Assignment - 1 :

Task-1:

1. Create the database named "TicketBookingSystem".

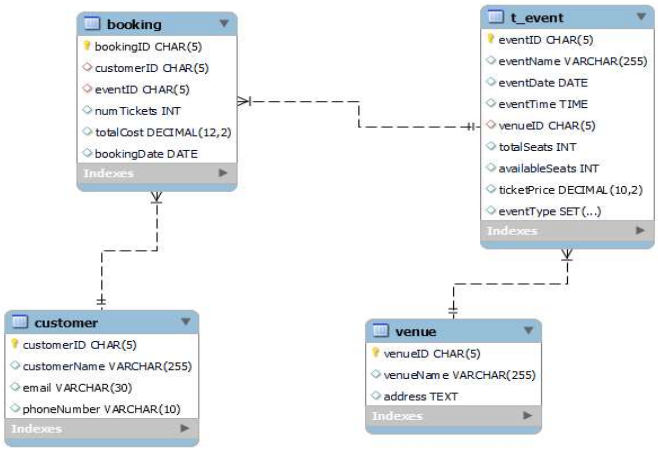
A. create database TicketBookingSystem;

2.

```
create table venue(
    venueID char(5) primary key,
    venueName varchar(255),
    address text
);

create table t_event(
    eventID char(5) primary key,
    eventName varchar(255),
    eventDate date,
    eventTime time,
    venueID char(5),
    totalSeats int,
    availableSeats int,
    ticketPrice decimal(10,2),
    eventType set('Movie', 'Concert', 'Sports'),
    foreign key(venueID) references venue(venueID) on delete cascade on update cascade
);

create table customer(
    customerID char(5) primary key,
    customerName varchar(255),
    email varchar(30) unique,
    phoneNumber varchar(10) unique
);

create table booking(
    bookingID char(5) primary key,
    customerID char(5),
    eventID char(5),
    numTickets int,
    totalCost decimal(12, 2),
    bookingDate date,
    foreign key(customerID) references customer(customerID) on delete cascade on update cascade,
    foreign key(eventID) references t_event(eventID) on delete cascade on update cascade
);
```

✓	250	16:36:42	create table venue(venueID char(5) primary key, venueName varchar(255), address text)	0 row(s) affected
✓	251	16:36:49	create table t_event(eventID char(5) primary key, eventName varchar(255), eventDate date, event...	0 row(s) affected
✓	252	16:36:53	create table customer(customerID char(5) primary key, customerName varchar(255), email varchar(30) ...	0 row(s) affected
✓	253	16:36:57	create table booking(bookingID char(5) primary key, customerID char(5), eventID char(5), numTicke...	0 row(s) affected

3.



Task-2:

1.

```
INSERT INTO venue (venueID, venueName, address) VALUES
('V001', 'City Stadium', '123 Main Street, New York'),
('V002', 'Concert Hall', '456 Oak Avenue, Washington'),
('V003', 'Movieplex Arena', '789 Maple Road, Chicago'),
('V004', 'Sports Arena', '101 Pine Boulevard, Dallas'),
('V005', 'Grand Theater', '202 Cedar Lane, Austin'),
('V006', 'Event Center', '303 Elm Street, New York'),
('V007', 'The Arena', '404 Birch Avenue, Boston'),
('V008', 'Film Palace', '505 Walnut Road, Indianapolis'),
('V009', 'Stadium Square', '606 Pine Street, Dallas'),
('V010', 'Concert Pavilion', '707 Oak Avenue, Austin'),
('V011', 'Sporting Ground', '808 Maple Road, Chicago'),
('V012', 'Cinema Plaza', '909 Cedar Lane, Seattle'),
('V013', 'Music Hall', '1010 Elm Street, New York'),
('V014', 'Theater Square', '1111 Birch Avenue, Chicago'),
('V015', 'Ballgame Park', '1212 Walnut Road, Indianapolis'),
('V016', 'Film Festival Plaza', '1313 Pine Street, Dallas'),
('V017', 'Performance Venue', '1414 Oak Avenue, Washington'),
('V018', 'Game Arena', '1515 Maple Road, Chicago'),
('V019', 'Showcase Center', '1616 Cedar Lane, Austin'),
('V020', 'Entertainment Plaza', '1717 Elm Street, New York');
```

```
INSERT INTO customer (customerID, customerName, email, phoneNumber) VALUES
('C001', 'John Doe', 'john.doe@example.com', '1234567890'),
('C002', 'Jane Smith', 'jane.smith@example.com', '9876541151'),
('C003', 'Bob Johnson', 'bob.johnson@example.com', '5678901515'),
('C004', 'Alice Williams', 'alice.williams@example.com', '8901234567'),
('C005', 'Charlie Brown', 'charlie.brown@example.com', '2345678901'),
('C006', 'Eva Davis', 'eva.davis@example.com', '6789012345'),
('C007', 'Frank Miller', 'frank.miller@example.com', '3456789012'),
('C008', 'Grace Wilson', 'grace.wilson@example.com', '9012345678'),
('C009', 'David Lee', 'david.lee@example.com', '4567890123'),
('C010', 'Sophie Taylor', 'sophie.taylor@example.com', '1230987654'),
('C011', 'Michael Anderson', 'michael.anderson@example.com', '9876543413'),
('C012', 'Emma Martinez', 'emma.martinez@example.com', '5678901234'),
('C013', 'James Wright', 'james.wright@example.com', '8901362567'),
('C014', 'Olivia Brown', 'olivia.brown@example.com', '2342843011'),
('C015', 'Daniel White', 'daniel.white@example.com', '6721162166');
```

```
INSERT INTO t_event (eventID, eventName, eventDate, eventTime, venueID, totalSeats, availableSeats, ticketPrice, eventType) VALUES
('E001', 'Sports Cup 2024', '2024-01-15', '18:00:00', 'V001', 10000, 0, 1500.00, 'Sports'), -- 0 - 1
('E002', 'Concert Spectacular', '2024-02-20', '20:30:00', 'V002', 1500, 1200, 2500.00, 'Concert'),
('E003', 'Movie Night: Blockbuster Marathon', '2024-02-25', '19:00:00', 'V003', 500, 500, 200.00, 'Movie'), -- Changed Price
('E004', 'Football League Cup', '2024-03-10', '16:45:00', 'V004', 20000, 0, 800.00, 'Sports'), -- Changed Price -- 0 - 4
('E005', 'Cinematic Experience: Classics Revisited', '2024-03-12', '18:30:00', 'V005', 800, 700, 300.00, 'Movie'), -- Changed Price
('E006', 'Concert in the Park', '2024-03-18', '21:00:00', 'V013', 3000, 2500, 2800.00, 'Concert'), -- Changed Venue
('E007', 'Basketball Showdown', '2024-03-05', '17:15:00', 'V007', 12000, 0, 2000.00, 'Sports'), -- 0 - 7
('E008', 'Drama Night: Theatrical Delight', '2024-05-22', '19:45:00', 'V008', 600, 450, 100.00, 'Movie'), -- Changed Price
('E009', 'Music Festival Extravaganza', '2024-05-30', '22:00:00', 'V009', 10000, 8000, 1100.00, 'Concert'), -- Changed Price
('E010', 'Soccer Showpiece', '2024-06-05', '15:30:00', 'V010', 25000, 0, 2200.00, 'Sports'), -- 0 - 10
('E011', 'Classic Film Marathon', '2024-06-12', '20:15:00', 'V011', 700, 600, 1300.00, 'Movie'),
('E012', 'Rock Concert Blast', '2024-07-18', '21:30:00', 'V012', 4000, 3500, 2700.00, 'Concert'),
('E013', 'Hockey Cup', '2024-08-12', '18:45:00', 'V011', 18000, 0, 1500.00, 'Sports'), -- Changed Price, Venue -- 0 - 13
('E014', 'Film Noir Night', '2024-08-30', '19:30:00', 'V014', 900, 800, 1100.00, 'Movie'),
('E015', 'Symphony Orchestra Showcase', '2024-09-15', '20:00:00', 'V013', 1200, 1000, 2600.00, 'Concert'), -- Changed Venue
('E016', 'Baseball Championship', '2024-10-02', '16:00:00', 'V019', 22000, 0, 2400.00, 'Sports'), -- Changed Venue -- 0 - 16
('E017', 'Science Fiction Film Festival', '2024-10-22', '18:00:00', 'V016', 1000, 900, 400.00, 'Movie'), -- Changed Venue, Price
('E018', 'Jazz Night: Smooth Sounds', '2024-11-28', '21:15:00', 'V016', 5000, 4500, 2900.00, 'Concert'), -- Changed Venue
('E019', 'Tennis Championship', '2024-12-05', '17:30:00', 'V019', 28000, 0, 2700.00, 'Sports'), -- 0 - 19
('E020', 'Zero Night Cocert', '2024-12-31', '22:00:00', 'V010', 25000, 5000, 1500.00, 'Concert'); -- Changed Venue
```

```sql
-- CHECK BOOKIN DATE
INSERT INTO booking (bookingID, customerID, eventID, numTickets, totalCost, bookingDate) VALUES
('B001', 'C007', 'E002', 2, 5000.00, '2024-02-10'), -- D
('B002', 'C001', 'E002', 3, 7500.00, '2024-02-18'), -- D
('B003', 'C005', 'E003', 2, 400.00, '2024-02-20'),
('B004', 'C002', 'E006', 2, 5600.00, '2024-03-01'), -- D
('B005', 'C008', 'E007', 6, 12000.00, '2024-03-01'), -- D
('B006', 'C012', 'E004', 7, 5600.00, '2024-03-05'), -- D
('B007', 'C001', 'E005', 4, 1200.00, '2024-03-10'),
('B008', 'C008', 'E008', 1, 100.00, '2024-05-18'),
('B009', 'C003', 'E010', 5, 11000.00, '2024-05-25'), -- D
('B010', 'C013', 'E009', 4, 4400.00, '2024-05-25'), -- D
('B011', 'C015', 'E011', 5, 6500.00, '2024-06-08'), -- D
('B012', 'C009', 'E012', 2, 5400.00, '2024-07-10'), -- D
('B013', 'C004', 'E013', 4, 6000.00, '2024-07-12'), -- D
('B014', 'C014', 'E014', 2, 2200.00, '2024-08-21'), -- D
('B015', 'C010', 'E016', 8, 19200.00, '2024-09-10'), -- D
('B016', 'C005', 'E015', 1, 2600.00, '2024-09-11'), -- D
('B017', 'C012', 'E017', 3, 1200.00, '2024-10-21'),
('B018', 'C006', 'E018', 3, 8700.00, '2024-11-25'), -- D
('B019', 'C011', 'E020', 3, 4500.00, '2024-11-25'), -- D
('B020', 'C010', 'E019', 6, 16200.00, '2024-12-01'), -- D
('B021', 'C007', 'E020', 5, 7500.00, '2024-12-15'); -- D
```

```
254 16:50:00 INSERT INTO venue (venueID, venueName, address) VALUES ('V001', 'City Stadium', '123 Main Street, New ... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0 0.000 sec
255 17:14:49 INSERT INTO t_event (eventID, eventName, eventDate, eventTime, venueID, totalSeats, availableSeats, tic... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0 0.000 sec

261 19:05:31 INSERT INTO customer (customerID, customerName, email, phoneNumber) VALUES ('C001', 'John Doe', 'john... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.000 sec
262 19:05:41 INSERT INTO booking (bookingID, customerID, eventID, numTickets, totalCost, bookingDate) VALUES ('B00... 21 row(s) affected Records: 21 Duplicates: 0 Warnings: 0 0.000 sec
```

2.

```sql
-- Write a SQL query to list all Events.
select * from t_event;
```

eventID	eventName	eventDate	eventTime	venueID	totalSeats	availableSeats	ticketPrice	eventType
E001	Sports Cup 2024	2024-01-15	18:00:00	V001	10000	0	1500.00	Sports
E002	Concert Spectacular	2024-02-20	20:30:00	V002	1500	1200	2500.00	Concert
E003	Movie Night: Blockbuster Marathon	2024-02-25	19:00:00	V003	500	500	200.00	Movie
E004	Football League Cup	2024-03-10	16:45:00	V004	20000	0	800.00	Sports
E005	Cinematic Experience: Classics Revisited	2024-03-12	18:30:00	V005	800	700	300.00	Movie
E006	Concert in the Park	2024-03-18	21:00:00	V013	3000	2500	2800.00	Concert
E007	Basketball Showdown	2024-03-05	17:15:00	V007	12000	0	2000.00	Sports
E008	Drama Night: Theatrical Delight	2024-05-22	19:45:00	V008	600	450	100.00	Movie
E009	Music Festival Extravaganza	2024-05-30	22:00:00	V009	10000	8000	1100.00	Concert
E010	Soccer Showpiece	2024-06-05	15:30:00	V010	25000	0	2200.00	Sports
E011	Classic Film Marathon	2024-06-12	20:15:00	V011	700	600	1300.00	Movie
E012	Rock Concert Blast	2024-07-18	21:30:00	V012	4000	3500	2700.00	Concert
E013	Hockey Cup	2024-08-12	18:45:00	V011	18000	0	1500.00	Sports
E014	Film Noir Night	2024-08-30	19:30:00	V014	900	800	1100.00	Movie
E015	Symphony Orchestra Showcase	2024-09-15	20:00:00	V013	1200	1000	2600.00	Concert
E016	Baseball Championship	2024-10-02	16:00:00	V019	22000	0	2400.00	Sports
E017	Science Fiction Film Festival	2024-10-22	18:00:00	V016	1000	900	400.00	Movie
E018	Jazz Night: Smooth Sounds	2024-11-28	21:15:00	V016	5000	4500	2900.00	Concert
E019	Tennis Championship	2024-12-05	17:30:00	V019	28000	0	2700.00	Sports
E020	Zero Night Cocert	2024-12-31	22:00:00	V010	25000	5000	1500.00	Concert
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.

```sql
-- Write a SQL query to select events with available tickets.
select *
from t_event
where availableSeats>0;
```

eventID	eventName	eventDate	eventTime	venueID	totalSeats	availableSeats	ticketPrice	eventType
E002	Concert Spectacular	2024-02-20	20:30:00	V002	1500	1200	2500.00	Concert
E003	Movie Night: Blockbuster Marathon	2024-02-25	19:00:00	V003	500	500	200.00	Movie
E005	Cinematic Experience: Classics Revisited	2024-03-12	18:30:00	V005	800	700	300.00	Movie
E006	Concert in the Park	2024-03-18	21:00:00	V013	3000	2500	2800.00	Concert
E008	Drama Night: Theatrical Delight	2024-05-22	19:45:00	V008	600	450	100.00	Movie
E009	Music Festival Extravaganza	2024-05-30	22:00:00	V009	10000	8000	1100.00	Concert
E011	Classic Film Marathon	2024-06-12	20:15:00	V011	700	600	1300.00	Movie
E012	Rock Concert Blast	2024-07-18	21:30:00	V012	4000	3500	2700.00	Concert
E014	Film Noir Night	2024-08-30	19:30:00	V014	900	800	1100.00	Movie
E015	Symphony Orchestra Showcase	2024-09-15	20:00:00	V013	1200	1000	2600.00	Concert
E017	Science Fiction Film Festival	2024-10-22	18:00:00	V016	1000	900	400.00	Movie
E018	Jazz Night: Smooth Sounds	2024-11-28	21:15:00	V016	5000	4500	2900.00	Concert
E020	Zero Night Cocert	2024-12-31	22:00:00	V010	25000	5000	1500.00	Concert
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4.

```sql
-- Write a SQL query to select events name partial match with ‘cup’.
select *
from t_event
where eventName like '%cup%';
```

eventID	eventName	eventDate	eventTime	venueID	totalSeats	availableSeats	ticketPrice	eventType
E001	Sports Cup 2024	2024-01-15	18:00:00	V001	10000	0	1500.00	Sports
E004	Football League Cup	2024-03-10	16:45:00	V004	20000	0	800.00	Sports
E013	Hockey Cup	2024-08-12	18:45:00	V011	18000	0	1500.00	Sports
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5.

```sql
-- Write a SQL query to select events with ticket price range is between 1000 to 2500.
select eventID, eventName, ticketPrice, eventType
from t_event
where ticketPrice between 1000 and 2500;
```

eventID	eventName	ticketPrice	eventType
E001	Sports Cup 2024	1500.00	Sports
E002	Concert Spectacular	2500.00	Concert
E007	Basketball Showdown	2000.00	Sports
E009	Music Festival Extravaganza	1100.00	Concert
E010	Soccer Showpiece	2200.00	Sports
E011	Classic Film Marathon	1300.00	Movie
E013	Hockey Cup	1500.00	Sports
E014	Film Noir Night	1100.00	Movie
E016	Baseball Championship	2400.00	Sports
E020	Zero Night Cocert	1500.00	Concert
NULL	NULL	NULL	NULL

6.

```sql
-- Write a SQL query to retrieve events with dates falling within a specific range.
select eventID, eventName, eventDate, eventType
from t_event
where eventDate between '2024-04-01' and '2024-07-31';
```

eventID	eventName	eventDate	eventType
E008	Drama Night: Theatrical Delight	2024-05-22	Movie
E009	Music Festival Extravaganza	2024-05-30	Concert
E010	Soccer Showpiece	2024-06-05	Sports
E011	Classic Film Marathon	2024-06-12	Movie
E012	Rock Concert Blast	2024-07-18	Concert
NULL	NULL	NULL	NULL

7.

```sql
-- Write a SQL query to retrieve events with available tickets that also have "Concert" in their name.
select eventID, eventName, availableSeats as `Available Tickets`, eventType
from t_event
where eventName like '%Concert%';
```

	eventID	eventName	Available Tickets	eventType
▶	E002	Concert Spectacular	1200	Concert
	E006	Concert in the Park	2500	Concert
	E012	Rock Concert Blast	3500	Concert
*	NULL	NULL	NULL	NULL

8.

```
-- Write a SQL query to retrieve users in batches of 5, starting from the 6th user.
select *
from customer
order by customerID
limit 5
offset 5;
```

	customerID	customerName	email	phoneNumber
▶	C006	Eva Davis	eva.davis@example.com	6789012345
	C007	Frank Miller	frank.miller@example.com	3456789012
	C008	Grace Wilson	grace.wilson@example.com	9012345678
	C009	David Lee	david.lee@example.com	4567890123
	C010	Sophie Taylor	sophie.taylor@example.com	1230987654
*	NULL	NULL	NULL	NULL

9.

```
-- Write a SQL query to retrieve bookings details contains booked no of ticket more than 4.
select *
from booking
where numTickets>4;
```

	bookingID	customerID	eventID	numTickets	totalCost	bookingDate
▶	B005	C008	E007	6	12000.00	2024-03-01
	B006	C012	E004	7	5600.00	2024-03-05
	B009	C003	E010	5	11000.00	2024-05-25
	B011	C015	E011	5	6500.00	2024-06-08
	B015	C010	E016	8	19200.00	2024-09-10
	B020	C010	E019	6	16200.00	2024-12-01
	B021	C007	E020	5	7500.00	2024-12-15
*	NULL	NULL	NULL	NULL	NULL	NULL

10.

```
-- Write a SQL query to retrieve customer information whose phone number end with '000'
select *
from customer
where phoneNumber like '%000';
```

	customerID	customerName	email	phoneNumber
▶	C001	John Doe	john.doe@example.com	8358328000
	C010	Sophie Taylor	sophie.taylor@example.com	7583128000
*	NULL	NULL	NULL	NULL

11.

```sql
-- Write a SQL query to retrieve the events in order whose seat capacity more than 15000.
select eventID, eventName, totalSeats as Capacity, eventType
from t_event
where totalSeats>15000
order by totalSeats desc;
```

eventID	eventName	Capacity	eventType
E019	Tennis Championship	28000	Sports
E010	Soccer Showpiece	25000	Sports
E020	Zero Night Cocert	25000	Concert
E016	Baseball Championship	22000	Sports
E004	Football League Cup	20000	Sports
E013	Hockey Cup	18000	Sports
NULL	NULL	NULL	NULL

12.

```sql
-- Write a SQL query to select events name not start with ‘x’, ‘y’, ‘z’
select eventID, eventName, eventType
from t_event
where eventName not like '[xyz]%';
```

eventID	eventName	eventType
E001	Sports Cup 2024	Sports
E002	Concert Spectacular	Concert
E003	Movie Night: Blockbuster Marathon	Movie
E004	Football League Cup	Sports
E005	Cinematic Experience: Classics Revisited	Movie
E006	Concert in the Park	Concert
E007	Basketball Showdown	Sports
E008	Drama Night: Theatrical Delight	Movie
E009	Music Festival Extravaganza	Concert
E010	Soccer Showpiece	Sports
E011	Classic Film Marathon	Movie
E012	Rock Concert Blast	Concert
E013	Hockey Cup	Sports
E014	Film Noir Night	Movie

Task-3:

1.

```sql
-- Write a SQL query to List Events and Their Average Ticket Prices.
select eventID, eventName, round(avg(ticketPrice), 2) as `Average Ticket Price`
from t_event
group by eventID, eventname;
```

eventID	eventName	Average Ticket Price
E001	Sports Cup 2024	1500.00
E002	Concert Spectacular	2500.00
E003	Movie Night: Blockbuster Marathon	200.00
E004	Football League Cup	800.00
E005	Cinematic Experience: Classics Revisited	300.00
E006	Concert in the Park	2800.00
E007	Basketball Showdown	2000.00
E008	Drama Night: Theatrical Delight	100.00
E009	Music Festival Extravaganza	1100.00
E010	Soccer Showpiece	2200.00
E011	Classic Film Marathon	1300.00
E012	Rock Concert Blast	2700.00
E013	Hockey Cup	1500.00
E014	Film Noir Night	1100.00
E015	Symphony Orchestra Showcase	2600.00
E016	Baseball Championship	2400.00
E017	Science Fiction Film Festival	400.00
E018	Jazz Night: Smooth Sounds	2900.00

2.

```sql
-- Write a SQL query to Calculate the Total Revenue Generated by Events.
select eventName, SUM(((totalSeats-availableSeats)*ticketPrice)) as `Total Revenue`
from t_event
group by eventName
order by `Total Revenue` desc;
```

eventName	Total Revenue
Tennis Championship	75600000.00
Soccer Showpiece	55000000.00
Baseball Championship	52800000.00
Zero Night Cocert	30000000.00
Hockey Cup	27000000.00
Basketball Showdown	24000000.00
Football League Cup	16000000.00
Sports Cup 2024	15000000.00
Music Festival Extravaganza	2200000.00
Jazz Night: Smooth Sounds	1450000.00
Concert in the Park	1400000.00
Rock Concert Blast	1350000.00
Concert Spectacular	750000.00
Symphony Orchestra Show...	520000.00
Classic Film Marathon	130000.00
Film Noir Night	110000.00
Science Fiction Film Festival	40000.00
Cinematic Experience: Clas...	30000.00
Drama Night: Theatrical Deli...	15000.00
Movie Night: Blockbuster M...	0.00

3.

```sql
-- Write a SQL query to find the event with the highest ticket sales.
select b.eventID, e.eventName, sum(numTickets) as total_tickets
from booking b
join t_event e on b.eventID = e.eventID
group by b.eventID
having total_tickets = (
                        select max(total_tickets)
                        from (select eventID, sum(numTickets) as total_tickets
                                from booking
                                group by eventID) x);
```

	eventID	eventName	total_tickets
▶	E016	Baseball Championship	8
	E020	Zero Night Cocert	8

4.

```sql
-- Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event.
select eventID, sum(numTickets) as total_tickets
from booking
group by eventID;
```

	eventID	total_tickets
▶	E002	5
	E003	2
	E004	7
	E005	4
	E006	2
	E007	6
	E008	1
	E009	4
	E010	5
	E011	5
	E012	2
	E013	4
	E014	2
	E015	1
	E016	8
	E017	3
	E018	3
	E019	6
	E020	8

5.

```sql
-- Write a SQL query to Find Events with No Ticket Sales.
select e.*
from t_event e
left join booking b on e.eventID = b.eventID
where b.eventID is null;
```

	eventID	eventName	eventDate	eventTime	venueID	totalSeats	availableSeats	ticketPrice	eventType
▶	E001	Sports Cup 2024	2024-01-15	18:00:00	V001	10000	0	1500.00	Sports

6.

```sql
-- Write a SQL query to Find the User Who Has Booked the Most Tickets.
select c.*
from customer c
join booking b on c.customerID = b.customerID
group by b.customerID
having sum(numTickets) = (
        select max(total_tickets)
        from(
            select customerID, sum(numTickets) as total_tickets
            from booking
            group by customerID) a);
```

	customerID	customerName	email	phoneNumber	total_tickets
▶	C010	Sophie Taylor	sophie.taylor@example.com	7583128000	14

7.

```sql
-- Write a SQL query to List Events and the total number of tickets sold for each month.
select b.eventID,
       e.eventName,
       e.eventDate,
       monthname (bookingDate) as `Booking Month`,
       sum(numTickets) as `Total Tickets`
from booking b
join t_event e on b.eventId = e.eventId
group by eventID, `Booking Month`;
```

eventID	eventName	eventDate	Booking Month	Total Tickets
E002	Concert Spectacular	2024-02-20	February	5
E003	Movie Night: Blockbuster Marathon	2024-02-25	February	2
E006	Concert in the Park	2024-03-18	March	2
E007	Basketball Showdown	2024-03-05	March	6
E004	Football League Cup	2024-03-10	March	7
E005	Cinematic Experience: Classics Revisited	2024-03-12	March	4
E008	Drama Night: Theatrical Delight	2024-05-22	May	1
E010	Soccer Showpiece	2024-06-05	May	5
E009	Music Festival Extravaganza	2024-05-30	May	4
E011	Classic Film Marathon	2024-06-12	June	5
E012	Rock Concert Blast	2024-07-18	July	2
E013	Hockey Cup	2024-08-12	July	4
E014	Film Noir Night	2024-08-30	August	2
E016	Baseball Championship	2024-10-02	September	8
E015	Symphony Orchestra Showcase	2024-09-15	September	1
E017	Science Fiction Film Festival	2024-10-22	October	3
E018	Jazz Night: Smooth Sounds	2024-11-28	November	3
E020	Zero Night Cocert	2024-12-31	November	3
E019	Tennis Championship	2024-12-05	December	6
E020	Zero Night Cocert	2024-12-31	December	5

8.

```sql
-- Write a SQL query to calculate the average Ticket Price for Events in Each Venue.
select venueID, round(avg(ticketPrice), 2)
from t_event
group by venueID;
```

venueID	round(avg(ticketPrice), 2)
V001	1500.00
V002	2500.00
V003	200.00
V004	800.00
V005	300.00
V007	2000.00
V008	100.00
V009	1100.00
V010	1850.00
V011	1400.00
V012	2700.00
V013	2700.00
V014	1100.00
V016	1650.00
V019	2550.00

9.

```sql
-- Write a SQL query to calculate the total Number of Tickets Sold for Each Event Type.
select eventType, sum(totalSeats-availableSeats) as `Tickets Sold`
from t_event
group by eventType;
```

	eventType	Tickets Sold
▶	Sports	135000
	Concert	24000
	Movie	550

10.

```sql
-- Write a SQL query to calculate the total Revenue Generated by Events in Each Year.
select eventName, Year(eventDate), SUM((((totalSeats + availableSeats)-availableSeats)*ticketPrice)) as `Total Revenue`
from t_event
group by eventName, Year(eventDate)
order by `Total Revenue` desc;
```

	eventName	Year(eventDate)	Total Revenue
▶	Tennis Championship	2024	75600000.00
	Soccer Showpiece	2024	55000000.00
	Baseball Championship	2024	52800000.00
	Zero Night Cocert	2024	37500000.00
	Hockey Cup	2024	27000000.00
	Basketball Showdown	2024	24000000.00
	Football League Cup	2024	16000000.00
	Sports Cup 2024	2024	15000000.00
	Jazz Night: Smooth Sounds	2024	14500000.00
	Music Festival Extravaganza	2024	11000000.00
	Rock Concert Blast	2024	10800000.00
	Concert in the Park	2024	8400000.00
	Concert Spectacular	2024	3750000.00
	Symphony Orchestra Show...	2024	3120000.00
	Film Noir Night	2024	990000.00
	Classic Film Marathon	2024	910000.00
	Science Fiction Film Festival	2024	400000.00
	Cinematic Experience: Clas...	2024	240000.00
	Movie Night: Blockbuster M...	2024	100000.00
	Drama Night: Theatrical Deli...	2024	60000.00

11.

```sql
-- Write a SQL query to list users who have booked tickets for multiple events.
select b.customerID, c.customerName, count(distinct b.eventID) as Events_Booked
from booking b
join customer c on b.customerID = c.customerID
group by b.customerID
having count(distinct b.eventID)>1;
```

	customerID	customerName	Events_Booked
▶	C001	John Doe	2
	C005	Charlie Brown	2
	C007	Frank Miller	2
	C008	Grace Wilson	2
	C010	Sophie Taylor	2
	C012	Emma Martinez	2

12.

```sql
-- Write a SQL query to calculate the Total Revenue Generated by Events for Each User.
select customerID, eventID, sum(totalCost) as `Total Revenue`
from booking
group by customerID, eventID
order by `Total Revenue` desc;
```

customerID	eventID	Total Revenue
C010	E016	19200.00
C010	E019	16200.00
C008	E007	12000.00
C003	E010	11000.00
C006	E018	8700.00
C001	E002	7500.00
C007	E020	7500.00
C015	E011	6500.00
C004	E013	6000.00
C002	E006	5600.00
C012	E004	5600.00
C009	E012	5400.00
C007	E002	5000.00
C011	E020	4500.00
C013	E009	4400.00
C005	E015	2600.00
C014	E014	2200.00
C001	E005	1200.00
C012	E017	1200.00
C005	E003	400.00
C008	E008	100.00

13.

```sql
-- Write a SQL query to calculate the Average Ticket Price for Events in Each Category and Venue.
select eventType, venueID, round(avg(TicketPrice), 2) as `Average Ticket Price`
from t_event
group by eventType, venueID
order by eventType;
```

eventType	venueID	Average Ticket Price
Movie	V003	200.00
Movie	V005	300.00
Movie	V008	100.00
Movie	V011	1300.00
Movie	V014	1100.00
Movie	V016	400.00
Concert	V002	2500.00
Concert	V009	1100.00
Concert	V010	1500.00
Concert	V012	2700.00
Concert	V013	2700.00
Concert	V016	2900.00
Sports	V001	1500.00
Sports	V004	800.00
Sports	V007	2000.00
Sports	V010	2200.00
Sports	V011	1500.00
Sports	V019	2550.00

14.

```sql
-- Write a SQL query to list Users and the Total Number of Tickets They've Purchased in the Last 30 Days.
select customerID, sum(numTickets) as `Total Tickets`
from booking
where bookingDate between ('2024-03-31' - interval 30 day) and '2024-03-31'
group by customerID;
```

customerID	Total Tickets
C001	4
C002	2
C008	6
C012	7

Task-4:

1.

```sql
-- Calculate the Average Ticket Price for Events in Each Venue Using a Subquery.
select venueID, avg(ticketPrice)
from (select venueId, ticketPrice from t_event) a
group by venueID;
```

venueId	avg(ticketPrice)
V001	1500.000000
V002	2500.000000
V003	200.000000
V004	800.000000
V005	300.000000
V007	2000.000000
V008	100.000000
V009	1100.000000
V010	1850.000000
V011	1400.000000
V012	2700.000000
V013	2700.000000
V014	1100.000000
V016	1650.000000

2.

```sql
-- Find Events with More Than 50% of Tickets Sold using subquery.
select eventID
from t_event
where eventID in (
                select eventID
                from t_event
                where availableSeats = 0 OR (totalSeats-availableSeats)>availableSeats
                );
```

eventID
E001
E004
E007
E010
E013
E016
E019
E020
NULL

3.

```
-- Calculate the Total Number of Tickets Sold for Each Event.
select eventID, totalSeats-availableSeats as `Tickets Sold`
from t_event
order by `Tickets Sold`;
```

eventID	Tickets Sold
E003	0
E005	100
E011	100
E014	100
E017	100
E008	150
E015	200
E002	300
E006	500
E012	500
E018	500
E009	2000
E001	10000
E007	12000
E013	18000
E004	20000
E020	20000
E016	22000
E010	25000
E019	28000

4.

```
-- Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery.
 select *
 from customer c
 where NOT EXISTS(Select distinct customerID -- distinct customerID is a placeholder
                  from booking b
                  where b.customerID = c.customerID);
```

customerID	customerName	email	phoneNumber
C016	Danny White	White@holdingbusiness.com	8374839215
NULL	NULL	NULL	NULL

5.

```
-- List Events with No Ticket Sales Using a NOT IN Subquery
select *
from t_event
where eventID NOT IN (select eventID from t_event where totalSeats!=availableSeats);
```

eventID	eventName	eventDate	eventTime	venueID	totalSeats	availableSeats	ticketPrice	eventType
E003	Movie Night: Blockbuster Marathon	2024-02-25	19:00:00	V003	500	500	200.00	Movie
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6.

```
-- Calculate the Total Number of Tickets Sold for Each Event Type
-- Using a Subquery in the FROM Clause.
select ev.eventType, sum(ev.totalSeats-ev.availableSeats) as `Ticket Sold`
from (
    select eventType, totalSeats, availableSeats
    from t_event) ev
group by ev.eventType;
```

	eventType	Ticket Sold
▶	Sports	135000
	Concert	24000
	Movie	550

7.

```
-- Find Events with Ticket Prices Higher Than the
-- Average Ticket Price Using a Subquery in the WHERE Clause.
select eventID, eventName, ticketPrice
from t_event
where ticketPrice > (select avg(ticketPrice)
                     from t_event);
```

	eventID	eventName	ticketPrice
▶	E002	Concert Spectacular	2500.00
	E006	Concert in the Park	2800.00
	E007	Basketball Showdown	2000.00
	E010	Soccer Showpiece	2200.00
	E012	Rock Concert Blast	2700.00
	E015	Symphony Orchestra Showcase	2600.00
	E016	Baseball Championship	2400.00
	E018	Jazz Night: Smooth Sounds	2900.00
	E019	Tennis Championship	2700.00
*	NULL	NULL	NULL

8.

```
-- Calculate the Total Revenue Generated by Events for Each User Using a Correlated Subquery.
select c.customerID,
       c.customerName,
       coalesce((
            select SUM(totalCost)
            from booking b
            where b.customerID = c.customerID
        ), 0) as `Total Revenue`
from customer c;
```

customerID	customerName	Total Revenue
C001	John Doe	8700.00
C002	Jane Smith	5600.00
C003	Bob Johnson	11000.00
C004	Alice Williams	6000.00
C005	Charlie Brown	3000.00
C006	Eva Davis	8700.00
C007	Frank Miller	12500.00
C008	Grace Wilson	12100.00
C009	David Lee	5400.00
C010	Sophie Taylor	35400.00
C011	Michael Ander...	4500.00
C012	Emma Martinez	6800.00
C013	James Wright	4400.00
C014	Olivia Brown	2200.00
C015	Daniel White	6500.00
C016	Danny White	0.00

9.

```sql
-- List Users Who Have Booked Tickets for Events in a
-- Given Venue Using a Subquery in the WHERE Clause.
select b.customerID, c.customerName
from booking b
join customer c on b.customerID = c.customerID
where eventID in (
                select eventID
                from t_event e
                where e.eventID = b.eventID and e.venueID = 'V010');
```

customerID	customerName
C003	Bob Johnson
C011	Michael Anderson
C007	Frank Miller

10.

```sql
-- Calculate the Total Number of Tickets Sold for
-- Each Event Category Using a Subquery with GROUP BY.
select ev.eventType, `Ticket Sold`
from (
    select eventType, sum(totalSeats-availableSeats) as `Ticket Sold`
    from t_event
    group by eventType) ev;
```

	eventType	Ticket Sold
▶	Sports	135000
	Concert	24000
	Movie	550

12.

```
-- Calculate the Average Ticket Price for Events in Each Venue Using a Subquery
select v.venueID, venueName, coalesce(ven.avg_price, 0) as `Average Price`
from venue v
left join (select venueID, avg(TicketPrice) as avg_price
        from t_event
        group by venueID) ven on v.venueID = ven.venueID;
```

	venueID	venueName	Average Price
▶	V001	City Stadium	1500.000000
	V002	Concert Hall	2500.000000
	V003	Movieplex Arena	200.000000
	V004	Sports Arena	800.000000
	V005	Grand Theater	300.000000
	V006	Event Center	0.000000
	V007	The Arena	2000.000000
	V008	Film Palace	100.000000
	V009	Stadium Square	1100.000000
	V010	Concert Pavilion	1850.000000
	V011	Sporting Ground	1400.000000
	V012	Cinema Plaza	2700.000000
	V013	Music Hall	2700.000000
	V014	Theater Square	1100.000000
	V015	Ballgame Park	0.000000
	V016	Film Festival Plaza	1650.000000
	V017	Performance Ve...	0.000000
	V018	Game Arena	0.000000
	V019	Showcase Center	2550.000000
	V020	Entertainment P...	0.000000