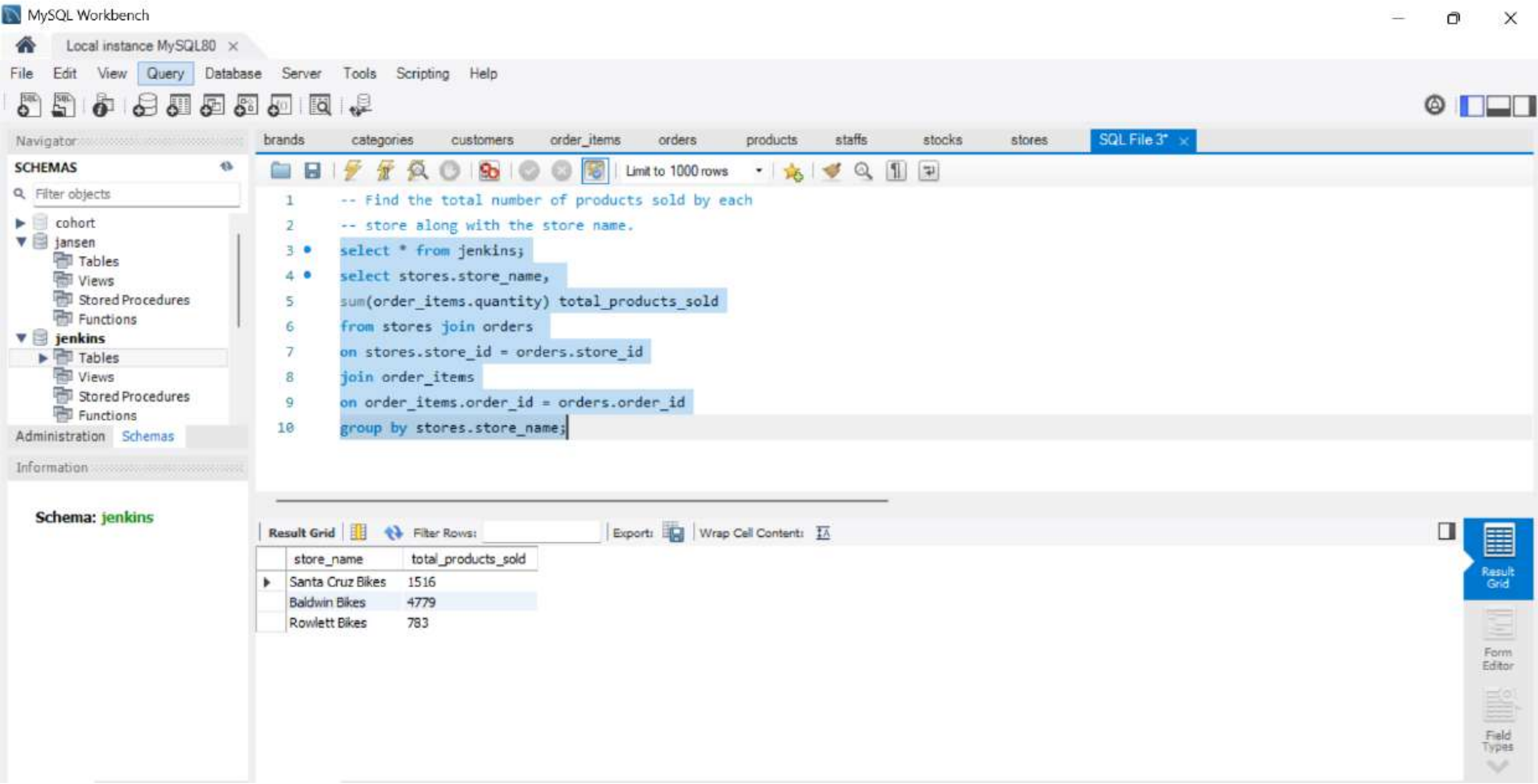
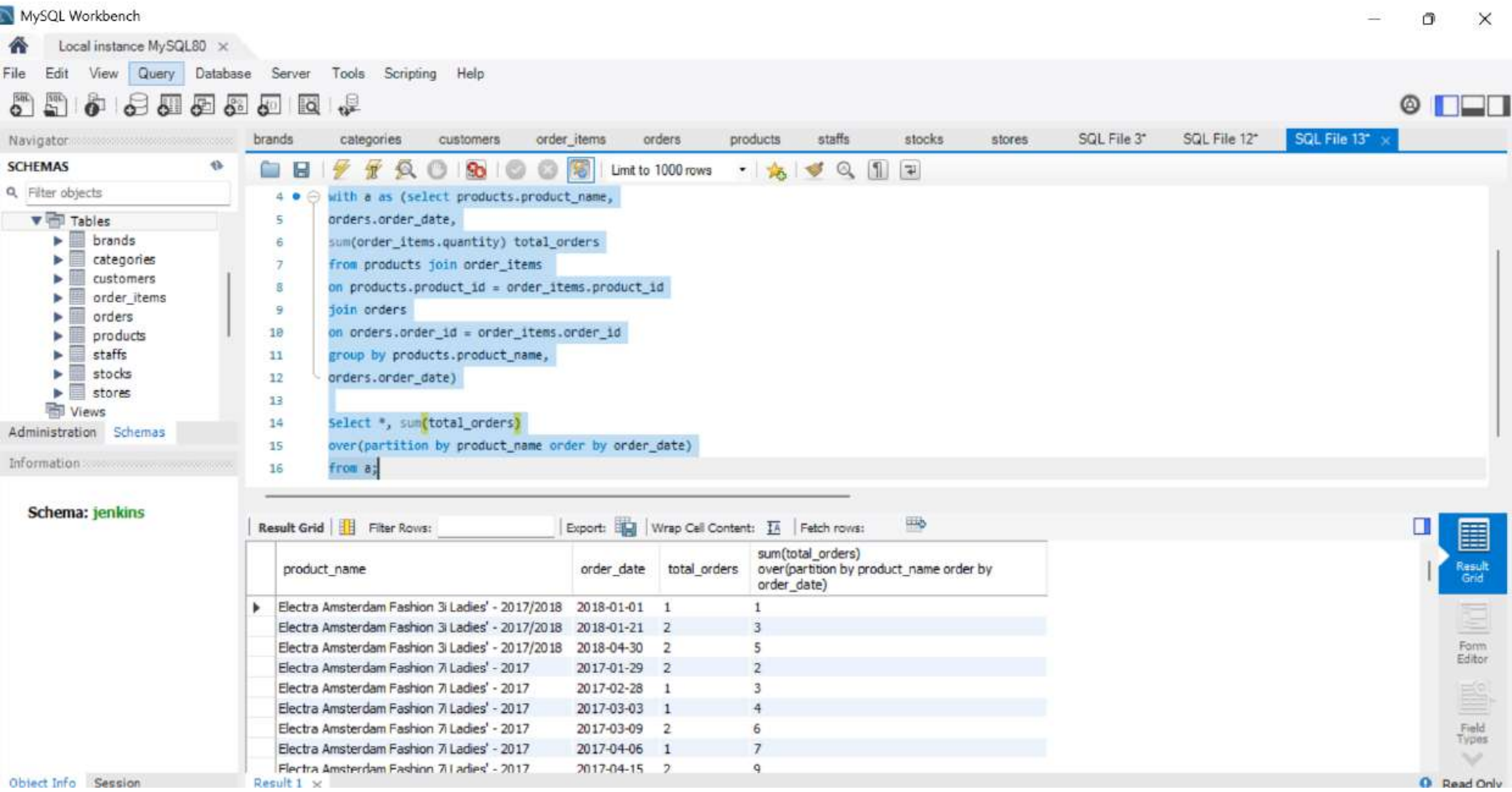


Schema: jenkins

```sql
-- Find the total number of products sold by each
-- store along with the store name.
select * from jenkins;
select stores.store_name,
sum(order_items.quantity) total_products_sold
from stores join orders
on stores.store_id = orders.store_id
join order_items
on order_items.order_id = orders.order_id
group by stores.store_name;
```

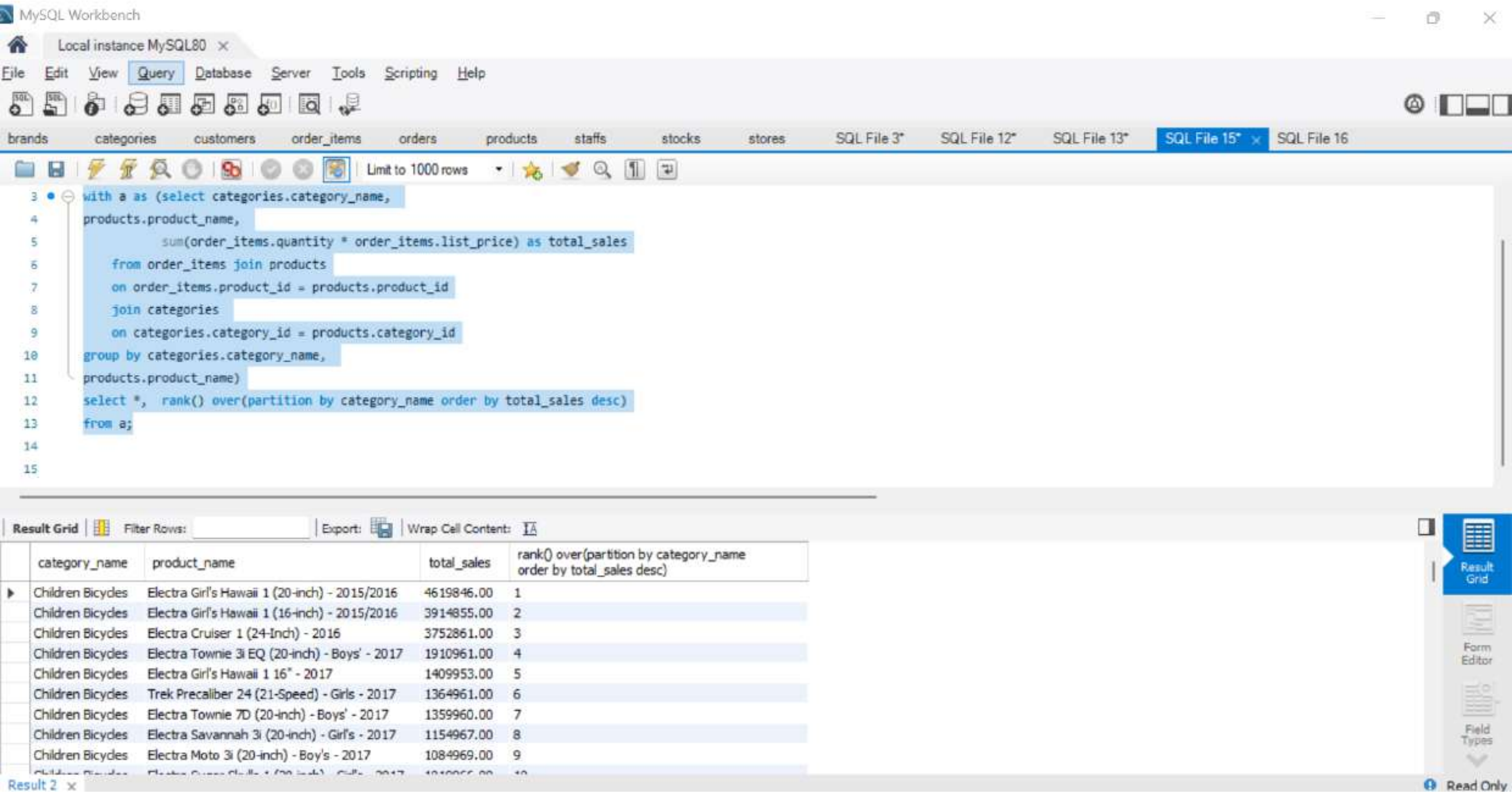
store_name	total_products_sold
Santa Cruz Bikes	1516
Baldwin Bikes	4779
Rowlett Bikes	783



Schema: jenkins

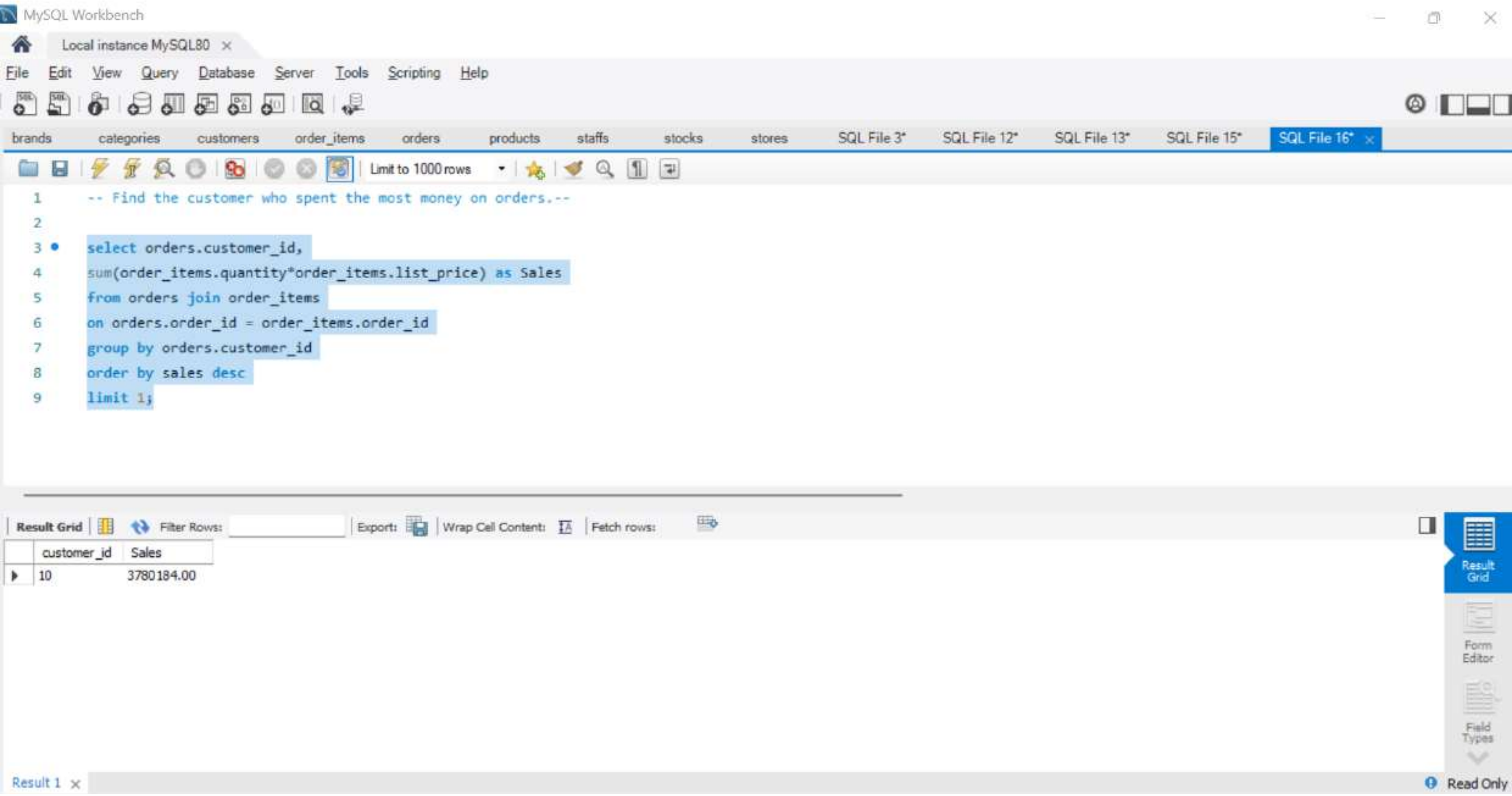
```sql
with a as (select products.product_name,
orders.order_date,
sum(order_items.quantity) total_orders
from products join order_items
on products.product_id = order_items.product_id
join orders
on orders.order_id = order_items.order_id
group by products.product_name,
orders.order_date)

Select *, sum(total_orders)
over(partition by product_name order by order_date)
from a;
```

product_name	order_date	total_orders	sum(total_orders) over(partition by product_name order by order_date)
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-01	1	1
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-21	2	3
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-04-30	2	5
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-01-29	2	2
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-02-28	1	3
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-03	1	4
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-09	2	6
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-06	1	7
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-15	2	9


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
brands categories customers order_items orders products staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16
Limit to 1000 rows
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Result Grid
Form Editor
Field Types
Result 2
Read Only

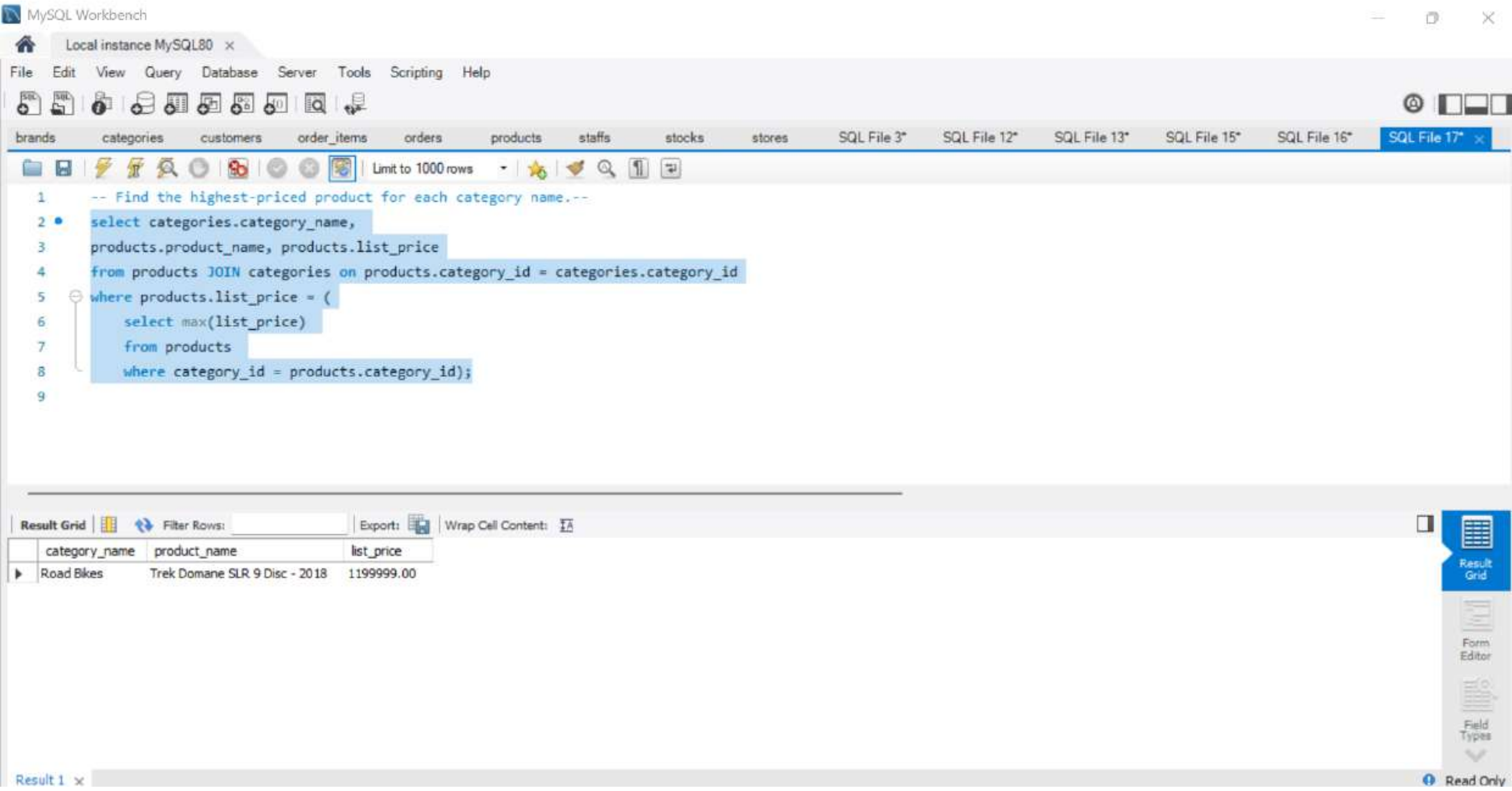
```
with a as (select categories.category_name,
products.product_name,
            sum(order_items.quantity * order_items.list_price) as total_sales
    from order_items join products
    on order_items.product_id = products.product_id
    join categories
    on categories.category_id = products.category_id
group by categories.category_name,
products.product_name)
select *,  rank() over(partition by category_name order by total_sales desc)
from a;
```

category_name	product_name	total_sales	rank() over(partition by category_name order by total_sales desc)
Children Bicycles	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	4619846.00	1
Children Bicycles	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	3914855.00	2
Children Bicycles	Electra Cruiser 1 (24-Inch) - 2016	3752861.00	3
Children Bicycles	Electra Townie 3i EQ (20-inch) - Boys' - 2017	1910961.00	4
Children Bicycles	Electra Girl's Hawaii 1 16" - 2017	1409953.00	5
Children Bicycles	Trek Precaliber 24 (21-Speed) - Girls - 2017	1364961.00	6
Children Bicycles	Electra Townie 7D (20-inch) - Boys' - 2017	1359960.00	7
Children Bicycles	Electra Savannah 3i (20-inch) - Girl's - 2017	1154967.00	8
Children Bicycles	Electra Moto 3i (20-inch) - Boy's - 2017	1084969.00	9
]


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
brands categories customers order_items orders products staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
Result Grid
Form Editor
Field Types
Result 1
Read Only

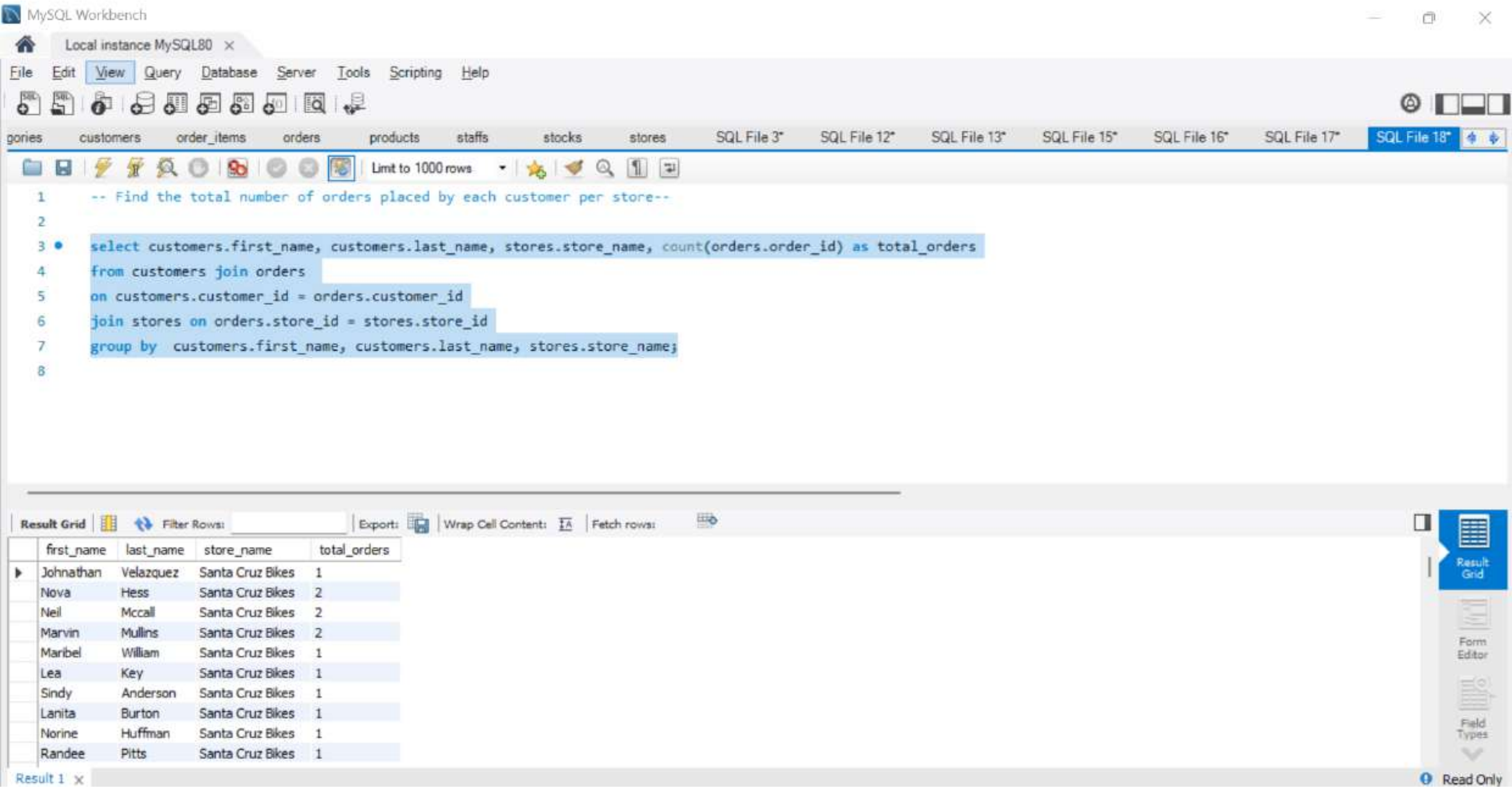
```
-- Find the customer who spent the most money on orders.--

select orders.customer_id,
sum(order_items.quantity*order_items.list_price) as Sales
from orders join order_items
on orders.order_id = order_items.order_id
group by orders.customer_id
order by sales desc
limit 1;
```

customer_id	Sales
10	3780184.00


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
brands categories customers order_items orders products staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16* SQL File 17*
Limit to 1000 rows
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Result Grid
Form Editor
Field Types
Result 1
Read Only

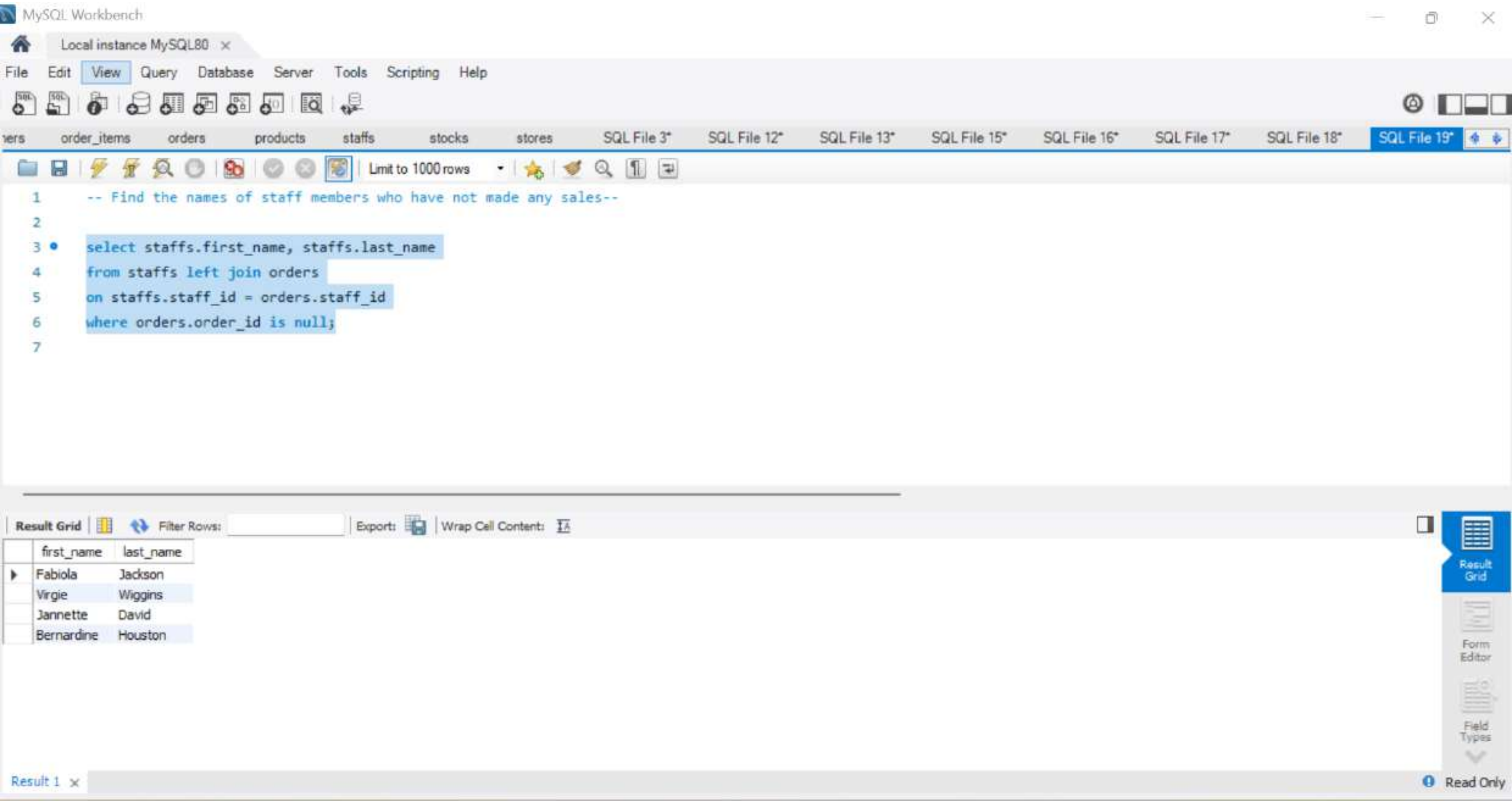
```sql
-- Find the highest-priced product for each category name.--
select categories.category_name,
products.product_name, products.list_price
from products JOIN categories on products.category_id = categories.category_id
where products.list_price = (
    select max(list_price)
    from products
    where category_id = products.category_id);
```

category_name	product_name	list_price
Road Bikes	Trek Domane SLR 9 Disc - 2018	1199999.00


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
gories customers order_items orders products staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16* SQL File 17* SQL File 18*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
Result Grid
Form Editor
Field Types
Result 1
Read Only

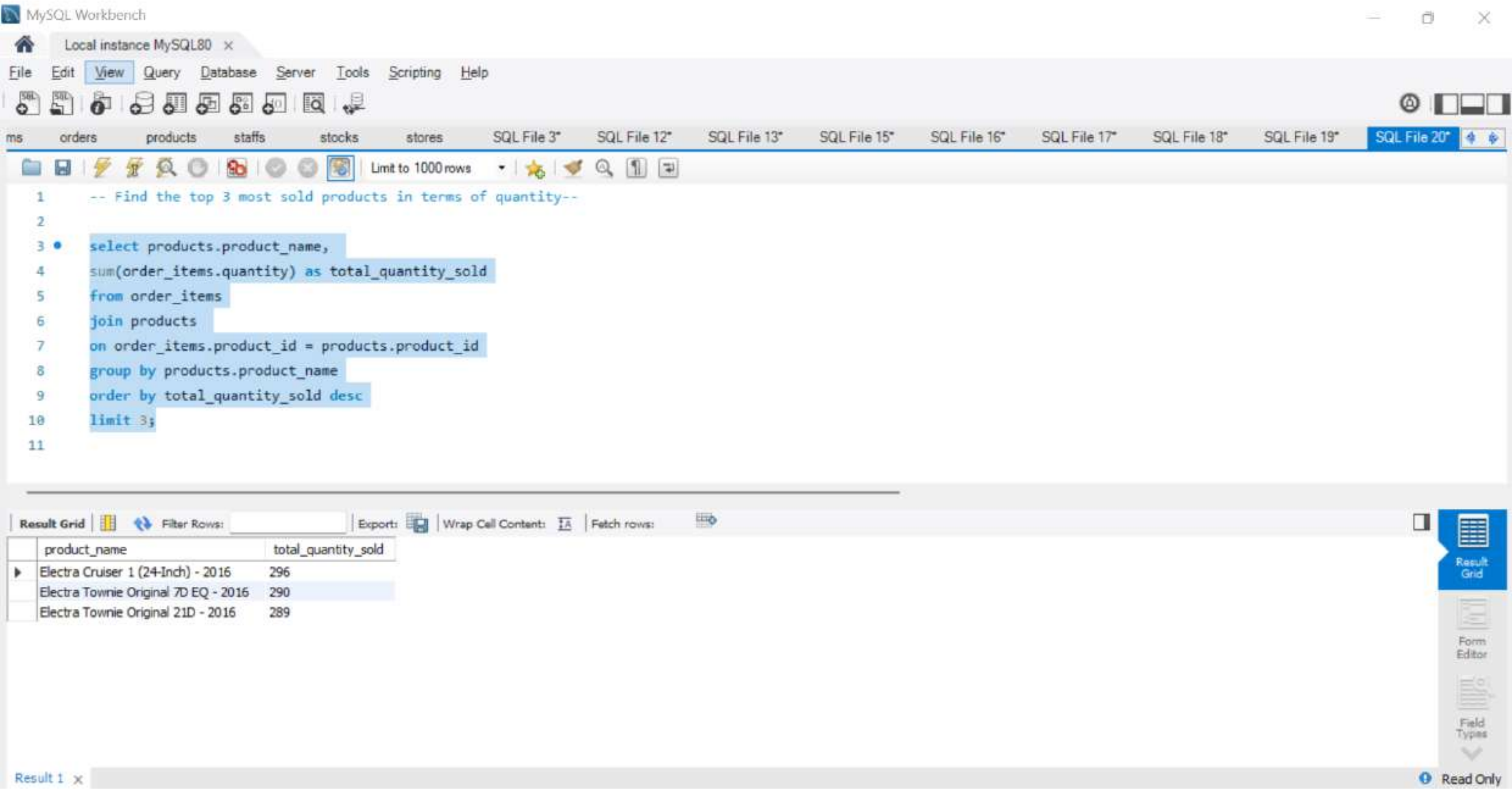
```sql
-- Find the total number of orders placed by each customer per store--

select customers.first_name, customers.last_name, stores.store_name, count(orders.order_id) as total_orders
from customers join orders
on customers.customer_id = orders.customer_id
join stores on orders.store_id = stores.store_id
group by  customers.first_name, customers.last_name, stores.store_name;
```

first_name	last_name	store_name	total_orders
Johnathan	Velazquez	Santa Cruz Bikes	1
Nova	Hess	Santa Cruz Bikes	2
Neil	Mccall	Santa Cruz Bikes	2
Marvin	Mullins	Santa Cruz Bikes	2
Maribel	William	Santa Cruz Bikes	1
Lea	Key	Santa Cruz Bikes	1
Sindy	Anderson	Santa Cruz Bikes	1
Lanita	Burton	Santa Cruz Bikes	1
Norine	Huffman	Santa Cruz Bikes	1
Randee	Pitts	Santa Cruz Bikes	1



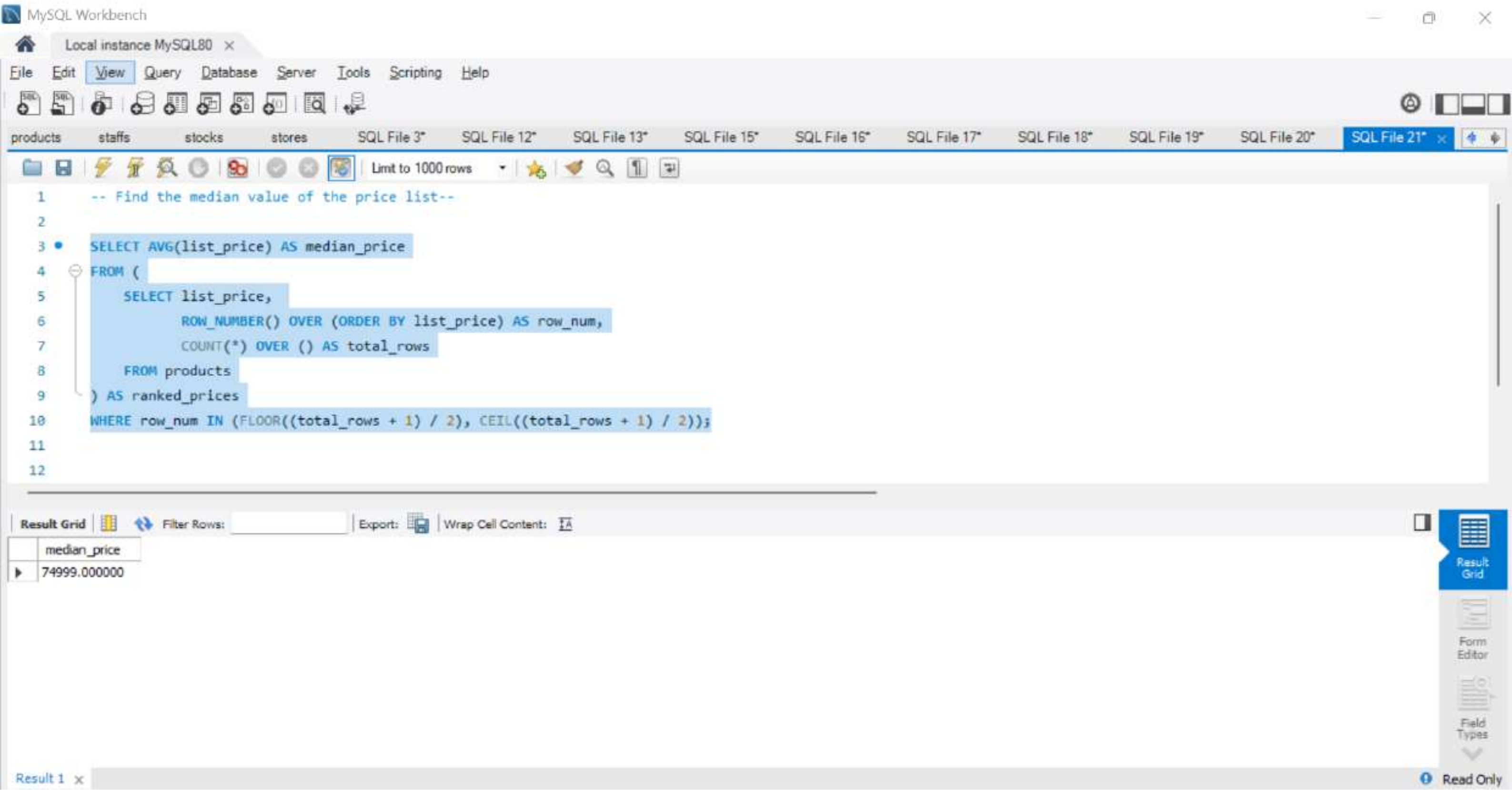
```sql
-- Find the names of staff members who have not made any sales--

select staffs.first_name, staffs.last_name
from staffs left join orders
on staffs.staff_id = orders.staff_id
where orders.order_id is null;
```

first_name	last_name
Fabiola	Jackson
Virgie	Wiggins
Jannette	David
Bernardine	Houston


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
ms orders products staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* SQL File 20*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
Result Grid
Form Editor
Field Types
Result 1
Read Only

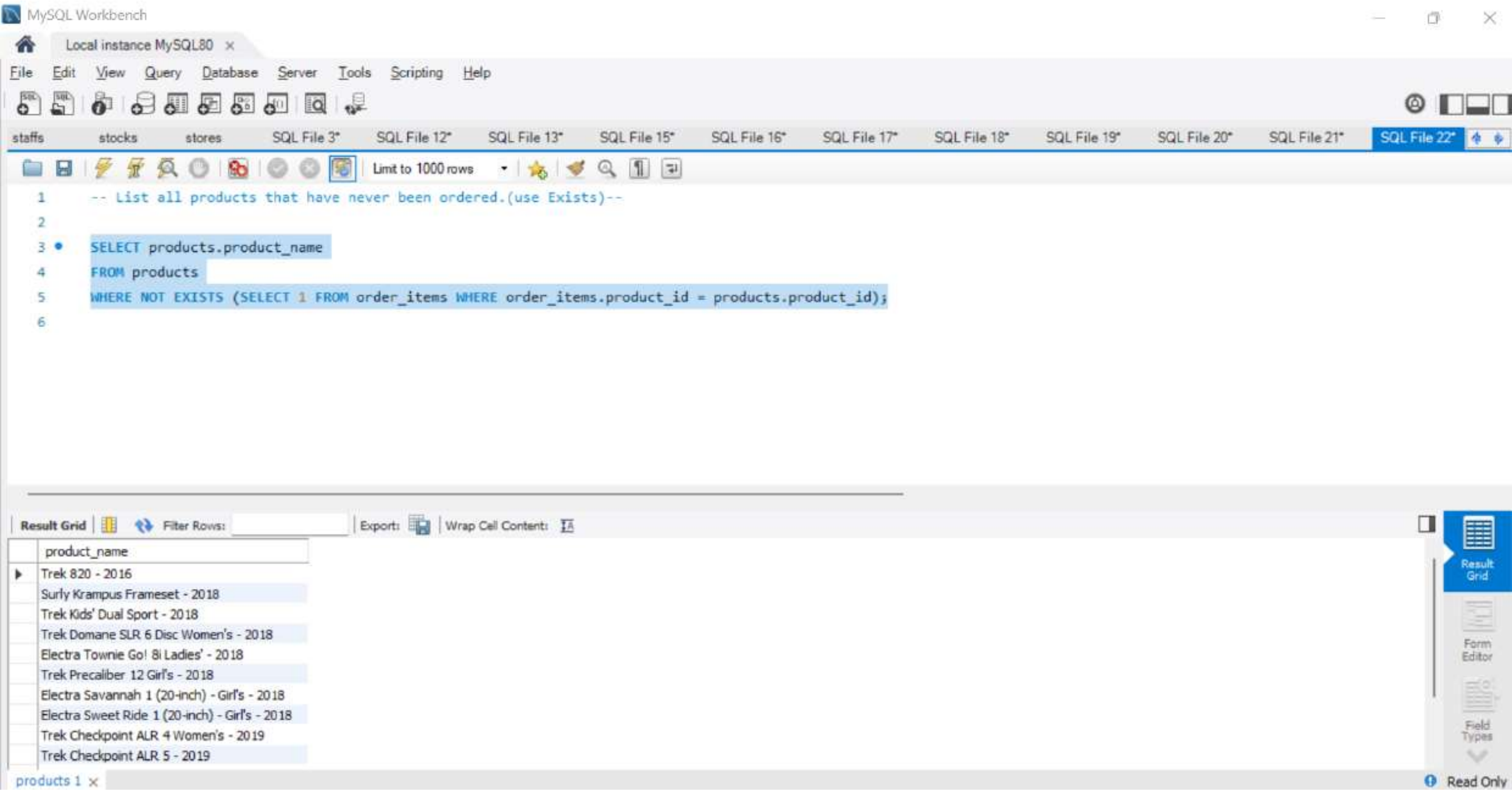
```
-- Find the top 3 most sold products in terms of quantity--

select products.product_name,
sum(order_items.quantity) as total_quantity_sold
from order_items
join products
on order_items.product_id = products.product_id
group by products.product_name
order by total_quantity_sold desc
limit 3;
```

product_name	total_quantity_sold
Electra Cruiser 1 (24-Inch) - 2016	296
Electra Townie Original 7D EQ - 2016	290
Electra Townie Original 21D - 2016	289


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
products staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* SQL File 20* SQL File 21*
Limit to 1000 rows
Result Grid Filter Rows: Export: Wrap Cell Content:
Result Grid
Form Editor
Field Types
Result 1
Read Only

```
-- Find the median value of the price list--

SELECT AVG(list_price) AS median_price
FROM (
    SELECT list_price,
           ROW_NUMBER() OVER (ORDER BY list_price) AS row_num,
           COUNT(*) OVER () AS total_rows
    FROM products
) AS ranked_prices
WHERE row_num IN (FLOOR((total_rows + 1) / 2), CEIL((total_rows + 1) / 2));
```

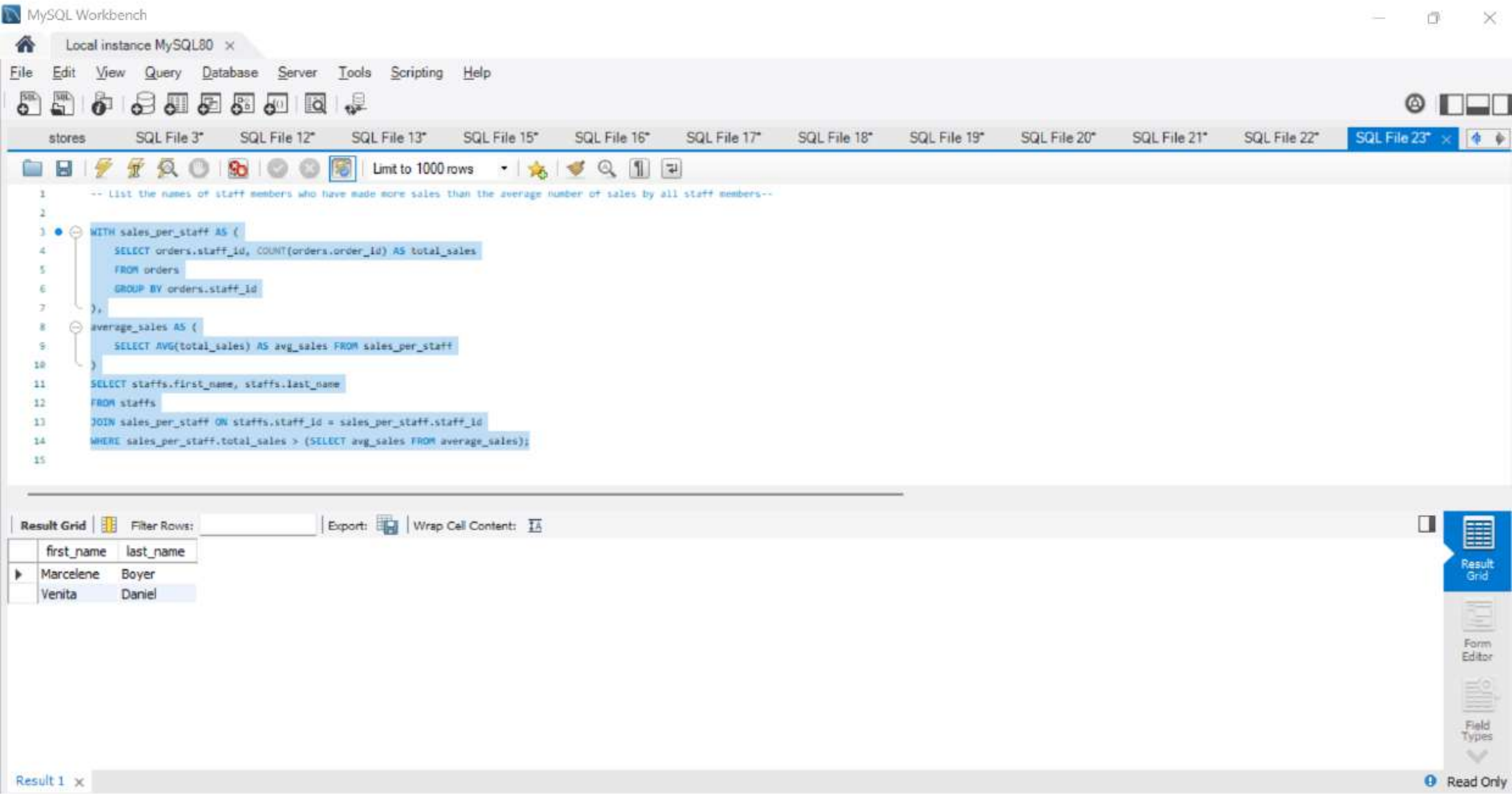
```
median_price
74999.000000
```



MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
staffs stocks stores SQL File 3* SQL File 12* SQL File 13* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* SQL File 20* SQL File 21* SQL File 22*
Limit to 1000 rows
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Result Grid
Form Editor
Field Types
products 1
Read Only

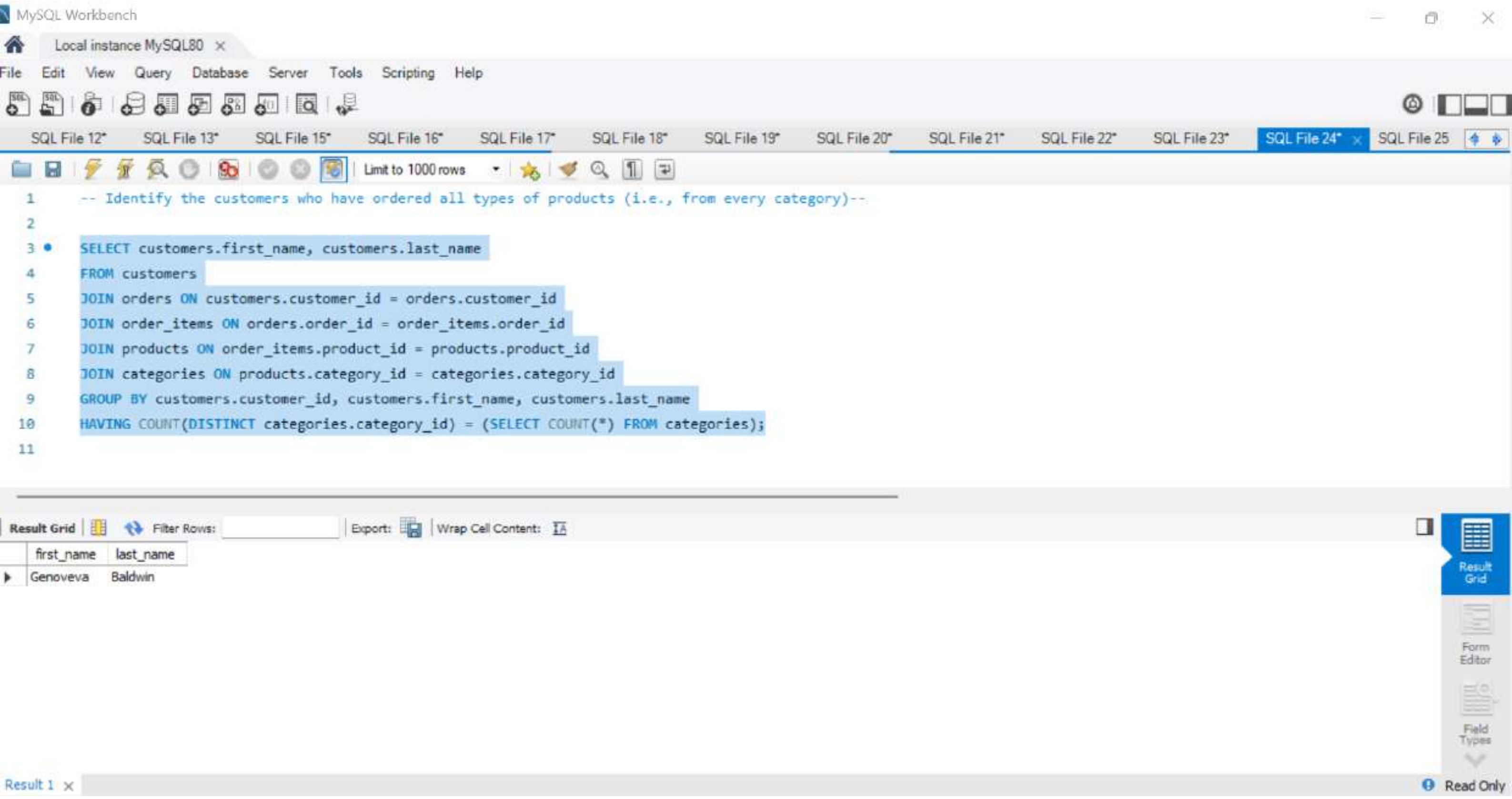
```sql
-- List all products that have never been ordered.(use Exists)--

SELECT products.product_name
FROM products
WHERE NOT EXISTS (SELECT 1 FROM order_items WHERE order_items.product_id = products.product_id);
```

product_name
Trek 820 - 2016
Surly Krampus Frameset - 2018
Trek Kids' Dual Sport - 2018
Trek Domane SLR 6 Disc Women's - 2018
Electra Townie Go! 8i Ladies' - 2018
Trek Precaliber 12 Girl's - 2018
Electra Savannah 1 (20-inch) - Girl's - 2018
Electra Sweet Ride 1 (20-inch) - Girl's - 2018
Trek Checkpoint ALR 4 Women's - 2019
Trek Checkpoint ALR 5 - 2019



```sql
-- List the names of staff members who have made more sales than the average number of sales by all staff members--

WITH sales_per_staff AS (
    SELECT orders.staff_id, COUNT(orders.order_id) AS total_sales
    FROM orders
    GROUP BY orders.staff_id
),
average_sales AS (
    SELECT AVG(total_sales) AS avg_sales FROM sales_per_staff
)
SELECT staffs.first_name, staffs.last_name
FROM staffs
JOIN sales_per_staff ON staffs.staff_id = sales_per_staff.staff_id
WHERE sales_per_staff.total_sales > (SELECT avg_sales FROM average_sales);
```

first_name	last_name
Marcelene	Boyer
Venita	Daniel


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 12* SQL File 13* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* SQL File 20* SQL File 21* SQL File 22* SQL File 23* SQL File 24* SQL File 25
Limit to 1000 rows
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Result Grid
Form Editor
Field Types
Result 1
Read Only

```
-- Identify the customers who have ordered all types of products (i.e., from every category)--

SELECT customers.first_name, customers.last_name
FROM customers
JOIN orders ON customers.customer_id = orders.customer_id
JOIN order_items ON orders.order_id = order_items.order_id
JOIN products ON order_items.product_id = products.product_id
JOIN categories ON products.category_id = categories.category_id
GROUP BY customers.customer_id, customers.first_name, customers.last_name
HAVING COUNT(DISTINCT categories.category_id) = (SELECT COUNT(*) FROM categories);
```

first_name	last_name
Genoveva	Baldwin