

```
create database ticketbookingsystem;

use ticketbookingsystem;

SET @OLD_FOREIGN_KEY_CHECKS = @@FOREIGN_KEY_CHECKS;

SET FOREIGN_KEY_CHECKS = 0;

create table venue(
venue_id int primary key,
venue_name varchar(100),
address varchar(100)
);

create table event(
event_id int primary key,
event_name varchar(100),
event_date date,
event_time time,
venue_id int null ,
foreign key(venue_id) references venue(venue_id),
total_seats int,
available_Seats int,
ticket_price int,
event_type varchar(100),
booking_id int null,
foreign key(booking_id) references booking(booking_id)
);

create table customer(
customer_id int primary key,
customer_name varchar(100),
email varchar(100),
phone_number int,
booking_id int,
foreign key(booking_id) references booking(booking_id)
);
```

```

create table booking(
booking_id int primary key,
customer_id int null,
foreign key(customer_id) references customer(customer_id),
num_tickets int,
total_cost int,
booking_date date
);

-- 1.

-- Insert into the "venue" table
INSERT INTO venue (venue_id, venue_name, address)
VALUES
(1, 'Concert Hall', '123 Main Street'),
(2, 'Conference Center', '456 Oak Avenue'),
(3, 'Sports Stadium', '789 Maple Lane'),
(4, 'Comedy Club', '101 Pine Road'),
(5, 'Art Gallery', '202 Artist Street');

-- Insert into the "event" table
INSERT INTO event (event_id, event_name, event_date, event_time, venue_id, total_seats,
available_seats, ticket_price, event_type, booking_id)
VALUES
(101, 'Music Festival', '2023-12-15', '19:00:00', 1, 2000, 1500, 50, 'Music', 201),
(102, 'Tech Summit', '2023-11-25', '10:00:00', 2, 500, 300, 100, 'Conference', 202),
(103, 'Football Match', '2023-11-30', '15:30:00', 3, 10000, 8000, 30, 'Sports', NULL),
(104, 'Stand-up Comedy Show', '2023-12-05', '20:30:00', 4, 800, 400, 25, 'Comedy', NULL),
(105, 'Art Exhibition', '2023-11-20', '11:00:00', 5, 300, 200, 20, 'Exhibition', 203);

-- Insert into the "customer" table
INSERT INTO customer (customer_id, customer_name, email, phone_number, booking_id)
VALUES

```

```
(501, 'Alice Johnson', 'alice.johnson@email.com', 11122233, 301),  
(502, 'Bob Smith', 'bob.smith@email.com', 44455566, 302),  
(503, 'Charlie Davis', 'charlie.davis@email.com', 77788899, 303),  
(504, 'David Brown', 'david.brown@email.com', 12345678, NULL),  
(505, 'Eva White', 'eva.white@email.com', 98765432, NULL);
```

```
-- Insert into the "booking" table
```

```
INSERT INTO booking (booking_id, customer_id, num_tickets, total_cost, booking_date)  
VALUES  
(201, 501, 3, 150, '2023-11-22'),  
(202, 502, 2, 200, '2023-11-10'),  
(203, 503, 1, 20, '2023-11-18'),  
(204, NULL, 5, 250, '2023-12-01'),  
(205, NULL, 4, 100, '2023-11-15');
```

```
select * from customer;
```

```
-- 2.
```

```
SELECT * FROM event;
```

```
-- 3.
```

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

```
-- 4.
```

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

```
-- 5.
```

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

```
-- 6.
```

```
SELECT * FROM event WHERE event_date BETWEEN '2023-11-01' AND '2023-12-31';
```

-- 7.

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

-- 8.

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

-- 9.

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

-- 10.

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

-- 11.

```
SELECT * FROM event WHERE total_seats > 15000 ORDER BY total_seats;
```

-- 12.

```
SELECT * FROM event WHERE NOT (event_name LIKE 'x%' OR event_name LIKE 'y%' OR event_name  
LIKE 'z%');
```

-- -----AGGREGATE FUNCTIONS, GROUP BY , AND JOIN-----

-- 1

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

-- 2

```
SELECT event_id, event_name, SUM(total_cost) AS total_revenue  
FROM booking  
JOIN event ON booking.event_id = event.event_id
```

```
GROUP BY event_id, event_name;
```

```
-- 3
```

```
SELECT event_id, event_name, SUM(num_tickets) AS total_tickets_sold  
FROM booking  
JOIN event ON booking.event_id = event.event_id  
GROUP BY event_id, event_name  
ORDER BY total_tickets_sold DESC  
LIMIT 1;
```

```
-- 4
```

```
SELECT event_id, event_name, SUM(num_tickets) AS total_tickets_sold  
FROM booking  
JOIN event ON booking.event_id = event.event_id  
GROUP BY event_id, event_name;
```

```
-- 5
```

```
SELECT event_id, event_name  
FROM event  
WHERE event_id NOT IN (SELECT DISTINCT event_id FROM booking);
```

```
-- 6
```

```
SELECT customer_id, customer_name, SUM(num_tickets) AS total_tickets_purchased  
FROM booking  
JOIN customer ON booking.customer_id = customer.customer_id  
GROUP BY customer_id, customer_name  
ORDER BY total_tickets_purchased DESC  
LIMIT 1;
```

```
-- 7
```

```
SELECT EXTRACT(MONTH FROM booking_date) AS month, event_id, event_name, SUM(num_tickets)
AS total_tickets_sold
```

```
FROM booking
```

```
JOIN event ON booking.event_id = event.event_id
```

```
GROUP BY month, event_id, event_name;
```

```
-- 8
```

```
SELECT venue_id, AVG(ticket_price) AS average_ticket_price
```

```
FROM event
```

```
GROUP BY venue_id;
```

```
-- 9
```

```
SELECT event_type, SUM(num_tickets) AS total_tickets_sold
```

```
FROM booking
```

```
JOIN event ON booking.event_id = event.event_id
```

```
GROUP BY event_type;
```

```
-- 10
```

```
SELECT EXTRACT(YEAR FROM booking_date) AS year, SUM(total_cost) AS total_revenue
```

```
FROM booking
```

```
GROUP BY year;
```

```
-- 11
```

```
SELECT customer_id, customer_name
```

```
FROM customer
```

```
WHERE customer_id IN (SELECT customer_id FROM booking GROUP BY customer_id HAVING
COUNT(DISTINCT event_id) > 1);
```

```
-- 12
```

```
SELECT customer_id, customer_name, SUM(total_cost) AS total_revenue
```

```
FROM booking
```

```
JOIN customer ON booking.customer_id = customer.customer_id
```

```
GROUP BY customer_id, customer_name;
```

```
-- 13
```

```
SELECT venue_id, event_type, AVG(ticket_price) AS average_ticket_price  
FROM event  
GROUP BY venue_id, event_type;
```

```
-- 14
```

```
SELECT customer_id, customer_name, SUM(num_tickets) AS total_tickets_purchased  
FROM booking  
JOIN customer ON booking.customer_id = customer.customer_id  
WHERE booking_date >= CURRENT_DATE - INTERVAL '30' DAY  
GROUP BY customer_id, customer_name;
```

```
-----SUBQUERY-----  
-----
```

```
-- 1
```

```
SELECT venue_id, AVG(ticket_price) AS average_ticket_price  
FROM event  
WHERE venue_id IN (SELECT DISTINCT venue_id FROM event)  
GROUP BY venue_id;
```

```
-- 2
```

```
SELECT event_id, event_name  
FROM event  
WHERE (SELECT SUM(num_tickets) FROM booking WHERE booking.event_id = event.event_id) > 0.5  
* total_seats;
```

```
-- 3
```

```
SELECT event_id, event_name, (SELECT SUM(num_tickets) FROM booking WHERE booking.event_id  
= event.event_id) AS total_tickets_sold
```

```
FROM event;
```

```
-- 4
```

```
SELECT customer_id, customer_name
```

```
FROM customer
```

```
WHERE NOT EXISTS (SELECT 1 FROM booking WHERE booking.customer_id = customer.customer_id);
```

```
-- 5
```

```
SELECT event_id, event_name
```

```
FROM event
```

```
WHERE event_id NOT IN (SELECT DISTINCT event_id FROM booking);
```

```
-- 6
```

```
SELECT event_type, SUM(total_tickets_sold) AS total_tickets_sold
```

```
FROM (SELECT event_type, event_id, (SELECT SUM(num_tickets) FROM booking WHERE  
booking.event_id = event.event_id) AS total_tickets_sold FROM event) AS subquery
```

```
GROUP BY event_type;
```

```
-- 7
```

```
SELECT event_id, event_name, ticket_price
```

```
FROM event
```

```
WHERE ticket_price > (SELECT AVG(ticket_price) FROM event);
```

```
-- 8
```

```
SELECT customer_id, customer_name, (SELECT SUM(total_cost) FROM booking WHERE  
booking.customer_id = customer.customer_id) AS total_revenue
```

```
FROM customer;
```

```
-- 9
```

```
SELECT customer_id, customer_name
```

```
FROM customer
```



```
WHERE customer_id IN (SELECT DISTINCT customer_id FROM booking WHERE booking.event_id IN  
(SELECT event_id FROM event WHERE venue_id = 1));
```

```
-- 10
```

```
SELECT event_type, SUM(total_tickets_sold) AS total_tickets_sold  
  
FROM (SELECT event_type, (SELECT SUM(num_tickets) FROM booking WHERE booking.event_id =  
event.event_id) AS total_tickets_sold FROM event) AS subquery  
  
GROUP BY event_type;
```

```
-- 11
```

```
SELECT customer_id, customer_name  
  
FROM customer  
  
WHERE customer_id IN (SELECT DISTINCT customer_id FROM booking WHERE  
DATE_FORMAT(booking_date, '%Y-%m') = '2023-11');
```

```
-- 12
```

```
SELECT venue_id, AVG(ticket_price) AS average_ticket_price  
  
FROM event  
  
WHERE venue_id IN (SELECT DISTINCT venue_id FROM event)  
  
GROUP BY venue_id;
```