Name:- Sneha Das

Assignment 5

Tasks 1: Database Design:

1. Create the database named "TicketBookingSystem"

```
mysql> create database TicketBookingSystem;
Query OK, 1 row affected (0.02 sec)
```

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

- Venue

```
mysql> create table Venue(venue_id int primary key
    -> ,venue_name varchar(30),
    -> address text);
Query OK, 0 rows affected (0.10 sec)

mysql> desc Venue;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| venue_id   | int         | NO   | PRI | NULL    |       |
| venue_name | varchar(30) | YES  |     | NULL    |       |
| address    | text        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

- Events

```
mysql> create table Events(event_id int primary key,
    -> event_name varchar(20),
    -> event_date date,
    -> event_time time,
    -> venue_id int,
    -> total_seats int, available_seats int,
    -> ticket_price decimal,
    -> event_type varchar(20),
    -> foreign key(venue_id) references Venue(venue_id));
Query OK, 0 rows affected (0.08 sec)

mysql> desc events;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| event_id        | int           | NO   | PRI | NULL    |       |
| event_name      | varchar(20)   | YES  |     | NULL    |       |
| event_date      | date          | YES  |     | NULL    |       |
| event_time      | time          | YES  |     | NULL    |       |
| venue_id        | int           | YES  | MUL | NULL    |       |
| total_seats     | int           | YES  |     | NULL    |       |
| available_seats | int           | YES  |     | NULL    |       |
| ticket_price    | decimal(10,0) | YES  |     | NULL    |       |
| event_type      | varchar(20)   | YES  |     | NULL    |       |
+-----------------+---------------+------+-----+---------+-------+
9 rows in set (0.01 sec)
```

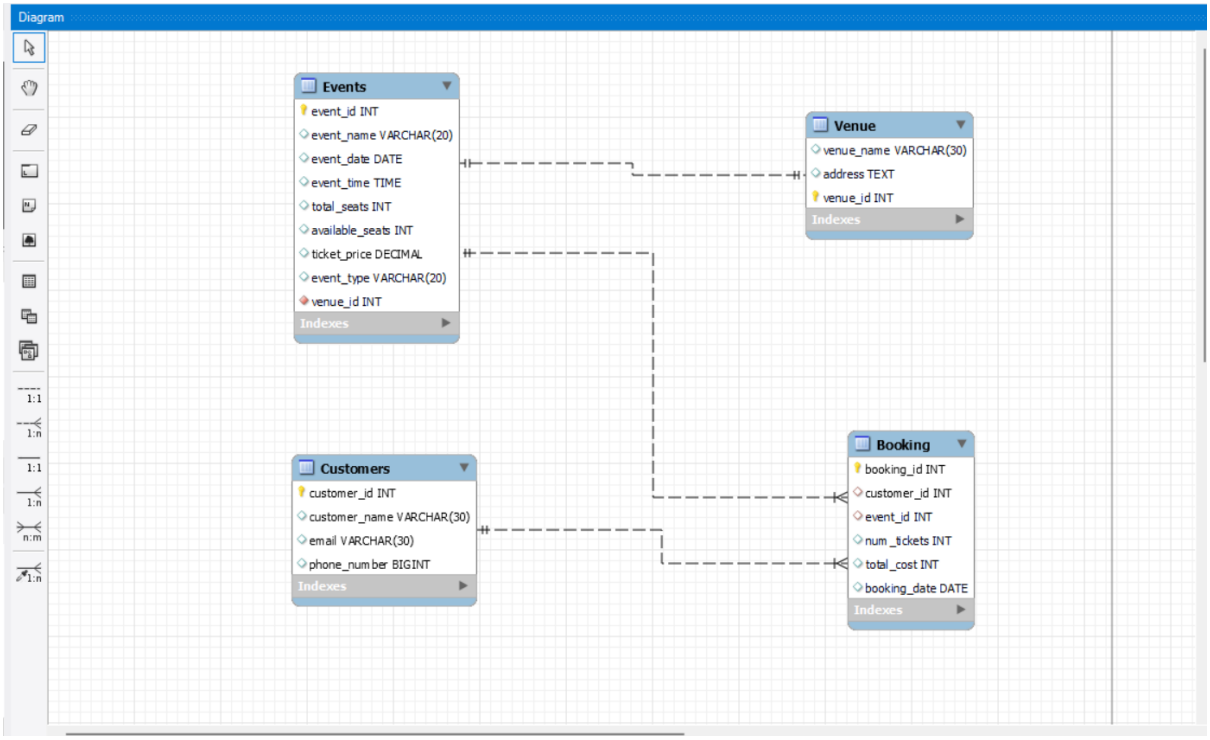
- Customer

```
mysql> create table Customer(
    -> customer_id int primary key,
    -> customer_name varchar(30),
    -> email varchar(30),
    -> phone_number long);
Query OK, 0 rows affected (0.06 sec)

mysql> desc customer;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| customer_id   | int         | NO   | PRI | NULL    |       |
| customer_name | varchar(30) | YES  |     | NULL    |       |
| email         | varchar(30) | YES  |     | NULL    |       |
| phone_number  | mediumtext  | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

- Bookings

```
mysql> create table Bookings(
    -> booking_id int primary key,
    -> customer_id int,
    -> event_id int,
    -> num_tickets int,
    -> total_cost int,
    -> booking_date date,
    -> foreign key (customer_id) references Customer(customer_id),
    -> foreign key (event_id) references Events(event_id));
Query OK, 0 rows affected (0.08 sec)

mysql> desc bookings;
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| booking_id   | int  | NO   | PRI | NULL    |       |
| customer_id  | int  | YES  | MUL | NULL    |       |
| event_id     | int  | YES  | MUL | NULL    |       |
| num_tickets  | int  | YES  |     | NULL    |       |
| total_cost   | int  | YES  |     | NULL    |       |
| booking_date | date | YES  |     | NULL    |       |
+--------------+------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

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



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

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

- Events

```
mysql> insert into events values
    -> (103,"Tech Summit",'2024-04-25','10:00:00',3,500,400,20.00,"Conference"),
    -> (104,"Comedy Show",'2024-05-10','20:30:00',4,200,180,15.00,"Comedy"),
    -> (105,"Classical Music Gala",'2024-06-05','18:00:00',5,400,350,18.00,"Concert"),
    -> (106,"IPL Final",'2024-06-15','19:30:00',2,800,700,30.00,"Sports"),
    -> (107,"IT Conference",'2024-07-12','09:00:00',3,600,550,25.00,"Conference"),
    -> (108,"Stand-up Comedy Night",'2024-08-20','21:00:00',4,250,230,12.00,"Comedy"),
    -> (109,"Sufi Music Festival",'2024-09-10','17:30:00',5,350,300,15.00,"Concert"),
    -> (110,"Football League",'2024-09-25','15:00:00',6,700,650,20.00,"Sports"),
    -> (111,"Movie Premiere",'2024-10-05','20:00:00',7,150,120,10.00,"Movie"),
    -> ( 112,"Action Blockbuster",'2024-10-15','18:30:00',8,180,150,11.00,"Movie"),
    -> (113,"Romantic Movie Night",'2024-11-01','19:30:00',9,200,180,12.00,"Movie");
Query OK, 11 rows affected (0.02 sec)
Records: 11  Duplicates: 0  Warnings: 0
```

- Venue

```
mysql> insert into venue values
    -> (1,"Taj Convention Center","22 MG Road,Mumbai,Maharashtra"),
    -> (2,"Royal Stadium", "8A Eden Gardens,Kolkata, West Bengal"),
    -> (3,"Lotus Hall","15 Residency Road, Bangalore, Karnataka"),
    -> (4,"Grand Auditorium","42 Connaught Place, Delhi, NCR"),
    -> (5,"Pearl Palace","5 Jubilee Hills, Hyderabad, Telangana"),
    -> (6,"Queen's Pavillion","18 Park Street, Kolkata, West Bengal"),
    -> (7,"Saffron Gardens","10 MG Road, Pune, Maharashtra"),
    -> (8,"Starry Arena","3 MG Road, Chennai, Tamil Nadu"),
    -> (9,"Velvet Lounge","25 Brigade Road, Bangalore, Karnataka"),
    -> (10,"Harmony Hall","12 Banjara Hills, Hyderabad, Telangana");
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

- Customers

```
mysql> insert into customer values
    -> (1001,"Priya Sharma","priya.sharma@email.com",9876543210),
    -> (1002,"Rahul Verma","rahul.verma@email.com",87654321091),
    -> (1003,"Pooja Singh","pooja.singh@email.com",7654321098),
    -> (1004,"Aman Gupta","aman.gupta@email.com",65432109879),
    -> (1005,"Nisha Patel","nisha.patel@email.com",5432109876),
    -> (1006,"Sameer Shah","sameer.shah@email.com",4321098765),
    -> (1007,"Anjali Desai","anjali.desai@email.com",3210987654),
    -> (1008,"Rohan Malhotra","rohan.malhotra@email.com",2109876543),
    -> (1009,"Shreya Kapoor","shreya.kapoor@email.com",1098765432),
    -> (1010,"Kartik Joshi","kartik.joshi@email.com",9876543211);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

- Booking

```
mysql> insert into bookings values
    -> (5001,1001,101,2,25.00,'2024-01-20'),
    -> (5002,1002,102,5,125.00,'2024-01-21'),
    -> (5003,1003,103,3,60.00,'2024-01-22'),
    -> (5004,1004,104,4,60.00,'2024-01-23'),
    -> (5005,1005,105,2,36.00,'2024-01-24'),
    -> (5006,1006,106,3,90.00,'2024-01-25'),
    -> (5007,1007,107,2,50.00,'2024-01-26'),
    -> (5008,1008,108,2,24.00,'2024-01-27'),
    -> (5009,1009,109,4,60.00,'2024-01-28'),
    -> (5010,1010,110,5,100.00,'2024-01-29'),
    -> (5011,1001,111,3,30.00,'2024-02-01'),
    -> (5014,1004,106,4,52.00,'2024-06-15'),
    -> (5015,1005,106,2,25.00,'2024-06-15');
Query OK, 13 rows affected (0.01 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

2. Write a SQL query to list all Events.

```
mysql> select * from events;
+----------+----------------------+------------+------------+----------+-------------+----------------+-------------+------------+
| event_id | event_name           | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+----------------------+------------+------------+----------+-------------+----------------+-------------+------------+
|      101 | Bollywood Night      | 2024-02-15 | 19:00:00   |        1 |         300 |            250 |          13 | Concert    |
|      102 | Cricket Championship | 2024-03-20 | 16:30:00   |        2 |        1000 |            800 |          25 | Sports     |
|      103 | Tech Summit          | 2024-04-25 | 10:00:00   |        3 |         500 |            400 |          20 | Conference |
|      104 | Comedy Show          | 2024-05-10 | 20:30:00   |        4 |         200 |            180 |          15 | Comedy     |
|      105 | Classical Music Gala | 2024-06-05 | 18:00:00   |        5 |         400 |            350 |          18 | Concert    |
|      106 | IPL Final            | 2024-06-15 | 19:30:00   |        2 |         800 |            700 |          30 | Sports     |
|      107 | IT Conference        | 2024-07-12 | 09:00:00   |        3 |         600 |            550 |          25 | Conference |
|      108 | Stand-up Comedy Night | 2024-08-20 | 21:00:00   |        4 |         250 |            230 |          12 | Comedy     |
|      109 | Sufi Music Festival  | 2024-09-10 | 17:30:00   |        5 |         350 |            300 |          15 | Concert    |
|      110 | Football League      | 2024-09-25 | 15:00:00   |        6 |         700 |            650 |          20 | Sports     |
|      111 | Movie Premiere       | 2024-10-05 | 20:00:00   |        7 |         150 |            120 |          10 | Movie      |
|      112 | Action Blockbuster   | 2024-10-15 | 18:30:00   |        8 |         180 |            150 |          11 | Movie      |
|      113 | Romantic Movie Night | 2024-11-01 | 19:30:00   |        9 |         200 |            180 |          12 | Movie      |
+----------+----------------------+------------+------------+----------+-------------+----------------+-------------+------------+
13 rows in set (0.00 sec)
```

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

```
mysql> select event_id,event_name,available_seats from events;
+----------+----------------------+-----------------+
| event_id | event_name           | available_seats |
+----------+----------------------+-----------------+
|      101 | Bollywood Night      |             250 |
|      102 | Cricket Championship |             800 |
|      103 | Tech Summit          |             400 |
|      104 | Comedy Show          |             180 |
|      105 | Classical Music Gala |             350 |
|      106 | IPL Final            |             700 |
|      107 | IT Conference        |             550 |
|      108 | Stand-up Comedy Night |             230 |
|      109 | Sufi Music Festival  |             300 |
|      110 | Football League      |             650 |
|      111 | Movie Premiere       |             120 |
|      112 | Action Blockbuster   |             150 |
|      113 | Romantic Movie Night |             180 |
+----------+----------------------+-----------------+
13 rows in set (0.00 sec)
```

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

```
mysql> select event_name from events where event_name like '%league';
+-----------------+
| event_name      |
+-----------------+
| Football League |
+-----------------+
1 row in set (0.00 sec)

mysql> insert into events values
    -> (114,"FIFA CUP",'2024-05-12','21:00:00',6,400,120,150.25,"Sports");
Query OK, 1 row affected, 1 warning (0.02 sec)

mysql> select event_name from events where event_name like '%cup%';
+------------+
| event_name |
+------------+
| FIFA CUP   |
+------------+
1 row in set (0.00 sec)
```

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

```
mysql> select * from events 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 |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
|      101 | Bollywood Night      | 2024-02-15 | 19:00:00   |        1 |         300 |             250 |         1300 | Concert    |
|      102 | Cricket Championship | 2024-03-20 | 16:30:00   |        2 |        1000 |             800 |         2500 | Sports     |
|      103 | Tech Summit          | 2024-04-25 | 10:00:00   |        3 |         500 |             400 |         2000 | Conference |
|      104 | Comedy Show          | 2024-05-10 | 20:30:00   |        4 |         200 |             180 |         1500 | Comedy     |
|      105 | Classical Music Gala | 2024-06-05 | 18:00:00   |        5 |         400 |             350 |         1800 | Concert    |
|      107 | IT Conference        | 2024-07-12 | 09:00:00   |        3 |         600 |             550 |         2500 | Conference |
|      108 | Stand-up Comedy Night| 2024-08-20 | 21:00:00   |        4 |         250 |             230 |         1200 | Comedy     |
|      109 | Sufi Music Festival  | 2024-09-10 | 17:30:00   |        5 |         350 |             300 |         1500 | Concert    |
|      110 | Football League      | 2024-09-25 | 15:00:00   |        6 |         700 |             650 |         2000 | Sports     |
|      111 | Movie Premiere       | 2024-10-05 | 20:00:00   |        7 |         150 |             120 |         1000 | Movie      |
|      112 | Action Blockbuster   | 2024-10-15 | 18:30:00   |        8 |         180 |             150 |         1100 | Movie      |
|      113 | Romantic Movie Night | 2024-11-01 | 19:30:00   |        9 |         200 |             180 |         1200 | Movie      |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
12 rows in set (0.00 sec)
```

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

```
mysql> select * from events where event_date not between '2024-02-12' and '2024-05-12';
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name           | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
|      105 | Classical Music Gala | 2024-06-05 | 18:00:00   |        5 |         400 |             350 |         1800 | Concert    |
|      106 | IPL Final            | 2024-06-15 | 19:30:00   |        2 |         800 |             700 |         3000 | Sports     |
|      107 | IT Conference        | 2024-07-12 | 09:00:00   |        3 |         600 |             550 |         2500 | Conference |
|      108 | Stand-up Comedy Night| 2024-08-20 | 21:00:00   |        4 |         250 |             230 |         1200 | Comedy     |
|      109 | Sufi Music Festival  | 2024-09-10 | 17:30:00   |        5 |         350 |             300 |         1500 | Concert    |
|      110 | Football League      | 2024-09-25 | 15:00:00   |        6 |         700 |             650 |         2000 | Sports     |
|      111 | Movie Premiere       | 2024-10-05 | 20:00:00   |        7 |         150 |             120 |         1000 | Movie      |
|      112 | Action Blockbuster   | 2024-10-15 | 18:30:00   |        8 |         180 |             150 |         1100 | Movie      |
|      113 | Romantic Movie Night | 2024-11-01 | 19:30:00   |        9 |         200 |             180 |         1200 | Movie      |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
9 rows in set (0.00 sec)
```

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

```
mysql> select event_id,event_name,available_seats from events
    -> where event_name like '%Concert%';
Empty set (0.00 sec)

mysql> select event_id,event_name,available_seats from events
    -> where event_type="Concert";
+----------+----------------------+-----------------+
| event_id | event_name           | available_seats |
+----------+----------------------+-----------------+
|      101 | Bollywood Night      |             250 |
|      105 | Classical Music Gala |             350 |
|      109 | Sufi Music Festival  |             300 |
+----------+----------------------+-----------------+
3 rows in set (0.00 sec)
```

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

```
mysql> create procedure printInBatches()
    -> begin
    -> declare start int default 5;
    -> declare size int default 5;
    -> declare total int;
    -> select count(customer_id) from customer into total;
    -> while start <= total do
    -> select * from customer order by customer_id
    -> limit 5 offset start;
    -> set start = start+size;
    -> end while;
    -> end @@
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
mysql> call printInBatches();
+-------------+---------------+-------------------------+--------------+
| customer_id | customer_name | email                   | phone_number |
+-------------+---------------+-------------------------+--------------+
|        1006 | Sameer Shah   | sameer.shah@email.com   | 4321098765   |
|        1007 | Anjali Desai  | anjali.desai@email.com  | 3210987654   |
|        1008 | Rohan Malhotra| rohan.malhotra@email.com| 2109876543   |
|        1009 | Shreya Kapoor | shreya.kapoor@email.com | 1098765432   |
|        1010 | Kartik Joshi  | kartik.joshi@email.com  | 9876543211   |
+-------------+---------------+-------------------------+--------------+
5 rows in set (0.00 sec)

Empty set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

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

```
mysql> select * from bookings where num_tickets>4;
+------------+-------------+----------+-------------+------------+--------------+
| booking_id | customer_id | event_id | num_tickets | total_cost | booking_date |
+------------+-------------+----------+-------------+------------+--------------+
|       5002 |        1002 |      102 |           5 |        125 | 2024-01-21   |
|       5010 |        1010 |      110 |           5 |        100 | 2024-01-29   |
+------------+-------------+----------+-------------+------------+--------------+
2 rows in set (0.00 sec)
```

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

```
mysql> select * from customer where right(phone_number,3)='000';
Empty set (0.00 sec)

mysql> select * from customer where right(phone_number,3)='879';
+-------------+---------------+----------------------+--------------+
| customer_id | customer_name | email                | phone_number |
+-------------+---------------+----------------------+--------------+
|        1004 | Aman Gupta    | aman.gupta@email.com | 65432109879  |
+-------------+---------------+----------------------+--------------+
1 row in set (0.00 sec)
```

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

```
mysql> select * from events where total_seats > 500;
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name           | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
|      102 | Cricket Championship | 2024-03-20 | 16:30:00   |        2 |        1000 |             800 |         2500 | Sports     |
|      106 | IPL Final            | 2024-06-15 | 19:30:00   |        2 |         800 |             700 |         3000 | Sports     |
|      107 | IT Conference        | 2024-07-12 | 09:00:00   |        3 |         600 |             550 |         2500 | Conference |
|      110 | Football League      | 2024-09-25 | 15:00:00   |        6 |         700 |             650 |         2000 | Sports     |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
4 rows in set (0.00 sec)

mysql> select * from events where total_seats > 15000;
Empty set (0.00 sec)
```

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

```
mysql> select * from events where left(event_name,1) not in ('x','y','z');
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name           | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
|      101 | Bollywood Night      | 2024-02-15 | 19:00:00   |        1 |         300 |             250 |         1300 | Concert    |
|      102 | Cricket Championship | 2024-03-20 | 16:30:00   |        2 |        1000 |             800 |         2500 | Sports     |
|      103 | Tech Summit          | 2024-04-25 | 10:00:00   |        3 |         500 |             400 |         2000 | Conference |
|      104 | Comedy Show          | 2024-05-10 | 20:30:00   |        4 |         200 |             180 |         1500 | Comedy     |
|      105 | Classical Music Gala | 2024-06-05 | 18:00:00   |        5 |         400 |             350 |         1800 | Concert    |
|      106 | IPL Final            | 2024-06-15 | 19:30:00   |        2 |         800 |             700 |         3000 | Sports     |
|      107 | IT Conference        | 2024-07-12 | 09:00:00   |        3 |         600 |             550 |         2500 | Conference |
|      108 | Stand-up Comedy Night | 2024-08-20 | 21:00:00   |        4 |         250 |             230 |         1200 | Comedy     |
|      109 | Sufi Music Festival  | 2024-09-10 | 17:30:00   |        5 |         350 |             300 |         1500 | Concert    |
|      110 | Football League      | 2024-09-25 | 15:00:00   |        6 |         700 |             650 |         2000 | Sports     |
|      111 | Movie Premiere       | 2024-10-05 | 20:00:00   |        7 |         150 |             120 |         1000 | Movie      |
|      112 | Action Blockbuster   | 2024-10-15 | 18:30:00   |        8 |         180 |             150 |         1100 | Movie      |
|      113 | Romantic Movie Night | 2024-11-01 | 19:30:00   |        9 |         200 |             180 |         1200 | Movie      |
|      114 | FIFA CUP             | 2024-05-12 | 21:00:00   |        6 |         400 |             120 |        15000 | Sports     |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
14 rows in set (0.01 sec)
```

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

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

```
mysql> select event_name,avg(ticket_price) from events group by event_type,event_name;
+----------------------+--------------------+
| event_name           | avg(ticket_price) |
+----------------------+--------------------+
| Bollywood Night      |          1300.0000 |
| Cricket Championship |          2500.0000 |
| Tech Summit          |          2000.0000 |
| Comedy Show          |          1500.0000 |
| Classical Music Gala |          1800.0000 |
| IPL Final            |          3000.0000 |
| IT Conference        |          2500.0000 |
| Stand-up Comedy Night |          1200.0000 |
| Sufi Music Festival  |          1500.0000 |
| Football League      |          2000.0000 |
| Movie Premiere       |          1000.0000 |
| Action Blockbuster   |          1100.0000 |
| Romantic Movie Night |          1200.0000 |
| FIFA CUP             |         15000.0000 |
+----------------------+--------------------+
14 rows in set (0.00 sec)
```

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

```
mysql> select sum(b.num_tickets*e.ticket_price) as TotalRevenue
    -> from events e
    -> join bookings b on e.event_id=b.event_id;
+--------------+
| TotalRevenue |
+--------------+
|        74100 |
+--------------+
1 row in set (0.00 sec)
```

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

```
mysql> select e.event_id,e.event_name,sum(b.num_tickets) as TotalTicketsSold
    -> from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by e.event_id,e.event_name
    -> order by TotalTicketsSold desc
    -> limit 1;
+----------+------------+------------------+
| event_id | event_name | TotalTicketsSold |
+----------+------------+------------------+
|      106 | IPL Final  |                9 |
+----------+------------+------------------+
1 row in set (0.00 sec)
```

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

```
mysql> select e.event_id,e.event_name,sum(b.num_tickets) as NumberOfTickets
    -> from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by e.event_id
    -> order by NumberOfTickets desc
    -> limit 1;
+----------+------------+-----------------+
| event_id | event_name | NumberOfTickets |
+----------+------------+-----------------+
|      106 | IPL Final  |               9 |
+----------+------------+-----------------+
1 row in set (0.00 sec)
```

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

```
mysql> select e.event_name,sum(b.num_tickets) as NumberOfTickets
    -> from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by e.event_id;
+----------------------+-----------------+
| event_name           | NumberOfTickets |
+----------------------+-----------------+
| Bollywood Night      |               2 |
| Cricket Championship |               5 |
| Tech Summit          |               3 |
| Comedy Show          |               4 |
| Classical Music Gala |               2 |
| IPL Final            |               9 |
| IT Conference        |               2 |
| Stand-up Comedy Night|               2 |
| Sufi Music Festival  |               4 |
| Football League      |               5 |
| Movie Premiere       |               3 |
+----------------------+-----------------+
11 rows in set (0.00 sec)
```

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

```
mysql> select c.customer_id,c.customer_name,sum(b.num_tickets) as NumberOfTickets
    -> from customer c
    -> join bookings b on c.customer_id=b.customer_id
    -> group by c.customer_id
    -> order by NumberOfTickets desc
    -> limit 1;
+-------------+--------------+----------------+
| customer_id | customer_name | NumberOfTickets |
+-------------+--------------+----------------+
|        1004 | Aman Gupta   |              8 |
+-------------+--------------+----------------+
1 row in set (0.00 sec)
```

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

```
mysql> select e.event_id,e.event_name,sum(b.num_tickets) as NumberOfTickets,
    -> monthname(b.booking_date) as Month
    -> from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by e.event_id,Month;
+----------+----------------------+-----------------+----------+
| event_id | event_name           | NumberOfTickets | Month    |
+----------+----------------------+-----------------+----------+
|      101 | Bollywood Night      |               2 | January  |
|      102 | Cricket Championship |               5 | January  |
|      103 | Tech Summit          |               3 | January  |
|      104 | Comedy Show          |               4 | January  |
|      105 | Classical Music Gala |               2 | January  |
|      106 | IPL Final            |               3 | January  |
|      107 | IT Conference        |               2 | January  |
|      108 | Stand-up Comedy Night|               2 | January  |
|      109 | Sufi Music Festival  |               4 | January  |
|      110 | Football League      |               5 | January  |
|      111 | Movie Premiere       |               3 | February |
|      106 | IPL Final            |               6 | June     |
+----------+----------------------+-----------------+----------+
12 rows in set (0.01 sec)
```

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

```
mysql> select avg(e.ticket_price) as AveragePrice,v.venue_id as venue_id,v.venue_name as VenueName
    -> from events e
    -> join venue v on e.venue_id=v.venue_id
    -> group by venue_id;
+--------------+----------+----------------------+
| AveragePrice | venue_id | VenueName            |
+--------------+----------+----------------------+
|    1300.0000 |        1 | Taj Convention Center|
|    2750.0000 |        2 | Royal Stadium        |
|    2250.0000 |        3 | Lotus Hall           |
|    1350.0000 |        4 | Grand Auditorium     |
|    1650.0000 |        5 | Pearl Palace         |
|    8500.0000 |        6 | Queen's Pavillion    |
|    1000.0000 |        7 | Saffron Gardens      |
|    1100.0000 |        8 | Starry Arena         |
|    1200.0000 |        9 | Velvet Lounge        |
+--------------+----------+----------------------+
9 rows in set (0.00 sec)
```

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

```
mysql> select e.event_type as EventType,sum(b.num_tickets) as NumberOfSoldTickets
    -> from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by EventType;
+------------+----------------------+
| EventType  | NumberOfSoldTickets |
+------------+----------------------+
| Concert    |                    8 |
| Sports     |                   19 |
| Conference |                    5 |
| Comedy     |                    6 |
| Movie      |                    3 |
+------------+----------------------+
5 rows in set (0.00 sec)
```

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

```
mysql> select e.event_name as EventName,sum((e.total_seats-e.available_seats)*ticket_price) as Total_Revenue,year(e.event_date) as Year
    -> from events e
    -> group by EventName,Year;
+----------------------+---------------+------+
| EventName            | Total_Revenue | Year |
+----------------------+---------------+------+
| Bollywood Night      |         65000 | 2024 |
| Cricket Championship |        500000 | 2024 |
| Tech Summit          |        200000 | 2024 |
| Comedy Show          |         30000 | 2024 |
| Classical Music Gala |         90000 | 2024 |
| IPL Final            |        300000 | 2024 |
| IT Conference        |        125000 | 2024 |
| Stand-up Comedy Night |         24000 | 2024 |
| Sufi Music Festival  |         75000 | 2024 |
| Football League      |        100000 | 2024 |
| Movie Premiere       |         30000 | 2024 |
| Action Blockbuster   |         33000 | 2024 |
| Romantic Movie Night |         24000 | 2024 |
| FIFA CUP             |       4200000 | 2024 |
+----------------------+---------------+------+
14 rows in set (0.00 sec)
```

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

```
mysql> select c.customer_id,c.customer_name,count(b.booking_id) as NoOfEvents
    -> from customer c
    -> join bookings b on c.customer_id=b.customer_id
    -> group by c.customer_id
    -> having NoOfEvents > 1;
+-------------+---------------+------------+
| customer_id | customer_name | NoOfEvents |
+-------------+---------------+------------+
|        1001 | Priya Sharma  |          2 |
|        1004 | Aman Gupta    |          2 |
|        1005 | Nisha Patel   |          2 |
+-------------+---------------+------------+
3 rows in set (0.00 sec)
```

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

```
mysql> select c.customer_id,c.customer_name,e.event_name,sum(b.total_cost) as TotalRevenue
    -> from customer c
    -> join bookings b on c.customer_id=b.customer_id
    -> join events e on e.event_id=b.event_id
    -> group by c.customer_id,e.event_id,c.customer_name;
+-------------+---------------+----------------------+--------------+
| customer_id | customer_name | event_name           | TotalRevenue |
+-------------+---------------+----------------------+--------------+
|        1001 | Priya Sharma  | Bollywood Night      |         2500 |
|        1001 | Priya Sharma  | Movie Premiere       |         3000 |
|        1002 | Rahul Verma   | Cricket Championship |        12500 |
|        1003 | Pooja Singh   | Tech Summit          |         6000 |
|        1004 | Aman Gupta    | Comedy Show          |         6000 |
|        1004 | Aman Gupta    | IPL Final            |         5200 |
|        1005 | Nisha Patel   | Classical Music Gala |         3600 |
|        1005 | Nisha Patel   | IPL Final            |         2500 |
|        1006 | Sameer Shah   | IPL Final            |         9000 |
|        1007 | Anjali Desai  | IT Conference        |         5000 |
|        1008 | Rohan Malhotra | Stand-up Comedy Night |         2400 |
|        1009 | Shreya Kapoor | Sufi Music Festival  |         6000 |
|        1010 | Kartik Joshi  | Football League      |        10000 |
+-------------+---------------+----------------------+--------------+
13 rows in set (0.00 sec)
```

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

```
mysql> select e.event_type,v.venue_name,avg(e.ticket_price) as AVGTCKTPRICE
    -> from events e
    -> join venue v on e.venue_id=v.venue_id
    -> group by e.event_type,v.venue_name;
+------------+----------------------+--------------+
| event_type | venue_name           | AVGTCKTPRICE |
+------------+----------------------+--------------+
| Concert    | Taj Convention Center |    1300.0000 |
| Sports     | Royal Stadium        |    2750.0000 |
| Conference | Lotus Hall           |    2250.0000 |
| Comedy     | Grand Auditorium     |    1350.0000 |
| Concert    | Pearl Palace         |    1650.0000 |
| Sports     | Queen's Pavillion    |    8500.0000 |
| Movie      | Saffron Gardens      |    1000.0000 |
| Movie      | Starry Arena         |    1100.0000 |
| Movie      | Velvet Lounge        |    1200.0000 |
+------------+----------------------+--------------+
9 rows in set (0.00 sec)
```

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

```
mysql> select c.customer_id,c.customer_name,sum(b.num_tickets) as Total_tickets
    -> from customer c
    -> join bookings b on c.customer_id=b.customer_id
    -> where b.booking_date >= curdate() - interval 30 day
    -> group by c.customer_id,c.customer_name;
+-------------+---------------+---------------+
| customer_id | customer_name  | Total_tickets |
+-------------+---------------+---------------+
|        1001 | Priya Sharma   |             5 |
|        1002 | Rahul Verma    |             5 |
|        1003 | Pooja Singh    |             3 |
|        1004 | Aman Gupta     |             8 |
|        1005 | Nisha Patel    |             4 |
|        1006 | Sameer Shah    |             3 |
|        1007 | Anjali Desai   |             2 |
|        1008 | Rohan Malhotra |             2 |
|        1009 | Shreya Kapoor  |             4 |
|        1010 | Kartik Joshi   |             5 |
+-------------+---------------+---------------+
10 rows in set (0.01 sec)
```

Tasks 4: Subquery and its types

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

```
mysql> select venue_id,avg(ticket_price)
    -> from events
    -> where venue_id in(select venue_id from venue)
    -> group by venue_id;
+----------+--------------------+
| venue_id | avg(ticket_price) |
+----------+--------------------+
|        1 |          1300.0000 |
|        2 |          2750.0000 |
|        3 |          2250.0000 |
|        4 |          1350.0000 |
|        5 |          1650.0000 |
|        6 |          8500.0000 |
|        7 |          1000.0000 |
|        8 |          1100.0000 |
|        9 |          1200.0000 |
+----------+--------------------+
9 rows in set (0.00 sec)
```

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

```
mysql> select * from events;
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name           | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
|      101 | Bollywood Night      | 2024-02-15 | 19:00:00   |        1 |         300 |             250 |         1300 | Concert    |
|      102 | Cricket Championship | 2024-03-20 | 16:30:00   |        2 |        1000 |             800 |         2500 | Sports     |
|      103 | Tech Summit          | 2024-04-25 | 10:00:00   |        3 |         500 |             400 |         2000 | Conference |
|      104 | Comedy Show          | 2024-05-10 | 20:30:00   |        4 |         200 |             180 |         1500 | Comedy     |
|      105 | Classical Music Gala | 2024-06-05 | 18:00:00   |        5 |         400 |             350 |         1800 | Concert    |
|      106 | IPL Final            | 2024-06-15 | 19:30:00   |        2 |         800 |             700 |         3000 | Sports     |
|      107 | IT Conference        | 2024-07-12 | 09:00:00   |        3 |         600 |             550 |         2500 | Conference |
|      108 | Stand-up Comedy Night | 2024-08-20 | 21:00:00   |        4 |         250 |             230 |         1200 | Comedy     |
|      109 | Sufi Music Festival  | 2024-09-10 | 17:30:00   |        5 |         350 |             300 |         1500 | Concert    |
|      110 | Football League      | 2024-09-25 | 15:00:00   |        6 |         700 |             650 |         2000 | Sports     |
|      111 | Movie Premiere       | 2024-10-05 | 20:00:00   |        7 |         150 |             120 |         1000 | Movie      |
|      112 | Action Blockbuster   | 2024-10-15 | 18:30:00   |        8 |         180 |             150 |         1100 | Movie      |
|      113 | Romantic Movie Night | 2024-11-01 | 19:30:00   |        9 |         200 |             180 |         1200 | Movie      |
|      114 | FIFA CUP             | 2024-05-12 | 21:00:00   |        6 |         400 |             120 |        15000 | Sports     |
+----------+----------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
14 rows in set (0.00 sec)

mysql> select event_id,event_name,total_seats-available_seats as TotalSoldTickets
    -> from events
    -> where event_id in(
    -> select event_id from events where total_seats-available_seats >= total_seats*0.5);
+----------+------------+------------------+
| event_id | event_name | TotalSoldTickets |
+----------+------------+------------------+
|      114 | FIFA CUP   |              280 |
+----------+------------+------------------+
1 row in set (0.00 sec)
```

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

```
mysql> select event_id,event_name,total_seats-available_seats as TicketsSold
    -> from events;
+----------+----------------------+-------------+
| event_id | event_name           | TicketsSold |
+----------+----------------------+-------------+
|      101 | Bollywood Night      |          50 |
|      102 | Cricket Championship |         200 |
|      103 | Tech Summit          |         100 |
|      104 | Comedy Show          |          20 |
|      105 | Classical Music Gala |          50 |
|      106 | IPL Final            |         100 |
|      107 | IT Conference        |          50 |
|      108 | Stand-up Comedy Night |          20 |
|      109 | Sufi Music Festival  |          50 |
|      110 | Football League      |          50 |
|      111 | Movie Premiere       |          30 |
|      112 | Action Blockbuster   |          30 |
|      113 | Romantic Movie Night |          20 |
|      114 | FIFA CUP             |         280 |
+----------+----------------------+-------------+
14 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 c
    -> where not exists(select customer_id from bookings b where b.customer_id=c.customer_id);
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
|        1010 | Kartik Joshi  |
+-------------+---------------+
1 row in set (0.00 sec)
```

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

```
mysql> select event_id,event_name
    -> from events e
    -> where event_id not in(select event_id from bookings);
+----------+----------------------+
| event_id | event_name           |
+----------+----------------------+
|      110 | Football League      |
|      112 | Action Blockbuster   |
|      113 | Romantic Movie Night |
|      114 | FIFA CUP             |
+----------+----------------------+
4 rows in set (0.00 sec)

mysql> select * from bookings;
+------------+-------------+----------+-------------+------------+--------------+
| booking_id | customer_id | event_id | num_tickets | total_cost | booking_date |
+------------+-------------+----------+-------------+------------+--------------+
|       5001 |        1001 |      101 |           2 |       2500 | 2024-01-20   |
|       5002 |        1002 |      102 |           5 |      12500 | 2024-01-21   |
|       5003 |        1003 |      103 |           3 |       6000 | 2024-01-22   |
|       5004 |        1004 |      104 |           4 |       6000 | 2024-01-23   |
|       5005 |        1005 |      105 |           2 |       3600 | 2024-01-24   |
|       5006 |        1006 |      106 |           3 |       9000 | 2024-01-25   |
|       5007 |        1007 |      107 |           2 |       5000 | 2024-01-26   |
|       5008 |        1008 |      108 |           2 |       2400 | 2024-01-27   |
|       5009 |        1009 |      109 |           4 |       6000 | 2024-01-28   |
|       5011 |        1001 |      111 |           3 |       3000 | 2024-02-01   |
|       5014 |        1004 |      106 |           4 |       5200 | 2024-06-15   |
|       5015 |        1005 |      106 |           2 |       2500 | 2024-06-15   |
+------------+-------------+----------+-------------+------------+--------------+
12 rows in 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,total_tickets_sold
    -> from (select e.event_type,sum(b.num_tickets) as total_tickets_sold
    -> from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by event_type) as total_event_summary;
+------------+--------------------+
| event_type | total_tickets_sold |
+------------+--------------------+
| Concert    |                  8 |
| Sports     |                 14 |
| Conference |                  5 |
| Comedy     |                  6 |
| Movie      |                  3 |
+------------+--------------------+
5 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 events
    -> where ticket_price > (select avg(ticket_price) from events);
+----------+------------+--------------+
| event_id | event_name | ticket_price |
+----------+------------+--------------+
|      106 | IPL Final  |         3000 |
|      114 | FIFA CUP   |        15000 |
+----------+------------+--------------+
2 rows in set (0.00 sec)

mysql> select avg(ticket_price) from events;
+-------------------+
| avg(ticket_price) |
+-------------------+
|         2685.7143 |
+-------------------+
1 row in set (0.00 sec)
```

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

```
mysql> select c.customer_id,c.customer_name,
    -> (select sum(b.num_tickets*e.ticket_price) from bookings b
    -> join events e on b.event_id=e.event_id
    -> where b.customer_id=c.customer_id) as TotalRevenue
    -> from customer c;
+-------------+---------------+--------------+
| customer_id | customer_name | TotalRevenue |
+-------------+---------------+--------------+
|        1001 | Priya Sharma  |         5600 |
|        1002 | Rahul Verma   |        12500 |
|        1003 | Pooja Singh   |         6000 |
|        1004 | Aman Gupta    |        18000 |
|        1005 | Nisha Patel   |         9600 |
|        1006 | Sameer Shah   |         9000 |
|        1007 | Anjali Desai  |         5000 |
|        1008 | Rohan Malhotra|         2400 |
|        1009 | Shreya Kapoor |         6000 |
|        1010 | Kartik Joshi  |         NULL |
+-------------+---------------+--------------+
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 c.customer_id,c.customer_name
    -> from customer c
    -> where exists(select c.customer_id from bookings b
    -> join events e on b.event_id=e.event_id
    -> where b.customer_id=c.customer_id
    -> and e.venue_id=2);
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
|        1002 | Rahul Verma   |
|        1004 | Aman Gupta    |
|        1005 | Nisha Patel   |
|        1006 | Sameer Shah   |
+-------------+---------------+
4 rows in set (0.01 sec)
```

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

```
mysql> select event_type,sum(totalTickets) as totalTickets
    -> from
    -> (select e.event_type,sum(b.num_tickets) as totalTickets from events e
    -> join bookings b on e.event_id=b.event_id
    -> group by e.event_type,b.event_id) as tickets
    -> group by event_type;
+------------+--------------+
| event_type | totalTickets |
+------------+--------------+
| Concert    |            8 |
| Sports     |           14 |
| Conference |            5 |
| Comedy     |            6 |
| Movie      |            3 |
+------------+--------------+
5 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 c.customer_id,c.customer_name
    -> from customer c
    -> where exists(select c.customer_id from bookings b
    -> join events e on b.event_id=e.event_id
    -> where b.customer_id=c.customer_id
    -> and date_format(b.booking_date,'%Y-%m') = '2024-06');
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
|        1004 | Aman Gupta    |
|        1005 | Nisha Patel   |
+-------------+---------------+
2 rows in set (0.00 sec)
```

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

```
mysql> select avg(ticket_price) as AveragePrice,venue_id
    -> from events
    -> where venue_id in(select venue_id from venue)
    -> group by venue_id;
+--------------+----------+
| AveragePrice | venue_id |
+--------------+----------+
|    1300.0000 |        1 |
|    2750.0000 |        2 |
|    2250.0000 |        3 |
|    1350.0000 |        4 |
|    1650.0000 |        5 |
|    8500.0000 |        6 |
|    1000.0000 |        7 |
|    1100.0000 |        8 |
|    1200.0000 |        9 |
+--------------+----------+
9 rows in set (0.00 sec)
```