```
use ticketbookingsystem;

show tables;


insert into venue(venue_name, address)

 values

('mumbai', 'marol andheri(w)'),

('chennai', 'IT Park'),

('pondicherry', 'state beach');


select * from venue;


insert into customer(customer_name,email,phone_number)

values

('harry potter','harry@gmail.com','45454545'),

('ronald weasley','ron@gmail.com','45454545'),

('hermione granger','her@hmail.com','45454545'),

('draco malfoy','drac@gmail.com','45454545'),

('ginni weasley','ginni@gmail.co','45454545');


insert into customer(customer_name,email,phone_number)

values

('severus snape', 'sev@gmail.co','56556');


select * from customer;


insert into
event(event_name,event_date,event_time,total_seats,available_seats,ticket_price,event_type,venue_id)

values

('Late Ms. Lata Mangeshkar Musical', '2021-09-12', '20:00',320,270,600, 'concert',3),


('CSK vs RCB', '2024-04-11','19:30',23000,3,3600, 'sports', 2),
('CSK vs RR', '2024-04-19','19:30', 23000,10,3400, 'sports', 2),

('MI vs KKR', '2024-05-01', '15:30', 28000,100,8000, 'sports',1);

select * from event;

insert into booking(event_id,customer_id,num_tickets,total_cost,booking_date)
values(1,1,2,640,'2021-09-12'),
(1,4,3,960,'2021-09-12'),
(2,1,3,10800,'2024-04-11'),
(2,3,5,18000,'2024-04-10'),
(3,5,10,34000,'2024-04-15'),
(4,2,4,32000,'2024-05-01');

-- 3 SQL query to select events with available tickets.

select event_name,event_date, available_seats
from event
where available_seats>0;

/*
Late Ms. Lata Mangeshkar Musical      2021-09-12     270
CSK vs RCB      2024-04-11      3
CSK vs RR       2024-04-19      10
MI vs KKR       2024-05-01      100
*/

-- 4 SQL query to select events name partial match with 'cup'.

select *
```

```
from event
where event_name LIKE  '%cup%';


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


select *
from event
where ticket_price between 1000 AND 2500 ;


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


select *
from event
where event_date between '2024-04-12' AND '2024-10-10';


/*
3       CSK vs RR       2024-04-19      19:30:00        23000   10      3400    sports  2
4       MI vs KKR       2024-05-01      15:30:00        28000   100     8000    sports  1

*/
-- 7 SQL query to retrieve events with available tickets that also have "Concert" in their name.


select *
from event
where available_seats>0 AND event_type = 'concert';


/*
1       Late Ms. Lata Mangeshkar Musical        2021-09-12      20:00:00        320     270     600
        concert 3


*/
```

```sql
-- 8 SQL query to retrieve users in batches of 5, starting from the 6th user


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

select *
from booking
where num_tickets > 4;

/*
10      2       3       5       18000   2024-04-10
11      3       5       10      34000   2024-04-15

*/

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

select *
from customer
where phone_number LIKE '%000';

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

select *
from event
where total_seats > 15000;

/*
2       CSK vs RCB      2024-04-11      19:30:00        23000   3       3600    sports  2
3       CSK vs RR       2024-04-19      19:30:00        23000   10      3400    sports  2
```

```
4	MI vs KKR	2024-05-01	15:30:00	28000	100	8000	sports	1

*/

-- 12 Write a SQL query to select events name not start with 'x', 'v', '2'


select event_name
from event
where event_name NOT LIKE '[X,V,2]%';

/*
Late Ms. Lata Mangeshkar Musical
CSK vs RCB
CSK vs RR
MI vs KKR
*/



--  1 SQL query to List Events and Their Average Ticket Prices.
/*
 projection -- event
 criteria -- venue
 */
select e.event_name, avg(b.total_cost)
from event e, booking b
where e.event_id = b.event_id
group by e.event_id;

/*
Late Ms. Lata Mangeshkar Musical	800.0000
CSK vs RCB	14400.0000
CSK vs RR       34000.0000
MI vs KKR       32000.0000
*/

-- 2 SQL query to Calculate the Total Revenue Generated by Events.
/*
projection -- event
criteria -- booking
*/

select e.event_id,e.event_name, sum(b.total_cost * num_tickets) as total_revenue_generated
from event e, booking b
where e.event_id = b.event_id
group by e.event_id;

/*
1       Late Ms. Lata Mangeshkar Musical        4160
2       CSK vs RCB      122400
3       CSK vs RR       340000
4       MI vs KKR       128000
*/

-- 3 SQL query to find the event with the highest ticket sales.
/* projection -- event
   criteria -- booking
   */
select e.event_name, b.num_tickets
from event e,booking b
where e.event_id = b.event_id
group by e.event_name,b.num_tickets
order by b.num_tickets desc
```

```sql
limit 1;


/*
CSK vs RR        10
*/


-- 4 SQL query to Calculate the Total Number of Tickets Sold for Each Event.
/*
projection -- event
criteia -- booking
*/


select e.event_name, sum(distinct b.num_tickets) as total_ticket_sold
from event e, booking b
where e.event_id = b.event_id
group by event_name;


/*


CSK vs RCB       8
CSK vs RR        10
Late Ms. Lata Mangeshkar Musical        5
MI vs KKR        4
*/


-- 5 SQL query to Find Events with No Ticket Sales.


select e.event_name, sum(distinct b.num_tickets) as total_tickets_sold
from event e, booking b
where e.event_id = b.event_id
group by event_name
```

```
having total_tickets_sold = 0;

--  6 SQL query to Find the User Who Has Booked the Most Tickets.
/*
projection -- customer
criteria -- booking
*/

select c.customer_name,sum(distinct b.num_tickets) as total_tickets
from customer c, booking b
where c.customer_id = b.customer_id
group by c.customer_name
order by total_tickets desc
limit 1;

/*
ginni weasley     10
*/

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

-- not possible



-- 8 SQL query to calculate the average Ticket Price for Events in Each Venue.
/*
projection -- venue
criteria -- event
*/

select v.venue_name, avg(e.ticket_price) as average_price
```

```
from venue v, event e
where v.venue_id = e.venue_id
group by v.venue_name;

/*
mumbai		8000.0000
chennai3500.0000
pondicherry	600.0000
*/

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

/*
prjection -- event
criteria -- booking
*/
select e.event_type,sum(b.num_tickets) as total_tickets
from event e join booking b
ON e.event_id = b.event_id
group by e.event_type;

/*concert		5
sports	22*/

-- Task 4

-- 1. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery.
/*
projection -- ticket price of event
criteria -- venue
*/
```

```
select v.venue_name, AVG(e.ticket_price) as avg_ticket_price
from venue v join event e ON v.venue_id = e.venue_id
group by v.venue_name;



  -- 2 Find Events with More Than 50% of Tickets Sold using subquery.
   select event_id, event_name, available_seats
   from event
   where (total_seats-available_seats) > (total_seats/2);

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

 select event_name, SUM(total_seats - available_seats) as tickets_sold
 from event
 group by event_name;

/*
Late Ms. Lata Mangeshkar Musical          50
CSK vs RCB        22997
CSK vs RR         22990
MI vs KKR         27900
*/

 -- 4 Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery.
 insert into customer(customer_name,email,phone_number) values
 ('frodo baggins', 'frodo@latr.com', '3454');
 /*
 projection -- customer
 criteria -- booking
 */
 select *
 from customer

 where customer_id NOT IN (select distinct c.customer_id

                                          from customer c join

            booking b ON c.customer_id = b.customer_id);


            /*

            6       severus snape   sev@gmail.co    56556

7          frodo baggins    frodo@latr.com3454

                              */


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

select event_name, sum(total_seats-available_seats) as total_ticket_sold

from event

group by event_name;


/*


Late Ms. Lata Mangeshkar Musical         50

CSK vs RCB        22997

CSK vs RR         22990

MI vs KKR         27900


*/


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


/*

projection -- ticket price of event

criteria -- event

*/
```

```
select *

from event

where ticket_price > (select AVG (ticket_price)

            from event);


            /*4   MI vs KKR      2024-05-01     15:30:00        28000   100     8000    sports   1

                                                                */



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


select c.customer_name, SUM(b.total_cost)

from customer c join booking b

ON c.customer_id = b.customer_id

group by c.customer_name;


/*


output


harry potter     11440

draco malfoy     960

hermione granger         18000

ginni weasley    34000

ronald weasley  32000

*/




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

```
select *
from customer
where customer_id IN(select customer_id from booking
                     where event_id IN (select event_id from event
                                        where venue_id IN (select venue_id from venue
                                                           where venue_name = 'mumbai')));



                 /* output


                 2      ronald weasley  ron@gmail.com45454545
                        */
```

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

```
select event_type , sum(total_seats-available_seats) as tickets_sold
from event
group by event_type;


/*
output


concert 50
sports   73887


*/
```

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

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

```
select v.venue_name, AVG(e.ticket_price)
from venue v join event e
ON e.venue_id = v.venue_id
group by v.venue_name;

/* output

mumbai          8000.0000
chennai3500.0000
pondicherry     600.0000
*/
```