Hexaware Assignment-Ticket Booking

Tasks 1: Database Design:

1.Create the database named "TicketBookingSystem"

Create database Ticketbooking;

2. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

- **Venu**
- **Event**
- **Customers**
- **Booking**

Venu:

```
create table Venu(
venue_id varchar(20) PRIMARY KEY,
venue_name varchar(50),
address varchar(20));
```

Event:

```
create table Events_list(
event_id varchar(10) PRIMARY KEY,
event_name varchar(50),
event_date DATE,event_time TIME,
venue_id varchar(20),
total_seats int,
available_seats int,
ticket_price decimal(10,2),
event_type varchar(10),
booking_id varchar(10),
foreign key(venue_id)references Venu(venue_id));
```

Customer:

```
create table Customer(
customer_id varchar(10) PRIMARY KEY,
customer_name varchar(25),email varchar(30),
phone_number bigint,booking_id varchar(10),
```

foreign key(booking_id)references Booking(booking_id));

Booking:

Create table booking(

booking_id varchar(10) PRIMARY KEY,

event_id varchar(20),

num_tickets int

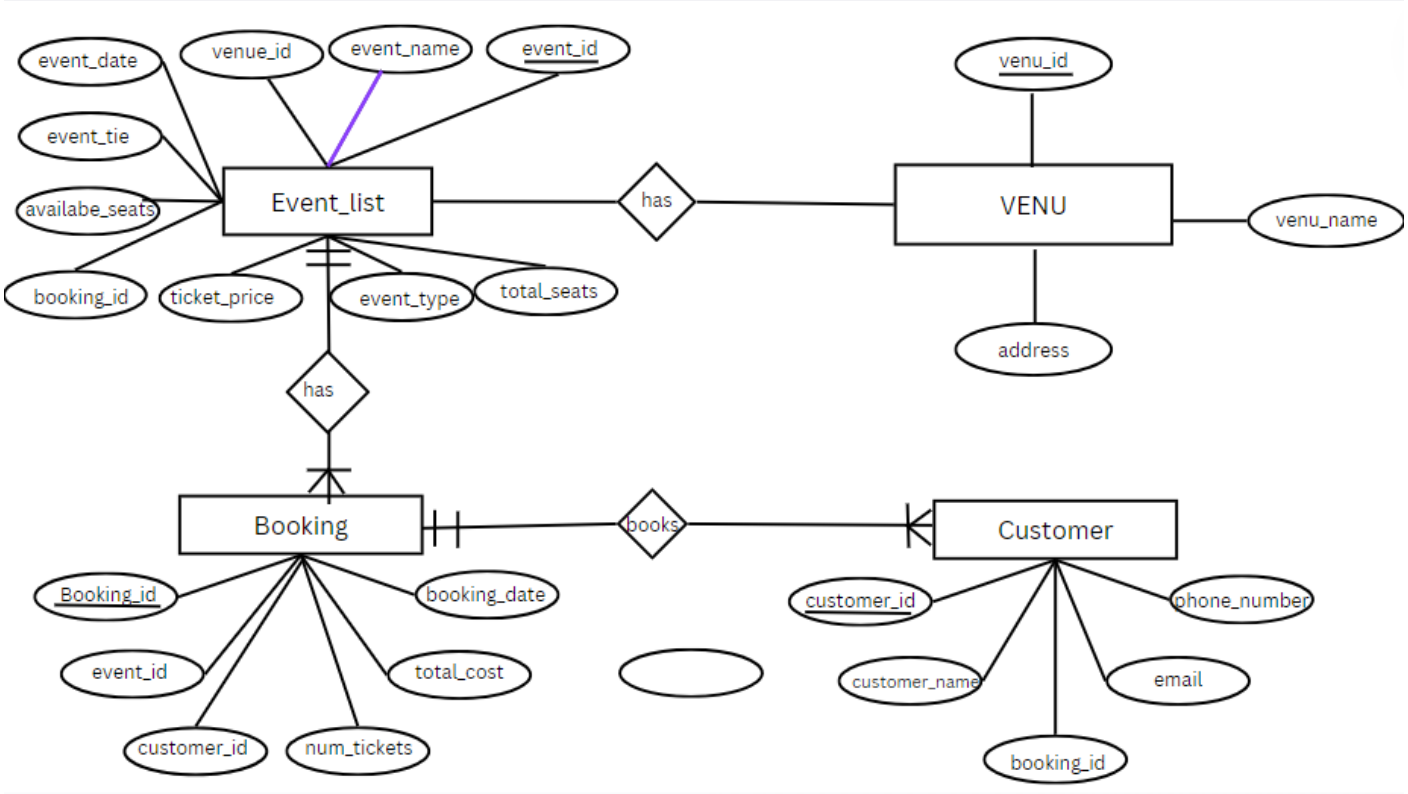
,total_cost decimal(10,2) ,

booking_date DATE,

customer_id varchar (10),

foreign key(event_id)references Events_list(event_id));

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



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

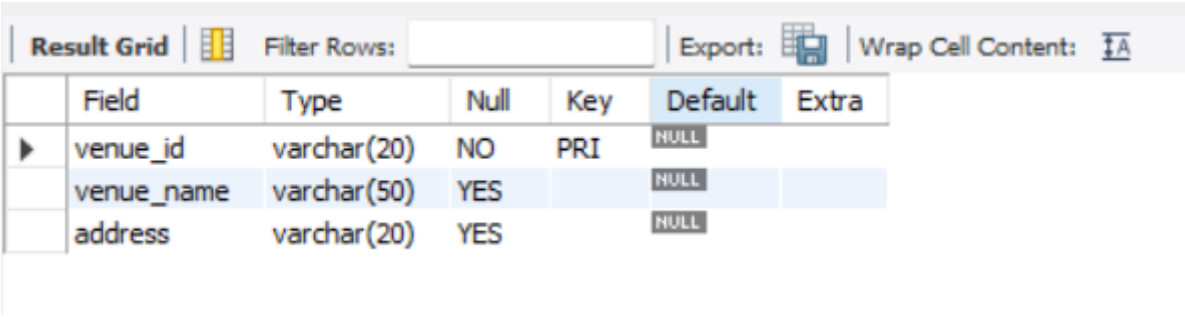
Venu table:

create table Venu(

venue_id varchar(20) PRIMARY KEY,

venue_name varchar(50),

address varchar(20));



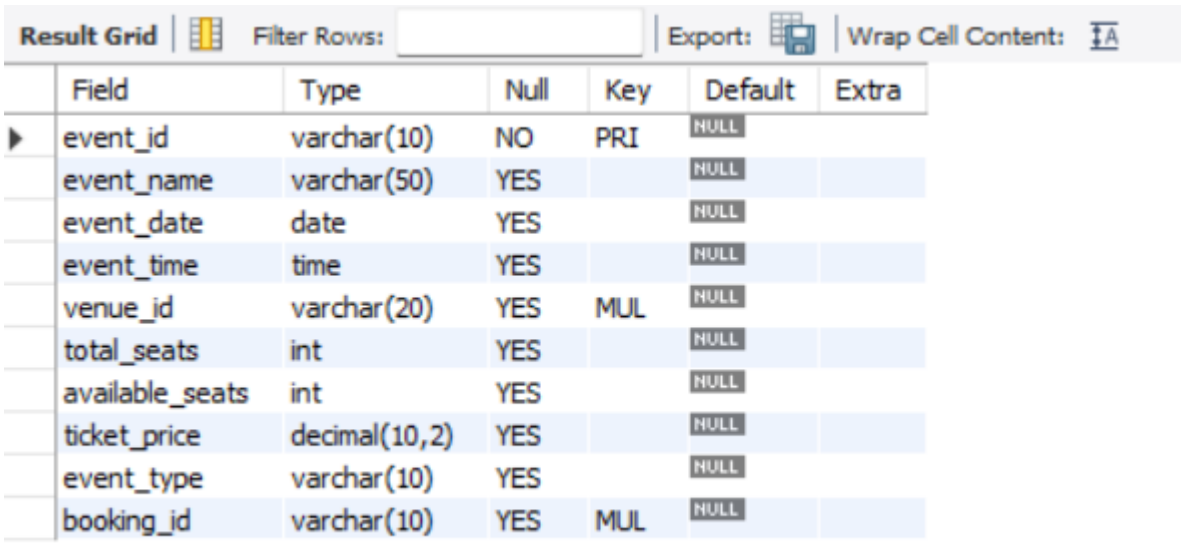
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
venue_id	varchar(20)	NO	PRI	NULL	
venue_name	varchar(50)	YES		NULL	
address	varchar(20)	YES		NULL	

Events_list:

```
create table Events_list(
event_id varchar(10) PRIMARY KEY,
event_name varchar(50),
event_date DATE,event_time TIME,
 venue_id varchar(20),
 total_seats int,available_seats int,
 ticket_price decimal(10,2),
 event_type varchar(10),
booking_id varchar(10),
foreign key(venue_id)references Venu(venue_id));
```

```
ALTER TABLE Events_list
ADD CONSTRAINT addforeignkey
FOREIGN KEY (booking_id) REFERENCES Booking(Booking_id);
```

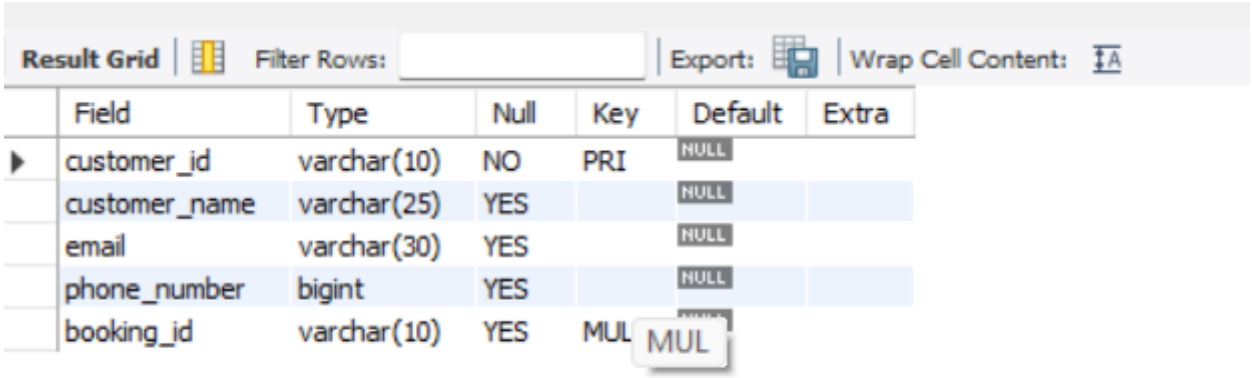


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
event_id	varchar(10)	NO	PRI	NULL	
event_name	varchar(50)	YES		NULL	
event_date	date	YES		NULL	
event_time	time	YES		NULL	
venue_id	varchar(20)	YES	MUL	NULL	
total_seats	int	YES		NULL	
available_seats	int	YES		NULL	
ticket_price	decimal(10,2)	YES		NULL	
event_type	varchar(10)	YES		NULL	
booking_id	varchar(10)	YES	MUL	NULL	

Customer:

```
create table Customer(
customer_id varchar(10) PRIMARY KEY,
customer_name varchar(25),email varchar(30),
phone_number bigint,booking_id varchar(10),
foreign key(booking_id)references Booking(booking_id));
```

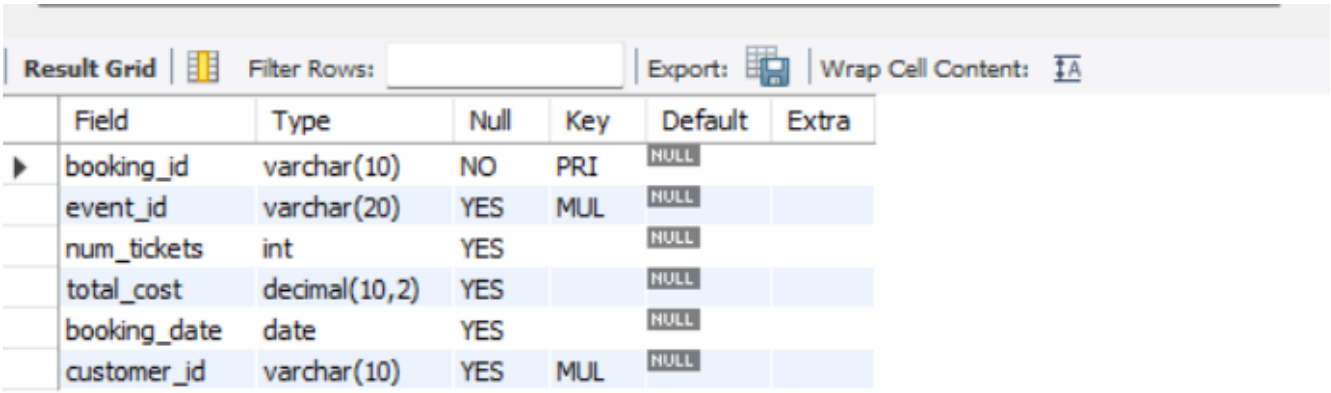


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
customer_id	varchar(10)	NO	PRI	NULL	
customer_name	varchar(25)	YES		NULL	
email	varchar(30)	YES		NULL	
phone_number	bigint	YES		NULL	
booking_id	varchar(10)	YES	MUL		

Booking:

```
Create table booking(
booking_id varchar(10) PRIMARY KEY,
event_id varchar(20),
num_tickets int
,total_cost decimal(10,2) ,
booking_date DATE,
customer_id varchar (10),
foreign key(event_id)references Events_list(event_id));
ALTER TABLE Booking
ADD CONSTRAINT add1foreignkey
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id);
```



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

Field	Type	Null	Key	Default	Extra
booking_id	varchar(10)	NO	PRI	NULL	
event_id	varchar(20)	YES	MUL	NULL	
num_tickets	int	YES		NULL	
total_cost	decimal(10,2)	YES		NULL	
booking_date	date	YES		NULL	
customer_id	varchar(10)	YES	MUL	NULL	

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

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

Venu Table:

```
Insert into Venu values
('A', 'Allan Batmintion club','Tirunelveli'),
('B', 'Kasi theatre','Chennai'),
('C','EVP Palace','Chennai'),
('D','Jazz cinemas', 'Tripur'),
('E','Madrass cricket club','Madurai'),
('F','GK cinemas','Salem'),
('G', 'Ashok sports club','Chennai'),
('H','Nani mahal','Chennai'),
('I','VMM sports','Madurai'),
('J','Chepauk Stadium','Chennai');
```

Events_list:

```
INSERT INTO Events_list VALUES
('e1','Batminton_cup_2026','2026-04-03','9:00:00','A',800,250,500.00,'Sports','aaa'),
('e2','Evaru','2024-04-26','11:30:00','B',700,275,250.00,'Movies','bbb'),
('e3','Harmony Fest of concert','2025-05-06','20:00:00','C',10000,3400,1100.00,'Music','ccc'),
('e4','Clap','2024-05-27','15:00:00','D',600,150,180.00,'Movies','ddd'),
('e5','Cricket_cup_2026','2026-06-12','20:30:00','E',30000,16040,1440.00,'Sports','fff'),
('e6','X_Men','2024-04-28','18:00:00','F',750,200,230.00,'Movies','hhh'),
('e7','Zorb_Football','2024-05-29','19:00:00','G',50000,49997,2450.00,'Sports','iii'),
('e8','Rhythm Rave concert','2024-05-03','18:30:00','H',13000,2350,1500.00,'Music','kkk'),
('e9','Zenith concert,2025','2025-04-24','19:30:00','I',17500,3500,500.00,'Music','qqq'),
('e0','world_cup_match','2024-05-16','21:30:00','J',45000,24000,2500.00,'Sports','mmm');
```

Customer:

```
INSERT INTO Customer VALUES
('111','Raj','raj@gmail.com',2345561878,'aaa'),
('222','Ram','ram@gmail.com',9798467325,'bbb'),
('333','Rajesh','rajesh@gmail.com',7676489000,'ccc'),
```

('444','Rohan','rohan@gmail.com',9009745354,'ddd'),
('555','keerthi','keerthi@gmail.com',6652454998,'fff'),
('666','suba','suba@gmail.com',998734267,'hhh'),
('777','sri','sri@gmail.com',9790292000,'iii'),
('888','Vaishnavi','vaishnavi@gmail.com',8097655287,'kkk'),
('999','sam','sam@gmail.com',9809645324,'qqq'),
('110','Rhythm','rhythm@gmail.com',9876123000,'mmm'),
('112','Arun','arun@gmail.com',98776753000,'eee'),
('113','Ronish','ronish@gmail.com',9865653800,'jjj');

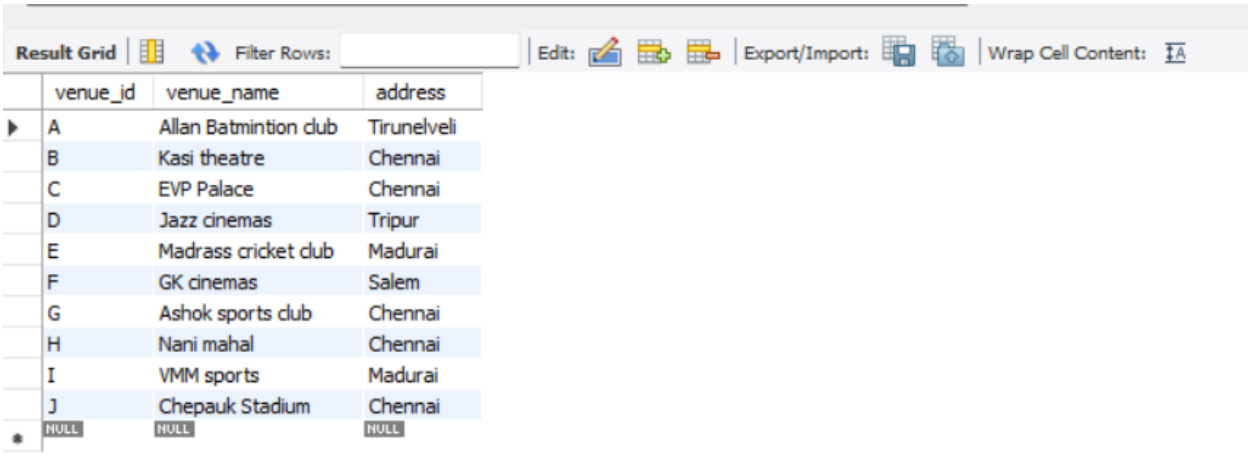
Booking:

INSERT INTO booking VALUES
('aaa', 'e1', 5, 2500.00, '2026-04-03','111'),
('bbb', 'e2', 4, 1000.00, '2024-04-26','222'),
('ccc', 'e3', 3, 3300.00, '2025-05-06','333'),
('ddd', 'e4', 6, 1080.00, '2024-05-27','444'),
('eee','e4',3,540.00,'2024-05-27','112'),
('fff', 'e5', 2, 2880.00, '2026-06-12','555'),
('hhh', 'e6', 1, 230.00, '2024-04-28','666'),
('iii', 'e7', 3, 7350.00, '2024-05-29','777'),
('jjj', 'e8', 2, 3000.00, '2024-05-03','113'),
('kkk', 'e8', 1, 1500.00, '2024-05-03','888'),
('qqq', 'e9', 2, 1000.00, '2025-04-24','999'),
('mmm', 'e0', 4, 12500.00, '2024-05-16','110');

2. **Write a SQL query to list all Events.**

Venu Table

Select * from Venu;



venue_id	venue_name	address
A	Allan Batmintion club	Tirunelveli
B	Kasi theatre	Chennai
C	EVP Palace	Chennai
D	Jazz cinemas	Tripur
E	Madrass cricket club	Madurai
F	GK cinemas	Salem
G	Ashok sports club	Chennai
H	Nani mahal	Chennai
I	VMM sports	Madurai
J	Chepauk Stadium	Chennai
NULL	NULL	NULL

Events_list Table:

Select *from Events_list;

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type	booking_id
e0	world_cup_match	2024-05-16	21:30:00	J	45000	24000	2500.00	Sports	mmm
e1	Batminton_cup_2026	2026-04-03	09:00:00	A	800	250	500.00	Sports	aaa
e2	Evaru	2024-04-03	11:30:00	B	700	275	250.00	Movies	bbb
e3	Harmony Fest of concert	2025-05-06	20:00:00	C	10000	3400	1100.00	Music	ccc
e4	Clap	2024-05-27	15:00:00	D	600	150	180.00	Movies	ddd
e5	Cricket_cup_2026	2026-06-12	20:30:00	E	30000	16040	1440.00	Sports	fff
e6	X_Men	2024-04-28	18:00:00	F	750	200	230.00	Movies	hhh
e7	Zorb_Football	2024-05-29	19:00:00	G	50000	49997	2450.00	Sports	iii
e8	Rhythm Rave concert	2024-05-03	18:30:00	H	13000	2350	1500.00	Music	kkk
e9	Zenith concert,2025	2025-04-24	19:30:00	I	17500	3500	500.00	Music	qqq
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customer Table:

Select * from Customer;

customer_id	customer_name	email	phone_number	booking_id
110	Rhythm	rhythm@gmail.com	9876123000	mmm
111	Raj	raj@gmail.com	2345561878	aaa
112	Arun	arun@gmail.com	98776753000	eee
113	Ronish	ronish@gmail.com	9865653800	jjj
222	Ram	ram@gmail.com	9798467325	bbb
333	Rajesh	rajesh@gmail.com	7676489000	ccc
444	Rohan	rohan@gmail.com	9009745354	ddd
555	keerthi	keerthi@gmail.com	6652454998	fff
666	suba	suba@gmail.com	998734267	hhh
777	sri	sri@gmail.com	9790292000	iii
888	Vaishnavi	vaishnavi@gmail.c...	8097655287	kkk
999	sam	sam@gmail.com	9809645324	qqq
NULL	NULL	NULL	NULL	NULL

Booking Table

Select * from Booking;

booking_id	event_id	num_tickets	total_cost	booking_date	customer_id
aaa	e1	5	2500.00	2026-04-03	111
bbb	e2	4	1000.00	2024-04-03	222
ccc	e3	3	3300.00	2025-05-06	333
ddd	e4	6	1080.00	2024-05-27	444
eee	e4	3	540.00	2024-05-27	112
fff	e5	2	2880.00	2026-06-12	555
hhh	e6	1	230.00	2024-04-28	666
iii	e7	3	7350.00	2024-05-29	777
jjj	e8	2	3000.00	2024-05-03	113
kkk	e8	1	1500.00	2024-05-03	888
mmm	e0	4	12500.00	2024-05-16	110
qqq	e9	2	1000.00	2025-04-24	999
NULL	NULL	NULL	NULL	NULL	NULL

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

select event_name , available_seats from Events_list;

event_name	available_seats
world_cup_match	24000
Batminton_cup_2026	250
Evaru	275
Harmony Fest of concert	3400
Clap	150
Cricket_cup_2026	16040
X_Men	200
Zorb_Football	49997
Rhythm Rave concert	2350
Zenith concert,2025	3500

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

select event_name from Events_list where event_name like '%cup%';

event_name
world_cup_match
Batminton_cup_2026
Cricket_cup_2026

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

select event_name , ticket_price from Events_list where ticket_price between 1000 and 2500;

event_name	ticket_price
world_cup_match	2500.00
Harmony Fest of concert	1100.00
Cricket_cup_2026	1440.00
Zorb_Football	2450.00
Rhythm Rave concert	1500.00

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

select event_name ,event_date from Events_list where event_date between '2024-04-01' and '2024-04-30';

event_name	event_date
world_cup_match	2024-05-16
Evaru	2024-04-03
Clap	2024-05-27
X_Men	2024-04-28
Zorb_Football	2024-05-29
Rhythm Rave concert	2024-05-03

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

select event_name , available_seats from Events_list where event_name like'%concert%';

event_name	available_seats
Harmony Fest of concert	3400
Rhythm Rave concert	2350
Zenith concert,2025	3500

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

select customer_name from Customer limit 5 offset 5;

customer_name
Rajesh
Rohan
keerthi
suba
sri

9.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	num_tickets	total_cost	booking_date	customer_id
aaa	e1	5	2500.00	2026-04-16	111
ddd	e4	6	1080.00	2024-05-27	444
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_number like '%000';

customer_id	customer_name	email	phone_number	booking_id
110	Rhythm	rhythm@gmail.com	9876123000	mmm
112	Arun	arun@gmail.com	98776753000	eee
333	Rajesh	rajesh@gmail.com	7676489000	ccc
777	sri	sri@gmail.com	9790292000	iii
NULL	NULL	NULL	NULL	NULL

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

select event_name from Events_list where total_seats >15000;

	event_name
▶	world_cup_match
	Cricket_cup_2026
	Zorb_Football
	Zenith concert,2025

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

select event_name from Events_list where event_name not like'X%' and event_name not like 'Y%' and event_name not like 'Z%';

	event_name
▶	world_cup_match
	Batminton_cup_2026
	Evaru
	Harmony Fest of concert
	Clap
	Cricket_cup_2026
	Rhythm Rave concert

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

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

select event_type ,avg(ticket_price) as average_ticketprice from Events_list GROUP BY event_type;

	event_type	average_ticketprice
▶	Sports	1722.500000
	Movies	220.000000
	Music	1033.333333

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

select sum((total_seats-available_seats)*(ticket_price)) as total_revenue from Events_list;

	total_revenue
▶	103433500.00

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

select event_name,(total_seats-available_seats)as highest_ticket_sales from events_list ORDER BY(total_seats-available_seats) desc limit 1;

event_name	highest_ticket_sales
world_cup_match	21000

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

select event_name , (total_seats-available_seats) as total_seats_sold from Events_list;

event_name	total_seats_sold
world_cup_match	21000
Batminton_cup_2024	550
Evaru	425
Harmony Fest of concert	6600
Clap	450
Cricket_cup_2024	13960
X_Men	550
Zorb_Football	3
Rhythm Rave concert	10650
Zenith concert,2024	14000

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

select event_name,event_id from Events_list having event_id not in(select distinct event_id from Booking);

event_name	event_id
NULL	NULL

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

select b.customer_id,c.customer_name,b.num_tickets from customer c inner join booking b

on c.customer_id = b.customer_id

having b.num_tickets = (select max(num_tickets) from booking);

customer_id	customer_name	num_tickets
444	Rohan	6

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

```
select e.event_name,MONTH (b.booking_date) as month,YEAR (b.booking_date) as year
,sum(b.num_tickets) as tickets_booked
from Events_list e join Booking b on e.event_id = b.event_id
group by month(b.booking_date),year(b.booking_date),e.event_name
order by year asc,month asc;
```

event_name	month	year	tickets_booked
Evaru	4	2024	4
X_Men	4	2024	1
world_cup_match	5	2024	4
Clap	5	2024	9
Zorb_Football	5	2024	3
Rhythm Rave concert	5	2024	3
Zenith concert,2025	4	2025	2
Harmony Fest of concert	5	2025	3
Batminton_cup_2026	4	2026	5
Cricket_cup_2026	6	2026	2

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

```
select v.venue_name,e.event_name,avg(e.ticket_price) as average_of_ticketprice
from Venu v join Events_list e
on v.venue_id=e.venue_id
group by event_name;
```

venue_name	event_name	average_of_ticketprice
Allan Batmintion club	Batminton_cup_2026	500.000000
Kasi theatre	Evaru	250.000000
EVP Palace	Harmony Fest of concert	1100.000000
Jazz cinemas	Clap	180.000000
Madrass cricket club	Cricket_cup_2026	1440.000000
GK cinemas	X_Men	230.000000
Ashok sports club	Zorb_Football	2450.000000
Nani mahal	Rhythm Rave concert	1500.000000
VMM sports	Zenith concert,2025	500.000000
Chepauk Stadium	world_cup_match	2500.000000

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

```
select event_type,sum(total_seats-available_seats) as total_seats_sold from Events_list
```

group by event_type;

event_type	total_seats_sold
Sports	35513
Movies	1425
Music	31250

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

```
select   YEAR(event_date) as Year,sum((total_seats-available_seats)*ticket_price)as Total_Revenue
from Events_list
group by year(event_date)
order by year asc;
```

Year	Total_Revenue
2024	68796100.00
2025	14260000.00
2026	20377400.00

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

select customer_ID from booking group by customer_ID having count(*)>1;

customer_ID

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

```
select b.event_id,c.customer_id,c.customer_name , b.total_cost from Customer c
join Booking b
on c.customer_id=b.customer_id
group by b.event_id;
```

event_id	customer_id	customer_name	total_cost
e0	110	Rhythm	12500.00
e1	111	Raj	2500.00
e4	112	Arun	540.00
e8	113	Ronish	3000.00
e2	222	Ram	1000.00
e3	333	Rajesh	3300.00
e5	555	keerthi	2880.00
e6	666	suba	230.00
e7	777	sri	7350.00
e9	999	sam	1000.00

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

```
select e.event_type,e.event_name,v.venue_name ,avg(ticket_price) as average_ticket_price from Events_list e
join Venu von e.venue_id=v.venue_id
group by event_type,evenue_name;
```

event_type	event_name	venue_name	average_ticket_price
Sports	world_cup_match	Chepauk Stadium	2500.000000
Sports	Batminton_cup_2026	Allan Batmintion club	500.000000
Movies	Evaru	Kasi theatre	250.000000
Music	Harmony Fest of concert	EVP Palace	1100.000000
Movies	Clap	Jazz cinemas	180.000000
Sports	Cricket_cup_2026	Madrass cricket club	1440.000000
Movies	X_Men	GK cinemas	230.000000
Sports	Zorb_Football	Ashok sports club	2450.000000
Music	Rhythm Rave concert	Nani mahal	1500.000000
Music	Zenith concert,2025	VMM sports	500.000000

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

```
SELECT c.customer_id,c.customer_name,b.booking_date,b.num_tickets
FROM Customer c
JOIN Booking b ON c.booking_id = b.booking_id
WHERE b.booking_date >= CURRENT_DATE - INTERVAL '30' DAY and b.booking_date<current_date
ORDER BY b.booking_date;
```

customer_id	customer_name	booking_date	num_tickets
222	Ram	2024-04-03	4

Tasks 4: Subquery and its types

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

```
select v.venue_name,(select avg(ticket_price) from events_list e where e.venue_id =
 v.venue_id) as avg_ticket_price from Venu v;
```

venue_name	avg_ticket_price
Allan Batmintion club	500.000000
Kasi theatre	250.000000
EVP Palace	1100.000000
Jazz cinemas	180.000000
Madrass cricket club	1440.000000
GK cinemas	230.000000
Ashok sports club	2450.000000
Nani mahal	1500.000000
VMM sports	500.000000
Chepauk Stadium	2500.000000

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

```
select event_id,event_name from events_list where (total_seats-available_seats)>(total_seats*0.5);
```

event_id	event_name
e1	Batminton_cup_2026
e2	Evaru
e3	Harmony Fest of concert
e4	Clap
e6	X_Men
e8	Rhythm Rave concert
e9	Zenith concert,2025
NULL	NULL

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

```
select e.event_id,e.event_name ,(select sum(num_tickets) from Booking b
where e.event_id=b.event_id)as tickets_sold from Events_list e;
```

event_id	event_name	tickets_sold
e0	world_cup_match	4
e1	Batminton_cup_2026	5
e2	Evaru	4
e3	Harmony Fest of concert	3
e4	Clap	9
e5	Cricket_cup_2026	2
e6	X_Men	1
e7	Zorb_Football	3
e8	Rhythm Rave concert	3
e9	Zenith concert,2025	2

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* from Booking b where b.booking_id=c.booking_id);

customer_id	customer_name
NULL	NULL

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

select event_id, event_name from events_list where event_id not in (select distinct

event_id from Booking);

event_id	event_name
NULL	NULL

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

```
SELECT
   event_type,
   SUM(tickets_sold) AS total_tickets_sold
FROM (select event_type,sum(total_seats-available_seats) AS tickets_sold
   FROM Events_lis GROUP BY event_type
) AS subquery GROUP BY event_type;
```

event_type	total_tickets_sold
Sports	35513
Movies	1425
Music	31250

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 events_list where ticket_price> (select

avg(ticket_price) from events_list);

event_id	event_name	ticket_price
e0	world_cup_match	2500.00
e3	Harmony Fest of concert	1100.00
e5	Cricket_cup_2026	1440.00
e7	Zorb_Football	2450.00
e8	Rhythm Rave concert	1500.00
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 sum(total_cost) from Booking b where b.customer_id=c.customer_id) as total_revenue from Customer c;

customer_id	customer_name	total_revenue
110	Rhythm	12500.00
111	Raj	2500.00
112	Arun	540.00
113	Ronish	3000.00
222	Ram	1000.00
333	Rajesh	3300.00
444	Rohan	1080.00
555	keerthi	2880.00
666	suba	230.00
777	sri	7350.00
888	Vaishnavi	1500.00
999	sam	1000.00

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

select customer_id,customer_name from customer where booking_id in

(select booking_id from booking where event_id in

(select event_id from events_list where venue_id ='B'));

	customer_id	customer_name
▶	222	Ram
*	NULL	NULL

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

```
select event_type, sum(total_tickets_sold) AS total_tickets_sold
 from(
select event_type, sum(total_seats-available_seats) AS total_tickets_sold
from Events_list
GROUP BY event_id
) as total_tickets
GROUP BY event_type;
```

	event_type	total_tickets_sold
▶	Sports	35513
	Movies	1425
	Music	31250

11.Find Users Who Have Booked Tickets for Events in each month.

```
select c.customer_id ,c.customer_name,(select b.booking_date from booking b where c.booking_id=b.booking_id) as booking_date
from customer c order by booking_date;
```

	customer_id	customer_name	booking_date
▶	222	Ram	2024-04-03
	666	suba	2024-04-28
	113	Ronish	2024-05-03
	888	Vaishnavi	2024-05-03
	110	Rhythm	2024-05-16
	112	Arun	2024-05-27
	444	Rohan	2024-05-27
	777	sri	2024-05-29
	999	sam	2025-04-24
	333	Rajesh	2025-05-06
	111	Raj	2026-04-03
	555	keerthi	2026-06-12

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

```
select v.venue_name,(select avg(ticket_price) from events_list e where e.venue_id =
 v.venue_id) as avg_ticket_price from Venu v;
```

	venue_name	avg_ticket_price
▶	Allan Batmintion club	500.000000
	Kasi theatre	250.000000
	EVP Palace	1100.000000
	Jazz cinemas	180.000000
	Madrass cricket club	1440.000000
	GK cinemas	230.000000
	Ashok sports club	2450.000000
	Nani mahal	1500.000000
	VMM sports	500.000000
	Chepauk Stadium	2500.000000