

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\* ×

Navigator

Filter objects

SCHEMAS

- cohort
- jansen
  - Tables
  - Views
  - Stored Procedures
  - Functions
- jenkins
  - Tables
  - Views
  - Stored Procedures
  - Functions

Administration Schemas

Information

Schema: jenkins

```
1 -- Find the total number of products sold by each
2 -- store along with the store name.
3 • select * from jenkins;
4 • select stores.store_name,
5 sum(order_items.quantity) total_products_sold
6 from stores join orders
7 on stores.store_id = orders.store_id
8 join order_items
9 on order_items.order_id = orders.order_id
10 group by stores.store_name;
```

Result Grid | Filter Rows: Export | Wrap Cell Content:

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

Result Grid

Form Editor

Field Types

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Query' is selected. The main area displays a SQL query in the 'SQL File 3\*' tab:

```
1 -- Find the total number of products sold by each
2 -- store along with the store name.
3 • select * from jenkins;
4 • select stores.store_name,
5 sum(order_items.quantity) total_products_sold
6 from stores join orders
7 on stores.store_id = orders.store_id
8 join order_items
9 on order_items.order_id = orders.order_id
10 group by stores.store_name;
```

The results of this query are shown in a 'Result Grid' below:

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

The 'Schemas' pane on the left shows the 'jenkins' schema selected, containing tables for 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Results' pane on the right shows the execution context.

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\* ×

Navigator

SCHEMAS

Filter objects

Tables

- brands
- categories
- customers
- order\_items
- orders
- products
- staffs
- stocks
- stores

Views

Administration Schemas

Information

Schema: jenkins

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

```
4 with a as (select products.product_name,
5     orders.order_date,
6     sum(order_items.quantity) total_orders
7     from products join order_items
8     on products.product_id = order_items.product_id
9     join orders
10    on orders.order_id = order_items.order_id
11    group by products.product_name,
12        orders.order_date)
13
14 select *, sum(total_orders)
15 over(partition by product_name order by order_date)
16 from a;
```

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

Object Info Session Result 1 × Read Only

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

```

3 with a as (select categories.category_name,
4   products.product_name,
5     sum(order_items.quantity * order_items.list_price) as total_sales
6   from order_items join products
7   on order_items.product_id = products.product_id
8   join categories
9   on categories.category_id = products.category_id
10  group by categories.category_name,
11    products.product_name)
12 select *, rank() over(partition by category_name order by total_sales desc)
13 from a;
14
15

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only

| 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   |
| Children Bicycles | Electra Go! 20 (20-inch) - Boys' - 2017       | 1010000.00  | 10  |

Result 2 ×

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\* ×

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\* ×

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

2

3 • select orders.customer\_id,  
4 sum(order\_items.quantity\*order\_items.list\_price) as Sales  
5 from orders join order\_items  
6 on orders.order\_id = order\_items.order\_id  
7 group by orders.customer\_id  
8 order by sales desc  
9 limit 1;

Result Grid | Filter Rows: \_\_\_\_\_ | Export: | Wrap Cell Content: | Fetch rows: |

| customer_id | Sales      |
|-------------|------------|
| 10          | 3780184.00 |

Result 1 ×

Result Grid

Form Editor

Field Types

Read Only

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\* ×

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\* ×

```
1 -- Find the highest-priced product for each category name.--
2 • select categories.category_name,
3 products.product_name, products.list_price
4 from products JOIN categories on products.category_id = categories.category_id
5 where products.list_price = (
6     select max(list_price)
7     from products
8     where category_id = products.category_id);
9
```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

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

Result 1 ×

Result Grid Form Editor Field Types Read Only

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

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\* SQL File 18\* SQL File 19\*

Limit to 1000 rows

```
1 -- Find the total number of orders placed by each customer per store--  
2  
3 • select customers.first_name, customers.last_name, stores.store_name, count(orders.order_id) as total_orders  
4 from customers join orders  
5 on customers.customer_id = orders.customer_id  
6 join stores on orders.store_id = stores.store_id  
7 group by customers.first_name, customers.last_name, stores.store_name;  
8
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid Form Editor Field Types Read Only

| 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            |

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

vers 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\* SQL File 19\*

1 -- Find the names of staff members who have not made any sales--  
2  
3 • select staffs.first\_name, staffs.last\_name  
4 from staffs left join orders  
5 on staffs.staff\_id = orders.staff\_id  
6 where orders.order\_id is null;  
7

Result Grid | Filter Rows: Export: Wrap Cell Content:

| first_name | last_name |
|------------|-----------|
| Fabiola    | Jackson   |
| Virgie     | Wiggins   |
| Jannette   | David     |
| Bernardine | Houston   |

Result 1 × Read Only

Result Grid Form Editor Field Types

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\* SQL File 21\*

1 -- Find the top 3 most sold products in terms of quantity--  
2  
3 • select products.product\_name,  
4 sum(order\_items.quantity) as total\_quantity\_sold  
5 from order\_items  
6 join products  
7 on order\_items.product\_id = products.product\_id  
8 group by products.product\_name  
9 order by total\_quantity\_sold desc  
10 limit 3;  
11

---

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

| 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                 |

Result 1 ×

Result Grid Form Editor Field Types Read Only

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\* SQL File 22\*

Limit to 1000 rows

```
1 -- Find the median value of the price list--
2
3 • SELECT AVG(list_price) AS median_price
4 FROM (
5     SELECT list_price,
6            ROW_NUMBER() OVER (ORDER BY list_price) AS row_num,
7            COUNT(*) OVER () AS total_rows
8     FROM products
9 ) AS ranked_prices
10 WHERE row_num IN (FLOOR((total_rows + 1) / 2), CEIL((total_rows + 1) / 2));
11
12
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Field Types Read Only

| median_price |
|--------------|
| 74999.000000 |

Result 1 ×

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\* SQL File 23\*

Limit to 1000 rows

```
1 -- List all products that have never been ordered.(use Exists)--  
2  
3 • SELECT products.product_name  
4 FROM products  
5 WHERE NOT EXISTS (SELECT 1 FROM order_items WHERE order_items.product_id = products.product_id);  
6
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

| 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                   |

products 1 ×

Result Grid Form Editor Field Types Read Only

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

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\* SQL File 23\* SQL File 24\*

1 -- List the names of staff members who have made more sales than the average number of sales by all staff members--  
2  
3 WITH sales\_per\_staff AS (  
4 SELECT orders.staff\_id, COUNT(orders.order\_id) AS total\_sales  
5 FROM orders  
6 GROUP BY orders.staff\_id  
7 ),  
8 average\_sales AS (  
9 SELECT AVG(total\_sales) AS avg\_sales FROM sales\_per\_staff  
10 )  
11 SELECT staffs.first\_name, staffs.last\_name  
12 FROM staffs  
13 JOIN sales\_per\_staff ON staffs.staff\_id = sales\_per\_staff.staff\_id  
14 WHERE sales\_per\_staff.total\_sales > (SELECT avg\_sales FROM average\_sales);  
15

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

| first_name | last_name |
|------------|-----------|
| Marcelene  | Boyer     |
| Venita     | Daniel    |

Result 1 ×

Result Grid Form Editor Field Types Read Only

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

```
1 -- Identify the customers who have ordered all types of products (i.e., from every category)--
2
3 • SELECT customers.first_name, customers.last_name
4   FROM customers
5     JOIN orders ON customers.customer_id = orders.customer_id
6     JOIN order_items ON orders.order_id = order_items.order_id
7     JOIN products ON order_items.product_id = products.product_id
8     JOIN categories ON products.category_id = categories.category_id
9   GROUP BY customers.customer_id, customers.first_name, customers.last_name
10  HAVING COUNT(DISTINCT categories.category_id) = (SELECT COUNT(*) FROM categories);
11
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

| first_name | last_name |
|------------|-----------|
| Genoveva   | Baldwin   |

Form Editor

Field Types

Read Only