Ticket Booking System

Name: J305-Samiksha Sandeep Patil

Superset ID: 5273555

DATABASE TABLES

Tasks 1: Database Design:

1. Create the database named "TicketBookingSystem"

```
CREATE DATABASE TicketBookingSystem;
USE TicketBookingSystem;
```

```
mysql> CREATE DATABASE TicketBookingSystem;
ERROR 1007 (HY000): Can't create database 'ticketbookingsystem'; database exists
mysql> USE TicketBookingSystem;
Database changed
mysql>
```

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

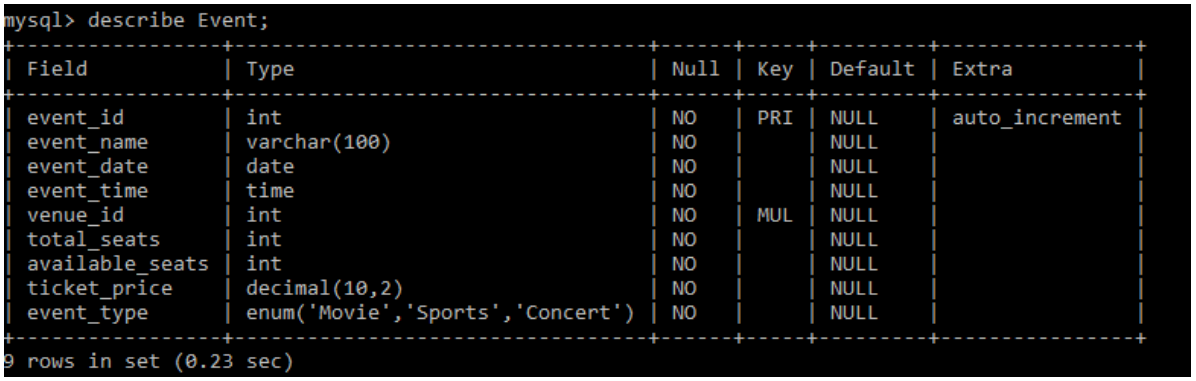
- **Venue**

```
CREATE TABLE Venue (venue_id INT PRIMARY KEY AUTO_INCREMENT,
venue_name VARCHAR(100) NOT NULL,address VARCHAR(255) NOT NULL);
```

```
mysql> describe Venue;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| venue_id   | int          | NO   | PRI | NULL    | auto_increment |
| venue_name | varchar(100) | NO   |     | NULL    |                |
| address    | varchar(255) | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

- **Event**

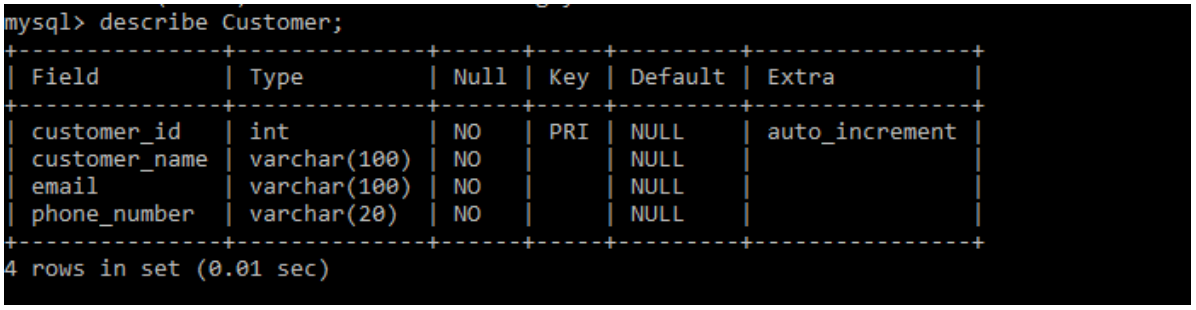
```
CREATE TABLE Event (
event_id INT PRIMARY KEY AUTO_INCREMENT,
event_name VARCHAR(100) NOT NULL,
event_date DATE NOT NULL,event_time TIME NOT NULL,
venue_id INT NOT NULL,total_seats INT NOT NULL,
available_seats INT NOT NULL,ticket_price DECIMAL(10, 2) NOT NULL,
event_type ENUM('Movie', 'Sports', 'Concert') NOT NULL,
FOREIGN KEY (venue_id) REFERENCES Venue(venue_id));
```



```
mysql> describe Event;
+-----------------+---------------------------------+------+-----+---------+----------------+
| Field           | Type                            | Null | Key | Default | Extra          |
+-----------------+---------------------------------+------+-----+---------+----------------+
| event_id        | int                             | NO   | PRI | NULL    | auto_increment |
| event_name      | varchar(100)                    | NO   |     | NULL    |                |
| event_date      | date                            | NO   |     | NULL    |                |
| event_time      | time                            | NO   |     | NULL    |                |
| venue_id        | int                             | NO   | MUL | NULL    |                |
| total_seats     | int                             | NO   |     | NULL    |                |
| available_seats | int                             | NO   |     | NULL    |                |
| ticket_price    | decimal(10,2)                   | NO   |     | NULL    |                |
| event_type      | enum('Movie','Sports','Concert') | NO   |     | NULL    |                |
+-----------------+---------------------------------+------+-----+---------+----------------+
9 rows in set (0.23 sec)
```

- **Customers**

```
CREATE TABLE Customer (customer_id INT PRIMARY KEY AUTO_INCREMENT
customer_name VARCHAR(100) NOT NULL,email VARCHAR(100) NOT NULL,
phone_number VARCHAR(20) NOT NULL);
```



```
mysql> describe Customer;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| customer_id   | int          | NO   | PRI | NULL    | auto_increment |
| customer_name | varchar(100) | NO   |     | NULL    |                |
| email         | varchar(100) | NO   |     | NULL    |                |
| phone_number  | varchar(20)  | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

•

Booking

```
CREATE TABLE Booking (booking_id INT PRIMARY KEY AUTO_INCREMENT,cu
event_id INT NOT NULL,
num_tickets INT NOT NULL,total_cost DECIMAL(10, 2) NOT NULL,
booking_date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
FOREIGN KEY (event_id) REFERENCES Event(event_id));
```

```
mysql> describe Booking;
+--------------+---------------+------+-----+-------------------+-------------------+
| Field        | Type          | Null | Key | Default           | Extra             |
+--------------+---------------+------+-----+-------------------+-------------------+
| booking_id   | int           | NO   | PRI | NULL              | auto_increment    |
| customer_id  | int           | NO   | MUL | NULL              |                   |
| event_id     | int           | NO   | MUL | NULL              |                   |
| num_tickets  | int           | NO   |     | NULL              |                   |
| total_cost   | decimal(10,2) | NO   |     | NULL              |                   |
| booking_date | datetime      | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+--------------+---------------+------+-----+-------------------+-------------------+
6 rows in set (0.01 sec)
```

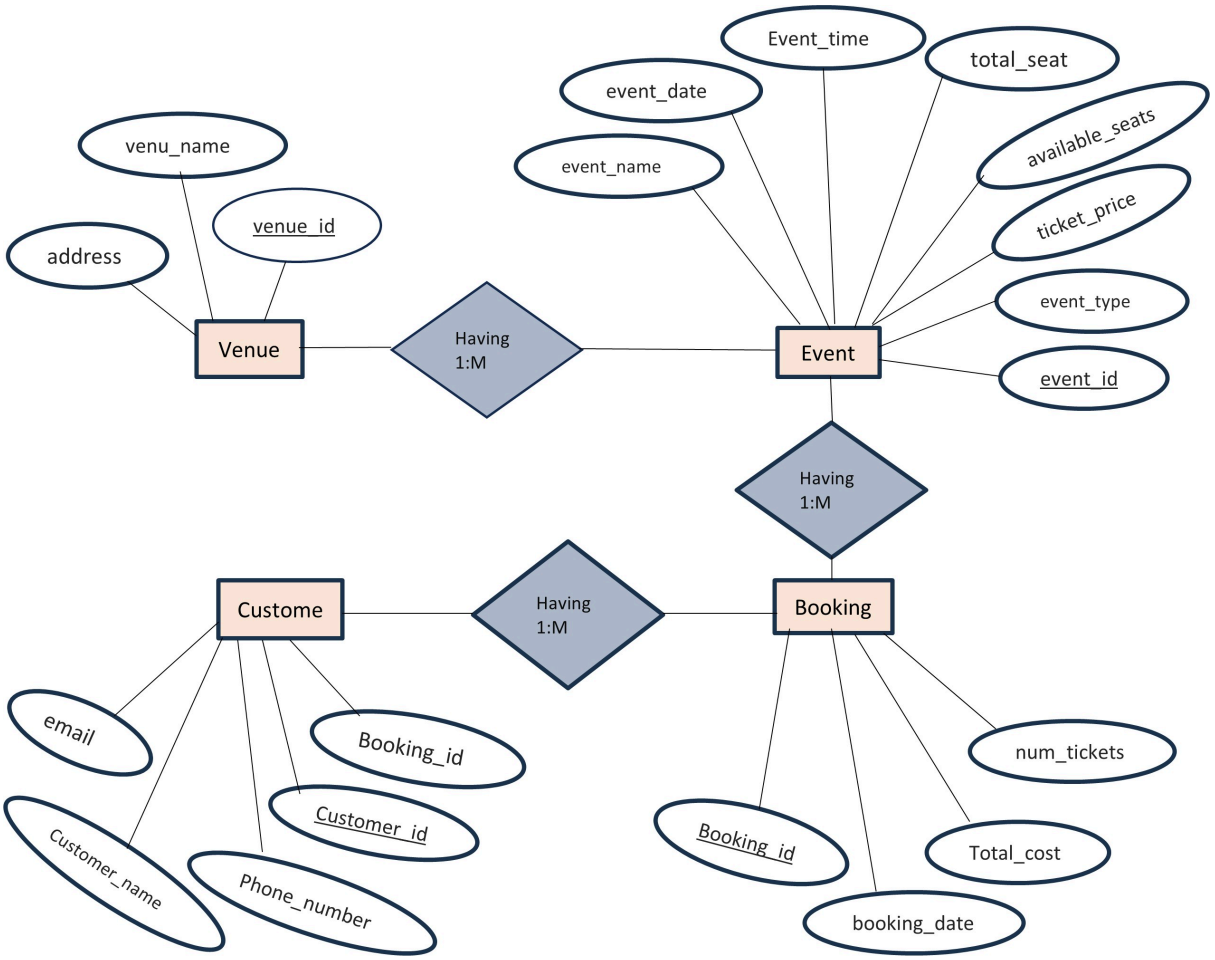
3. Create an ERD (Entity Relationship Diagram) for the database

1. **Venue Table**:
   - Primary Key: venue_id
   - Attributes: venue_name, address
2. **Event Table**:
   - Primary Key: event_id
   - Foreign Key: venue_id (references Venue)
   - Attributes: event_name, event_date, event_time, total_seats, available_seats, ticket_price, event_type
3. **Customer Table**:
   - Primary Key: customer_id
   - Attributes: customer_name, email, phone_number
4. **Booking Table**:
   - Primary Key: booking_id

- Foreign Keys: customer_id (references Customer), event_id (references Event)
- Attributes: num_tickets, total_cost, booking_date

The relationships between these entities are:

- One **Venue** can host many **Events** (**One-to-Many**)
- One **Event** can have many **Bookings** (**One-to-Many**)
- One **Customer** can make many **Bookings** (**One-to-Many**)



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

```
--Add booking_id foreign key to Event table (as mentioned in requirements)
ALTER TABLE Event ADD COLUMN booking_id INT;
ALTER TABLE Event ADD FOREIGN KEY (booking_id)
REFERENCES Booking(booking_id);
```

```
mysql> describe Event;
+----------------+---------------------------------+------+-----+---------+----------------+
| Field          | Type                            | Null | Key | Default | Extra          |
+----------------+---------------------------------+------+-----+---------+----------------+
| event_id       | int                             | NO   | PRI | NULL    | auto_increment |
| event_name     | varchar(100)                    | NO   |     | NULL    |                |
| event_date     | date                            | NO   |     | NULL    |                |
| event_time     | time                            | NO   |     | NULL    |                |
| venue_id       | int                             | NO   | MUL | NULL    |                |
| total_seats    | int                             | NO   |     | NULL    |                |
| available_seats| int                             | NO   |     | NULL    |                |
| ticket_price   | decimal(10,2)                   | NO   |     | NULL    |                |
| event_type     | enum('Movie','Sports','Concert') | NO   |     | NULL    |                |
| booking_id     | int                             | YES  | MUL | NULL    |                |
+----------------+---------------------------------+------+-----+---------+----------------+
10 rows in set (0.01 sec)
```

```
--Add booking_id foreign key to Customer table (as mentioned in requirement
ALTER TABLE Customer ADD COLUMN booking_id INT;
ALTER TABLE Customer ADD FOREIGN KEY (booking_id)
REFERENCES Booking(booking_id);
```

```
mysql> describe Customer;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| customer_id   | int          | NO   | PRI | NULL    | auto_increment |
| customer_name | varchar(100) | NO   |     | NULL    |                |
| email         | varchar(100) | NO   |     | NULL    |                |
| phone_number  | varchar(20)  | NO   |     | NULL    |                |
| booking_id    | int          | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

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

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

Insert Venue:

```
INSERT INTO Venue (venue_name, address) VALUES
('National Stadium', '123 Sports Avenue, Delhi'),
('Galaxy Cinemas', '456 Movie Lane, Mumbai'),
('Music Hall', '789 Concert Road, Bangalore'),
('Olympic Arena', '101 Olympic Street, Chennai'),
('Cityview Theater', '202 Broadway Avenue, Hyderabad'),
('Concert Dome', '303 Music Street, Kolkata'),
```

```
('Sports Complex', '404 Game Boulevard, Pune'),
('Grand Auditorium', '505 Show Street, Ahmedabad'),
('Festival Grounds', '606 Festival Road, Jaipur'),
('Community Center', '707 Community Avenue, Lucknow');
```

```
mysql> select * from Venue;
+----------+------------------+--------------------------------+
| venue_id | venue_name       | address                        |
+----------+------------------+--------------------------------+
|        1 | National Stadium | 123 Sports Avenue, Delhi       |
|        2 | Galaxy Cinemas   | 456 Movie Lane, Mumbai         |
|        3 | Music Hall       | 789 Concert Road, Bangalore    |
|        4 | Olympic Arena    | 101 Olympic Street, Chennai    |
|        5 | Cityview Theater | 202 Broadway Avenue, Hyderabad |
|        6 | Concert Dome     | 303 Music Street, Kolkata      |
|        7 | Sports Complex   | 404 Game Boulevard, Pune       |
|        8 | Grand Auditorium | 505 Show Street, Ahmedabad     |
|        9 | Festival Grounds | 606 Festival Road, Jaipur      |
|       10 | Community Center | 707 Community Avenue, Lucknow  |
+----------+------------------+--------------------------------+
10 rows in set (0.11 sec)
```

Insert Event:

```
INSERT INTO Event (event_name, event_date, event_time, venue_id,
total_seats, available_seats, ticket_price, event_type) VALUES
('World Cup Final', '2025-04-15', '14:00:00', 1, 20000, 5000, 2000.00, 'Sports
('Avengers 5 Premiere', '2025-04-20', '18:30:00', 2, 300, 50, 500.00, 'Movie'),
('Rock Concert', '2025-05-01', '20:00:00', 3, 5000, 2000, 1500.00, 'Concert'),
('Olympic Qualifying Match', '2025-05-10', '16:00:00', 4, 15000, 7000, 1200.00
('Bollywood Night', '2025-05-15', '19:00:00', 5, 400, 100, 800.00, 'Concert'),
('Classical Music Night', '2025-05-20', '18:30:00', 6, 1000, 300, 2200.00, 'Cor
('T20 Cricket Cup', '2025-06-01', '15:00:00', 7, 18000, 3000, 1800.00, 'Sports'
('Film Festival Screening', '2025-06-10', '17:00:00', 8, 600, 200, 650.00, 'Movi
('Summer Music Festival', '2025-06-15', '16:00:00', 9, 8000, 2500, 2500.00, '(
('Community Theater Play', '2025-06-20', '19:30:00', 10, 200, 50, 350.00, 'Mo
```

```
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 | World Cup Final          | 2025-04-15 | 14:00:00   |        1 |       20000 |           5000 |      2000.00 | Sports     |       NULL |
|        2 | Avengers 5 Premiere      | 2025-04-20 | 18:30:00   |        2 |         300 |             50 |       500.00 | Movie      |       NULL |
|        3 | Rock Concert             | 2025-05-01 | 20:00:00   |        3 |        5000 |           2000 |      1500.00 | Concert    |       NULL |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |           7000 |      1200.00 | Sports     |       NULL |
|        5 | Bollywood Night          | 2025-05-15 | 19:00:00   |        5 |         400 |            100 |       800.00 | Concert    |       NULL |
|        6 | Classical Music Night    | 2025-05-20 | 18:30:00   |        6 |        1000 |            300 |      2200.00 | Concert    |       NULL |
|        7 | T20 Cricket Cup          | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |       NULL |
|        8 | Film Festival Screening  | 2025-06-10 | 17:00:00   |        8 |         600 |            200 |       650.00 | Movie      |       NULL |
|        9 | Summer Music Festival    | 2025-06-15 | 16:00:00   |        9 |        8000 |           2500 |      2500.00 | Concert    |       NULL |
|       10 | Community Theater Play   | 2025-06-20 | 19:30:00   |       10 |         200 |             50 |       350.00 | Movie      |       NULL |
+----------+--------------------------+------------+------------+----------+-------------+----------------+--------------+------------+------------+
10 rows in set (0.02 sec)
```

Insert Customers:

```
INSERT INTO Customer (customer_name, email, phone_number) VALUES
('Samiksha Patil', 'samikshapatil419@gmail.com', '7097381008'),
('Priya Patel', 'priya.patel@gmail.com', '8765432109'),
('Amit Kumar', 'amit.kumar@yahoo.com', '7654321098'),
('Sneha Gupta', 'sneha.gupta@hotmail.com', '6543210987'),
('Vikram Singh', 'vikram.singh@gmail.com', '5432109876'),
('Neha Verma', 'neha.verma@outlook.com', '4321098765'),
('Ravi Kapoor', 'ravi.kapoor@gmail.com', '3210987654'),
('Anjali Desai', 'anjali.desai@yahoo.com', '2109876543'),
('Sanjay Joshi', 'sanjay.joshi@gmail.com', '1098765432'),
('Meera Reddy', 'meera.reddy@hotmail.com', '9876543000');
```

```
mysql> select * from customer;
+-------------+---------------+---------------------------+--------------+------------+
| customer_id | customer_name | email                     | phone_number | booking_id |
+-------------+---------------+---------------------------+--------------+------------+
|           1 | Samiksha Patil | samikshapatil419@gmail.com | 7097381008   |          1 |
|           2 | Priya Patel    | priya.patel@gmail.com      | 8765432109   |          2 |
|           3 | Amit Kumar     | amit.kumar@yahoo.com       | 7654321098   |          3 |
|           4 | Sneha Gupta    | sneha.gupta@hotmail.com    | 6543210987   |          4 |
|           5 | Vikram Singh   | vikram.singh@gmail.com     | 5432109876   |          5 |
|           6 | Neha Verma     | neha.verma@outlook.com     | 4321098765   |          6 |
|           7 | Ravi Kapoor    | ravi.kapoor@gmail.com      | 3210987654   |          7 |
|           8 | Anjali Desai   | anjali.desai@yahoo.com     | 2109876543   |          8 |
|           9 | Sanjay Joshi   | sanjay.joshi@gmail.com     | 1098765432   |          9 |
|          10 | Meera Reddy    | meera.reddy@hotmail.com    | 9876543000   |         10 |
+-------------+---------------+---------------------------+--------------+------------+
10 rows in set (0.02 sec)
```

Insert Booking:

```
INSERT INTO Booking (customer_id, event_id, num_tickets, total_cost, booking
(1, 1, 3, 6000.00, '2025-03-15 10:30:00'),
(2, 2, 2, 1000.00, '2025-03-16 11:45:00'),
(3, 3, 5, 7500.00, '2025-03-17 09:15:00'),
(4, 4, 4, 4800.00, '2025-03-18 14:20:00'),
(5, 5, 2, 1600.00, '2025-03-19 16:30:00'),
(6, 6, 3, 6600.00, '2025-03-20 10:00:00'),
(7, 7, 6, 10800.00, '2025-03-21 12:15:00'),
(8, 8, 2, 1300.00, '2025-03-22 17:45:00'),
(9, 9, 4, 10000.00, '2025-03-23 09:30:00'),
(10, 10, 5, 1750.00, '2025-03-24 11:00:00');
```

```
mysql> select * from Booking;
+------------+-------------+----------+-------------+------------+---------------------+
| booking_id | customer_id | event_id | num_tickets | total_cost | booking_date        |
+------------+-------------+----------+-------------+------------+---------------------+
|          1 |           1 |        1 |           3 |    6000.00 | 2025-03-15 10:30:00 |
|          2 |           2 |        2 |           2 |    1000.00 | 2025-03-16 11:45:00 |
|          3 |           3 |        3 |           5 |    7500.00 | 2025-03-17 09:15:00 |
|          4 |           4 |        4 |           4 |    4800.00 | 2025-03-18 14:20:00 |
|          5 |           5 |        5 |           2 |    1600.00 | 2025-03-19 16:30:00 |
|          6 |           6 |        6 |           3 |    6600.00 | 2025-03-20 10:00:00 |
|          7 |           7 |        7 |           6 |   10800.00 | 2025-03-21 12:15:00 |
|          8 |           8 |        8 |           2 |    1300.00 | 2025-03-22 17:45:00 |
|          9 |           9 |        9 |           4 |   10000.00 | 2025-03-23 09:30:00 |
|         10 |          10 |       10 |           5 |    1750.00 | 2025-03-24 11:00:00 |
+------------+-------------+----------+-------------+------------+---------------------+
10 rows in set (0.00 sec)
```

```
-- Update booking_id in Event and Customer tables
-- (This step is needed because of the circular foreign key references in th
e schema)
UPDATE Event SET booking_id = 1 WHERE event_id = 1;
UPDATE Event SET booking_id = 2 WHERE event_id = 2;
UPDATE Event SET booking_id = 3 WHERE event_id = 3;
UPDATE Event SET booking_id = 4 WHERE event_id = 4;
UPDATE Event SET booking_id = 5 WHERE event_id = 5;
UPDATE Event SET booking_id = 6 WHERE event_id = 6;
UPDATE Event SET booking_id = 7 WHERE event_id = 7;
UPDATE Event SET booking_id = 8 WHERE event_id = 8;
UPDATE Event SET booking_id = 9 WHERE event_id = 9;
UPDATE Event SET booking_id = 10 WHERE event_id = 10;

UPDATE Customer SET booking_id = 1 WHERE customer_id = 1;
UPDATE Customer SET booking_id = 2 WHERE customer_id = 2;
UPDATE Customer SET booking_id = 3 WHERE customer_id = 3;
UPDATE Customer SET booking_id = 4 WHERE customer_id = 4;
UPDATE Customer SET booking_id = 5 WHERE customer_id = 5;
UPDATE Customer SET booking_id = 6 WHERE customer_id = 6;
UPDATE Customer SET booking_id = 7 WHERE customer_id = 7;
UPDATE Customer SET booking_id = 8 WHERE customer_id = 8;
UPDATE Customer SET booking_id = 9 WHERE customer_id = 9;
UPDATE Customer SET booking_id = 10 WHERE customer_id = 10;
```

```
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 | World Cup Final          | 2025-04-15 | 14:00:00   |        1 |       20000 |           5000 |      2000.00 | Sports     |          1 |
|        2 | Avengers 5 Premiere      | 2025-04-20 | 18:30:00   |        2 |         300 |             50 |       500.00 | Movie      |          2 |
|        3 | Rock Concert             | 2025-05-01 | 20:00:00   |        3 |        5000 |           2000 |      1500.00 | Concert    |          3 |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |           7000 |      1200.00 | Sports     |          4 |
|        5 | Bollywood Night          | 2025-05-15 | 19:00:00   |        5 |         400 |            100 |       800.00 | Concert    |          5 |
|        6 | Classical Music Night    | 2025-05-20 | 18:30:00   |        6 |        1000 |            300 |      2200.00 | Concert    |          6 |
|        7 | T20 Cricket Cup          | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |          7 |
|        8 | Film Festival Screening  | 2025-06-10 | 17:00:00   |        8 |         600 |            200 |       650.00 | Movie      |          8 |
|        9 | Summer Music Festival    | 2025-06-15 | 16:00:00   |        9 |        8000 |           2500 |      2500.00 | Concert    |          9 |
|       10 | Community Theater Play   | 2025-06-20 | 19:30:00   |       10 |         200 |             50 |       350.00 | Movie      |         10 |
+----------+--------------------------+------------+------------+----------+-------------+----------------+--------------+------------+------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from customer;
+-------------+---------------+---------------------------+--------------+------------+
| customer_id | customer_name | email                     | phone_number | booking_id |
+-------------+---------------+---------------------------+--------------+------------+
|           1 | Samiksha Patil | samikshapatil419@gmail.com | 7097381008   |          1 |
|           2 | Priya Patel    | priya.patel@gmail.com      | 8765432109   |          2 |
|           3 | Amit Kumar     | amit.kumar@yahoo.com       | 7654321098   |          3 |
|           4 | Sneha Gupta    | sneha.gupta@hotmail.com    | 6543210987   |          4 |
|           5 | Vikram Singh   | vikram.singh@gmail.com     | 5432109876   |          5 |
|           6 | Neha Verma     | neha.verma@outlook.com     | 4321098765   |          6 |
|           7 | Ravi Kapoor    | ravi.kapoor@gmail.com      | 3210987654   |          7 |
|           8 | Anjali Desai   | anjali.desai@yahoo.com     | 2109876543   |          8 |
|           9 | Sanjay Joshi   | sanjay.joshi@gmail.com     | 1098765432   |          9 |
|          10 | Meera Reddy    | meera.reddy@hotmail.com    | 9876543000   |         10 |
+-------------+---------------+---------------------------+--------------+------------+
10 rows in set (0.02 sec)
```

2. Write a SQL query to list all Events.

```
SELECT * FROM Event;
```

```
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 | World Cup Final          | 2025-04-15 | 14:00:00   |        1 |       20000 |           5000 |      2000.00 | Sports     |          1 |
|        2 | Avengers 5 Premiere      | 2025-04-20 | 18:30:00   |        2 |         300 |             50 |       500.00 | Movie      |          2 |
|        3 | Rock Concert             | 2025-05-01 | 20:00:00   |        3 |        5000 |           2000 |      1500.00 | Concert    |          3 |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |           7000 |      1200.00 | Sports     |          4 |
|        5 | Bollywood Night          | 2025-05-15 | 19:00:00   |        5 |         400 |            100 |       800.00 | Concert    |          5 |
|        6 | Classical Music Night    | 2025-05-20 | 18:30:00   |        6 |        1000 |            300 |      2200.00 | Concert    |          6 |
|        7 | T20 Cricket Cup          | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |          7 |
|        8 | Film Festival Screening  | 2025-06-10 | 17:00:00   |        8 |         600 |            200 |       650.00 | Movie      |          8 |
|        9 | Summer Music Festival    | 2025-06-15 | 16:00:00   |        9 |        8000 |           2500 |      2500.00 | Concert    |          9 |
|       10 | Community Theater Play   | 2025-06-20 | 19:30:00   |       10 |         200 |             50 |       350.00 | Movie      |         10 |
+----------+--------------------------+------------+------------+----------+-------------+----------------+--------------+------------+------------+
10 rows in set (0.00 sec)
```

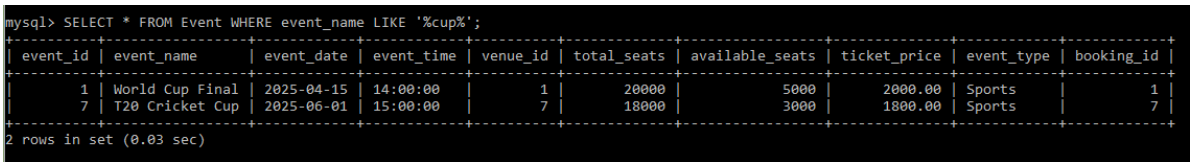
3. Select events with available tickets

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

```
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 | World Cup Final          | 2025-04-15 | 14:00:00   |        1 |       20000 |           5000 |      2000.00 | Sports     |          1 |
|        2 | Avengers 5 Premiere      | 2025-04-20 | 18:30:00   |        2 |         300 |             50 |       500.00 | Movie      |          2 |
|        3 | Rock Concert             | 2025-05-01 | 20:00:00   |        3 |        5000 |           2000 |      1500.00 | Concert    |          3 |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |           7000 |      1200.00 | Sports     |          4 |
|        5 | Bollywood Night          | 2025-05-15 | 19:00:00   |        5 |         400 |            100 |       800.00 | Concert    |          5 |
|        6 | Classical Music Night    | 2025-05-20 | 18:30:00   |        6 |        1000 |            300 |      2200.00 | Concert    |          6 |
|        7 | T20 Cricket Cup          | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |          7 |
|        8 | Film Festival Screening  | 2025-06-10 | 17:00:00   |        8 |         600 |            200 |       650.00 | Movie      |          8 |
|        9 | Summer Music Festival    | 2025-06-15 | 16:00:00   |        9 |        8000 |           2500 |      2500.00 | Concert    |          9 |
|       10 | Community Theater Play   | 2025-06-20 | 19:30:00   |       10 |         200 |             50 |       350.00 | Movie      |         10 |
+----------+--------------------------+------------+------------+----------+-------------+----------------+--------------+------------+------------+
10 rows in set (0.10 sec)
```

4. Select events with name partial match with 'cup'

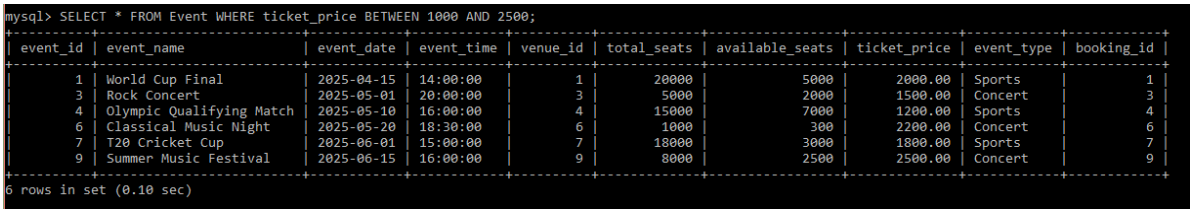
```
SELECT * FROM Event WHERE event_name LIKE '%cup%';
```



```
mysql> SELECT * FROM Event WHERE event_name LIKE '%cup%';
+----------+----------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
| event_id | event_name     | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type | booking_id |
+----------+----------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
|        1 | World Cup Final | 2025-04-15 | 14:00:00   |        1 |       20000 |           5000 |      2000.00 | Sports     |          1 |
|        7 | T20 Cricket Cup | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |          7 |
+----------+----------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
2 rows in set (0.03 sec)
```

5. Select events with ticket price range between 1000 to 2500

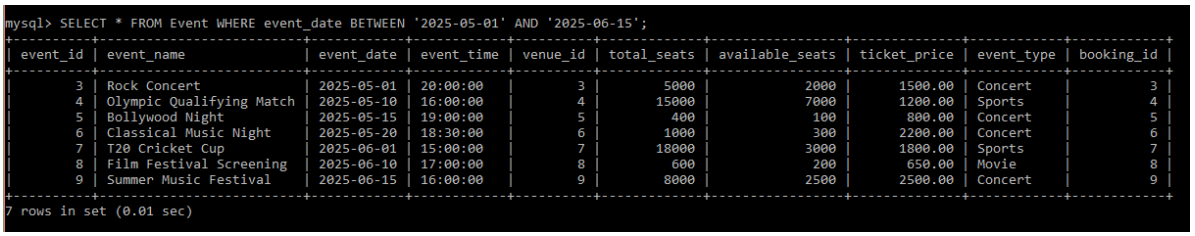
```
SELECT * FROM Event WHERE ticket_price BETWEEN 1000 AND 2500;
```



```
mysql> SELECT * FROM Event WHERE ticket_price BETWEEN 1000 AND 2500;
+----------+-------------------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
| event_id | event_name              | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type | booking_id |
+----------+-------------------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
|        1 | World Cup Final         | 2025-04-15 | 14:00:00   |        1 |       20000 |           5000 |      2000.00 | Sports     |          1 |
|        3 | Rock Concert            | 2025-05-01 | 20:00:00   |        3 |        5000 |           2000 |      1500.00 | Concert    |          3 |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |           7000 |      1200.00 | Sports     |          4 |
|        6 | Classical Music Night   | 2025-05-20 | 18:30:00   |        6 |        1000 |            300 |      2200.00 | Concert    |          6 |
|        7 | T20 Cricket Cup         | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |          7 |
|        9 | Summer Music Festival   | 2025-06-15 | 16:00:00   |        9 |        8000 |           2500 |      2500.00 | Concert    |          9 |
+----------+-------------------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
6 rows in set (0.10 sec)
```

6. Retrieve events with dates falling within a specific range

```
SELECT * FROM Event WHERE event_date BETWEEN '2025-05-01' AND '2
025-06-15';
```



```
mysql> SELECT * FROM Event WHERE event_date BETWEEN '2025-05-01' AND '2025-06-15';
+----------+-------------------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
| event_id | event_name              | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type | booking_id |
+----------+-------------------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
|        3 | Rock Concert            | 2025-05-01 | 20:00:00   |        3 |        5000 |           2000 |      1500.00 | Concert    |          3 |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |           7000 |      1200.00 | Sports     |          4 |
|        5 | Bollywood Night         | 2025-05-15 | 19:00:00   |        5 |         400 |            100 |       800.00 | Concert    |          5 |
|        6 | Classical Music Night   | 2025-05-20 | 18:30:00   |        6 |        1000 |            300 |      2200.00 | Concert    |          6 |
|        7 | T20 Cricket Cup         | 2025-06-01 | 15:00:00   |        7 |       18000 |           3000 |      1800.00 | Sports     |          7 |
|        8 | Film Festival Screening | 2025-06-10 | 17:00:00   |        8 |         600 |            200 |       650.00 | Movie      |          8 |
|        9 | Summer Music Festival   | 2025-06-15 | 16:00:00   |        9 |        8000 |           2500 |      2500.00 | Concert    |          9 |
+----------+-------------------------+------------+------------+----------+-------------+----------------+-------------+------------+------------+
7 rows in set (0.01 sec)
```

7. Retrieve events with available tickets that also have "Concert" in their name

```
SELECT * FROM Event WHERE available_seats > 0 AND event_name LIKE
'%Concert%';
```

```
mysql> SELECT * FROM Event WHERE available_seats > 0 AND event_name LIKE '%Concert%';
+----------+-------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
| event_id | event_name  | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type | booking_id |
+----------+-------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
|        3 | Rock Concert | 2025-05-01 | 20:00:00   |        3 |        5000 |            2000 |      1500.00 | Concert    |          3 |
+----------+-------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
1 row in set (0.00 sec)
```

8. Retrieve users in batches of 5, starting from the 6th user

```
SELECT * FROM Customer LIMIT 5 OFFSET 5;
```

```
mysql> SELECT * FROM Customer LIMIT 5 OFFSET 5;
+-------------+---------------+-------------------------+--------------+------------+
| customer_id | customer_name | email                   | phone_number | booking_id |
+-------------+---------------+-------------------------+--------------+------------+
|           6 | Neha Verma    | neha.verma@outlook.com  | 4321098765   |          6 |
|           7 | Ravi Kapoor   | ravi.kapoor@gmail.com   | 3210987654   |          7 |
|           8 | Anjali Desai  | anjali.desai@yahoo.com  | 2109876543   |          8 |
|           9 | Sanjay Joshi  | sanjay.joshi@gmail.com  | 1098765432   |          9 |
|          10 | Meera Reddy   | meera.reddy@hotmail.com | 9876543000   |         10 |
+-------------+---------------+-------------------------+--------------+------------+
5 rows in set (0.00 sec)
```

9. Retrieve bookings details with more than 4 tickets

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

```
mysql> SELECT * FROM Booking WHERE num_tickets > 4;
+------------+-------------+----------+-------------+------------+---------------------+
| booking_id | customer_id | event_id | num_tickets | total_cost | booking_date        |
+------------+-------------+----------+-------------+------------+---------------------+
|          3 |           3 |        3 |           5 |    7500.00 | 2025-03-17 09:15:00 |
|          7 |           7 |        7 |           6 |   10800.00 | 2025-03-21 12:15:00 |
|         10 |          10 |       10 |           5 |    1750.00 | 2025-03-24 11:00:00 |
+------------+-------------+----------+-------------+------------+---------------------+
3 rows in set (0.00 sec)
```

10. Retrieve customer information whose phone number ends with '000'

```
SELECT * FROM Customer WHERE phone_number LIKE '%000';
```

```
mysql> SELECT * FROM Customer WHERE phone_number LIKE '%000';
+-------------+---------------+-------------------------+--------------+------------+
| customer_id | customer_name | email                   | phone_number | booking_id |
+-------------+---------------+-------------------------+--------------+------------+
|          10 | Meera Reddy   | meera.reddy@hotmail.com | 9876543000   |         10 |
+-------------+---------------+-------------------------+--------------+------------+
1 row in set (0.00 sec)
```

11. Retrieve events in order whose seat capacity is more than 15000

```
SELECT * FROM Event WHERE total_seats > 15000 ORDER BY total_seats D
ESC;
```

```
mysql> SELECT * FROM Event WHERE total_seats > 15000 ORDER BY total_seats DESC;
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
| event_id | event_name      | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type | booking_id |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
|        1 | World Cup Final | 2025-04-15 | 14:00:00   |        1 |       20000 |            5000 |      2000.00 | Sports     |          1 |
|        7 | T20 Cricket Cup | 2025-06-01 | 15:00:00   |        7 |       18000 |            3000 |      1800.00 | Sports     |          7 |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
2 rows in set (0.00 sec)
```

12. Select events with names not starting with 'x', 'y', 'z'

```
SELECT * FROM Event WHERE event_name NOT LIKE 'x%' AND event_nam
e NOT LIKE 'y%' AND event_name NOT LIKE 'z%';
```

```
mysql> SELECT * FROM Event WHERE event_name NOT LIKE 'x%' AND event_name NOT LIKE 'y%' AND event_name NOT LIKE 'z%';
+----------+--------------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
| event_id | event_name               | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type | booking_id |
+----------+--------------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
|        1 | World Cup Final          | 2025-04-15 | 14:00:00   |        1 |       20000 |            5000 |      2000.00 | Sports     |          1 |
|        2 | Avengers 5 Premiere      | 2025-04-20 | 18:30:00   |        2 |         300 |              50 |       500.00 | Movie      |          2 |
|        3 | Rock Concert             | 2025-05-01 | 20:00:00   |        3 |        5000 |            2000 |      1500.00 | Concert    |          3 |
|        4 | Olympic Qualifying Match | 2025-05-10 | 16:00:00   |        4 |       15000 |            7000 |      1200.00 | Sports     |          4 |
|        5 | Bollywood Night          | 2025-05-15 | 19:00:00   |        5 |         400 |             100 |       800.00 | Concert    |          5 |
|        6 | Classical Music Night    | 2025-05-20 | 18:30:00   |        6 |        1000 |             300 |      2200.00 | Concert    |          6 |
|        7 | T20 Cricket Cup          | 2025-06-01 | 15:00:00   |        7 |       18000 |            3000 |      1800.00 | Sports     |          7 |
|        8 | Film Festival Screening  | 2025-06-10 | 17:00:00   |        8 |         600 |             200 |       650.00 | Movie      |          8 |
|        9 | Summer Music Festival    | 2025-06-15 | 16:00:00   |        9 |        8000 |            2500 |      2500.00 | Concert    |          9 |
|       10 | Community Theater Play   | 2025-06-20 | 19:30:00   |       10 |         200 |              50 |       350.00 | Movie      |         10 |
+----------+--------------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+
10 rows in set (0.12 sec)
```

Tasks 3: Aggregate functions, Having, Order By, Group By and Joins:

1. List Events and Their Average Ticket Prices.

```
SELECT event_name, AVG(ticket_price) AS average_ticket_price
FROM Event
GROUP BY event_name;
```

```
mysql> SELECT event_name, AVG(ticket_price) AS average_ticket_price
    -> FROM Event
    -> GROUP BY event_name;
+--------------------------+----------------------+
| event_name               | average_ticket_price |
+--------------------------+----------------------+
| World Cup Final          |          2000.000000 |
| Avengers 5 Premiere      |           500.000000 |
| Rock Concert             |          1500.000000 |
| Olympic Qualifying Match |          1200.000000 |
| Bollywood Night          |           800.000000 |
| Classical Music Night    |          2200.000000 |
| T20 Cricket Cup          |          1800.000000 |
| Film Festival Screening  |           650.000000 |
| Summer Music Festival    |          2500.000000 |
| Community Theater Play   |           350.000000 |
+--------------------------+----------------------+
10 rows in set (0.23 sec)
```

2. Calculate the Total Revenue Generated by Events

```
SELECT SUM(total_cost) AS total_revenue
FROM Booking;
```

```
mysql> SELECT SUM(total_cost) AS total_revenue
    -> FROM Booking;
+---------------+
| total_revenue |
+---------------+
|      51350.00 |
+---------------+
1 row in set (0.02 sec)
```

3. Find the Event with the Highest Ticket Sales

```
SELECT e.event_name, SUM(b.num_tickets) AS tickets_sold
FROM Event e
JOIN Booking b ON e.event_id = b.event_id
GROUP BY e.event_id, e.event_name
ORDER BY tickets_sold DESC
LIMIT 1;
```

```
mysql> SELECT e.event_name, SUM(b.num_tickets) AS tickets_sold
    -> FROM Event e
    -> JOIN Booking b ON e.event_id = b.event_id
    -> GROUP BY e.event_id, e.event_name
    -> ORDER BY tickets_sold DESC
    -> LIMIT 1;
+-----------------+--------------+
| event_name      | tickets_sold |
+-----------------+--------------+
| T20 Cricket Cup |            6 |
+-----------------+--------------+
1 row in set (0.03 sec)
```

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

```
SELECT e.event_name, SUM(b.num_tickets) AS total_tickets_sold FROM Ev
ent e
JOIN Booking b ON e.event_id = b.event_id
GROUP BY e.event_id, e.event_name;
```

```
mysql> SELECT e.event_name, 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_id, e.event_name;
+--------------------------+--------------------+
| event_name               | total_tickets_sold |
+--------------------------+--------------------+
| World Cup Final          |                  3 |
| Avengers 5 Premiere      |                  2 |
| Rock Concert             |                  5 |
| Olympic Qualifying Match |                  4 |
| Bollywood Night          |                  2 |
| Classical Music Night    |                  3 |
| T20 Cricket Cup          |                  6 |
| Film Festival Screening  |                  2 |
| Summer Music Festival    |                  4 |
| Community Theater Play   |                  5 |
+--------------------------+--------------------+
10 rows in set (0.00 sec)
```

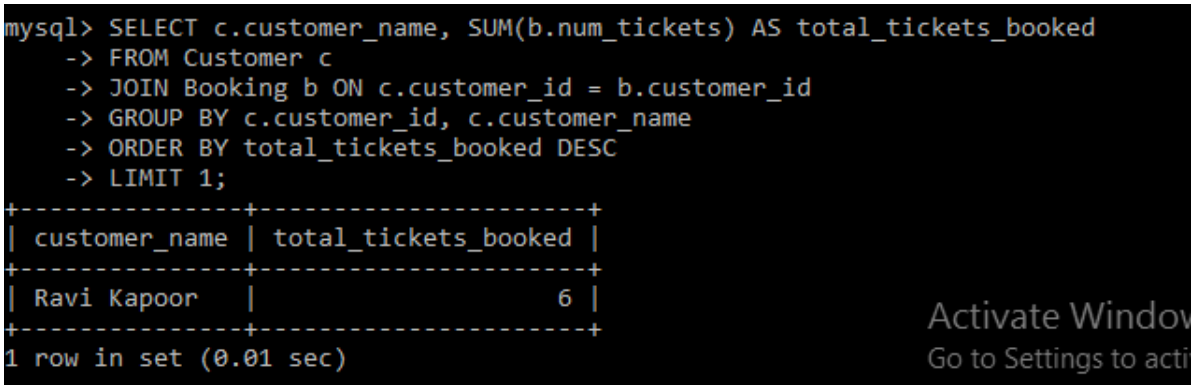
5. Find Events with No Ticket Sales

```
SELECT e.event_name, e.event_date, e.event_time FROM Event e
LEFT JOIN Booking b ON e.event_id = b.event_id
WHERE b.booking_id IS NULL;
```

```
mysql> SELECT e.event_name, e.event_date, e.event_time
    -> FROM Event e
    -> LEFT JOIN Booking b ON e.event_id = b.event_id
    -> WHERE b.booking_id IS NULL;
Empty set (0.03 sec)
```

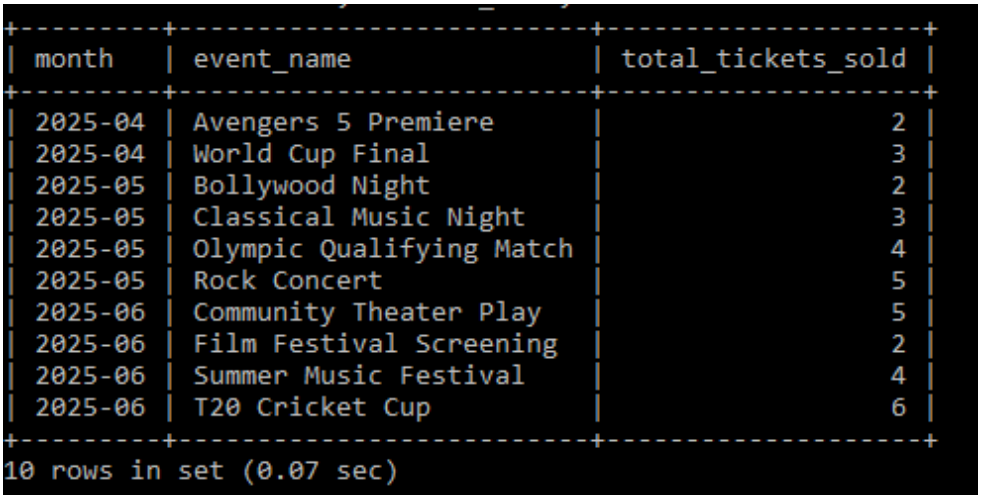
6. Find the User Who Has Booked the Most Tickets

```
SELECT c.customer_name, SUM(b.num_tickets) AS total_tickets_booked
FROM Customer c JOIN Booking b ON c.customer_id = b.customer_id
GROUP BY c.customer_id, c.customer_name
ORDER BY total_tickets_booked DESC LIMIT 1;
```



7. List Events and the Total Number of Tickets Sold for Each Month

```
SELECT DATE_FORMAT(e.event_date, '%Y-%m') AS month,e.event_name,
SUM(b.num_tickets) AS total_tickets_sold FROM Event e
JOIN Booking b ON e.event_id = b.event_id GROUP BY month, e.event_id,
e.event_name
ORDER BY month, e.event_name;
```

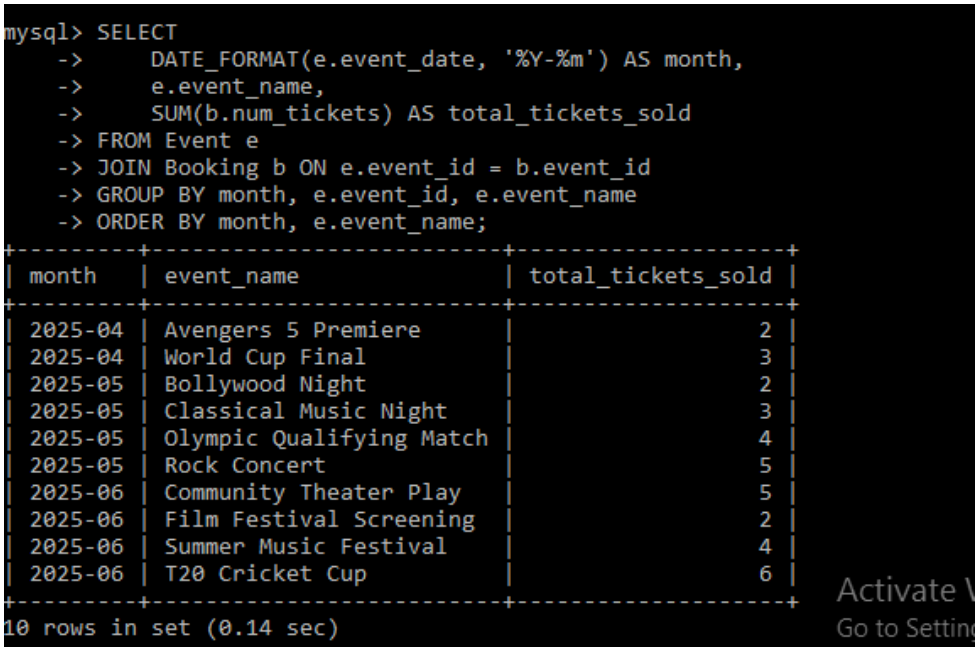


month	event_name	total_tickets_sold
2025-04	Avengers 5 Premiere	2
2025-04	World Cup Final	3
2025-05	Bollywood Night	2
2025-05	Classical Music Night	3
2025-05	Olympic Qualifying Match	4
2025-05	Rock Concert	5
2025-06	Community Theater Play	5
2025-06	Film Festival Screening	2
2025-06	Summer Music Festival	4
2025-06	T20 Cricket Cup	6

10 rows in set (0.07 sec)

8. Calculate the Average Ticket Price for Events in Each Venue

SELECT v.venue_name, AVG(e.ticket_price) AS average_ticket_price
FROM Venue v JOIN Event e ON v.venue_id = e.venue_id GROUP BY v.ven
ue_id, v.venue_name;



```
mysql> SELECT
    ->     DATE_FORMAT(e.event_date, '%Y-%m') AS month,
    ->     e.event_name,
    ->     SUM(b.num_tickets) AS total_tickets_sold
    -> FROM Event e
    -> JOIN Booking b ON e.event_id = b.event_id
    -> GROUP BY month, e.event_id, e.event_name
    -> ORDER BY month, e.event_name;
```

month	event_name	total_tickets_sold
2025-04	Avengers 5 Premiere	2
2025-04	World Cup Final	3
2025-05	Bollywood Night	2
2025-05	Classical Music Night	3
2025-05	Olympic Qualifying Match	4
2025-05	Rock Concert	5
2025-06	Community Theater Play	5
2025-06	Film Festival Screening	2
2025-06	Summer Music Festival	4
2025-06	T20 Cricket Cup	6

10 rows in set (0.14 sec)

9. Calculate the Total Number of Tickets Sold for Each Event Type

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.eve

```
nt_type;
```

```
mysql> 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 |
+------------+--------------------+
| Sports     |                 13 |
| Movie      |                  9 |
| Concert    |                 14 |
+------------+--------------------+
3 rows in set (0.00 sec)
```


10. Calculate the Total Revenue Generated by Events in Each Year

```
SELECT YEAR(e.event_date) AS event_year, SUM(b.total_cost) AS total_rev
enue
FROM Event e JOIN Booking b ON e.event_id = b.event_id
GROUP BY event_year ORDER BY event_year;
```

```
mysql> SELECT
    ->     YEAR(e.event_date) AS event_year,
    ->     SUM(b.total_cost) AS total_revenue
    -> FROM Event e
    -> JOIN Booking b ON e.event_id = b.event_id
    -> GROUP BY event_year
    -> ORDER BY event_year;
+------------+---------------+
| event_year | total_revenue |
+------------+---------------+
|       2025 |      51350.00 |
+------------+---------------+
1 row in set (0.10 sec)
```

11. List Users Who Have Booked Tickets for Multiple Events

```
SELECT c.customer_id,c.customer_name,
COUNT(DISTINCT b.event_id) AS number_of_events FROM Customer c JOI
N Booking b ON c.customer_id = b.customer_id GROUP BY c.customer_id,
c.customer_name HAVING number_of_events > 1;
```

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.customer_name,
    ->     COUNT(DISTINCT b.event_id) AS number_of_events
    -> FROM Customer c
    -> JOIN Booking b ON c.customer_id = b.customer_id
    -> GROUP BY c.customer_id, c.customer_name
    -> HAVING number_of_events > 1;
Empty set (0.06 sec)
```

12. Calculate the Total Revenue Generated by Events for Each User

SELECT c.customer_name,SUM(b.total_cost) AS total_spent FROM Custom
er c
JOIN Booking b ON c.customer_id = b.customer_id
GROUP BY c.customer_id, c.customer_name ORDER BY total_spent DESC;

```
mysql> SELECT
    ->     c.customer_name,
    ->     SUM(b.total_cost) AS total_spent
    -> FROM Customer c
    -> JOIN Booking b ON c.customer_id = b.customer_id
    -> GROUP BY c.customer_id, c.customer_name
    -> ORDER BY total_spent DESC;
+---------------+-------------+
| customer_name | total_spent |
+---------------+-------------+
| Ravi Kapoor   |    10800.00 |
| Sanjay Joshi  |    10000.00 |
| Amit Kumar    |     7500.00 |
| Neha Verma    |     6600.00 |
| Rahul Sharma  |     6000.00 |
| Sneha Gupta   |     4800.00 |
| Meera Reddy   |     1750.00 |
| Vikram Singh  |     1600.00 |
| Anjali Desai  |     1300.00 |
| Priya Patel   |     1000.00 |
+---------------+-------------+
10 rows in set (0.00 sec)
```

13. Calculate the Average Ticket Price for Events in Each Category and Venue

SELECT v.venue_name,e.event_type,AVG(e.ticket_price) AS average_ticket
_price
FROM Venue v JOIN Event e ON v.venue_id = e.venue_id
GROUP BY v.venue_id, v.venue_name, e.event_type
ORDER BY v.venue_name, e.event_type;

```
mysql> SELECT
    ->     v.venue_name,
    ->     e.event_type,
    ->     AVG(e.ticket_price) AS average_ticket_price
    -> FROM Venue v
    -> JOIN Event e ON v.venue_id = e.venue_id
    -> GROUP BY v.venue_id, v.venue_name, e.event_type
    -> ORDER BY v.venue_name, e.event_type;
+-----------------+------------+---------------------+
| venue_name      | event_type | average_ticket_price |
+-----------------+------------+---------------------+
| Cityview Theater | Concert    |           800.000000 |
| Community Center | Movie      |           350.000000 |
| Concert Dome     | Concert    |          2200.000000 |
| Festival Grounds | Concert    |          2500.000000 |
| Galaxy Cinemas   | Movie      |           500.000000 |
| Grand Auditorium | Movie      |           650.000000 |
| Music Hall       | Concert    |          1500.000000 |
| National Stadium | Sports     |          2000.000000 |
| Olympic Arena    | Sports     |          1200.000000 |
| Sports Complex   | Sports     |          1800.000000 |
+-----------------+------------+---------------------+
10 rows in set (0.02 sec)
```

14. List Users and the Total Number of Tickets They've Purchased in the Last 30 Days

```
SELECT c.customer_name,
SUM(b.num_tickets) AS tickets_purchased FROM Customer c
JOIN Booking b ON c.customer_id = b.customer_id
WHERE b.booking_date >= DATE_SUB(CURRENT_DATE(), INTERVAL 30 DA
Y)
GROUP BY c.customer_id, c.customer_name
ORDER BY tickets_purchased DESC;
```

```
mysql> SELECT
    ->     c.customer_name,
    ->     SUM(b.num_tickets) AS tickets_purchased
    -> FROM Customer c
    -> JOIN Booking b ON c.customer_id = b.customer_id
    -> WHERE b.booking_date >= DATE_SUB(CURRENT_DATE(), INTERVAL 30 DAY)
    -> GROUP BY c.customer_id, c.customer_name
    -> ORDER BY tickets_purchased DESC;
+---------------+-------------------+
| customer_name | tickets_purchased |
+---------------+-------------------+
| Ravi Kapoor   |                 6 |
| Amit Kumar    |                 5 |
| Meera Reddy   |                 5 |
| Sneha Gupta   |                 4 |
| Sanjay Joshi  |                 4 |
| Rahul Sharma  |                 3 |
| Neha Verma    |                 3 |
| Priya Patel   |                 2 |
| Vikram Singh  |                 2 |
| Anjali Desai  |                 2 |
+---------------+-------------------+
10 rows in set (0.12 sec)
```


Tasks 4: Subquery and its types

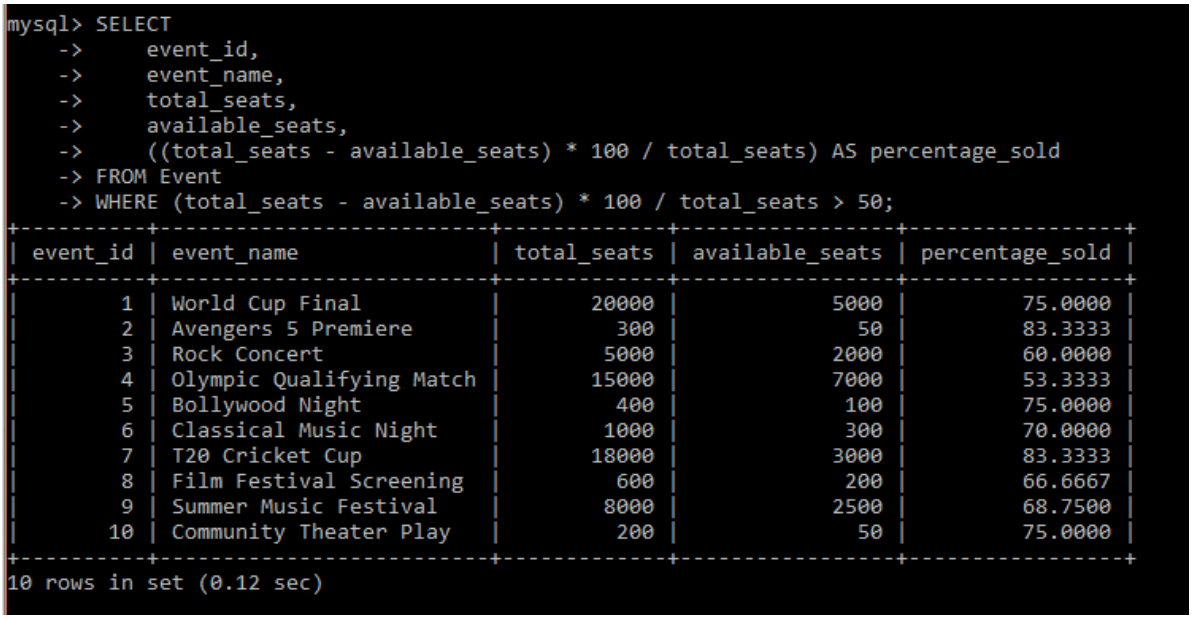
1. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery

SELECT v.venue_id,v.venue_name,(SELECT AVG(e.ticket_price FROM Even t e
WHERE e.venue_id = v.venue_id) AS average_ticket_price FROM Venue v;

```
mysql> SELECT
    ->     v.venue_id,
    ->     v.venue_name,
    ->     (SELECT AVG(e.ticket_price)
    ->      FROM Event e
    ->      WHERE e.venue_id = v.venue_id) AS average_ticket_price
    -> FROM Venue v;
+----------+------------------+----------------------+
| venue_id | venue_name       | average_ticket_price |
+----------+------------------+----------------------+
|        1 | National Stadium |          2000.000000 |
|        2 | Galaxy Cinemas   |           500.000000 |
|        3 | Music Hall       |          1500.000000 |
|        4 | Olympic Arena    |          1200.000000 |
|        5 | Cityview Theater |           800.000000 |
|        6 | Concert Dome     |          2200.000000 |
|        7 | Sports Complex   |          1800.000000 |
|        8 | Grand Auditorium |           650.000000 |
|        9 | Festival Grounds |          2500.000000 |
|       10 | Community Center |           350.000000 |
+----------+------------------+----------------------+
10 rows in set (0.03 sec)
```

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

```
SELECT event_id,event_name,total_seats,available_seats,
((total_seats - available_seats) * 100 / total_seats) AS percentage_sold FRO
M Event
WHERE (total_seats - available_seats) * 100 / total_seats > 50;
```



```
mysql> SELECT
    ->     event_id,
    ->     event_name,
    ->     total_seats,
    ->     available_seats,
    ->     ((total_seats - available_seats) * 100 / total_seats) AS percentage_sold
    -> FROM Event
    -> WHERE (total_seats - available_seats) * 100 / total_seats > 50;
+----------+--------------------------+-------------+-----------------+-----------------+
| event_id | event_name               | total_seats | available_seats | percentage_sold |
+----------+--------------------------+-------------+-----------------+-----------------+
|        1 | World Cup Final          |       20000 |            5000 |         75.0000 |
|        2 | Avengers 5 Premiere      |         300 |              50 |         83.3333 |
|        3 | Rock Concert             |        5000 |            2000 |         60.0000 |
|        4 | Olympic Qualifying Match |       15000 |            7000 |         53.3333 |
|        5 | Bollywood Night          |         400 |             100 |         75.0000 |
|        6 | Classical Music Night    |        1000 |             300 |         70.0000 |
|        7 | T20 Cricket Cup          |       18000 |            3000 |         83.3333 |
|        8 | Film Festival Screening  |         600 |             200 |         66.6667 |
|        9 | Summer Music Festival    |        8000 |            2500 |         68.7500 |
|       10 | Community Theater Play   |         200 |              50 |         75.0000 |
+----------+--------------------------+-------------+-----------------+-----------------+
10 rows in set (0.12 sec)
```

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

```
SELECT e.event_id,e.event_name,(SELECT SUM(b.num_tickets)
FROM Booking b WHERE b.event_id = e.event_id) AS tickets_sold FROM Ev
ent e;
```

```
mysql> SELECT
    ->     e.event_id,
    ->     e.event_name,
    ->     (SELECT SUM(b.num_tickets)
    ->      FROM Booking b
    ->      WHERE b.event_id = e.event_id) AS tickets_sold
    -> FROM Event e;
+----------+--------------------------+--------------+
| event_id | event_name               | tickets_sold |
+----------+--------------------------+--------------+
|        1 | World Cup Final          |            3 |
|        2 | Avengers 5 Premiere      |            2 |
|        3 | Rock Concert             |            5 |
|        4 | Olympic Qualifying Match |            4 |
|        5 | Bollywood Night          |            2 |
|        6 | Classical Music Night    |            3 |
|        7 | T20 Cricket Cup          |            6 |
|        8 | Film Festival Screening  |            2 |
|        9 | Summer Music Festival    |            4 |
|       10 | Community Theater Play   |            5 |
+----------+--------------------------+--------------+
10 rows in set (0.00 sec)
```

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

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

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.customer_name,
    ->     c.email
    -> FROM Customer c
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Booking b
    ->     WHERE b.customer_id = c.customer_id
    -> );
Empty set (0.02 sec)
```

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

```
SELECT e.event_id,e.event_name,e.event_date FROM Event e
WHERE e.event_id NOT IN (SELECT DISTINCT event_id FROM Booking);
```

```
mysql> SELECT
    ->     e.event_id,
    ->     e.event_name,
    ->     e.event_date
    -> FROM Event e
    -> WHERE e.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

```
SELECT event_type,SUM(tickets_sold) AS total_tickets_sold
FROM (SELECT e.event_type,e.event_id,(SELECT SUM(b.num_tickets)
FROM Booking b WHERE b.event_id = e.event_id) AS tickets_sold
FROM Event e) AS event_sales GROUP BY event_type;
```

```
mysql> SELECT
    ->     event_type,
    ->     SUM(tickets_sold) AS total_tickets_sold
    -> FROM (
    ->     SELECT
    ->         e.event_type,
    ->         e.event_id,
    ->         (SELECT SUM(b.num_tickets)
    ->          FROM Booking b
    ->          WHERE b.event_id = e.event_id) AS tickets_sold
    ->     FROM Event e
    -> ) AS event_sales
    -> GROUP BY event_type;
+------------+--------------------+
| event_type | total_tickets_sold |
+------------+--------------------+
| Sports     |                 13 |
| Movie      |                  9 |
| Concert    |                 14 |
+------------+--------------------+
3 rows in set (0.04 sec)
```

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

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

```
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 | World Cup Final      |      2000.00 |
|        3 | Rock Concert         |      1500.00 |
|        6 | Classical Music Night |      2200.00 |
|        7 | T20 Cricket Cup      |      1800.00 |
|        9 | Summer Music Festival |      2500.00 |
+----------+----------------------+--------------+
5 rows in set (0.01 sec)
```

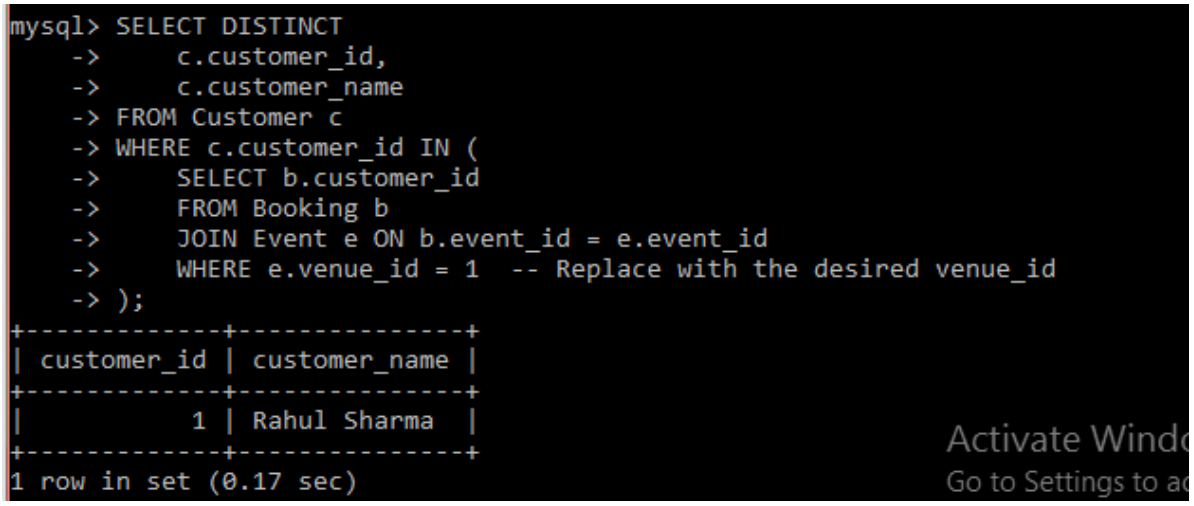
8. Calculate the Total Revenue Generated by Events for Each User Using a Correlated Subquery

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

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.customer_name,
    ->     (SELECT SUM(b.total_cost)
    ->      FROM Booking b
    ->      WHERE b.customer_id = c.customer_id) AS total_spent
    -> FROM Customer c;
+-------------+---------------+-------------+
| customer_id | customer_name | total_spent |
+-------------+---------------+-------------+
|           1 | Rahul Sharma  |     6000.00 |
|           2 | Priya Patel   |     1000.00 |
|           3 | Amit Kumar    |     7500.00 |
|           4 | Sneha Gupta   |     4800.00 |
|           5 | Vikram Singh  |     1600.00 |
|           6 | Neha Verma    |     6600.00 |
|           7 | Ravi Kapoor   |    10800.00 |
|           8 | Anjali Desai  |     1300.00 |
|           9 | Sanjay Joshi  |    10000.00 |
|          10 | Meera Reddy   |     1750.00 |
+-------------+---------------+-------------+
10 rows in set (0.02 sec)
```

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

```
SELECT DISTINCT c.customer_id,c.customer_name
FROM Customer c WHERE c.customer_id IN (
   SELECT b.customer_id
   FROM Booking b
   JOIN Event e ON b.event_id = e.event_id
   WHERE e.venue_id = 1  -- Replace with the desired venue_id);
```



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

```
SELECT e.event_type,(SELECT SUM(b.num_tickets)
    FROM Booking b
    JOIN Event e2 ON b.event_id = e2.event_id
    WHERE e2.event_type = e.event_type) AS total_tickets_sold
FROM Event e
GROUP BY e.event_type;
```

```
mysql> SELECT
    ->     e.event_type,
    ->     (SELECT SUM(b.num_tickets)
    ->      FROM Booking b
    ->      JOIN Event e2 ON b.event_id = e2.event_id
    ->      WHERE e2.event_type = e.event_type) AS total_tickets_sold
    -> FROM Event e
    -> GROUP BY e.event_type;
+------------+--------------------+
| event_type | total_tickets_sold |
+------------+--------------------+
| Sports     |                 13 |
| Movie      |                  9 |
| Concert    |                 14 |
+------------+--------------------+
3 rows in set (0.00 sec)
```

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

```
SELECT c.customer_id,c.customer_name,
   (SELECT DATE_FORMAT(MIN(e.event_date), '%Y-%m')
    FROM Booking b
    JOIN Event e ON b.event_id = e.event_id
    WHERE b.customer_id = c.customer_id) AS first_booking_month
FROM Customer c
WHERE c.customer_id IN (
   SELECT DISTINCT b.customer_id
   FROM Booking b
   JOIN Event e ON b.event_id = e.event_id);
```

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.customer_name,
    ->     (SELECT DATE_FORMAT(MIN(e.event_date), '%Y-%m')
    ->      FROM Booking b
    ->      JOIN Event e ON b.event_id = e.event_id
    ->      WHERE b.customer_id = c.customer_id) AS first_booking_month
    -> FROM Customer c
    -> WHERE c.customer_id IN (
    ->     SELECT DISTINCT b.customer_id
    ->     FROM Booking b
    ->     JOIN Event e ON b.event_id = e.event_id
    -> );
+-------------+---------------+----------------------+
| customer_id | customer_name | first_booking_month  |
+-------------+---------------+----------------------+
|           1 | Rahul Sharma  | 2025-04              |
|           2 | Priya Patel   | 2025-04              |
|           3 | Amit Kumar    | 2025-05              |
|           4 | Sneha Gupta   | 2025-05              |
|           5 | Vikram Singh  | 2025-05              |
|           6 | Neha Verma    | 2025-05              |
|           7 | Ravi Kapoor   | 2025-06              |
|           8 | Anjali Desai  | 2025-06              |
|           9 | Sanjay Joshi  | 2025-06              |
|          10 | Meera Reddy   | 2025-06              |
+-------------+---------------+----------------------+
10 rows in set (0.02 sec)
```


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

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

```
mysql> SELECT
    ->     v.venue_id,
    ->     v.venue_name,
    ->     (SELECT AVG(ticket_price)
    ->      FROM Event
    ->      WHERE venue_id = v.venue_id) AS avg_ticket_price
    -> FROM Venue v;
+----------+------------------+------------------+
| venue_id | venue_name       | avg_ticket_price |
+----------+------------------+------------------+
|        1 | National Stadium |      2000.000000 |
|        2 | Galaxy Cinemas   |       500.000000 |
|        3 | Music Hall       |      1500.000000 |
|        4 | Olympic Arena    |      1200.000000 |
|        5 | Cityview Theater |       800.000000 |
|        6 | Concert Dome     |      2200.000000 |
|        7 | Sports Complex   |      1800.000000 |
|        8 | Grand Auditorium |       650.000000 |
|        9 | Festival Grounds |      2500.000000 |
|       10 | Community Center |       350.000000 |
+----------+------------------+------------------+
10 rows in set (0.00 sec)
```