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

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

//query//

mysql> INSERT INTO Venue (venue\_id, venue\_name, address) VALUES

-> (1, 'Shree Ram Theatre', 'Bhopal'),

-> (2, 'Bhopal Haat', 'Bhopal'),

-> (3, 'Vidisha Heritage', 'Vidisha'),

-> (4, 'Kamal Palace', 'Bhopal'),

-> (5, 'Shankar Temple', 'Vidisha');

Query OK, 5 rows affected (0.02 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Event (event\_id, event\_name, event\_date, event\_time, venue\_id, total\_seats, available\_seats, ticket\_price, event\_type) VALUES

-> (1, 'Movie Night', '2024-10-10', '19:00:00', 1, 200, 150, 500.00, 'Movie'),

-> (2, 'Cricket Match', '2024-10-15', '16:00:00', 2, 5000, 4000, 1500.00, 'Sports'),

-> (3, 'Music Evening', '2024-11-01', '20:00:00', 3, 100, 80, 300.00, 'Concert'),

-> (4, 'Bhopal Carnival', '2024-10-20', '18:00:00', 4, 1000, 800, 700.00, 'Concert');

Query OK, 4 rows affected (0.02 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Booking (booking\_id, customer\_id, event\_id, num\_tickets, total\_cost, booking\_date) VALUES

-> (1, 1, 1, 2, 1000.00, '2024-10-01'),

-> (2, 2, 2, 5, 7500.00, '2024-10-02'),

-> (3, 3, 3, 1, 300.00, '2024-10-03'),

-> (4, 4, 4, 3, 2100.00, '2024-10-04');

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Customer (customer\_id, customer\_name, email, phone\_number, booking\_id) VALUES

-> (1, 'Aditi Mishra', 'aditi.mishra@example.com', '7896543210', 1),

-> (2, 'Rohan Verma', 'rohan.verma@example.com', '7896543211', 2),

-> (3, 'Neha Pandey', 'neha.pandey@example.com', '7896543212', 3),

-> (4, 'Vikrant Shukla', 'vikrant.shukla@example.com', '7896543213', 4);

2: List all Events

SELECT event\_id, event\_name, event\_date, event\_time, venue\_id, total\_seats, available\_seats, ticket\_price, event\_type

FROM Event;

Output:

| event\_id | event\_name | event\_date | event\_time | venue\_id | total\_seats | available\_seats | ticket\_price | event\_type |

|----------|-----------------|------------|------------|----------|-------------|-----------------|--------------|------------|

| 1 | Movie Night | 2024-10-10 | 19:00:00 | 1 | 200 | 150 | 500.00 | Movie |

| 2 | Cricket Match | 2024-10-15 | 16:00:00 | 2 | 5000 | 4000 | 1500.00 | Sports |

| 3 | Music Evening | 2024-11-01 | 20:00:00 | 3 | 100 | 80 | 300.00 | Concert |

| 4 | Bhopal Carnival | 2024-10-20 | 18:00:00 | 4 | 1000 | 800 | 700.00 | Concert |

3: Select events with available tickets

SELECT event\_name, total\_seats, available\_seats

FROM Event

WHERE available\_seats > 0;

Output:

| event\_name | total\_seats | available\_seats |

|-----------------|-------------|-----------------|

| Movie Night | 200 | 150 |

| Cricket Match | 5000 | 4000 |

| Music Evening | 100 | 80 |

| Bhopal Carnival | 1000 | 800 |

4: Select events whose name partially matches 'cup'

SELECT event\_name, event\_type, event\_date

FROM Event

WHERE event\_name LIKE '%cup%';

Output:

| event\_name | event\_type | event\_date |

|------------|------------|------------|

| \*No matching records\* (Assuming none of your event names include 'cup') |

5: Select events where the ticket price falls between 1000 and 2500

SELECT event\_name, ticket\_price

FROM Event

WHERE ticket\_price BETWEEN 1000 AND 2500;

Output:

| event\_name | ticket\_price |

|---------------|--------------|

| Cricket Match | 1500.00 |

| Bhopal Carnival | 700.00 |

6: Retrieve events that occur within a specific date range

SELECT event\_name, event\_date

FROM Event

WHERE event\_date BETWEEN '2024-10-01' AND '2024-11-01';

Output:

| event\_name | event\_date |

|-----------------|------------|

| Movie Night | 2024-10-10 |

| Cricket Match | 2024-10-15 |

| Bhopal Carnival | 2024-10-20 |

| Music Evening | 2024-11-01 |

7: Retrieve events with available tickets that are 'Concert' type

SELECT event\_name, available\_seats, event\_type

FROM Event

WHERE available\_seats > 0

AND event\_type = 'Concert';

Output:

| event\_name | available\_seats | event\_type |

|-----------------|-----------------|------------|

| Music Evening | 80 | Concert |

| Bhopal Carnival | 800 | Concert |

8: Retrieve customers in batches of 5, starting from the 6th record

SELECT customer\_id, customer\_name, email, phone\_number

FROM Customer

LIMIT 5 OFFSET 5;

Output:

| customer\_id | customer\_name | email | phone\_number |

|-------------|---------------|-------|--------------|

| \*No matching records\* (Assuming there are fewer than 6 customers in the `Customer` table) |

9: Retrieve booking details where the number of tickets booked exceeds 4

SELECT booking\_id, event\_id, customer\_id, num\_tickets, total\_cost

FROM Booking

WHERE num\_tickets > 4;

Output:

| booking\_id | event\_id | customer\_id | num\_tickets | total\_cost |

|------------ |---------- | ------------- |------------ -|------------ |

| 2 | 2 | 2 | 5 | 7500.00 |

10: Retrieve customer information whose phone number ends with '000'

SELECT customer\_id, customer\_name, email, phone\_number

FROM Customer

WHERE phone\_number LIKE '%000';

Output:

| customer\_id | customer\_name | email | phone\_number |

|------------- |--------------- |------- |-------------- |

| \*No matching records\* (Assuming none of the customers have phone numbers ending with '000') |

11: Retrieve the events in order whose seat capacity is more than 15000

SELECT event\_name, total\_seats

FROM Event

WHERE total\_seats > 15000

ORDER BY total\_seats;

Output:

| event\_name | total\_seats |

|------------ |------------- |

| \*No matching records\* (Assuming no events have a seat capacity greater than 15000) |

12: Select event names that do not start with 'x', 'y', or 'z'

SELECT event\_name

FROM Event

WHERE event\_name NOT LIKE 'x%'

AND event\_name NOT LIKE 'y%'

AND event\_name NOT LIKE 'z%';

Output:

| event\_name |

|----------------- |

| Movie Night |

| Cricket Match |

| Music Evening |

| Bhopal Carnival |