Online Shopping Platform

-By Akshay Gujare
77418analyst@gmail.com

ONLINE SHOPPING PLATFORM

SQL Assignment

We are analyzing a small online shopping platform where customers register and place orders. The company wants to track customer details and their purchases for reporting and analysis.

- Database Name: **RetailDB_1**
- Purpose: To understand who the customers are, when they joined, and what orders they placed.
- At this stage, we are only storing customer information and their orders.

Tables in RetailDB:

1. **Customers** – stores details of each registered customer.

   (**customer_id**, name, email, city, signup_date)

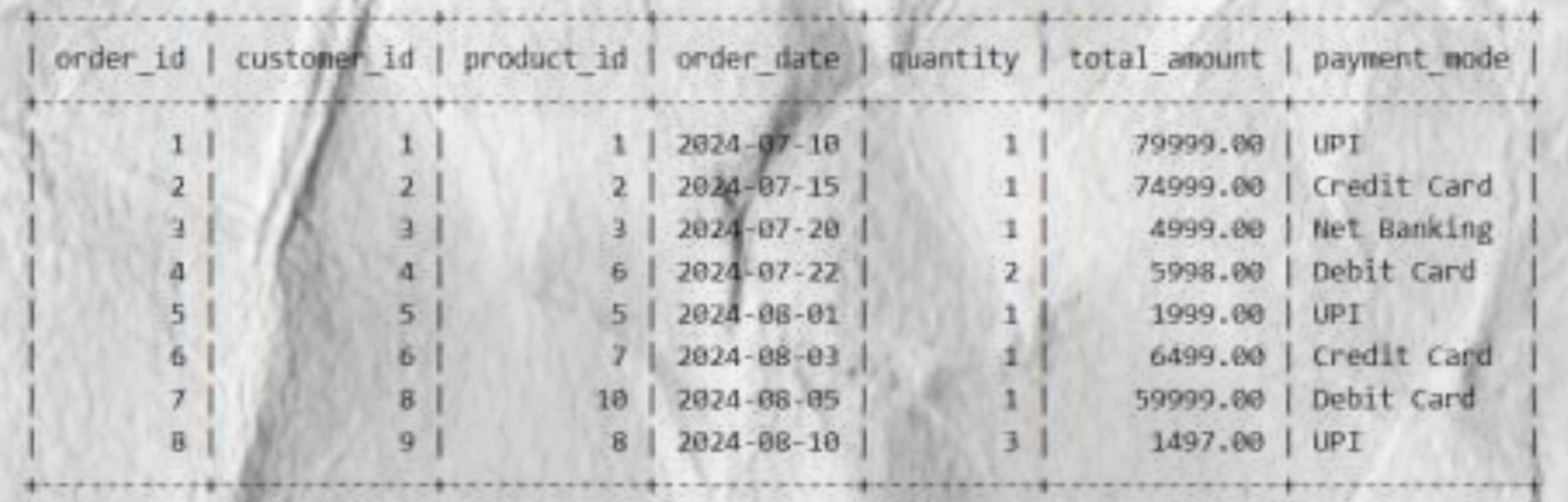
2. **Products** - holds all the products details

   (**product_id**, product_name, category, price)

3. **Orders** – keeps records of every order placed, linked to the customer.

   (**order_id**, customer_id, product_id, order_date, quantity, total_amount, payment_mode)

Orders



```
+----------+-------------+------------+------------+----------+--------------+--------------+
| order_id | customer_id | product_id | order_date | quantity | total_amount | payment_mode |
+----------+-------------+------------+------------+----------+--------------+--------------+
|        1 |           1 |          1 | 2024-07-10 |        1 |     79999.00 | UPI          |
|        2 |           2 |          2 | 2024-07-15 |        1 |     74999.00 | Credit Card  |
|        3 |           3 |          3 | 2024-07-20 |        1 |      4999.00 | Net Banking  |
|        4 |           4 |          6 | 2024-07-22 |        2 |      5998.00 | Debit Card   |
|        5 |           5 |          5 | 2024-08-01 |        1 |      1999.00 | UPI          |
|        6 |           6 |          7 | 2024-08-03 |        1 |      6499.00 | Credit Card  |
|        7 |           8 |         10 | 2024-08-05 |        1 |     59999.00 | Debit Card   |
|        8 |           9 |          8 | 2024-08-10 |        3 |      1497.00 | UPI          |
+----------+-------------+------------+------------+----------+--------------+--------------+
```

Customers

```
+-------------+--------------+-------------------------+-----------+-------------+
| customer_id | name         | email                   | city      | signup_date |
+-------------+--------------+-------------------------+-----------+-------------+
|           1 | Rohit Kumar  | rohit.kumar@gmail.com   | Delhi     | 2024-01-15  |
|           2 | Sneha Sharma | sneha.sharma@yahoo.com  | Mumbai    | 2024-02-20  |
|           3 | Amit Patel   | amit.patel@gmail.com    | Ahmedabad | 2024-03-05  |
|           4 | Priya Reddy  | priya.reddy@gmail.com   | Hyderabad | 2024-03-22  |
|           5 | Karan Singh  | karan.singh@outlook.com | Chennai   | 2024-04-10  |
|           6 | Neha Verma   | neha.verma@gmail.com    | Pune      | 2024-05-08  |
|           7 | Arjun Mehta  | arjun.mehta@gmail.com   | Jaipur    | 2024-05-15  |
|           8 | Ananya Iyer  | ananya.iyer@gmail.com   | Bengaluru | 2024-06-02  |
|           9 | Vikram Das   | vikram.das@gmail.com    | Kolkata   | 2024-06-20  |
|          10 | Meera Nair   | meera.nair@gmail.com    | Kochi     | 2024-07-01  |
+-------------+--------------+-------------------------+-----------+-------------+
```

Products

```
+------------+------------------------+-----------------+----------+
| product_id | product_name           | category        | price    |
+------------+------------------------+-----------------+----------+
|          1 | iPhone 15              | Electronics     | 79999.00 |
|          2 | Samsung Galaxy S24     | Electronics     | 74999.00 |
|          3 | Noise Smartwatch       | Wearables       |  4999.00 |
|          4 | Boat Earbuds           | Wearables       |  2499.00 |
|          5 | Kurta Set              | Fashion         |  1999.00 |
|          6 | Running Shoes          | Fashion         |  2999.00 |
|          7 | Prestige Mixer Grinder | Home Appliances |  6499.00 |
|          8 | Tata Tea 1kg Pack      | Groceries       |   499.00 |
|          9 | Amul Butter 500g       | Groceries       |   285.00 |
|         10 | Sony Bravia 55” TV     | Electronics     | 59999.00 |
+------------+------------------------+-----------------+----------+
```

Task: Solve the below mentioned questions by writing SQL queries

1. Fetch all customers from the database.

2. Show only the customer names and their cities.

3. Find customers who live in Mumbai.

4. Get all orders placed after 1st August 2024.

5. List all products priced greater than ₹5000.

6. Count how many customers exist in the system.

7. Update a customer’s city (e.g., change Rohit Kumar’s city to Hyderabad).

8. Delete an order (e.g., remove order with ID = 5).

9. Display product names with their original price and price increased by 10%.

10.Show only the unique cities where customers live.

11. Get the first 3 customers who signed up.

12.Skip the first 2 customers and fetch the next 3 customers.

13.Find products with prices between ₹2000 and ₹6000.

14.Find customers who are from Mumbai OR Chennai.

15.Find customers who are NOT from Delhi.

16.Find orders that are NOT paid by UPI.

17.Get the average order amount across all orders.

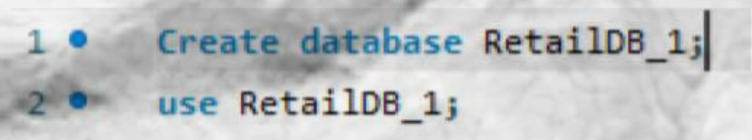
18.Show the highest order amount.

19.Show the lowest product price.

20.Find the total money spent across all orders

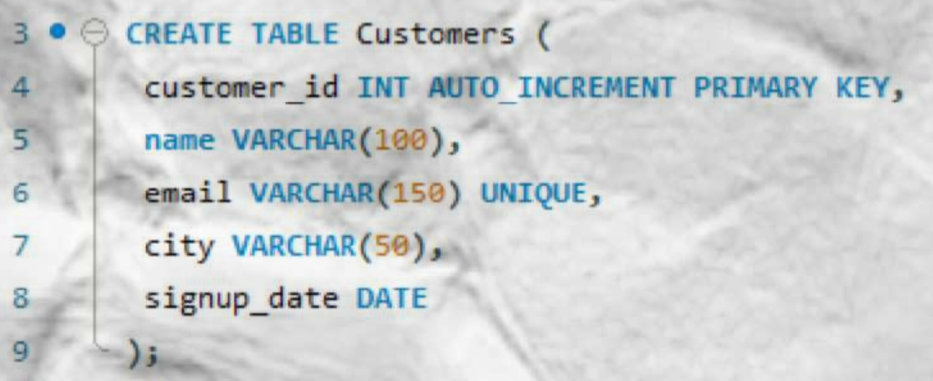
Table creation and Import of data from csv file

Step 1: Database creation



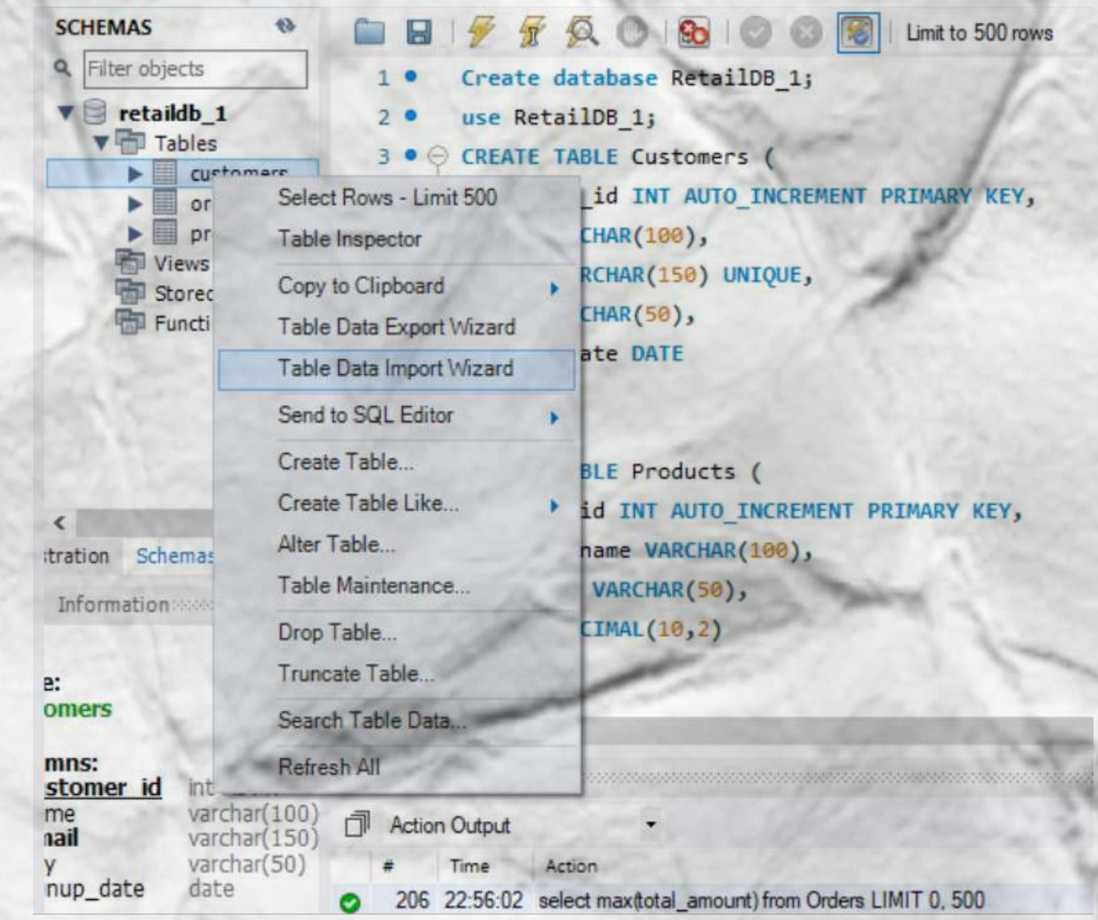
```
Create database RetailDB_1;
use RetailDB_1;
```

Step 2: Table creation (Customer)



```
CREATE TABLE Customers (
  customer_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(150) UNIQUE,
  city VARCHAR(50),
  signup_date DATE
);
```

Step 3: Import data from CSV file

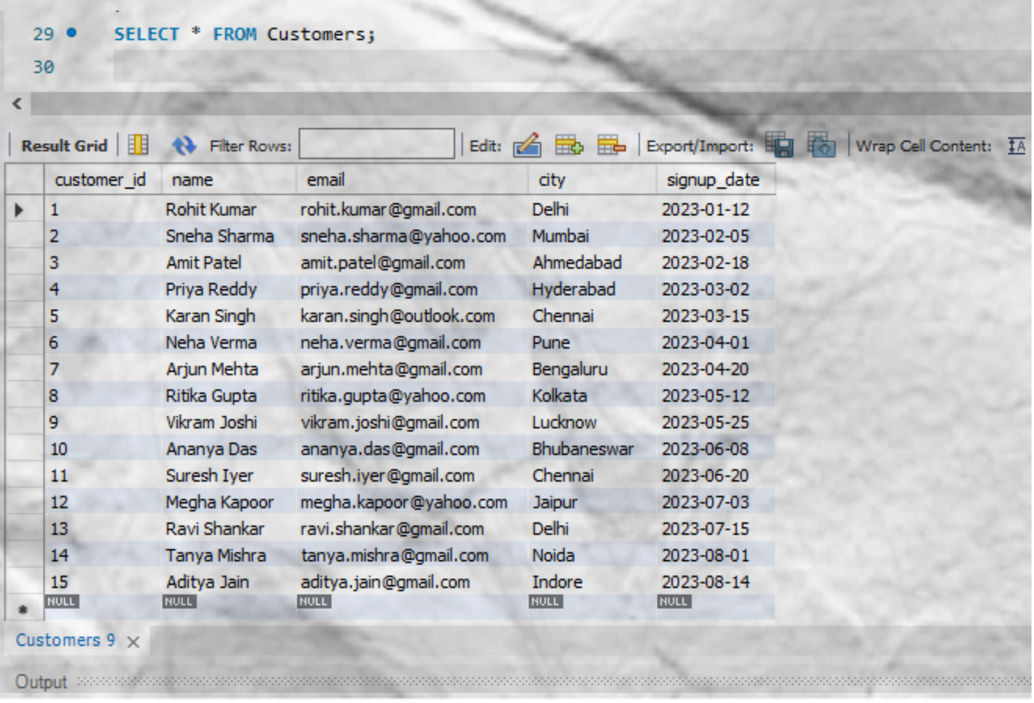


```
Create database RetailDB_1;
use RetailDB_1;
CREATE TABLE Customers (
_id INT AUTO_INCREMENT PRIMARY KEY,
CHAR(100),
RCHAR(150) UNIQUE,
CHAR(50),
ate DATE

BLE Products (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100),
 VARCHAR(50),
CIMAL(10,2)
```

```
e:
omers

mns:
stomer_id   int
me          varchar(100)
nail        varchar(150)
y           varchar(50)
nup_date    date
```

```
#    Time      Action
206  22:56:02  select max(total_amount) from Orders LIMIT 0, 500
```

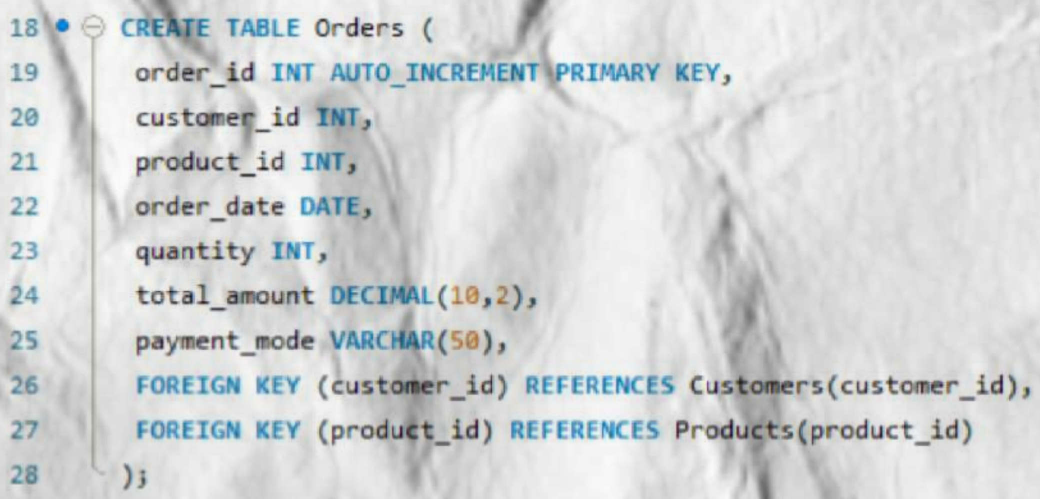
Step 4: Display Imported data



```
SELECT * FROM Customers;
```

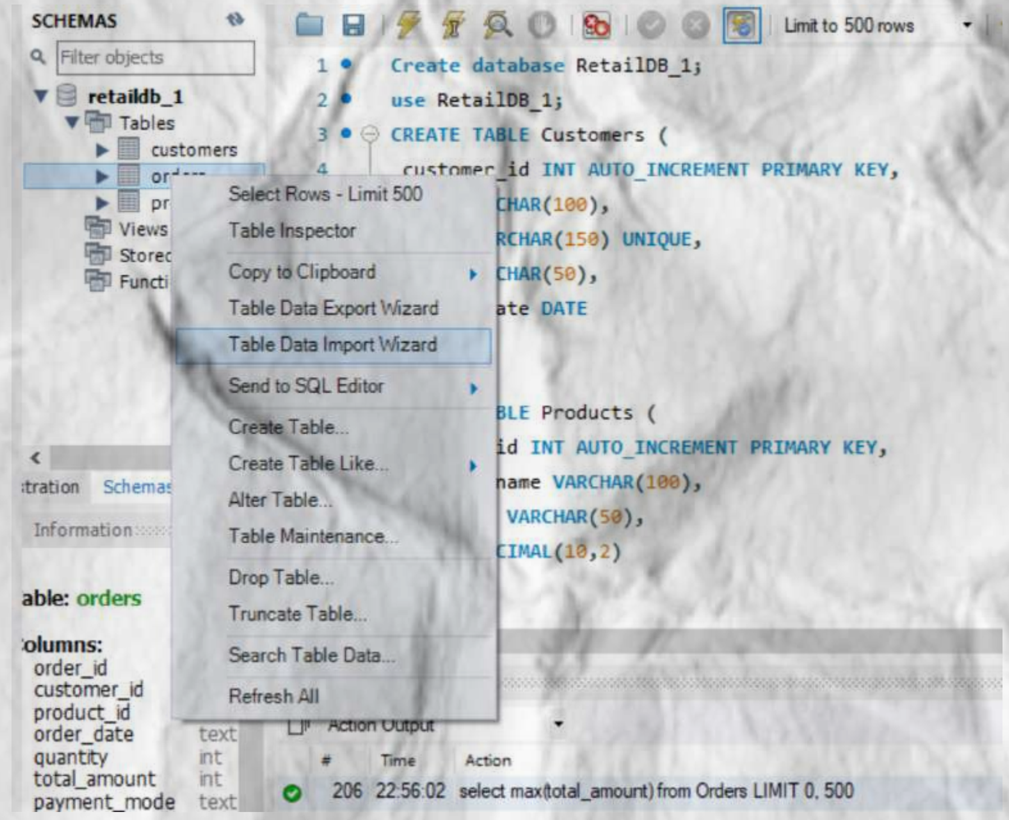
customer_id	name	email	city	signup_date
1	Rohit Kumar	rohit.kumar@gmail.com	Delhi	2023-01-12
2	Sneha Sharma	sneha.sharma@yahoo.com	Mumbai	2023-02-05
3	Amit Patel	amit.patel@gmail.com	Ahmedabad	2023-02-18
4	Priya Reddy	priya.reddy@gmail.com	Hyderabad	2023-03-02
5	Karan Singh	karan.singh@outlook.com	Chennai	2023-03-15
6	Neha Verma	neha.verma@gmail.com	Pune	2023-04-01
7	Arjun Mehta	arjun.mehta@gmail.com	Bengaluru	2023-04-20
8	Ritika Gupta	ritika.gupta@yahoo.com	Kolkata	2023-05-12
9	Vikram Joshi	vikram.joshi@gmail.com	Lucknow	2023-05-25
10	Ananya Das	ananya.das@gmail.com	Bhubaneswar	2023-06-08
11	Suresh Iyer	suresh.iyer@gmail.com	Chennai	2023-06-20
12	Megha Kapoor	megha.kapoor@yahoo.com	Jaipur	2023-07-03
13	Ravi Shankar	ravi.shankar@gmail.com	Delhi	2023-07-15
14	Tanya Mishra	tanya.mishra@gmail.com	Noida	2023-08-01
15	Aditya Jain	aditya.jain@gmail.com	Indore	2023-08-14
NULL	NULL	NULL	NULL	NULL

Step 5: Table creation (Orders)



```sql
CREATE TABLE Orders (
  order_id INT AUTO_INCREMENT PRIMARY KEY,
  customer_id INT,
  product_id INT,
  order_date DATE,
  quantity INT,
  total_amount DECIMAL(10,2),
  payment_mode VARCHAR(50),
  FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
  FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

Step 6: Import data from CSV file



```sql
Create database RetailDB_1;
use RetailDB_1;
CREATE TABLE Customers (
 customer_id INT AUTO_INCREMENT PRIMARY KEY,
                          CHAR(100),
                          RCHAR(150) UNIQUE,
                          CHAR(50),
                          ate DATE


                          BLE Products (
                          id INT AUTO_INCREMENT PRIMARY KEY,
                          name VARCHAR(100),
                           VARCHAR(50),
                          CIMAL(10,2)
```

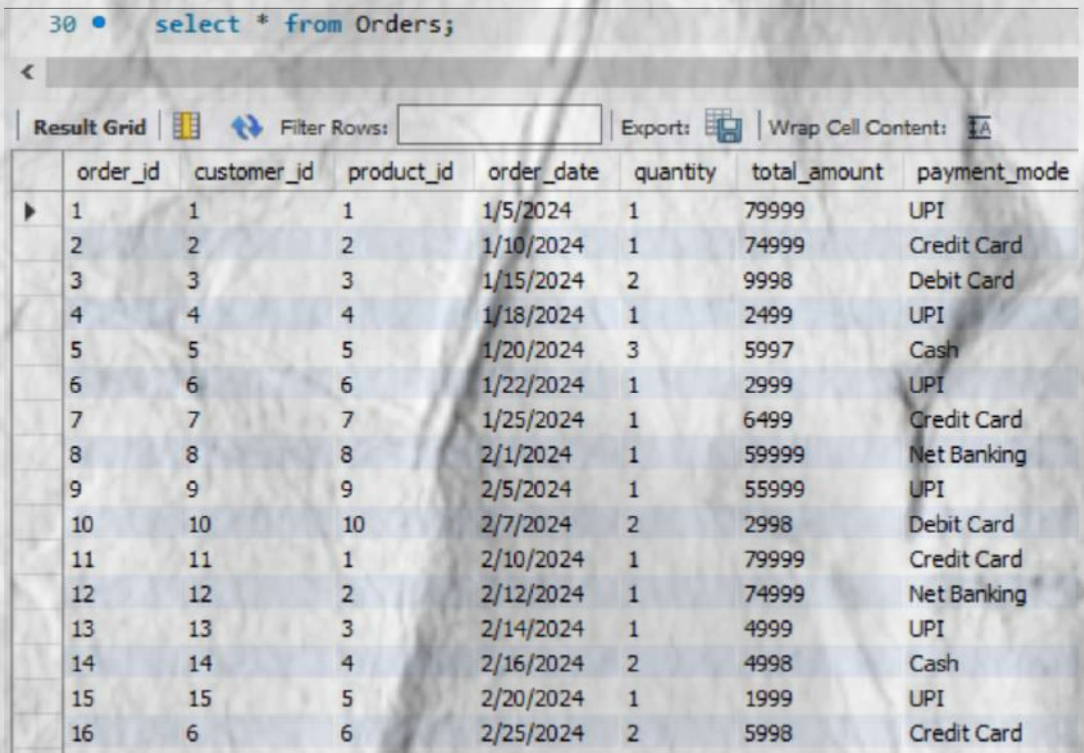
able: orders

olumns:

```
order_id
customer_id
product_id
order_date     text
quantity       int
total_amount   int
payment_mode   text
```

```
  #    Time      Action
  206  22:56:02  select max(total_amount) from Orders LIMIT 0, 500
```

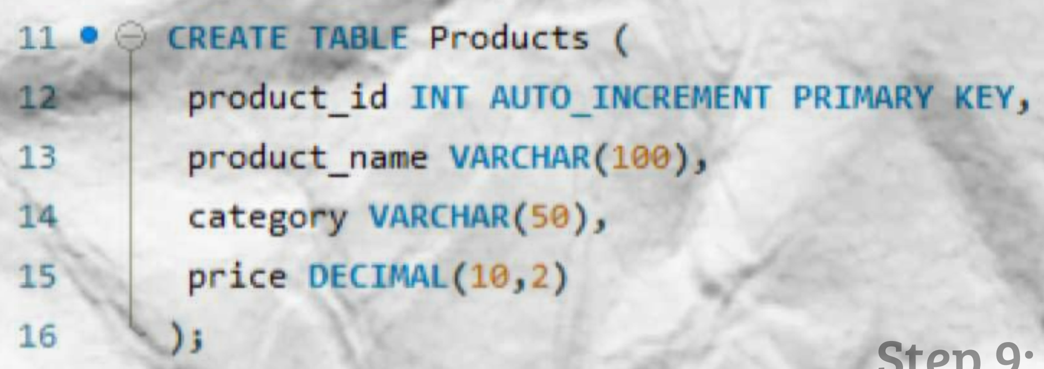
Step 7: Display Imported data



```sql
select * from Orders;
```

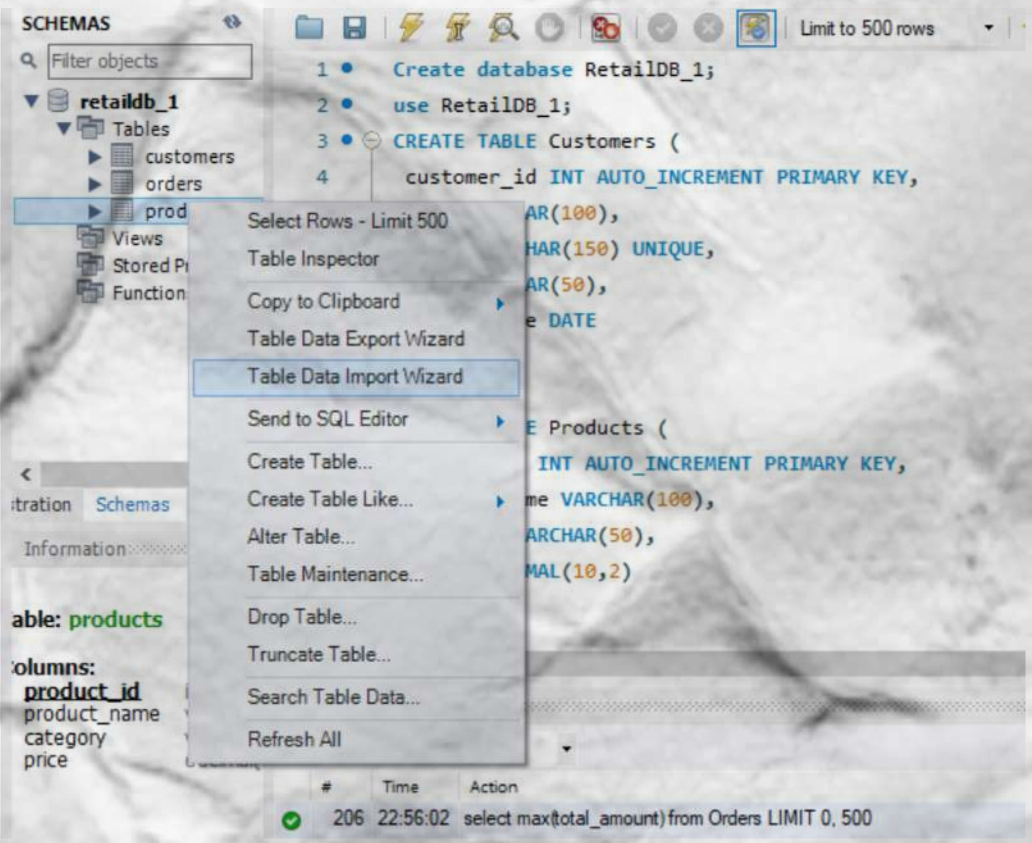
order_id	customer_id	product_id	order_date	quantity	total_amount	payment_mode
1	1	1	1/5/2024	1	79999	UPI
2	2	2	1/10/2024	1	74999	Credit Card
3	3	3	1/15/2024	2	9998	Debit Card
4	4	4	1/18/2024	1	2499	UPI
5	5	5	1/20/2024	3	5997	Cash
6	6	6	1/22/2024	1	2999	UPI
7	7	7	1/25/2024	1	6499	Credit Card
8	8	8	2/1/2024	1	59999	Net Banking
9	9	9	2/5/2024	1	55999	UPI
10	10	10	2/7/2024	2	2998	Debit Card
11	11	1	2/10/2024	1	79999	Credit Card
12	12	2	2/12/2024	1	74999	Net Banking
13	13	3	2/14/2024	1	4999	UPI
14	14	4	2/16/2024	2	4998	Cash
15	15	5	2/20/2024	1	1999	UPI
16	6	6	2/25/2024	2	5998	Credit Card

Step 8: Table creation (Products)



```
CREATE TABLE Products (
  product_id INT AUTO_INCREMENT PRIMARY KEY,
  product_name VARCHAR(100),
  category VARCHAR(50),
  price DECIMAL(10,2)
);
```

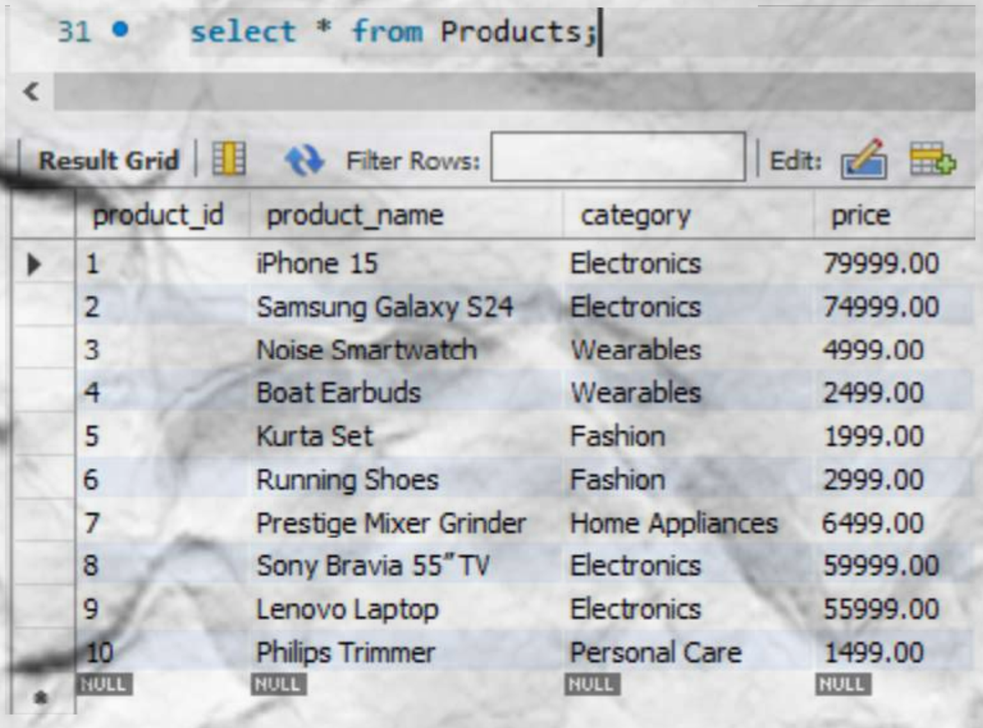
Step 9: Import data from CSV file



```
Create database RetailDB_1;
use RetailDB_1;
CREATE TABLE Customers (
  customer_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
#    Time      Action
206  22:56:02  select max(total_amount) from Orders LIMIT 0, 500
```

Step 10: Display Imported data



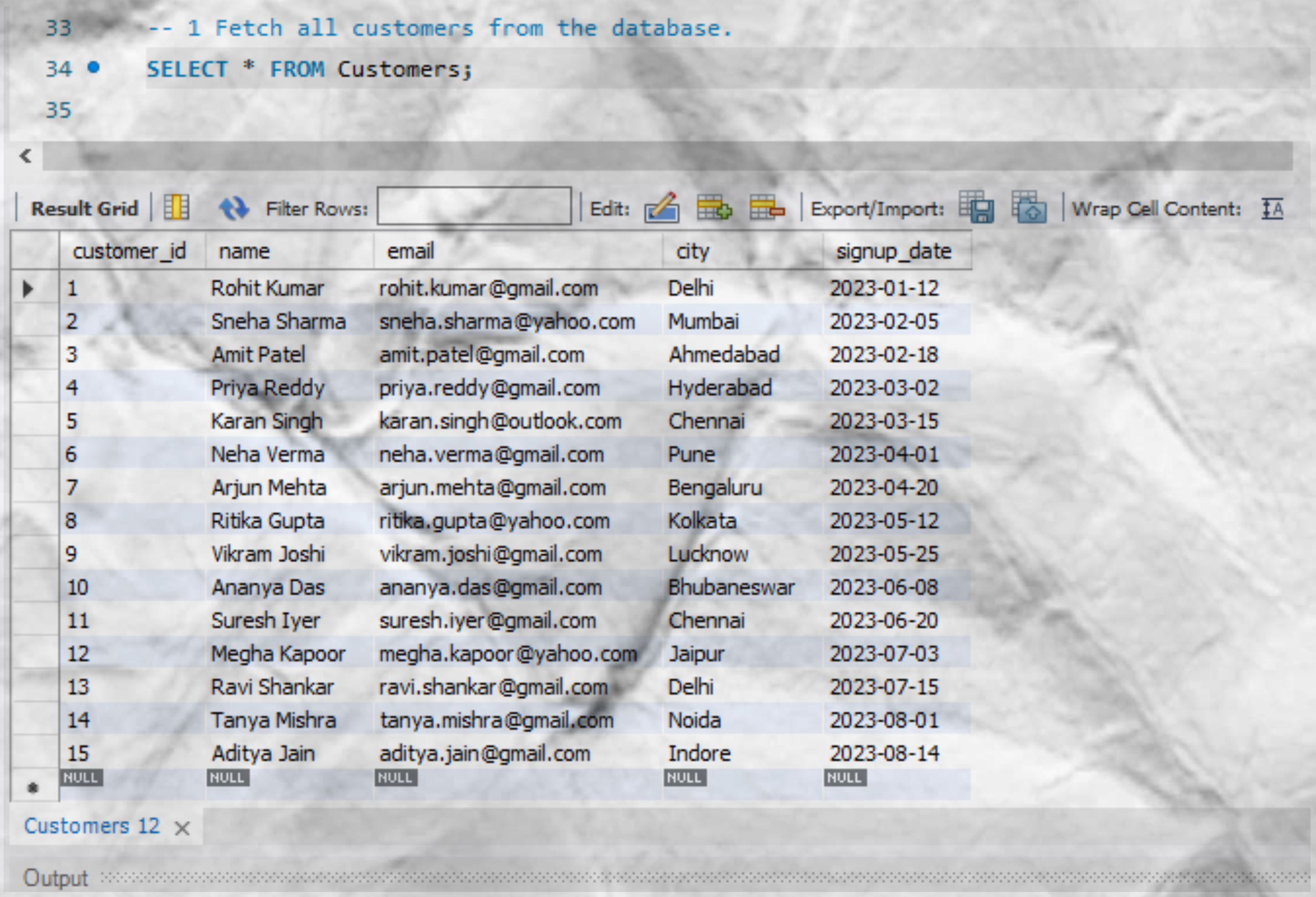
```
select * from Products;
```

product_id	product_name	category	price
1	iPhone 15	Electronics	79999.00
2	Samsung Galaxy S24	Electronics	74999.00
3	Noise Smartwatch	Wearables	4999.00
4	Boat Earbuds	Wearables	2499.00
5	Kurta Set	Fashion	1999.00
6	Running Shoes	Fashion	2999.00
7	Prestige Mixer Grinder	Home Appliances	6499.00
8	Sony Bravia 55" TV	Electronics	59999.00
9	Lenovo Laptop	Electronics	55999.00
10	Philips Trimmer	Personal Care	1499.00
NULL	NULL	NULL	NULL

Finally all three tables of Customers ,Orders & Products with data is ready to perform given operations or task.

Solved problem statements

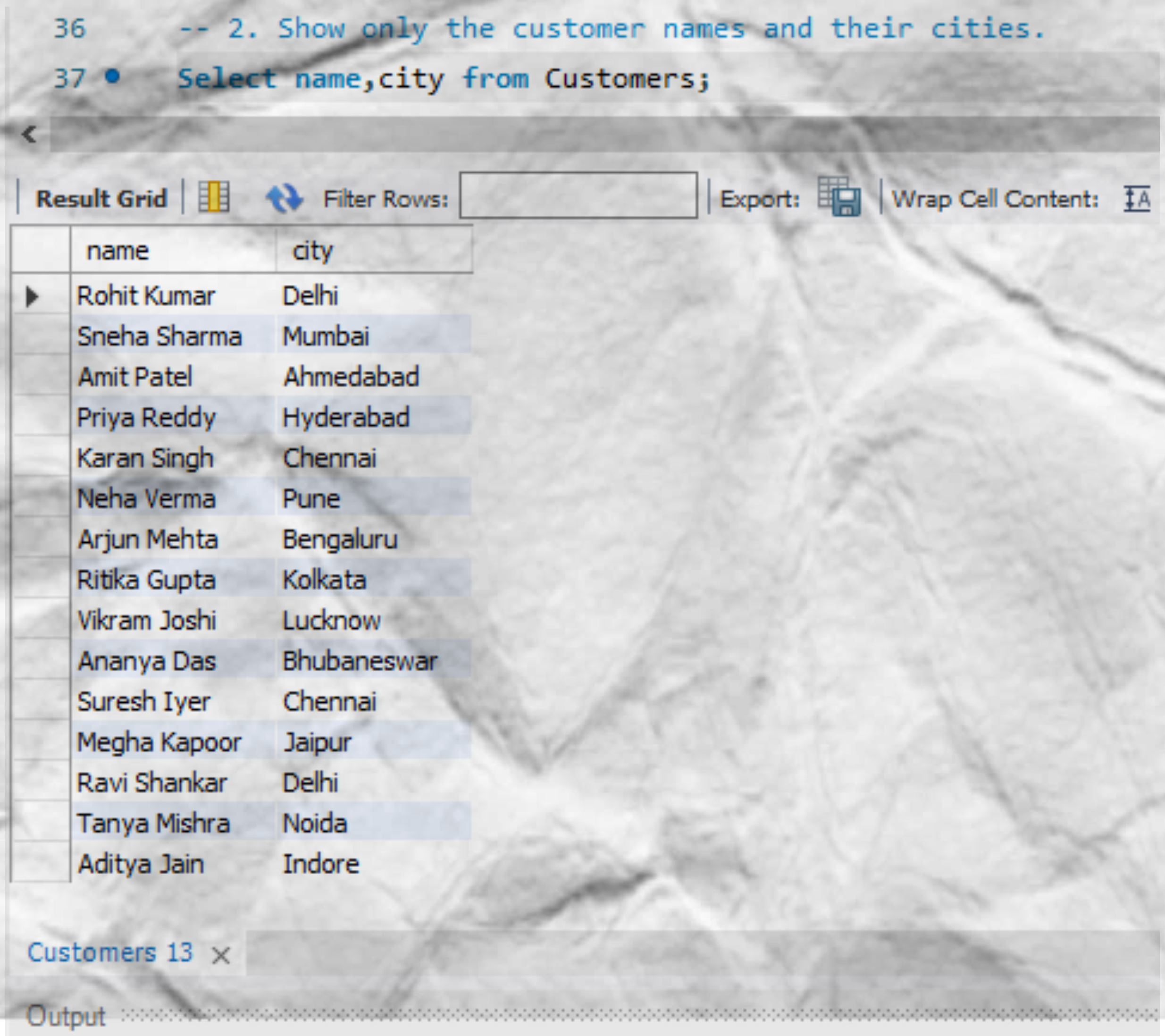
1. Fetch all customers from the database.



```sql
-- 1 Fetch all customers from the database.
SELECT * FROM Customers;
```

customer_id	name	email	city	signup_date
1	Rohit Kumar	rohit.kumar@gmail.com	Delhi	2023-01-12
2	Sneha Sharma	sneha.sharma@yahoo.com	Mumbai	2023-02-05
3	Amit Patel	amit.patel@gmail.com	Ahmedabad	2023-02-18
4	Priya Reddy	priya.reddy@gmail.com	Hyderabad	2023-03-02
5	Karan Singh	karan.singh@outlook.com	Chennai	2023-03-15
6	Neha Verma	neha.verma@gmail.com	Pune	2023-04-01
7	Arjun Mehta	arjun.mehta@gmail.com	Bengaluru	2023-04-20
8	Ritika Gupta	ritika.gupta@yahoo.com	Kolkata	2023-05-12
9	Vikram Joshi	vikram.joshi@gmail.com	Lucknow	2023-05-25
10	Ananya Das	ananya.das@gmail.com	Bhubaneswar	2023-06-08
11	Suresh Iyer	suresh.iyer@gmail.com	Chennai	2023-06-20
12	Megha Kapoor	megha.kapoor@yahoo.com	Jaipur	2023-07-03
13	Ravi Shankar	ravi.shankar@gmail.com	Delhi	2023-07-15
14	Tanya Mishra	tanya.mishra@gmail.com	Noida	2023-08-01
15	Aditya Jain	aditya.jain@gmail.com	Indore	2023-08-14
NULL	NULL	NULL	NULL	NULL

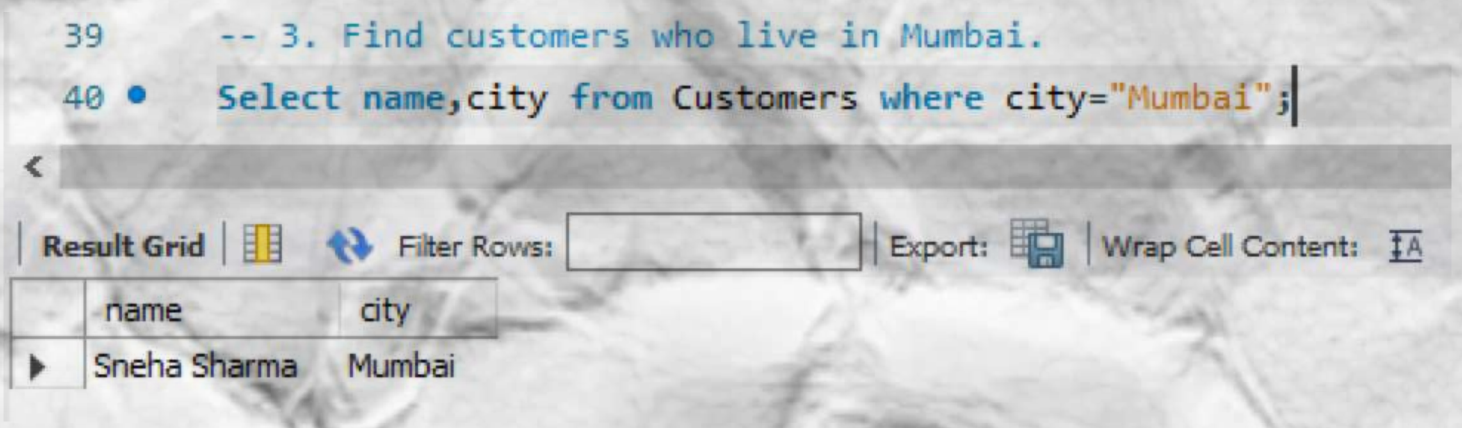
2. Show only the customer names and their cities.



```sql
-- 2. Show only the customer names and their cities.
Select name,city from Customers;
```

name	city
Rohit Kumar	Delhi
Sneha Sharma	Mumbai
Amit Patel	Ahmedabad
Priya Reddy	Hyderabad
Karan Singh	Chennai
Neha Verma	Pune
Arjun Mehta	Bengaluru
Ritika Gupta	Kolkata
Vikram Joshi	Lucknow
Ananya Das	Bhubaneswar
Suresh Iyer	Chennai
Megha Kapoor	Jaipur
Ravi Shankar	Delhi
Tanya Mishra	Noida
Aditya Jain	Indore

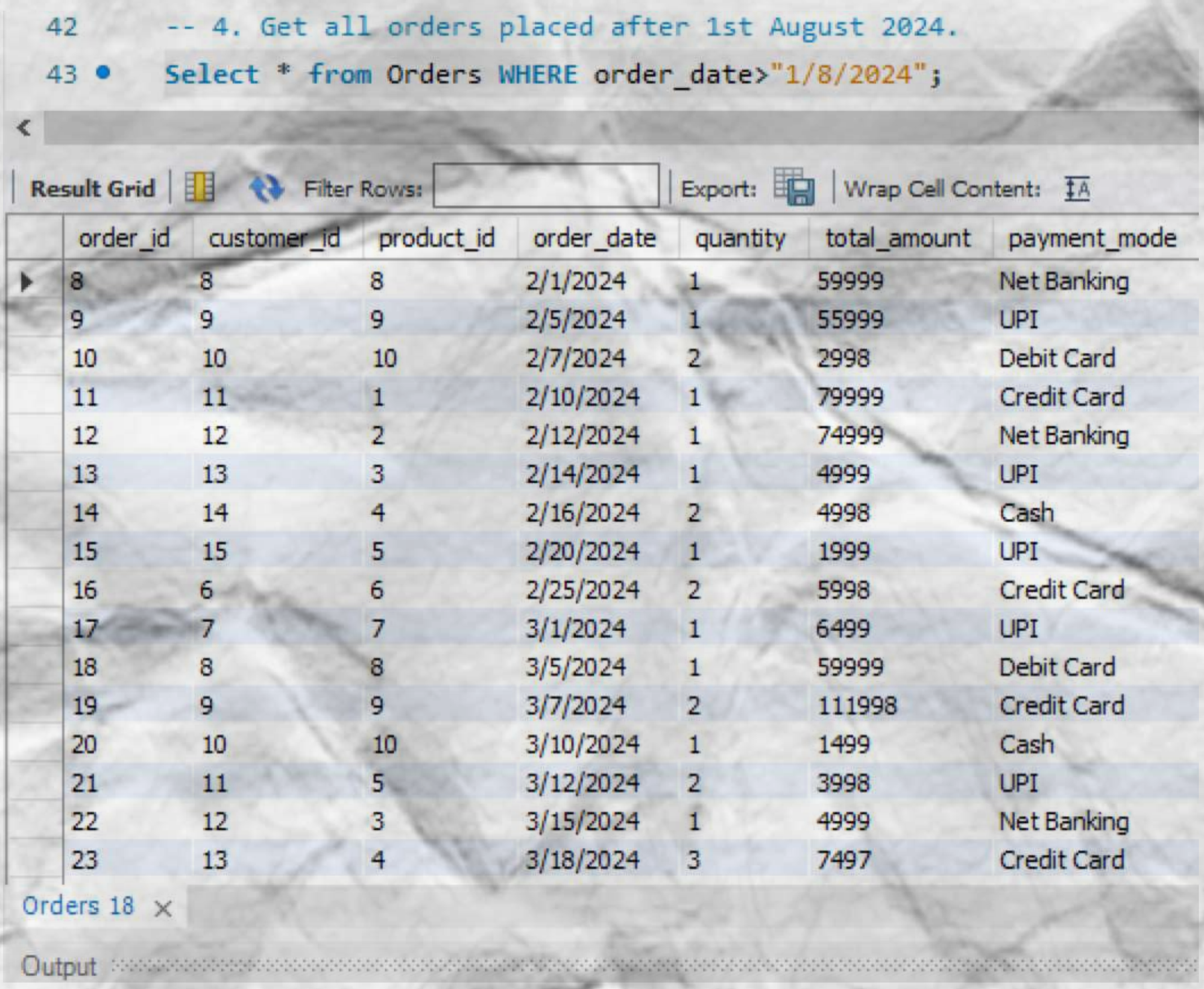
3. Find customers who live in Mumbai.



```sql
-- 3. Find customers who live in Mumbai.
Select name,city from Customers where city="Mumbai";
```

name	city
Sneha Sharma	Mumbai

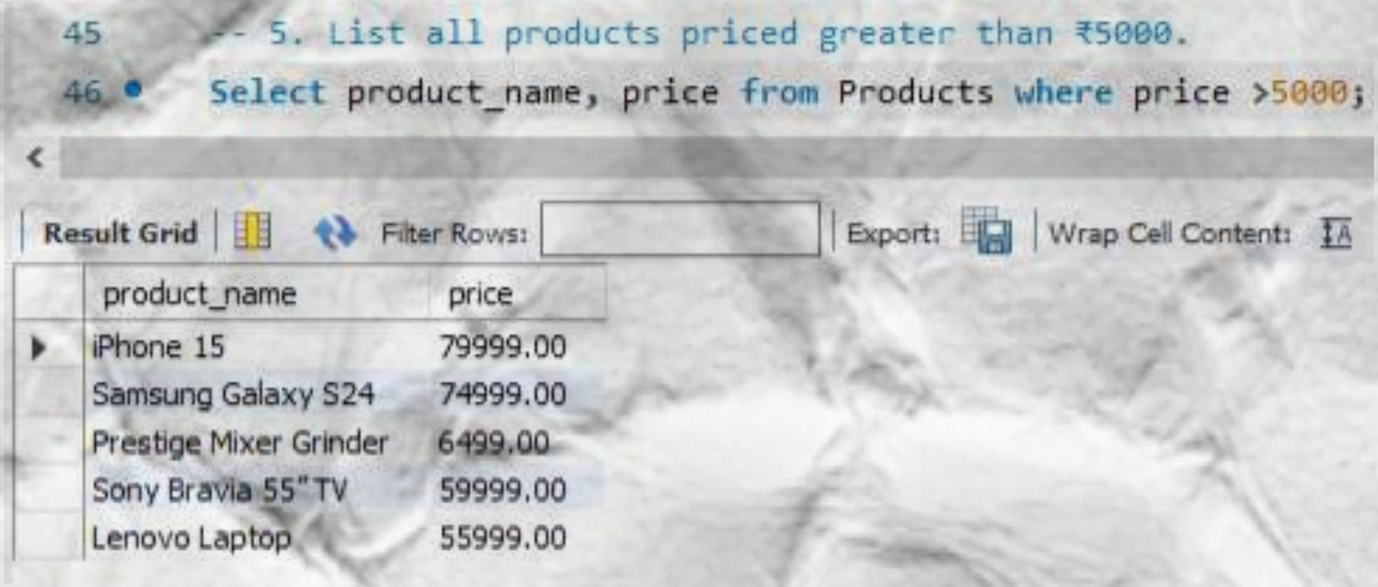
4. Get all orders placed after 1st August 2024.



```sql
-- 4. Get all orders placed after 1st August 2024.
Select * from Orders WHERE order_date>"1/8/2024";
```

order_id	customer_id	product_id	order_date	quantity	total_amount	payment_mode
8	8	8	2/1/2024	1	59999	Net Banking
9	9	9	2/5/2024	1	55999	UPI
10	10	10	2/7/2024	2	2998	Debit Card
11	11	1	2/10/2024	1	79999	Credit Card
12	12	2	2/12/2024	1	74999	Net Banking
13	13	3	2/14/2024	1	4999	UPI
14	14	4	2/16/2024	2	4998	Cash
15	15	5	2/20/2024	1	1999	UPI
16	6	6	2/25/2024	2	5998	Credit Card
17	7	7	3/1/2024	1	6499	UPI
18	8	8	3/5/2024	1	59999	Debit Card
19	9	9	3/7/2024	2	111998	Credit Card
20	10	10	3/10/2024	1	1499	Cash
21	11	5	3/12/2024	2	3998	UPI
22	12	3	3/15/2024	1	4999	Net Banking
23	13	4	3/18/2024	3	7497	Credit Card

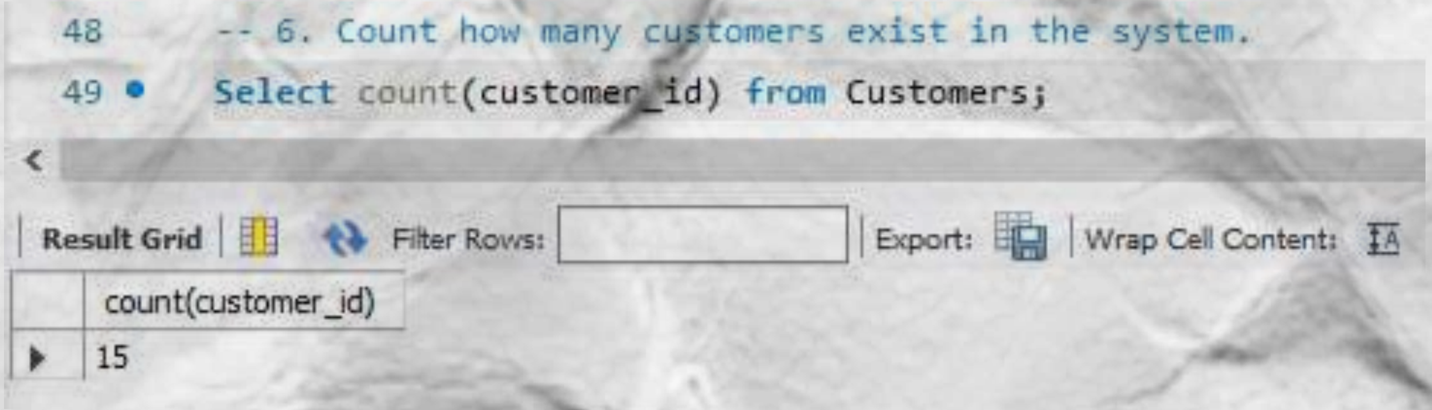
5. List all products priced greater than ₹5000.



```
-- 5. List all products priced greater than ₹5000.
Select product_name, price from Products where price >5000;
```

product_name	price
iPhone 15	79999.00
Samsung Galaxy S24	74999.00
Prestige Mixer Grinder	6499.00
Sony Bravia 55" TV	59999.00
Lenovo Laptop	55999.00

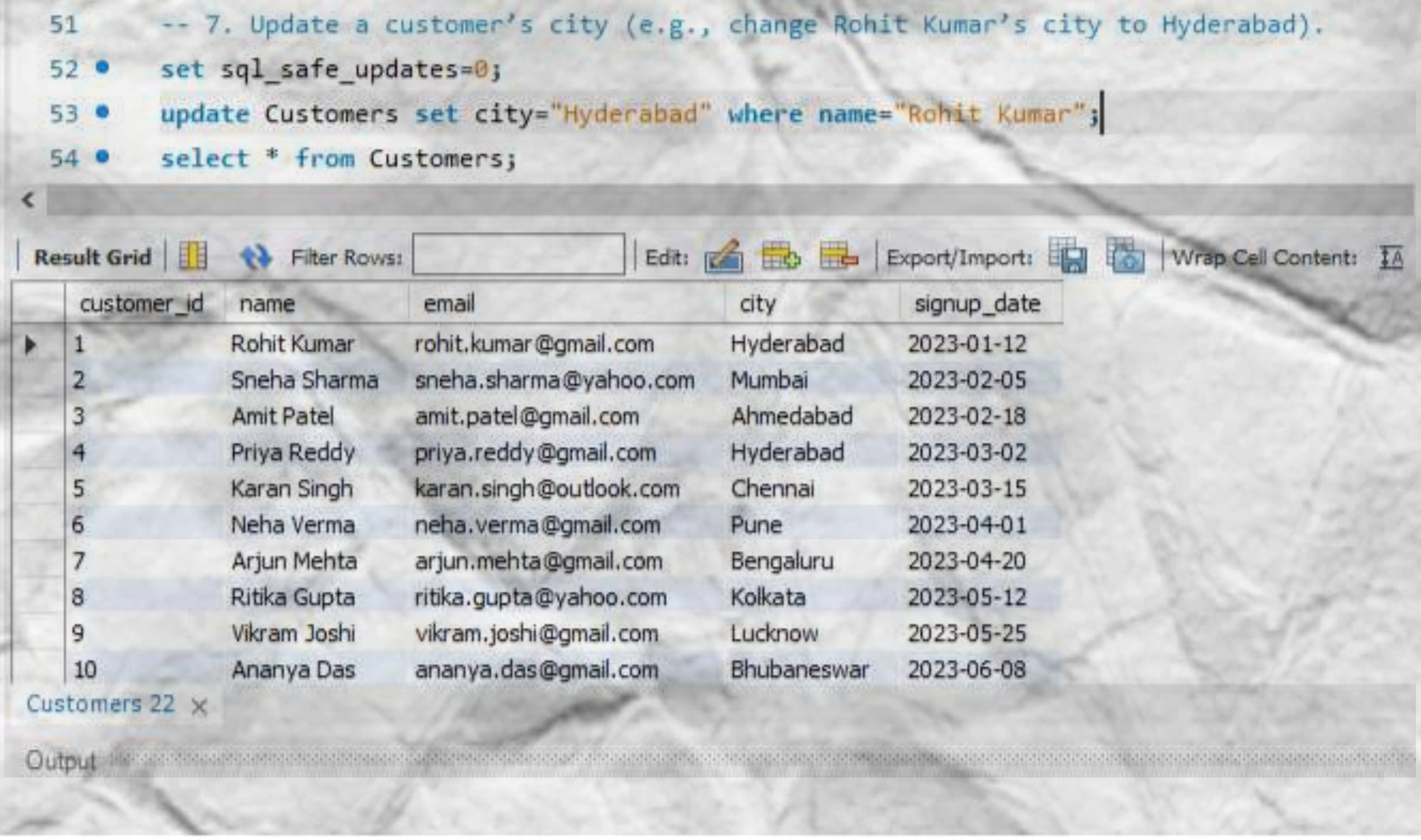
6. Count how many customers exist in the system.



```
-- 6. Count how many customers exist in the system.
Select count(customer_id) from Customers;
```

count(customer_id)
15

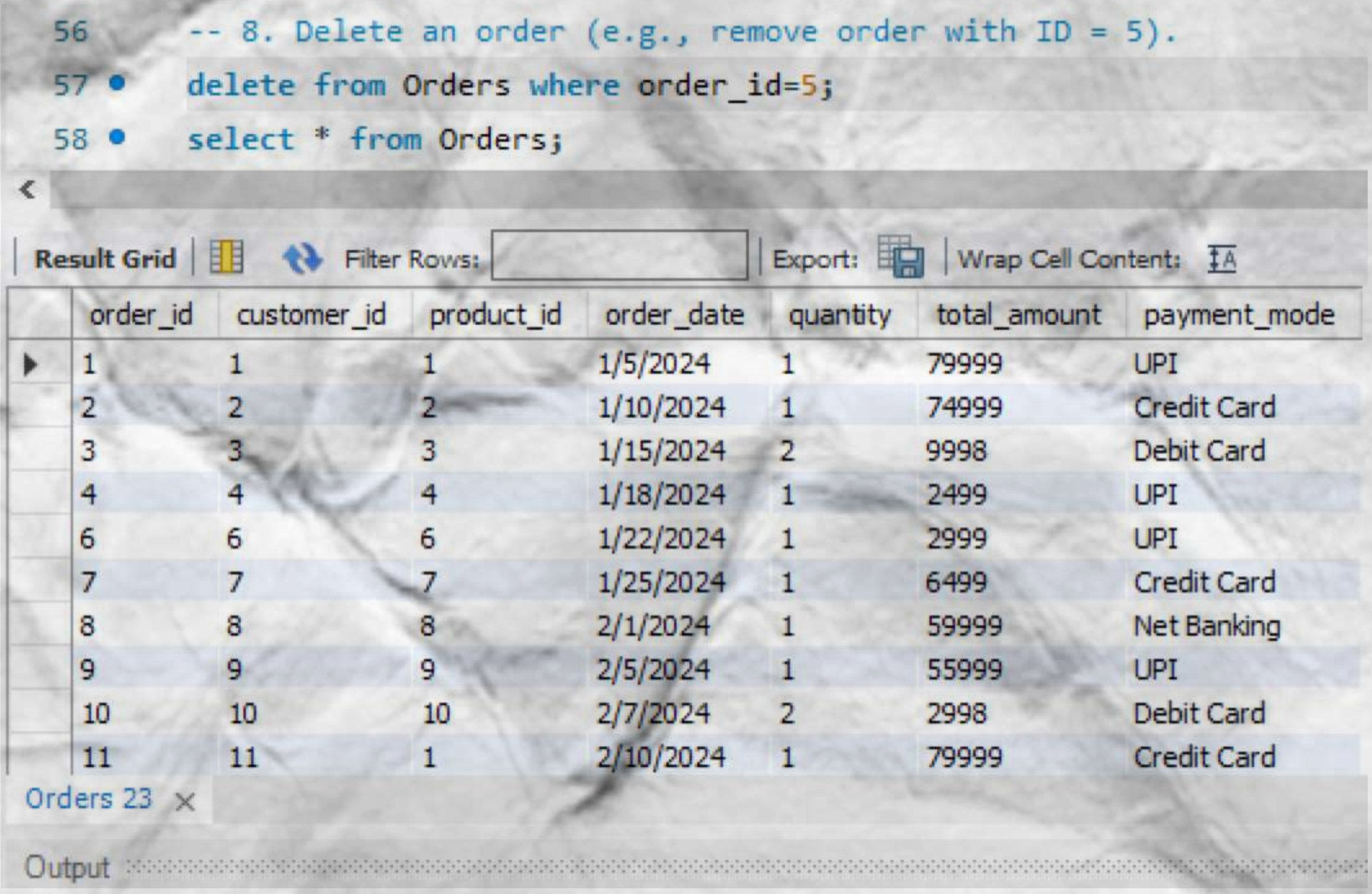
7. Update a customer's city (e.g., change Rohit Kumar's city to Hyderabad).



```
-- 7. Update a customer's city (e.g., change Rohit Kumar's city to Hyderabad).
set sql_safe_updates=0;
update Customers set city="Hyderabad" where name="Rohit Kumar";
select * from Customers;
```

customer_id	name	email	city	signup_date
1	Rohit Kumar	rohit.kumar@gmail.com	Hyderabad	2023-01-12
2	Sneha Sharma	sneha.sharma@yahoo.com	Mumbai	2023-02-05
3	Amit Patel	amit.patel@gmail.com	Ahmedabad	2023-02-18
4	Priya Reddy	priya.reddy@gmail.com	Hyderabad	2023-03-02
5	Karan Singh	karan.singh@outlook.com	Chennai	2023-03-15
6	Neha Verma	neha.verma@gmail.com	Pune	2023-04-01
7	Arjun Mehta	arjun.mehta@gmail.com	Bengaluru	2023-04-20
8	Ritika Gupta	ritika.gupta@yahoo.com	Kolkata	2023-05-12
9	Vikram Joshi	vikram.joshi@gmail.com	Lucknow	2023-05-25
10	Ananya Das	ananya.das@gmail.com	Bhubaneswar	2023-06-08

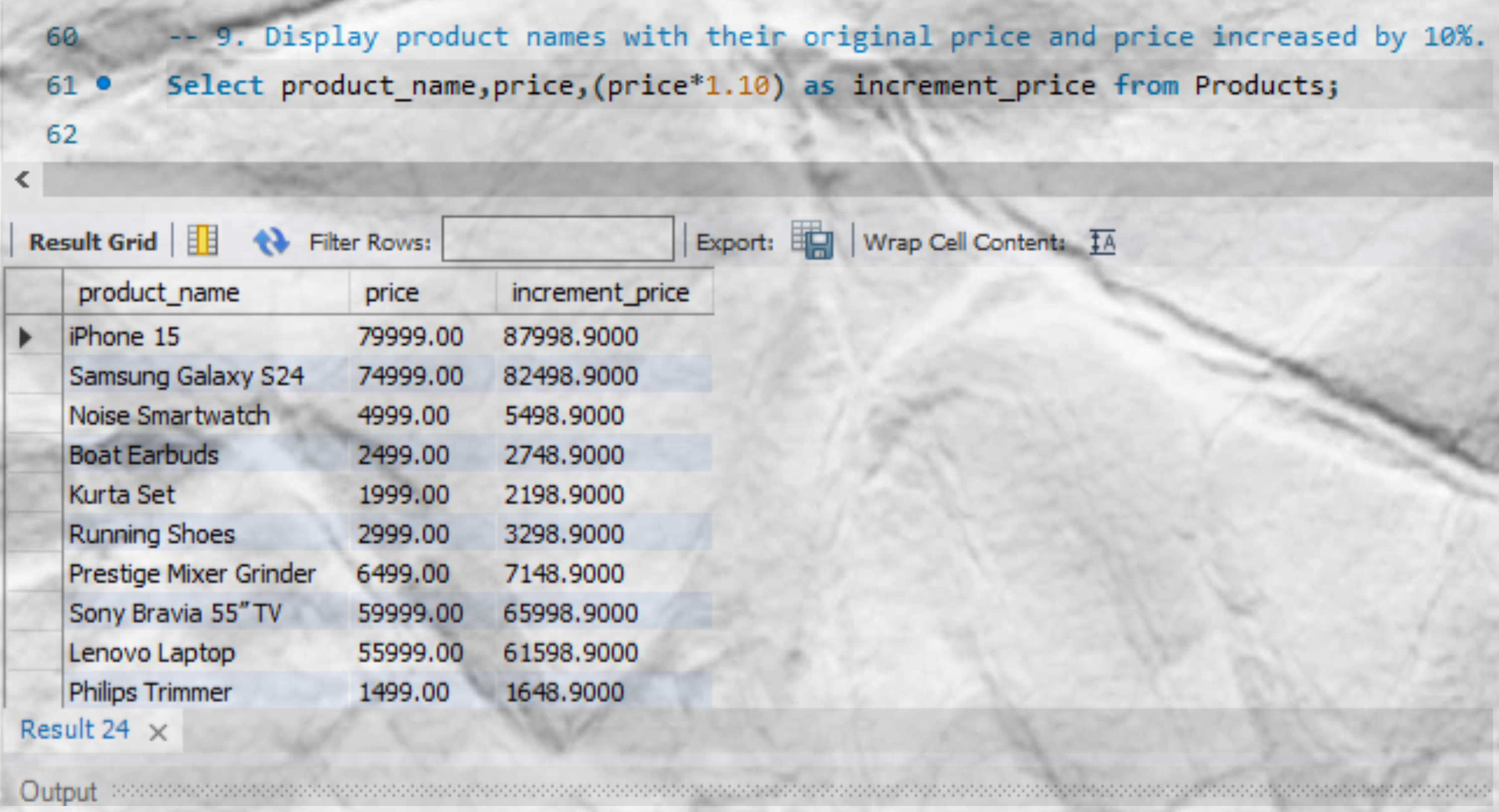
8. Delete an order (e.g., remove order with ID = 5).



```sql
-- 8. Delete an order (e.g., remove order with ID = 5).
delete from Orders where order_id=5;
select * from Orders;
```

order_id	customer_id	product_id	order_date	quantity	total_amount	payment_mode
1	1	1	1/5/2024	1	79999	UPI
2	2	2	1/10/2024	1	74999	Credit Card
3	3	3	1/15/2024	2	9998	Debit Card
4	4	4	1/18/2024	1	2499	UPI
6	6	6	1/22/2024	1	2999	UPI
7	7	7	1/25/2024	1	6499	Credit Card
8	8	8	2/1/2024	1	59999	Net Banking
9	9	9	2/5/2024	1	55999	UPI
10	10	10	2/7/2024	2	2998	Debit Card
11	11	1	2/10/2024	1	79999	Credit Card

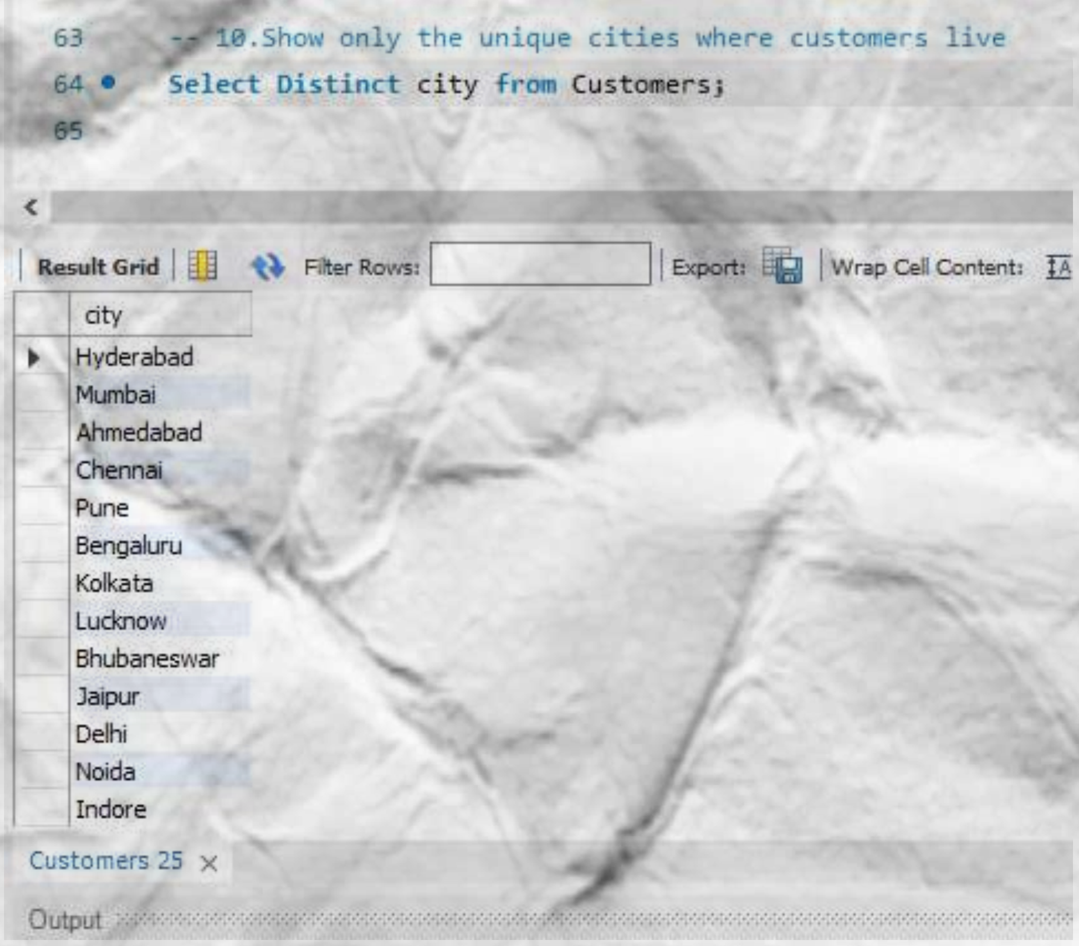
9. Display product names with their original price and price increased by 10%.



```sql
-- 9. Display product names with their original price and price increased by 10%.
Select product_name,price,(price*1.10) as increment_price from Products;
```

product_name	price	increment_price
iPhone 15	79999.00	87998.9000
Samsung Galaxy S24	74999.00	82498.9000
Noise Smartwatch	4999.00	5498.9000
Boat Earbuds	2499.00	2748.9000
Kurta Set	1999.00	2198.9000
Running Shoes	2999.00	3298.9000
Prestige Mixer Grinder	6499.00	7148.9000
Sony Bravia 55" TV	59999.00	65998.9000
Lenovo Laptop	55999.00	61598.9000
Philips Trimmer	1499.00	1648.9000

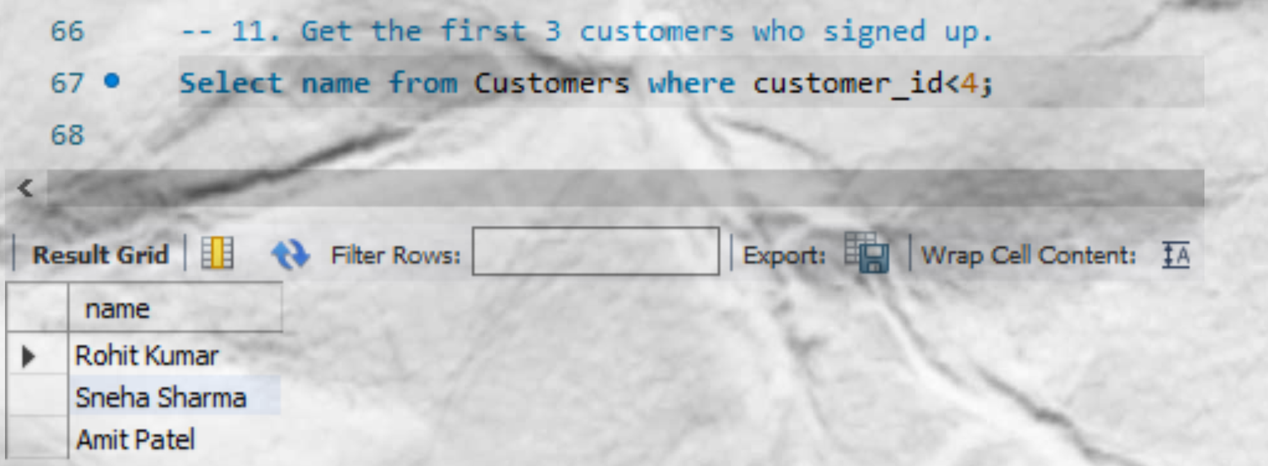
10.Show only the unique cities where customers live.



```
-- 10.Show only the unique cities where customers live
Select Distinct city from Customers;
```

city
Hyderabad
Mumbai
Ahmedabad
Chennai
Pune
Bengaluru
Kolkata
Lucknow
Bhubaneswar
Jaipur
Delhi
Noida
Indore

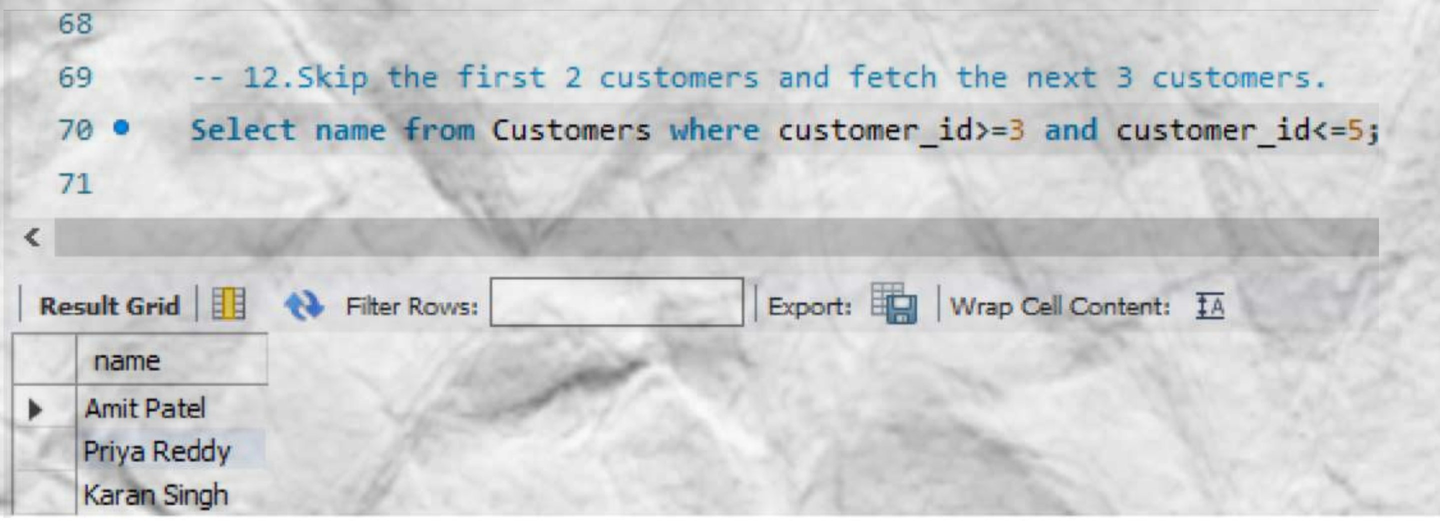
11. Get the first 3 customers who signed up.



```
-- 11. Get the first 3 customers who signed up.
Select name from Customers where customer_id<4;
```

name
Rohit Kumar
Sneha Sharma
Amit Patel

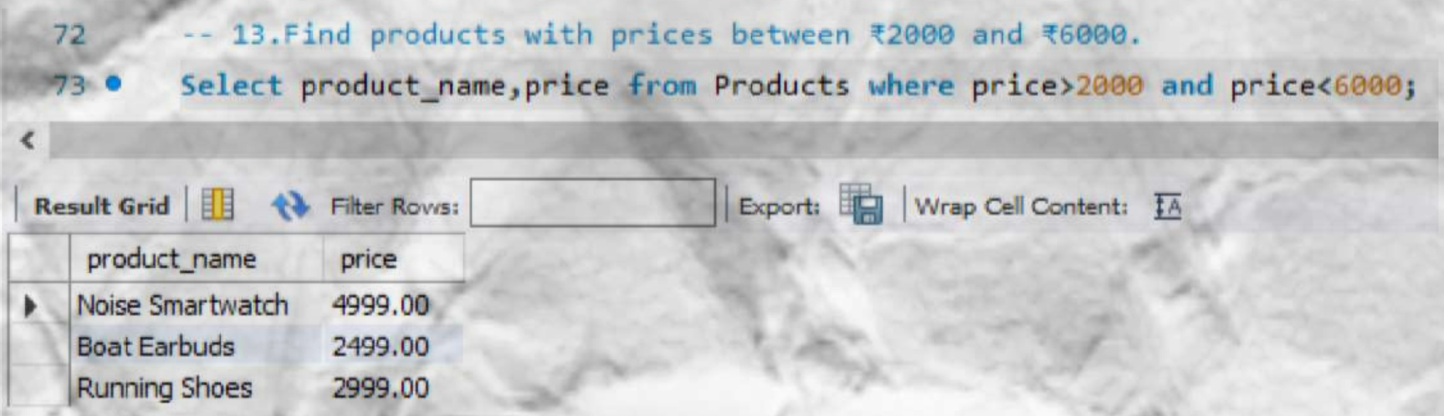
12.Skip the first 2 customers and fetch the next 3 customers.



```
-- 12.Skip the first 2 customers and fetch the next 3 customers.
Select name from Customers where customer_id>=3 and customer_id<=5;
```

name
Amit Patel
Priya Reddy
Karan Singh

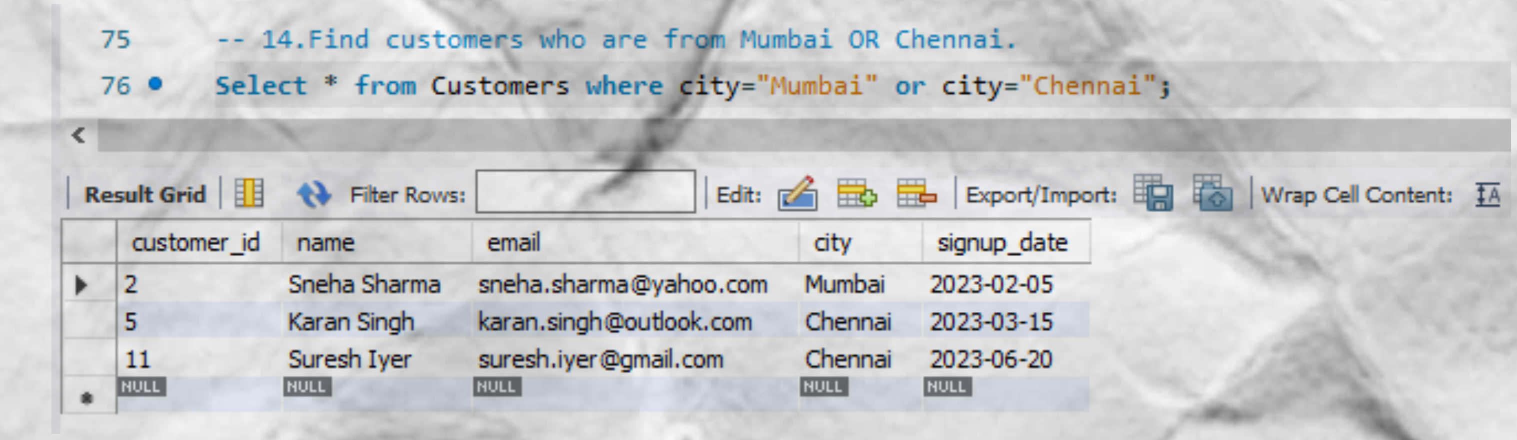
13.Find products with prices between ₹2000 and ₹6000.



```
-- 13.Find products with prices between ₹2000 and ₹6000.
Select product_name,price from Products where price>2000 and price<6000;
```

product_name	price
Noise Smartwatch	4999.00
Boat Earbuds	2499.00
Running Shoes	2999.00

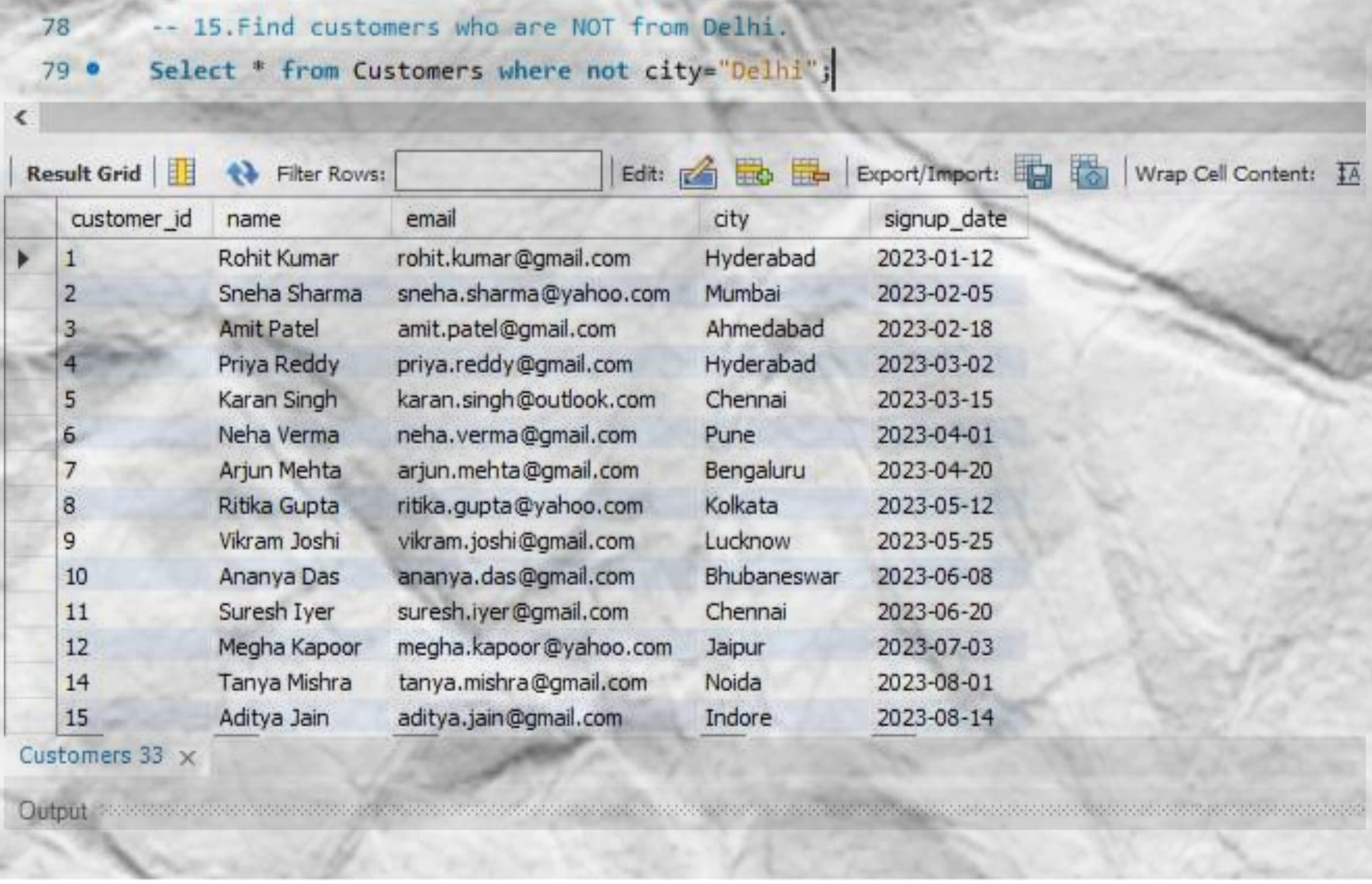
14.Find customers who are from Mumbai OR Chennai.



```
-- 14.Find customers who are from Mumbai OR Chennai.
Select * from Customers where city="Mumbai" or city="Chennai";
```

customer_id	name	email	city	signup_date
2	Sneha Sharma	sneha.sharma@yahoo.com	Mumbai	2023-02-05
5	Karan Singh	karan.singh@outlook.com	Chennai	2023-03-15
11	Suresh Iyer	suresh.iyer@gmail.com	Chennai	2023-06-20
NULL	NULL	NULL	NULL	NULL

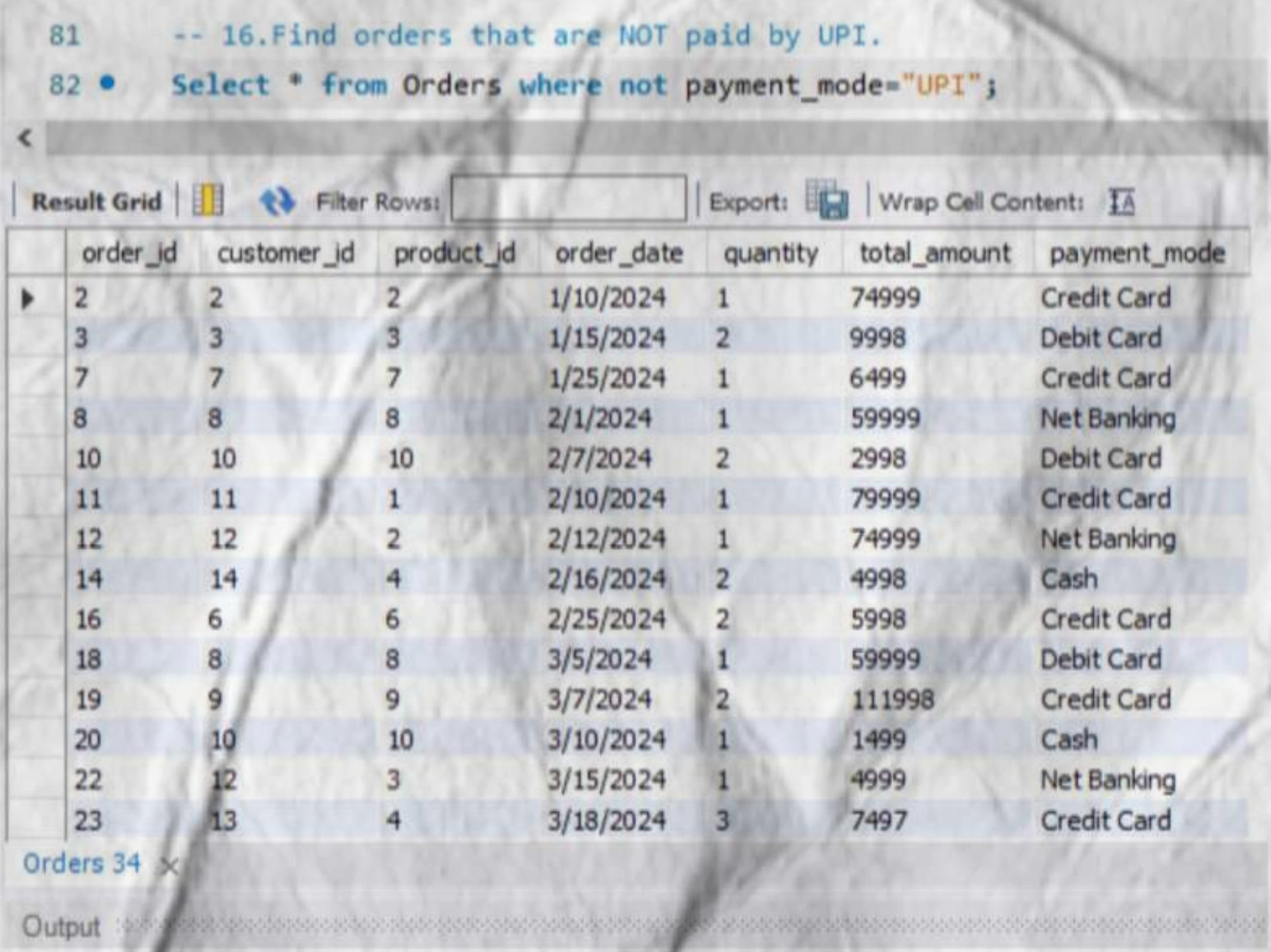
15.Find customers who are NOT from Delhi.



```
-- 15.Find customers who are NOT from Delhi.
Select * from Customers where not city="Delhi";
```

customer_id	name	email	city	signup_date
1	Rohit Kumar	rohit.kumar@gmail.com	Hyderabad	2023-01-12
2	Sneha Sharma	sneha.sharma@yahoo.com	Mumbai	2023-02-05
3	Amit Patel	amit.patel@gmail.com	Ahmedabad	2023-02-18
4	Priya Reddy	priya.reddy@gmail.com	Hyderabad	2023-03-02
5	Karan Singh	karan.singh@outlook.com	Chennai	2023-03-15
6	Neha Verma	neha.verma@gmail.com	Pune	2023-04-01
7	Arjun Mehta	arjun.mehta@gmail.com	Bengaluru	2023-04-20
8	Ritika Gupta	ritika.gupta@yahoo.com	Kolkata	2023-05-12
9	Vikram Joshi	vikram.joshi@gmail.com	Lucknow	2023-05-25
10	Ananya Das	ananya.das@gmail.com	Bhubaneswar	2023-06-08
11	Suresh Iyer	suresh.iyer@gmail.com	Chennai	2023-06-20
12	Megha Kapoor	megha.kapoor@yahoo.com	Jaipur	2023-07-03
14	Tanya Mishra	tanya.mishra@gmail.com	Noida	2023-08-01
15	Aditya Jain	aditya.jain@gmail.com	Indore	2023-08-14

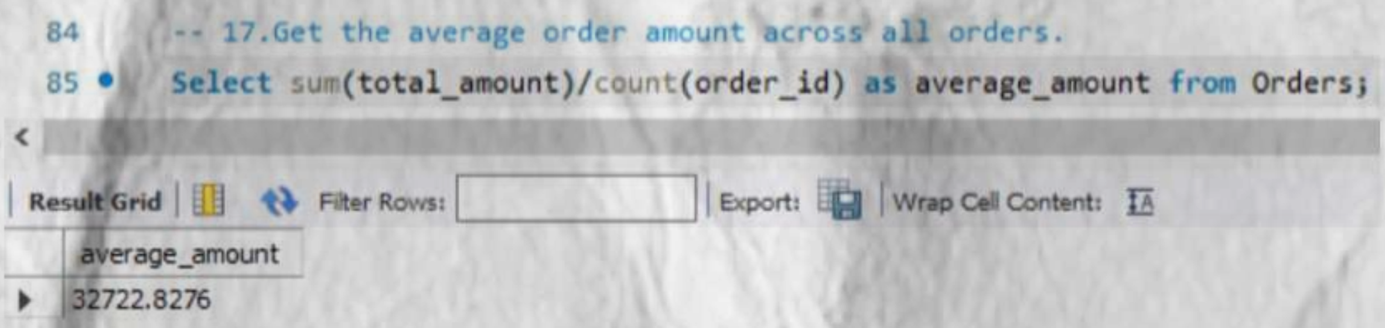
16.Find orders that are NOT paid by UPI.



```
-- 16.Find orders that are NOT paid by UPI.
Select * from Orders where not payment_mode="UPI";
```

order_id	customer_id	product_id	order_date	quantity	total_amount	payment_mode
2	2	2	1/10/2024	1	74999	Credit Card
3	3	3	1/15/2024	2	9998	Debit Card
7	7	7	1/25/2024	1	6499	Credit Card
8	8	8	2/1/2024	1	59999	Net Banking
10	10	10	2/7/2024	2	2998	Debit Card
11	11	1	2/10/2024	1	79999	Credit Card
12	12	2	2/12/2024	1	74999	Net Banking
14	14	4	2/16/2024	2	4998	Cash
16	6	6	2/25/2024	2	5998	Credit Card
18	8	8	3/5/2024	1	59999	Debit Card
19	9	9	3/7/2024	2	111998	Credit Card
20	10	10	3/10/2024	1	1499	Cash
22	12	3	3/15/2024	1	4999	Net Banking
23	13	4	3/18/2024	3	7497	Credit Card

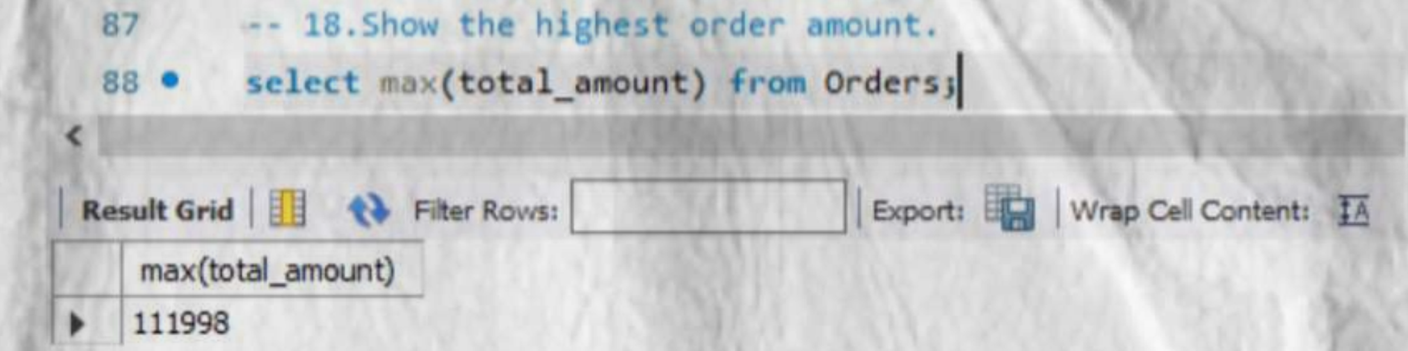
17.Get the average order amount across all orders.



```
-- 17.Get the average order amount across all orders.
Select sum(total_amount)/count(order_id) as average_amount from Orders;
```

average_amount
32722.8276

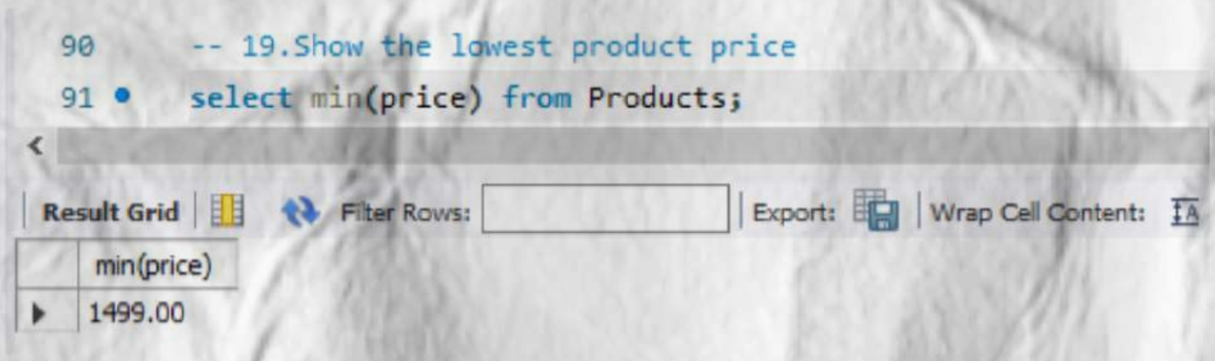
18.Show the highest order amount.



```
-- 18.Show the highest order amount.
select max(total_amount) from Orders;
```

max(total_amount)
111998

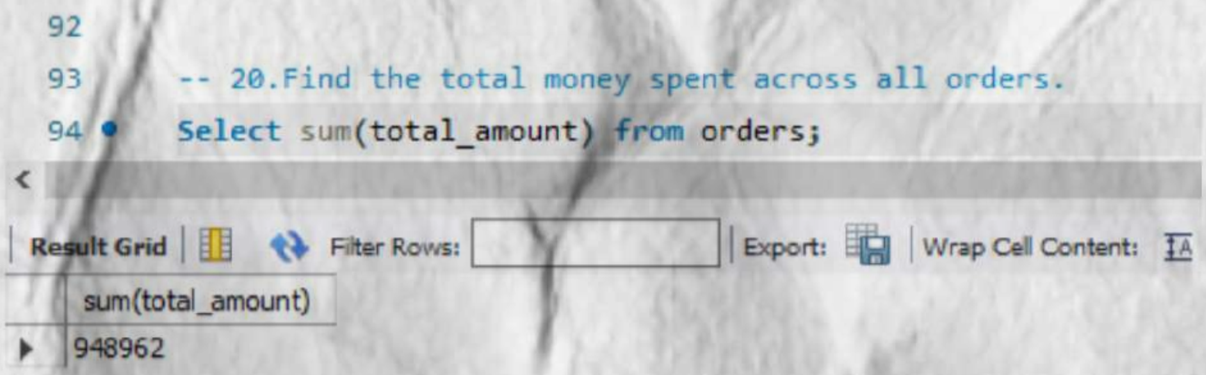
19.Show the lowest product price.



```sql
-- 19.Show the lowest product price
select min(price) from Products;
```

```
min(price)
1499.00
```

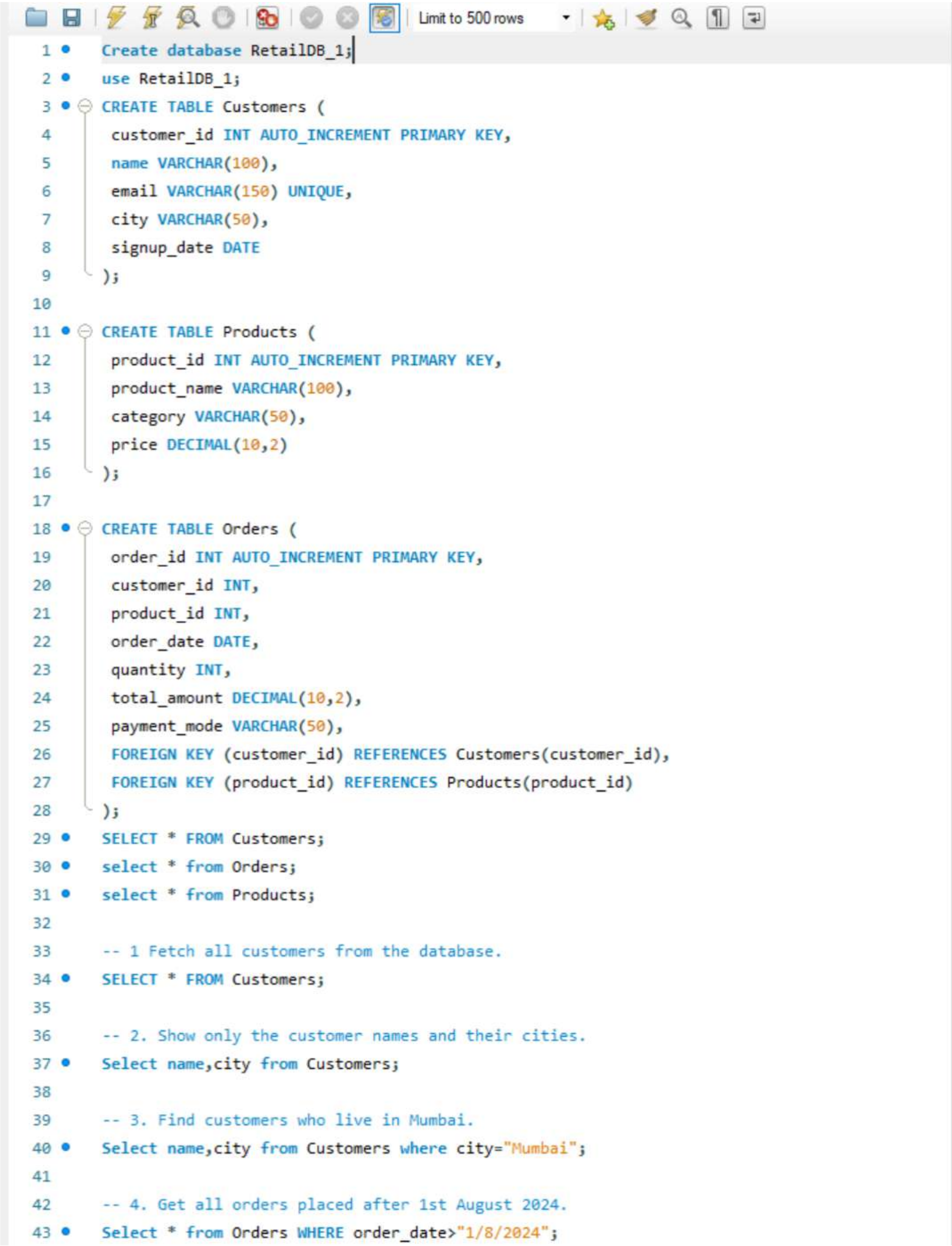
20.Find the total money spent across all orders



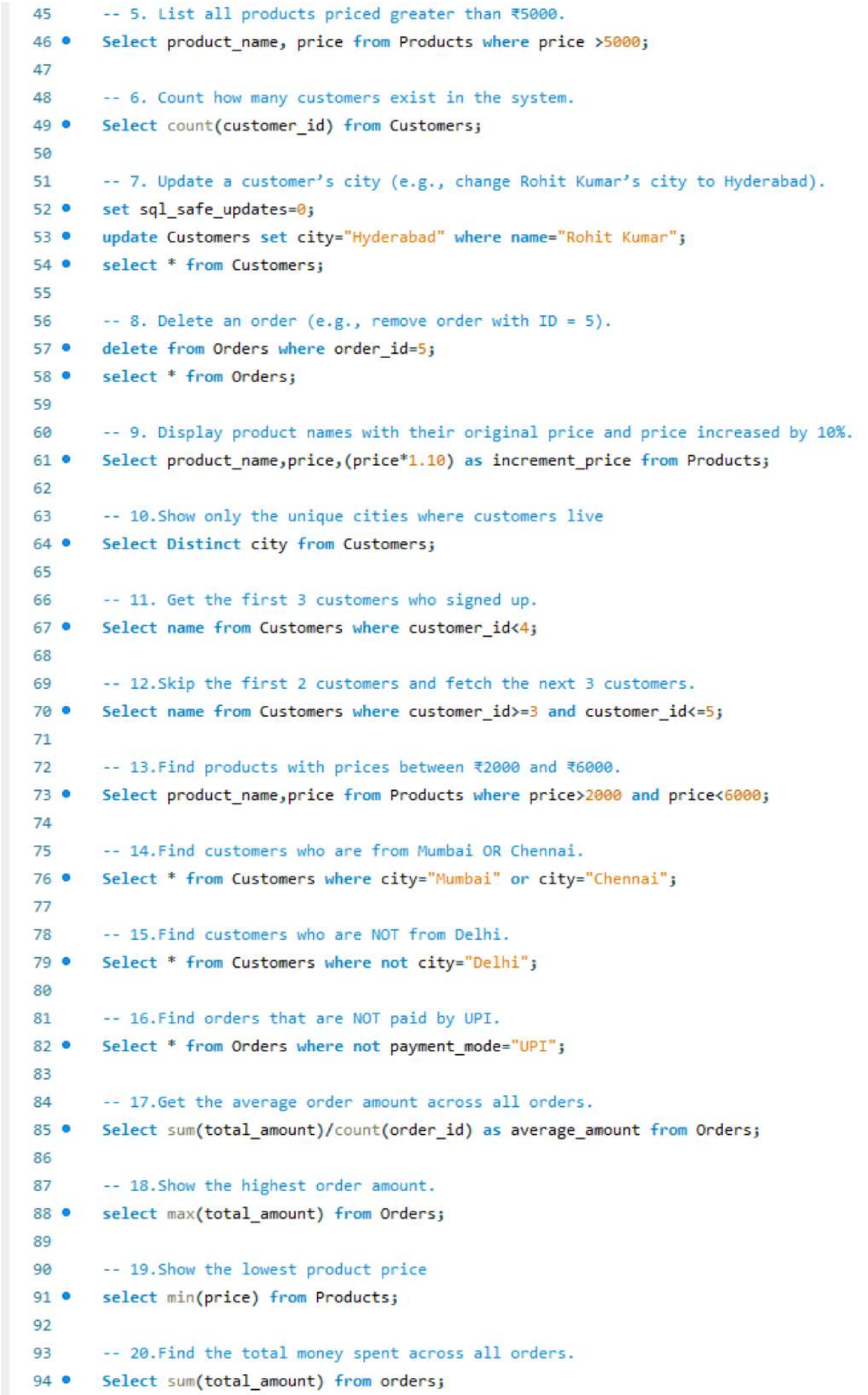
```sql
-- 20.Find the total money spent across all orders.
Select sum(total_amount) from orders;
```

```
sum(total_amount)
948962
```

Overview of MySQL workbench



```sql
Create database RetailDB_1;
use RetailDB_1;
CREATE TABLE Customers (
 customer_id INT AUTO_INCREMENT PRIMARY KEY,
 name VARCHAR(100),
 email VARCHAR(150) UNIQUE,
 city VARCHAR(50),
 signup_date DATE
);

CREATE TABLE Products (
 product_id INT AUTO_INCREMENT PRIMARY KEY,
 product_name VARCHAR(100),
 category VARCHAR(50),
 price DECIMAL(10,2)
);

CREATE TABLE Orders (
 order_id INT AUTO_INCREMENT PRIMARY KEY,
 customer_id INT,
 product_id INT,
 order_date DATE,
 quantity INT,
 total_amount DECIMAL(10,2),
 payment_mode VARCHAR(50),
 FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
 FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
SELECT * FROM Customers;
select * from Orders;
select * from Products;

-- 1 Fetch all customers from the database.
SELECT * FROM Customers;

-- 2. Show only the customer names and their cities.
Select name,city from Customers;

-- 3. Find customers who live in Mumbai.
Select name,city from Customers where city="Mumbai";

-- 4. Get all orders placed after 1st August 2024.
Select * from Orders WHERE order_date>"1/8/2024";
```



```sql
-- 5. List all products priced greater than ₹5000.
Select product_name, price from Products where price >5000;

-- 6. Count how many customers exist in the system.
Select count(customer_id) from Customers;

-- 7. Update a customer’s city (e.g., change Rohit Kumar’s city to Hyderabad).
set sql_safe_updates=0;
update Customers set city="Hyderabad" where name="Rohit Kumar";
select * from Customers;

-- 8. Delete an order (e.g., remove order with ID = 5).
delete from Orders where order_id=5;
select * from Orders;

-- 9. Display product names with their original price and price increased by 10%.
Select product_name,price,(price*1.10) as increment_price from Products;

-- 10.Show only the unique cities where customers live
Select Distinct city from Customers;

-- 11. Get the first 3 customers who signed up.
Select name from Customers where customer_id<4;

-- 12.Skip the first 2 customers and fetch the next 3 customers.
Select name from Customers where customer_id>=3 and customer_id<=5;

-- 13.Find products with prices between ₹2000 and ₹6000.
Select product_name,price from Products where price>2000 and price<6000;

-- 14.Find customers who are from Mumbai OR Chennai.
Select * from Customers where city="Mumbai" or city="Chennai";

-- 15.Find customers who are NOT from Delhi.
Select * from Customers where not city="Delhi";

-- 16.Find orders that are NOT paid by UPI.
Select * from Orders where not payment_mode="UPI";

-- 17.Get the average order amount across all orders.
Select sum(total_amount)/count(order_id) as average_amount from Orders;

-- 18.Show the highest order amount.
select max(total_amount) from Orders;

-- 19.Show the lowest product price
select min(price) from Products;

-- 20.Find the total money spent across all orders.
Select sum(total_amount) from orders;
```

Thank

you