ASSIGNMENT TICKET BOOKING SYSTEM

create database ticket\_booking\_system;

use ticket\_booking\_system;

CREATE TABLE venue (

-> venue\_id INT PRIMARY KEY,

-> venue\_name varchar(255),

-> address varchar(255)

-> );

desc venue;

+------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| venue\_id | int | NO | PRI | NULL | |

| venue\_name | varchar(255) | YES | | NULL | |

| address | varchar(255) | YES | | NULL | |

+------------+--------------+------+-----+---------+-------+

CREATE TABLE event (

-> event\_id INT PRIMARY KEY,

-> event\_name varchar(255) NOT NULL,

-> event\_date DATE,

-> event\_time TIME,

-> venue\_id INT,

-> total\_seats INT,

-> available\_seats INT,

-> ticket\_price DECIMAL(10,2),

-> event\_type ENUM('Movie','Sports','Concert') NOT NULL,

-> booking\_id INT,

-> FOREIGN KEY (venue\_id) REFERENCES venue(venue\_id)

-> );

desc event;

+-----------------+----------------------------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+----------------------------------+------+-----+---------+-------+

| event\_id | int | NO | PRI | NULL | |

| event\_name | varchar(255) | NO | | NULL | |

| event\_date | date | YES | | NULL | |

| event\_time | time | YES | | NULL | |

| venue\_id | int | YES | MUL | NULL | |

| total\_seats | int | YES | | NULL | |

| available\_seats | int | YES | | NULL | |

| ticket\_price | decimal(10,2) | YES | | NULL | |

| event\_type | enum('Movie','Sports','Concert') | NO | | NULL | |

| booking\_id | int | YES | | NULL | |

+-----------------+----------------------------------+------+-----+---------+-------+

CREATE TABLE customer (

-> customer\_id INT NOT NULL,

-> customer\_name varchar(255),

-> email varchar(255),

-> phone\_number varchar(11),

-> booking\_id INT,

-> PRIMARY KEY (customer\_id)

-> );

desc customer;

+---------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+--------------+------+-----+---------+-------+

| customer\_id | int | NO | PRI | NULL | |

| customer\_name | varchar(255) | YES | | NULL | |

| email | varchar(255) | YES | | NULL | |

| phone\_number | varchar(11) | YES | | NULL | |

| booking\_id | int | YES | | NULL | |

+---------------+--------------+------+-----+---------+-------+

CREATE TABLE booking (

-> booking\_id INT PRIMARY KEY,

-> customer\_id INT,

-> event\_id INT,

-> num\_tickets INT,

-> total\_cost varchar(255),

-> booking\_date DATE,

-> FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id),

-> FOREIGN KEY (event\_id) REFERENCES event(event\_id)

-> );

desc booking;

+--------------+--------------+------+-----+---------+-------+

| 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 | varchar(255) | YES | | NULL | |

| booking\_date | date | YES | | NULL | |

+--------------+--------------+------+-----+---------+-------+

mysql> ALTER TABLE event

-> ADD FOREIGN KEY (booking\_id) REFERENCES booking(booking\_id);

ALTER TABLE customer

-> ADD FOREIGN KEY (booking\_id) REFERENCES booking(booking\_id);

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

VALUES

(1, NULL, NULL, 4, 8000, '2024-09-05'),

(2, NULL, NULL, 2, 5000, '2024-09-10'),

(3, NULL, NULL, 6, 12000, '2024-09-12'),

(4, NULL, NULL, 3, 6000, '2024-09-18'),

(5, NULL, NULL, 5, 7000, '2024-09-22'),

(6, NULL, NULL, 7, 14000, '2024-09-25'),

(7, NULL, NULL, 1, 3000, '2024-09-27'),

(8, NULL, NULL, 4, 8000, '2024-09-30'),

(9, NULL, NULL, 8, 16000, '2024-10-01'),

(10, NULL, NULL, 2, 5000, '2024-10-03'),

(11, NULL, NULL, 4, 10000, '2024-10-05'),

(12, NULL, NULL, 1, 2000, '2024-10-10');

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

VALUES

(1, 'John Doe', 'john.doe@example.com', '9876543210', 1),

(2, 'Jane Smith', 'jane.smith@example.com', '9876540000', 2),

(3, 'Alice Brown', 'alice.brown@example.com', '9876543333', 3),

(4, 'Michael Lee', 'michael.lee@example.com', '9876544444', 4),

(5, 'David Johnson', 'david.johnson@example.com', '9876545555', 5),

(6, 'Emily White', 'emily.white@example.com', '9876546666', 6),

(7, 'Sarah Wilson', 'sarah.wilson@example.com', '9876547777', 7),

(8, 'Chris Evans', 'chris.evans@example.com', '9876548888', 8),

(9, 'Sophia Taylor', 'sophia.taylor@example.com', '9876549999', 9),

(10, 'Olivia Brown', 'olivia.brown@example.com', '9876541234', 10),

(11, 'James Black', 'james.black@example.com', '9876542345', 11),

(12, 'Liam Green', 'liam.green@example.com', '9876543456', 12);

INSERT INTO Venue (venue\_id, venue\_name, address)

VALUES

(1, 'City Concert Hall', '123 Main St'),

(2, 'Open Air Stadium', '456 Stadium Ave'),

(3, 'Grand Theater', '789 Broadway'),

(4, 'Metropolitan Arena', '101 High St'),

(5, 'Sunshine Auditorium', '202 Hilltop Ave'),

(6, 'Lakeview Stage', '303 Lakeside Rd'),

(7, 'Downtown Stadium', '404 Downtown Blvd'),

(8, 'Skyline Theater', '505 Skyline Dr'),

(9, 'Westside Pavilion', '606 Westside St'),

(10, 'Heritage Hall', '707 Heritage Rd');

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

VALUES

(1, 'Rock Concert', '2024-09-10', '19:00:00', 1, 5000, 2000, 7500, 'Concert', 1),

(2, 'Football Match', '2024-09-15', '17:00:00', 2, 10000, 5000, 6000, 'Sports', 2),

(3, 'Broadway Play', '2024-09-20', '20:00:00', 3, 2000, 800, 4000, 'Movie', 3),

(4, 'Jazz Night', '2024-09-25', '18:00:00', 4, 1500, 600, 5500, 'Concert', 4),

(5, 'Opera Show', '2024-09-28', '21:00:00', 5, 1000, 300, 8000, 'Movie', 5),

(6, 'Cricket Match', '2024-09-30', '16:00:00', 6, 15000, 5000, 5000, 'Sports', 6),

(7, 'Pop Music Festival', '2024-10-02', '19:00:00', 7, 8000, 3000, 9000, 'Concert', 7),

(8, 'Theater Play', '2024-10-05', '20:00:00', 8, 2500, 1200, 4500, 'Movie', 8),

(9, 'Boxing Championship', '2024-10-08', '18:30:00', 9, 10000, 4000, 10000, 'Sports', 9),

(10, 'Ballet Performance', '2024-10-10', '19:30:00', 10, 2000, 500, 7000, 'Movie', 10),

(11, 'Symphony Orchestra', '2024-10-12', '20:00:00', 1, 3000, 1500, 9500, 'Concert', 11),

(12, 'Stand-Up Comedy', '2024-10-15', '19:00:00', 2, 4000, 1000, 6500, 'Movie', 12);

UPDATE Booking

SET customer\_id = (SELECT customer\_id FROM Customer WHERE Booking.booking\_id = Customer.booking\_id),

event\_id = (SELECT event\_id FROM Event WHERE Booking.booking\_id = Event.booking\_id);

TASK 2

2. select \* from event;

+----------+---------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

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

+----------+---------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

| 1 | Rock Concert | 2024-09-10 | 19:00:00 | 1 | 5000 | 2000 | 7500.00 | Concert | 1 |

| 2 | Football Match | 2024-09-15 | 17:00:00 | 2 | 10000 | 5000 | 6000.00 | Sports | 2 |

| 3 | Broadway Play | 2024-09-20 | 20:00:00 | 3 | 2000 | 800 | 4000.00 | Movie | 3 |

| 4 | Jazz Night | 2024-09-25 | 18:00:00 | 4 | 1500 | 600 | 5500.00 | Concert | 4 |

| 5 | Opera Show | 2024-09-28 | 21:00:00 | 5 | 1000 | 300 | 8000.00 | Movie | 5 |

| 6 | Cricket Match | 2024-09-30 | 16:00:00 | 6 | 15000 | 5000 | 5000.00 | Sports | 6 |

| 7 | Pop Music Festival | 2024-10-02 | 19:00:00 | 7 | 8000 | 3000 | 9000.00 | Concert | 7 |

| 8 | Theater Play | 2024-10-05 | 20:00:00 | 8 | 2500 | 1200 | 4500.00 | Movie | 8 |

| 9 | Boxing Championship | 2024-10-08 | 18:30:00 | 9 | 10000 | 4000 | 10000.00 | Sports | 9 |

| 10 | Ballet Performance | 2024-10-10 | 19:30:00 | 10 | 2000 | 500 | 7000.00 | Movie | 10 |

| 11 | Symphony Orchestra | 2024-10-12 | 20:00:00 | 1 | 3000 | 1500 | 9500.00 | Concert | 11 |

| 12 | Stand-Up Comedy | 2024-10-15 | 19:00:00 | 2 | 4000 | 1000 | 6500.00 | Movie | 12 |

+----------+---------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

select event\_name from event;

+---------------------+

| event\_name |

+---------------------+

| Rock Concert |

| Football Match |

| Broadway Play |

| Jazz Night |

| Opera Show |

| Cricket Match |

| Pop Music Festival |

| Theater Play |

| Boxing Championship |

| Ballet Performance |

| Symphony Orchestra |

| Stand-Up Comedy |

+---------------------+

3.select event\_name from event

-> where available\_seats > 0;

+---------------------+

| event\_name |

+---------------------+

| Rock Concert |

| Football Match |

| Broadway Play |

| Jazz Night |

| Opera Show |

| Cricket Match |

| Pop Music Festival |

| Theater Play |

| Boxing Championship |

| Ballet Performance |

| Symphony Orchestra |

| Stand-Up Comedy |

+---------------------+

4.

mysql> select event\_name from event

-> where event\_name LIKE '%cup%';

Empty set (0.04 sec)

5. select \* from event

-> where ticket\_price BETWEEN 1000 AND 2500;

Empty set (0.00 sec)

select \* from event

-> where ticket\_price BETWEEN 4000 AND 7000;

+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

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

+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

| 2 | Football Match | 2024-09-15 | 17:00:00 | 2 | 10000 | 5000 | 6000.00 | Sports | 2 |

| 3 | Broadway Play | 2024-09-20 | 20:00:00 | 3 | 2000 | 800 | 4000.00 | Movie | 3 |

| 4 | Jazz Night | 2024-09-25 | 18:00:00 | 4 | 1500 | 600 | 5500.00 | Concert | 4 |

| 6 | Cricket Match | 2024-09-30 | 16:00:00 | 6 | 15000 | 5000 | 5000.00 | Sports | 6 |

| 8 | Theater Play | 2024-10-05 | 20:00:00 | 8 | 2500 | 1200 | 4500.00 | Movie | 8 |

| 10 | Ballet Performance | 2024-10-10 | 19:30:00 | 10 | 2000 | 500 | 7000.00 | Movie | 10 |

| 12 | Stand-Up Comedy | 2024-10-15 | 19:00:00 | 2 | 4000 | 1000 | 6500.00 | Movie | 12 |

+----------+--------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

6.select \* from event

-> where event\_date BETWEEN '2024-09-20' AND '2024-10-20';

+----------+---------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

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

+----------+---------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

| 3 | Broadway Play | 2024-09-20 | 20:00:00 | 3 | 2000 | 800 | 4000.00 | Movie | 3 |

| 4 | Jazz Night | 2024-09-25 | 18:00:00 | 4 | 1500 | 600 | 5500.00 | Concert | 4 |

| 5 | Opera Show | 2024-09-28 | 21:00:00 | 5 | 1000 | 300 | 8000.00 | Movie | 5 |

| 6 | Cricket Match | 2024-09-30 | 16:00:00 | 6 | 15000 | 5000 | 5000.00 | Sports | 6 |

| 7 | Pop Music Festival | 2024-10-02 | 19:00:00 | 7 | 8000 | 3000 | 9000.00 | Concert | 7 |

| 8 | Theater Play | 2024-10-05 | 20:00:00 | 8 | 2500 | 1200 | 4500.00 | Movie | 8 |

| 9 | Boxing Championship | 2024-10-08 | 18:30:00 | 9 | 10000 | 4000 | 10000.00 | Sports | 9 |

| 10 | Ballet Performance | 2024-10-10 | 19:30:00 | 10 | 2000 | 500 | 7000.00 | Movie | 10 |

| 11 | Symphony Orchestra | 2024-10-12 | 20:00:00 | 1 | 3000 | 1500 | 9500.00 | Concert | 11 |

| 12 | Stand-Up Comedy | 2024-10-15 | 19:00:00 | 2 | 4000 | 1000 | 6500.00 | Movie | 12 |

+----------+---------------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

7. select \* from event

-> where available\_seats > 0 AND event\_name LIKE '%concert%';

+----------+--------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

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

+----------+--------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

| 1 | Rock Concert | 2024-09-10 | 19:00:00 | 1 | 5000 | 2000 | 7500.00 | Concert | 1 |

+----------+--------------+------------+------------+----------+-------------+-----------------+--------------+------------+------------+

8. select customer\_name from customer

-> LIMIT 5 OFFSET 5;

+---------------+

| customer\_name |

+---------------+

| Emily White |

| Sarah Wilson |

| Chris Evans |

| Sophia Taylor |

| Olivia Brown |

+---------------+

9. select \* from booking

-> where num\_tickets > 4;

+------------+-------------+----------+-------------+------------+--------------+

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

+------------+-------------+----------+-------------+------------+--------------+

| 3 | 3 | 3 | 6 | 12000 | 2024-09-12 |

| 5 | 5 | 5 | 5 | 7000 | 2024-09-22 |

| 6 | 6 | 6 | 7 | 14000 | 2024-09-25 |

| 9 | 9 | 9 | 8 | 16000 | 2024-10-01 |

+------------+-------------+----------+-------------+------------+--------------+

10. select \* from customer

-> where phone\_number LIKE '%000';

+-------------+---------------+------------------------+--------------+------------+

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

+-------------+---------------+------------------------+--------------+------------+

| 2 | Jane Smith | jane.smith@example.com | 9876540000 | 2 |

+-------------+---------------+------------------------+--------------+------------+

11. select event\_name from event

-> where available\_seats > 15000

-> Order BY event\_name;

Empty set (0.04 sec)

12. select event\_name from event

-> where event\_name NOT LIKE 'X%' AND event\_name NOT LIKE 'Y%' AND event\_name NOT LIKE 'Z%';

+---------------------+

| event\_name |

+---------------------+

| Rock Concert |

| Football Match |

| Broadway Play |

| Jazz Night |

| Opera Show |

| Cricket Match |

| Pop Music Festival |

| Theater Play |

| Boxing Championship |

| Ballet Performance |

| Symphony Orchestra |

| Stand-Up Comedy |

+---------------------+

TASK 3

1. select event\_name, avg(ticket\_price) AS average\_ticket\_price

-> from event

-> GROUP BY event\_name;

+---------------------+----------------------+

| event\_name | average\_ticket\_price |

+---------------------+----------------------+

| Rock Concert | 7500.000000 |

| Football Match | 6000.000000 |

| Broadway Play | 4000.000000 |

| Jazz Night | 5500.000000 |

| Opera Show | 8000.000000 |

| Cricket Match | 5000.000000 |

| Pop Music Festival | 9000.000000 |

| Theater Play | 4500.000000 |

| Boxing Championship | 10000.000000 |

| Ballet Performance | 7000.000000 |

| Symphony Orchestra | 9500.000000 |

| Stand-Up Comedy | 6500.000000 |

+---------------------+----------------------+

2. select sum(total\_cost) AS revenue\_generated

-> from booking;

+-------------------+

| revenue\_generated |

+-------------------+

| 96000 |

+-------------------+

3. select e.event\_name, sum(b.num\_tickets) AS total\_tickets\_sold

-> from event e

-> JOIN booking b ON e.event\_id = b.event\_id

-> GROUP BY e.event\_name

-> ORDER BY total\_tickets\_sold DESC

-> LIMIT 1;

+---------------------+--------------------+

| event\_name | total\_tickets\_sold |

+---------------------+--------------------+

| Boxing Championship | 8 |

+---------------------+--------------------+

4. select e.event\_name, sum(b.num\_tickets) as total\_tickets\_sold

-> from event e

-> join booking b on e.event\_id = b.event\_id

-> group by e.event\_id;

+---------------------+--------------------+

| event\_name | total\_tickets\_sold |

+---------------------+--------------------+

| Rock Concert | 4 |

| Football Match | 2 |

| Broadway Play | 6 |

| Jazz Night | 3 |

| Opera Show | 5 |

| Cricket Match | 7 |

| Pop Music Festival | 1 |

| Theater Play | 4 |

| Boxing Championship | 8 |

| Ballet Performance | 2 |

| Symphony Orchestra | 4 |

| Stand-Up Comedy | 1 |

+---------------------+--------------------+

5. SELECT e.event\_id, e.event\_name

-> FROM Event e

-> LEFT JOIN Booking b ON e.event\_id = b.event\_id

-> WHERE b.event\_id IS NULL;

Empty set (0.00 sec)

6. select c.customer\_name, sum(b.num\_tickets) AS tickets\_bought

-> from customer c

-> JOIN booking b ON c.customer\_id = b.customer\_id

-> GROUP BY c.customer\_name

-> ORDER BY tickets\_bought desc

-> LIMIT 1;

+---------------+----------------+

| customer\_name | tickets\_bought |

+---------------+----------------+

| Sophia Taylor | 8 |

+---------------+----------------+

7. select e.event\_name , MONTH(booking\_date) as month, SUM(b.num\_tickets) AS tickets\_sold

-> from event e

-> JOIN booking b ON e.event\_id = b.event\_id

-> GROUP BY month,e.event\_name

-> ORDER BY month;

+---------------------+-------+--------------+

| event\_name | month | tickets\_sold |

+---------------------+-------+--------------+

| Broadway Play | 9 | 6 |

| Cricket Match | 9 | 7 |

| Football Match | 9 | 2 |

| Jazz Night | 9 | 3 |

| Opera Show | 9 | 5 |

| Pop Music Festival | 9 | 1 |

| Rock Concert | 9 | 4 |

| Theater Play | 9 | 4 |

| Ballet Performance | 10 | 2 |

| Boxing Championship | 10 | 8 |

| Stand-Up Comedy | 10 | 1 |

| Symphony Orchestra | 10 | 4 |

+---------------------+-------+--------------+

8. select v.venue\_name, avg(ticket\_price) as avg\_ticket\_price

-> from venue v

-> JOIN event e ON v.venue\_id = e.venue\_id

-> group by v.venue\_id;

+---------------------+------------------+

| venue\_name | avg\_ticket\_price |

+---------------------+------------------+

| City Concert Hall | 8500.000000 |

| Open Air Stadium | 6250.000000 |

| Grand Theater | 4000.000000 |

| Metropolitan Arena | 5500.000000 |

| Sunshine Auditorium | 8000.000000 |

| Lakeview Stage | 5000.000000 |

| Downtown Stadium | 9000.000000 |

| Skyline Theater | 4500.000000 |

| Westside Pavilion | 10000.000000 |

| Heritage Hall | 7000.000000 |

+---------------------+------------------+

9. select e.event\_name, sum(b.num\_tickets) as total\_tickets\_sold

-> from event e

-> JOIN booking b ON e.event\_id = b.event\_id

-> GROUP BY e.event\_name;

+---------------------+--------------------+

| event\_name | total\_tickets\_sold |

+---------------------+--------------------+

| Rock Concert | 4 |

| Football Match | 2 |

| Broadway Play | 6 |

| Jazz Night | 3 |

| Opera Show | 5 |

| Cricket Match | 7 |

| Pop Music Festival | 1 |

| Theater Play | 4 |

| Boxing Championship | 8 |

| Ballet Performance | 2 |

| Symphony Orchestra | 4 |

| Stand-Up Comedy | 1 |

+---------------------+--------------------+

10. select YEAR(event\_date) AS event\_year, sum(b.total\_cost) AS total\_revenue

-> from event e

-> JOIN booking b ON e.event\_id = b.event\_id

-> Group by event\_year;

+------------+---------------+

| event\_year | total\_revenue |

+------------+---------------+

| 2024 | 96000 |

+------------+---------------+

11. SELECT c.customer\_name, COUNT(DISTINCT b.event\_id) AS number\_of\_events

-> FROM Customer c

-> JOIN Booking b ON c.customer\_id = b.customer\_id

-> GROUP BY c.customer\_id

-> HAVING number\_of\_events > 1;

Empty set (0.04 sec)

12. SELECT c.customer\_name, SUM(b.total\_cost) AS total\_revenue

-> FROM Customer c

-> JOIN Booking b ON c.customer\_id = b.customer\_id

-> GROUP BY c.customer\_id;

+---------------+---------------+

| customer\_name | total\_revenue |

+---------------+---------------+

| John Doe | 8000 |

| Jane Smith | 5000 |

| Alice Brown | 12000 |

| Michael Lee | 6000 |

| David Johnson | 7000 |

| Emily White | 14000 |

| Sarah Wilson | 3000 |

| Chris Evans | 8000 |

| Sophia Taylor | 16000 |

| Olivia Brown | 5000 |

| James Black | 10000 |

| Liam Green | 2000 |

+---------------+---------------+

13. SELECT e.event\_type, v.venue\_name, AVG(e.ticket\_price) AS average\_ticket\_price

-> FROM Event e

-> JOIN Venue v ON e.venue\_id = v.venue\_id

-> GROUP BY e.event\_type, v.venue\_id;

+------------+---------------------+----------------------+

| event\_type | venue\_name | average\_ticket\_price |

+------------+---------------------+----------------------+

| Concert | City Concert Hall | 8500.000000 |

| Sports | Open Air Stadium | 6000.000000 |

| Movie | Open Air Stadium | 6500.000000 |

| Movie | Grand Theater | 4000.000000 |

| Concert | Metropolitan Arena | 5500.000000 |

| Movie | Sunshine Auditorium | 8000.000000 |

| Sports | Lakeview Stage | 5000.000000 |

| Concert | Downtown Stadium | 9000.000000 |

| Movie | Skyline Theater | 4500.000000 |

| Sports | Westside Pavilion | 10000.000000 |

| Movie | Heritage Hall | 7000.000000 |

+------------+---------------------+----------------------+

14. SELECT c.customer\_name, SUM(b.num\_tickets) AS total\_tickets

-> FROM Customer c

-> JOIN Booking b ON c.customer\_id = b.customer\_id

-> WHERE b.booking\_date >= DATE\_SUB(CURDATE(), INTERVAL 30 DAY)

-> GROUP BY c.customer\_id;

+---------------+---------------+

| customer\_name | total\_tickets |

+---------------+---------------+

| John Doe | 4 |

| Jane Smith | 2 |

| Alice Brown | 6 |

| Michael Lee | 3 |

| David Johnson | 5 |

| Emily White | 7 |

| Sarah Wilson | 1 |

| Chris Evans | 4 |

| Sophia Taylor | 8 |

| Olivia Brown | 2 |

| James Black | 4 |

| Liam Green | 1 |

+---------------+---------------+

Task 4

1. 146 (42S02): Table 'ticket\_booking\_system.venu' doesn't exist

mysql> SELECT v.venue\_id, v.venue\_name,

-> (SELECT AVG(e.ticket\_price)

-> FROM Event e

-> WHERE e.venue\_id = v.venue\_id) AS average\_ticket\_price

-> FROM Venue v;

+----------+---------------------+----------------------+

| venue\_id | venue\_name | average\_ticket\_price |

+----------+---------------------+----------------------+

| 1 | City Concert Hall | 8500.000000 |

| 2 | Open Air Stadium | 6250.000000 |

| 3 | Grand Theater | 4000.000000 |

| 4 | Metropolitan Arena | 5500.000000 |

| 5 | Sunshine Auditorium | 8000.000000 |

| 6 | Lakeview Stage | 5000.000000 |

| 7 | Downtown Stadium | 9000.000000 |

| 8 | Skyline Theater | 4500.000000 |

| 9 | Westside Pavilion | 10000.000000 |

| 10 | Heritage Hall | 7000.000000 |

+----------+---------------------+----------------------+

2. SELECT e.event\_id, e.event\_name, e.available\_seats, e.total\_seats

-> FROM Event e

-> WHERE (e.total\_seats - e.available\_seats) > (0.5 \* e.total\_seats);

+----------+---------------------+-----------------+-------------+

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

+----------+---------------------+-----------------+-------------+

| 1 | Rock Concert | 2000 | 5000 |

| 3 | Broadway Play | 800 | 2000 |

| 4 | Jazz Night | 600 | 1500 |

| 5 | Opera Show | 300 | 1000 |

| 6 | Cricket Match | 5000 | 15000 |

| 7 | Pop Music Festival | 3000 | 8000 |

| 8 | Theater Play | 1200 | 2500 |

| 9 | Boxing Championship | 4000 | 10000 |

| 10 | Ballet Performance | 500 | 2000 |

| 12 | Stand-Up Comedy | 1000 | 4000 |

+----------+---------------------+-----------------+-------------+

3. SELECT e.event\_id, e.event\_name, SUM(b.num\_tickets) AS total\_tickets\_sold

-> FROM Event e

-> JOIN Booking b ON e.event\_id = b.event\_id

-> GROUP BY e.event\_id, e.event\_name;

+----------+---------------------+--------------------+

| event\_id | event\_name | total\_tickets\_sold |

+----------+---------------------+--------------------+

| 1 | Rock Concert | 4 |

| 2 | Football Match | 2 |

| 3 | Broadway Play | 6 |

| 4 | Jazz Night | 3 |

| 5 | Opera Show | 5 |

| 6 | Cricket Match | 7 |

| 7 | Pop Music Festival | 1 |

| 8 | Theater Play | 4 |

| 9 | Boxing Championship | 8 |

| 10 | Ballet Performance | 2 |

| 11 | Symphony Orchestra | 4 |

| 12 | Stand-Up Comedy | 1 |

+----------+---------------------+--------------------+

4.

mysql> SELECT c.customer\_id, c.customer\_name

-> FROM Customer c

-> WHERE NOT EXISTS (

-> SELECT 1

-> FROM Booking b

-> WHERE b.customer\_id = c.customer\_id

-> );

Empty set (0.05 sec)

5. SELECT et.event\_type, SUM(total\_tickets\_sold) AS total\_sold

-> FROM (

-> SELECT e.event\_type, SUM(b.num\_tickets) AS total\_tickets\_sold

-> FROM Event e

-> JOIN Booking b ON e.event\_id = b.event\_id

-> GROUP BY e.event\_type, e.event\_id

-> ) AS et

-> GROUP BY et.event\_type;

+------------+------------+

| event\_type | total\_sold |

+------------+------------+

| Concert | 12 |

| Sports | 17 |

| Movie | 18 |

+------------+------------+

6. SELECT et.event\_type, SUM(total\_tickets\_sold) AS total\_sold

-> FROM (

-> SELECT e.event\_type, SUM(b.num\_tickets) AS total\_tickets\_sold

-> FROM Event e

-> JOIN Booking b ON e.event\_id = b.event\_id

-> GROUP BY e.event\_type, e.event\_id

-> ) AS et

-> GROUP BY et.event\_type;

+------------+------------+

| event\_type | total\_sold |

+------------+------------+

| Concert | 12 |

| Sports | 17 |

| Movie | 18 |

+------------+------------+

7. SELECT e.event\_id, e.event\_name, e.ticket\_price

-> FROM Event e

-> WHERE e.ticket\_price > (

-> SELECT AVG(ticket\_price)

-> FROM Event

-> );

+----------+---------------------+--------------+

| event\_id | event\_name | ticket\_price |

+----------+---------------------+--------------+

| 1 | Rock Concert | 7500.00 |

| 5 | Opera Show | 8000.00 |

| 7 | Pop Music Festival | 9000.00 |

| 9 | Boxing Championship | 10000.00 |

| 10 | Ballet Performance | 7000.00 |

| 11 | Symphony Orchestra | 9500.00 |

+----------+---------------------+--------------+

8. SELECT c.customer\_id, c.customer\_name,

-> (SELECT SUM(b.total\_cost)

-> FROM Booking b

-> WHERE b.customer\_id = c.customer\_id) AS total\_revenue

-> FROM Customer c;

+-------------+---------------+---------------+

| customer\_id | customer\_name | total\_revenue |

+-------------+---------------+---------------+

| 1 | John Doe | 8000 |

| 2 | Jane Smith | 5000 |

| 3 | Alice Brown | 12000 |

| 4 | Michael Lee | 6000 |

| 5 | David Johnson | 7000 |

| 6 | Emily White | 14000 |

| 7 | Sarah Wilson | 3000 |

| 8 | Chris Evans | 8000 |

| 9 | Sophia Taylor | 16000 |

| 10 | Olivia Brown | 5000 |

| 11 | James Black | 10000 |

| 12 | Liam Green | 2000 |

+-------------+---------------+---------------+

9. SELECT DISTINCT c.customer\_id, c.customer\_name

-> FROM Customer c

-> WHERE c.customer\_id IN (

-> SELECT b.customer\_id

-> FROM Booking b

-> JOIN Event e ON b.event\_id = e.event\_id

-> WHERE e.venue\_id = 3

-> );

+-------------+---------------+

| customer\_id | customer\_name |

+-------------+---------------+

| 3 | Alice Brown |

+-------------+---------------+

10. SELECT e.event\_type, SUM(b.num\_tickets) AS total\_tickets\_sold

-> FROM Event e

-> JOIN Booking b ON e.event\_id = b.event\_id

-> GROUP BY e.event\_type;

+------------+--------------------+

| event\_type | total\_tickets\_sold |

+------------+--------------------+

| Concert | 12 |

| Sports | 17 |

| Movie | 18 |

+------------+--------------------+

11. SELECT c.customer\_id, c.customer\_name

-> FROM Customer c

-> WHERE c.customer\_id IN (

-> SELECT b.customer\_id

-> FROM Booking b

-> JOIN Event e ON b.event\_id = e.event\_id

-> WHERE DATE\_FORMAT(e.event\_date, '%Y-%m') = '2024-09' -- Specify the desired month here

-> );

+-------------+---------------+

| customer\_id | customer\_name |

+-------------+---------------+

| 1 | John Doe |

| 2 | Jane Smith |

| 3 | Alice Brown |

| 4 | Michael Lee |

| 5 | David Johnson |

| 6 | Emily White |

+-------------+---------------+

12. SELECT v.venue\_id, v.venue\_name,

-> (SELECT AVG(e.ticket\_price)

-> FROM Event e

-> WHERE e.venue\_id = v.venue\_id) AS average\_ticket\_price

-> FROM Venue v;

+----------+---------------------+----------------------+

| venue\_id | venue\_name | average\_ticket\_price |

+----------+---------------------+----------------------+

| 1 | City Concert Hall | 8500.000000 |

| 2 | Open Air Stadium | 6250.000000 |

| 3 | Grand Theater | 4000.000000 |

| 4 | Metropolitan Arena | 5500.000000 |

| 5 | Sunshine Auditorium | 8000.000000 |

| 6 | Lakeview Stage | 5000.000000 |

| 7 | Downtown Stadium | 9000.000000 |

| 8 | Skyline Theater | 4500.000000 |

| 9 | Westside Pavilion | 10000.000000 |

| 10 | Heritage Hall | 7000.000000 |

+----------+---------------------+----------------------+