Inventory Management System

By-Pranali Bhosale

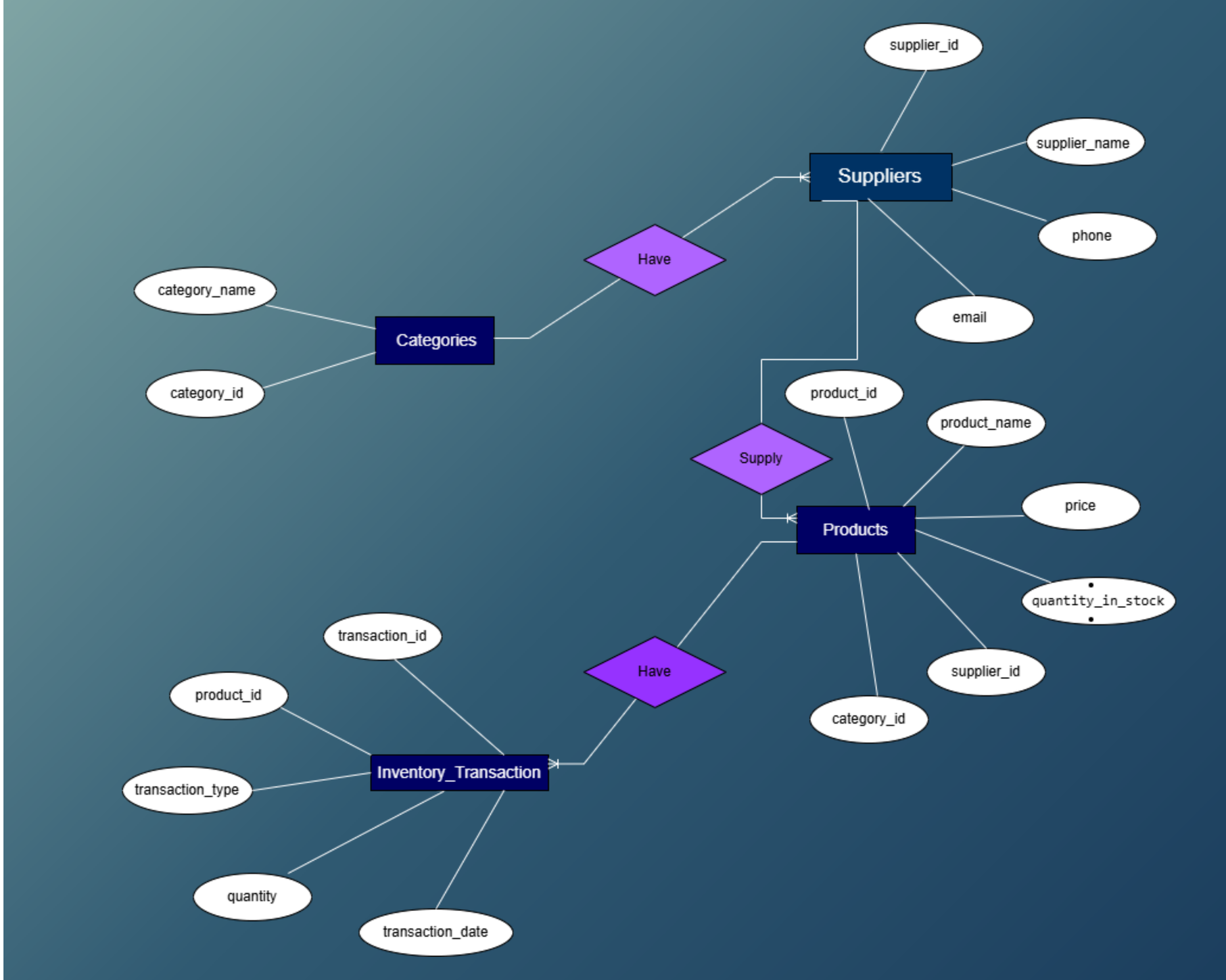
Introduction:

An Inventory Management System (IMS) is a software solution designed to track, manage, and optimize inventory levels across a business's supply chain. It provides a centralized platform to oversee the flow of goods, from raw materials to finished products, across storage locations and distribution channels.

Purpose :

An Inventory Management System (IMS) helps businesses keep optimal stock levels, reducing costs and preventing overstock or stockouts. It tracks inventory in real-time, automates order processes, and supports demand forecasting. IMS improves decision-making with data-driven insights, enhancing efficiency. By streamlining inventory control, it boosts customer satisfaction and operational effectiveness.

ER Diagram


supplier_id
supplier_name
Suppliers
phone
Have
email
category_name
Categories
category_id
product_id
product_name
Supply
price
Products
quantity_in_stock
transaction_id
Have
supplier_id
product_id
category_id
Inventory_Transaction
transaction_type
quantity
transaction_date

Tables Created :

1 .Categories

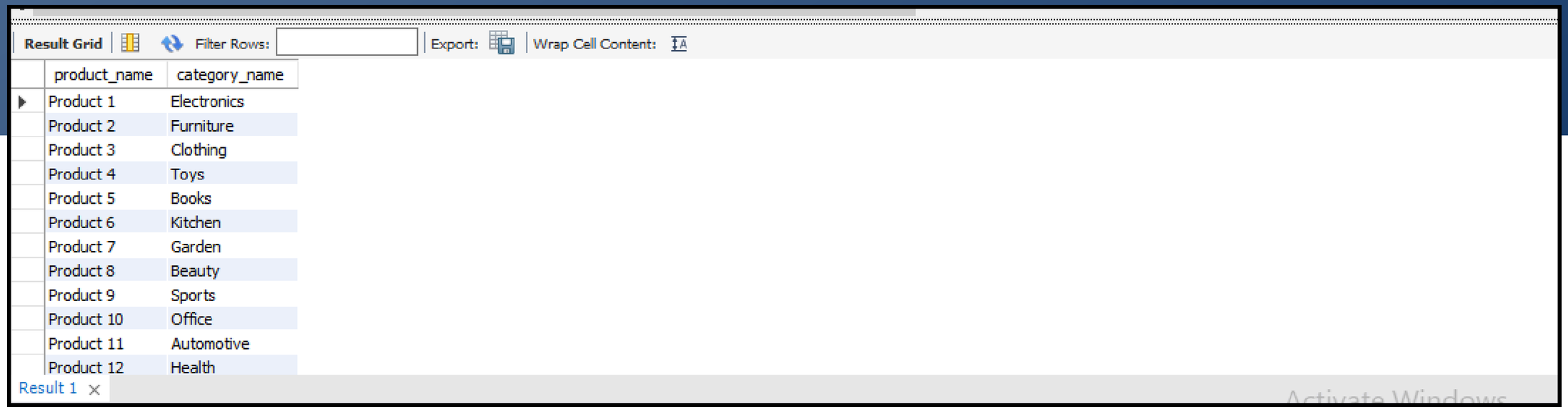
2 . Suppliers

3 . Products

4 . Inventory_Transaction

1. Retrieve all product names and their categories.

```
SELECT product_name, category_name
FROM Products
JOIN Categories ON Products.category_id = Categories.category_id;
```



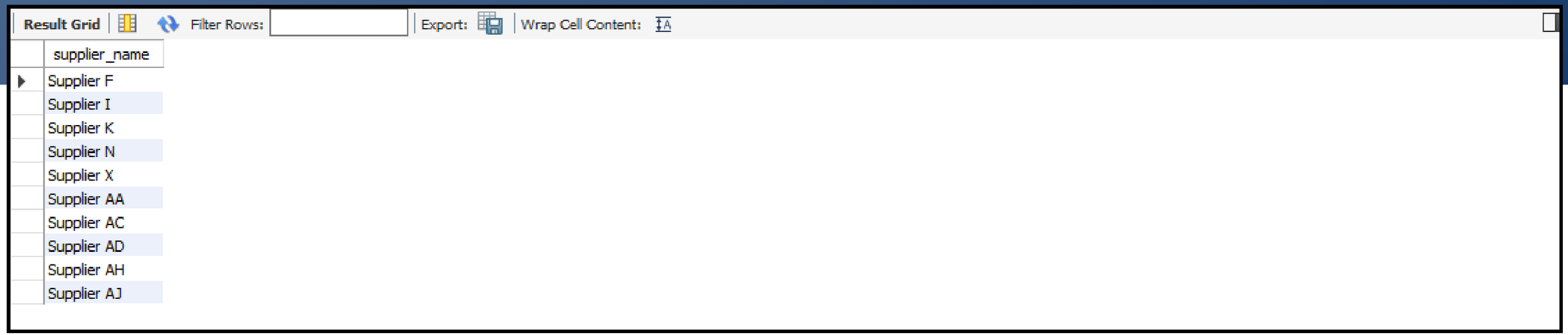
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	category_name
Product 1	Electronics
Product 2	Furniture
Product 3	Clothing
Product 4	Toys
Product 5	Books
Product 6	Kitchen
Product 7	Garden
Product 8	Beauty
Product 9	Sports
Product 10	Office
Product 11	Automotive
Product 12	Health

Result 1

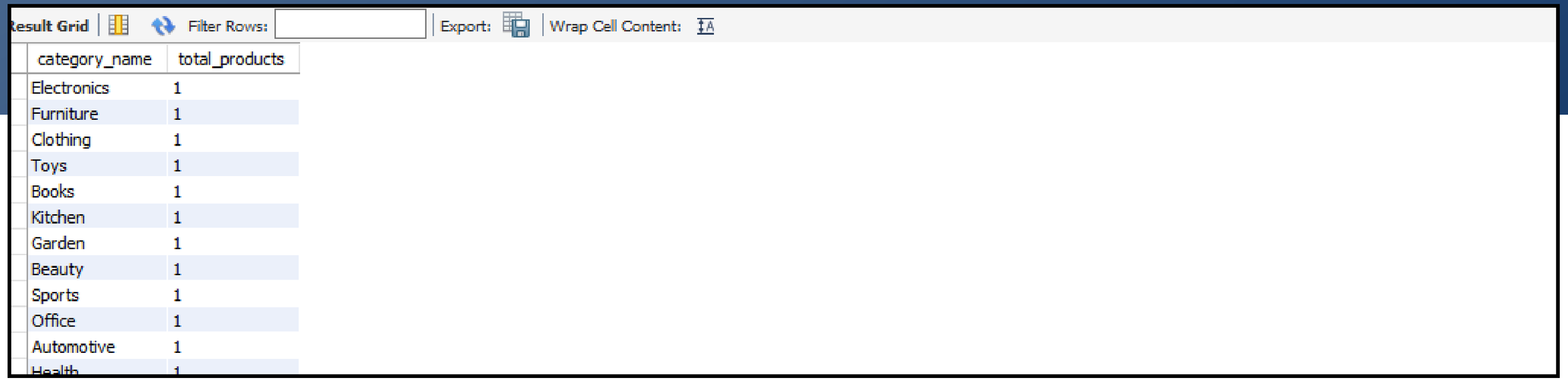
2. List suppliers who provide products with a quantity in stock of less than 100.

```
SELECT supplier_name
FROM Suppliers
JOIN Products ON Suppliers.supplier_id = Products.supplier_id
WHERE Products.quantity_in_stock < 100;
```



3. Find the total number of products in each category.

```
SELECT category_name, COUNT(*) AS total_products
FROM Products
JOIN Categories ON Products.category_id = Categories.category_id
GROUP BY category_name;
```

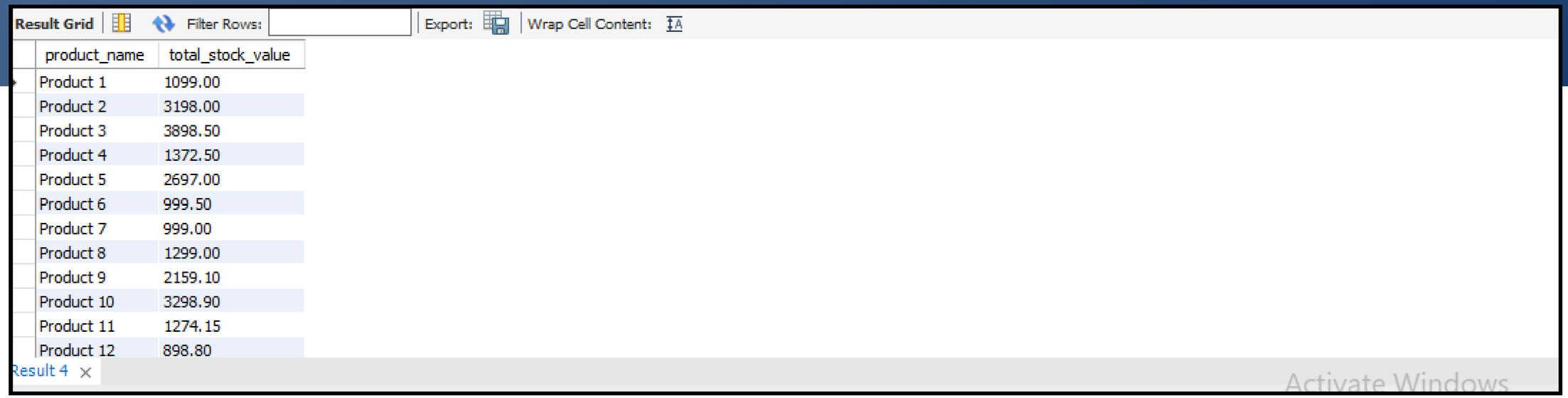


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category_name	total_products
Electronics	1
Furniture	1
Clothing	1
Toys	1
Books	1
Kitchen	1
Garden	1
Beauty	1
Sports	1
Office	1
Automotive	1
Health	1

4. Calculate the total stock value (quantity * price) for each product.

```
SELECT product_name, (price * quantity_in_stock) AS
total_stock_value
FROM Products;
```



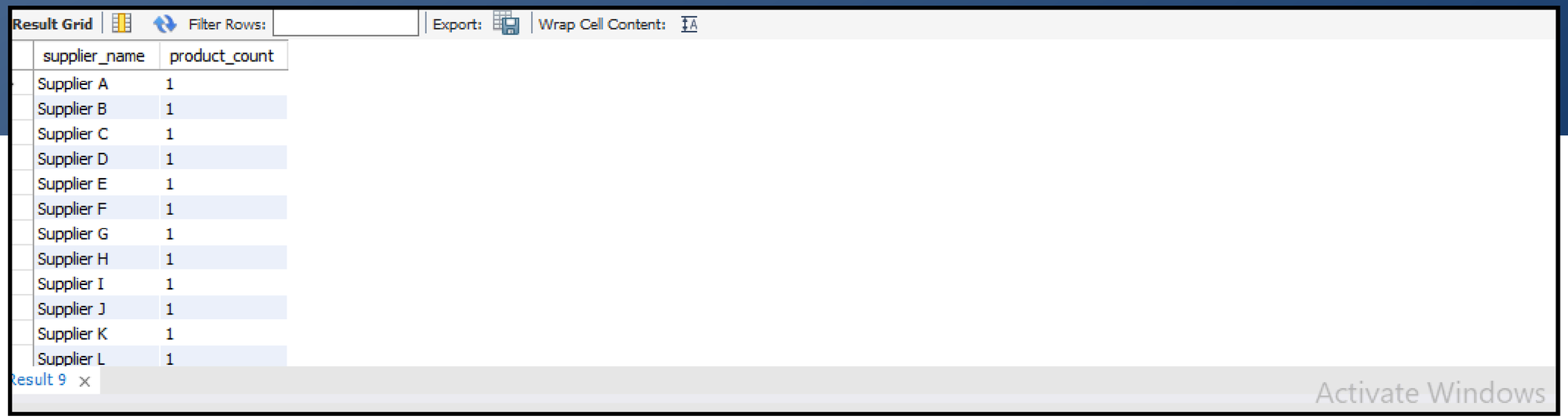
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	total_stock_value
Product 1	1099.00
Product 2	3198.00
Product 3	3898.50
Product 4	1372.50
Product 5	2697.00
Product 6	999.50
Product 7	999.00
Product 8	1299.00
Product 9	2159.10
Product 10	3298.90
Product 11	1274.15
Product 12	898.80

Result 4

5. Get the total number of products supplied by each supplier.

```
SELECT supplier_name, COUNT(*) AS product_count
FROM Products
JOIN Suppliers ON Products.supplier_id = Suppliers.supplier_id
GROUP BY supplier_name;
```



6. Retrieve the latest 10 transactions, sorted by date.

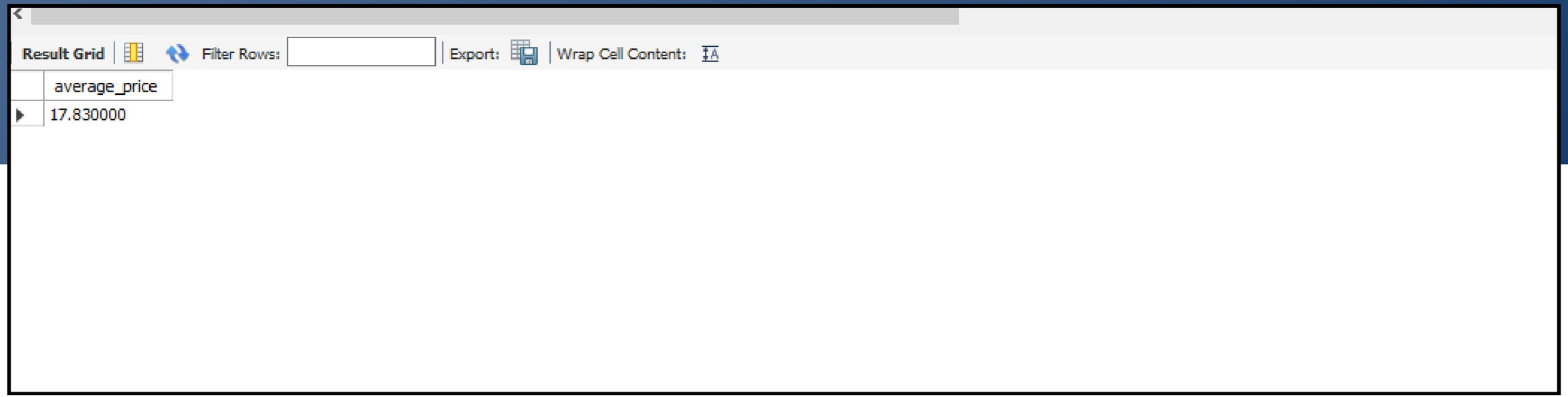
```
SELECT *
FROM Inventory_Transaction
ORDER BY transaction_date DESC
LIMIT 10;
```

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

transaction_id	product_id	transaction_type	quantity	transaction_date
1	1	purchase	10	2024-11-07 11:40:31
2	2	sale	5	2024-11-07 11:40:31
3	3	purchase	15	2024-11-07 11:40:31
4	4	sale	10	2024-11-07 11:40:31
5	5	purchase	20	2024-11-07 11:40:31
6	6	sale	5	2024-11-07 11:40:31
7	7	purchase	10	2024-11-07 11:40:31
8	8	sale	8	2024-11-07 11:40:31
9	9	purchase	10	2024-11-07 11:40:31
10	10	sale	4	2024-11-07 11:40:31
NULL	NULL	NULL	NULL	NULL

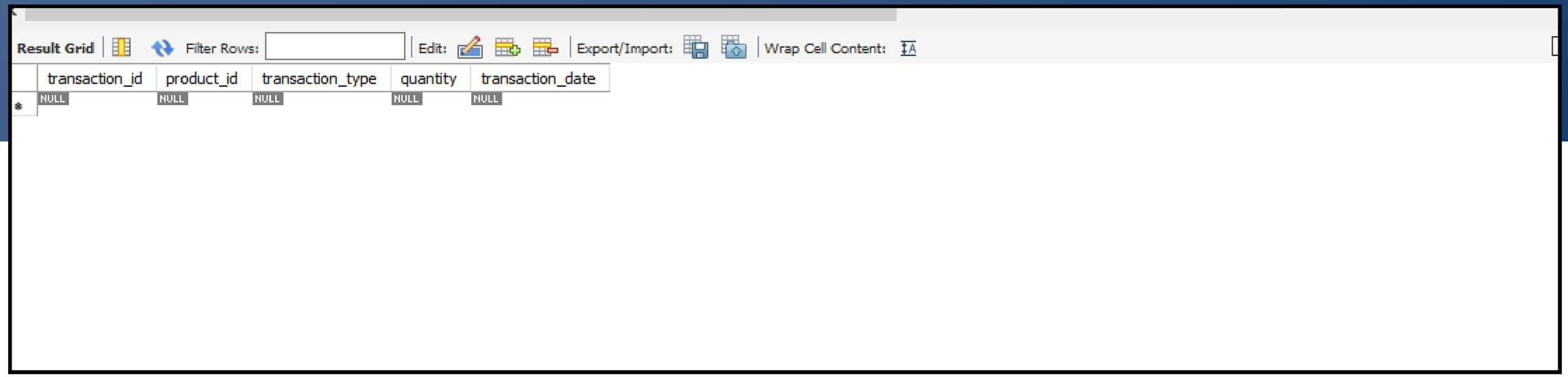
7. Calculate the average price of all products.

```
SELECT AVG(price) AS average_price
FROM Products;
```



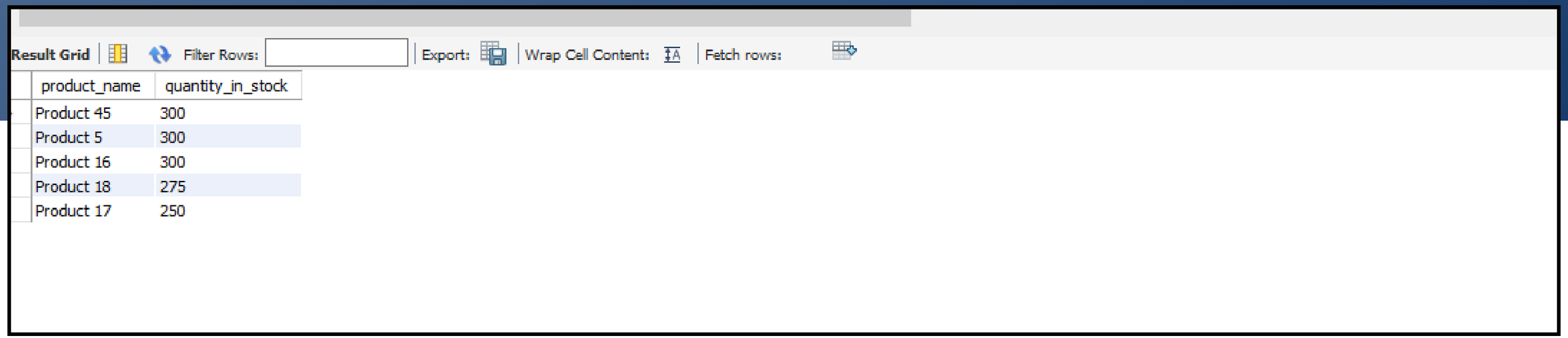
8. List all transactions of type 'sale' in the past 30 days.

```
SELECT *
FROM Inventory_Transactions
WHERE transaction_type = 'sale'
AND transaction_date >= CURDATE() - INTERVAL 30 DAY;
```



9 . Find the top 5 products with the highest quantity in stock.

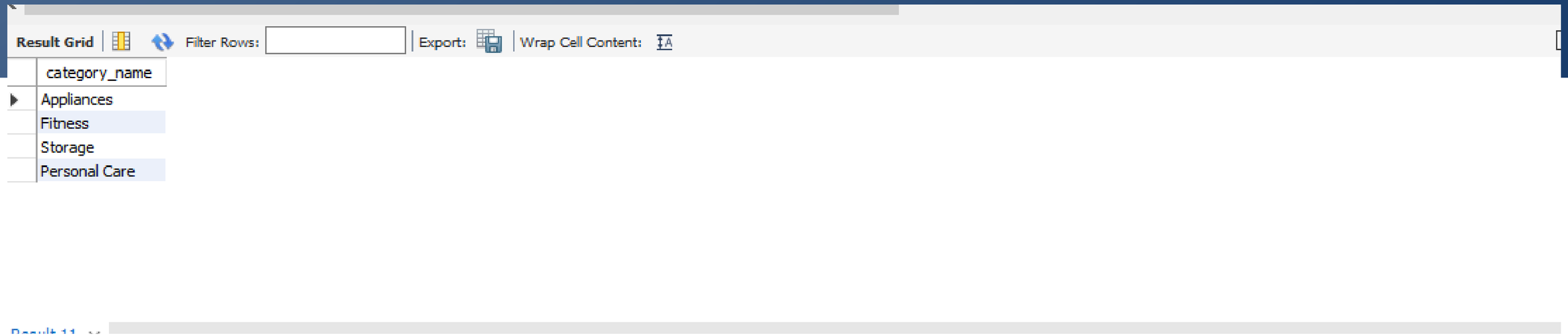
```
SELECT product_name, quantity_in_stock
FROM Products
ORDER BY quantity_in_stock DESC
LIMIT 5;
```



product_name	quantity_in_stock
Product 45	300
Product 5	300
Product 16	300
Product 18	275
Product 17	250

10. List categories with at least one product priced above $30.

```
SELECT DISTINCT category_name
FROM Products
JOIN Categories ON Products.category_id = Categories.category_id
WHERE price > 30.00;
```



THANK YOU