Assignment-5

Task-1

1, Create the database named "TicketBookingSystem"

```
create database TicketBookingSystem;
use TicketBookingSystem;
```

64	15:02:49	create database TicketBookingSystem	1 row(s) affected
65	15:02:53	use TicketBookingSystem	0 row(s) affected

2, Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships. • Venu • Event • Customers • Booking

Venue_table:

```
create table venue_table
(venue_id int primary key, venue_name varchar(255), address varchar(255));
desc venue_table;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
venue_id	int	NO	PRI	NULL	
venue_name	varchar(255)	YES		NULL	
address	varchar(255)	YES		NULL	

event:

```
create table event
(event_id int primary key,event_name varchar(255), event_date date, event_time time, venue_id int, foreign key(venue_id) references
venue_table(venue_id),total_seats bigint, avaliable_seats bigint, ticket_price DECIMAL(10,2),event_type text);
alter table event add column booking_id int;
alter table event add
foreign key(booking_id) references booking(booking_id);
desc event;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
event_id	int	NO	PRI	NULL	
event_name	varchar(255)	YES		NULL	
event_date	date	YES		NULL	
event_time	time	YES		NULL	
venue_id	int	YES	MUL	NULL	
total_seats	bigint	YES		NULL	
avaliable_seats	bigint	YES		NULL	
ticket_price	decimal(10,2)	YES		NULL	
event_type	text	YES		NULL	
booking_id	int	YES	MUL	NULL	

Customer:

```
create table customer
(customer_id int primary key, customer_name text, email text, phone varchar(11));
alter table customer add column booking_id int;
alter table customer add
foreign key(booking_id) references booking(booking_id);
desc customer;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
customer_name	text	YES		NULL	
email	text	YES		NULL	
phone	varchar(11)	YES		NULL	
booking_id	int	YES		NULL	

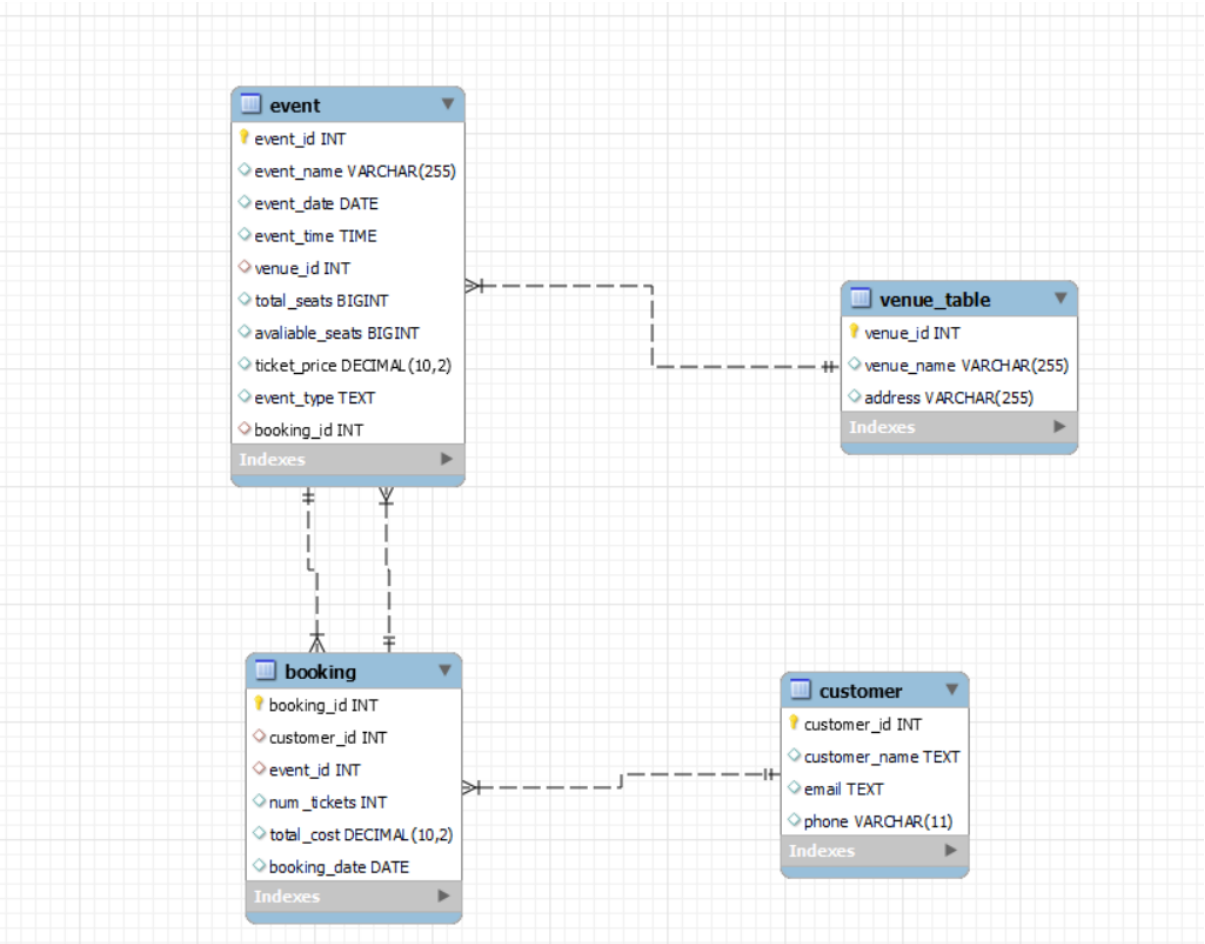
Booking:

```
--
create table booking
(booking_id  int primary key, customer_id int ,foreign key(customer_id) references customer(customer_id), event_id int,
foreign key(event_id) references event(event_id), num_tickets int, total_cost decimal(10,2), booking_date date);
desc booking;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
booking_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
event_id	int	YES	MUL	NULL	
num_tickets	int	YES		NULL	
total_cost	decimal(10,2)	YES		NULL	
booking_date	date	YES		NULL	

3, Create an ERD (Entity Relationship Diagram) for the database.



Task-2

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

Venue:

```sql
INSERT INTO Venue_table (venue_id, venue_name, address)
VALUES
    (1, 'Royal Palace Banquet Hall', '123 Main Street, Mumbai, Maharashtra, India'),
    (2, 'Green Valley Gardens', '456 Avenue, Bangalore, Karnataka, India'),
    (3, 'Grand Plaza Convention Center', '789 Pine Road, Delhi, NCR, India'),
    (4, 'Elegant Halls and Events', '101 Maple Lane, Chennai, Tamil Nadu, India'),
    (5, 'Celestial Ballroom', '202 Cedar Street, Kolkata, West Bengal, India'),
    (6, 'Sapphire Wedding Venues', '303 Birch Avenue, Hyderabad, Telangana, India'),
    (7, 'Golden Sands Resort', '404 Elm Lane, Pune, Maharashtra, India'),
    (8, 'Crystal Gardens', '505 Redwood Road, Ahmedabad, Gujarat, India'),
    (9, 'Heritage Hall', '606 Willow Lane, Jaipur, Rajasthan, India'),
    (10, 'Starlight Convention Center', '707 Oak Street, Lucknow, Uttar Pradesh, India');
    select * from venue_table;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

venue_id	venue_name	address
1	Royal Palace Banquet Hall	123 Main Street, Mumbai, Maharashtra, India
2	Green Valley Gardens	456 Avenue, Bangalore, Karnataka, India
3	Grand Plaza Convention Center	789 Pine Road, Delhi, NCR, India
4	Elegant Halls and Events	101 Maple Lane, Chennai, Tamil Nadu, India
5	Celestial Ballroom	202 Cedar Street, Kolkata, West Bengal, India
6	Sapphire Wedding Venues	303 Birch Avenue, Hyderabad, Telangana, India
7	Golden Sands Resort	404 Elm Lane, Pune, Maharashtra, India
8	Crystal Gardens	505 Redwood Road, Ahmedabad, Gujarat, India
9	Heritage Hall	606 Willow Lane, Jaipur, Rajasthan, India
10	Starlight Convention Center	707 Oak Street, Lucknow, Uttar Pradesh, India
NULL	NULL	NULL

Event:

```sql
alter table event add
foreign key(booking_id) references booking(booking_id);
INSERT INTO Event (event_id, event_name, event_date, event_time, venue_id, total_seats, avaliable_seats, ticket_price, event_type, booking_id)
VALUES
    (1, 'Bollywood Night Extravaganza','2024-06-17', '18:30:00', 1, 50, 47, 250.00, 'Concert',null),
    (2, 'Cricket Tournament Finals world cup', '2024-05-20', '15:00:00', 2, 15500, 15480, 2400.00, 'Sports',null),
    (3, 'Classic Movie Marathon', '2024-07-21', '12:00:00', 3, 100, 95, 50.00, 'Movie', null),
    (4, 'Live Stand-Up Comedy Show', '2024-05-22', '19:00:00', 4, 50, 45, 120.00, 'Concert',null),
    (5, 'Football Championship', '2024-05-01', '16:30:00', 5, 100, 100, 80.00, 'Sports',null),
    (6, 'Melodious Sufi Night', '2024-08-10', '20:00:00', 6, 50, 49, 200.00, 'Concert',null),
    (7, 'Outdoor Movie Night', '2024-06-25', '19:30:00', 7, 50, 45, 75.00, 'Movie',null),
    (8, 'Basketball Showdown', '2024-07-15', '14:00:00', 8, 150, 145, 60.00, 'Sports',null),
    (9, 'Folk Music Festival', '2024-09-03', '17:00:00', 9, 50, 47, 150.00, 'Concert',null),
    (10, 'Tennis Championship asian cup', '2024-09-11', '12:30:00', 10, 200, 190, 1000.00, 'Sports',null);
   select * from event;

    update event set booking_id=101 where event_id=1;
    update event set booking_id=102 where event_id=2;
    update event set booking_id=103 where event_id=3;
    update event set booking_id=104 where event_id=4;
```

```
update event set booking_id=105 where event_id=6;
update event set booking_id=106 where event_id=7;
update event set booking_id=107 where event_id=8;
update event set booking_id=108 where event_id=9;
update event set booking_id=109 where event_id=10;
create table customer
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
1	Bollywood Night Extravaganza	2024-06-17	18:30:00	1	50	47	250.00	Concert	101
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
3	Classic Movie Marathon	2024-07-21	12:00:00	3	100	95	50.00	Movie	103
4	Live Stand-Up Comedy Show	2024-05-22	19:00:00	4	50	45	120.00	Concert	104
5	Football Championship	2024-05-01	16:30:00	5	100	100	80.00	Sports	NULL
6	Melodious Sufi Night	2024-08-10	20:00:00	6	50	49	200.00	Concert	105
7	Outdoor Movie Night	2024-06-25	19:30:00	7	50	45	75.00	Movie	106
8	Basketball Showdown	2024-07-15	14:00:00	8	150	145	60.00	Sports	107
9	Folk Music Festival	2024-09-03	17:00:00	9	50	47	150.00	Concert	108
10	Tennis Championship asian cup	2024-09-11	12:30:00	10	200	190	1000.00	Sports	109
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customer:

```
INSERT INTO Customer (customer_id, customer_name, email, phone, booking_id)
VALUES
    (1, 'Aditya Verma', 'aditya@gmail.com', '1234567890', null),
    (2, 'Sneha Patel', 'sneha@gmail.com', '1234567891', null),
    (3, 'Aryan Singh', 'aryan@gmail.com', '1234567892', null),
    (4, 'Kavita Gupta', 'kavita@gmail.com', '1234567893', null),
    (5, 'Rahul Sharma', 'rahul@gmail.com', '1234567894', null),
    (6, 'Priya Mishra', 'priya@gmail.com', '1234567895', null),
    (7, 'Vikram Malhotra', 'vikram@gmail.com', '1234567896', null),
    (8, 'Anjali sharma', 'anjali@gmail.com', '1234567000', null),
    (9, 'Neha Kapoor', 'neha@gmail.com', '1234567898', null),
    (10, 'Ravi Verma', 'ravi@gmail.com', '1134567000', null),
    (11,"akash devaki","akash@gmail.com","1478523691",null),
    (12,"sai teja","sai@gmail.com","1478523692",null);

update customer set booking_id=101 where customer_id=1;
update customer set booking_id=102 where customer_id=2;
update customer set booking_id=103 where customer_id=3;
update customer set booking_id=104 where customer_id=4;
update customer set booking_id=105 where customer_id=6;
update customer set booking_id=106 where customer_id=7;
```

```sql
update customer set booking_id=106 where customer_id=7;
update customer set booking_id=107 where customer_id=8;
update customer set booking_id=108 where customer_id=9;
update customer set booking_id=109 where customer_id=10;
select * from customer;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

customer_id	customer_name	email	phone	booking_id
1	Aditya Verma	aditya@gmail.com	1234567890	101
2	Sneha Patel	sneha@gmail.com	1234567891	102
3	Aryan Singh	aryan@gmail.com	1234567892	103
4	Kavita Gupta	kavita@gmail.com	1234567893	104
5	Rahul Sharma	rahul@gmail.com	1234567894	NULL
6	Priya Mishra	priya@gmail.com	1234567895	105
7	Vikram Malhotra	vikram@gmail.com	1234567896	106
8	Anjali sharma	anjali@gmail.com	1234567000	107
9	Neha Kapoor	neha@gmail.com	1234567898	108
10	Ravi Verma	ravi@gmail.com	1134567000	109
11	akash devaki	akash@gmail.com	1478523691	NULL
12	sai teja	sai@gmail.com	1478523692	NULL
NULL	NULL	NULL	NULL	NULL

Booking:

```sql
INSERT INTO Booking (booking_id, customer_id, event_id, num_tickets, total_cost, booking_date)
VALUES
    (101, 1, 1, 3, 750.00, '2024-01-10'),
    (102, 2, 2, 20, 48000.00, '2024-01-12'),
    (103, 3, 3, 5, 250.00, '2024-01-15'),
    (104, 4, 4, 5, 600.00, '2024-01-17'),
    (105, 6, 6, 1, 200.00, '2024-02-20'),
    (106, 7, 7, 5, 375.00, '2024-02-25'),
    (107, 8, 8, 5, 300.00, '2024-03-05'),
    (108, 9, 9, 3, 450.00, '2024-03-07'),
    (109, 10, 10, 10, 10000.00, '2024-03-10');
select * from booking;

    /*task 2 question 3*/
    select * from event where avaliable_seats <> 0;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

booking_id	customer_id	event_id	num_tickets	total_cost	booking_date
101	1	1	3	750.00	2024-01-10
102	2	2	20	48000.00	2024-01-12
103	3	3	5	250.00	2024-01-15
104	4	4	5	600.00	2024-01-17
105	6	6	1	200.00	2024-02-20
106	7	7	5	375.00	2024-02-25
107	8	8	5	300.00	2024-03-05
108	9	9	3	450.00	2024-03-07
109	10	10	10	10000.00	2024-03-10
NULL	NULL	NULL	NULL	NULL	NULL

2, Write a SQL query to list all Events.

```
select * from event;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
1	Bollywood Night Extravaganza	2024-02-15	18:30:00	1	500	300	250.00	Concert	101
2	Cricket Tournament Finals world cup	2024-02-20	15:00:00	2	1600	500	2600.00	Sports	102
3	Classic Movie Marathon	2024-03-05	12:00:00	3	300	150	50.00	Movie	103
4	Live Stand-Up Comedy Show	2024-03-10	19:00:00	4	200	0	120.00	Concert	104
5	Football Championship	2024-03-15	16:30:00	5	800	400	80.00	Sports	105
6	Melodious Sufi Night	2024-03-20	20:00:00	6	400	250	200.00	Concert	106
7	Outdoor Movie Night	2024-04-05	19:30:00	7	150	0	75.00	Movie	107
8	Basketball Showdown	2024-04-10	14:00:00	8	500	150	60.00	Sports	108
9	Folk Music Festival	2024-04-15	17:00:00	9	300	50	150.00	Concert	109
10	Tennis Championship asian cup	2024-04-20	12:30:00	10	600	100	1000.00	Sports	110
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3, Write a SQL query to select events with available tickets

```
select * from event where avaliable_seats <> 0;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
1	Bollywood Night Extravaganza	2024-06-17	18:30:00	1	50	47	250.00	Concert	101
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
3	Classic Movie Marathon	2024-07-21	12:00:00	3	100	95	50.00	Movie	103
4	Live Stand-Up Comedy Show	2024-05-22	19:00:00	4	50	45	120.00	Concert	104
5	Football Championship	2024-05-01	16:30:00	5	100	100	80.00	Sports	NULL
6	Melodious Sufi Night	2024-08-10	20:00:00	6	50	49	200.00	Concert	105
7	Outdoor Movie Night	2024-06-25	19:30:00	7	50	45	75.00	Movie	106
8	Basketball Showdown	2024-07-15	14:00:00	8	150	145	60.00	Sports	107
9	Folk Music Festival	2024-09-03	17:00:00	9	50	47	150.00	Concert	108
10	Tennis Championship asian cup	2024-09-11	12:30:00	10	200	190	1000.00	Sports	109
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

event 3

4, Write a SQL query to select events name partial match with ‘cup’.

```
select * from Event where event_name LIKE '%cup%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
10	Tennis Championship asian cup	2024-09-11	12:30:00	10	200	190	1000.00	Sports	109
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
select * from Event where ticket_price between 1000 and 2500;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
10	Tennis Championship asian cup	2024-09-11	12:30:00	10	200	190	1000.00	Sports	109
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
select * from Event where event_date between '2024-05-01' and '2024-06-15';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
4	Live Stand-Up Comedy Show	2024-05-22	19:00:00	4	50	45	120.00	Concert	104
5	Football Championship	2024-05-01	16:30:00	5	100	100	80.00	Sports	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
select * from event where avaliable_seats <> 0 and event_name="%concert%";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Here we got this output because there is no event name which has concert in their name.

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;
```

Result Grid | Filter Rows: | Edit: | Export

customer_id	customer_name	email	phone	booking_id
6	Priya Mishra	priya@gmail.com	1234567895	105
7	Vikram Malhotra	vikram@gmail.com	1234567896	106
8	Anjali sharma	anjali@gmail.com	1234567000	107
9	Neha Kapoor	neha@gmail.com	1234567898	108
10	Ravi Verma	ravi@gmail.com	1134567000	109
NULL	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 num_tickets > 4;
```

Result Grid | Filter Rows: | Edit: | Export/Impor

booking_id	customer_id	event_id	num_tickets	total_cost	booking_date
102	2	2	20	48000.00	2024-01-12
103	3	3	5	250.00	2024-01-15
104	4	4	5	600.00	2024-01-17
106	7	7	5	375.00	2024-02-25
107	8	8	5	300.00	2024-03-05
109	10	10	10	10000.00	2024-03-10
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 phone like"%000";

/*task 2 question 11*/
```

Result Grid | Filter Rows: | Edit: | Export/Im

customer_id	customer_name	email	phone	booking_id
8	Anjali sharma	anjali@gmail.com	1234567000	107
10	Ravi Verma	ravi@gmail.com	1134567000	109
NULL	NULL	NULL	NULL	NULL

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

```
select * from event where total_seats>15000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

12, Write a SQL query to select events name not start with ‘x’, ‘y’, ‘z’

```
select * from Event where event_name not like 'x%' and event_name not like 'y%'
and event_name not like 'z%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
1	Bollywood Night Extravaganza	2024-06-17	18:30:00	1	50	47	250.00	Concert	101
2	Cricket Tournament Finals world cup	2024-05-20	15:00:00	2	15500	15480	2400.00	Sports	102
3	Classic Movie Marathon	2024-07-21	12:00:00	3	100	95	50.00	Movie	103
4	Live Stand-Up Comedy Show	2024-05-22	19:00:00	4	50	45	120.00	Concert	104
5	Football Championship	2024-05-01	16:30:00	5	100	100	80.00	Sports	NULL
6	Melodious Sufi Night	2024-08-10	20:00:00	6	50	49	200.00	Concert	105
7	Outdoor Movie Night	2024-06-25	19:30:00	7	50	45	75.00	Movie	106
8	Basketball Showdown	2024-07-15	14:00:00	8	150	145	60.00	Sports	107
9	Folk Music Festival	2024-09-03	17:00:00	9	50	47	150.00	Concert	108
10	Tennis Championship asian cup	2024-09-11	12:30:00	10	200	190	1000.00	Sports	109
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Task – 3

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

```
select Event_ID, Event_Name,
avg(Ticket_Price) as AverageTicketPrice
from Event e group by Event_ID, Event_Name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

Event_ID	Event_Name	AverageTicketPrice
1	Bollywood Night Extravaganza	250.000000
2	Cricket Tournament Finals world cup	2400.000000
3	Classic Movie Marathon	50.000000
4	Live Stand-Up Comedy Show	120.000000
5	Football Championship	80.000000
6	Melodious Sufi Night	200.000000
7	Outdoor Movie Night	75.000000
8	Basketball Showdown	60.000000
9	Folk Music Festival	150.000000
10	Tennis Championship asian cup	1000.000000

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

```
SELECT sum(total_cost) as revenue_generated from booking;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

revenue_generated
60925.00

3, Write a SQL query to find the event with the highest ticket sales.

```
select e.Event_ID,e.Event_Name,SUM(total_Seats-avaliable_seats) as TotalTicketSales
from event e group by e.Event_ID, e.Event_Name order by TotalTicketSales desc
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Event_ID	Event_Name	TotalTicketSales
2	Cricket Tournament Finals world cup	20

4, Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event.

```
select e.event_id,event_name, b.num_tickets as total_tickets_saled  from event e
join booking b on e.booking_id=b.booking_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_id	event_name	total_tickets_saled
1	Bollywood Night Extravaganza	3
2	Cricket Tournament Finals world cup	20
3	Classic Movie Marathon	5
4	Live Stand-Up Comedy Show	5
6	Melodious Sufi Night	1
7	Outdoor Movie Night	5
8	Basketball Showdown	5
9	Folk Music Festival	3
10	Tennis Championship asian cup	10

5, Write a SQL query to Find Events with No Ticket Sales.

```
select * from event where total_seats=avaliable_seats;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

event_id	event_name	event_date	event_time	venue_id	total_seats	avaliable_seats	ticket_price	event_type	booking_id
5	Football Championship	2024-05-01	16:30:00	5	100	100	80.00	Sports	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6, Write a SQL query to Find the User Who Has Booked the Most Tickets.

```sql
select c.customer_name, b.customer_ID, SUM(Num_Tickets) as TotalTicketsBooked
from Booking b join customer c on c.customer_id=b.customer_id
group by customer_id
order by TotalTicketsBooked desc limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_name	customer_ID	TotalTicketsBooked
Sneha Patel	2	20

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

```sql
select b.Event_ID,e.event_name, month(booking_date) as SaleMonth, year(booking_date) as SaleYear,
SUM(Num_Tickets) AS TotalTicketsSold FROM booking b join event e on e.event_id=b.event_id
group by Event_ID, SaleMonth, SaleYear order by SaleYear, SaleMonth, Event_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Event_ID	event_name	SaleMonth	SaleYear	TotalTicketsSold
1	Bollywood Night Extravaganza	1	2024	3
2	Cricket Tournament Finals world cup	1	2024	20
3	Classic Movie Marathon	1	2024	5
4	Live Stand-Up Comedy Show	1	2024	5
6	Melodious Sufi Night	2	2024	1
7	Outdoor Movie Night	2	2024	5
8	Basketball Showdown	3	2024	5
9	Folk Music Festival	3	2024	3
10	Tennis Championship asian cup	3	2024	10

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

```
select e.Venue_ID,v.Venue_Name,avg(t.total_cost ) as AverageTicketPrice
from event e join booking t on e.Event_ID = t.Event_ID
join Venue_table v on e.Venue_ID = v.Venue_ID
group by e.Venue_ID, v.Venue_Name order by e.Venue_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Venue_ID	Venue_Name	AverageTicketPrice
1	Royal Palace Banquet Hall	750.000000
2	Green Valley Gardens	48000.000000
3	Grand Plaza Convention Center	250.000000
4	Elegant Halls and Events	600.000000
6	Sapphire Wedding Venues	200.000000
7	Golden Sands Resort	375.000000
8	Crystal Gardens	300.000000
9	Heritage Hall	450.000000
10	Starlight Convention Center	10000.000000

9, 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 tickets from event e join booking b on e.event_id=b.event_id
group by event_type order by event_type;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_type	tickets
Concert	12
Movie	10
Sports	35

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

```
select year(Event_Date) as EventYear,
SUM((total_seats-avaliable_Seats)*ticket_price) as TotalRevenue
from event group by year(Event_Date) order by EventYear;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EventYear	TotalRevenue
2024	60925.00

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

```
select customer_ID, COUNT(distinct Event_ID) as NumberOfEventsBooked
from Booking group by customer_ID having COUNT(distinct Event_ID) > 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_ID	NumberOfEventsBooked

Here we got no id because there is no customer in the database those who made bookings for multiple events.

12, Write a SQL query to calculate the Total Revenue Generated by Events for Each User

```
select c.customer_id,c.customer_name,b.total_cost from customer c left join booking
b on c.customer_id=b.customer_id join event e on b.event_id=e.event_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name	total_cost
1	Aditya Verma	750.00
2	Sneha Patel	48000.00
3	Aryan Singh	250.00
4	Kavita Gupta	600.00
6	Priya Mishra	200.00
7	Vikram Malhotra	375.00
8	Anjali sharma	300.00
9	Neha Kapoor	450.00
10	Ravi Verma	10000.00

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

Category:

```
204 •   select event_type, avg(ticket_price) from event group by event_type;
205
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_type	avg(ticket_price)
Concert	180.000000
Sports	885.000000
Movie	62.500000

Venue:

```
205 •   select v.venue_name, v.venue_id,avg(e.ticket_price) from venue_table v join event e on e.venue_id=v.venue_id group by venue_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

venue_name	venue_id	avg(e.ticket_price)
Royal Palace Banquet Hall	1	250.000000
Green Valley Gardens	2	2400.000000
Grand Plaza Convention Center	3	50.000000
Elegant Halls and Events	4	120.000000
Celestial Ballroom	5	80.000000
Sapphire Wedding Venues	6	200.000000
Golden Sands Resort	7	75.000000
Crystal Gardens	8	60.000000
Heritage Hall	9	150.000000
Starlight Convention Center	10	1000.000000

14, Write a SQL query to list Users and the Total Number of Tickets They've Purchased in the Last 30 Days

```
SELECT customer_id,SUM(Num_Tickets) AS TotalTicketsPurchased
FROM booking WHERE booking_Date >= curdate() - interval 30 day
GROUP BY customer_id ORDER BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	TotalTicketsPurchased
1	3
2	20
3	5
4	5
6	1
7	5
8	5
9	3
10	10

Task-4

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

```
SELECT
    v.Venue_ID,
    v.Venue_Name,
    (SELECT AVG(total_cost) FROM booking t WHERE t.Event_ID IN (SELECT Event_ID FROM Event e WHERE e.Venue_ID = v.Venue_ID))
    AS AverageTicketPrice
FROM
    Venue_table v;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Venue_ID	Venue_Name	AverageTicketPrice
1	Royal Palace Banquet Hall	750.000000
2	Green Valley Gardens	48000.000000
3	Grand Plaza Convention Center	250.000000
4	Elegant Halls and Events	600.000000
5	Celestial Ballroom	NULL
6	Sapphire Wedding Venues	200.000000
7	Golden Sands Resort	375.000000
8	Crystal Gardens	300.000000
9	Heritage Hall	450.000000
10	Starlight Convention Center	10000.000000

2, Find Events with More Than 50% of Tickets Sold using subquery.

```
SELECT
    Event_ID,
    Event_Name,
    sum(total_Seats-avaliable_seats) as tickets_sold,
    Total_seats,
    ((total_seats-avaliable_Seats) / Total_Seats) * 100 AS PercentageSold
FROM
    Event
WHERE
    ((total_seats-avaliable_Seats) / Total_Seats) * 100 > 50
group by Event_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Event_ID	Event_Name	tickets_sold	Total_seats	PercentageSold

Here we got this output because there is no event which has sold 50% of their tickets so we didn't get any event name as output.

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

```
select  e.event_id,COALESCE(b.num_tickets,0) as totaltickets from event e left join booking b on  e.event_id=b.event_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_id	totaltickets
1	3
2	20
3	5
4	5
5	0
6	1
7	5
8	5
9	3
10	10

4, Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery.

```
select customer_ID,customer_Name from customer c
where not exists (select 1 from Booking b where b.customer_ID = c.customer_ID);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_ID	customer_Name
5	Rahul Sharma
11	akash devaki
12	sai teja

5, List Events with No Ticket Sales Using a NOT IN Subquery.

```
select event_ID,event_Name from event e
where not exists (select 1 from Booking b where b.event_ID = e.event_ID);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_ID	event_Name
5	Football Championship

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

```
select e.event_type,(select COALESCE(SUM(b.Num_Tickets), 0)
from booking b where e.event_id=b.event_id) as TotalTicketsSold From event e  group by event_type;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_type	TotalTicketsSold
Concert	3
Sports	20
Movie	5

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

```
select event_id,event_name,ticket_price from event
where ticket_price > (select avg(ticket_price) from event);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

event_id	event_name	ticket_price
2	Cricket Tournament Finals world cup	2400.00
10	Tennis Championship asian cup	1000.00
NULL	NULL	NULL

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

```
select c.customer_ID,c.customer_Name,
(
        select COALESCE(SUM(b.total_cost), 0)
        from event e
        join booking b on e.Event_ID = b.Event_ID
        where b.customer_ID = c.customer_id
    ) as TotalRevenue
from customer c;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

customer_ID	customer_Name	TotalRevenue
1	Aditya Verma	750.00
2	Sneha Patel	48000.00
3	Aryan Singh	250.00
4	Kavita Gupta	600.00
5	Rahul Sharma	0.00
6	Priya Mishra	200.00
7	Vikram Malhotra	375.00
8	Anjali sharma	300.00
9	Neha Kapoor	450.00
10	Ravi Verma	10000.00
11	akash devaki	0.00
12	sai teja	0.00

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

```
select c.customer_ID,c.customer_Name from customer c
where exists (select 1 from Booking b
join event e on b.Event_ID = e.Event_ID
where b.customer_ID = c.customer_ID and e.Venue_ID = 3 );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_ID	customer_Name
3	Aryan Singh

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

```
select e.event_type,(select COALESCE(SUM(b.Num_Tickets), 0)
from booking b where e.event_id=b.event_id) as TotalTicketsSold From event e group by event_id, event_type ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

event_type	TotalTicketsSold
Concert	3
Sports	20
Movie	5
Concert	5
Sports	0
Concert	1
Movie	5
Sports	5
Concert	3
Sports	10

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

```
select c.customer_id,c.customer_Name,DATE_FORMAT(b.booking_date, '%Y-%m') as PurchaseMonth
from customer c join Booking b on c.customer_id = b.customer_id
join Event e on b.Event_ID = e.Event_ID
where DATE_FORMAT(b.booking_date, '%Y-%m') IN (
select distinct  DATE_FORMAT(booking_date, '%Y-%m') as BookingMonth from Booking b)
order by c.customer_id, PurchaseMonth;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_Name	PurchaseMonth
1	Aditya Verma	2024-01
2	Sneha Patel	2024-01
3	Aryan Singh	2024-01
4	Kavita Gupta	2024-01
6	Priya Mishra	2024-02
7	Vikram Malhotra	2024-02
8	Anjali sharma	2024-03
9	Neha Kapoor	2024-03
10	Ravi Verma	2024-03

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

```
select v.Venue_ID,v.Venue_Name,
(select COALESCE(avg(e.Ticket_Price), 0) from event e
join booking b on e.Event_ID = b.Event_ID
where e.Venue_ID = v.Venue_ID) as AverageTicketPrice
from Venue_table v;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Venue_ID	Venue_Name	AverageTicketPrice
1	Royal Palace Banquet Hall	250.000000
2	Green Valley Gardens	2400.000000
3	Grand Plaza Convention Center	50.000000
4	Elegant Halls and Events	120.000000
5	Celestial Ballroom	0.000000
6	Sapphire Wedding Venues	200.000000
7	Golden Sands Resort	75.000000
8	Crystal Gardens	60.000000
9	Heritage Hall	150.000000
10	Starlight Convention Center	1000.000000

Submitted By:

Devaki Akash