

```
use mydb;
```

```
show tables;
```

```
select * from booking;
```

```
-- drop database mydb;
```

```
describe booking;
```

```
describe customer;
```

```
describe event;
```

```
describe venue;
```

```
-- INSERT into VENUE
```

```
insert into venue
```

```
(venue_name, venue_address) values
```

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

```
('chennai', 'IT Park'),
```

```
('pondicherry', 'state bank');
```

```
select * from venue;
```

```
update venue set venue_address = 'state beach' where venue_id = 3;
```

```
-- INSERT into customer
```

```
insert into customer
```

```
(customer_name, customer_email, customer_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),
```

```
('ginny weasley', 'ginny@gmail.com', 45454545),
```

```
('severus snape', 'sev@gmail.com', 565556);
```

```
select * from customer;
```

```
-- INSERT into event
```

```
insert into event
```

```
(event_name, event_date, event_time, event_total_seats, event_available_seats, event_ticket_price, event_type, venue_id) values
```

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

```
insert into event
```

```
(event_name, event_date, event_time, event_total_seats, event_available_seats, event_ticket_price, event_type, venue_id) values
```

```
('CSK vs RCB', '2024-04-11', '19:30:00', 23000, 3, 3600, 'sports', 2),
```

```
('CSK vs RR', '2024-04-19', '19:30:00', 23000, 10, 3400, 'sports', 2),
```

```
('Conference CUP', '2024-05-01', '15:30:00', 28000, 99, 8000, 'sports', 1),
```

```
('Krsna Hip-Hop Concert', '2024-04-13', '20:30:00', 2000, 13, 4000, 'concert', 3);
```

```
select * from event;
```

```
-- INSERT into booking
```

```
insert into booking
```

```
(event_id, customer_id, booking_num_tickets, booking_total_cost, booking_date) values
```

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

```
(2, 3, 5, 3000, '2024-03-15'),
```

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

```
(4, 1, 3, 10800, '2024-04-11'),
```

```
(4, 3, 5, 18000, '2024-04-10'),
```

```
(4, 5, 10, 34000, '2024-04-15'),
```

```
(4, 2, 4, 32000, '2024-05-01'),
```

```
(1, 6, 1, 8000, '2024-03-15');
```

```
select * from booking;
```

```
alter table booking rename column booking_total_coast to booking_total_cost;
```

```
-- Tasks 2: Select, Where, Between, AND, LIKE:
```

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

```
-- Done above
```

```
-- 2. Write a SQL query to list all Events.
```

```
select * from event;
```

```
/* output
```

```
'1','CSK vs RCB','2024-04-11','19:30:00','23000','3','3600','sports','2'
```

```
'2','CSK vs RR','2024-04-19','19:30:00','23000','10','3400','sports','2'
```

```
'3','Conference CUP','2024-05-01','15:30:00','28000','99','8000','sports','1'
```

```
'4','Krsna Hip-Hop Concert','2024-04-13','20:30:00','2000','13','4000','concert','3'
```

```
'5','Late Ms. Lata Mangeshkar Muscial','2021-09-12','20:00:00','320','270','600','concert','3'
```

```
*/
```

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

```
select * from event where event_available_seats > 0;
```

```
/* output
```

```
'1','CSK vs RCB','2024-04-11','19:30:00','23000','3','3600','sports','2'
```

```
'2','CSK vs RR','2024-04-19','19:30:00','23000','10','3400','sports','2'
```

```
'3','Conference CUP','2024-05-01','15:30:00','28000','99','8000','sports','1'
```

```
'4','Krsna Hip-Hop Concert','2024-04-13','20:30:00','2000','13','4000','concert','3'
```

```
'5','Late Ms. Lata Mangeshkar Muscial','2021-09-12','20:00:00','320','270','600','concert','3'
```

```
*/
```

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

```
select * from event where event_name like '%cup%';
```

```
/* output
```

```
'3','Conference CUP','2024-05-01','15:30:00','28000','99','8000','sports','1'
```

```
*/
```

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

```
select * from event where event_ticket_price > 100 and event_ticket_price < 2500;
```

```
select * from event where event_ticket_price between 100 and 2500;
```

```
/* output
```

```
'5','Late Ms. Lata Mangeshkar Muscial','2021-09-12','20:00:00','320','270','600','concert','3'
```

```
*/
```

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

```
select * from event where event_date between '2024-04-1' and '2024-04-29';
```

```
/* output
```

```
'1','CSK vs RCB','2024-04-11','19:30:00','23000','3','3600','sports','2'
```

```
'2','CSK vs RR','2024-04-19','19:30:00','23000','10','3400','sports','2'
```

```
'4','Krsna Hip-Hop Concert','2024-04-13','20:30:00','2000','13','4000','concert','3'
```

```
*/
```

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

```
select * from event where event_available_seats > 0 and event_name like '%Concert%';
```

```
/* output
```

```
*/
```

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

```
select * from customer order by customer_id limit 5 offset 5;
```

```
/* output
```

```
'4','Krsna Hip-Hop Concert','2024-04-13','20:30:00','2000','13','4000','concert','3'
```

```
*/
```

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

```
select * from booking where booking_num_tickets > 4;
```

```
/* output
```

```
*/
```

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

```
select * from customer where customer_phone_number like '%000';
```

```
/* output
```

```
'2','3','5','3000','2024-03-15'
```

```
'4','3','5','18000','2024-04-10'
```

```
'4','5','10','34000','2024-04-15'
```

```
*/
```

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

```
select * from event where event_total_seats > 15000 order by event_total_seats asc;
```

```
/* output
```

```
'1','CSK vs RCB','2024-04-11','19:30:00','23000','3','3600','sports','2'
```

```
'2','CSK vs RR','2024-04-19','19:30:00','23000','10','3400','sports','2'
```

```
'3','Conference CUP','2024-05-01','15:30:00','28000','99','8000','sports','1'
```

```
*/
```

-- 12. Write a SQL query to select events name not start with 'x', 'Y', '7'

```
select event_name from event where event_name not like 'x%' and event_name not like 'Y%' and event_name not like '7%';
```

```
/* output
```

```
'CSK vs RCB'
```

```
'CSK vs RR'
```

```
'Conference CUP'
```

```
'Krsna Hip-Hop Concert'
```

```
'Late Ms. Lata Mangeshkar Muscial'
```

```
*/
```

-- Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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

```
select e.event_type, avg(e.event_ticket_price)
```

```
from event e
```

```
group by e.event_type;
```

```
/* output
```

```
'sports','5000'
```

```
'concert','2300'
```

```
*/
```

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

-- projection: booking

-- criteria: event

```
select e.event_name, sum(booking_total_cost) as 'Total_Revenue'
```

```
from booking b join event e on b.event_id = e.event_id
```

```
group by e.event_name;
```

```
/* output
```

```
'CSK vs RCB','8640'
```

```
'CSK vs RR','3000'  
'Conference CUP','960'  
'Krsna Hip-Hop Concert','94800'
```

```
*/
```

```
-- 3. Write a SQL query to find the event with the highest ticket sales.  
-- projection: event  
-- criteria: event  
select e.event_name, e.event_total_seats - e.event_available_seats as 'Sold_Tickets'  
from event e  
order by e.event_total_seats - e.event_available_seats desc  
limit 1;  
/* output  
'Conference CUP','27901'
```

```
*/
```

```
-- 4. Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event.  
select e.event_name, e.event_total_seats - e.event_available_seats as 'Sold_Tickets'  
from event e;  
/* output  
'CSK vs RCB','22997'  
'CSK vs RR','22990'  
'Conference CUP','27901'  
'Krsna Hip-Hop Concert','1987'  
'Late Ms. Lata Mangeshkar Muscial','50'
```

```
*/
```

```
-- 5. Write a SQL query to Find Events with No Ticket Sales.  
select *  
from event e  
where e.event_total_seats = e.event_available_seats;  
/* output  
null  
*/
```

```
-- 6. Write a SQL query to Find the User Who Has Booked the Most Tickets.  
-- Projection:customer, Criteria: booking  
select c.customer_id, c.customer_name, b.booking_num_tickets  
from customer c, booking b  
where c.customer_id = b.customer_id  
order by b.booking_num_tickets desc  
limit 1;  
/* output
```

```
'5','ginni weasley','10'
```

```
*/
```

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

```
-- NOT Possible without 'Month' column
```

```
/* output
```

```
*/
```

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

```
-- Projection: event criteria: venue
```

```
select v.venue_name, avg(e.event_ticket_price) as 'Average Ticket Price'
```

```
from event e, venue v
```

```
group by v.venue_name;
```

```
/* output
```

```
'pondicherry','3920'
```

```
'chennai','3920'
```

```
'mumbai','3920'
```

```
*/
```

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

```
-- Projection: booking Criteria: event
```

```
select e.event_type, sum(b.booking_num_tickets)
```

```
from event e, booking b
```

```
group by e.event_type;
```

```
/* output
```

```
'concert','66'
```

```
'sports','99'
```

```
*/
```

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

```
-- NOT possible without 'Year' column
```

```
/* output
```

```
*/
```

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

```
-- projection: customer
```

```
-- criteria: booking
```

```
select c.customer_name
```

```
from booking b join customer c on b.customer_id = c.customer_id
```

```
having b.event_id > 1;
```

```
/* output
```

```
*/
```

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

-- projection: event

-- criteria: customer

```
select c.customer_name, sum(b.booking_total_cost) as 'Total_Revenue_Generated'
```

```
from event e join booking b on e.event_id = b.event_id
```

```
join customer c on b.customer_id = c.customer_id
```

```
group by c.customer_name;
```

/\* output

'harry potter','11440'

'ronald weasley','32000'

'hermione granger','21000'

'draco malfoy','960'

'ginny weasley','34000'

'severus snape','8000'

\*/

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

-- projection: event

-- criteria: event, venue

```
select event_type, avg(event_ticket_price)
```

```
from event
```

```
group by event_type;
```

/\* output

'sports','5000'

'concert','2300'

\*/

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

-- Question NOT clearly visible in PDF

/\* output

\*/

-- Tasks 4: Subquery and its types

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

-- projection: event

-- criteria: venue

```
select v.venue_name, avg(e.event_ticket_price)
```

```
from event e join venue v on e.venue_id = v.venue_id
```

```
group by v.venue_name;
```

/\* output

'mumbai','8000'

'chennai','3500'

'pondicherry','2300'

```
*/
```

```
-- 2. Find Events with More Than 50% of Tickets Sold using subquery.
-- projection: event
-- criteria: event
select *
from event
where (event_total_seats - event_available_seats) > (event_total_seats/2);
/* output
'1','CSK vs RCB','2024-04-11','19:30:00','23000','3','3600','sports','2'
'2','CSK vs RR','2024-04-19','19:30:00','23000','10','3400','sports','2'
'3','Conference CUP','2024-05-01','15:30:00','28000','99','8000','sports','1'
'4','Krsna Hip-Hop Concert','2024-04-13','20:30:00','2000','13','4000','concert','3'
```

```
*/
```

```
-- 3. Calculate the Total Number of Tickets Sold for Each Event.
-- projection: event
-- criteria: event
select sum(event_total_seats - event_available_seats)
from event
group by event_name;
/* output
'22997'
'22990'
'27901'
'1987'
'50'
*/
```

```
-- 4. Find customer Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery.
-- projection: customer
-- criteria: booking
select *
from customer
where customer_id not in (select customer_id
                           from booking);
/* output
null
*/
```

```
-- 5. List Events with No Ticket Sales Using a NOT IN Subquery.
-- projection: event
-- criteria: booking
```



```

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

/* output
'5','Late Ms. Lata Mangeshkar Muscial','2021-09-12','20:00:00','320','270','600','concert','3'

*/

```

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

```

-- projection: event
-- criteria: event
select sum(event_total_seats - event_available_seats)
from event
group by event_type;

/* output
'73888'
'2037'

*/

```

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

```

-- projection: event
-- criteria: event
select *
from event
where event_id in (select event_id
                   where event_ticket_price > (select avg(event_ticket_price) from event));

/* output
'3','Conference CUP','2024-05-01','15:30:00','28000','99','8000','sports','1'
'4','Krsna Hip-Hop Concert','2024-04-13','20:30:00','2000','13','4000','concert','3'

*/

```

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

```

-- projection: customer
-- criteria: booking
select c.customer_name, sum(b.booking_total_cost)
from customer c join booking b on c.customer_id = b.customer_id
group by c.customer_name;

/* output
'harry potter','11440'
'ronald weasley','32000'
'hermione granger','21000'
'draco malfoy','960'

```

```
'ginni weasley','34000'  
'severus snape','8000'
```

```
*/
```

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

```
-- projection: customers
```

```
-- criteria: venue
```

```
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
```

```
'4','draco malfoy','drac@gmail.com','45454545'
```

```
*/
```

```
-- 10. Calculate the Total Number of Tickets Sold for Each Event Category Using a Subquery with Group by
```

```
-- projection: event
```

```
-- criteria: event
```

```
select event_type, sum(event_total_seats-event_available_seats)
```

```
from event e
```

```
group by event_type;
```

```
/* output
```

```
'sports','73888'
```

```
'concert','2037'
```

```
*/
```

```
-- 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
```

```
-- projection: event
```

```
-- criteria: venue
```

```
select v.venue_name, avg(e.event_ticket_price)
```

```
from venue v join event e on v.venue_id = e.venue_id
```

```
group by v.venue_name;
```

```
/* output
```

```
'mumbai','8000'
```

```
'chennai','3500'
```

'pondicherry','2300'

\*/