



# **Inventory Management System**

By-Pranali Bhosale

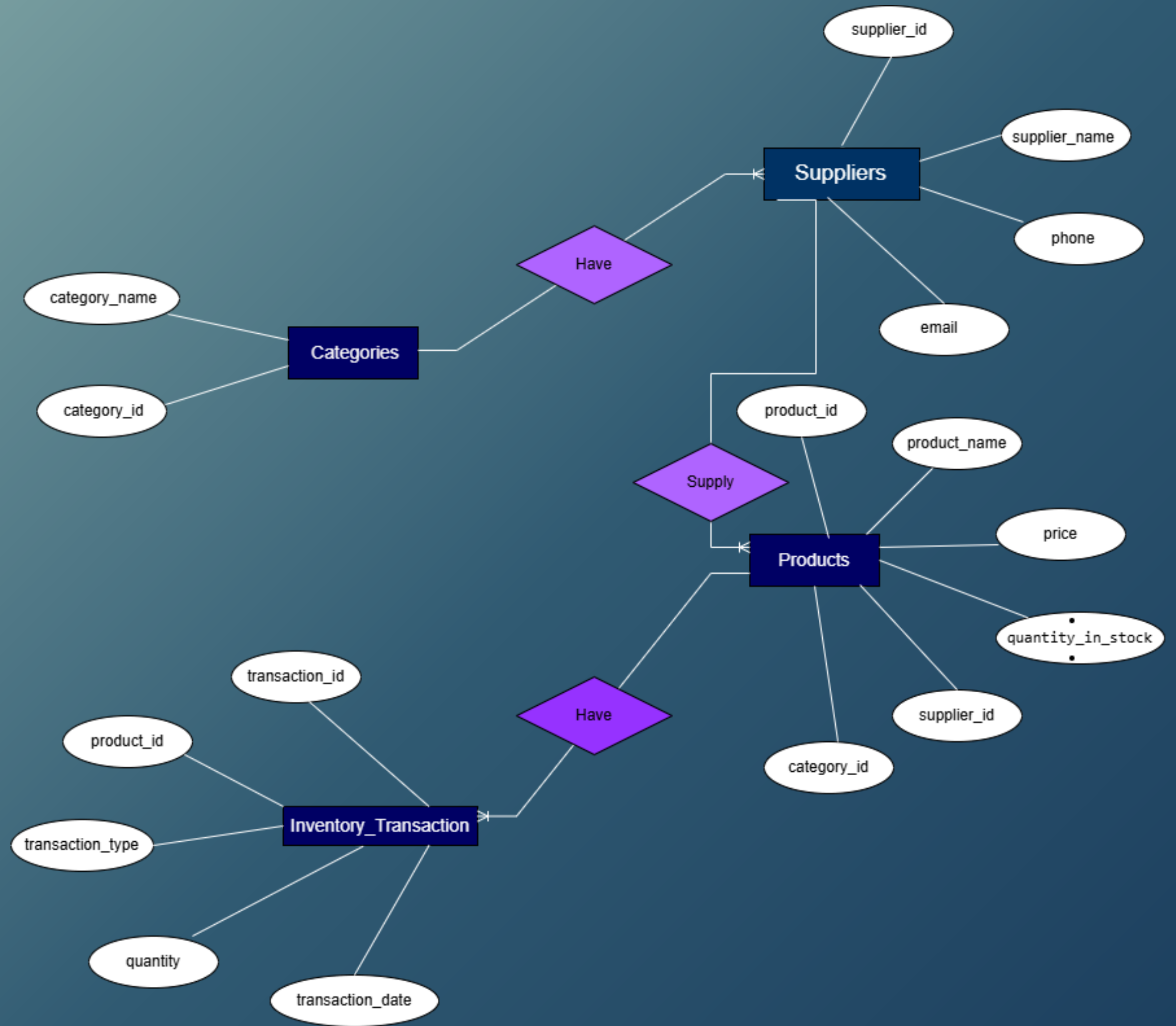
# Introduction:

**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





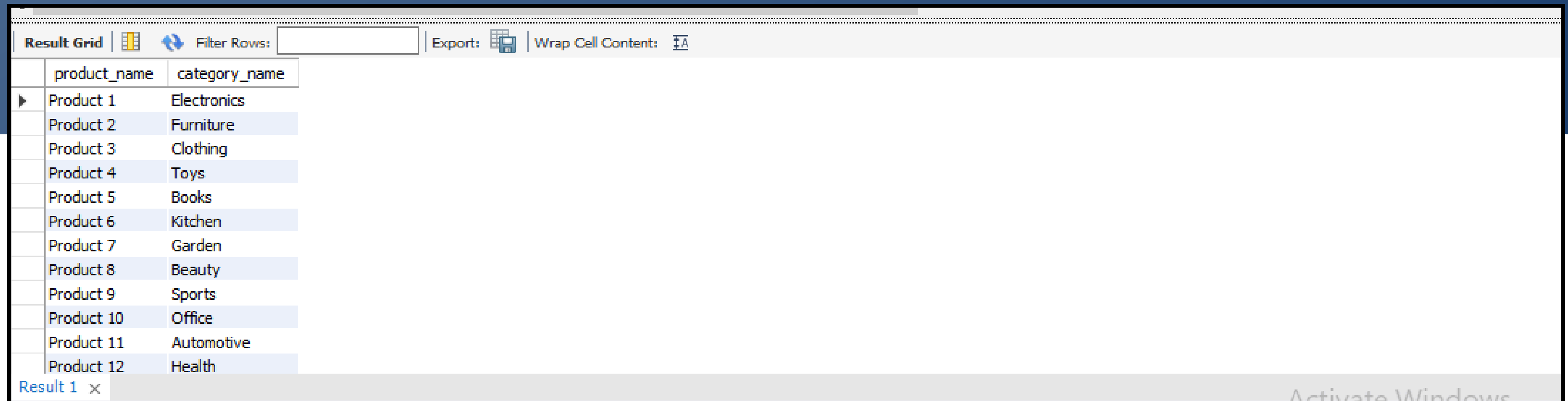
# Tables Created :

- 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;
```



The screenshot shows a database query result grid with the following data:

	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 x

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;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	supplier_name			
▶	Supplier F			
	Supplier I			
	Supplier K			
	Supplier N			
	Supplier X			
	Supplier AA			
	Supplier AC			
	Supplier AD			
	Supplier AH			
	Supplier AJ			



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: <input type="text"/>   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			

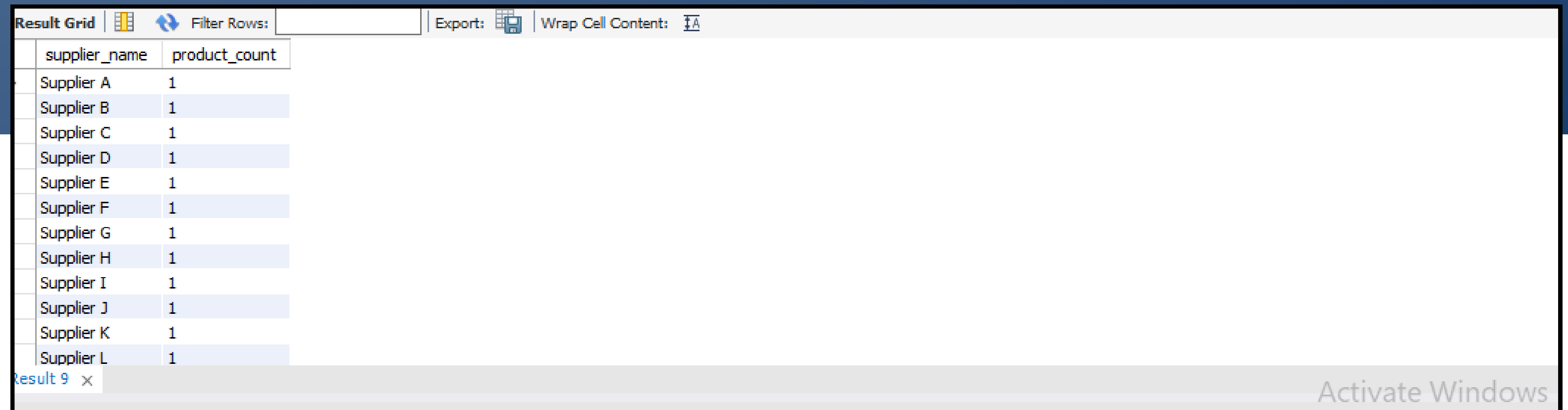
Result 4 ×

Activate Windows

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;
```



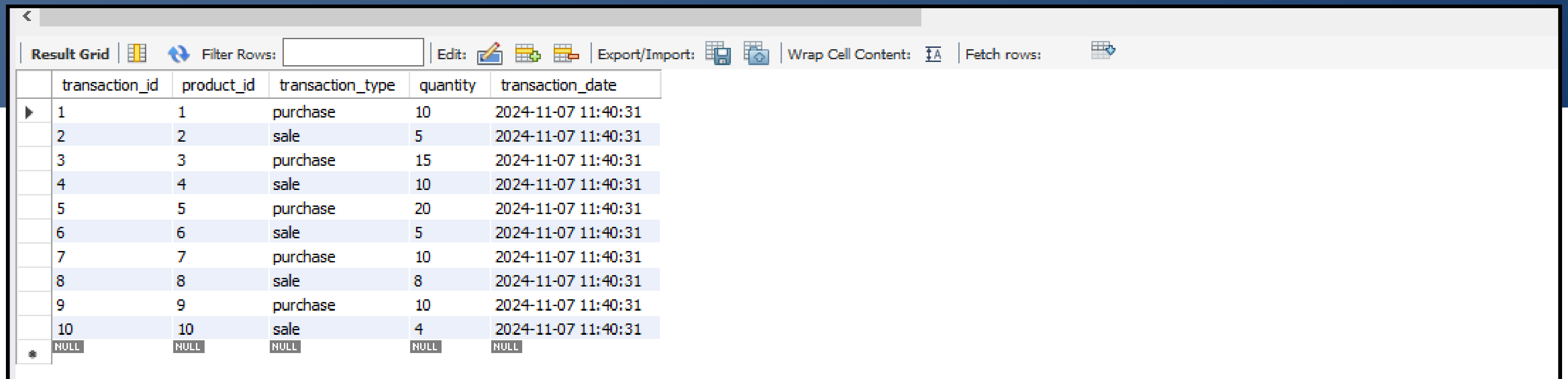
The screenshot shows a database query result grid with the following columns: `supplier_name` and `product_count`. The results are as follows:

supplier_name	product_count
Supplier A	1
Supplier B	1
Supplier C	1
Supplier D	1
Supplier E	1
Supplier F	1
Supplier G	1
Supplier H	1
Supplier I	1
Supplier J	1
Supplier K	1
Supplier L	1

The interface includes a toolbar with options like 'Filter Rows', 'Export', and 'Wrap Cell Content'. The window title is 'Result 9' and there is an 'Activate Windows' watermark in the bottom right corner.

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

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

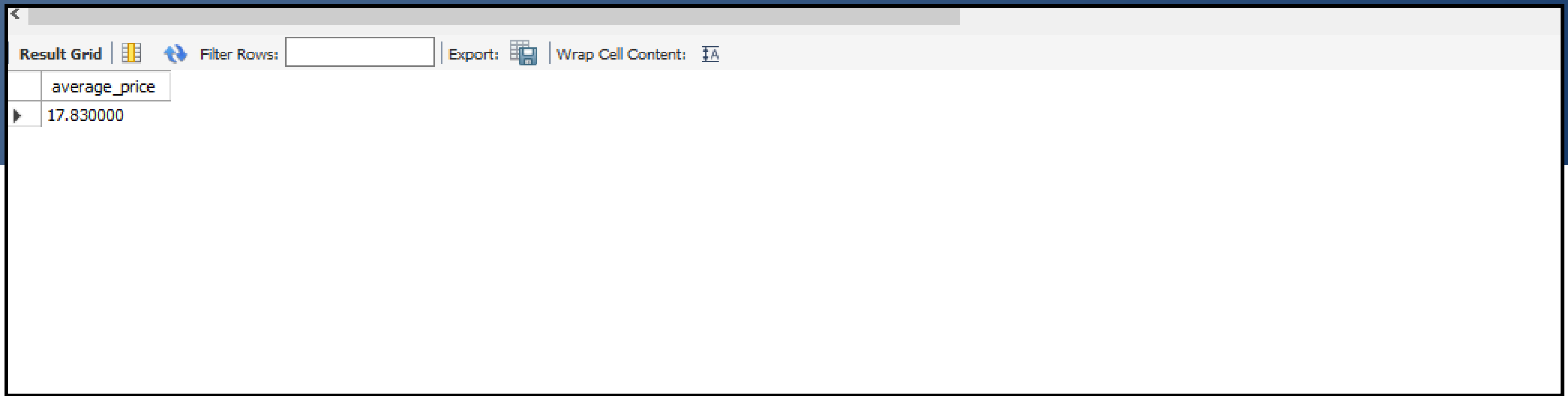


The screenshot shows a database query result grid with a toolbar at the top. The toolbar includes options for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', 'Wrap Cell Content', and 'Fetch rows'. The grid displays 10 rows of transaction data, sorted by 'transaction\_date' in descending order. Each row contains the transaction ID, product ID, transaction type, quantity, and the transaction date. The dates are all '2024-11-07 11:40:31'. The last row shows NULL values for all fields.

	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;
```

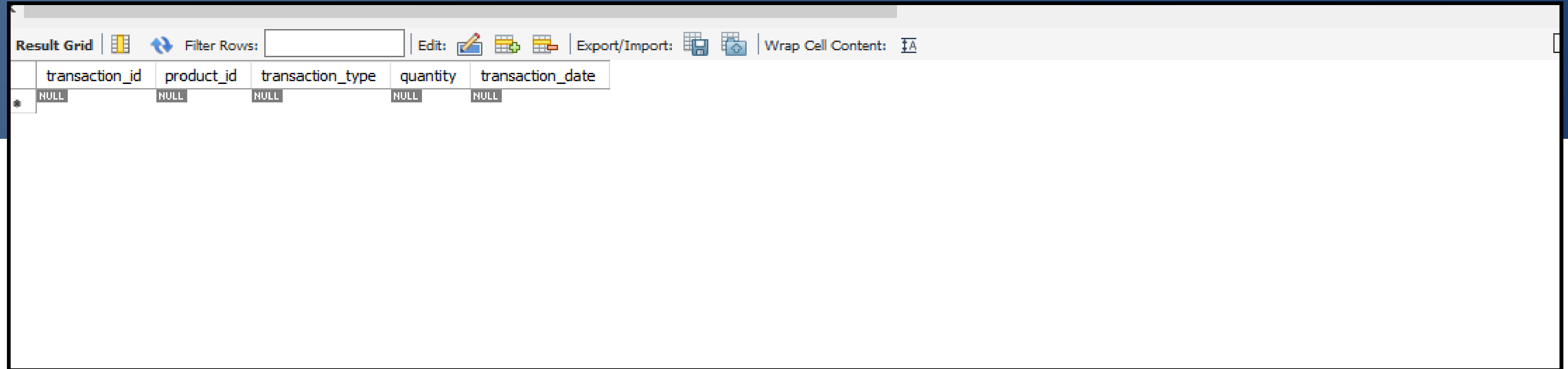


The screenshot shows a database query result grid. The grid has a header row with the column name 'average\_price' and a data row with the value '17.830000'. The grid is part of a software interface with a toolbar at the top containing icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

average_price
17.830000

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;
```



The screenshot shows a database application window with a toolbar at the top. The toolbar includes buttons for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below the toolbar is a table with five columns: 'transaction\_id', 'product\_id', 'transaction\_type', 'quantity', and 'transaction\_date'. The first row of the table contains five 'NULL' values. A small asterisk icon is visible in the first column of the first row.

transaction_id	product_id	transaction_type	quantity	transaction_date
NULL	NULL	NULL	NULL	NULL

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;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	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;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	category_name			
▶	Appliances			
	Fitness			
	Storage			
	Personal Care			





THANK YOU

