Akilesh K

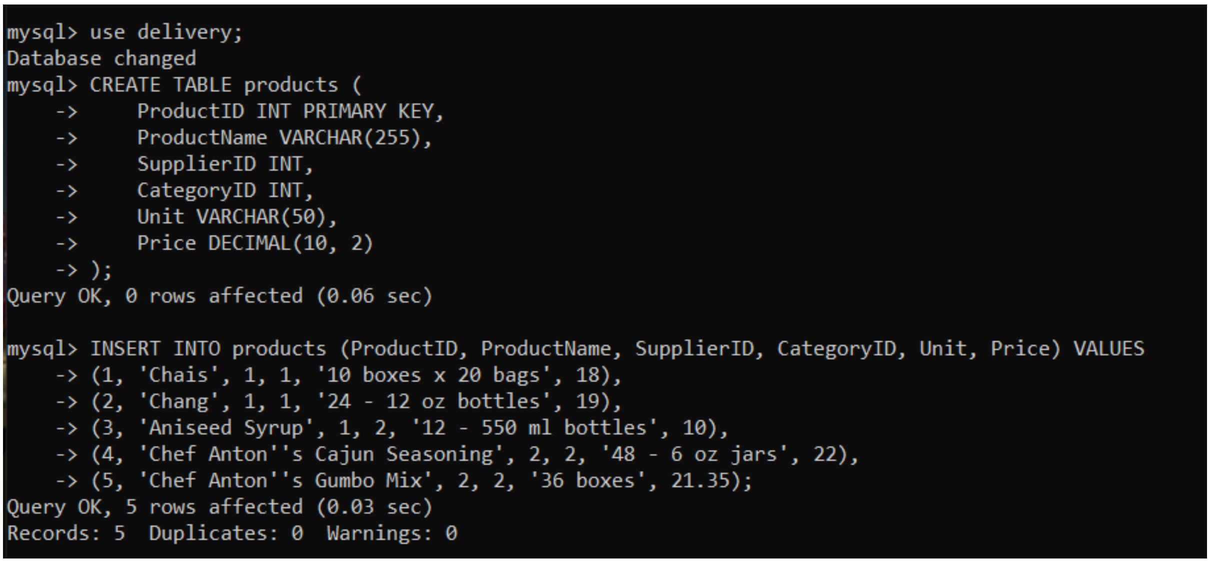
k.akilesh123@gmail.com

Data engineering - Batch 1

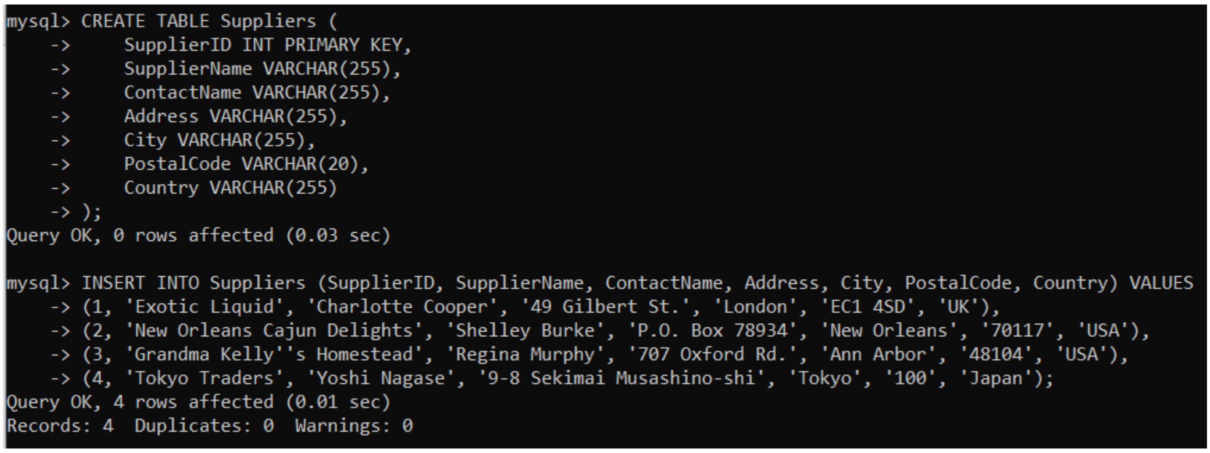
Date: 22-01-24

SQL- EXISTS , SUBQUERY

Table for finding exists

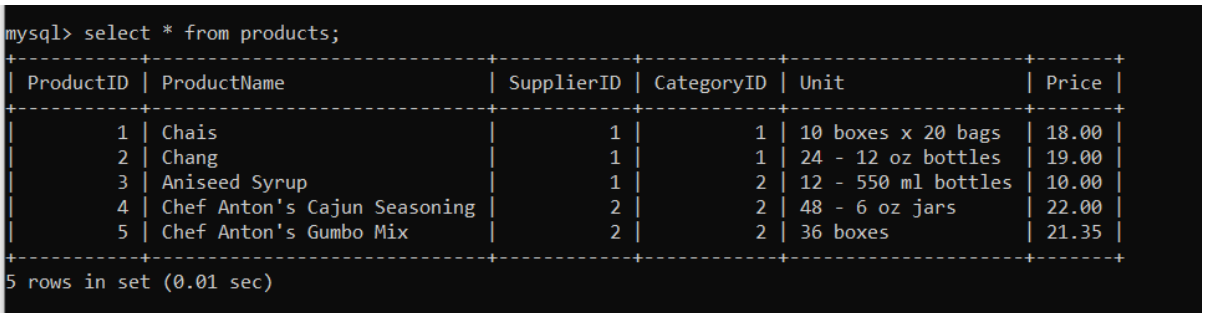


```
mysql> use delivery;
Database changed
mysql> CREATE TABLE products (
    ->     ProductID INT PRIMARY KEY,
    ->     ProductName VARCHAR(255),
    ->     SupplierID INT,
    ->     CategoryID INT,
    ->     Unit VARCHAR(50),
    ->     Price DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO products (ProductID, ProductName, SupplierID, CategoryID, Unit, Price) VALUES
    -> (1, 'Chais', 1, 1, '10 boxes x 20 bags', 18),
    -> (2, 'Chang', 1, 1, '24 - 12 oz bottles', 19),
    -> (3, 'Aniseed Syrup', 1, 2, '12 - 550 ml bottles', 10),
    -> (4, 'Chef Anton''s Cajun Seasoning', 2, 2, '48 - 6 oz jars', 22),
    -> (5, 'Chef Anton''s Gumbo Mix', 2, 2, '36 boxes', 21.35);
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0
```



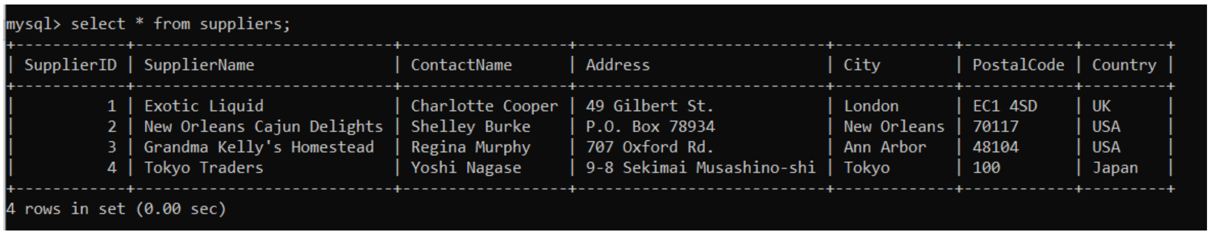
```
mysql> CREATE TABLE Suppliers (
    ->     SupplierID INT PRIMARY KEY,
    ->     SupplierName VARCHAR(255),
    ->     ContactName VARCHAR(255),
    ->     Address VARCHAR(255),
    ->     City VARCHAR(255),
    ->     PostalCode VARCHAR(20),
    ->     Country VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO Suppliers (SupplierID, SupplierName, ContactName, Address, City, PostalCode, Country) VALUES
    -> (1, 'Exotic Liquid', 'Charlotte Cooper', '49 Gilbert St.', 'London', 'EC1 4SD', 'UK'),
    -> (2, 'New Orleans Cajun Delights', 'Shelley Burke', 'P.O. Box 78934', 'New Orleans', '70117', 'USA'),
    -> (3, 'Grandma Kelly''s Homestead', 'Regina Murphy', '707 Oxford Rd.', 'Ann Arbor', '48104', 'USA'),
    -> (4, 'Tokyo Traders', 'Yoshi Nagase', '9-8 Sekimai Musashino-shi', 'Tokyo', '100', 'Japan');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Display products



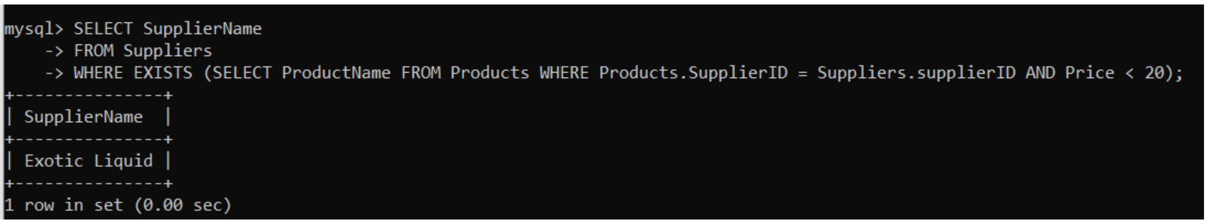
```
mysql> select * from products;
+-----------+------------------------------+------------+------------+---------------------+-------+
| ProductID | ProductName                  | SupplierID | CategoryID | Unit                | Price |
+-----------+------------------------------+------------+------------+---------------------+-------+
|         1 | Chais                        |          1 |          1 | 10 boxes x 20 bags  | 18.00 |
|         2 | Chang                        |          1 |          1 | 24 - 12 oz bottles  | 19.00 |
|         3 | Aniseed Syrup                |          1 |          2 | 12 - 550 ml bottles | 10.00 |
|         4 | Chef Anton's Cajun Seasoning |          2 |          2 | 48 - 6 oz jars      | 22.00 |
|         5 | Chef Anton's Gumbo Mix       |          2 |          2 | 36 boxes            | 21.35 |
+-----------+------------------------------+------------+------------+---------------------+-------+
5 rows in set (0.01 sec)
```

Display supliers



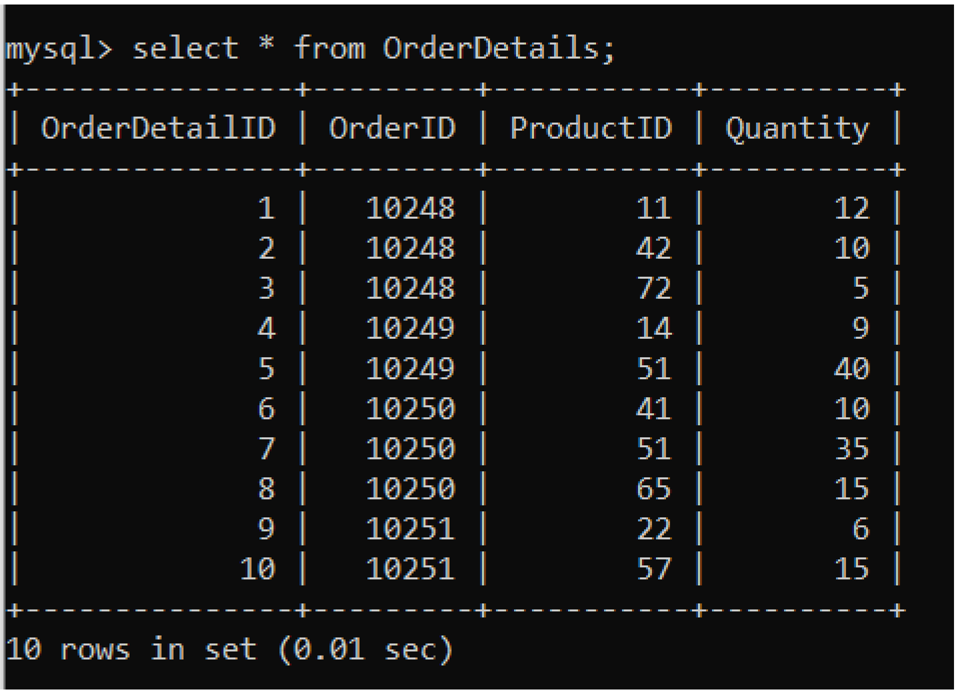
```
mysql> select * from suppliers;
+------------+----------------------------+------------------+---------------------------+-------------+------------+---------+
| SupplierID | SupplierName               | ContactName      | Address                   | City        | PostalCode | Country |
+------------+----------------------------+------------------+---------------------------+-------------+------------+---------+
|          1 | Exotic Liquid              | Charlotte Cooper | 49 Gilbert St.            | London      | EC1 4SD    | UK      |
|          2 | New Orleans Cajun Delights | Shelley Burke    | P.O. Box 78934            | New Orleans | 70117      | USA     |
|          3 | Grandma Kelly's Homestead  | Regina Murphy    | 707 Oxford Rd.            | Ann Arbor   | 48104      | USA     |
|          4 | Tokyo Traders              | Yoshi Nagase     | 9-8 Sekimai Musashino-shi | Tokyo       | 100        | Japan   |
+------------+----------------------------+------------------+---------------------------+-------------+------------+---------+
4 rows in set (0.00 sec)
```

Where exists



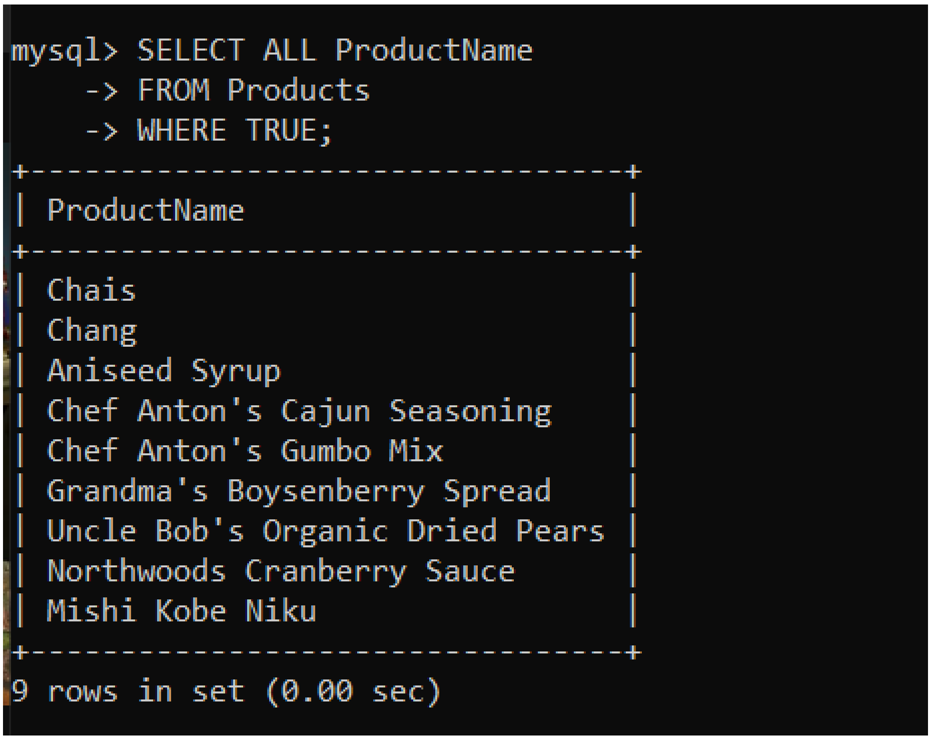
```
mysql> SELECT SupplierName
    -> FROM Suppliers
    -> WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);
+---------------+
| SupplierName  |
+---------------+
| Exotic Liquid |
+---------------+
1 row in set (0.00 sec)
```

Order details



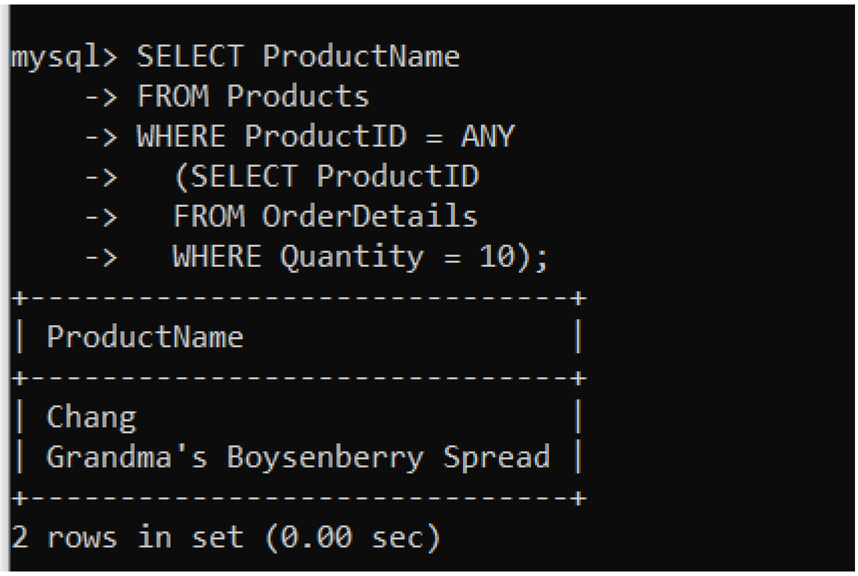
```
mysql> select * from OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |   10248 |        11 |       12 |
|             2 |   10248 |        42 |       10 |
|             3 |   10248 |        72 |        5 |
|             4 |   10249 |        14 |        9 |
|             5 |   10249 |        51 |       40 |
|             6 |   10250 |        41 |       10 |
|             7 |   10250 |        51 |       35 |
|             8 |   10250 |        65 |       15 |
|             9 |   10251 |        22 |        6 |
|            10 |   10251 |        57 |       15 |
+---------------+---------+-----------+----------+
10 rows in set (0.01 sec)
```

Where attribute is true



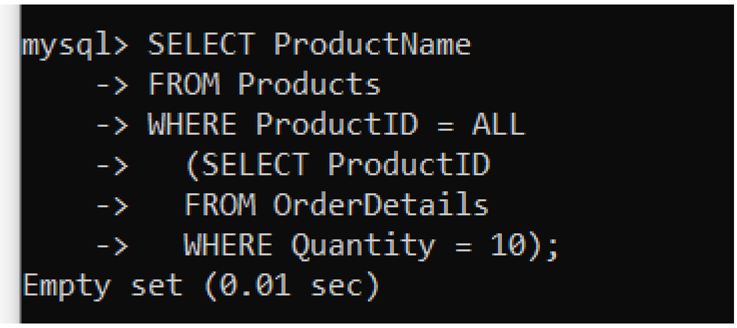
```
mysql> SELECT ALL ProductName
    -> FROM Products
    -> WHERE TRUE;
+----------------------------------+
| ProductName                      |
+----------------------------------+
| Chais                            |
| Chang                            |
| Aniseed Syrup                    |
| Chef Anton's Cajun Seasoning     |
| Chef Anton's Gumbo Mix           |
| Grandma's Boysenberry Spread     |
| Uncle Bob's Organic Dried Pears |
| Northwoods Cranberry Sauce       |
| Mishi Kobe Niku                  |
+----------------------------------+
9 rows in set (0.00 sec)
```

ANY



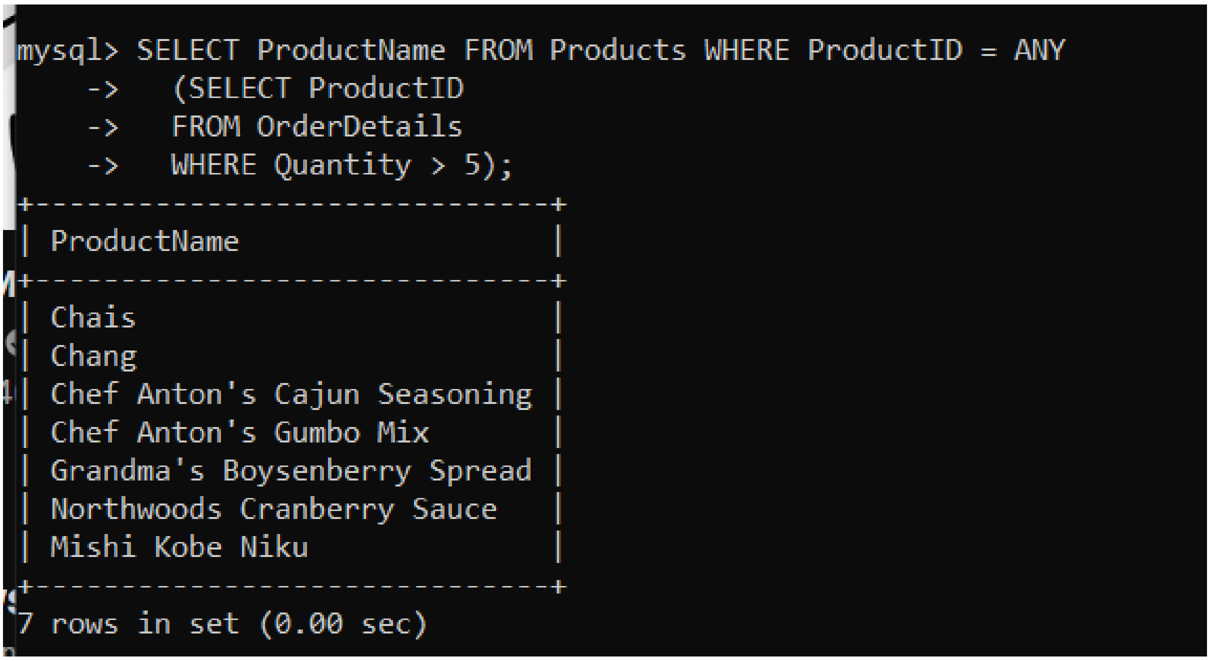
```
mysql> SELECT ProductName
    -> FROM Products
    -> WHERE ProductID = ANY
    ->   (SELECT ProductID
    ->   FROM OrderDetails
    ->   WHERE Quantity = 10);
+------------------------------+
| ProductName                  |
+------------------------------+
| Chang                        |
| Grandma's Boysenberry Spread |
+------------------------------+
2 rows in set (0.00 sec)
```

ALL



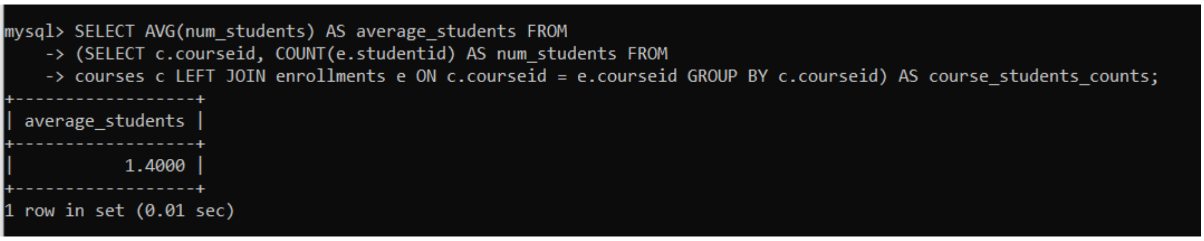
```
mysql> SELECT ProductName
    -> FROM Products
    -> WHERE ProductID = ALL
    ->   (SELECT ProductID
    ->   FROM OrderDetails
    ->   WHERE Quantity = 10);
Empty set (0.01 sec)
```

ANY

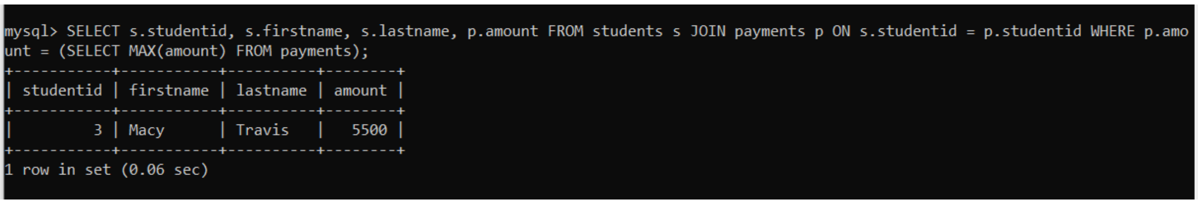


```
mysql> SELECT ProductName FROM Products WHERE ProductID = ANY
    ->   (SELECT ProductID
    ->   FROM OrderDetails
    ->   WHERE Quantity > 5);
+------------------------------+
| ProductName                  |
+------------------------------+
| Chais                        |
| Chang                        |
| Chef Anton's Cajun Seasoning |
| Chef Anton's Gumbo Mix       |
| Grandma's Boysenberry Spread |
| Northwoods Cranberry Sauce   |
| Mishi Kobe Niku              |
+------------------------------+
7 rows in set (0.00 sec)
```

SUBQUERY

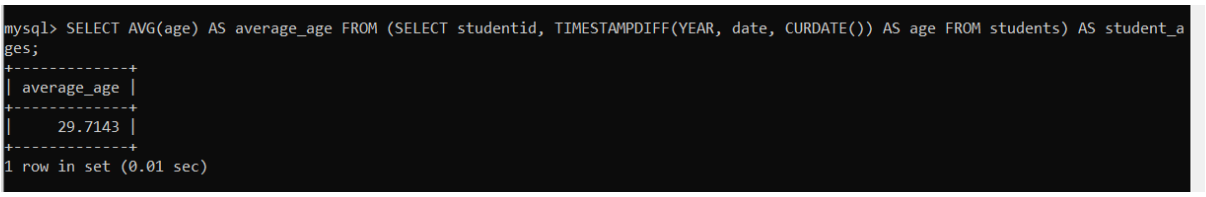


```
mysql> SELECT AVG(num_students) AS average_students FROM
    -> (SELECT c.courseid, COUNT(e.studentid) AS num_students FROM
    -> courses c LEFT JOIN enrollments e ON c.courseid = e.courseid GROUP BY c.courseid) AS course_students_counts;
+------------------+
| average_students |
+------------------+
|           1.4000 |
+------------------+
1 row in set (0.01 sec)
```



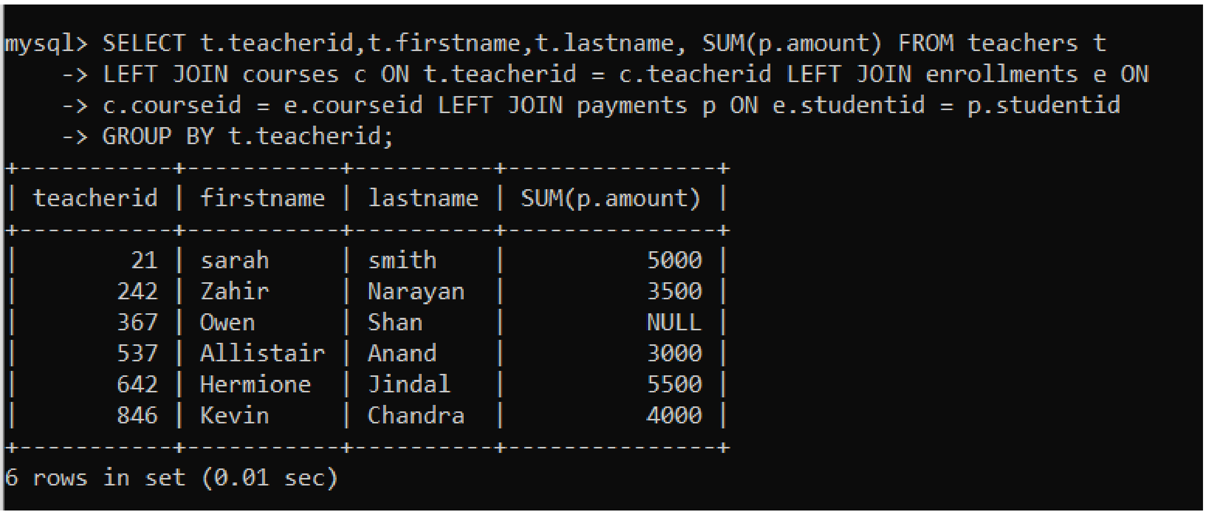
```
mysql> SELECT s.studentid, s.firstname, s.lastname, p.amount FROM students s JOIN payments p ON s.studentid = p.studentid WHERE p.amo
unt = (SELECT MAX(amount) FROM payments);
+-----------+-----------+----------+--------+
| studentid | firstname | lastname | amount |
+-----------+-----------+----------+--------+
|         3 | Macy      | Travis   |   5500 |
+-----------+-----------+----------+--------+
1 row in set (0.06 sec)
```

Current date calculated using date function



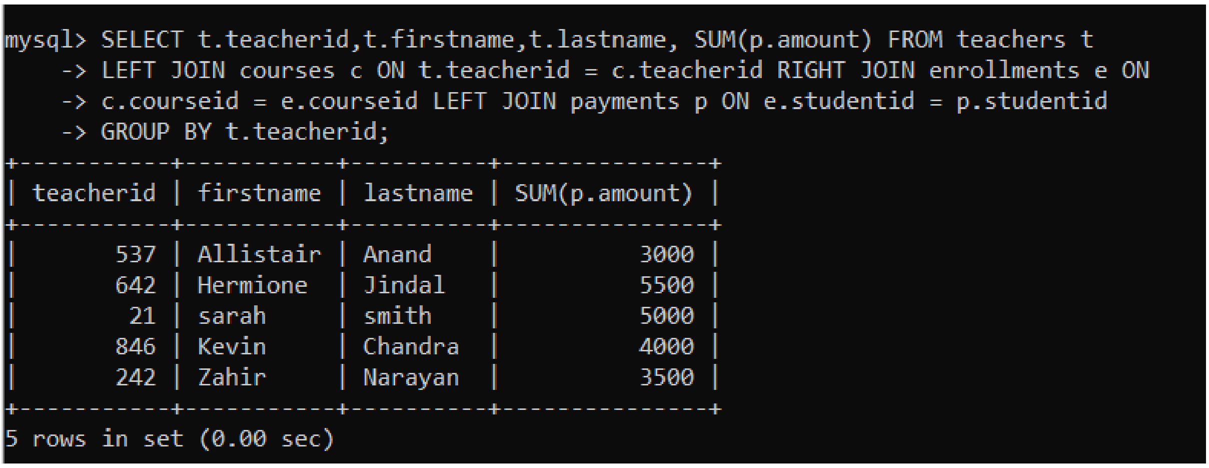
```
mysql> SELECT AVG(age) AS average_age FROM (SELECT studentid, TIMESTAMPDIFF(YEAR, date, CURDATE()) AS age FROM students) AS student_a
ges;
+-------------+
| average_age |
+-------------+
|     29.7143 |
+-------------+
1 row in set (0.01 sec)
```

Left join



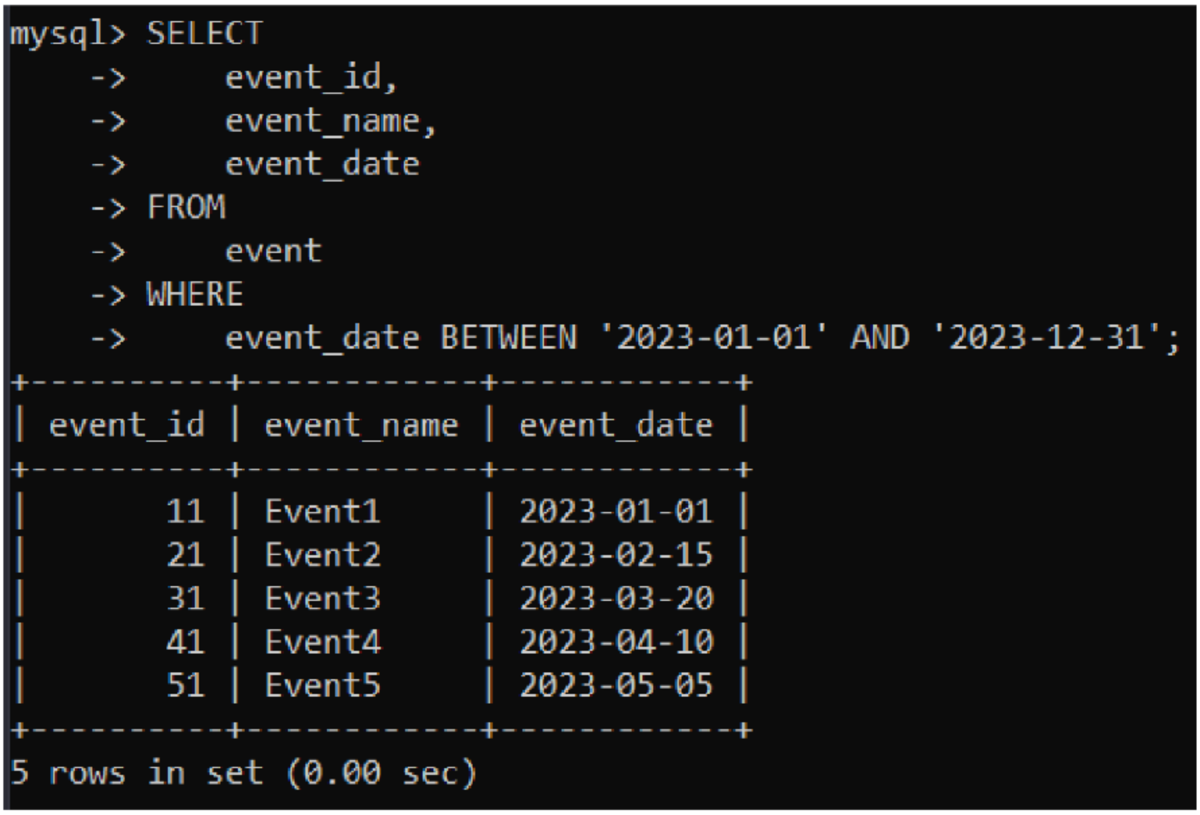
```
mysql> SELECT t.teacherid,t.firstname,t.lastname, SUM(p.amount) FROM teachers t
    -> LEFT JOIN courses c ON t.teacherid = c.teacherid LEFT JOIN enrollments e ON
    -> c.courseid = e.courseid LEFT JOIN payments p ON e.studentid = p.studentid
    -> GROUP BY t.teacherid;
+-----------+-----------+----------+---------------+
| teacherid | firstname | lastname | SUM(p.amount) |
+-----------+-----------+----------+---------------+
|        21 | sarah     | smith    |          5000 |
|       242 | Zahir     | Narayan  |          3500 |
|       367 | Owen      | Shan     |          NULL |
|       537 | Allistair | Anand    |          3000 |
|       642 | Hermione  | Jindal   |          5500 |
|       846 | Kevin     | Chandra  |          4000 |
+-----------+-----------+----------+---------------+
6 rows in set (0.01 sec)
```

Right join

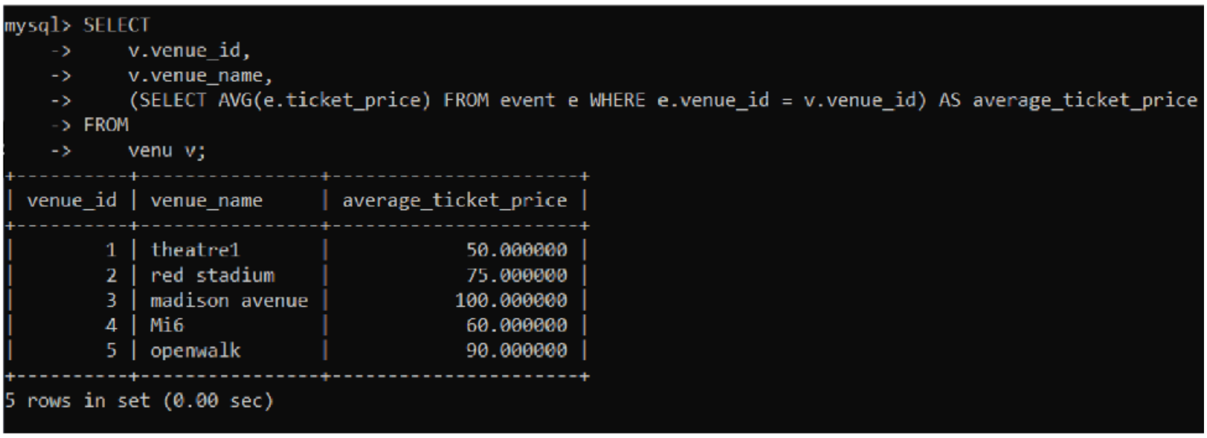


```
mysql> SELECT t.teacherid,t.firstname,t.lastname, SUM(p.amount) FROM teachers t
    -> LEFT JOIN courses c ON t.teacherid = c.teacherid RIGHT JOIN enrollments e ON
    -> c.courseid = e.courseid LEFT JOIN payments p ON e.studentid = p.studentid
    -> GROUP BY t.teacherid;
+-----------+-----------+----------+---------------+
| teacherid | firstname | lastname | SUM(p.amount) |
+-----------+-----------+----------+---------------+
|       537 | Allistair | Anand    |          3000 |
|       642 | Hermione  | Jindal   |          5500 |
|        21 | sarah     | smith    |          5000 |
|       846 | Kevin     | Chandra  |          4000 |
|       242 | Zahir     | Narayan  |          3500 |
+-----------+-----------+----------+---------------+
5 rows in set (0.00 sec)
```

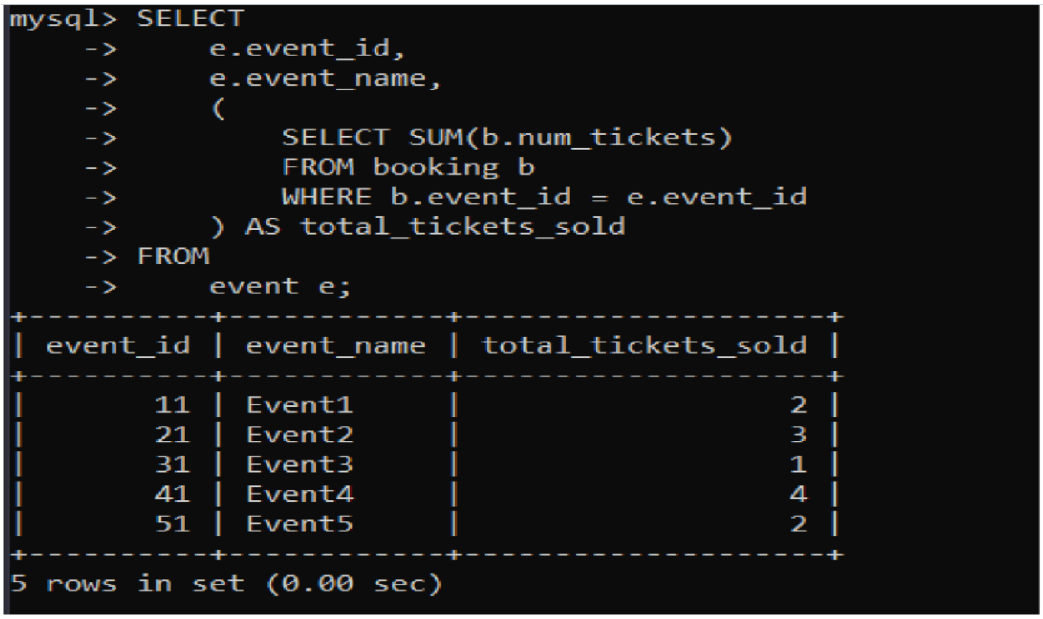
Date between two dates



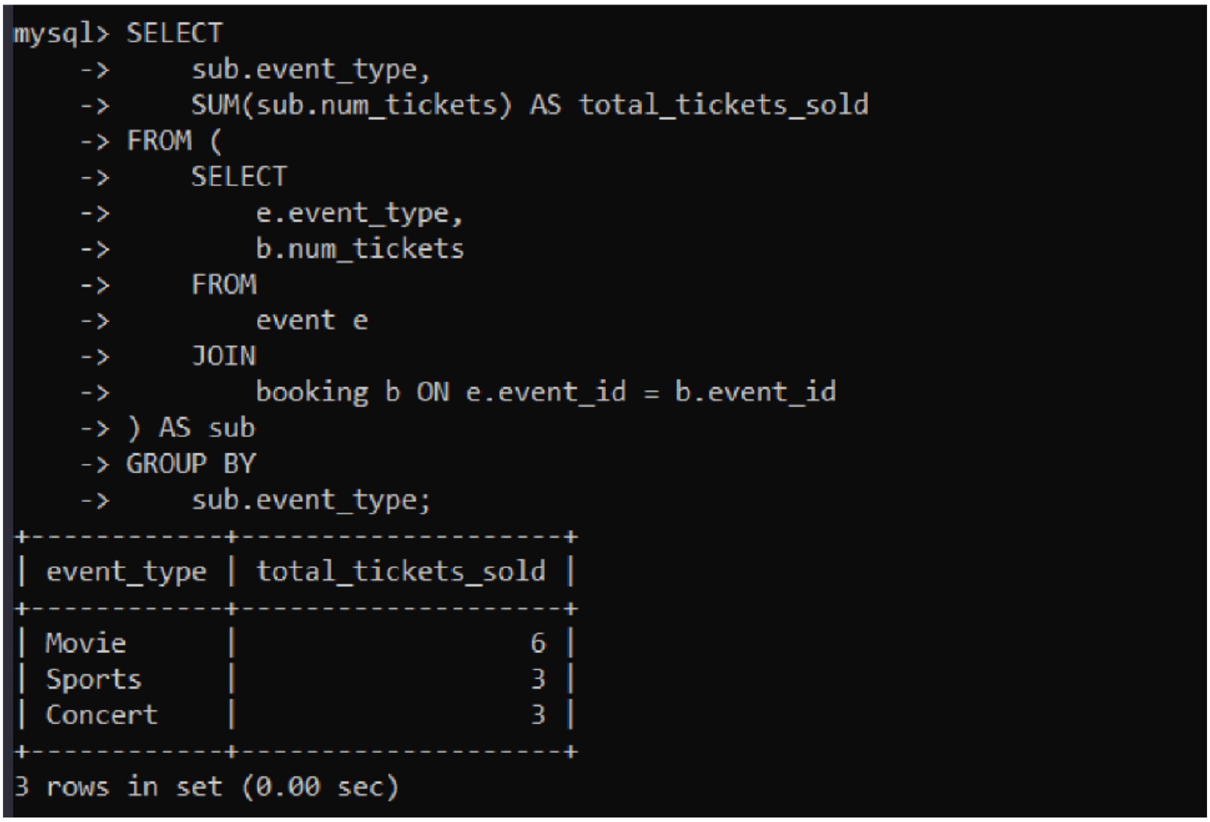
```
mysql> SELECT
    ->     event_id,
    ->     event_name,
    ->     event_date
    -> FROM
    ->     event
    -> WHERE
    ->     event_date BETWEEN '2023-01-01' AND '2023-12-31';
+----------+------------+------------+
| event_id | event_name | event_date |
+----------+------------+------------+
|       11 | Event1     | 2023-01-01 |
|       21 | Event2     | 2023-02-15 |
|       31 | Event3     | 2023-03-20 |
|       41 | Event4     | 2023-04-10 |
|       51 | Event5     | 2023-05-05 |
+----------+------------+------------+
5 rows in set (0.00 sec)
```



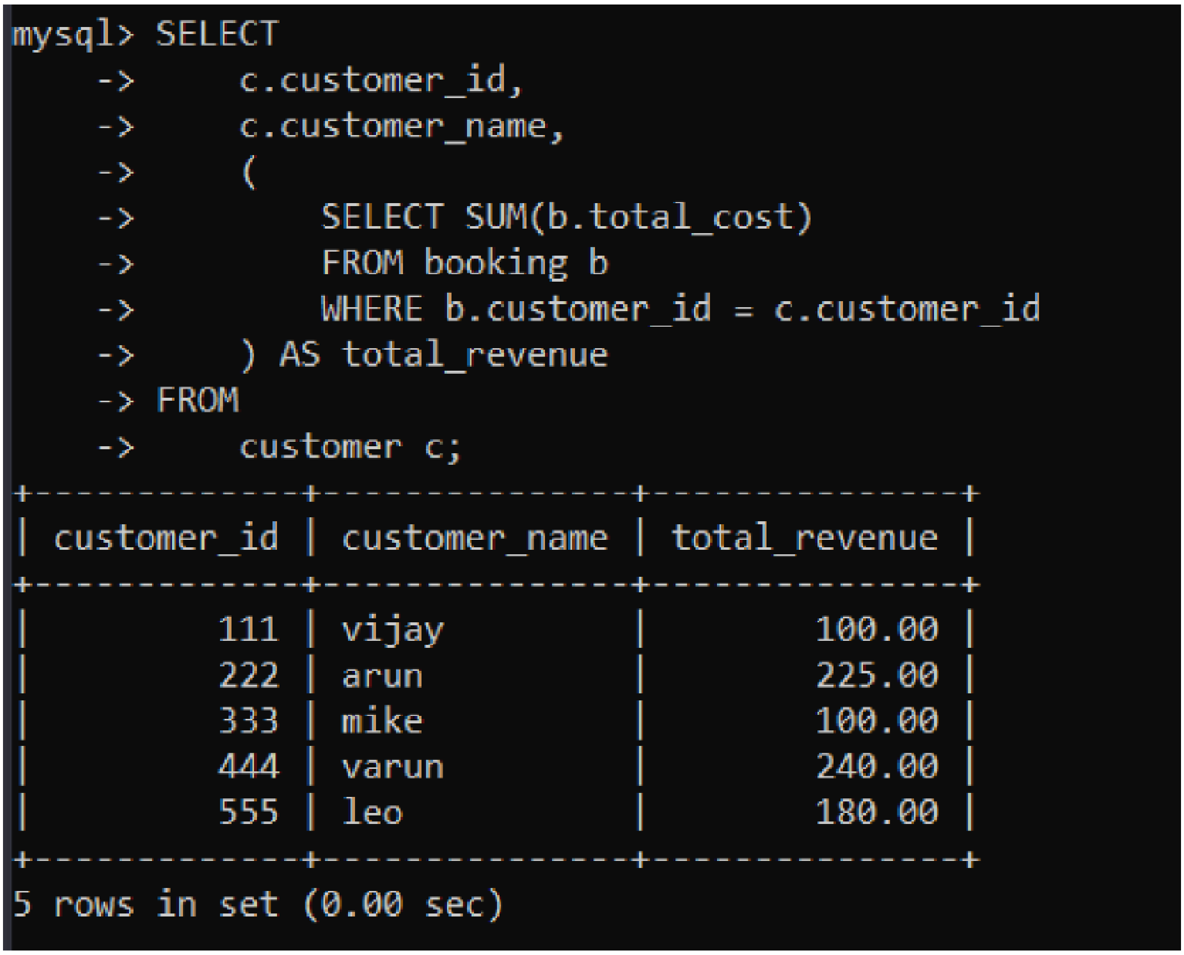
```
mysql> SELECT
    ->     v.venue_id,
    ->     v.venue_name,
    ->     (SELECT AVG(e.ticket_price) FROM event e WHERE e.venue_id = v.venue_id) AS average_ticket_price
    -> FROM
    ->     venu v;
+----------+----------------+----------------------+
| venue_id | venue_name     | average_ticket_price |
+----------+----------------+----------------------+
|        1 | theatre1       |            50.000000 |
|        2 | red stadium    |            75.000000 |
|        3 | madison avenue |           100.000000 |
|        4 | Mi6            |            60.000000 |
|        5 | openwalk       |            90.000000 |
+----------+----------------+----------------------+
5 rows in set (0.00 sec)
```



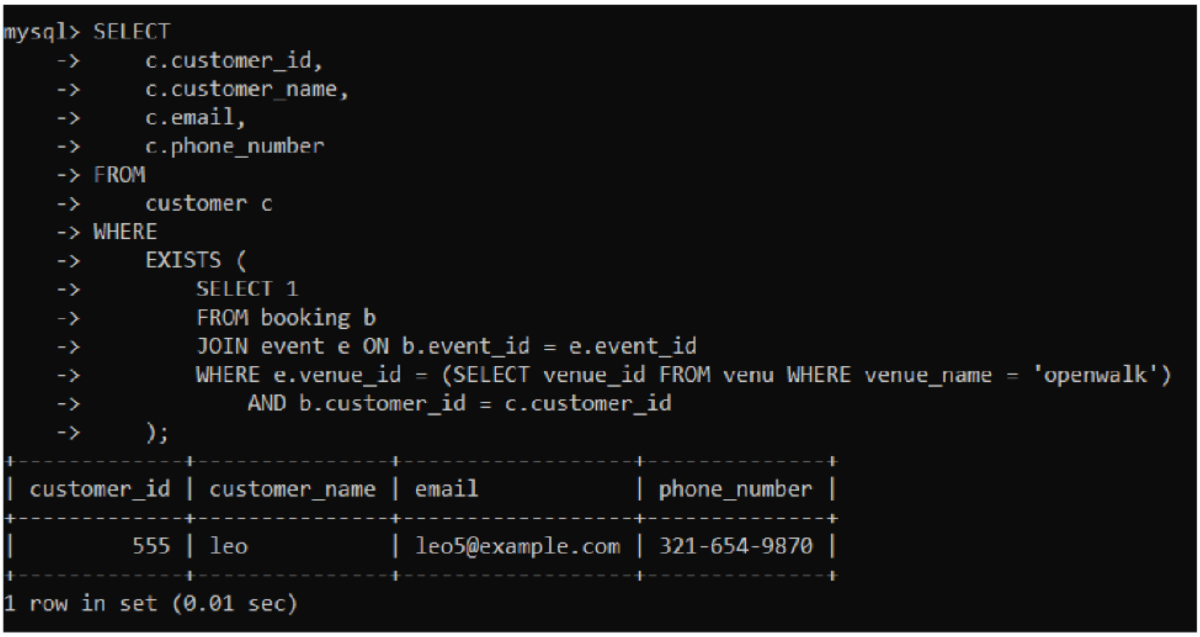
```
mysql> SELECT
    ->     e.event_id,
    ->     e.event_name,
    ->     (
    ->         SELECT SUM(b.num_tickets)
    ->         FROM booking b
    ->         WHERE b.event_id = e.event_id
    ->     ) AS total_tickets_sold
    -> FROM
    ->     event e;
+----------+------------+--------------------+
| event_id | event_name | total_tickets_sold |
+----------+------------+--------------------+
|       11 | Event1     |                  2 |
|       21 | Event2     |                  3 |
|       31 | Event3     |                  1 |
|       41 | Event4     |                  4 |
|       51 | Event5     |                  2 |
+----------+------------+--------------------+
5 rows in set (0.00 sec)
```



```
mysql> SELECT
    ->     sub.event_type,
    ->     SUM(sub.num_tickets) AS total_tickets_sold
    -> FROM (
    ->     SELECT
    ->         e.event_type,
    ->         b.num_tickets
    ->     FROM
    ->         event e
    ->     JOIN
    ->         booking b ON e.event_id = b.event_id
    -> ) AS sub
    -> GROUP BY
    ->     sub.event_type;
+------------+--------------------+
| event_type | total_tickets_sold |
+------------+--------------------+
| Movie      |                  6 |
| Sports     |                  3 |
| Concert    |                  3 |
+------------+--------------------+
3 rows in set (0.00 sec)
```



```
mysql> 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 |
+-------------+---------------+---------------+
|         111 | vijay         |        100.00 |
|         222 | arun          |        225.00 |
|         333 | mike          |        100.00 |
|         444 | varun         |        240.00 |
|         555 | leo           |        180.00 |
+-------------+---------------+---------------+
5 rows in set (0.00 sec)
```



```
mysql> SELECT
    ->     c.customer_id,
    ->     c.customer_name,
    ->     c.email,
    ->     c.phone_number
    -> FROM
    ->     customer c
    -> WHERE
    ->     EXISTS (
    ->         SELECT 1
    ->         FROM booking b
    ->         JOIN event e ON b.event_id = e.event_id
    ->         WHERE e.venue_id = (SELECT venue_id FROM venu WHERE venue_name = 'openwalk')
    ->             AND b.customer_id = c.customer_id
    ->     );
+-------------+---------------+-------------------+--------------+
| customer_id | customer_name | email             | phone_number |
+-------------+---------------+-------------------+--------------+
|         555 | leo           | leo5@example.com  | 321-654-9870 |
+-------------+---------------+-------------------+--------------+
1 row in set (0.01 sec)
```