Ticket Booking System

Submitted by: Anirudh Suryawanshi

anirudhsuryawanshi759@gmail.com

```
-- ----------------------------Task 1 ------------------------------------------

use TicketBookingSystem;
show tables;
describe event;

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@gmail.com', '45454545'),
('draco malfoy', 'drac@gmail.com', '45454545'),
('ginni weasley', 'ginni@gmail.com', '45454545'),
('severus snape','sev@gmail.com',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;

select * from booking;

insert into booking (event_id,customer_id,num_tickets,total_cost,booking_date) values

(4,1,2,640,'2021-09-12'),

(4,4,3,960,'2021-09-12'),

(5,1,3,10800,'2024-04-11'),

(5,3,5,18000,'2024-04-10'),

(6,5,10,34000,'2024-04-15'),

(7,2,4,32000,'2024-05-01');

-- ----------Task 2 ----------

-- Q2 Write a SQL query to list all Events.

select * from event;

/*

+----------+----------+----------+----------+----------+----------+----------+----------+

| event_id | event_name | event_date | event_time | total_seats | available_seats | ticket_price | event_type | venue_id |

+----------+----------+----------+----------+----------+----------+----------+----------+

| 4 | Late Ms. Lata Mangeshkar Musical | 2021-09-12 | 20:00 | 320 | 270 | 600 | concert | 3 |

| 5 | CSK vs RCB | 2024-04-11 | 19:30 | 23000 | 3 | 3600 | sports | 2 |

| 6 | CSK vs RR | 2024-04-19 | 19:30 | 23000 | 10 | 3400 | sports | 2 |

| 7 | MI vs KKR | 2024-05-01 | 15:30 | 28000 | 100 | 8000 | sports | 1 |

| 8 | Arijit Singh Musical | 2024-06-15 | 14:30 | 20000 | 150 | 5000 | concert | 1 |

+----------+---------------------------------+------------+------------+-------------+-----------------+--------------+-----------+----------+*/

-- Q3 Write a SQL query to select events with avallable tickets.

select event_name,available_seats from event;

/*

+---------------------------------+-----------------+

| event_name | available_seats |

+---------------------------------+-----------------+

| Late Ms. Lata Mangeshkar Musical | 270 |

| CSK vs RCB | 3 |

| CSK vs RR | 10 |

| MI vs KKR | 100 |

| Arijit Singh Musical | 150 |

+---------------------------------+-----------------+*/

-- Q4 Write a SQL query to select events name partial match with 'cup'.

select event_name

from event

where event_name LIKE '%csk%';

/*

+------------+

| event_name |

+------------+

| CSK vs RCB |

| CSK vs RR |

+------------+*/

-- Q5 Write a SQL query to select events with ticket price range is between 1000 to 2500.

```
select event_name,ticket_price
from event
where ticket_price between 500 AND 4000;
/*
+---------------------------------+--------------+
| event_name                      | ticket_price |
+---------------------------------+--------------+
| Late Ms. Lata Mangeshkar Musical |         600 |
| CSK vs RCB                      |        3600 |
| CSK vs RR                       |        3400 |
+---------------------------------+--------------+*/
```

-- Q6 Write a SQL query to retrieve events with dates falling within a specific range.

```
select event_name,event_date
from event
where event_date between '2024-1-1' AND '2024-5-1';
/*
+------------+------------+
| event_name | event_date |
+------------+------------+
| CSK vs RCB | 2024-04-11 |
| CSK vs RR  | 2024-04-19 |
| MI vs KKR  | 2024-05-01 |
+------------+------------+*/
```

-- Q7 Write a SQL query to retrieve events with available tickets that also have "Concert" in their name.

```
select event_name,available_seats
from event
where event_name LIKE '%lata%';
```

```
/*
+---------------------------------+----------------+
| event_name                      | available_seats |
+---------------------------------+----------------+
| Late Ms. Lata Mangeshkar Musical |            270 |
+---------------------------------+----------------+*/

-- Q8 Write a SQL query to retrieve users in batches of 5, starting from the 6th user.
select *
from customer
limit 2,3;
/*
+-------------+-----------------+----------------+--------------+
| customer_id | customer_name    | email          | phone_number |
+-------------+-----------------+----------------+--------------+
|           3 | hermione granger | her@gmail.com   | 45454545     |
|           4 | draco malfoy     | drac@gmail.com  | 45454545     |
|           5 | ginni weasley    | ginni@gmail.com | 45454545     |
+-------------+-----------------+----------------+--------------+*/

-- Q9 Write a SQL query to retrieve bookings details contains booked no of ticket more than 4.
select *
from booking
where num_tickets>4;
/*
+------------+----------+-------------+-------------+------------+--------------+
| booking_id | event_id | customer_id | num_tickets | total_cost | booking_date |
+------------+----------+-------------+-------------+------------+--------------+
|          5 |        5 |           3 |           5 |      18000 | 2024-04-10   |
|          6 |        6 |           5 |          10 |      34000 | 2024-04-15   |
+------------+----------+-------------+-------------+------------+--------------+*/
```

```
-- Q10 Write a SQL query to retrieve customer information whose phone number end with '000'
select *
from customer
where phone_number LIKE '%4545';
/*
+-------------+-----------------+----------------+--------------+
| customer_id | customer_name    | email          | phone_number |
+-------------+-----------------+----------------+--------------+
|           1 | harry potter     | harry@gmail.com | 45454545     |
|           2 | ronald weasley   | ron@gmail.com   | 45454545     |
|           3 | hermione granger | her@gmail.com   | 45454545     |
|           4 | draco malfoy     | drac@gmail.com  | 45454545     |
|           5 | ginni weasley    | ginni@gmail.com | 45454545     |
+-------------+-----------------+----------------+--------------+*/

-- Q11 Write a SQL query to retrieve the events in order whose seat capacity more than 15000.
select event_name,total_seats
from event
where total_seats>20000
order by total_seats DESC;
/*
+------------+-------------+
| event_name | total_seats |
+------------+-------------+
| MI vs KKR  |       28000 |
| CSK vs RCB |       23000 |
| CSK vs RR  |       23000 |
+------------+-------------+*/

-- Q12 Write a SQL query to select events name not start with 'x', 'y', 'z'
```

```
select event_name
from event
where event_name not like 'c%';
/*
+----------------------------------+
| event_name                       |
+----------------------------------+
| Late Ms. Lata Mangeshkar Musical |
| MI vs KKR                        |
| Arijit Singh Musical             |
+----------------------------------+*/

-- ----------------------------Task 3 ----------------------------------------
-- Q1 Write a SQL query to List Events and Their Average Ticket Prices.
select e.event_name, avg(b.total_cost) as average
from event e, booking b
where e.event_id=b.event_id
group by e.event_id;
/*
+----------------------------------+------------+
| event_name                       | average    |
+----------------------------------+------------+
| Late Ms. Lata Mangeshkar Musical |   800.0000 |
| CSK vs RCB                       | 14400.0000 |
| CSK vs RR                        | 34000.0000 |
| MI vs KKR                        | 32000.0000 |
+----------------------------------+------------+*/

-- Q2 Write a SQL query to Calculate the Total Revenue Generated by Events.
select e.event_name, sum(b.total_cost) as total
from event e, booking b
```

```
where e.event_id=b.event_id
group by e.event_id;
/*
+----------------------------------+-------+
| event_name                       | total |
+----------------------------------+-------+
| Late Ms. Lata Mangeshkar Musical |  1600 |
| CSK vs RCB                       | 28800 |
| CSK vs RR                        | 34000 |
| MI vs KKR                        | 32000 |
+----------------------------------+-------+*/

-- Q3 Write a SQL query to find the event with the highest ticket sales.
select e.event_name, sum(b.total_cost) as total
from event e, booking b
where e.event_id=b.event_id
group by e.event_id
order by total desc
limit 1;
/*
+------------+-------+
| event_name | total |
+------------+-------+
| CSK vs RR  | 34000 |
+------------+-------+*/

-- Q4 Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event.
select e.event_name, sum(b.num_tickets) as total_tickets
from event e, booking b
where e.event_id=b.event_id
group by e.event_id;
```

```
/*
+---------------------------------+--------------+
| event_name                      | total_tickets |
+---------------------------------+--------------+
| Late Ms. Lata Mangeshkar Musical |            5 |
| CSK vs RCB                      |            8 |
| CSK vs RR                       |           10 |
| MI vs KKR                       |            4 |
+---------------------------------+--------------+*/

-- Q5 Write a SQL query to Find Events with No Ticket Sales.
select *
from event
where event_id NOT IN (select event.event_id
                                        from event e, booking b
                                        where e.event_id=b.event_id);
/* no record available */

-- Q6 Write a SQL query to Find the User Who Has Booked the Most Tickets.
select c.customer_name, sum(b.num_tickets) as total_tickets
from event e, booking b, customer c
where e.event_id=b.event_id AND b.customer_id=c.customer_id
group by c.customer_id
order by total_tickets DESC
limit 1;
/*
+---------------+---------------+
| customer_name | total_tickets |
+---------------+---------------+
| ginni weasley |            10 |
+---------------+---------------+*/
```

-- Q7 Write a SQL query to List Events and the total number of tickets sold for each month. 8. Write a SQL query to calculate the average Ticket Price for Events in Each Venue.

```
select e.event_name, EXTRACT(MONTH FROM e.event_date) as month, sum(b.num_tickets) as sum
from event e, booking b
where e.event_id=b.event_id
group by EXTRACT(MONTH FROM e.event_date);
```

/*

```
+---------------------------------+-------+------+
| event_name                      | month | sum  |
+---------------------------------+-------+------+
| CSK vs RCB                      |     4 |   18 |
| MI vs KKR                       |     5 |    4 |
| Late Ms. Lata Mangeshkar Musical |     9 |    5 |
+---------------------------------+-------+------+*/
```

-- Q8 Write a SQL query to calculate the average Ticket Price for Events in Each Venue.

```
select v.venue_name, avg(ticket_price) as average
from event e, venue v
where e.venue_id=v.venue_id
group by v.venue_id;
```

/*

```
+-------------+---------+
| venue_name  | average |
+-------------+---------+
| mumbai      |    6500 |
| chennai     |    3500 |
| pondicherry |     600 |
+-------------+---------+*/
```

-- Q9 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
from event e, booking b
where e.event_id=b.event_id
group by e.event_type;
/*
+------------+---------------+
| event_type | total_tickets |
+------------+---------------+
| concert    |             5 |
| sports     |            22 |
+------------+---------------+*/

-- Q10 Write a SQL query to calculate the total Revenue Generated by Events in Each Year.
select EXTRACT(YEAR FROM e.event_date) as Year, sum(b.total_cost) as total_cost
from event e, booking b
where e.event_id=b.event_id
group by EXTRACT(YEAR FROM e.event_date);
/*
+------+------------+
| Year | total_cost |
+------+------------+
| 2021 |       1600 |
| 2024 |      94800 |
+------+------------+*/

-- Q11 Write a SQL query to list users who have booked tickets for multiple events..
select c.customer_name, c.email
from booking b, customer c
where c.customer_id = b.customer_id
group by c.customer_id
having count(c.customer_id) >1;
```

```
/*
+---------------+-----------------+
| customer_name | email           |
+---------------+-----------------+
| harry potter  | harry@gmail.com |
+---------------+-----------------+*/

-- Q12 Write a SQL query to calculate the Total Revenue Generated by Events for Each User.
select c.customer_name, sum(b.total_cost) as sum
from event e, booking b, customer c
where e.event_id=b.event_id AND b.customer_id=c.customer_id
group by b.customer_id;
/*
+------------------+-------+
| customer_name    | sum   |
+------------------+-------+
| harry potter     | 11440 |
| ronald weasley   | 32000 |
| hermione granger | 18000 |
| draco malfoy     |   960 |
| ginni weasley    | 34000 |
+------------------+-------+*/

-- Q13 Write a SQL query to calculate the Average Ticket Price for Events In Each Category and Venue.
select e.event_type, avg(e.ticket_price)
from venue v, event e, booking b
where e.venue_id=v.venue_id AND e.event_id=b.event_id
group by event_type;
/*
+------------+--------------------+
```

| event_type | avg(e.ticket_price) |

+------------+---------------------+

| concert | 600 |

| sports | 4650 |

+------------+---------------------+*/

-- ----------------------------Task 4 --

-- Q1 Calculate the Average Ticket Price for Events in Each Venue Using a Subquery.

select v.venue_name, avg(e.ticket_price) as average

from event e JOIN venue v ON e.venue_id=v.venue_id

group by v.venue_id;

/*

+-------------+---------+

| venue_name | average |

+-------------+---------+

| mumbai | 6500 |

| chennai | 3500 |

| pondicherry | 600 |

+-------------+---------+*/

-- Q2 Find Events with More Than 50% of Tickets Sold using subquery.

select *

from event

where (total_seats-available_seats)>(total_seats/2);

/*

+----------+---------------------+------------+------------+-------------+-----------------+--------------+------------+-------
---+

| event_id | event_name | event_date | event_time | total_seats | available_seats |
ticket_price | event_type | venue_id |

+----------+---------------------+------------+------------+-------------+-----------------+--------------+------------+-------
---+

| 5 | CSK vs RCB | 2024-04-11 | 19:30 | 23000 | 3 | 3600 | sports | 2 |

| 6 | CSK vs RR | 2024-04-19 | 19:30 | 23000 | 10 | 3400 | sports | 2 |

| 7 | MI vs KKR | 2024-05-01 | 15:30 | 28000 | 100 | 8000 | sports | 1 |

| 8 | Arijit Singh Musical | 2024-06-15 | 14:30 | 20000 | 150 | 5000 | concert | 1 |

+----------+----------------------+------------+------------+-------------+-----------------+--------------+------------+-----------+*/

-- Q3 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_id;

/*

+---------------------------------+--------------+

| event_name | tickets_sold |

+---------------------------------+--------------+

| Late Ms. Lata Mangeshkar Musical | 50 |

| CSK vs RCB | 22997 |

| CSK vs RR | 22990 |

| MI vs KKR | 27900 |

| Arijit Singh Musical | 19850 |

+---------------------------------+--------------+*/

-- Q4 Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery

select *

from customer

where customer_id NOT IN(select customer_id from booking);

/*

+-------------+---------------+---------------+--------------+

| customer_id | customer_name | email | phone_number |

```
+-------------+---------------+---------------+--------------+

|          6 | severus snape | sev@gmail.com | 56556        |

|          7 | tom holland   | tom@gmail.com | 12345        |

+-------------+---------------+---------------+--------------+*/


-- Q5 List Events with No Ticket Sales Using a NOT IN Subquery.

select *

from event

where event_id NOT IN(select event_id from booking );

/*+----------+---------------------+------------+------------+-------------+-----------------+--------------+------------+-------------+
| event_id | event_name          | event_date | event_time | total_seats | available_seats | ticket_price | event_type | venue_id |
+----------+---------------------+------------+------------+-------------+-----------------+--------------+------------+-------------+
|        8 | Arijit Singh Musical | 2024-06-15 | 14:30      |       20000 |             150 |         5000 | concert    |        1 |
+----------+---------------------+------------+------------+-------------+-----------------+--------------+------------+-------------+*/


-- Q6 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)

from (select * from event) as dt

group by event_type;

/*

+---------------------------------+---------------------------------+

| event_name                      | sum(total_seats-available_seats) |

+---------------------------------+---------------------------------+

| Late Ms. Lata Mangeshkar Musical |                           19900 |

| CSK vs RCB                      |                           73887 |

+---------------------------------+---------------------------------+*/
```

-- Q7 Find Events with Ticket Prices Higher Than the Average Ticket Price Using a Subquery in the WHERE Clause.

```
select *

from event

where ticket_price>(select avg(ticket_price)from event);

/*

+----------+---------------------+------------+------------+-------------+----------------+--------------+------------+-------
---+

| event_id | event_name          | event_date | event_time | total_seats | available_seats |
ticket_price | event_type | venue_id |

+----------+---------------------+------------+------------+-------------+----------------+--------------+------------+-------
---+

|       7 | MI vs KKR           | 2024-05-01 | 15:30      |       28000 |           100 |        8000 | sports     |
1 |

|       8 | Arijit Singh Musical | 2024-06-15 | 14:30      |       20000 |           150 |        5000 | concert
|       1 |

+----------+---------------------+------------+------------+-------------+----------------+--------------+------------+-------
---+*/


-- Q8 Calculate the Total Revenue Generated by Events for Each User Using a Correlated Subquery.

select c.customer_name, sum(b.total_cost) as total

from customer c JOIN booking b ON c.customer_id=b.customer_id

group by c.customer_id;

/*

+------------------+-------+

| customer_name    | total |

+------------------+-------+

| harry potter     | 11440 |

| ronald weasley   | 32000 |

| hermione granger | 18000 |

| draco malfoy     |   960 |

| ginni weasley    | 34000 |

+------------------+-------+*/
```

```
-- Q9 List Users Who Have Booked Tickets for Events in a Given Venue Using a Subquery in the WHERE Clause.
select c.customer_name,c.email
from customer c
        JOIN booking b ON c.customer_id=b.customer_id
    JOIN event e ON b.event_id=e.event_id
    JOIN venue v ON e.venue_id=v.venue_id
where v.venue_name='chennai';
/*
+------------------+-----------------+
| customer_name    | email           |
+------------------+-----------------+
| harry potter     | harry@gmail.com |
| hermione granger | her@gmail.com   |
| ginni weasley    | ginni@gmail.com |
+------------------+-----------------+*/

-- Using sub query
select customer_name,email
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='chennai')));
/*
+------------------+-----------------+
| customer_name    | email           |
```

```
+-----------------+-----------------+
| harry potter    | harry@gmail.com |
| hermione granger | her@gmail.com   |
| ginni weasley   | ginni@gmail.com |
+-----------------+-----------------+*/

-- Q10 Calculate the Total Number of Tickets Sold for Each Event Category Using a Subquery with GROUP BY.
select e.event_type, sum(b.num_tickets)
from event e JOIN booking b ON e.event_id=b.event_id
group by e.event_type;
/*
+------------+--------------------+
| event_type | sum(b.num_tickets) |
+------------+--------------------+
| concert    |                  5 |
| sports     |                 22 |
+------------+--------------------+*/

-- Q11 Find Users Who Have Booked Tickets for Events in each Month Using a Subquery with DATE FORMAT.
select EXTRACT(MONTH FROM e.event_date), count(c.customer_id) as tickets_bought
from event e
        JOIN booking b on e.event_id=b.event_id
    JOIN customer c on c.customer_id=b.customer_id
group by EXTRACT(MONTH FROM e.event_date);
/*
+---------------------------------+----------------+
| EXTRACT(MONTH FROM e.event_date) | tickets_bought |
+---------------------------------+----------------+
|                               4 |              3 |
|                               5 |              1 |
```

```
|                        9 |           2 |
+----------------------------------+---------------+*/

-- Q12 Calculate the Average Ticket Price for Events in Each Venue Using a Subquery
select v.venue_name, avg(e.ticket_price)
from event e JOIN venue v ON e.venue_id=v.venue_id
group by EXTRACT(MONTH FROM e.event_date);
/*
+-------------+--------------------+
| venue_name  | avg(e.ticket_price) |
+-------------+--------------------+
| chennai     |             3500 |
| mumbai      |             8000 |
| mumbai      |             5000 |
| pondicherry |              600 |
+-------------+--------------------+*/
```