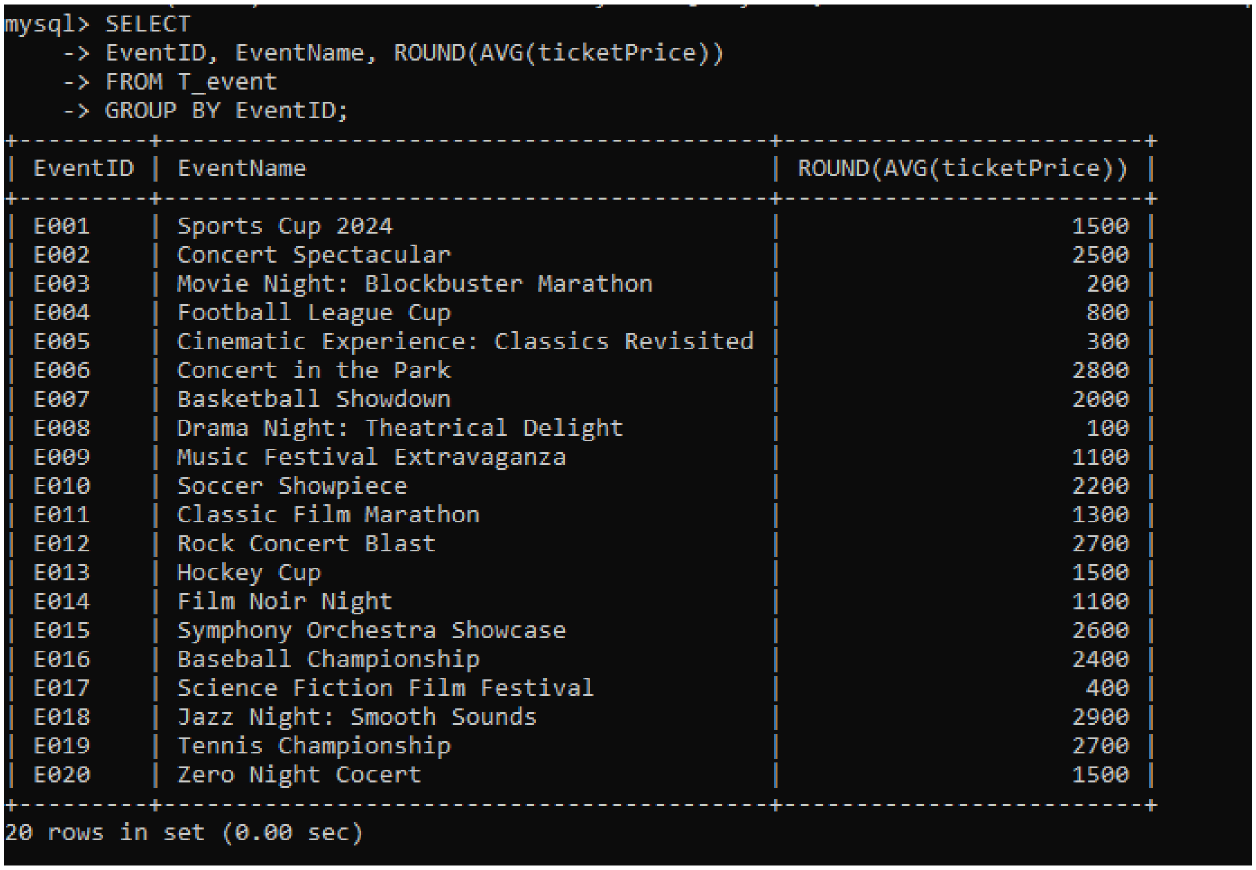
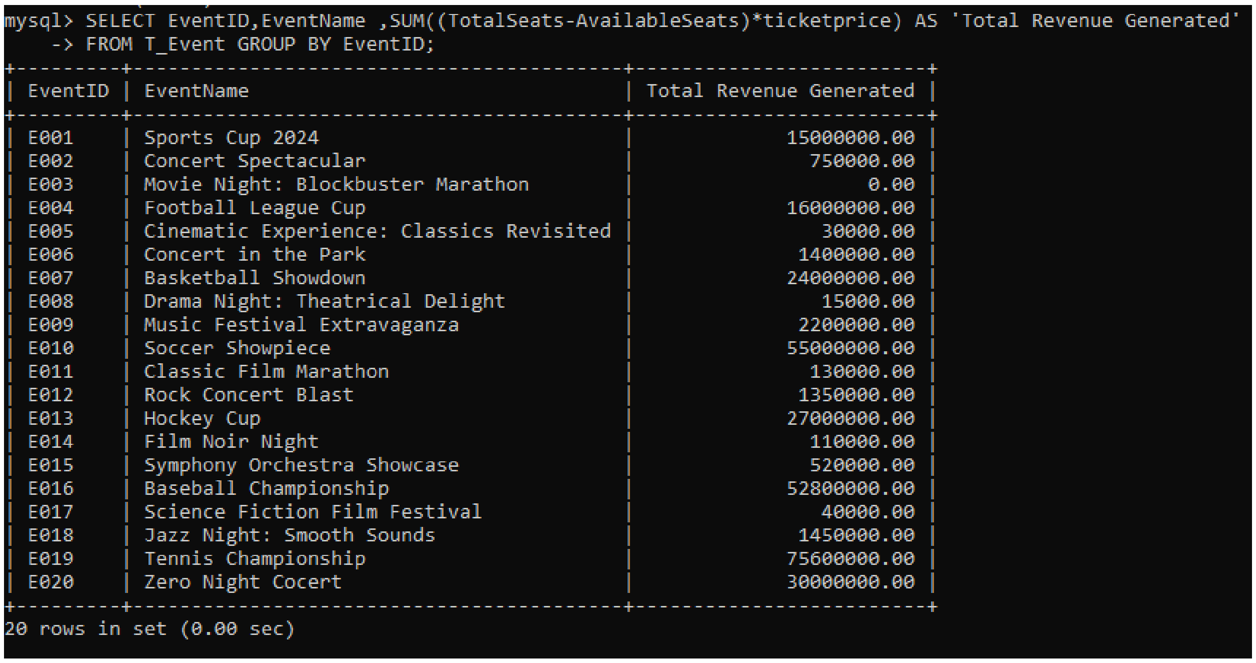
Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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



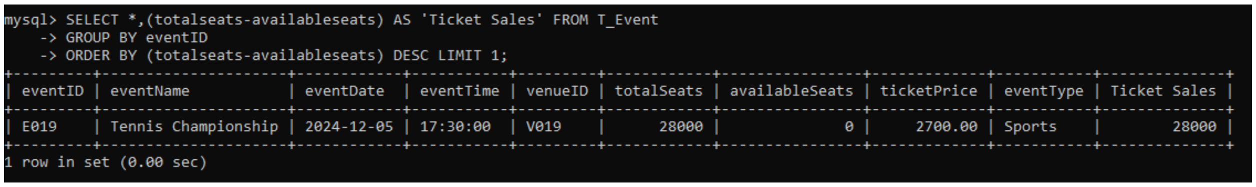
```
mysql> SELECT
    -> EventID, EventName, ROUND(AVG(ticketPrice))
    -> FROM T_event
    -> GROUP BY EventID;
+---------+-----------------------------------------+-------------------------+
| EventID | EventName                               | ROUND(AVG(ticketPrice)) |
+---------+-----------------------------------------+-------------------------+
| E001    | Sports Cup 2024                         |                    1500 |
| E002    | Concert Spectacular                     |                    2500 |
| E003    | Movie Night: Blockbuster Marathon       |                     200 |
| E004    | Football League Cup                     |                     800 |
| E005    | Cinematic Experience: Classics Revisited|                     300 |
| E006    | Concert in the Park                     |                    2800 |
| E007    | Basketball Showdown                     |                    2000 |
| E008    | Drama Night: Theatrical Delight         |                     100 |
| E009    | Music Festival Extravaganza             |                    1100 |
| E010    | Soccer Showpiece                        |                    2200 |
| E011    | Classic Film Marathon                   |                    1300 |
| E012    | Rock Concert Blast                      |                    2700 |
| E013    | Hockey Cup                              |                    1500 |
| E014    | Film Noir Night                         |                    1100 |
| E015    | Symphony Orchestra Showcase             |                    2600 |
| E016    | Baseball Championship                   |                    2400 |
| E017    | Science Fiction Film Festival           |                     400 |
| E018    | Jazz Night: Smooth Sounds               |                    2900 |
| E019    | Tennis Championship                     |                    2700 |
| E020    | Zero Night Cocert                       |                    1500 |
+---------+-----------------------------------------+-------------------------+
20 rows in set (0.00 sec)
```

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



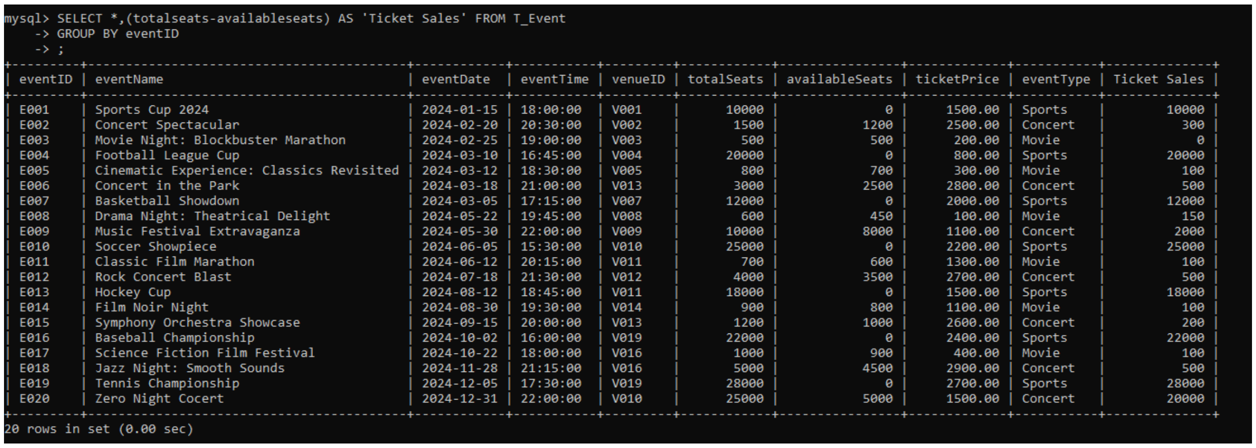
```
mysql> SELECT EventID,EventName ,SUM((TotalSeats-AvailableSeats)*ticketprice) AS 'Total Revenue Generated'
    -> FROM T_Event GROUP BY EventID;
+---------+------------------------------------------+-------------------------+
| EventID | EventName                                | Total Revenue Generated |
+---------+------------------------------------------+-------------------------+
| E001    | Sports Cup 2024                          |             15000000.00 |
| E002    | Concert Spectacular                      |               750000.00 |
| E003    | Movie Night: Blockbuster Marathon        |                    0.00 |
| E004    | Football League Cup                      |             16000000.00 |
| E005    | Cinematic Experience: Classics Revisited |                30000.00 |
| E006    | Concert in the Park                      |              1400000.00 |
| E007    | Basketball Showdown                      |             24000000.00 |
| E008    | Drama Night: Theatrical Delight          |                15000.00 |
| E009    | Music Festival Extravaganza              |              2200000.00 |
| E010    | Soccer Showpiece                         |             55000000.00 |
| E011    | Classic Film Marathon                    |               130000.00 |
| E012    | Rock Concert Blast                       |              1350000.00 |
| E013    | Hockey Cup                               |             27000000.00 |
| E014    | Film Noir Night                          |               110000.00 |
| E015    | Symphony Orchestra Showcase              |               520000.00 |
| E016    | Baseball Championship                    |             52800000.00 |
| E017    | Science Fiction Film Festival            |                40000.00 |
| E018    | Jazz Night: Smooth Sounds                |              1450000.00 |
| E019    | Tennis Championship                      |             75600000.00 |
| E020    | Zero Night Cocert                        |             30000000.00 |
+---------+------------------------------------------+-------------------------+
20 rows in set (0.00 sec)
```

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



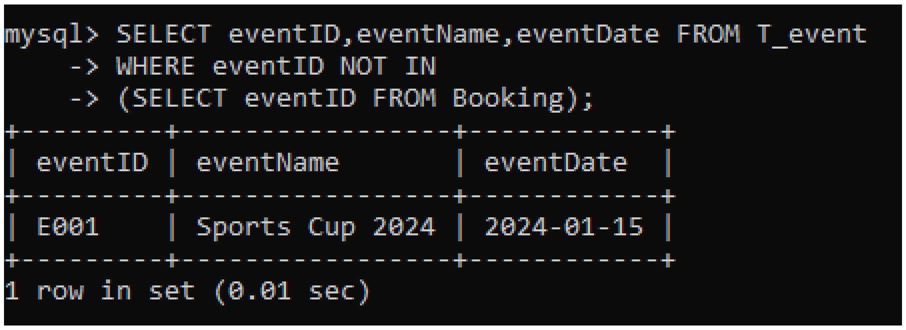
```
mysql> SELECT *,(totalseats-availableseats) AS 'Ticket Sales' FROM T_Event
    -> GROUP BY eventID
    -> ORDER BY (totalseats-availableseats) DESC LIMIT 1;
+---------+---------------------+------------+-----------+---------+------------+----------------+-------------+-----------+--------------+
| eventID | eventName           | eventDate  | eventTime | venueID | totalSeats | availableSeats | ticketPrice | eventType | Ticket Sales |
+---------+---------------------+------------+-----------+---------+------------+----------------+-------------+-----------+--------------+
| E019    | Tennis Championship | 2024-12-05 | 17:30:00  | V019    |      28000 |              0 |     2700.00 | Sports    |        28000 |
+---------+---------------------+------------+-----------+---------+------------+----------------+-------------+-----------+--------------+
1 row in set (0.00 sec)
```

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



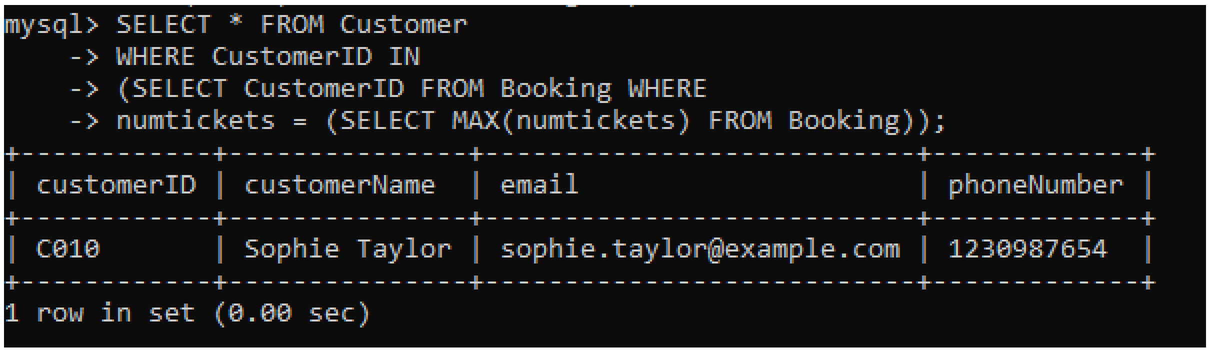
```
mysql> SELECT *,(totalseats-availableseats) AS 'Ticket Sales' FROM T_Event
    -> GROUP BY eventID
    -> ;
+---------+------------------------------------------+------------+-----------+---------+------------+----------------+-------------+-----------+--------------+
| eventID | eventName                                | eventDate  | eventTime | venueID | totalSeats | availableSeats | ticketPrice | eventType | Ticket Sales |
+---------+------------------------------------------+------------+-----------+---------+------------+----------------+-------------+-----------+--------------+
| E001    | Sports Cup 2024                          | 2024-01-15 | 18:00:00  | V001    |      10000 |              0 |     1500.00 | Sports    |        10000 |
| E002    | Concert Spectacular                      | 2024-02-20 | 20:30:00  | V002    |       1500 |           1200 |     2500.00 | Concert   |          300 |
| E003    | Movie Night: Blockbuster Marathon        | 2024-02-25 | 19:00:00  | V003    |        500 |            500 |      200.00 | Movie     |            0 |
| E004    | Football League Cup                      | 2024-03-10 | 16:45:00  | V004    |      20000 |              0 |      800.00 | Sports    |        20000 |
| E005    | Cinematic Experience: Classics Revisited | 2024-03-12 | 18:30:00  | V005    |        800 |            700 |      300.00 | Movie     |          100 |
| E006    | Concert in the Park                      | 2024-03-18 | 21:00:00  | V013    |       3000 |           2500 |     2800.00 | Concert   |          500 |
| E007    | Basketball Showdown                      | 2024-03-05 | 17:15:00  | V007    |      12000 |              0 |     2000.00 | Sports    |        12000 |
| E008    | Drama Night: Theatrical Delight          | 2024-05-22 | 19:45:00  | V008    |        600 |            450 |      100.00 | Movie     |          150 |
| E009    | Music Festival Extravaganza              | 2024-05-30 | 22:00:00  | V009    |      10000 |           8000 |     1100.00 | Concert   |         2000 |
| E010    | Soccer Showpiece                         | 2024-06-05 | 15:30:00  | V010    |      25000 |              0 |     2200.00 | Sports    |        25000 |
| E011    | Classic Film Marathon                    | 2024-06-12 | 20:15:00  | V011    |        700 |            600 |     1300.00 | Movie     |          100 |
| E012    | Rock Concert Blast                       | 2024-07-18 | 21:30:00  | V012    |       4000 |           3500 |     2700.00 | Concert   |          500 |
| E013    | Hockey Cup                               | 2024-08-12 | 18:45:00  | V011    |      18000 |              0 |     1500.00 | Sports    |        18000 |
| E014    | Film Noir Night                          | 2024-08-30 | 19:30:00  | V014    |        900 |            800 |     1100.00 | Movie     |          100 |
| E015    | Symphony Orchestra Showcase              | 2024-09-15 | 20:00:00  | V013    |       1200 |           1000 |     2600.00 | Concert   |          200 |
| E016    | Baseball Championship                    | 2024-10-02 | 16:00:00  | V019    |      22000 |              0 |     2400.00 | Sports    |        22000 |
| E017    | Science Fiction Film Festival            | 2024-10-22 | 18:00:00  | V016    |       1000 |            900 |      400.00 | Movie     |          100 |
| E018    | Jazz Night: Smooth Sounds                | 2024-11-28 | 21:15:00  | V016    |       5000 |           4500 |     2900.00 | Concert   |          500 |
| E019    | Tennis Championship                      | 2024-12-05 | 17:30:00  | V019    |      28000 |              0 |     2700.00 | Sports    |        28000 |
| E020    | Zero Night Cocert                        | 2024-12-31 | 22:00:00  | V010    |      25000 |           5000 |     1500.00 | Concert   |        20000 |
+---------+------------------------------------------+------------+-----------+---------+------------+----------------+-------------+-----------+--------------+
20 rows in set (0.00 sec)
```

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



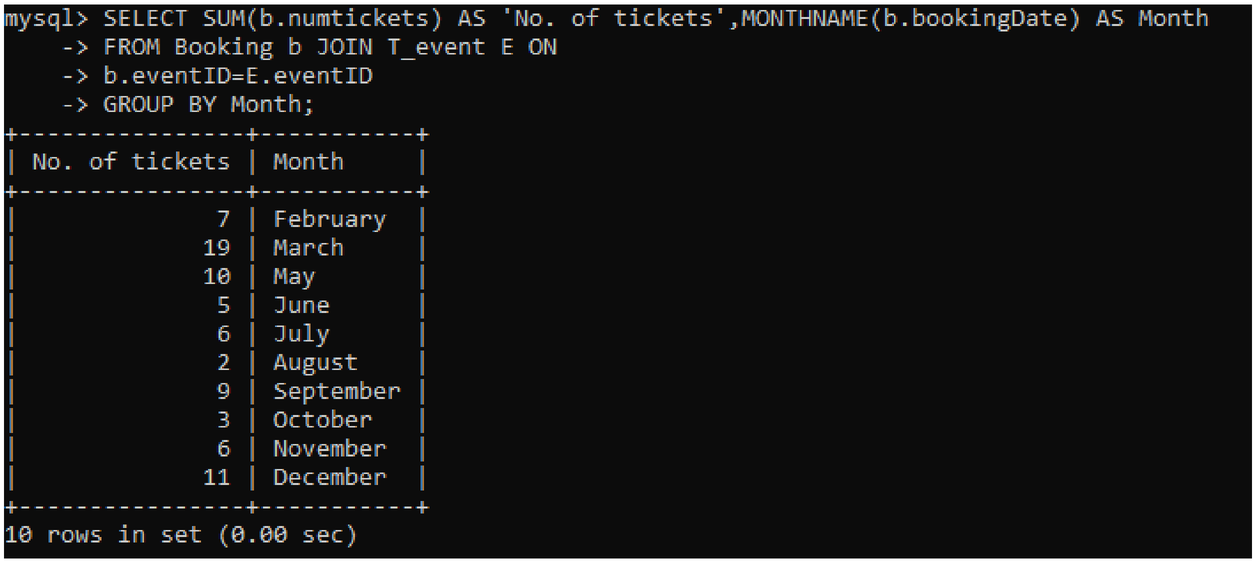
```
mysql> SELECT eventID,eventName,eventDate FROM T_event
    -> WHERE eventID NOT IN
    -> (SELECT eventID FROM Booking);
+---------+-----------------+------------+
| eventID | eventName       | eventDate  |
+---------+-----------------+------------+
| E001    | Sports Cup 2024 | 2024-01-15 |
+---------+-----------------+------------+
1 row in set (0.01 sec)
```

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



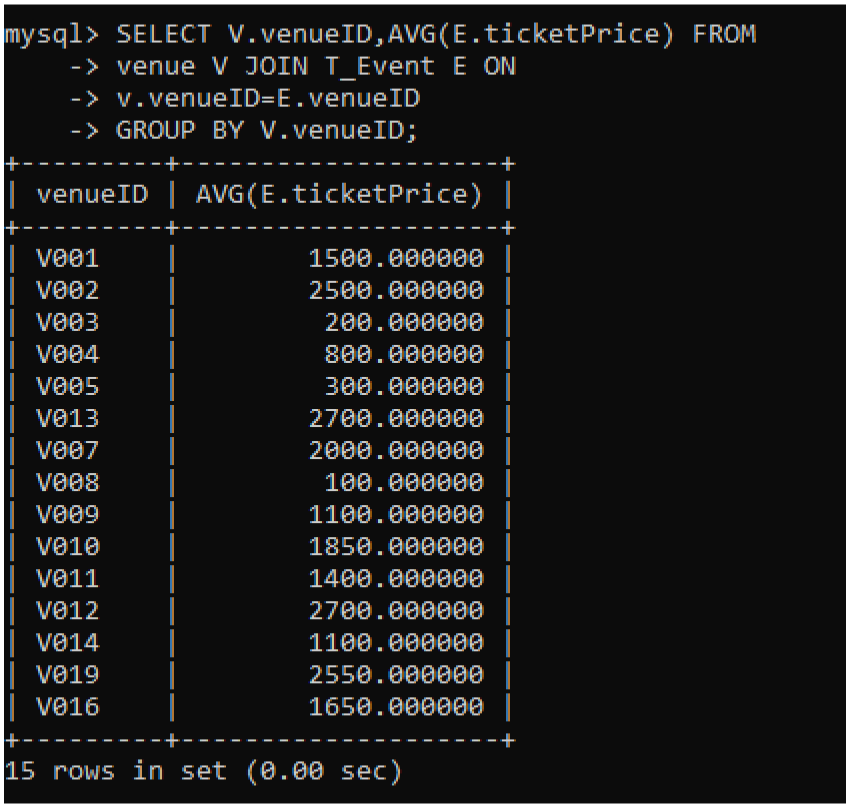
```
mysql> SELECT * FROM Customer
    -> WHERE CustomerID IN
    -> (SELECT CustomerID FROM Booking WHERE
    -> numtickets = (SELECT MAX(numtickets) FROM Booking));
+------------+---------------+---------------------------+-------------+
| customerID | customerName  | email                     | phoneNumber |
+------------+---------------+---------------------------+-------------+
| C010       | Sophie Taylor | sophie.taylor@example.com | 1230987654  |
+------------+---------------+---------------------------+-------------+
1 row in set (0.00 sec)
```

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



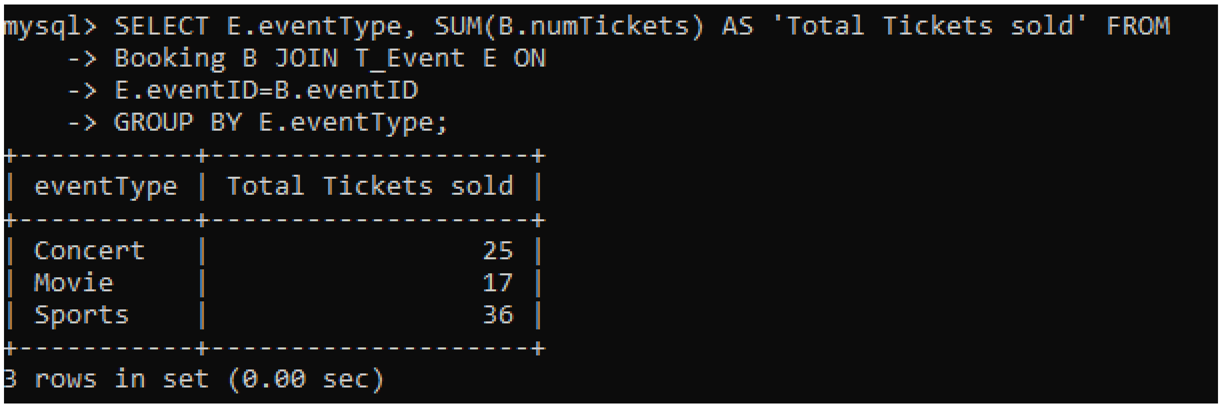
```
mysql> SELECT SUM(b.numtickets) AS 'No. of tickets',MONTHNAME(b.bookingDate) AS Month
    -> FROM Booking b JOIN T_event E ON
    -> b.eventID=E.eventID
    -> GROUP BY Month;
+----------------+-----------+
| No. of tickets | Month     |
+----------------+-----------+
|              7 | February  |
|             19 | March     |
|             10 | May       |
|              5 | June      |
|              6 | July      |
|              2 | August    |
|              9 | September |
|              3 | October   |
|              6 | November  |
|             11 | December  |
+----------------+-----------+
10 rows in set (0.00 sec)
```

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



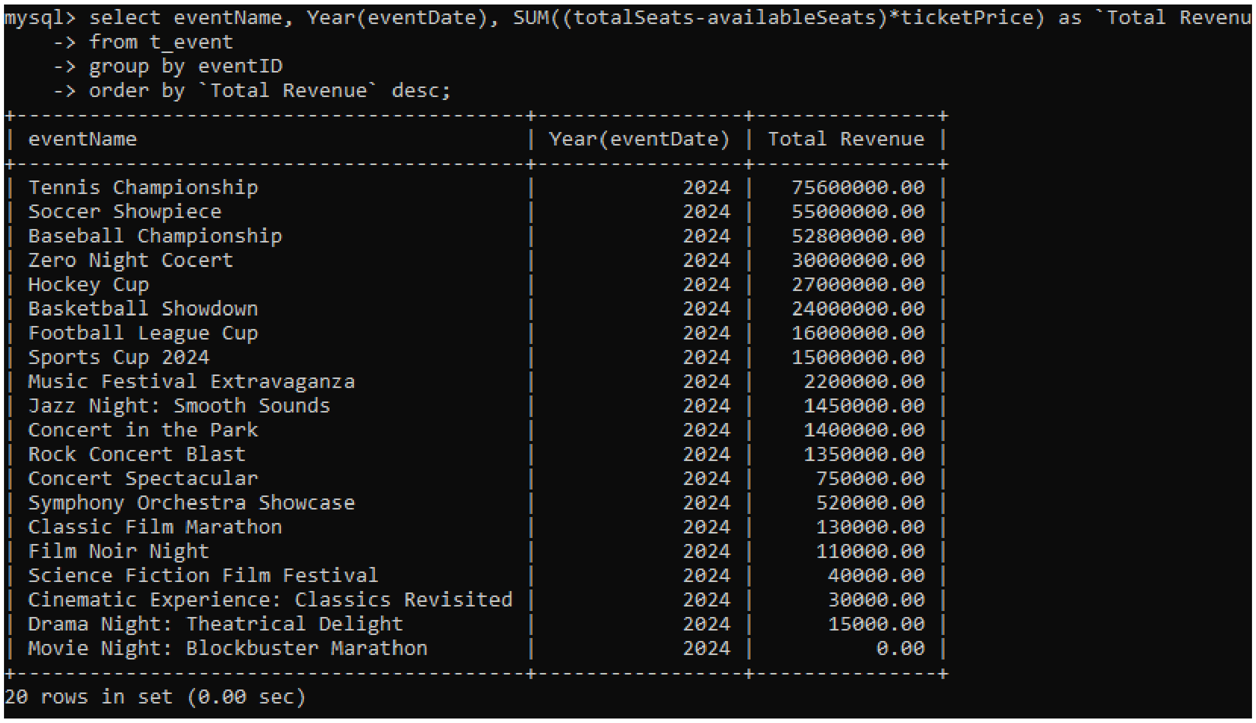
```
mysql> SELECT V.venueID,AVG(E.ticketPrice) FROM
    -> venue V JOIN T_Event E ON
    -> v.venueID=E.venueID
    -> GROUP BY V.venueID;
+---------+--------------------+
| venueID | AVG(E.ticketPrice) |
+---------+--------------------+
| V001    |        1500.000000 |
| V002    |        2500.000000 |
| V003    |         200.000000 |
| V004    |         800.000000 |
| V005    |         300.000000 |
| V013    |        2700.000000 |
| V007    |        2000.000000 |
| V008    |         100.000000 |
| V009    |        1100.000000 |
| V010    |        1850.000000 |
| V011    |        1400.000000 |
| V012    |        2700.000000 |
| V014    |        1100.000000 |
| V019    |        2550.000000 |
| V016    |        1650.000000 |
+---------+--------------------+
15 rows in set (0.00 sec)
```

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



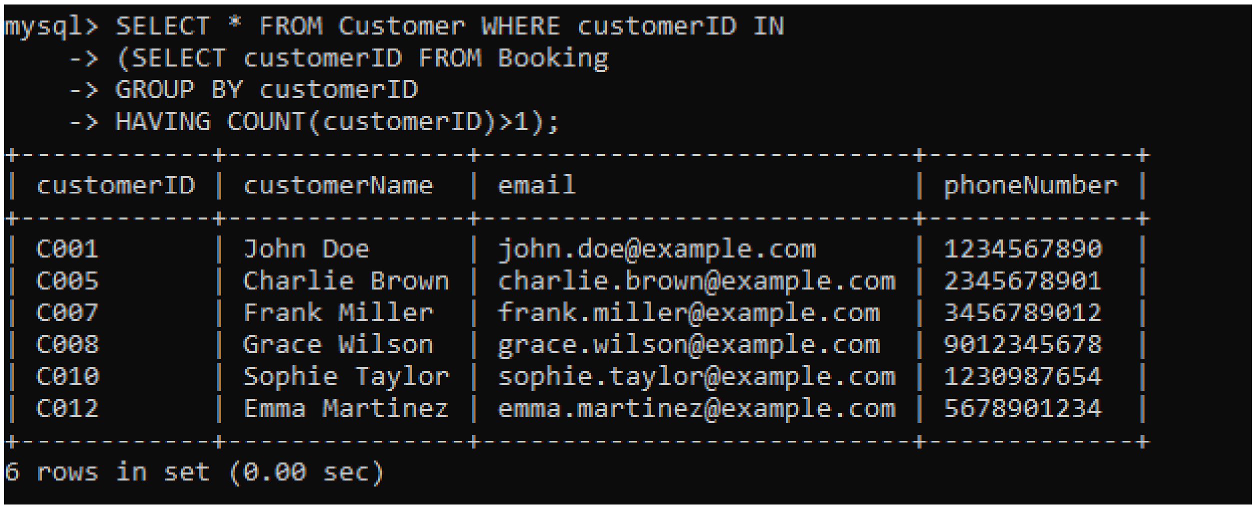
```
mysql> SELECT E.eventType, SUM(B.numTickets) AS 'Total Tickets sold' FROM
    -> Booking B JOIN T_Event E ON
    -> E.eventID=B.eventID
    -> GROUP BY E.eventType;
+-----------+--------------------+
| eventType | Total Tickets sold |
+-----------+--------------------+
| Concert   |                 25 |
| Movie     |                 17 |
| Sports    |                 36 |
+-----------+--------------------+
3 rows in set (0.00 sec)
```

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



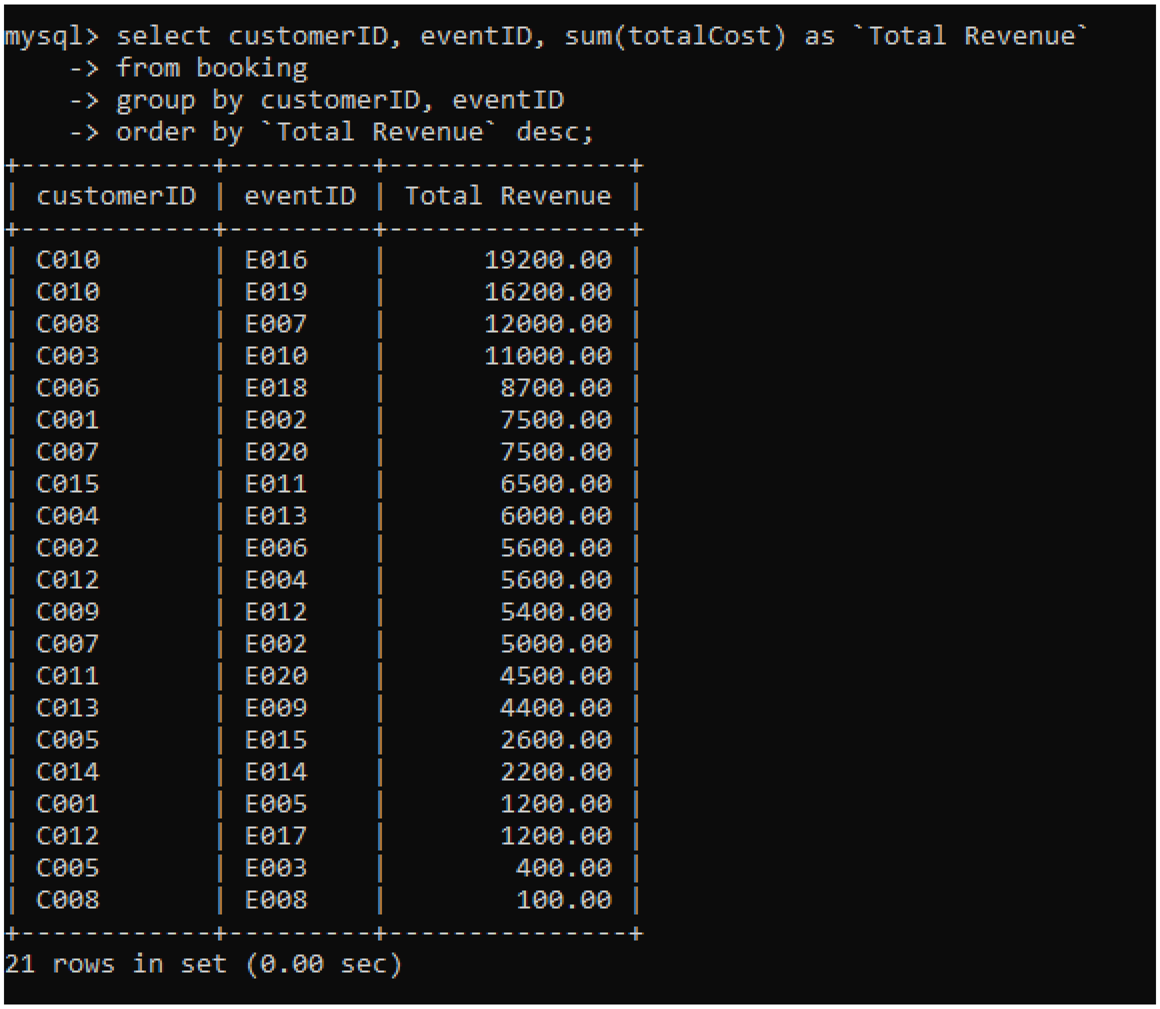
```
mysql> select eventName, Year(eventDate), SUM((totalSeats-availableSeats)*ticketPrice) as `Total Revenu
    -> from t_event
    -> group by eventID
    -> order by `Total Revenue` desc;
+---------------------------------------+-----------------+---------------+
| eventName                             | Year(eventDate) | Total Revenue |
+---------------------------------------+-----------------+---------------+
| Tennis Championship                   |            2024 |   75600000.00 |
| Soccer Showpiece                      |            2024 |   55000000.00 |
| Baseball Championship                 |            2024 |   52800000.00 |
| Zero Night Cocert                     |            2024 |   30000000.00 |
| Hockey Cup                            |            2024 |   27000000.00 |
| Basketball Showdown                   |            2024 |   24000000.00 |
| Football League Cup                   |            2024 |   16000000.00 |
| Sports Cup 2024                       |            2024 |   15000000.00 |
| Music Festival Extravaganza           |            2024 |    2200000.00 |
| Jazz Night: Smooth Sounds             |            2024 |    1450000.00 |
| Concert in the Park                   |            2024 |    1400000.00 |
| Rock Concert Blast                    |            2024 |    1350000.00 |
| Concert Spectacular                   |            2024 |     750000.00 |
| Symphony Orchestra Showcase           |            2024 |     520000.00 |
| Classic Film Marathon                 |            2024 |     130000.00 |
| Film Noir Night                       |            2024 |     110000.00 |
| Science Fiction Film Festival         |            2024 |      40000.00 |
| Cinematic Experience: Classics Revisited |         2024 |      30000.00 |
| Drama Night: Theatrical Delight       |            2024 |      15000.00 |
| Movie Night: Blockbuster Marathon     |            2024 |          0.00 |
+---------------------------------------+-----------------+---------------+
20 rows in set (0.00 sec)
```

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



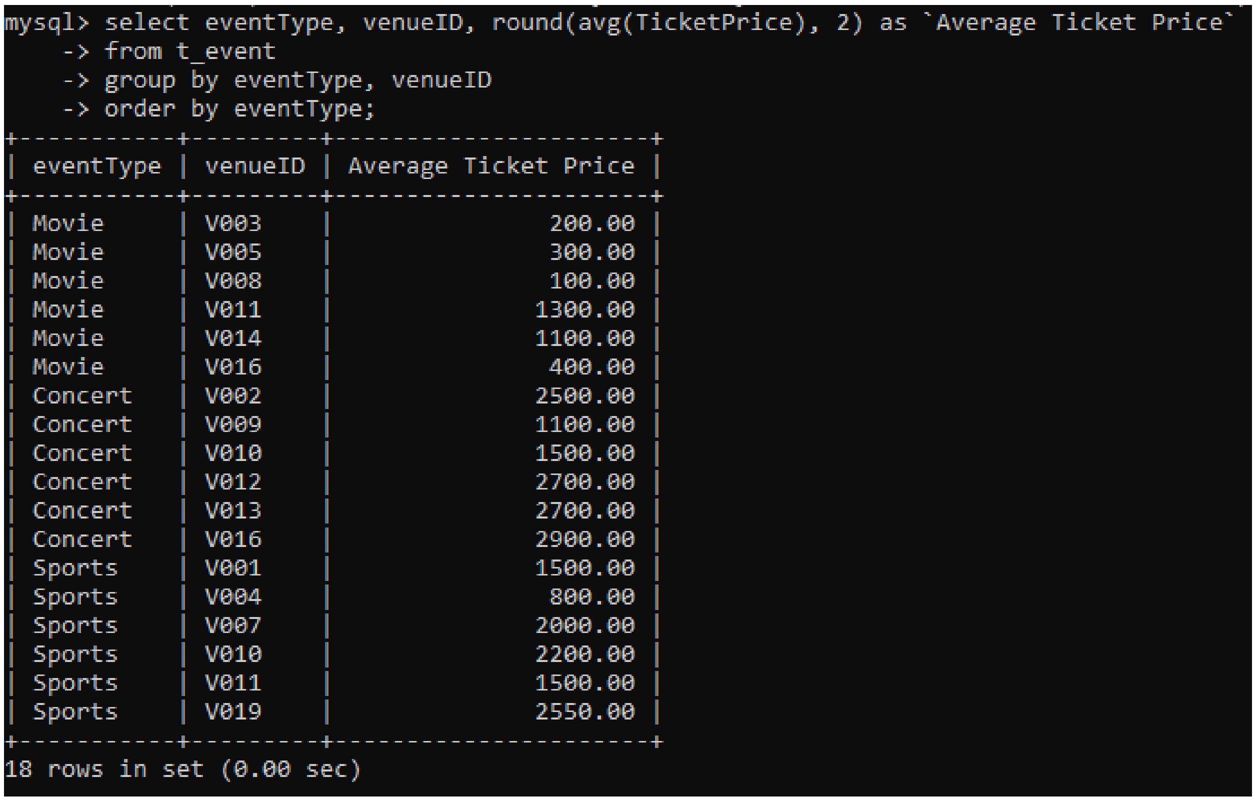
```
mysql> SELECT * FROM Customer WHERE customerID IN
    -> (SELECT customerID FROM Booking
    -> GROUP BY customerID
    -> HAVING COUNT(customerID)>1);
+------------+---------------+----------------------------+-------------+
| customerID | customerName  | email                      | phoneNumber |
+------------+---------------+----------------------------+-------------+
| C001       | John Doe      | john.doe@example.com       | 1234567890  |
| C005       | Charlie Brown | charlie.brown@example.com  | 2345678901  |
| C007       | Frank Miller  | frank.miller@example.com   | 3456789012  |
| C008       | Grace Wilson  | grace.wilson@example.com   | 9012345678  |
| C010       | Sophie Taylor | sophie.taylor@example.com  | 1230987654  |
| C012       | Emma Martinez | emma.martinez@example.com  | 5678901234  |
+------------+---------------+----------------------------+-------------+
6 rows in set (0.00 sec)
```

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



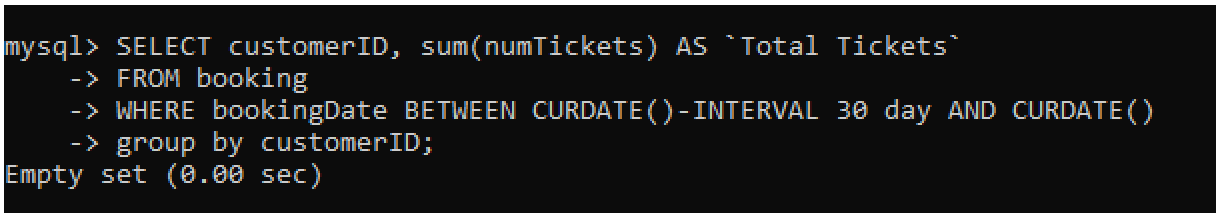
```
mysql> select customerID, eventID, sum(totalCost) as `Total Revenue`
    -> from booking
    -> group by customerID, eventID
    -> order by `Total Revenue` desc;
+------------+---------+---------------+
| customerID | eventID | Total Revenue |
+------------+---------+---------------+
| C010       | E016    |      19200.00 |
| C010       | E019    |      16200.00 |
| C008       | E007    |      12000.00 |
| C003       | E010    |      11000.00 |
| C006       | E018    |       8700.00 |
| C001       | E002    |       7500.00 |
| C007       | E020    |       7500.00 |
| C015       | E011    |       6500.00 |
| C004       | E013    |       6000.00 |
| C002       | E006    |       5600.00 |
| C012       | E004    |       5600.00 |
| C009       | E012    |       5400.00 |
| C007       | E002    |       5000.00 |
| C011       | E020    |       4500.00 |
| C013       | E009    |       4400.00 |
| C005       | E015    |       2600.00 |
| C014       | E014    |       2200.00 |
| C001       | E005    |       1200.00 |
| C012       | E017    |       1200.00 |
| C005       | E003    |        400.00 |
| C008       | E008    |        100.00 |
+------------+---------+---------------+
21 rows in set (0.00 sec)
```

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



```
mysql> select eventType, venueID, round(avg(TicketPrice), 2) as `Average Ticket Price`
    -> from t_event
    -> group by eventType, venueID
    -> order by eventType;
+-----------+---------+----------------------+
| eventType | venueID | Average Ticket Price |
+-----------+---------+----------------------+
| Movie     | V003    |               200.00 |
| Movie     | V005    |               300.00 |
| Movie     | V008    |               100.00 |
| Movie     | V011    |              1300.00 |
| Movie     | V014    |              1100.00 |
| Movie     | V016    |               400.00 |
| Concert   | V002    |              2500.00 |
| Concert   | V009    |              1100.00 |
| Concert   | V010    |              1500.00 |
| Concert   | V012    |              2700.00 |
| Concert   | V013    |              2700.00 |
| Concert   | V016    |              2900.00 |
| Sports    | V001    |              1500.00 |
| Sports    | V004    |               800.00 |
| Sports    | V007    |              2000.00 |
| Sports    | V010    |              2200.00 |
| Sports    | V011    |              1500.00 |
| Sports    | V019    |              2550.00 |
+-----------+---------+----------------------+
18 rows in set (0.00 sec)
```

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



```
mysql> SELECT customerID, sum(numTickets) AS `Total Tickets`
    -> FROM booking
    -> WHERE bookingDate BETWEEN CURDATE()-INTERVAL 30 day AND CURDATE()
    -> group by customerID;
Empty set (0.00 sec)
```