ASSIGNMENT -1 SQL & OOPS

TICKET BOOKING SYSTEM

CREATING A DATABASE

Task 1: Database Design

```
CREATE DATABASE TicketBookingSystem;
USE TicketBookingSystem;

CREATE TABLE Venu (
    venue_id INT AUTO_INCREMENT PRIMARY KEY,
    venue_name VARCHAR(100) NOT NULL,
    address VARCHAR(255) NOT NULL
);

CREATE TABLE Customer (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE,
    phone_number VARCHAR(15)
);

CREATE TABLE Event (
    event_id INT AUTO_INCREMENT PRIMARY KEY,
    event_name VARCHAR(100) NOT NULL,
    event_date DATE NOT NULL,
    event_time TIME NOT NULL,
    venue_id INT,
    total_seats INT NOT NULL,
    available_seats INT NOT NULL,
```

```
    ticket_price DECIMAL(10, 2) NOT NULL,
    event_type ENUM('Movie', 'Sports', 'Concert') NOT NULL,
    FOREIGN KEY (venue_id) REFERENCES Venu(venue_id) ON
DELETE CASCADE
);

CREATE TABLE Booking (
    booking_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    event_id INT,
    num_tickets INT NOT NULL,
    total_cost DECIMAL(10, 2) NOT NULL,
    booking_date DATE NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
ON DELETE CASCADE,
    FOREIGN KEY (event_id) REFERENCES Event(event_id) ON
DELETE CASCADE
);
```

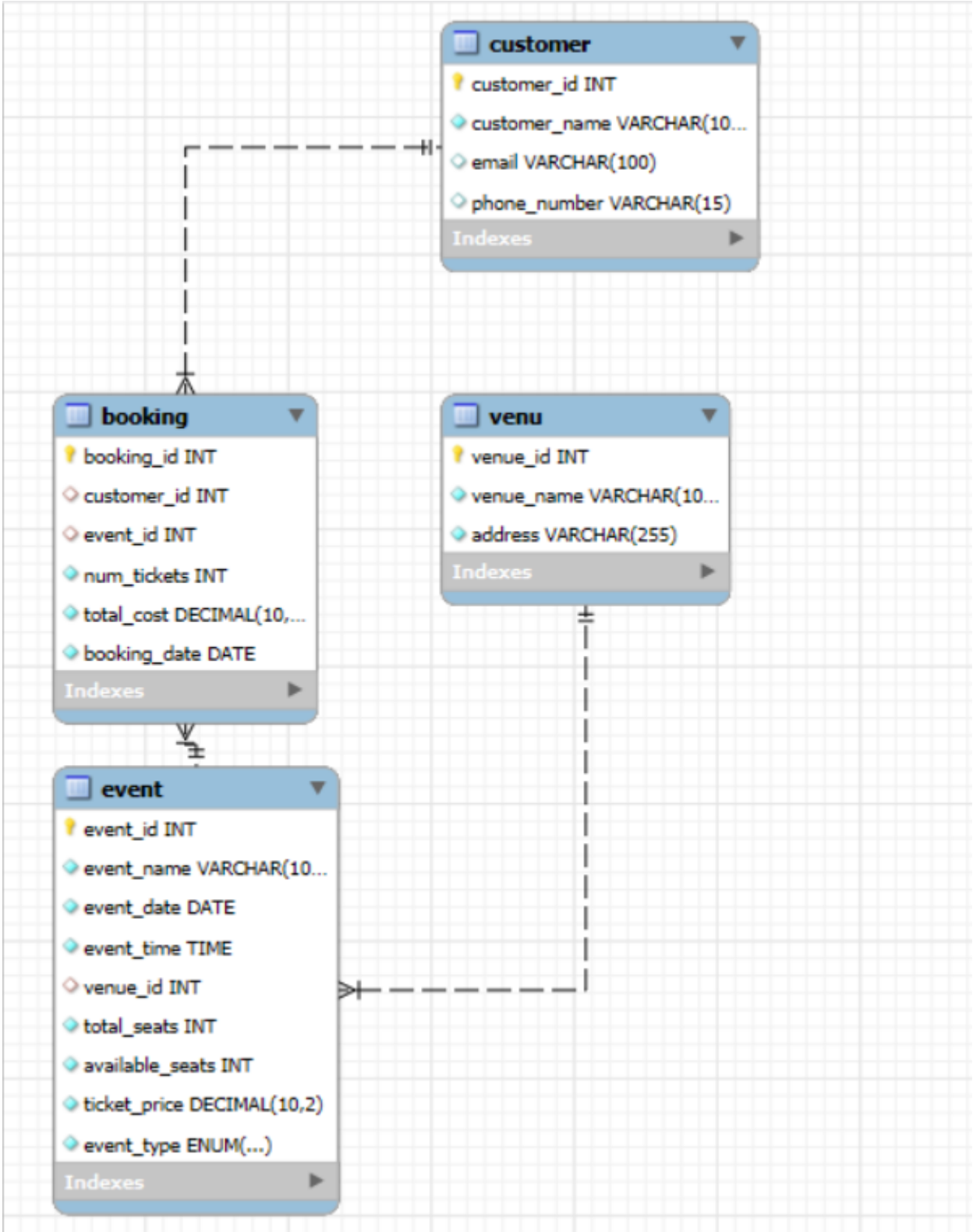
OUTPUT:

Output

Action Output

#	Time	Action	Message	Duration / Fetch
63	12:22:13	CREATE DATABASE TicketBookingSystem	1 row(s) affected	0.016 sec
64	12:22:16	USE TicketBookingSystem	0 row(s) affected	0.000 sec
65	12:22:20	CREATE TABLE Venu (venue_id INT AUTO_INCREMENT PRI...	0 row(s) affected	0.016 sec
66	12:22:23	CREATE TABLE Customer (customer_id INT AUTO_INCREMEN...	0 row(s) affected	0.015 sec
67	12:22:30	CREATE TABLE Event (event_id INT AUTO_INCREMENT PRI...	0 row(s) affected	0.016 sec
68	12:23:16	CREATE TABLE Booking (booking_id INT AUTO_INCREMENT ...	0 row(s) affected	0.047 sec

ER DIAGRAM:


customer
customer_id INT
customer_name VARCHAR(10...
email VARCHAR(100)
phone_number VARCHAR(15)
Indexes
booking
booking_id INT
customer_id INT
event_id INT
num_tickets INT
total_cost DECIMAL(10,...
booking_date DATE
Indexes
venu
venue_id INT
venue_name VARCHAR(10...
address VARCHAR(255)
Indexes
event
event_id INT
event_name VARCHAR(10...
event_date DATE
event_time TIME
venue_id INT
total_seats INT
available_seats INT
ticket_price DECIMAL(10,2)
event_type ENUM(...)
Indexes

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

-- 1) INSERTING THE VALUES

```
-- Insert into Venu
INSERT INTO Venu (venue_name, address) VALUES
('City Arena', 'Downtown Street 1'),
('Sunrise Hall', 'Green Park Avenue'),
('Sky Theatre', 'Skyline Road 23'),
('Ocean Dome', 'Beachside Blvd'),
('Grand Hall', 'City Center'),
('Riverstage', 'Riverbank Road'),
('Moonlight Pavilion', 'Lunar Lane'),
('Phoenix Grounds', 'Desert Circle'),
('Metro Auditorium', 'Metro Line'),
('Mountain View Hall', 'Hilltop Road');

-- Insert into Customer
INSERT INTO Customer (customer_name, email, phone_number)
VALUES
('Amit Roy', 'amit@gmail.com', '9876540001'),
('Sneha Patil', 'sneha@gmail.com', '9876540002'),
('John Deo', 'john@gmail.com', '9876543000'),
('Anjali Mehra', 'anjali@gmail.com', '9876540003'),
('Arun Kumar', 'arun@gmail.com', '9876540004'),
('Priya Singh', 'priya@gmail.com', '9876545000'),
('David Smith', 'david@gmail.com', '9876540005'),
('Meena Kapoor', 'meena@gmail.com', '9876540006'),
```

```
('Rahul Jain', 'rahul@gmail.com', '9876540007'),
('Sara Khan', 'sara@gmail.com', '9876548000');

-- Insert into Event
INSERT INTO Event (event_name, event_date, event_time, venue_id, total_seats, available_seats, ticket_price, event_type) VALUES
('Cricket Cup', '2025-08-10', '18:00:00', 1, 20000, 15000, 1500.00, 'Sports'),
('Movie Premiere', '2025-06-20', '20:00:00', 2, 500, 250, 250.00, 'Movie'),
('Rock Concert', '2025-07-15', '19:30:00', 3, 8000, 2000, 1800.00, 'Concert'),
('Drama Play', '2025-07-01', '17:00:00', 4, 400, 150, 700.00, 'Movie'),
('Football Cup', '2025-09-12', '16:00:00', 5, 30000, 22000, 2000.00, 'Sports'),
('Jazz Concert', '2025-07-20', '20:00:00', 6, 1000, 500, 1300.00, 'Concert'),
('Dance Show', '2025-08-05', '18:30:00', 7, 1500, 1000, 1200.00, 'Movie'),
('HipHop Cup', '2025-07-22', '19:00:00', 8, 25000, 24000, 1100.00, 'Sports'),
('Indie Concert', '2025-08-25', '21:00:00', 9, 3000, 1500, 2100.00, 'Concert'),
('Opera Night', '2025-10-01', '20:30:00', 10, 5000, 1000, 3000.00, 'Concert');

-- Insert into Booking
INSERT INTO Booking (customer_id, event_id, num_tickets, total_cost, booking_date) VALUES
(1, 1, 2, 3000.00, '2025-06-01'),
(2, 2, 1, 250.00, '2025-06-05'),
(3, 3, 3, 5400.00, '2025-06-10'),
```

(4, 4, 2, 1400.00, '2025-06-12'),

(5, 5, 5, 10000.00, '2025-06-15'),

(6, 6, 1, 1300.00, '2025-06-18'),

(7, 7, 2, 2400.00, '2025-06-20'),

(8, 8, 6, 6600.00, '2025-06-22'),

(9, 9, 4, 8400.00, '2025-06-25'),

(10, 10, 2, 6000.00, '2025-06-27');

OUTPUT:

Output

Action Output

#	Time	Action	Message	Duration / Fetch
69	12:46:30	INSERT INTO Venu (venue_name, address) VALUES ('City Arena', '...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
70	12:46:35	INSERT INTO Customer (customer_name, email, phone_number) VA...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
71	12:46:39	INSERT INTO Event (event_name, event_date, event_time, venue...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
72	12:46:47	INSERT INTO Booking (customer_id, event_id, num_tickets, total_c...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec

-- 2) LIST ALL EVENTS

SELECT * FROM Event;

OUTPUT:

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

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
2	Movie Premiere	2025-06-20	20:00:00	2	500	250	250.00	Movie
3	Rock Concert	2025-07-15	19:30:00	3	8000	2000	1800.00	Concert
4	Drama Play	2025-07-01	17:00:00	4	400	150	700.00	Movie
5	Football Cup	2025-09-12	16:00:00	5	30000	22000	2000.00	Sports
6	Jazz Concert	2025-07-20	20:00:00	6	1000	500	1300.00	Concert
7	Dance Show	2025-08-05	18:30:00	7	1500	1000	1200.00	Movie
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	Sports
9	Indie Concert	2025-08-25	21:00:00	9	3000	1500	2100.00	Concert
10	Opera Night	2025-10-01	20:30:00	10	5000	1000	3000.00	Concert
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types

-- 3) EVENTS WITH AVAILABLE TICKETS

SELECT * FROM Event WHERE available_seats > 0;

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
2	Movie Premiere	2025-06-20	20:00:00	2	500	250	250.00	Movie
3	Rock Concert	2025-07-15	19:30:00	3	8000	2000	1800.00	Concert
4	Drama Play	2025-07-01	17:00:00	4	400	150	700.00	Movie
5	Football Cup	2025-09-12	16:00:00	5	30000	22000	2000.00	Sports
6	Jazz Concert	2025-07-20	20:00:00	6	1000	500	1300.00	Concert
7	Dance Show	2025-08-05	18:30:00	7	1500	1000	1200.00	Movie
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	Sports
9	Indie Concert	2025-08-25	21:00:00	9	3000	1500	2100.00	Concert
10	Opera Night	2025-10-01	20:30:00	10	5000	1000	3000.00	Concert

-- 4) EVENTS NAME PARTIAL MATCH WITH 'CUP'

SELECT * FROM Event WHERE event_name LIKE '%cup%';

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
5	Football Cup	2025-09-12	16:00:00	5	30000	22000	2000.00	Sports
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	2000.00

-- 5) EVENTS WITH TICKET PRICE BETWEEN 1000 & 2500

SELECT * FROM Event WHERE ticket_price BETWEEN 1000 AND 2500;

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
3	Rock Concert	2025-07-15	19:30:00	3	8000	2000	1800.00	Concert
5	Football Cup	2025-09-12	16:00:00	5	30000	22000	2000.00	Sports
6	Jazz Concert	2025-07-20	20:00:00	6	1000	500	1300.00	Concert
7	Dance Show	2025-08-05	18:30:00	7	1500	1000	1200.00	Movie
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	Sports
9	Indie Concert	2025-08-25	21:00:00	9	3000	1500	2100.00	Concert

-- 6) EVENTS WITH DATES IN A SPECIFIC RANGE

SELECT * FROM Event

WHERE event_date BETWEEN '2025-07-01' AND '2025-08-31';

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
3	Rock Concert	2025-07-15	19:30:00	3	8000	2000	1800.00	Concert
4	Drama Play	2025-07-01	17:00:00	4	400	150	700.00	Movie
6	Jazz Concert	2025-07-20	20:00:00	6	1000	500	1300.00	Concert
7	Dance Show	2025-08-05	18:30:00	7	1500	1000	1200.00	Movie
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	Sports
9	Indie Concert	2025-08-25	21:00:00	9	3000	1500	2100.00	Concert
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- 7) EVENTS WITH AVAILABLE TICKETS AND "CONCERT" IN NAME

SELECT * FROM Event

WHERE available_seats > 0 AND event_name LIKE '%Concert%';

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
3	Rock Concert	2025-07-15	19:30:00	3	8000	2000	1800.00	Concert
6	Jazz Concert	2025-07-20	20:00:00	6	1000	500	1300.00	Concert
9	Indie Concert	2025-08-25	21:00:00	9	3000	1500	2100.00	Concert
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- 8) RETRIEVE USERS IN BATCHES OF 5 STARTING FROM 6TH USER

SELECT * FROM Customer LIMIT 5 OFFSET 5;

OUTPUT:

customer_id	customer_name	email	phone_number
6	Priya Singh	priya@gmail.com	9876545000
7	David Smith	david@gmail.com	9876540005
8	Meena Kapoor	meena@gmail.com	9876540006
9	Rahul Jain	rahul@gmail.com	9876540007
10	Sara Khan	sara@gmail.com	9876548000
NULL	NULL	NULL	NULL

-- 9) BOOKINGS WHERE NUMBER OF TICKETS > 4

SELECT * FROM Booking WHERE num_tickets > 4;

OUTPUT:

booking_id	customer_id	event_id	num_tickets	total_cost	booking_date
5	5	5	5	10000.00	2025-06-15
8	8	8	6	6600.00	2025-06-22
NULL	NULL	NULL	NULL	NULL	NULL

-- 10) CUSTOMERS WHOSE PHONE NUMBER ENDS WITH '000'

SELECT * FROM Customer WHERE phone_number LIKE '%000';

OUTPUT:

customer_id	customer_name	email	phone_number
3	John Deo	john@gmail.com	9876543000
6	Priya Singh	priya@gmail.com	9876545000
10	Sara Khan	sara@gmail.com	9876548000
NULL	NULL	NULL	NULL

-- 11) EVENTS ORDERED BY SEAT CAPACITY > 15000

SELECT * FROM Event

WHERE total_seats > 15000 ORDER BY total_seats DESC;

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
5	Football Cup	2025-09-12	16:00:00	5	30000	22000	2000.00	Sports
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	Sports
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- 12) EVENTS NAME NOT STARTING WITH 'x', 'y' OR 'z'

SELECT * FROM Event

WHERE event_name NOT LIKE 'x%'

AND event_name NOT LIKE 'y%'

AND event_name NOT LIKE 'z%';

OUTPUT:

event_id	event_name	event_date	event_time	venue_id	total_seats	available_seats	ticket_price	event_type
1	Cricket Cup	2025-08-10	18:00:00	1	20000	15000	1500.00	Sports
2	Movie Premiere	2025-06-20	20:00:00	2	500	250	250.00	Movie
3	Rock Concert	2025-07-15	19:30:00	3	8000	2000	1800.00	Concert
4	Drama Play	2025-07-01	17:00:00	4	400	150	700.00	Movie
5	Football Cup	2025-09-12	16:00:00	5	30000	22000	2000.00	Sports
6	Jazz Concert	2025-07-20	20:00:00	6	1000	500	1300.00	Concert
7	Dance Show	2025-08-05	18:30:00	7	1500	1000	1200.00	Movie
8	HipHop Cup	2025-07-22	19:00:00	8	25000	24000	1100.00	Sports
9	Indie Concert	2025-08-25	21:00:00	9	3000	1500	2100.00	Concert
10	Opera Night	2025-10-01	20:30:00	10	5000	1000	3000.00	Concert
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Task 3: Aggregate functions, Having, Order By, Group By and Joins

-- 1) LIST EVENTS AND THEIR AVERAGE TICKET PRICE

SELECT event_name, AVG (ticket_price) AS average_price

FROM Event GROUP BY event_name;

OUTPUT:

event_name	average_price
Cricket Cup	1500.000000
Movie Premiere	250.000000
Rock Concert	1800.000000
Drama Play	700.000000
Football Cup	2000.000000
Jazz Concert	1300.000000
Dance Show	1200.000000
HipHop Cup	1100.000000
Indie Concert	2100.000000
Opera Night	3000.000000

-- 2) TOTAL REVENUE GENERATED BY EVENTS

```
SELECT e.event_name, SUM(b.total_cost) AS total_revenue
FROM Booking b
JOIN Event e ON b.event_id = e.event_id GROUP BY e.event_name;
```

OUTPUT:

Result Grid | Filter Rows:

event_name	total_revenue
Cricket Cup	3000.00
Movie Premiere	250.00
Rock Concert	5400.00
Drama Play	1400.00
Football Cup	10000.00
Jazz Concert	1300.00
Dance Show	2400.00
HipHop Cup	6600.00
Indie Concert	8400.00
Opera Night	6000.00

-- 3) EVENT WITH HIGHEST TICKET SALES

```
SELECT e.event_name, SUM (b.num_tickets) AS total_tickets
FROM Booking b
JOIN Event e ON b.event_id = e.event_id
GROUP BY e.event_name ORDER BY total_tickets DESC
LIMIT 1;
```

OUTPUT:

Result Grid | Filter Rows:

event_name	total_tickets
HipHop Cup	6

-- 4) TOTAL NUMBER OF TICKETS SOLD FOR EACH EVENT

```
SELECT e.event_name, SUM (b.num_tickets) AS tickets_sold
FROM Booking b
JOIN Event e ON b.event_id = e.event_id GROUP BY e.event_name;
```

OUTPUT:

Result Grid | Filter Rows:

event_name	tickets_sold
Cricket Cup	2
Movie Premiere	1
Rock Concert	3
Drama Play	2
Football Cup	5
Jazz Concert	1
Dance Show	2
HipHop Cup	6
Indie Concert	4
Opera Night	2

-- 5) EVENTS WITH NO TICKET SALES

```
SELECT e.event_name
FROM Event e
LEFT JOIN Booking b ON e.event_id = b.event_id
WHERE b.booking_id IS NULL;
```

OUTPUT:

Result Grid | Filter

event_name

-- 6) USER WHO HAS BOOKED THE MOST TICKETS

```
SELECT c.customer_name, SUM(b.num_tickets) AS total_tickets
FROM Booking b
```

```
JOIN Customer c ON b.customer_id = c.customer_id
GROUP BY c.customer_name ORDER BY total_tickets DESC
LIMIT 1;
```

OUTPUT:

Result Grid | Filter Rows:

customer_name	total_tickets
Meena Kapoor	6

-- 7) EVENTS AND TICKETS SOLD PER MONTH

```
SELECT
    MONTH(booking_date) AS month,
    e.event_name,
    SUM(b.num_tickets) AS tickets_sold
FROM Booking b
JOIN Event e ON b.event_id = e.event_id
GROUP BY MONTH(booking_date), e.event_name ORDER BY month;
```

OUTPUT:

Result Grid | Filter Rows:

month	event_name	tickets_sold
6	Cricket Cup	2
6	Dance Show	2
6	Drama Play	2
6	Football Cup	5
6	HipHop Cup	6
6	Indie Concert	4
6	Jazz Concert	1
6	Movie Premiere	1
6	Opera Night	2
6	Rock Concert	3

-- 8) AVERAGE TICKET PRICE FOR EVENTS IN EACH VENUE

```
SELECT v.venue_name, AVG(e.ticket_price) AS avg_price
FROM Event e
JOIN Venu v ON e.venue_id = v.venue_id GROUP BY v.venue_name;
```

OUTPUT:

Result Grid | Filter Rows:

venue_name	avg_price
City Arena	1500.000000
Sunrise Hall	250.000000
Sky Theatre	1800.000000
Ocean Dome	700.000000
Grand Hall	2000.000000
Riverstage	1300.000000
Moonlight Pavilion	1200.000000
Phoenix Grounds	1100.000000
Metro Auditorium	2100.000000
Mountain View Hall	3000.000000

-- 9) TOTAL TICKETS SOLD BY EVENT TYPE

```
SELECT e.event_type, SUM(b.num_tickets) AS total_tickets
FROM Booking b
JOIN Event e ON b.event_id = e.event_id
GROUP BY e.event_type;
```

OUTPUT:

Result Grid | Filter Rows:

event_type	total_tickets
Sports	13
Movie	5
Concert	10

-- 10) TOTAL REVENUE BY YEAR

SELECT YEAR(booking_date) AS year, SUM(total_cost) AS total_revenue

FROM Booking GROUP BY YEAR(booking_date);

OUTPUT:

Result Grid | Filter Rows:

year	total_revenue
2025	44750.00

-- 11) TOTAL REVENUE BY USER

SELECT c.customer_name, SUM(b.total_cost) AS total_spent

FROM Booking b

JOIN Customer c ON b.customer_id = c.customer_id

GROUP BY c.customer_name;

OUTPUT:

Result Grid | Filter Rows:

customer_name	total_spent
Amit Roy	3000.00
Sneha Patil	250.00
John Deo	5400.00
Anjali Mehra	1400.00
Arun Kumar	10000.00
Priya Singh	1300.00
David Smith	2400.00
Meena Kapoor	6600.00
Rahul Jain	8400.00
Sara Khan	6000.00

-- 12) USERS WHO BOOKED FOR MULTIPLE EVENTS

SELECT customer_id

FROM Booking

```
GROUP BY customer_id
HAVING COUNT(DISTINCT event_id) > 1;
```

OUTPUT:

Result Grid | Filter Rows:

customer_id

-- 13) AVERAGE TICKET PRICE BY CATEGORY AND VENUE

```
SELECT e.event_type, v.venue_name, AVG(e.ticket_price) AS avg_price
FROM Event e JOIN Venu v ON e.venue_id = v.venue_id
GROUP BY e.event_type, v.venue_name;
```

OUTPUT:

Result Grid | Filter Rows:

event_type	venue_name	avg_price
Sports	City Arena	1500.000000
Movie	Sunrise Hall	250.000000
Concert	Sky Theatre	1800.000000
Movie	Ocean Dome	700.000000
Sports	Grand Hall	2000.000000
Concert	Riverstage	1300.000000
Movie	Moonlight Pavilion	1200.000000
Sports	Phoenix Grounds	1100.000000
Concert	Metro Auditorium	2100.000000
Concert	Mountain View Hall	3000.000000

-- 14) USERS AND TICKETS PURCHASED IN LAST 30 DAYS

```
SELECT c.customer_name, SUM(b.num_tickets) AS tickets_last_30_days
FROM Booking b
JOIN Customer c ON b.customer_id = c.customer_id
WHERE booking_date >= CURDATE() - INTERVAL 30 DAY
GROUP BY c.customer_name;
```

OUTPUT:

Result Grid | Filter Rows:

customer_name	tickets_last_30_days
Amit Roy	2
Sn Sneha Patil	1
John Deo	3
Anjali Mehra	2
Arun Kumar	5
Priya Singh	1
David Smith	2
Meena Kapoor	6
Rahul Jain	4
Sara Khan	2

Task 4: Subquery and its types

-- 1) AVERAGE TICKET PRICE FOR EVENTS IN EACH VENUE (Subquery in SELECT)

SELECT v.venue_name,

(SELECT AVG(ticket_price) FROM Event e WHERE e.venue_id = v.venue_id) AS avg_ticket_price

FROM Venu v;

OUTPUT:

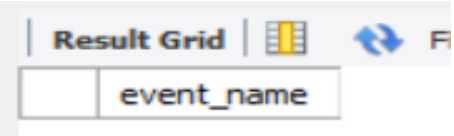
Result Grid | Filter Rows:

venue_name	avg_ticket_price
City Arena	1500.000000
Sunrise Hall	250.000000
Sky Theatre	1800.000000
Ocean Dome	700.000000
Grand Hall	2000.000000
Riverstage	1300.000000
Moonlight Pavilion	1200.000000
Phoenix Grounds	1100.000000
Metro Auditorium	2100.000000
Mountain View Hall	3000.000000

-- 2) EVENTS WITH MORE THAN 50% OF TICKETS SOLD (Subquery in WHERE)

```
SELECT event_name
FROM Event
WHERE event_id IN (
    SELECT e.event_id
    FROM Event e
    JOIN Booking b ON e.event_id = b.event_id
    GROUP BY e.event_id, e.total_seats
    HAVING SUM(b.num_tickets) > (e.total_seats * 0.5)
);
```

OUTPUT:



-- 3) TOTAL NUMBER OF TICKETS SOLD FOR EACH EVENT (Subquery in FROM)

```
SELECT e.event_name, t.total_tickets
FROM Event e
JOIN (
    SELECT event_id, SUM(num_tickets) AS total_tickets
    FROM Booking
    GROUP BY event_id
) t ON e.event_id = t.event_id;
```

OUTPUT:

Result Grid | Filter Rows:

event_name	total_tickets
Cricket Cup	2
Movie Premiere	1
Rock Concert	3
Drama Play	2
Football Cup	5
Jazz Concert	1
Dance Show	2
HipHop Cup	6
Indie Concert	4
Opera Night	2

-- 4) USERS WHO HAVE NOT BOOKED TICKETS (Subquery with NOT EXISTS)

```
SELECT customer_name
FROM Customer c
WHERE NOT EXISTS (
    SELECT 1
    FROM Booking b
    WHERE b.customer_id = c.customer_id
);
```

OUTPUT:

Result Grid | Filt

customer_name

-- 5) EVENTS WITH NO TICKET SALES (Subquery with NOT IN)

```
SELECT event_name
FROM Event
```

```
WHERE event_id NOT IN (
    SELECT event_id FROM Booking
);
```

OUTPUT:

Result Grid | Fil

event_name

-- 6) TICKETS SOLD FOR EACH EVENT TYPE (Subquery in FROM)

```
SELECT event_type, SUM(total_tickets) AS tickets_sold
FROM (
    SELECT e.event_type, b.num_tickets AS total_tickets
    FROM Booking b
    JOIN Event e ON b.event_id = e.event_id
) AS sub
GROUP BY event_type;
```

OUTPUT:

Result Grid | Filter Row

event_type	tickets_sold
Sports	13
Movie	5
Concert	10

-- 7) EVENTS WITH TICKET PRICE HIGHER THAN AVERAGE (Subquery in WHERE)

```
SELECT event_name, ticket_price
FROM Event
WHERE ticket_price > (
```

```
    SELECT AVG(ticket_price) FROM Event
);
```

OUTPUT:

Result Grid | Filter Rows:

event_name	ticket_price
Cricket Cup	1500.00
Rock Concert	1800.00
Football Cup	2000.00
Indie Concert	2100.00
Opera Night	3000.00

-- 8) REVENUE GENERATED BY EVENTS FOR EACH USER (Correlated subquery)

```
SELECT c.customer_name,
       (SELECT SUM(b.total_cost)
        FROM Booking b
        WHERE b.customer_id = c.customer_id) AS total_revenue
FROM Customer c;
```

OUTPUT:

Result Grid | Filter Rows:

customer_name	total_revenue
Amit Roy	3000.00
Sneha Patil	250.00
John Deo	5400.00
Anjali Mehra	1400.00
Arun Kumar	10000.00
Priya Singh	1300.00
David Smith	2400.00
Meena Kapoor	6600.00
Rahul Jain	8400.00
Sara Khan	6000.00

-- 9) USERS WHO BOOKED TICKETS FOR EVENTS IN A GIVEN VENUE (Subquery in WHERE)

```
SELECT customer_name
FROM Customer
WHERE customer_id IN (
    SELECT b.customer_id
    FROM Booking b
    JOIN Event e ON b.event_id = e.event_id
    WHERE e.venue_id = 1  -- Replace 1 with the venue_id you want
);
```

OUTPUT:

Result Grid

customer_name
Amit Roy

-- 10) TICKETS SOLD PER EVENT CATEGORY (Subquery with GROUP BY)

```
SELECT event_type, SUM(num_tickets) AS tickets_sold
FROM (
    SELECT e.event_type, b.num_tickets
    FROM Booking b
    JOIN Event e ON b.event_id = e.event_id
) AS sub
GROUP BY event_type;
```

OUTPUT:

Result Grid | Filter Rows:

event_type	tickets_sold
Sports	13
Movie	5
Concert	10

-- 11) USERS WHO BOOKED TICKETS EACH MONTH (Subquery with DATE_FORMAT)

```
SELECT DISTINCT c.customer_name
FROM Customer c
WHERE c.customer_id IN (
    SELECT b.customer_id
    FROM Booking b
    WHERE DATE_FORMAT(b.booking_date, '%Y-%m') IS NOT NULL
);
```

OUTPUT:

Result Grid

customer_name
Amit Roy
Sneha Patil
John Deo
Anjali Mehra
Arun Kumar
Priya Singh
David Smith
Meena Kapoor
Rahul Jain
Sara Khan

-- 12) AVERAGE TICKET PRICE PER VENUE (Duplicate of query 1 for revision)

```
SELECT v.venue_name,
```

```
(SELECT AVG(ticket_price)
 FROM Event e
 WHERE e.venue_id = v.venue_id) AS avg_ticket_price
FROM Venu v;
```

OUTPUT:

Result Grid | Filter Rows:

venue_name	avg_ticket_price
City Arena	1500.000000
Sunrise Hall	250.000000
Sky Theatre	1800.000000
Ocean Dome	700.000000
Grand Hall	2000.000000
Riverstage	1300.000000
Moonlight Pavilion	1200.000000
Phoenix Grounds	1100.000000
Metro Auditorium	2100.000000
Mountain View Hall	3000.000000