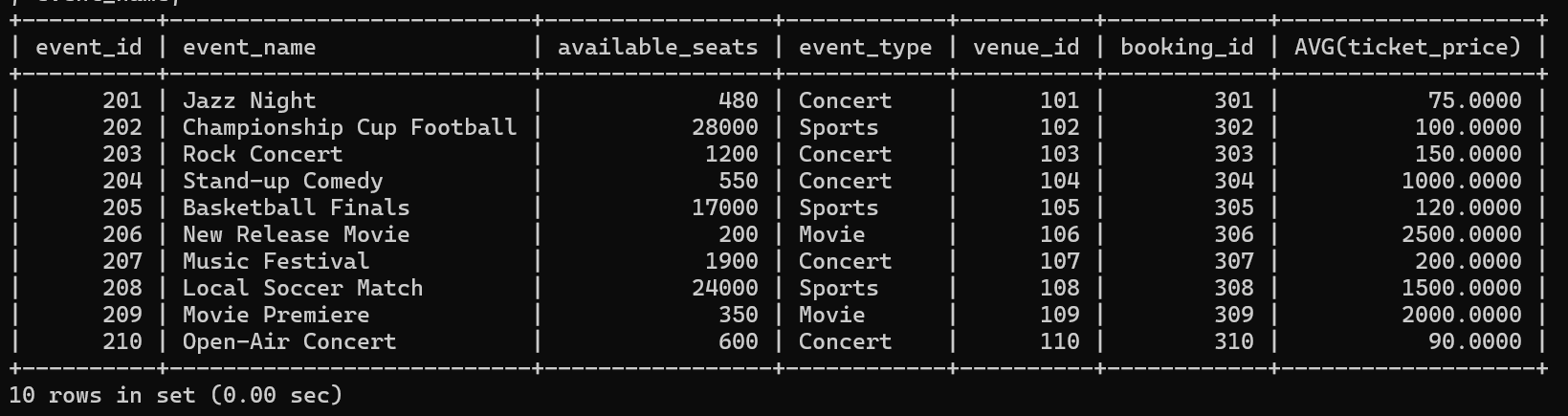
TASK 3 – TICKET BOOKING SYSTEM

(Aggregate functions, Having, Order By, GroupBy and Joins)

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

mysql> SELECT event\_id, event\_name, available\_seats, event\_type, venue\_id, booking\_id, AVG(ticket\_price) FROM Event GROUP BY event\_id

, event\_name;



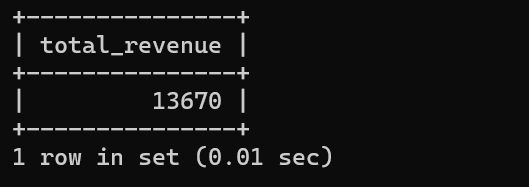
1. Write a SQL query to Calculate the Total Revenue Generated by Events.

mysql> SELECT SUM(b.num\_tickets \* e.ticket\_price) AS total\_revenue

-> FROM Booking b

-> INNER JOIN

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



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

mysql> SELECT e.event\_name, e.event\_id, b.num\_tickets FROM Event e INNER JOIN Booking b ON e.event\_id = b.event\_id ORDER BY num\_ticke

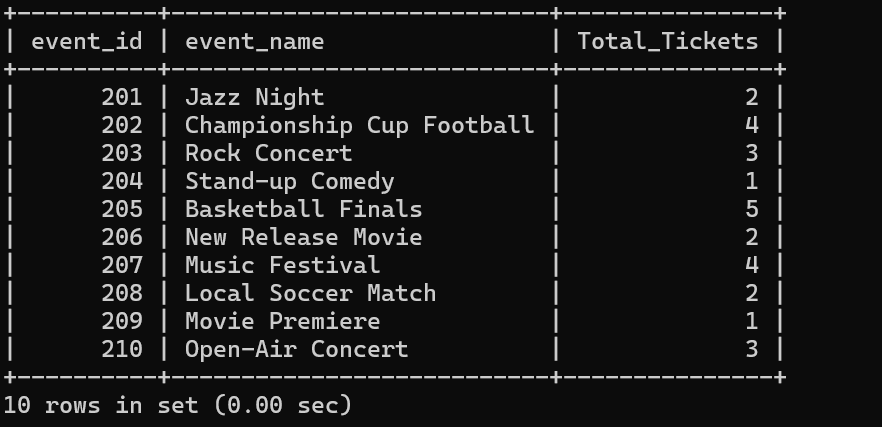
ts DESC;



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

mysql> SELECT DISTINCT e.event\_id, e.event\_name, SUM(b.num\_tickets) AS Total\_Tickets FROM Event e INNER JOIN Booking b ON e.event\_id

= b.event\_id GROUP BY e.event\_id, e.event\_name;



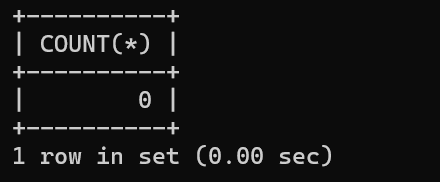
1. Write a SQL query to Find Events with No Ticket Sales.

mysql> SELECT COUNT(\*)

-> FROM Event e

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

-> WHERE b.event\_id IS NULL;



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

mysql> SELECT c.customer\_id, c.customer\_name, SUM(b.num\_tickets) AS total\_tickets

-> FROM Customers c

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

-> GROUP BY c.customer\_id, c.customer\_name

-> ORDER BY total\_tickets DESC;



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

mysql> SELECT e.event\_id, e.event\_name, MONTH(b.booking\_date) AS month,SUM(b.num\_tickets) AS total\_tickets

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

-> GROUP BY e.event\_id, e.event\_name, MONTH(b.booking\_date)

-> ;



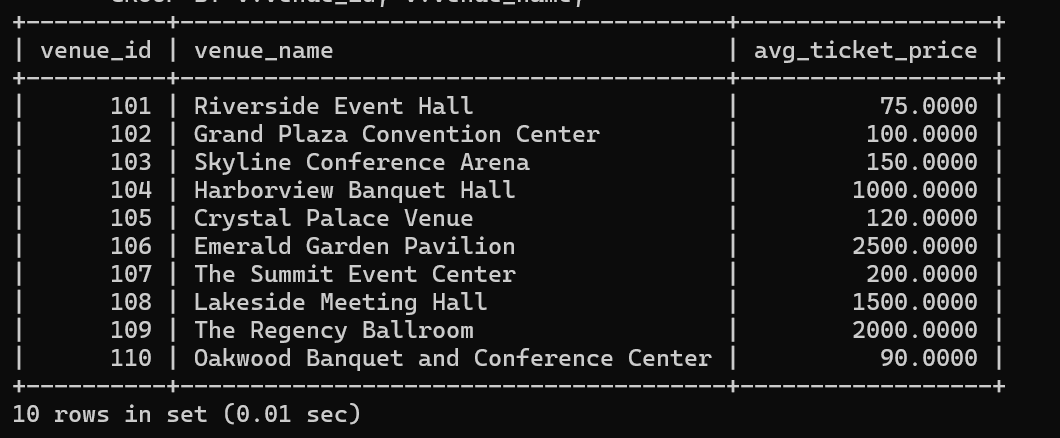
1. Write a SQL query to calculate the average Ticket Price for Events in Each Venue.

mysql> SELECT v.venue\_id, v.venue\_name, AVG(e.ticket\_price) AS avg\_ticket\_price

-> FROM Venu v INNER JOIN Event e

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

-> GROUP BY v.venue\_id, v.venue\_name;



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

mysql> SELECT

-> e.event\_type,

-> SUM(b.num\_tickets) AS total\_tickets\_sold FROM Event e JOIN

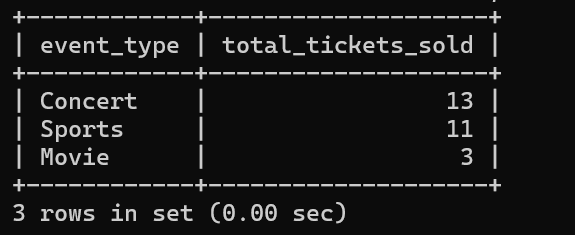
-> Booking b ON e.event\_id = b.event\_id

-> GROUP BY

-> e.event\_type

-> ORDER BY

-> total\_tickets\_sold DESC;



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

mysql> SELECT

-> YEAR(b.booking\_date) AS year,

-> SUM(b.num\_tickets \* e.ticket\_price) AS total\_revenue FROM

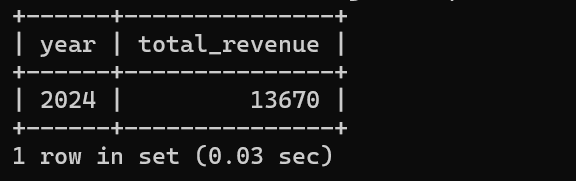
-> Booking b JOIN Event e ON b.event\_id = e.event\_id

-> GROUP BY

-> YEAR(b.booking\_date)

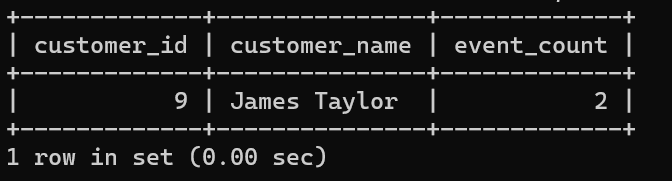
-> ORDER BY

-> YEAR(b.booking\_date);



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

mysql> SELECT c.customer\_id, c.customer\_name, COUNT(b.event\_id) AS event\_count FROM Customers c INNER JOIN Booking b ON c.customer\_id = b.customer\_id GROUP BY c.customer\_id, c.customer\_name HAVING event\_count > 1;



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

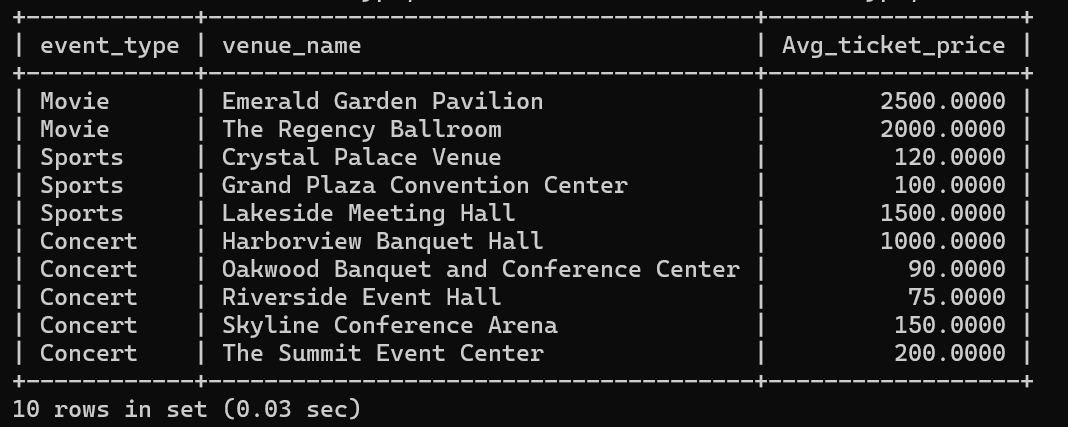
mysql> SELECT c.customer\_id, c.customer\_name, SUM(b.num\_tickets \* e.ticket\_price) AS total\_revenue FROM Customers c INNER JOIN Booking b ON c.customer\_id = b.customer\_id INNER JOIN Event e ON b.event\_id = e.event\_id GROUP BY c.customer\_id, c.customer\_name ORDER BY total\_revenue DESC;



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

mysql> SELECT e.event\_type, v.venue\_name, AVG(e.ticket\_price) AS Avg\_ticket\_price FROM Event e INNER JOIN Venu v ON e.venue\_id = v.venue\_id

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



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

mysql> SELECT c.customer\_id, c.customer\_name, SUM(b.num\_tickets) AS total\_tickets\_purchased FROM Customers c INNER JOIN Booking b ON c.customer\_id = b.customer\_id WHERE b.booking\_date >= CURDATE() - INTERVAL 30 DAY

-> GROUP BY c.customer\_id, c.customer\_name

-> ORDER BY total\_tickets\_purchased DESC;

