Assignment – 5

Ticket Booking System

By Esaq A

Task 1:

1. **Creating a database:**
   create database ticketbookingsystem;

```
mysql> create database TicketBookingSystem;
Query OK, 1 row affected (0.503 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| ticketbookingsystem |
| training           |
| workbench          |
| world              |
+--------------------+
8 rows in set (0.051 sec)
```

2. Creating Tables:

- **Venu table:**

  create table venue(venue_id varchar(4) primary key, venue_name varchar(50), address varchar(150));

```
mysql> create table venu(
    -> venue_id varchar(4) primary key,
    -> venue_name varchar(50),
    -> address varchar(150));
Query OK, 0 rows affected (0.259 sec)
```

- **Event Table:**

  create table event(event_id varchar(4) primary key, event_name varchar(40), event_date date, event_time time, venue_id varchar(4), total_seats int, available_seats int, ticket_price decimal(6,2), event_type enum('movie', 'sports', 'concert'));

```
mysql> create table event(event_id varchar(4) primary key,
    -> event_name varchar(40), event_date date,
    -> venue_id varchar(4), total_seats int,
    -> available_seats int, ticket_price decimal(6,2),
    -> event_type enum('movie', 'sports', 'concert'),
    -> foreign key (venue_id) references venue(venue_id));
Query OK, 0 rows affected (0.564 sec)
```

- **Customer Table:**

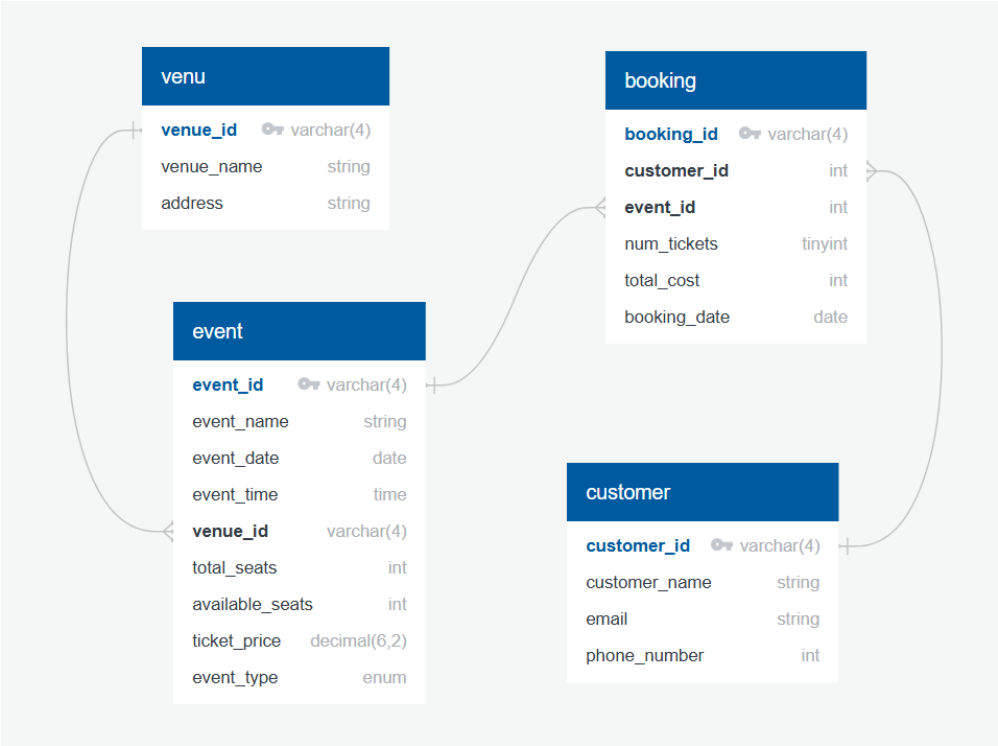
  create table customer(customer_id varchar(4) primary key, customer_name varchar(40), email varchar(150) unique, phone_number varchar(15) unique);

```
mysql> create table customer(
    -> customer_id varchar(4) primary key,
    -> customer_name varchar(40),
    -> email varchar(50) unique,
    -> phone_number varchar(15) unique);
Query OK, 0 rows affected (1.092 sec)
```

- **Booking Table:**
  create table booking(booking_id varchar(4) primary key, customer_id varchar(4), event_id varchar(4), num_tickets tinyint, total_cost int, booking_date date, foreign key (customer_id) references customer(customer_id), foreign key (event_id) references event(event_id));

```
mysql> create table booking(
    -> booking_id varchar(4) primary key,
    -> customer_id varchar(4),
    -> event_id varchar(4),
    -> num_tickets tinyint,
    -> total_cost int,
    -> booking_date date,
    -> foreign key (customer_id) references customer(customer_id),
    -> foreign key (event_id) references event(event_id));
Query OK, 0 rows affected (0.857 sec)
```

3. Create ER Diagram for the Database:



4. Created the primary and foreign constraints while creating tables.

Task 2:

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

Venue table:

insert into venu values

('V001', 'Mayajaal Cinemas', 'ECR, Chennai'),

('V002', 'SDAT Aquatic Complex', 'Velachery, Chennai'),

('V003', 'Music Academy', 'TTK Rd, Chennai'),

('V004', 'Express Avenue', 'Royapettah, Chennai'),

('V005', 'Chennai Trade Centre', 'Nandambakkam, Chennai'),

('V006', 'Sathyam Cinemas', 'Royapettah, Chennai'),

('V007', 'M. A. Chidambaram Stadium', 'Chepauk, Chennai'),

('V008', 'Nehru Indoor Stadium', 'Periamet, Chennai'),

('V009', 'Phoenix Market City', 'Velachery, Chennai'),

('V010', 'VGP Universal Kingdom', 'ECR, Chennai');

```
mysql> insert into venu values
    -> ('V001', 'Mayajaal Cinemas', 'ECR, Chennai'),
    -> ('V002', 'SDAT Aquatic Complex', 'Velachery, Chennai'),
    -> ('V003', 'Music Academy', 'TTK Rd, Chennai'),
    -> ('V004', 'Express Avenue', 'Royapettah, Chennai'),
    -> ('V005', 'Chennai Trade Centre', 'Nandambakkam, Chennai'),
    -> ('V006', 'Sathyam Cinemas', 'Royapettah, Chennai'),
    -> ('V007', 'M. A. Chidambaram Stadium', 'Chepauk, Chennai'),
    -> ('V008', 'Nehru Indoor Stadium', 'Periamet, Chennai'),
    -> ('V009', 'Phoenix Market City', 'Velachery, Chennai'),
    -> ('V010', 'VGP Universal Kingdom', 'ECR, Chennai');
Query OK, 10 rows affected (0.477 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Event table:

insert into event values

('E001', 'World Cup Final', '2023-01-28', '14:00:00', 'V007', 40000, 15000, 2500.00, 'sports'),

('E002', 'Rock Concert', '2024-05-08', '18:00:00', 'V003', 3000, 1200, 1500.00, 'concert'),

('E003', 'IPL Match', '2024-02-29', '14:00:00', 'V007', 38000, 5000, 2000.00, 'sports'),

('E004', 'Movie Premiere', '2024-06-26', '20:00:00', 'V001', 600, 150, 300.00, 'movie'),

('E005', 'Jazz Night Concert', '2023-09-03', '14:00:00', 'V003', 1200, 0, 1000.00, 'concert'),

('E006', 'Music Fest Chennai', '2023-07-14', '15:00:00', 'V008', 8000, 4000, 1800.00, 'concert'),

('E007', 'Aquatic Showdown', '2023-07-22', '16:00:00', 'V002', 5000, 3000, 1300.00, 'sports'),

('E008', 'Comedy Night Live', '2024-05-02', '17:00:00', 'V006', 400, 200, 450.00, 'movie'),

('E009', 'Carnatic Concert', '2024-08-07', '20:00:00', 'V003', 1000, 800, 700.00, 'concert'),

('E010', 'Youth Cup Finals', '2023-03-11', '17:00:00', 'V002', 7000, 6000, 1200.00, 'sports'),

('E011', 'Zest of Music', '2025-09-22', '15:00:00', 'V009', 3000, 500, 1600.00, 'concert'),

('E012', 'Indie Movie Night', '2023-01-09', '19:00:00', 'V004', 500, 100, 350.00, 'movie'),

('E013', 'Fantasy Cup Match', '2024-12-18', '17:00:00', 'V007', 45000, 20000, 2200.00, 'sports'),

('E014', 'VGP Concert Fiesta', '2023-09-06', '17:00:00', 'V010', 1500, 1000, 1100.00, 'concert'),

('E015', 'Summer Blockbuster', '2025-08-08', '21:00:00', 'V001', 800, 250, 300.00, 'movie');

```
mysql> insert into event values
    -> ('E001', 'World Cup Final', '2023-01-28', '14:00:00', 'V007', 40000, 15000, 2500.00, 'sports'),
    -> ('E002', 'Rock Concert', '2024-05-08', '18:00:00', 'V003', 3000, 1200, 1500.00, 'concert'),
    -> ('E003', 'IPL Match', '2024-02-29', '14:00:00', 'V007', 38000, 5000, 2000.00, 'sports'),
    -> ('E004', 'Movie Premiere', '2024-06-26', '20:00:00', 'V001', 600, 150, 300.00, 'movie'),
    -> ('E005', 'Jazz Night Concert', '2023-09-03', '14:00:00', 'V003', 1200, 0, 1000.00, 'concert'),
    -> ('E006', 'Music Fest Chennai', '2023-07-14', '15:00:00', 'V008', 8000, 4000, 1800.00, 'concert'),
    -> ('E007', 'Aquatic Showdown', '2023-07-22', '16:00:00', 'V002', 5000, 3000, 1300.00, 'sports'),
    -> ('E008', 'Comedy Night Live', '2024-05-02', '17:00:00', 'V006', 400, 200, 450.00, 'movie'),
    -> ('E009', 'Carnatic Concert', '2024-08-07', '20:00:00', 'V003', 1000, 800, 700.00, 'concert'),
    -> ('E010', 'Youth Cup Finals', '2023-03-11', '17:00:00', 'V002', 7000, 6000, 1200.00, 'sports'),
    -> ('E011', 'Zest of Music', '2025-09-22', '15:00:00', 'V009', 3000, 500, 1600.00, 'concert'),
    -> ('E012', 'Indie Movie Night', '2023-01-09', '19:00:00', 'V004', 500, 100, 350.00, 'movie'),
    -> ('E013', 'Fantasy Cup Match', '2024-12-18', '17:00:00', 'V007', 45000, 20000, 2200.00, 'sports'),
    -> ('E014', 'VGP Concert Fiesta', '2023-09-06', '17:00:00', 'V010', 1500, 1000, 1100.00, 'concert'),
    -> ('E015', 'Summer Blockbuster', '2025-08-08', '21:00:00', 'V001', 800, 250, 300.00, 'movie');
Query OK, 15 rows affected (0.074 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

Customer table:

insert into customer values

('C001', 'Aarav', 'aarav@mail.com', '9876510000'),

('C002', 'Diya', 'diya@mail.com', '9876510001'),

('C003', 'Karan', 'karan@mail.com', '9876510002'),

('C004', 'Meena', 'meena@mail.com', '9876510003'),

('C005', 'Ravi', 'ravi@mail.com', '9876510004'),

('C006', 'Sana', 'sana@mail.com', '9876510005'),

('C007', 'Vikram', 'vikram@mail.com', '9876510006'),

('C008', 'Neha', 'neha@mail.com', '9876510007'),

('C009', 'Amit', 'amit@mail.com', '9876510008'),

('C010', 'Riya', 'riya@mail.com', '9876510009'),

('C011', 'Tarun', 'tarun@mail.com', '9876510010'),

('C012', 'Divya', 'divya@mail.com', '9876510011'),

('C013', 'Kabir', 'kabir@mail.com', '9876510012'),

('C014', 'Leela', 'leela@mail.com', '9876510013'),

('C015', 'Farhan', 'farhan@mail.com', '9876510014');

```
mysql> insert into customer values
    -> ('C001', 'Aarav', 'aarav@mail.com', '9876510000'),
    -> ('C002', 'Diya', 'diya@mail.com', '9876510001'),
    -> ('C003', 'Karan', 'karan@mail.com', '9876510002'),
    -> ('C004', 'Meena', 'meena@mail.com', '9876510003'),
    -> ('C005', 'Ravi', 'ravi@mail.com', '9876510004'),
    -> ('C006', 'Sana', 'sana@mail.com', '9876510005'),
    -> ('C007', 'Vikram', 'vikram@mail.com', '9876510006'),
    -> ('C008', 'Neha', 'neha@mail.com', '9876510007'),
    -> ('C009', 'Amit', 'amit@mail.com', '9876510008'),
    -> ('C010', 'Riya', 'riya@mail.com', '9876510009'),
    -> ('C011', 'Tarun', 'tarun@mail.com', '9876510010'),
    -> ('C012', 'Divya', 'divya@mail.com', '9876510011'),
    -> ('C013', 'Kabir', 'kabir@mail.com', '9876510012'),
    -> ('C014', 'Leela', 'leela@mail.com', '9876510013'),
    -> ('C015', 'Farhan', 'farhan@mail.com', '9876510014');
Query OK, 15 rows affected (0.085 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

Booking table:

insert into booking values ('B001', 'C001', 'E013', 5, 11450, '2025-06-02'),

('B002', 'C001', 'E007', 4, 6168, '2025-04-19'), ('B003', 'C001', 'E001', 4, 7064, '2025-05-25'),

('B004', 'C001', 'E005', 5, 5970, '2025-06-07'), ('B005', 'C002', 'E005', 1, 1326, '2025-04-22'),

('B006', 'C002', 'E003', 5, 4505, '2025-03-16'), ('B007', 'C003', 'E002', 6, 9912, '2025-04-07'),

('B008', 'C003', 'E012', 4, 2848, '2025-05-01'), ('B009', 'C003', 'E015', 3, 6234, '2025-04-29'),

('B010', 'C004', 'E010', 5, 11265, '2025-04-21'), ('B011', 'C004', 'E011', 4, 9740, '2025-06-10'),

('B012', 'C004', 'E004', 3, 1665, '2025-05-28'),('B013', 'C005', 'E002', 6, 11598, '2025-06-01'),

('B014', 'C006', 'E010', 4, 6656, '2025-04-11'), ('B015', 'C007', 'E012', 6, 3342, '2025-04-28'),

('B016', 'C007', 'E006', 2, 2416, '2025-04-16'),('B017', 'C008', 'E013', 5, 10670, '2025-05-08'),

('B018', 'C008', 'E003', 1, 629, '2025-06-07'), ('B019', 'C009', 'E015', 1, 1534, '2025-06-09'),

('B020', 'C009', 'E009', 5, 7460, '2025-04-03'),('B021', 'C009', 'E008', 6, 4866, '2025-05-01'),

('B022', 'C010', 'E014', 5, 7390, '2025-03-26'),('B023', 'C010', 'E009', 4, 2700, '2025-04-09'),

('B024', 'C010', 'E004', 5, 9380, '2025-03-26'),('B025', 'C011', 'E010', 2, 2150, '2025-05-25'),

('B026', 'C011', 'E004', 2, 870, '2025-03-22'),('B027', 'C011', 'E005', 5, 6825, '2025-05-12'),

('B028', 'C012', 'E002', 2, 916, '2025-05-29'),('B029', 'C012', 'E015', 1, 1902, '2025-04-06'),

('B030', 'C012', 'E011', 6, 14688, '2025-06-09');

```
mysql> insert into booking values
    -> ('B001', 'C001', 'E013', 5, 11450, '2025-06-02'),
    -> ('B002', 'C001', 'E007', 4, 6168, '2025-04-19'),
    -> ('B003', 'C001', 'E001', 4, 7064, '2025-05-25'),
    -> ('B004', 'C001', 'E005', 5, 5970, '2025-06-07'),
    -> ('B005', 'C002', 'E005', 1, 1326, '2025-04-22'),
    -> ('B006', 'C002', 'E003', 5, 4505, '2025-03-16'),
    -> ('B007', 'C003', 'E002', 6, 9912, '2025-04-07'),
    -> ('B008', 'C003', 'E012', 4, 2848, '2025-05-01'),
    -> ('B009', 'C003', 'E015', 3, 6234, '2025-04-29'),
    -> ('B010', 'C004', 'E010', 5, 11265, '2025-04-21'),
    -> ('B011', 'C004', 'E011', 4, 9740, '2025-06-10'),
    -> ('B012', 'C004', 'E004', 3, 1665, '2025-05-28'),
    -> ('B013', 'C005', 'E002', 6, 11598, '2025-06-01'),
    -> ('B014', 'C006', 'E010', 4, 6656, '2025-04-11'),
    -> ('B015', 'C007', 'E012', 6, 3342, '2025-04-28'),
    -> ('B016', 'C007', 'E006', 2, 2416, '2025-04-16'),
    -> ('B017', 'C008', 'E013', 5, 10670, '2025-05-08'),
    -> ('B018', 'C008', 'E003', 1, 629, '2025-06-07'),
    -> ('B019', 'C009', 'E015', 1, 1534, '2025-06-09'),
    -> ('B020', 'C009', 'E009', 5, 7460, '2025-04-03'),
    -> ('B021', 'C009', 'E008', 6, 4866, '2025-05-01'),
    -> ('B022', 'C010', 'E014', 5, 7390, '2025-03-26'),
    -> ('B023', 'C010', 'E009', 4, 2700, '2025-04-09'),
    -> ('B024', 'C010', 'E004', 5, 9380, '2025-03-26'),
    -> ('B025', 'C011', 'E010', 2, 2150, '2025-05-25'),
    -> ('B026', 'C011', 'E004', 2, 870, '2025-03-22'),
    -> ('B027', 'C011', 'E005', 5, 6825, '2025-05-12'),
    -> ('B028', 'C012', 'E002', 2, 916, '2025-05-29'),
    -> ('B029', 'C012', 'E015', 1, 1902, '2025-04-06'),
    -> ('B030', 'C012', 'E011', 6, 14688, '2025-06-09');
Query OK, 30 rows affected (0.057 sec)
Records: 30  Duplicates: 0  Warnings: 0
```

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 |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E001     | World Cup Final    | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E002     | Rock Concert       | 2025-05-08 | 18:00:00   | V003     |        3000 |            1200 |      1500.00 | concert    |
| E003     | IPL Match          | 2025-02-28 | 14:00:00   | V007     |       38000 |            5000 |      2000.00 | sports     |
| E004     | Movie Premiere     | 2025-06-26 | 20:00:00   | V001     |         600 |             150 |       300.00 | movie      |
| E005     | Jazz Night Concert | 2025-09-03 | 14:00:00   | V003     |        1200 |               0 |      1000.00 | concert    |
| E006     | Music Fest Chennai | 2025-07-14 | 15:00:00   | V008     |        8000 |            4000 |      1800.00 | concert    |
| E007     | Aquatic Showdown   | 2025-07-22 | 16:00:00   | V002     |        5000 |            3000 |      1300.00 | sports     |
| E008     | Comedy Night Live  | 2025-05-02 | 17:00:00   | V006     |         400 |             200 |       450.00 | movie      |
| E009     | Carnatic Concert   | 2025-08-07 | 20:00:00   | V003     |        1000 |             800 |       700.00 | concert    |
| E010     | Youth Cup Finals   | 2025-03-11 | 17:00:00   | V002     |        7000 |            6000 |      1200.00 | sports     |
| E011     | Zest of Music      | 2025-09-22 | 15:00:00   | V009     |        3000 |             500 |      1600.00 | concert    |
| E012     | Indie Movie Night  | 2025-01-09 | 19:00:00   | V004     |         500 |             100 |       350.00 | movie      |
| E013     | Fantasy Cup Match  | 2025-12-18 | 17:00:00   | V007     |       45000 |           20000 |      2200.00 | sports     |
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00   | V010     |        1500 |            1000 |      1100.00 | concert    |
| E015     | Summer Blockbuster | 2025-08-08 | 21:00:00   | V001     |         800 |             250 |       300.00 | movie      |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
15 rows in set (0.010 sec)
```

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

*select * from event where available_tickets > 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 |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E001     | World Cup Final    | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E004     | Movie Premiere     | 2025-06-26 | 20:00:00   | V001     |         600 |             150 |       300.00 | movie      |
| E006     | Music Fest Chennai | 2025-07-14 | 15:00:00   | V008     |        8000 |            4000 |      1800.00 | concert    |
| E007     | Aquatic Showdown   | 2025-07-22 | 16:00:00   | V002     |        5000 |            3000 |      1300.00 | sports     |
| E008     | Comedy Night Live  | 2025-05-02 | 17:00:00   | V006     |         400 |              35 |       450.00 | movie      |
| E009     | Carnatic Concert   | 2025-08-07 | 20:00:00   | V003     |        1000 |             800 |       700.00 | concert    |
| E010     | Youth Cup Finals   | 2025-03-11 | 17:00:00   | V002     |        7000 |            1063 |      1200.00 | sports     |
| E011     | Zest of Music      | 2025-09-22 | 15:00:00   | V009     |        3000 |             500 |      1600.00 | concert    |
| E013     | Fantasy Cup Match  | 2025-12-18 | 17:00:00   | V007     |       45000 |           20000 |      2200.00 | sports     |
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00   | V010     |        1500 |            1000 |      1100.00 | concert    |
| E015     | Summer Blockbuster | 2025-08-08 | 21:00:00   | V001     |         800 |             250 |       300.00 | movie      |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
11 rows in set (0.028 sec)
```

4.Write a SQL query to select events 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 |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E001     | World Cup Final | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E010     | Youth Cup Finals| 2025-03-11 | 17:00:00   | V002     |        7000 |            6000 |      1200.00 | sports     |
| E013     | Fantasy Cup Match| 2025-12-18 | 17:00:00   | V007     |       45000 |           20000 |      2200.00 | sports     |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
3 rows in set (0.057 sec)
```

5.Write a SQL query to select events with ticket price range is 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 |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E001     | World Cup Final    | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E002     | Rock Concert       | 2025-05-08 | 18:00:00   | V003     |        3000 |               0 |      1500.00 | concert    |
| E003     | IPL Match          | 2025-02-28 | 14:00:00   | V007     |       38000 |            5000 |      2000.00 | sports     |
| E005     | Jazz Night Concert | 2025-09-03 | 14:00:00   | V003     |        1200 |               0 |      1000.00 | concert    |
| E006     | Music Fest Chennai | 2025-07-14 | 15:00:00   | V008     |        8000 |            4000 |      1800.00 | concert    |
| E007     | Aquatic Showdown   | 2025-07-22 | 16:00:00   | V002     |        5000 |            3000 |      1300.00 | sports     |
| E010     | Youth Cup Finals   | 2025-03-11 | 17:00:00   | V002     |        7000 |            6000 |      1200.00 | sports     |
| E011     | Zest of Music      | 2025-09-22 | 15:00:00   | V009     |        3000 |             500 |      1600.00 | concert    |
| E013     | Fantasy Cup Match  | 2025-12-18 | 17:00:00   | V007     |       45000 |           20000 |      2200.00 | sports     |
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00   | V010     |        1500 |            1000 |      1100.00 | concert    |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
10 rows in set (0.028 sec)
```

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

*select * from event where event_date between '2024-01-01' and '2024-12-31';*

```
mysql> select * from event where event_date between '2025-05-01' and '2025-06-30';
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name      | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E002     | Rock Concert    | 2025-05-08 | 18:00:00   | V003     |        3000 |               0 |      1500.00 | concert    |
| E003     | IPL Match       | 2025-05-28 | 14:00:00   | V007     |       38000 |               0 |      2000.00 | sports     |
| E004     | Movie Premiere  | 2025-06-26 | 20:00:00   | V001     |         600 |             150 |       300.00 | movie      |
| E008     | Comedy Night Live | 2025-05-02 | 17:00:00   | V006     |         400 |              35 |       450.00 | movie      |
| E012     | Indie Movie Night | 2025-06-09 | 19:00:00   | V004     |         500 |               0 |       350.00 | movie      |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
5 rows in set (0.015 sec)
```

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

*select * from event where available_tickets > 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 |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E009     | Carnatic Concert   | 2025-08-07 | 20:00:00   | V003     |        1000 |             800 |       700.00 | concert    |
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00   | V010     |        1500 |            1000 |      1100.00 | concert    |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
2 rows in set (0.016 sec)
```

8.Write a SQL query to 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 |
+-------------+---------------+-----------------+--------------+
| C006        | Sana          | sana@mail.com   | 9876510005   |
| C007        | Vikram        | vikram@mail.com | 9876510006   |
| C008        | Neha          | neha@mail.com   | 9876510007   |
| C009        | Amit          | amit@mail.com   | 9876510008   |
| C010        | Riya          | riya@mail.com   | 9876510009   |
+-------------+---------------+-----------------+--------------+
5 rows in set (0.068 sec)
```

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

*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 |
+------------+-------------+----------+-------------+------------+--------------+
| B001       | C001        | E013     |           5 |      11000 | 2025-06-02   |
| B004       | C001        | E005     |           5 |       5000 | 2025-06-07   |
| B006       | C002        | E003     |           5 |      10000 | 2025-03-16   |
| B007       | C003        | E002     |           6 |       9000 | 2025-04-07   |
| B010       | C004        | E010     |           5 |       6000 | 2025-04-21   |
| B013       | C005        | E002     |           6 |       9000 | 2025-06-01   |
| B015       | C007        | E012     |           6 |       2100 | 2025-04-28   |
| B017       | C008        | E013     |           5 |      11000 | 2025-05-08   |
| B020       | C009        | E009     |           5 |       3500 | 2025-04-03   |
| B021       | C009        | E008     |           6 |       2700 | 2025-05-01   |
| B022       | C010        | E014     |           5 |       5500 | 2025-03-26   |
| B024       | C010        | E004     |           5 |       1500 | 2025-03-26   |
| B027       | C011        | E005     |           5 |       5000 | 2025-05-12   |
| B030       | C012        | E011     |           6 |       9600 | 2025-06-09   |
+------------+-------------+----------+-------------+------------+--------------+
14 rows in set (0.382 sec)
```

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

*select * from customer where phone like '%000';*

```
mysql> select * from customer where phone_number like '%000';
+-------------+---------------+---------------+--------------+
| customer_id | customer_name | email         | phone_number |
+-------------+---------------+---------------+--------------+
| C001        | Aarav         | aarav@mail.com | 9876510000   |
+-------------+---------------+---------------+--------------+
1 row in set (0.015 sec)
```

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

*select * from event where total_seats > 15000 order by event_date;*

```
mysql> select * from event where total_seats > 15000 order by total_seats;
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name      | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E003     | IPL Match       | 2025-05-28 | 14:00:00   | V007     |       38000 |               0 |      2000.00 | sports     |
| E001     | World Cup Final | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E013     | Fantasy Cup Match | 2025-12-18 | 17:00:00 | V007     |       45000 |           20000 |      2200.00 | sports     |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
3 rows in set (0.103 sec)
```

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

*select * from event where event_name not like 'x%' and event_name 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 |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E001     | World Cup Final | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E002     | Rock Concert    | 2025-05-08 | 18:00:00   | V003     |        3000 |               0 |      1500.00 | concert    |
| E003     | IPL Match       | 2025-05-28 | 14:00:00   | V007     |       38000 |               0 |      2000.00 | sports     |
| E004     | Movie Premiere  | 2025-06-26 | 20:00:00   | V001     |         600 |             150 |       300.00 | movie      |
| E005     | Jazz Night Concert | 2025-09-03 | 14:00:00 | V003     |        1200 |               0 |      1000.00 | concert    |
| E006     | Music Fest Chennai | 2025-07-14 | 15:00:00 | V008     |        8000 |            4000 |      1800.00 | concert    |
| E007     | Aquatic Showdown | 2025-07-22 | 16:00:00   | V002     |        5000 |            3000 |      1300.00 | sports     |
| E008     | Comedy Night Live | 2025-05-02 | 17:00:00 | V006     |         400 |              35 |       450.00 | movie      |
| E009     | Carnatic Concert | 2025-08-07 | 20:00:00   | V003     |        1000 |             800 |       700.00 | concert    |
| E012     | Indie Movie Night | 2025-06-09 | 19:00:00 | V004     |         500 |               0 |       350.00 | movie      |
| E013     | Fantasy Cup Match | 2025-12-18 | 17:00:00 | V007     |       45000 |           20000 |      2200.00 | sports     |
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00 | V010     |        1500 |            1000 |      1100.00 | concert    |
| E015     | Summer Blockbuster | 2025-08-08 | 21:00:00 | V001     |         800 |             250 |       300.00 | movie      |
+----------+-----------------+------------+------------+----------+-------------+-----------------+--------------+------------+
13 rows in set (0.040 sec)
```

Task 3:

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

select event_name, avg(ticket_price) from event group by event_name;

```
mysql> select event_name, avg(ticket_price) as average_ticket from event group by event_name;
+--------------------+----------------+
| event_name         | average_ticket |
+--------------------+----------------+
| World Cup Final    |    2500.000000 |
| Rock Concert       |    1500.000000 |
| IPL Match          |    2000.000000 |
| Movie Premiere     |     300.000000 |
| Jazz Night Concert |    1000.000000 |
| Music Fest Chennai |    1800.000000 |
| Aquatic Showdown   |    1300.000000 |
| Comedy Night Live  |     450.000000 |
| Carnatic Concert   |     700.000000 |
| Youth Cup Finals   |    1200.000000 |
| Zest of Music      |    1600.000000 |
| Indie Movie Night  |     350.000000 |
| Fantasy Cup Match  |    2200.000000 |
| VGP Concert Fiesta |    1100.000000 |
| Summer Blockbuster |     300.000000 |
+--------------------+----------------+
15 rows in set (0.128 sec)
```

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

*select sum(ticket_price * num_tickets) as total_revenue from booking join event on booking.event_id = event.event_id;*

```
mysql> select event_id, event_name, ((total_seats - available_seats) * ticket_price) as total_revenue from event;
+----------+--------------------+---------------+
| event_id | event_name         | total_revenue |
+----------+--------------------+---------------+
| E001     | World Cup Final    |   62500000.00 |
| E002     | Rock Concert       |    4500000.00 |
| E003     | IPL Match          |   76000000.00 |
| E004     | Movie Premiere     |     135000.00 |
| E005     | Jazz Night Concert |    1200000.00 |
| E006     | Music Fest Chennai |    7200000.00 |
| E007     | Aquatic Showdown   |    2600000.00 |
| E008     | Comedy Night Live  |     164250.00 |
| E009     | Carnatic Concert   |     140000.00 |
| E010     | Youth Cup Finals   |    7124400.00 |
| E011     | Zest of Music      |    4000000.00 |
| E012     | Indie Movie Night  |     175000.00 |
| E013     | Fantasy Cup Match  |   55000000.00 |
| E014     | VGP Concert Fiesta |     550000.00 |
| E015     | Summer Blockbuster |     165000.00 |
+----------+--------------------+---------------+
15 rows in set (0.034 sec)
```

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

select event_id, sum(num_tickets) from booking group by event_id;

```
mysql> select * from event where available_seats < 50 order by total_seats desc
    -> limit 1;
+----------+------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E003     | IPL Match  | 2025-05-28 | 14:00:00   | V007     |       38000 |               0 |      2000.00 | sports     |
+----------+------------+------------+------------+----------+-------------+-----------------+--------------+------------+
1 row in set (0.085 sec)
```

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

select event_id, sum(num_tickets) from booking group by event_id;

```
mysql> select event_id, event_name, total_seats , total_seats-available_seats as tickets_sold from event;
+----------+--------------------+-------------+--------------+
| event_id | event_name         | total_seats | tickets_sold |
+----------+--------------------+-------------+--------------+
| E001     | World Cup Final    |       40000 |        25000 |
| E002     | Rock Concert       |        3000 |         3000 |
| E003     | IPL Match          |       38000 |        38000 |
| E004     | Movie Premiere     |         600 |          450 |
| E005     | Jazz Night Concert |        1200 |         1200 |
| E006     | Music Fest Chennai |        8000 |         4000 |
| E007     | Aquatic Showdown   |        5000 |         2000 |
| E008     | Comedy Night Live  |         400 |          365 |
| E009     | Carnatic Concert   |        1000 |          200 |
| E010     | Youth Cup Finals   |        7000 |         5937 |
| E011     | Zest of Music      |        3000 |         2500 |
| E012     | Indie Movie Night  |         500 |          500 |
| E013     | Fantasy Cup Match  |       45000 |        25000 |
| E014     | VGP Concert Fiesta |        1500 |          500 |
| E015     | Summer Blockbuster |         800 |          550 |
+----------+--------------------+-------------+--------------+
15 rows in set (0.010 sec)
```

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

*select * from event where event_id not in (select event_id from booking);*

```
mysql> select * from event where available_seats = total_seats;
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name         | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00   | V010     |        1500 |            1500 |      1100.00 | concert    |
+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
1 row in set (0.012 sec)
```

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

select customer_id, sum(num_tickets) as total_booked from booking group by customer_id order by total_booked desc limit 1;

```
mysql> select customer_id, sum(num_tickets) as total_tickets
    -> from booking
    -> group by customer_id
    -> order by total_tickets desc
    -> limit 1;
+-------------+---------------+
| customer_id | total_tickets |
+-------------+---------------+
| C001        |            18 |
+-------------+---------------+
1 row in set (0.012 sec)
```

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

select date_format(booking_date, '%Y-%m') as month, sum(num_tickets) from booking group by month;

```
mysql> select e.event_name, monthname(b.booking_date) as month, sum(num_tickets) as tickets_sold
    -> from booking b join event e on b.event_id = e.event_id
    -> group by e.event_name, month
    -> order by e.event_name, month;
+--------------------+-------+--------------+
| event_name         | month | tickets_sold |
+--------------------+-------+--------------+
| Aquatic Showdown   | April |            4 |
| Carnatic Concert   | April |            9 |
| Comedy Night Live  | May   |            6 |
| Fantasy Cup Match  | June  |            5 |
| Fantasy Cup Match  | May   |            5 |
| Indie Movie Night  | April |            6 |
| Indie Movie Night  | May   |            4 |
| IPL Match          | June  |            1 |
| IPL Match          | March |           12 |
| Jazz Night Concert | April |            1 |
| Jazz Night Concert | June  |            5 |
| Jazz Night Concert | May   |            5 |
| Movie Premiere     | March |            7 |
| Movie Premiere     | May   |            3 |
| Music Fest Chennai | April |            2 |
| Rock Concert       | April |            6 |
| Rock Concert       | June  |            6 |
| Rock Concert       | May   |            2 |
| Summer Blockbuster | April |            4 |
| Summer Blockbuster | June  |            1 |
| World Cup Final    | May   |            4 |
| Youth Cup Finals   | April |            9 |
| Youth Cup Finals   | May   |            2 |
| Zest of Music      | June  |           10 |
+--------------------+-------+--------------+
24 rows in set (0.015 sec)
```

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

Select v.venue_id, v.venue_name, avg(e.ticket_price) as avg_ticketprice from event e join venue v on v.venue_id = e.venue_id group by venue_id order by v.venue_id;

```
mysql> select v.venue_id, v.venue_name, avg(e.ticket_price) as avg_ticketprice from event e
    -> join venue v on v.venue_id = e.venue_id group by v.venue_id order by v.venue_id;
+----------+---------------------------+-----------------+
| venue_id | venue_name                | avg_ticketprice |
+----------+---------------------------+-----------------+
| V001     | Mayajaal Cinemas          |      300.000000 |
| V002     | SDAT Aquatic Complex      |     1250.000000 |
| V003     | Music Academy             |     1066.666667 |
| V004     | Express Avenue            |      350.000000 |
| V006     | Sathyam Cinemas           |      450.000000 |
| V007     | M. A. Chidambaram Stadium |     2233.333333 |
| V008     | Nehru Indoor Stadium      |     1800.000000 |
| V009     | Phoenix Market City       |     1600.000000 |
| V010     | VGP Universal Kingdom     |     1100.000000 |
+----------+---------------------------+-----------------+
9 rows in set (0.013 sec)
```

9. Write a SQL query to 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.event_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     |                  42 |
| concert    |                  46 |
| movie      |                  31 |
+------------+---------------------+
3 rows in set (0.041 sec)
```

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

*select e.event_name, year(b.booking_date) as year, sum(num_tickets*ticket_price) as revenue from booking b join event e on b.event_id = e.event_id group by e.event_name, year order by e.event_name;*

```
mysql> select e.event_name, year(b.booking_date) as year, sum(num_tickets*ticket_price) as revenue from
booking b join event e on b.event_id = e.event_id group by e.event_name, year order by e.event_name;
+--------------------+------+----------+
| event_name         | year | revenue  |
+--------------------+------+----------+
| Aquatic Showdown   | 2025 |  5200.00 |
| Carnatic Concert   | 2025 |  6300.00 |
| Comedy Night Live  | 2025 |  2700.00 |
| Fantasy Cup Match  | 2025 | 22000.00 |
| Indie Movie Night  | 2025 |  3500.00 |
| IPL Match          | 2025 | 26000.00 |
| Jazz Night Concert | 2025 | 11000.00 |
| Movie Premiere     | 2025 |  3000.00 |
| Music Fest Chennai | 2025 |  3600.00 |
| Rock Concert       | 2025 | 21000.00 |
| Summer Blockbuster | 2025 |  1500.00 |
| World Cup Final    | 2025 | 10000.00 |
| Youth Cup Finals   | 2025 | 13200.00 |
| Zest of Music      | 2025 | 16000.00 |
+--------------------+------+----------+
14 rows in set (0.013 sec)
```

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

select b.customer_id, c.customer_name, count(distinct b.event_id) as no_of_event from booking b join customer c on b.customer_id =c.customer_id group by b.customer_id having no_of_event>1;

```
mysql> select b.customer_id, c.customer_name, count(distinct b.event_id) as no_of_event from booking b
    -> join customer c on b.customer_id = c.customer_id group by b.customer_id having no_of_event>1;
+-------------+---------------+--------------+
| customer_id | customer_name | no_of_event  |
+-------------+---------------+--------------+
| C001        | Aarav         |            4 |
| C002        | Diya          |            2 |
| C003        | Karan         |            3 |
| C004        | Meena         |            3 |
| C007        | Vikram        |            2 |
| C008        | Neha          |            2 |
| C009        | Amit          |            3 |
| C010        | Riya          |            3 |
| C011        | Tarun         |            3 |
| C012        | Divya         |            3 |
+-------------+---------------+--------------+
10 rows in set (0.016 sec)
```

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

select e.event_name, c.customer_name, sum(total_cost) as total_revenue from booking b join customer c on b.customer_id = c.customer_id join event e on b.event_id=e.event_id group by e.event_id, c.customer_id order by b.event_id, b.customer_id;

```
mysql> select e.event_name, c.customer_name, sum(total_cost) as total_revenue from booking b
    -> join customer c on b.customer_id = c.customer_id join event e on b.event_id = e.event_id
    -> group by e.event_id, c.customer_id order by b.event_id, b.customer_id;
+--------------------+---------------+---------------+
| event_name         | customer_name | total_revenue |
+--------------------+---------------+---------------+
| World Cup Final    | Aarav         |         10000 |
| Rock Concert       | Karan         |          9000 |
| Rock Concert       | Ravi          |          9000 |
| Rock Concert       | Divya         |          3000 |
| IPL Match          | Diya          |         10000 |
| IPL Match          | Neha          |          2000 |
| IPL Match          | Riya          |         14000 |
| Movie Premiere     | Meena         |           900 |
| Movie Premiere     | Riya          |          1500 |
| Movie Premiere     | Tarun         |           600 |
| Jazz Night Concert | Aarav         |          5000 |
| Jazz Night Concert | Diya          |          1000 |
| Jazz Night Concert | Tarun         |          5000 |
| Music Fest Chennai | Vikram        |          3600 |
| Aquatic Showdown   | Aarav         |          5200 |
| Comedy Night Live  | Amit          |          2700 |
| Carnatic Concert   | Amit          |          3500 |
| Carnatic Concert   | Riya          |          2800 |
| Youth Cup Finals   | Meena         |          6000 |
| Youth Cup Finals   | Sana          |          4800 |
| Youth Cup Finals   | Tarun         |          2400 |
| Zest of Music      | Meena         |          6400 |
| Zest of Music      | Divya         |          9600 |
| Indie Movie Night  | Karan         |          1400 |
| Indie Movie Night  | Vikram        |          2100 |
| Fantasy Cup Match  | Aarav         |         11000 |
| Fantasy Cup Match  | Neha          |         11000 |
| Summer Blockbuster | Karan         |           900 |
| Summer Blockbuster | Amit          |           300 |
| Summer Blockbuster | Divya         |           300 |
+--------------------+---------------+---------------+
30 rows in set (0.014 sec)
```

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

select v.venue_name, e.event_type, avg(ticket_price) as average_price from event e join venue v on e.venue_id=v.venue_id group by v.venue_id, e.event_type order by v.venue_id;

```
mysql> select v.venue_name, e.event_type, avg(ticket_price) as average_price
    -> from event e join venue v on e.venue_id = v.venue_id
    -> group by v.venue_id, e.event_type order by v.venue_id;
+--------------------------+------------+-----------------+
| venue_name               | event_type | average_price   |
+--------------------------+------------+-----------------+
| Mayajaal Cinemas         | movie      |     300.000000 |
| SDAT Aquatic Complex     | sports     |    1250.000000 |
| Music Academy            | concert    |    1066.666667 |
| Express Avenue           | movie      |     350.000000 |
| Sathyam Cinemas          | movie      |     450.000000 |
| M. A. Chidambaram Stadium | sports     |    2233.333333 |
| Nehru Indoor Stadium     | concert    |    1800.000000 |
| Phoenix Market City      | concert    |    1600.000000 |
| VGP Universal Kingdom    | concert    |    1100.000000 |
+--------------------------+------------+-----------------+
9 rows in set (0.034 sec)
```

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

select c.customer_id, c.customer_name, sum(b.num_tickets) as tickets_bought from booking b join customer c on b.customer_id = c.customer_id where datediff(curdate(), booking_date)<=30 group by c.customer_id order by c.customer_id;

```
mysql> select c.customer_id, c.customer_name, sum(b.num_tickets) as tickets_bought
    -> from booking b join customer c on b.customer_id = c.customer_id
    -> where datediff(curdate(),booking_date)<= 30 group by c.customer_id
    -> order by c.customer_id;
+-------------+---------------+----------------+
| customer_id | customer_name | tickets_bought |
+-------------+---------------+----------------+
| C001        | Aarav         |             14 |
| C004        | Meena         |              7 |
| C005        | Ravi          |              6 |
| C008        | Neha          |              1 |
| C009        | Amit          |              1 |
| C011        | Tarun         |              2 |
| C012        | Divya         |              8 |
+-------------+---------------+----------------+
7 rows in set (0.019 sec)
```

Task 4:

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

select venue_id, avg(ticket_price) as avg_ticketprice from event where venue_id in (select distinct venue_id from event) group by venue_id order by venue_id;

```
mysql> select venue_id, avg(ticket_price) as avg_ticketprice
    -> from event where venue_id in
    -> (select distinct venue_id from event)
    -> group by venue_id order by venue_id;
+----------+------------------+
| venue_id | avg_ticketprice  |
+----------+------------------+
| V001     |       300.000000 |
| V002     |      1250.000000 |
| V003     |      1066.666667 |
| V004     |       350.000000 |
| V006     |       450.000000 |
| V007     |      2233.333333 |
| V008     |      1800.000000 |
| V009     |      1600.000000 |
| V010     |      1100.000000 |
+----------+------------------+
9 rows in set (0.011 sec)
```

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

*select * from event where event_id in (select event_id from event where (total_seats -available_seats)>(total_seats/2));*

```
mysql> select * from event where event_id in (select event_id from event where (total_seats - available_seats)>(total_seats/2));
+----------+-------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name        | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+-------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E001     | World Cup Final   | 2025-01-28 | 14:00:00   | V007     |       40000 |           15000 |      2500.00 | sports     |
| E002     | Rock Concert      | 2025-05-08 | 18:00:00   | V003     |        3000 |               0 |      1500.00 | concert    |
| E003     | IPL Match         | 2025-05-28 | 14:00:00   | V007     |       38000 |               0 |      2000.00 | sports     |
| E004     | Movie Premiere    | 2025-06-26 | 20:00:00   | V001     |         600 |             150 |       300.00 | movie      |
| E005     | Jazz Night Concert | 2025-09-03 | 14:00:00   | V003     |        1200 |               0 |      1000.00 | concert    |
| E008     | Comedy Night Live | 2025-05-02 | 17:00:00   | V006     |         400 |              35 |       450.00 | movie      |
| E010     | Youth Cup Finals  | 2025-03-11 | 17:00:00   | V002     |        7000 |            1063 |      1200.00 | sports     |
| E011     | Zest of Music     | 2025-09-22 | 15:00:00   | V009     |        3000 |             500 |      1600.00 | concert    |
| E012     | Indie Movie Night | 2025-06-09 | 19:00:00   | V004     |         500 |               0 |       350.00 | movie      |
| E013     | Fantasy Cup Match | 2025-12-18 | 17:00:00   | V007     |       45000 |           20000 |      2200.00 | sports     |
| E015     | Summer Blockbuster | 2025-08-08 | 21:00:00   | V001     |         800 |             250 |       300.00 | movie      |
+----------+-------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
11 rows in set (0.009 sec)
```

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

select e.event_id, e.event_name, tb.total_sold from event e join (select event_id, sum(num_tickets) as total_sold from booking group by event_id) as tb on e.event_id = tb.event_id;

```
mysql> select e.event_id, e.event_name, tb.total_sold from event e
    -> join (select event_id, sum(num_tickets) as total_sold from booking group by event_id)
    -> as tb on e.event_id = tb.event_id;
+----------+--------------------+------------+
| event_id | event_name         | total_sold |
+----------+--------------------+------------+
| E001     | World Cup Final    |          4 |
| E002     | Rock Concert       |         14 |
| E003     | IPL Match          |         13 |
| E004     | Movie Premiere     |         10 |
| E005     | Jazz Night Concert |         11 |
| E006     | Music Fest Chennai |          2 |
| E007     | Aquatic Showdown   |          4 |
| E008     | Comedy Night Live  |          6 |
| E009     | Carnatic Concert   |          9 |
| E010     | Youth Cup Finals   |         11 |
| E011     | Zest of Music      |         10 |
| E012     | Indie Movie Night  |         10 |
| E013     | Fantasy Cup Match  |         10 |
| E015     | Summer Blockbuster |          5 |
+----------+--------------------+------------+
14 rows in set (0.011 sec)
```

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

select customer_id, customer_name from customer where not exists (select customer_id from booking where booking.customer_id = customer.customer_id);

```
mysql> select customer_id, customer_name from customer where not exists
    -> (select customer_id from booking where booking.customer_id = customer.customer_id);
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
| C013        | Kabir         |
| C014        | Leela         |
| C015        | Farhan        |
+-------------+---------------+
3 rows in set (0.009 sec)
```

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

*select * from event where event_id not in (select event_id from booking);*

```
mysql> select * from event where event_id not in (select event_id from booking);
+----------+------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| event_id | event_name       | event_date | event_time | venue_id | total_seats | available_seats | ticket_price | event_type |
+----------+------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
| E014     | VGP Concert Fiesta | 2025-09-06 | 17:00:00   | V010     |        1500 |            1500 |      1100.00 | concert    |
+----------+------------------+------------+------------+----------+-------------+-----------------+--------------+------------+
1 row in set (0.050 sec)
```

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

select e.event_type. sum(t.tickets) as ticket_sold from event e join (select event_id, num_tickets as tickets from booking) as t on e.event_id=t.event_id group by e.event_type;

```
mysql> select e.event_type, sum(t.tickets) as ticket_sold
    -> from event e join (select event_id, num_tickets as tickets from booking) as t
    -> on e.event_id=t.event_id group by e.event_type;
+------------+-------------+
| event_type | ticket_sold |
+------------+-------------+
| sports     |          42 |
| concert    |          46 |
| movie      |          31 |
+------------+-------------+
3 rows in set (0.013 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 |
+----------+--------------------+--------------+
| E001     | World Cup Final    |      2500.00 |
| E002     | Rock Concert       |      1500.00 |
| E003     | IPL Match          |      2000.00 |
| E006     | Music Fest Chennai |      1800.00 |
| E007     | Aquatic Showdown   |      1300.00 |
| E011     | Zest of Music      |      1600.00 |
| E013     | Fantasy Cup Match  |      2200.00 |
+----------+--------------------+--------------+
7 rows in set (0.010 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_revenue 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_revenue
    -> from customer c;
+-------------+---------------+---------------+
| customer_id | customer_name | total_revenue |
+-------------+---------------+---------------+
| C001        | Aarav         |         31200 |
| C002        | Diya          |         11000 |
| C003        | Karan         |         11300 |
| C004        | Meena         |         13300 |
| C005        | Ravi          |          9000 |
| C006        | Sana          |          4800 |
| C007        | Vikram        |          5700 |
| C008        | Neha          |         13000 |
| C009        | Amit          |          6500 |
| C010        | Riya          |         18300 |
| C011        | Tarun         |          8000 |
| C012        | Divya         |         12900 |
| C013        | Kabir         |          NULL |
| C014        | Leela         |          NULL |
| C015        | Farhan        |          NULL |
+-------------+---------------+---------------+
15 rows in set (0.010 sec)
```

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

Select customer_id, customer_name from customer where customer_id in (select b.customer_id from booking b where b.event_id in (select event_id from where venue_id = 'V004'));

```
mysql> select customer_id, customer_name from customer
    -> where customer_id in (select b.customer_id from booking b
    -> where b.event_id in ( select event_id
    -> from event where venue_id = 'V004'));
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
| C003        | Karan         |
| C007        | Vikram        |
+-------------+---------------+
2 rows in set (0.029 sec)
```

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

select e.event_type, sum(t.tickets) as ticket_sold from event e join (select event_id, num_tickets as tickets from booking) as t on e.event_id = t.event_id group by e.event_type;

```
mysql> select e.event_type, sum(t.tickets) as ticket_sold
    -> from event e join (select event_id, num_tickets as tickets from booking) as t
    -> on e.event_id=t.event_id group by e.event_type;
+------------+-------------+
| event_type | ticket_sold |
+------------+-------------+
| sports     |          42 |
| concert    |          46 |
| movie      |          31 |
+------------+-------------+
3 rows in set (0.013 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, b.month from customer c join (select customer_id, date_format(booking,'%b') as month from booking group by month, customer_id) as b on c.customer_id=b.customer_id;

```
mysql> select c.customer_id, c.customer_name, b.month from customer c join (select customer_id, date_format(booking_date,'%b') as month from book
ing group by month, customer_id) as b on c.customer_id = b.customer_id;
+-------------+---------------+-------+
| customer_id | customer_name | month |
+-------------+---------------+-------+
| C001        | Aarav         | Jun   |
| C001        | Aarav         | Apr   |
| C001        | Aarav         | May   |
| C002        | Diya          | Apr   |
| C002        | Diya          | Mar   |
| C003        | Karan         | Apr   |
| C003        | Karan         | May   |
| C004        | Meena         | Apr   |
| C004        | Meena         | Jun   |
| C004        | Meena         | May   |
| C005        | Ravi          | Jun   |
| C006        | Sana          | Apr   |
| C007        | Vikram        | Apr   |
| C008        | Neha          | May   |
| C008        | Neha          | Jun   |
| C009        | Amit          | Jun   |
| C009        | Amit          | Apr   |
| C009        | Amit          | May   |
| C010        | Riya          | Mar   |
| C010        | Riya          | Apr   |
| C011        | Tarun         | May   |
| C011        | Tarun         | Mar   |
| C012        | Divya         | May   |
| C012        | Divya         | Apr   |
| C012        | Divya         | Jun   |
+-------------+---------------+-------+
25 rows in set (0.011 sec)
```

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

select venue_id, avg(ticket_price) as avg_ticketprice from event where venue_id in (select distinct venue_id from event) group by venue_id order by venue_id;

```
mysql> select venue_id, avg(ticket_price) as avg_ticketprice
    -> from event where venue_id in
    -> (select distinct venue_id from event)
    -> group by venue_id order by venue_id;
+----------+-----------------+
| venue_id | avg_ticketprice |
+----------+-----------------+
| V001     |      300.000000 |
| V002     |     1250.000000 |
| V003     |     1066.666667 |
| V004     |      350.000000 |
| V006     |      450.000000 |
| V007     |     2233.333333 |
| V008     |     1800.000000 |
| V009     |     1600.000000 |
| V010     |     1100.000000 |
+----------+-----------------+
9 rows in set (0.011 sec)
```