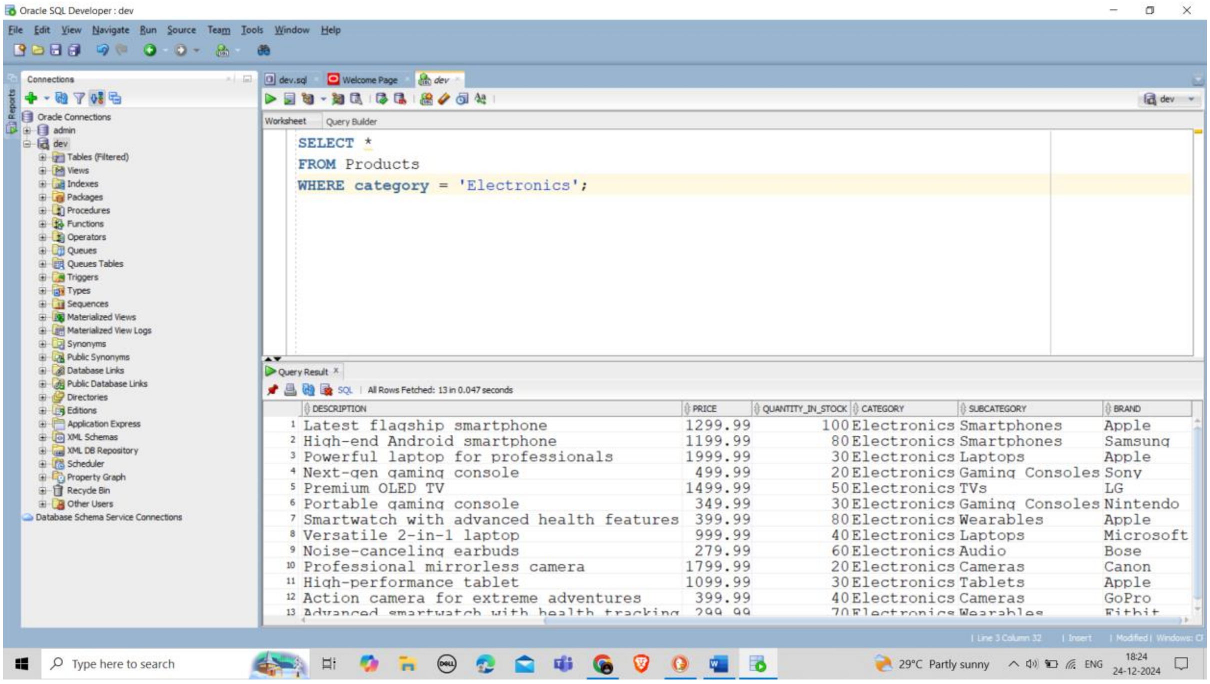
# ECOMMERCE STORE DATA REPORT

## 1. Find all products in a specific category:

SELECT *

FROM Products

WHERE category = 'Electronics';



```
SELECT *
FROM Products
WHERE category = 'Electronics';
```

All Rows Fetched: 13 in 0.047 seconds

| | DESCRIPTION | PRICE | QUANTITY_IN_STOCK | CATEGORY | SUBCATEGORY | BRAND |
|---|---|---|---|---|---|---|
| 1 | Latest flagship smartphone | 1299.99 | 100 | Electronics | Smartphones | Apple |
| 2 | High-end Android smartphone | 1199.99 | 80 | Electronics | Smartphones | Samsung |
| 3 | Powerful laptop for professionals | 1999.99 | 30 | Electronics | Laptops | Apple |
| 4 | Next-gen gaming console | 499.99 | 20 | Electronics | Gaming Consoles | Sony |
| 5 | Premium OLED TV | 1499.99 | 50 | Electronics | TVs | LG |
| 6 | Portable gaming console | 349.99 | 30 | Electronics | Gaming Consoles | Nintendo |
| 7 | Smartwatch with advanced health features | 399.99 | 80 | Electronics | Wearables | Apple |
| 8 | Versatile 2-in-1 laptop | 999.99 | 40 | Electronics | Laptops | Microsoft |
| 9 | Noise-canceling earbuds | 279.99 | 60 | Electronics | Audio | Bose |
| 10 | Professional mirrorless camera | 1799.99 | 20 | Electronics | Cameras | Canon |
| 11 | High-performance tablet | 1099.99 | 30 | Electronics | Tablets | Apple |
| 12 | Action camera for extreme adventures | 399.99 | 40 | Electronics | Cameras | GoPro |
| 13 | Advanced smartwatch with health tracking | 299.99 | 70 | Electronics | Wearables | Fitbit |

## 2. Find the most expensive product:

SELECT *

FROM (

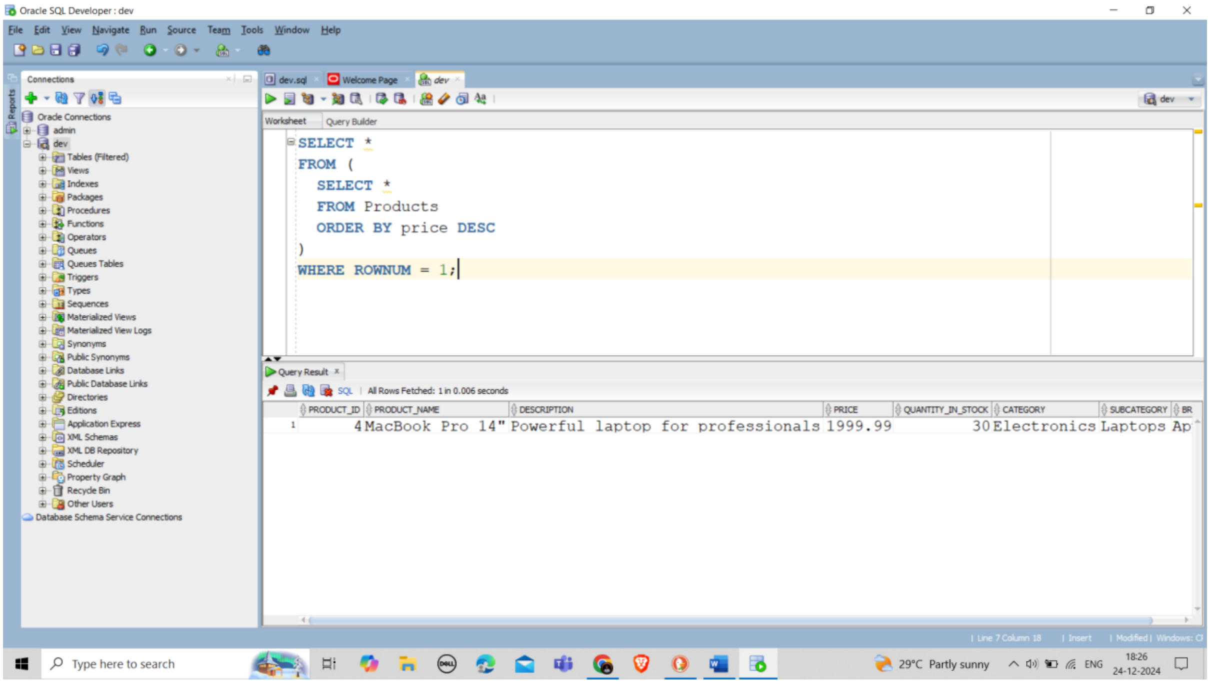
  SELECT *

  FROM Products

  ORDER BY price DESC

)

WHERE ROWNUM = 1;



```
SELECT *
FROM (
  SELECT *
  FROM Products
  ORDER BY price DESC
)
WHERE ROWNUM = 1;
```

Query Result

All Rows Fetched: 1 in 0.006 seconds

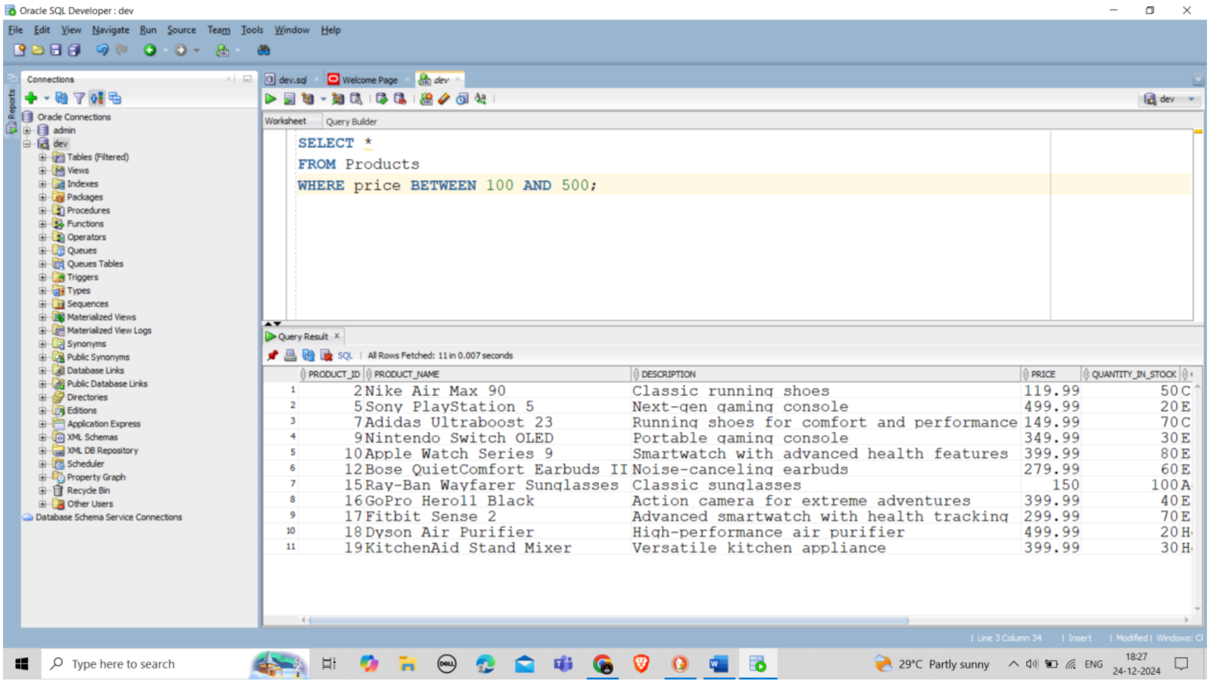
| | PRODUCT_ID | PRODUCT_NAME | DESCRIPTION | PRICE | QUANTITY_IN_STOCK | CATEGORY | SUBCATEGORY | BR |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | MacBook Pro 14" | Powerful laptop for professionals | 1999.99 | 30 | Electronics | Laptops | Ap |

## 3. Find products with a price range:

SELECT *

FROM Products

WHERE price BETWEEN 100 AND 500;



```
SELECT *
FROM Products
WHERE price BETWEEN 100 AND 500;
```

Query Result

All Rows Fetched: 11 in 0.007 seconds

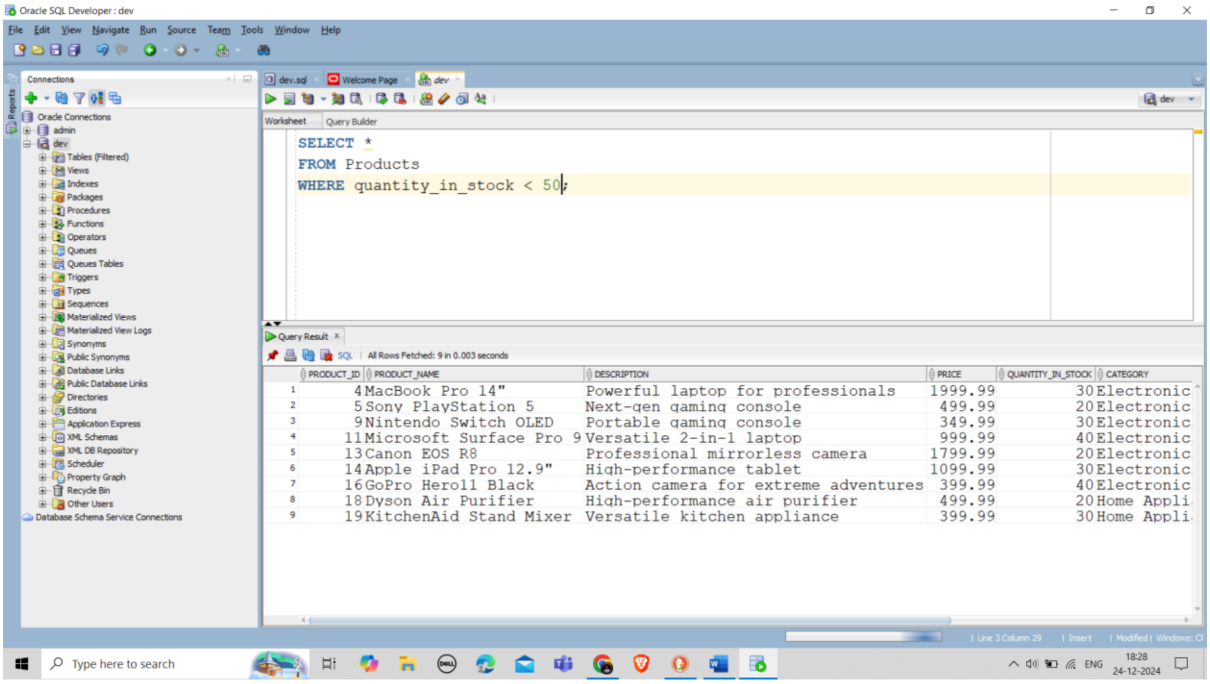
| | PRODUCT_ID | PRODUCT_NAME | DESCRIPTION | PRICE | QUANTITY_IN_STOCK | |
|---|---|---|---|---|---|---|
| 1 | 2 | Nike Air Max 90 | Classic running shoes | 119.99 | 50 | C |
| 2 | 5 | Sony PlayStation 5 | Next-gen gaming console | 499.99 | 20 | E |
| 3 | 7 | Adidas Ultraboost 23 | Running shoes for comfort and performance | 149.99 | 70 | C |
| 4 | 9 | Nintendo Switch OLED | Portable gaming console | 349.99 | 30 | E |
| 5 | 10 | Apple Watch Series 9 | Smartwatch with advanced health features | 399.99 | 80 | E |
| 6 | 12 | Bose QuietComfort Earbuds II | Noise-canceling earbuds | 279.99 | 60 | E |
| 7 | 15 | Ray-Ban Wayfarer Sunglasses | Classic sunglasses | 150 | 100 | A |
| 8 | 16 | GoPro Hero11 Black | Action camera for extreme adventures | 399.99 | 40 | E |
| 9 | 17 | Fitbit Sense 2 | Advanced smartwatch with health tracking | 299.99 | 70 | E |
| 10 | 18 | Dyson Air Purifier | High-performance air purifier | 499.99 | 20 | H |
| 11 | 19 | KitchenAid Stand Mixer | Versatile kitchen appliance | 399.99 | 30 | H |

## 4. Find products with low stock:

SELECT *

FROM Products

WHERE quantity_in_stock < 50;



```
SELECT *
FROM Products
WHERE quantity_in_stock < 50;
```

All Rows Fetched: 9 in 0.003 seconds

| | PRODUCT_ID | PRODUCT_NAME | DESCRIPTION | PRICE | QUANTITY_IN_STOCK | CATEGORY |
|---|---|---|---|---|---|---|
| 1 | 4 | MacBook Pro 14" | Powerful laptop for professionals | 1999.99 | 30 | Electronic |
| 2 | 5 | Sony PlayStation 5 | Next-gen gaming console | 499.99 | 20 | Electronic |
| 3 | 9 | Nintendo Switch OLED | Portable gaming console | 349.99 | 30 | Electronic |
| 4 | 11 | Microsoft Surface Pro 9 | Versatile 2-in-1 laptop | 999.99 | 40 | Electronic |
| 5 | 13 | Canon EOS R8 | Professional mirrorless camera | 1799.99 | 20 | Electronic |
| 6 | 14 | Apple iPad Pro 12.9" | High-performance tablet | 1099.99 | 30 | Electronic |
| 7 | 16 | GoPro Hero11 Black | Action camera for extreme adventures | 399.99 | 40 | Electronic |
| 8 | 18 | Dyson Air Purifier | High-performance air purifier | 499.99 | 20 | Home Appli |
| 9 | 19 | KitchenAid Stand Mixer | Versatile kitchen appliance | 399.99 | 30 | Home Appli |

## 5.Find customers who have made more than a certain number of orders:

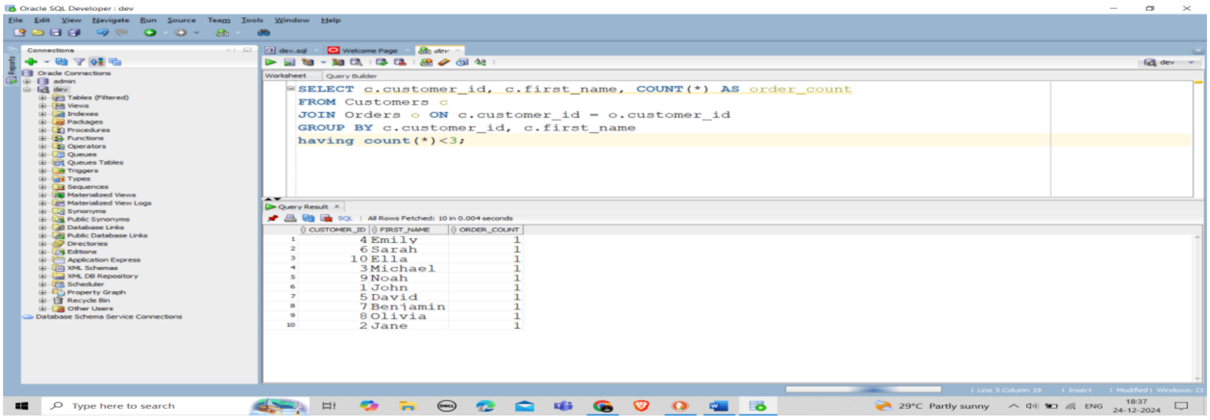
SELECT c.customer_id, c.first_name, COUNT(*) AS order_count

FROM Customers c

JOIN Orders o ON c.customer_id = o.customer_id

GROUP BY c.customer_id, c.first_name

having count(*)<3;



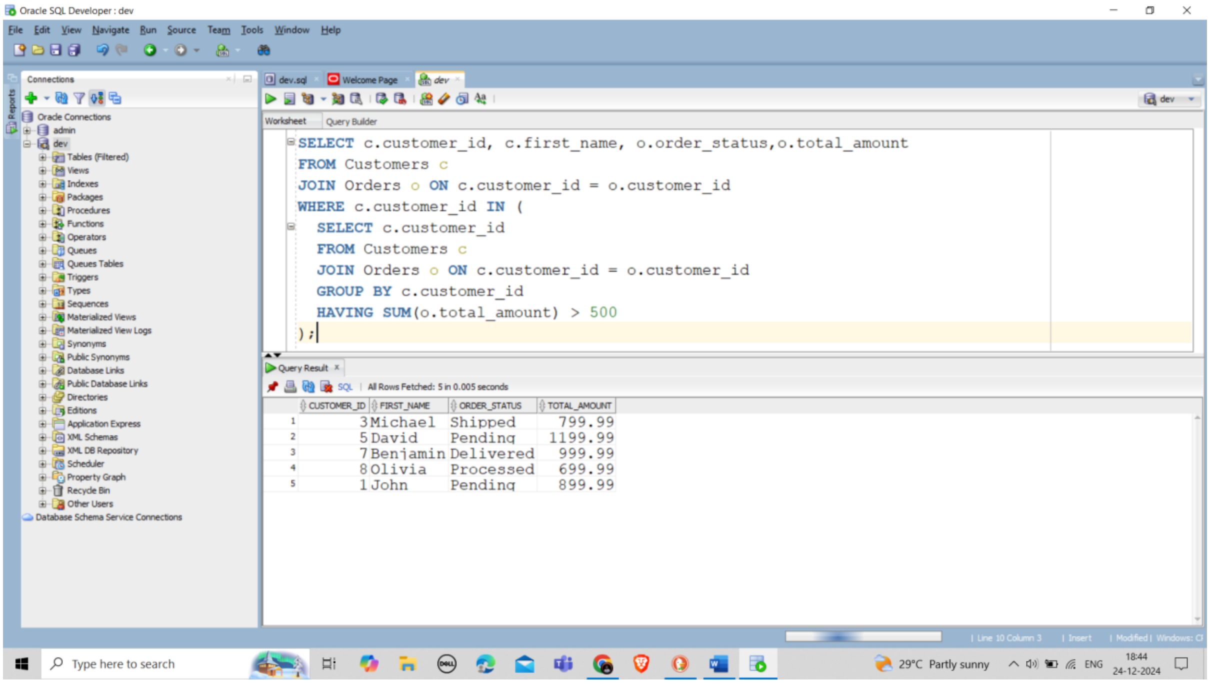
```
SELECT c.customer_id, c.first_name, COUNT(*) AS order_count
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.first_name
having count(*)<3;
```

All Rows Fetched: 10 in 0.004 seconds

| | CUSTOMER_ID | FIRST_NAME | ORDER_COUNT |
|---|---|---|---|
| 1 | 4 | Emily | 1 |
| 2 | 6 | Sarah | 1 |
| 3 | 10 | Ella | 1 |
| 4 | 3 | Michael | 1 |
| 5 | 9 | Noah | 1 |
| 6 | 1 | John | 1 |
| 7 | 5 | David | 1 |
| 8 | 7 | Benjamin | 1 |
| 9 | 8 | Olivia | 1 |
| 10 | 2 | Jane | 1 |

## 6.Find customers who have spent more than a certain amount:

```
SELECT c.customer_id, c.first_name, o.order_status,o.total_amount

FROM Customers c

JOIN Orders o ON c.customer_id = o.customer_id

WHERE c.customer_id IN (

  SELECT c.customer_id

  FROM Customers c

  JOIN Orders o ON c.customer_id = o.customer_id

  GROUP BY c.customer_id

  HAVING SUM(o.total_amount) > 500

);
```



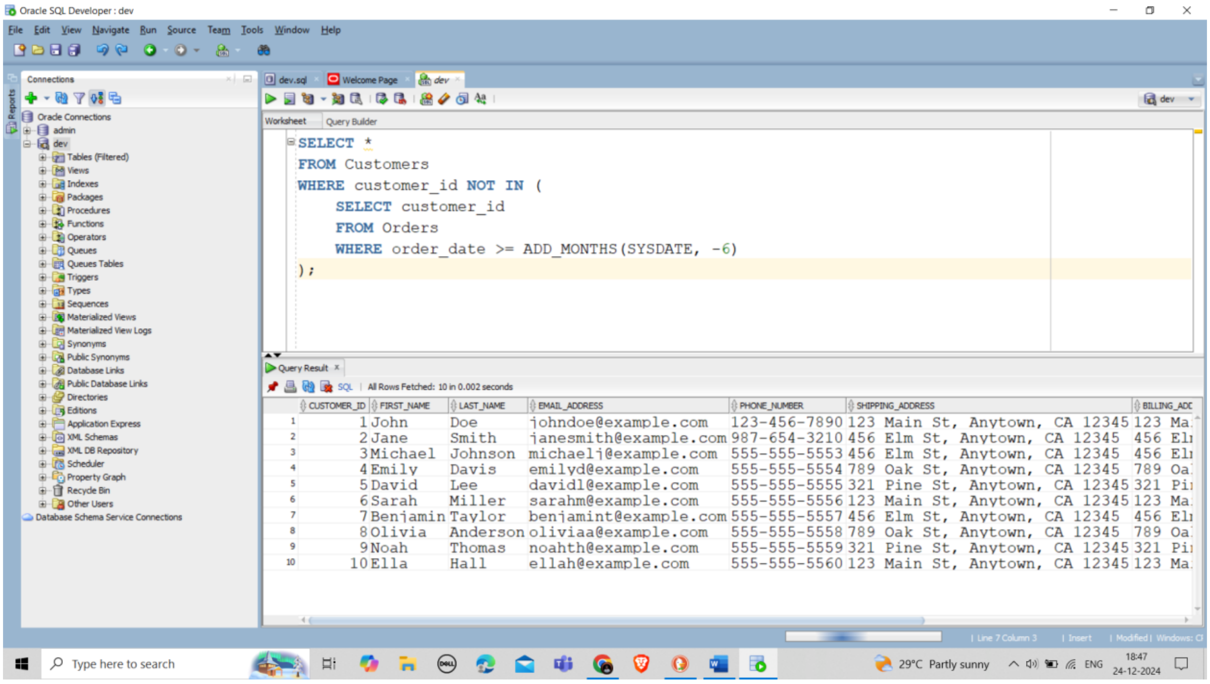
```
SELECT c.customer_id, c.first_name, o.order_status,o.total_amount
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
WHERE c.customer_id IN (
  SELECT c.customer_id
  FROM Customers c
  JOIN Orders o ON c.customer_id = o.customer_id
  GROUP BY c.customer_id
  HAVING SUM(o.total_amount) > 500
);
```

All Rows Fetched: 5 in 0.005 seconds

```
   CUSTOMER_ID FIRST_NAME ORDER_STATUS TOTAL_AMOUNT
1            3 Michael    Shipped            799.99
2            5 David      Pending           1199.99
3            7 Benjamin   Delivered          999.99
4            8 Olivia     Processed          699.99
5            1 John       Pending            899.99
```

## 7.Find customers who have not placed an order in the last 6 months:

```
SELECT *

FROM Customers

WHERE customer_id NOT IN (

    SELECT customer_id

    FROM Orders

    WHERE order_date >= ADD_MONTHS(SYSDATE, -6)

);
```



```sql
SELECT *
FROM Customers
WHERE customer_id NOT IN (
    SELECT customer_id
    FROM Orders
    WHERE order_date >= ADD_MONTHS(SYSDATE, -6)
);
```

All Rows Fetched: 10 in 0.002 seconds

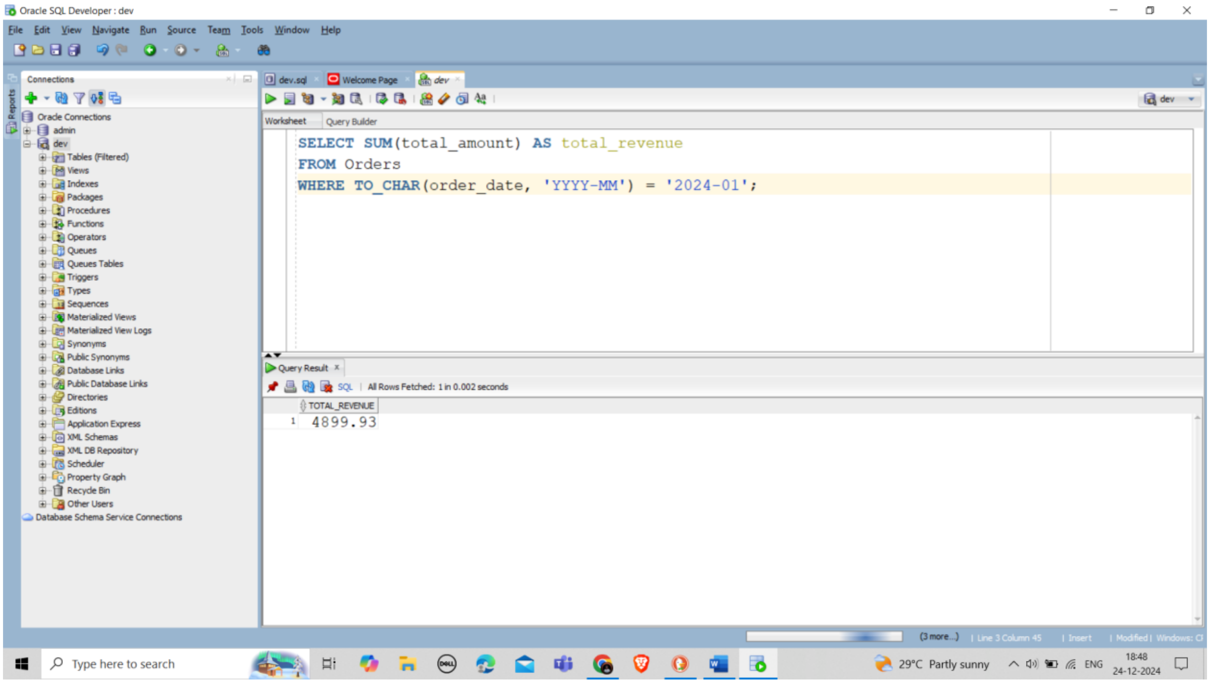
| | CUSTOMER_ID | FIRST_NAME | LAST_NAME | EMAIL_ADDRESS | PHONE_NUMBER | SHIPPING_ADDRESS | BILLING_ADD |
|---|---|---|---|---|---|---|---|
| 1 | 1 | John | Doe | johndoe@example.com | 123-456-7890 | 123 Main St, Anytown, CA 12345 | 123 Ma |
| 2 | 2 | Jane | Smith | janesmith@example.com | 987-654-3210 | 456 Elm St, Anytown, CA 12345 | 456 El |
| 3 | 3 | Michael | Johnson | michaelj@example.com | 555-555-5553 | 456 Elm St, Anytown, CA 12345 | 456 El |
| 4 | 4 | Emily | Davis | emilyd@example.com | 555-555-5554 | 789 Oak St, Anytown, CA 12345 | 789 Oa |
| 5 | 5 | David | Lee | davidl@example.com | 555-555-5555 | 321 Pine St, Anytown, CA 12345 | 321 Pi |
| 6 | 6 | Sarah | Miller | sarahm@example.com | 555-555-5556 | 123 Main St, Anytown, CA 12345 | 123 Ma |
| 7 | 7 | Benjamin | Taylor | benjamint@example.com | 555-555-5557 | 456 Elm St, Anytown, CA 12345 | 456 El |
| 8 | 8 | Olivia | Anderson | oliviaa@example.com | 555-555-5558 | 789 Oak St, Anytown, CA 12345 | 789 Oa |
| 9 | 9 | Noah | Thomas | noahth@example.com | 555-555-5559 | 321 Pine St, Anytown, CA 12345 | 321 Pi |
| 10 | 10 | Ella | Hall | ellah@example.com | 555-555-5560 | 123 Main St, Anytown, CA 12345 | 123 Ma |

## 8.Find the total revenue for a specific month:

SELECT SUM(total_amount) AS total_revenue

FROM Orders

WHERE TO_CHAR(order_date, 'YYYY-MM') = ' 2024-01;



```sql
SELECT SUM(total_amount) AS total_revenue
FROM Orders
WHERE TO_CHAR(order_date, 'YYYY-MM') = '2024-01';
```

All Rows Fetched: 1 in 0.002 seconds

| | TOTAL_REVENUE |
|---|---|
| 1 | 4899.93 |

## 9.Find the most popular product:

select * from(SELECT p.product_name, SUM(oi.quantity) AS total_quantity

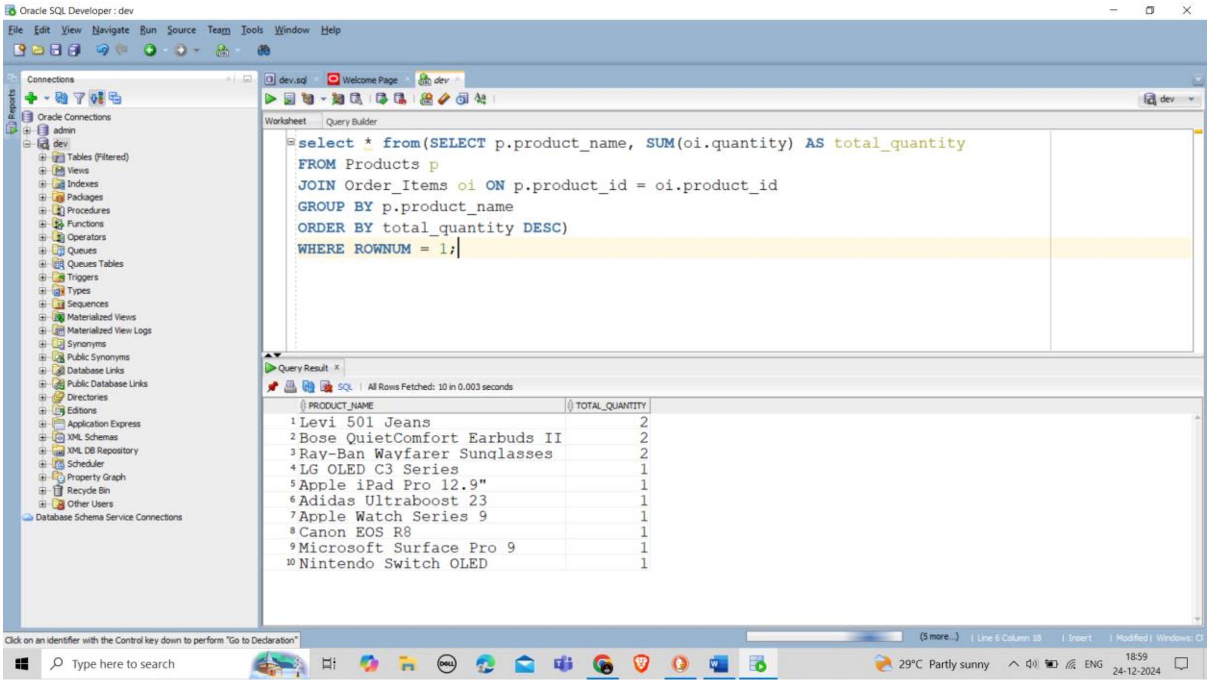
FROM Products p

JOIN Order_Items oi ON p.product_id = oi.product_id

GROUP BY p.product_name

ORDER BY total_quantity DESC)

WHERE ROWNUM = 1;



```
select * from(SELECT p.product_name, SUM(oi.quantity) AS total_quantity
FROM Products p
JOIN Order_Items oi ON p.product_id = oi.product_id
GROUP BY p.product_name
ORDER BY total_quantity DESC)
WHERE ROWNUM = 1;
```

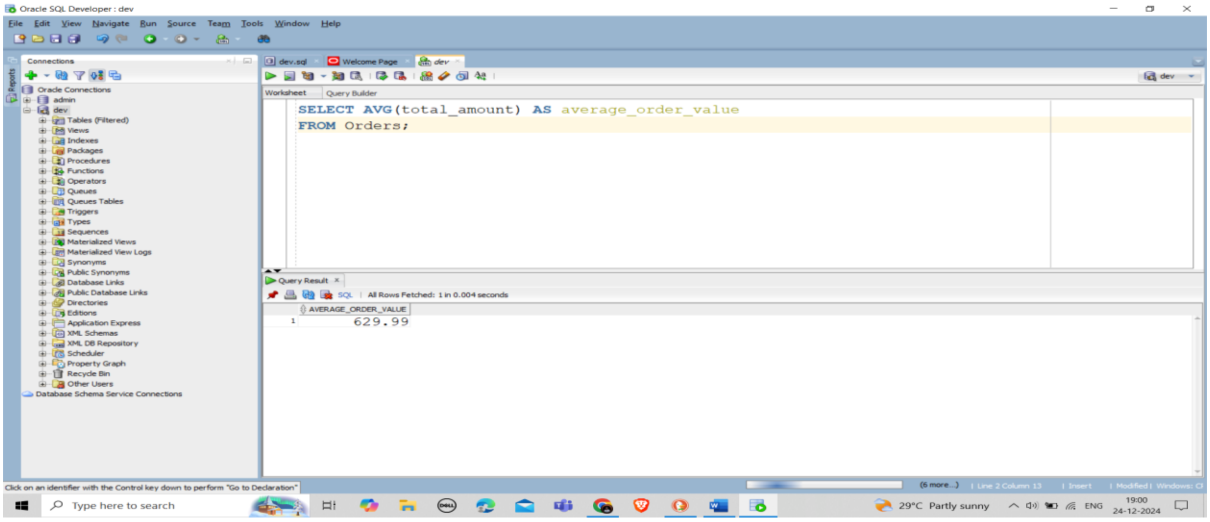
All Rows Fetched: 10 in 0.003 seconds

|  | PRODUCT_NAME | TOTAL_QUANTITY |
|---|---|---|
| 1 | Levi 501 Jeans | 2 |
| 2 | Bose QuietComfort Earbuds II | 2 |
| 3 | Ray-Ban Wayfarer Sunglasses | 2 |
| 4 | LG OLED C3 Series | 1 |
| 5 | Apple iPad Pro 12.9" | 1 |
| 6 | Adidas Ultraboost 23 | 1 |
| 7 | Apple Watch Series 9 | 1 |
| 8 | Canon EOS R8 | 1 |
| 9 | Microsoft Surface Pro 9 | 1 |
| 10 | Nintendo Switch OLED | 1 |

## 10.Find the average order value:

SELECT AVG(total_amount) AS average_order_value

FROM Orders;



```
SELECT AVG(total_amount) AS average_order_value
FROM Orders;
```

All Rows Fetched: 1 in 0.004 seconds

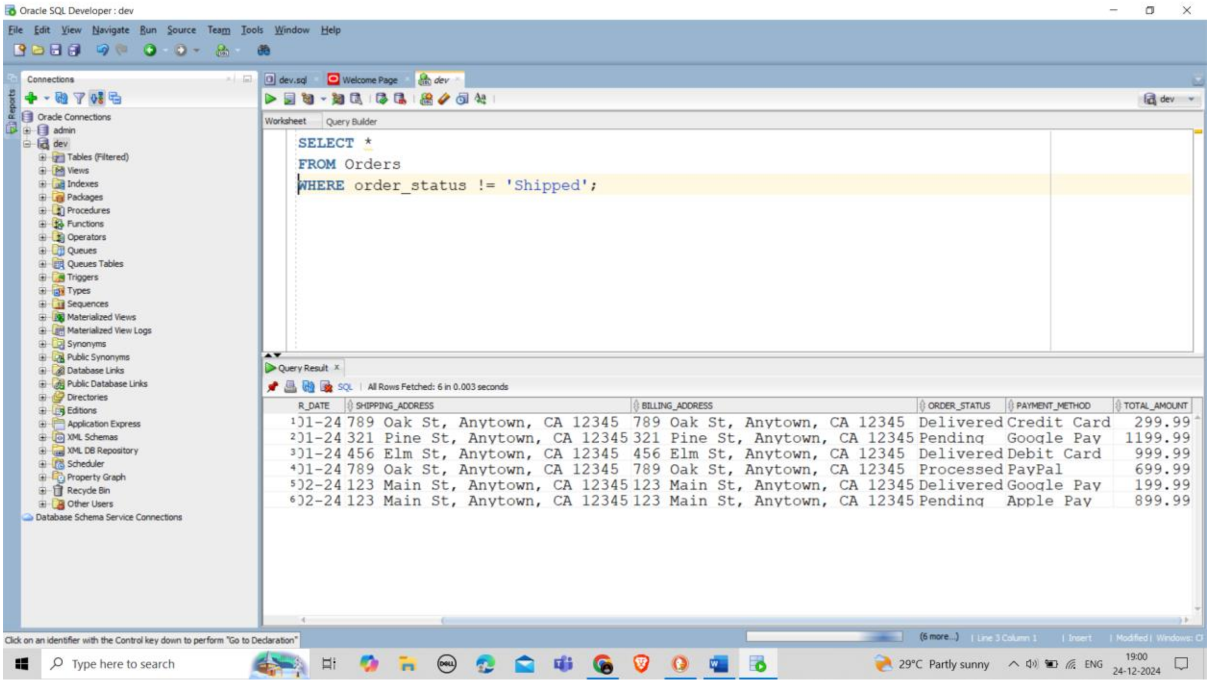
|  | AVERAGE_ORDER_VALUE |
|---|---|
| 1 | 629.99 |

## 11.Find orders that have not been shipped yet:

SELECT *

FROM Orders

WHERE order_status != 'Shipped';



```
SELECT *
FROM Orders
WHERE order_status != 'Shipped';
```

All Rows Fetched: 6 in 0.003 seconds

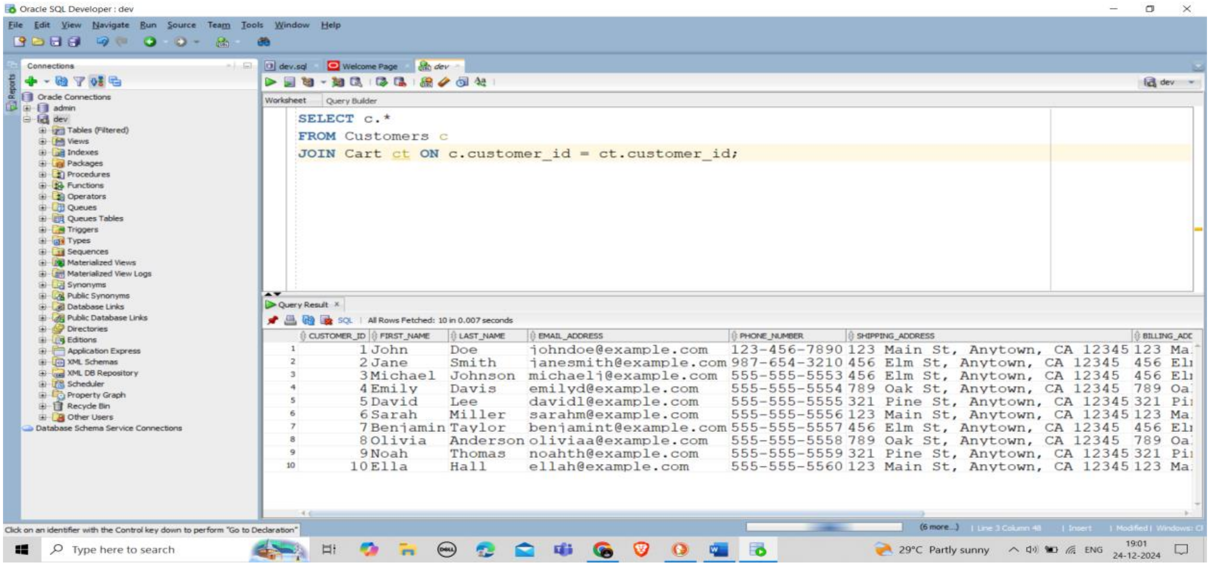
```
  R_DATE  SHIPPING_ADDRESS                 BILLING_ADDRESS                  ORDER_STATUS PAYMENT_METHOD TOTAL_AMOUNT
1 )1-24   789 Oak St, Anytown, CA 12345    789 Oak St, Anytown, CA 12345    Delivered    Credit Card    299.99
2 )1-24   321 Pine St, Anytown, CA 12345   321 Pine St, Anytown, CA 12345   Pending      Google Pay     1199.99
3 )1-24   456 Elm St, Anytown, CA 12345    456 Elm St, Anytown, CA 12345    Delivered    Debit Card     999.99
4 )1-24   789 Oak St, Anytown, CA 12345    789 Oak St, Anytown, CA 12345    Processed    PayPal         699.99
5 )2-24   123 Main St, Anytown, CA 12345   123 Main St, Anytown, CA 12345   Delivered    Google Pay     199.99
6 )2-24   123 Main St, Anytown, CA 12345   123 Main St, Anytown, CA 12345   Pending      Apple Pay      899.99
```

## 12.Find customers with items in their cart:

SELECT c.*

FROM Customers c

JOIN Cart ct ON c.customer_id = ct.customer_id;



```
SELECT c.*
FROM Customers c
JOIN Cart ct ON c.customer_id = ct.customer_id;
```

All Rows Fetched: 10 in 0.007 seconds

```
   CUSTOMER_ID FIRST_NAME LAST_NAME EMAIL_ADDRESS          PHONE_NUMBER SHIPPING_ADDRESS                 BILLING_ADD
1            1 John       Doe       johndoe@example.com    123-456-7890 123 Main St, Anytown, CA 12345   123 Ma
2            2 Jane       Smith     janesmith@example.com  987-654-3210 456 Elm St, Anytown, CA 12345    456 El
3            3 Michael    Johnson   michaelj@example.com   555-555-5553 456 Elm St, Anytown, CA 12345    456 El
4            4 Emily      Davis     emilyd@example.com     555-555-5554 789 Oak St, Anytown, CA 12345    789 Oa
5            5 David      Lee       davidl@example.com     555-555-5555 321 Pine St, Anytown, CA 12345   321 Pi
6            6 Sarah      Miller    sarahm@example.com     555-555-5556 123 Main St, Anytown, CA 12345   123 Ma
7            7 Benjamin   Taylor    benjamint@example.com  555-555-5557 456 Elm St, Anytown, CA 12345    456 El
8            8 Olivia     Anderson  oliviaa@example.com    555-555-5558 789 Oak St, Anytown, CA 12345    789 Oa
9            9 Noah       Thomas    noahth@example.com     555-555-5559 321 Pine St, Anytown, CA 12345   321 Pi
10          10 Ella       Hall      ellah@example.com      555-555-5560 123 Main St, Anytown, CA 12345   123 Ma
```

## 13.Find the total value of items in a customer's cart:

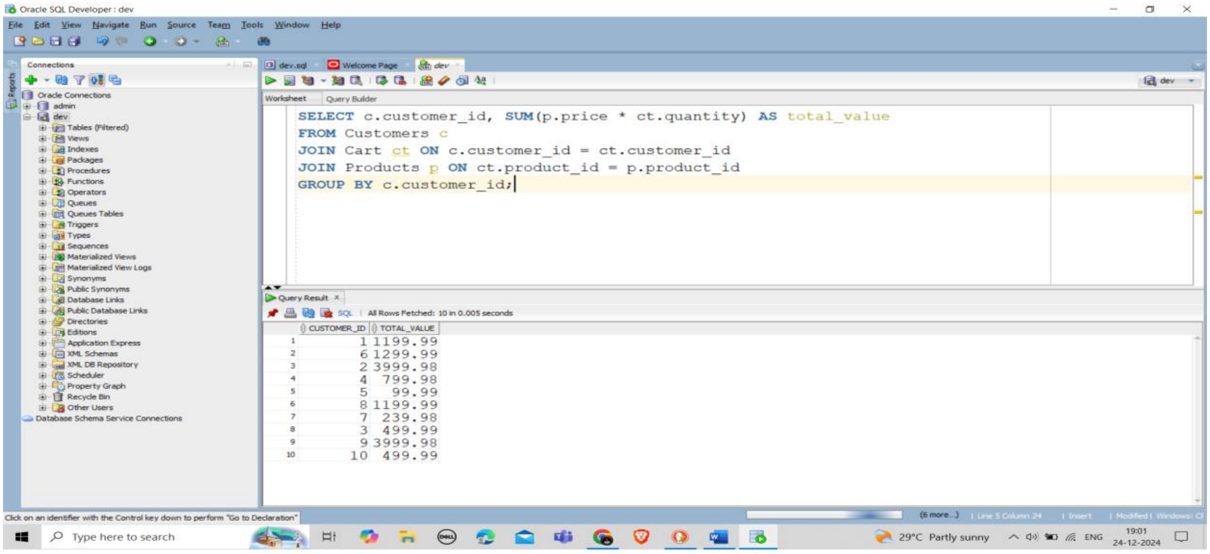
SELECT c.customer_id, SUM(p.price * ct.quantity) AS total_value

FROM Customers c

JOIN Cart ct ON c.customer_id = ct.customer_id

JOIN Products p ON ct.product_id = p.product_id

GROUP BY c.customer_id;



```
SELECT c.customer_id, SUM(p.price * ct.quantity) AS total_value
FROM Customers c
JOIN Cart ct ON c.customer_id = ct.customer_id
JOIN Products p ON ct.product_id = p.product_id
GROUP BY c.customer_id;
```

All Rows Fetched: 10 in 0.005 seconds

```
    CUSTOMER_ID TOTAL_VALUE
1             1 1199.99
2             6 1299.99
3             2 3999.98
4             4  799.98
5             5   99.99
6             8 1199.99
7             7  239.98
8             3  499.99
9             9 3999.98
10           10  499.99
```

## 14.Find the most popular items in carts:

select * from(SELECT p.product_name, SUM(ct.quantity) AS total_quantity

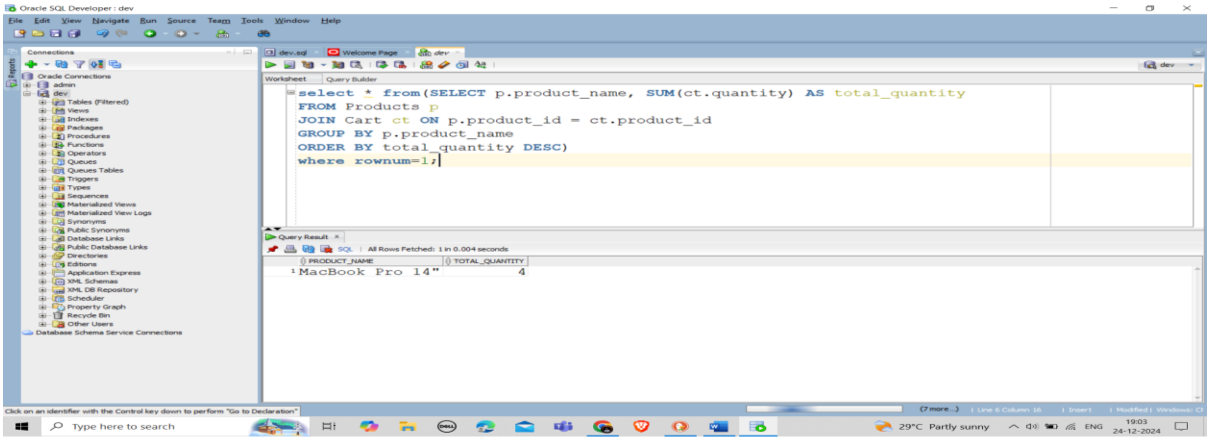
FROM Products p

JOIN Cart ct ON p.product_id = ct.product_id

GROUP BY p.product_name

ORDER BY total_quantity DESC)

where rownum=1;



```
select * from(SELECT p.product_name, SUM(ct.quantity) AS total_quantity
FROM Products p
JOIN Cart ct ON p.product_id = ct.product_id
GROUP BY p.product_name
ORDER BY total_quantity DESC)
where rownum=1;
```

All Rows Fetched: 1 in 0.004 seconds

```
  PRODUCT_NAME    TOTAL_QUANTITY
1 MacBook Pro 14"              4
```

## JOINS

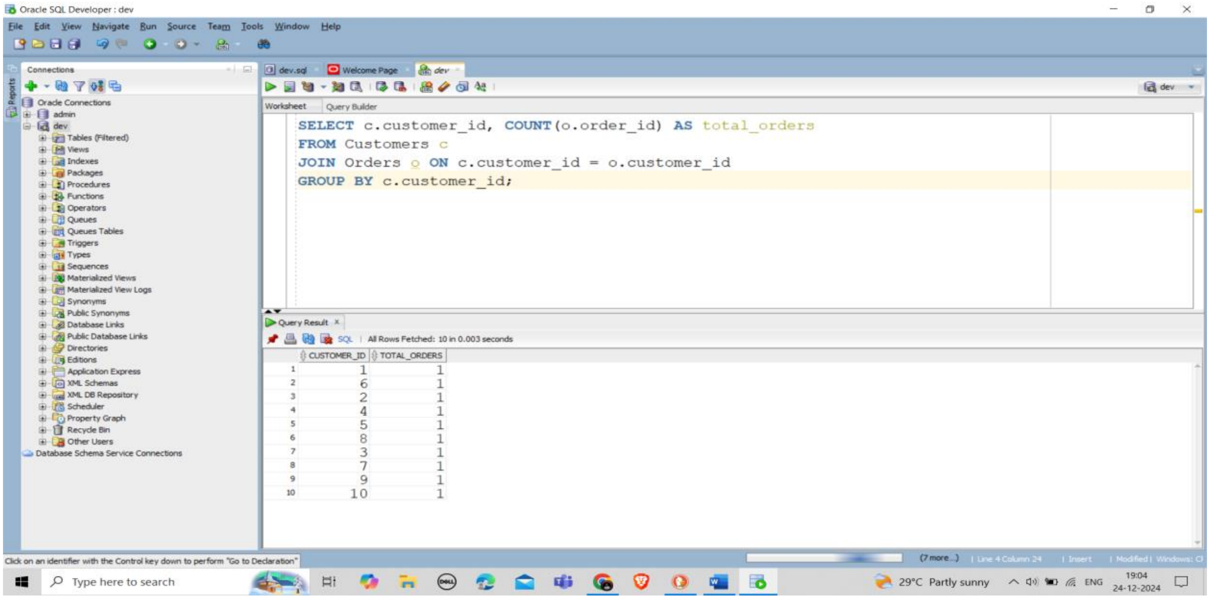
## 1. Find the total number of orders placed by each customer.

SELECT c.customer_id, COUNT(o.order_id) AS total_orders

FROM Customers c

JOIN Orders o ON c.customer_id = o.customer_id

GROUP BY c.customer_id;


Oracle SQL Developer : dev
File Edit View Navigate Run Source Team Tools Window Help
Connections
dev.sql
Welcome Page
dev
Worksheet
Query Builder

```
SELECT c.customer_id, COUNT(o.order_id) AS total_orders
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id;
```

Query Result

All Rows Fetched: 10 in 0.003 seconds

| | CUSTOMER_ID | TOTAL_ORDERS |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 6 | 1 |
| 3 | 2 | 1 |
| 4 | 4 | 1 |
| 5 | 5 | 1 |
| 6 | 8 | 1 |
| 7 | 3 | 1 |
| 8 | 7 | 1 |
| 9 | 9 | 1 |
| 10 | 10 | 1 |

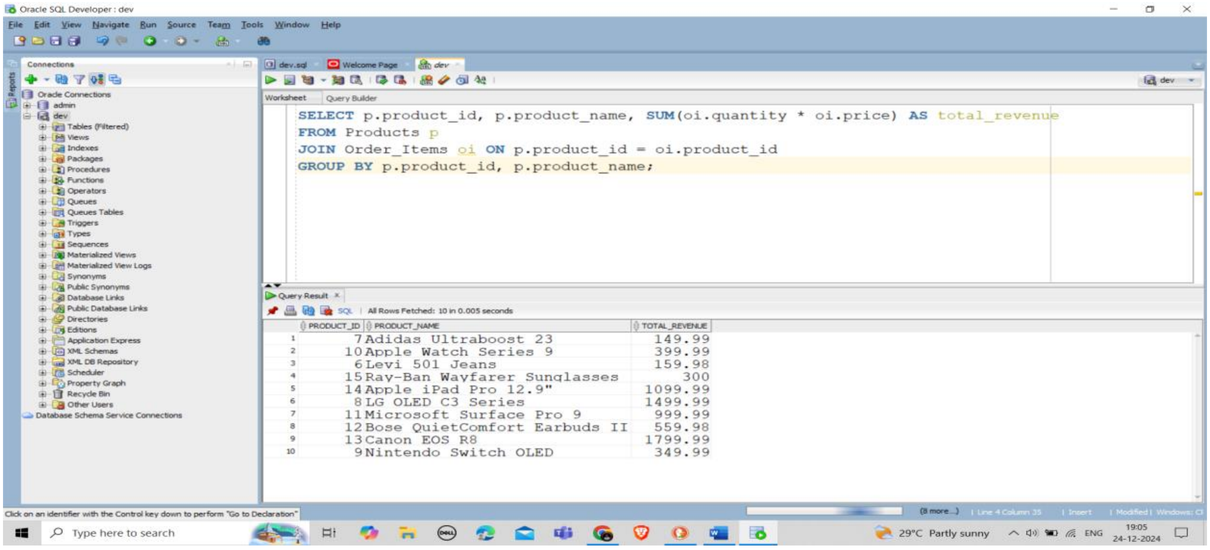
## 2. Find the total revenue generated by each product.

SELECT p.product_id, p.product_name, SUM(oi.quantity * oi.price) AS total_revenue

FROM Products p

JOIN Order_Items oi ON p.product_id = oi.product_id

GROUP BY p.product_id, p.product_name;


Oracle SQL Developer : dev
File Edit View Navigate Run Source Team Tools Window Help
Connections
dev.sql
Welcome Page
dev
Worksheet
Query Builder

```
SELECT p.product_id, p.product_name, SUM(oi.quantity * oi.price) AS total_revenue
FROM Products p
JOIN Order_Items oi ON p.product_id = oi.product_id
GROUP BY p.product_id, p.product_name;
```

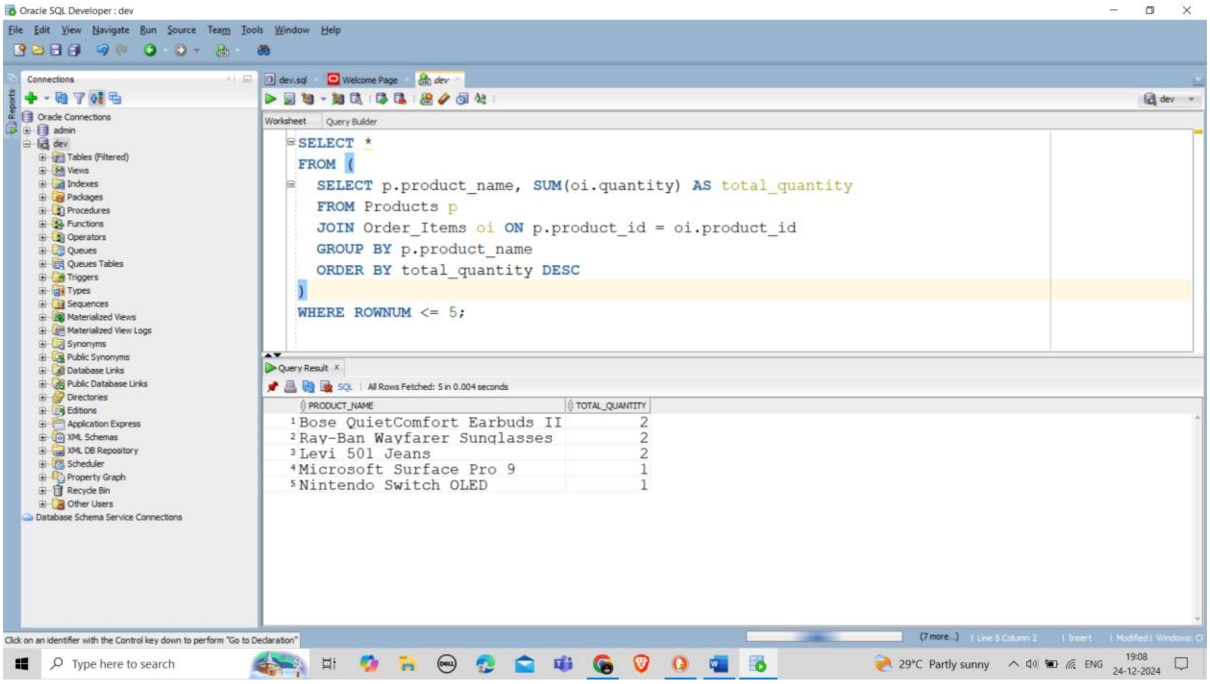
Query Result

All Rows Fetched: 10 in 0.005 seconds

| | PRODUCT_ID | PRODUCT_NAME | TOTAL_REVENUE |
|---|---|---|---|
| 1 | 7 | Adidas Ultraboost 23 | 149.99 |
| 2 | 10 | Apple Watch Series 9 | 399.99 |
| 3 | 6 | Levi 501 Jeans | 159.98 |
| 4 | 15 | Ray-Ban Wayfarer Sunglasses | 300 |
| 5 | 14 | Apple iPad Pro 12.9" | 1099.99 |
| 6 | 8 | LG OLED C3 Series | 1499.99 |
| 7 | 11 | Microsoft Surface Pro 9 | 999.99 |
| 8 | 12 | Bose QuietComfort Earbuds II | 559.98 |
| 9 | 13 | Canon EOS R8 | 1799.99 |
| 10 | 9 | Nintendo Switch OLED | 349.99 |

## 3. List the top 5 best-selling products.

```
SELECT *
FROM (
  SELECT p.product_name, SUM(oi.quantity) AS total_quantity
  FROM Products p
  JOIN Order_Items oi ON p.product_id = oi.product_id
  GROUP BY p.product_name
  ORDER BY total_quantity DESC
)
WHERE ROWNUM <= 5;
```



```
SELECT *
FROM (
  SELECT p.product_name, SUM(oi.quantity) AS total_quantity
  FROM Products p
  JOIN Order_Items oi ON p.product_id = oi.product_id
  GROUP BY p.product_name
  ORDER BY total_quantity DESC
)
WHERE ROWNUM <= 5;
```

All Rows Fetched: 5 in 0.004 seconds

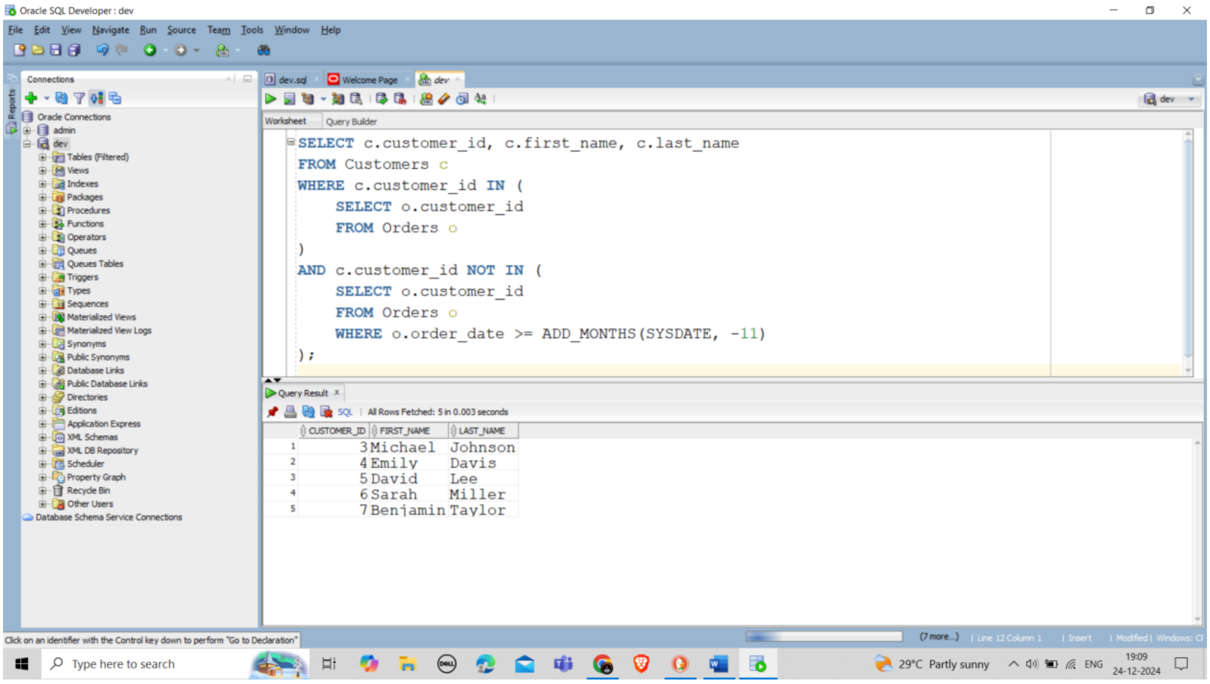
```
   PRODUCT_NAME                     TOTAL_QUANTITY
 1 Bose QuietComfort Earbuds II                  2
 2 Ray-Ban Wayfarer Sunglasses                   2
 3 Levi 501 Jeans                                2
 4 Microsoft Surface Pro 9                       1
 5 Nintendo Switch OLED                          1
```

## 4. Find customers who have placed orders but have not made any purchases in the last 11 months.

```
SELECT c.customer_id, c.first_name, c.last_name
FROM Customers c
WHERE c.customer_id IN (
    SELECT o.customer_id
    FROM Orders o
)
AND c.customer_id NOT IN (
    SELECT o.customer_id
    FROM Orders o
```

WHERE o.order_date >= ADD_MONTHS(SYSDATE, -11)

);



```
SELECT c.customer_id, c.first_name, c.last_name
FROM Customers c
WHERE c.customer_id IN (
    SELECT o.customer_id
    FROM Orders o
)
AND c.customer_id NOT IN (
    SELECT o.customer_id
    FROM Orders o
    WHERE o.order_date >= ADD_MONTHS(SYSDATE, -11)
);
```

Query Result

All Rows Fetched: 5 in 0.003 seconds

|  | CUSTOMER_ID | FIRST_NAME | LAST_NAME |
|---|---|---|---|
| 1 | 3 | Michael | Johnson |
| 2 | 4 | Emily | Davis |
| 3 | 5 | David | Lee |
| 4 | 6 | Sarah | Miller |
| 5 | 7 | Benjamin | Taylor |

## 5. List all products that have not been ordered in the last 12 months.

SQL

SELECT p.product_id, p.product_name

FROM Products p

WHERE p.product_id NOT IN (

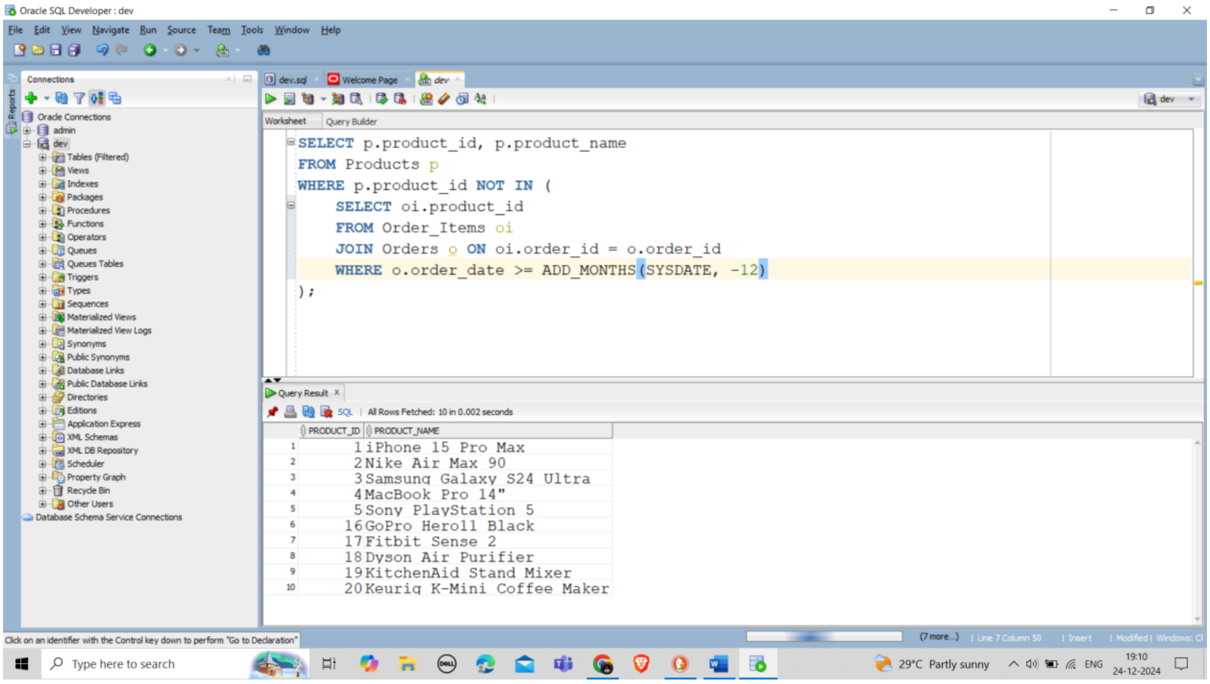
    SELECT oi.product_id

    FROM Order_Items oi

    JOIN Orders o ON oi.order_id = o.order_id

    WHERE o.order_date >= ADD_MONTHS(SYSDATE, -12)

);



```sql
SELECT p.product_id, p.product_name
FROM Products p
WHERE p.product_id NOT IN (
    SELECT oi.product_id
    FROM Order_Items oi
    JOIN Orders o ON oi.order_id = o.order_id
    WHERE o.order_date >= ADD_MONTHS(SYSDATE, -12)
);
```

All Rows Fetched: 10 in 0.002 seconds

|  | PRODUCT_ID | PRODUCT_NAME |
|---|---|---|
| 1 | 1 | iPhone 15 Pro Max |
| 2 | 2 | Nike Air Max 90 |
| 3 | 3 | Samsung Galaxy S24 Ultra |
| 4 | 4 | MacBook Pro 14" |
| 5 | 5 | Sony PlayStation 5 |
| 6 | 16 | GoPro Hero11 Black |
| 7 | 17 | Fitbit Sense 2 |
| 8 | 18 | Dyson Air Purifier |
| 9 | 19 | KitchenAid Stand Mixer |
| 10 | 20 | Keurig K-Mini Coffee Maker |

Click on an identifier with the Control key down to perform "Go to Declaration"

(7 more...) | Line 7 Column 50 | Insert | Modified | Windows: CI