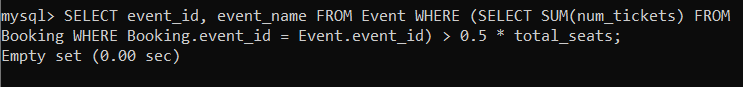
11. Find Users Who Have Booked Tickets for Events in each Month Using a Subquery with DATE\_FORMAT.

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

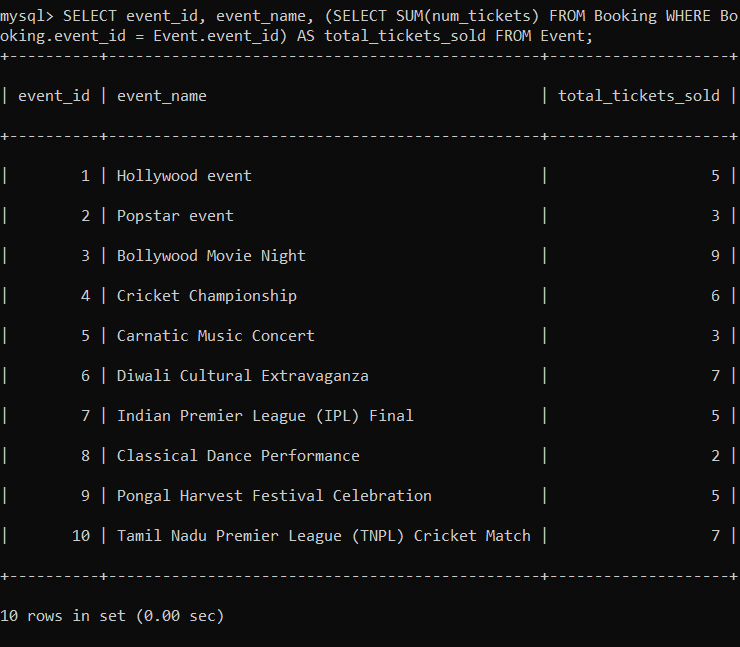
1.



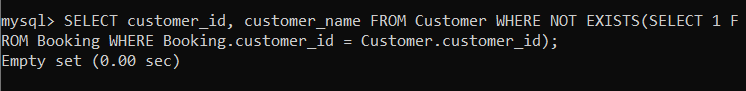
2.



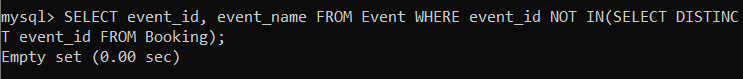
3.



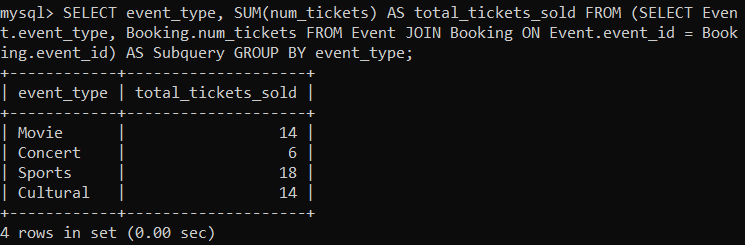
4.



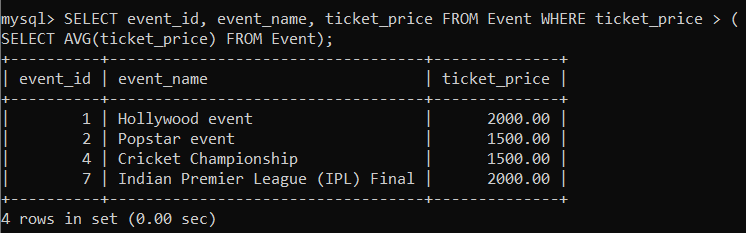
5.



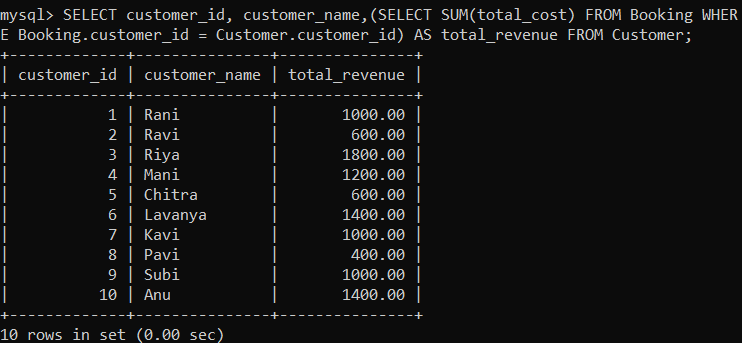
6.



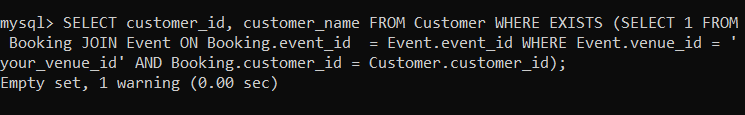
7.



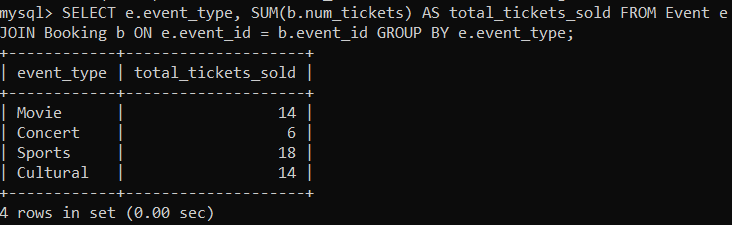
8.



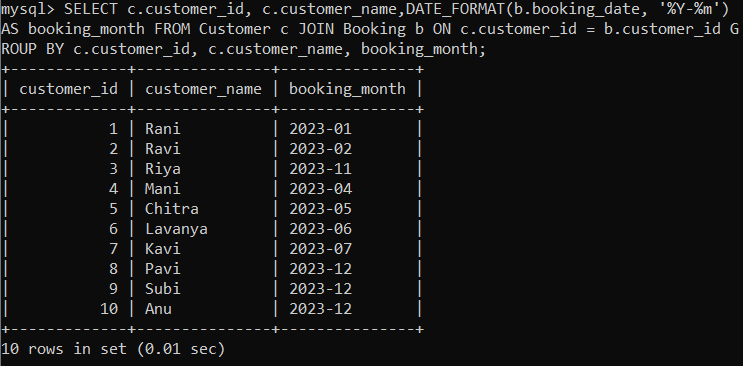
9.



10.



11.



12.

