



# amazone case study

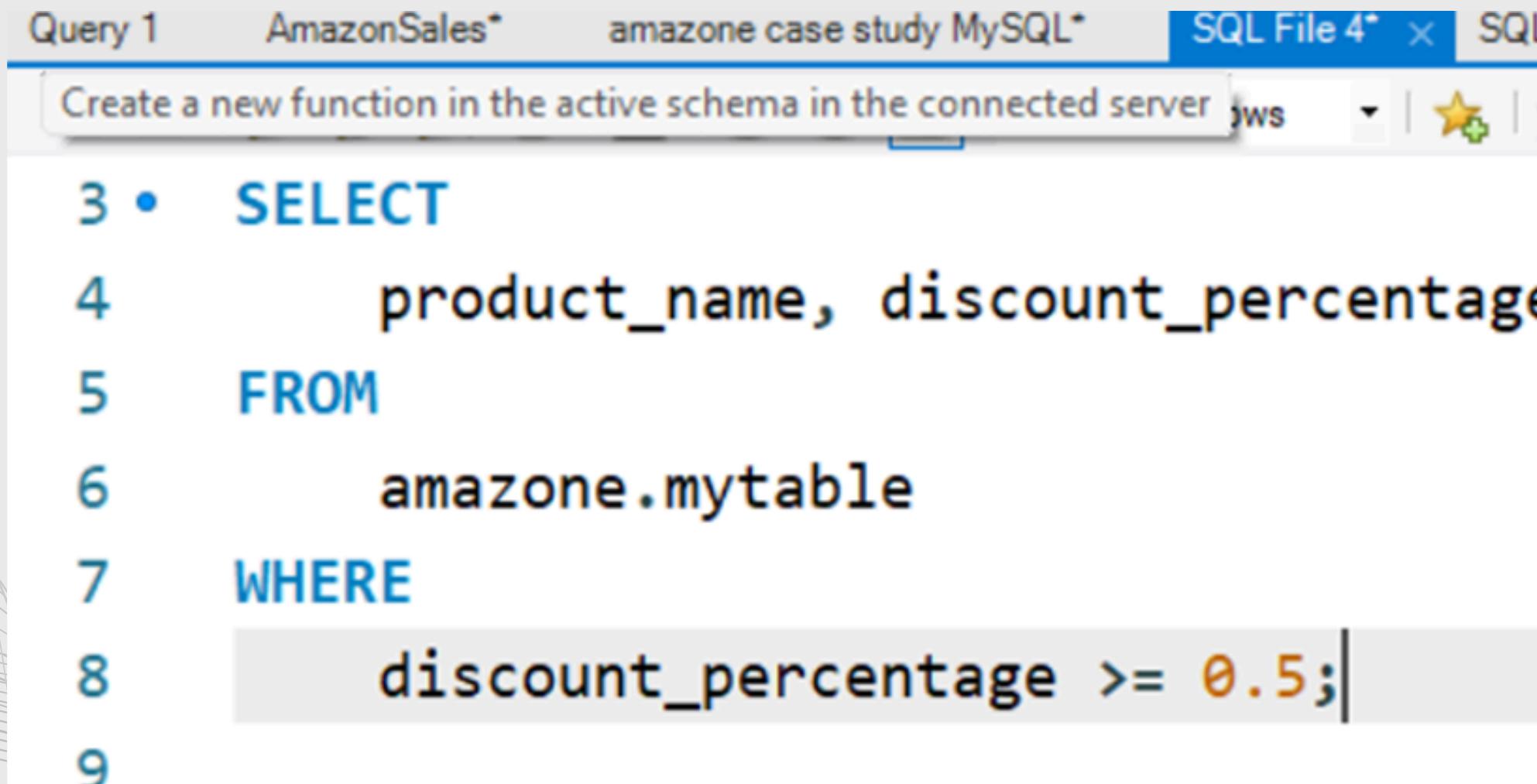
By: Renu Shokeen

# List all products with a discounted price below ₹500.

```
1 • SELECT
2      *
3  FROM
4    amazone.mytable
5 WHERE
6   discounted_price < 500 ;
```

**Purpose:**  
To find and list all  
products priced  
under ₹500 after  
discount.

# Find products with a discount percentage of 50% or more.

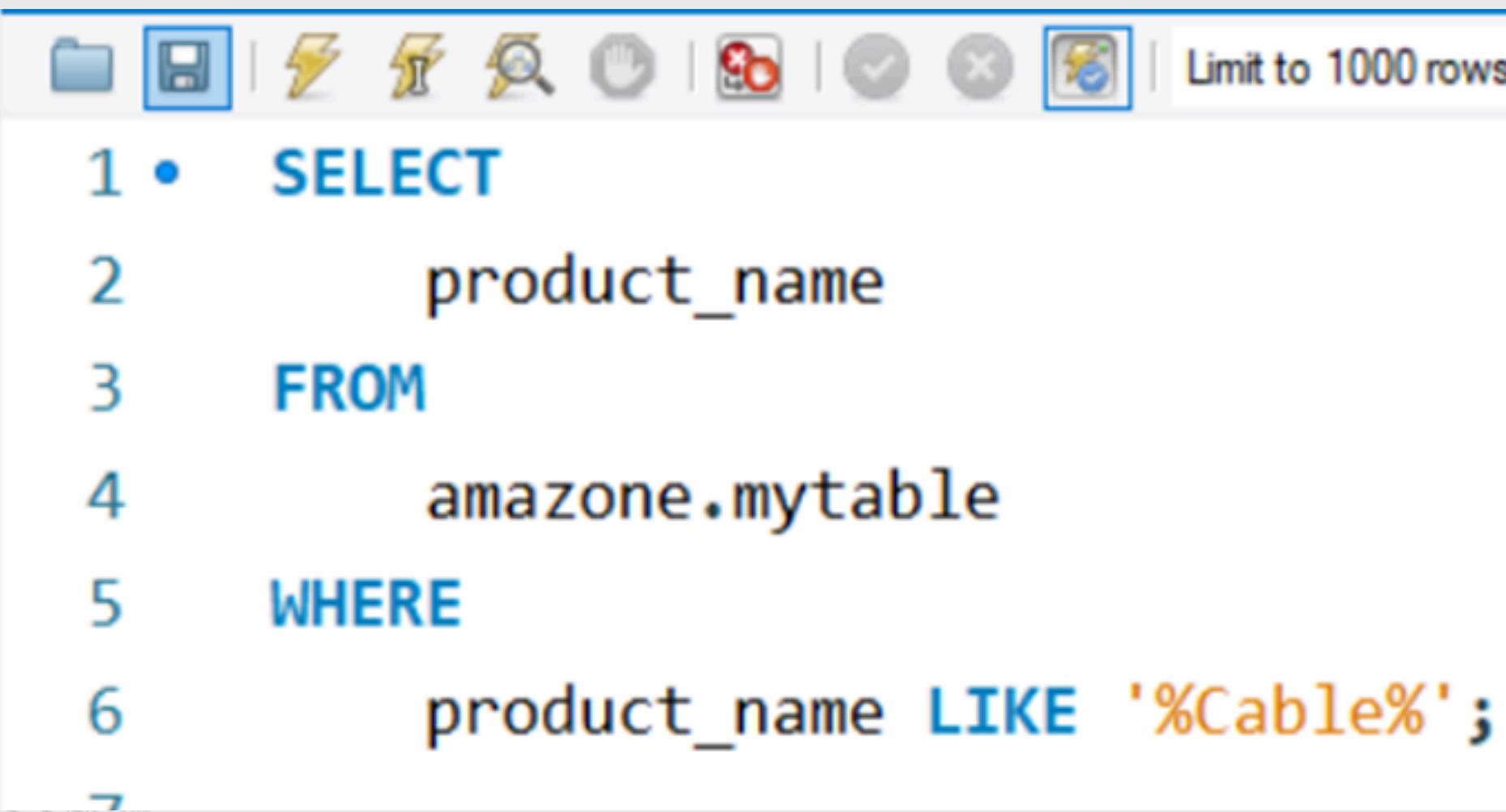


The screenshot shows a MySQL Workbench interface with multiple tabs at the top: 'Query 1', 'AmazonSales\*', 'amazone case study MySQL\*', 'SQL File 4\*', and 'SQL I'. The 'Query 1' tab is active. Below the tabs, there is a toolbar with icons for creating functions, opening files, and saving. The main area contains a SQL query:

```
3 • SELECT
4     product_name, discount_percentage
5 FROM
6     amazone.mytable
7 WHERE
8     discount_percentage >= 0.5;
9
```

**Purpose:**  
To find products offering significant discounts (50% or more)

# Retrieve all products where the name contains the word "Cable."

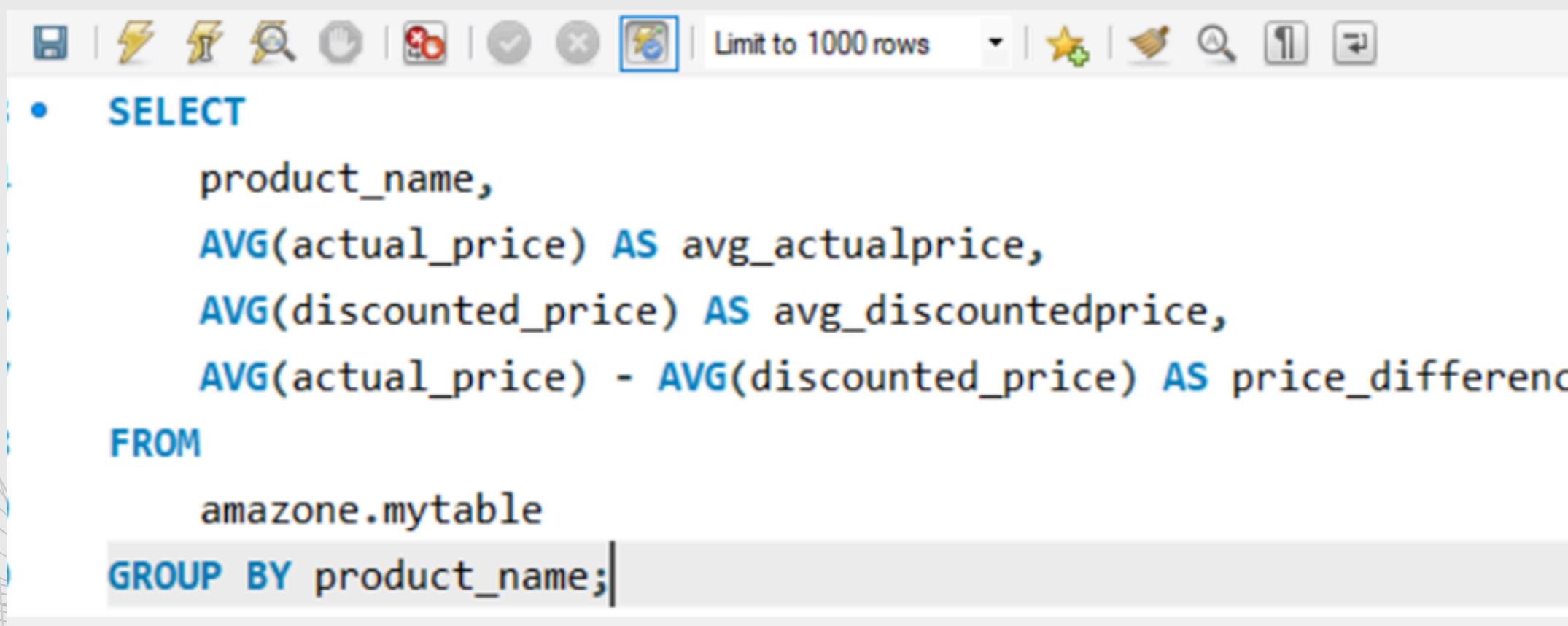


The screenshot shows a MySQL Workbench interface with a query editor window. The toolbar at the top includes icons for file operations, a refresh button, a search icon, a help icon, and a connection status. A limit of 1000 rows is specified. The query itself is a SELECT statement:

```
1 • SELECT
2     product_name
3 FROM
4     amazone.mytable
5 WHERE
6     product_name LIKE '%Cable%';
```

**Purpose:**  
To filter and display  
all cable-related  
products

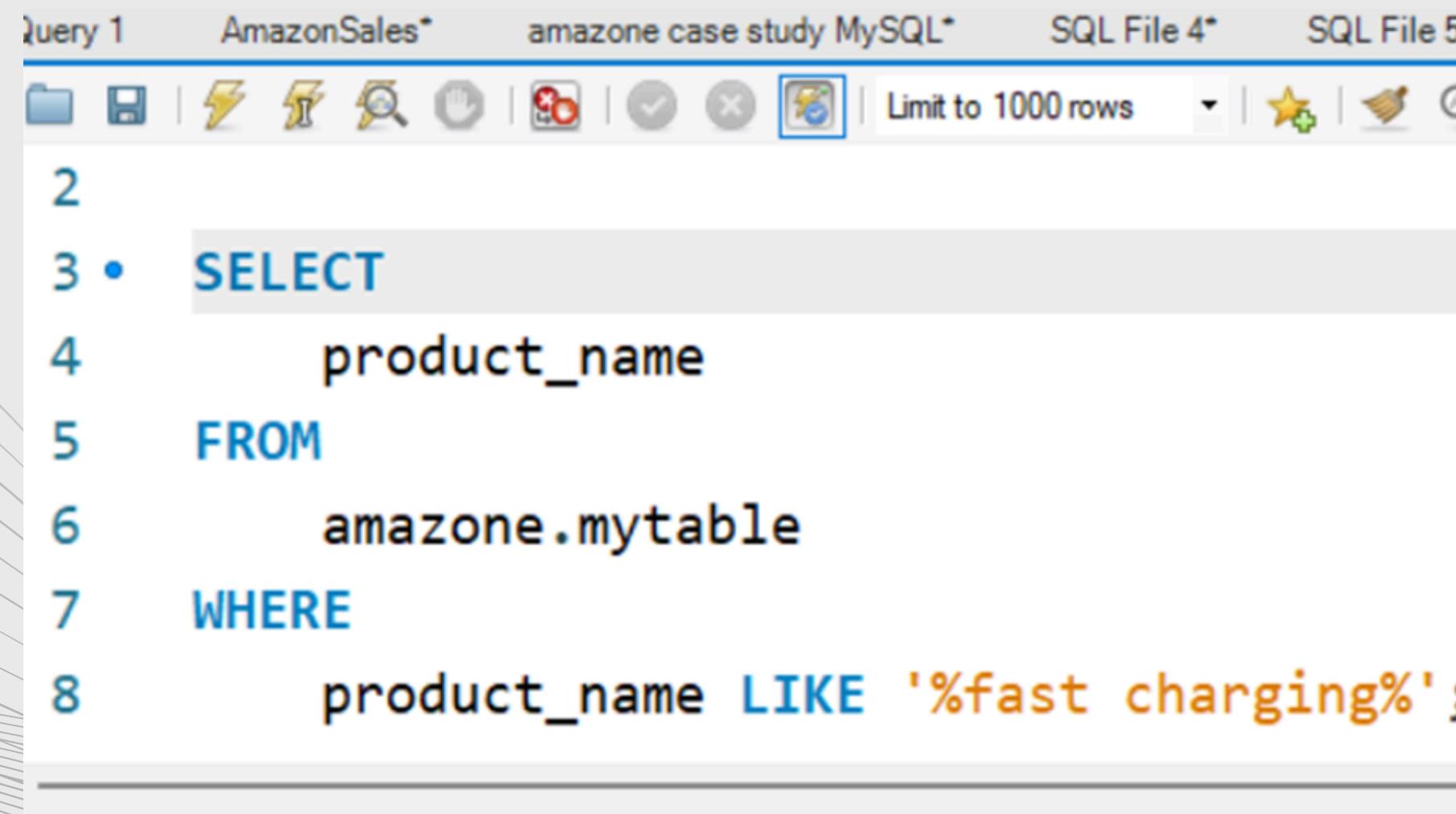
Display the difference between the average of the actual price and the average of discounted price for each product.



```
• SELECT
    product_name,
    AVG(actual_price) AS avg_actualprice,
    AVG(discounted_price) AS avg_discountedprice,
    AVG(actual_price) - AVG(discounted_price) AS price_difference
FROM
    amazone.mytable
GROUP BY product_name;
```

**Purpose:**  
To measure the average discount impact per product.

# Query reviews that mention "fast charging" in their content.

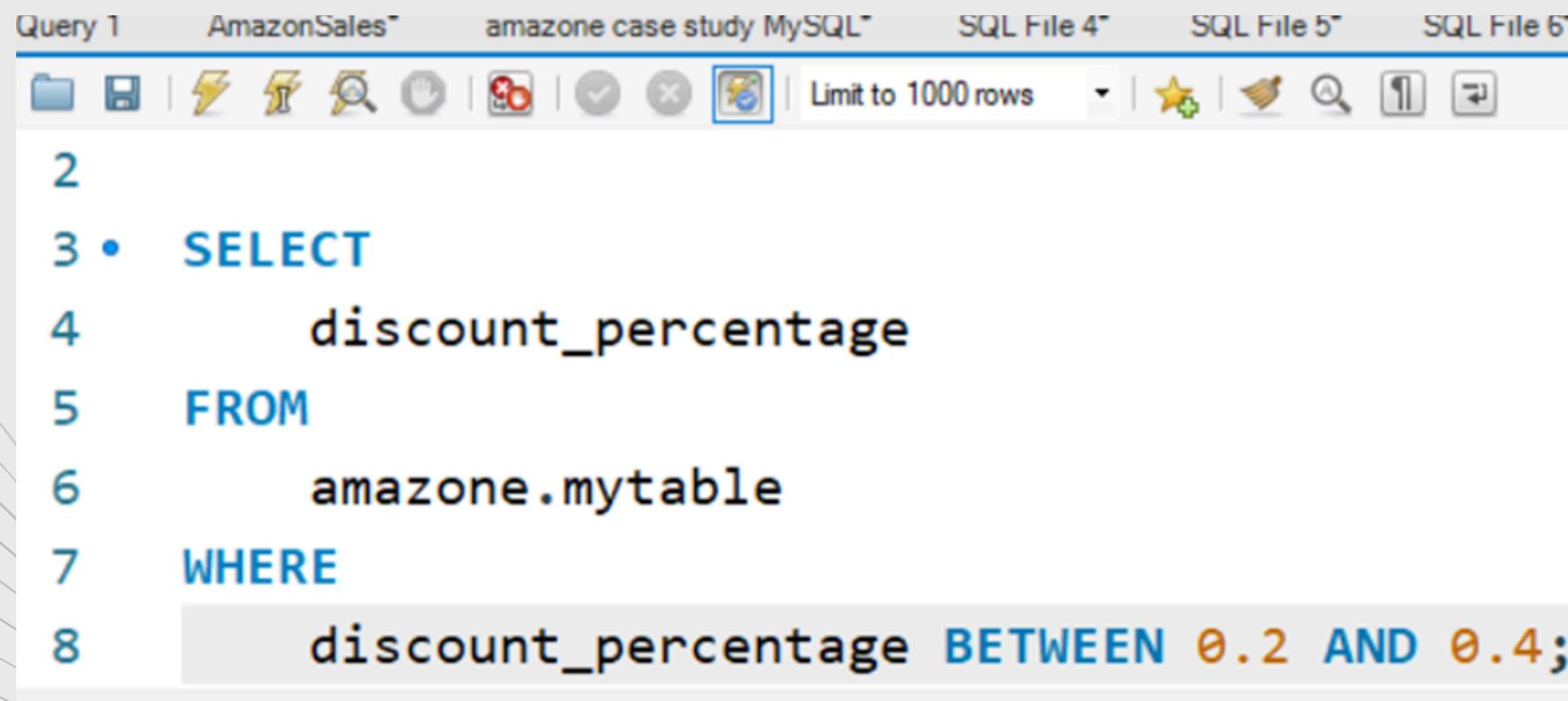


The screenshot shows a MySQL Workbench interface with multiple tabs at the top: 'Query 1' (selected), 'AmazonSales\*', 'amazone case study MySQL\*', 'SQL File 4\*', and 'SQL File 5\*'. The main area displays a SQL query:

```
2
3 • SELECT
4     product_name
5 FROM
6     amazone.mytable
7 WHERE
8     product_name LIKE '%fast charging%';
```

The 'WHERE' clause and the search term '%fast charging%' are highlighted in orange, while the rest of the query is in blue. The interface includes a toolbar with various icons and a 'Limit to 1000 rows' dropdown.

# Identify products with a discount percentage between 20% and 40%.



The screenshot shows a MySQL Workbench interface with multiple tabs at the top: Query 1, AmazonSales\*, amazone case study MySQL\*, SQL File 4\*, SQL File 5\*, and SQL File 6\*. The main area contains a SQL query:

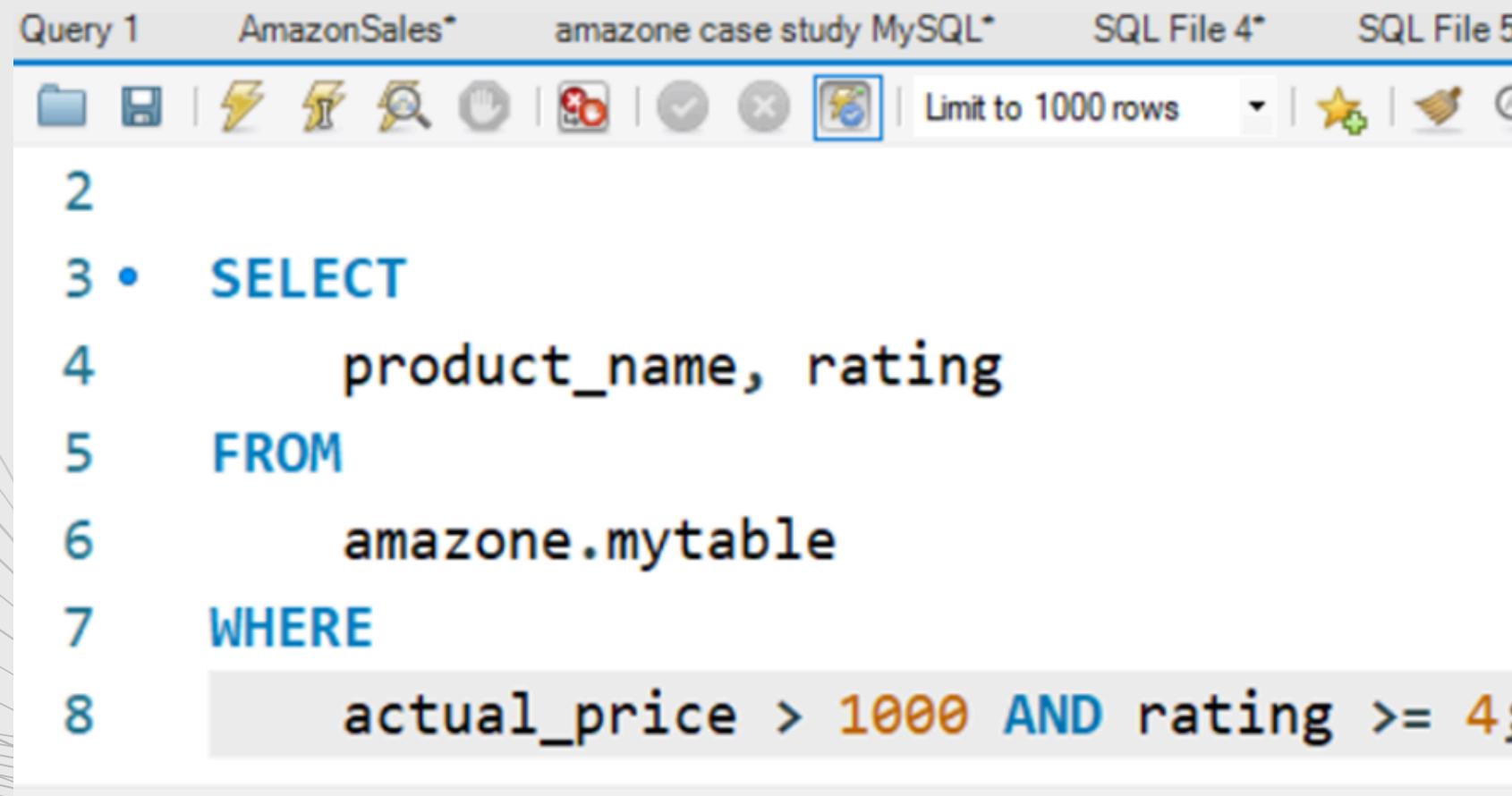
```
2
3 • SELECT
4     discount_percentage
5 FROM
6     amazone.mytable
7 WHERE
8     discount_percentage BETWEEN 0.2 AND 0.4;
```

The code uses backticks for the table name 'amazone.mytable' and the column name 'discount\_percentage'. The WHERE clause includes a condition using the BETWEEN operator to filter rows where the discount percentage is between 0.2 and 0.4.

**Purpose:**  
To list products offering mid-range discounts (20%–40%).

Find products that have an actual price above ₹1,000 and are rated 4 stars or above.

Query 1   AmazonSales\*   amazone case study MySQL\*   SQL File 4\*   SQL File 5\*



```
2
3 • SELECT
4     product_name, rating
5 FROM
6     amazone.mytable
7 WHERE
8     actual_price > 1000 AND rating >= 4;
```

**Purpose:**  
To list high-priced products with strong customer ratings.

# Find products where the discounted price ends with a 9 ?

```
2
3 • SELECT
4     product_name, discounted_price
5 FROM
6     amazone.mytable
7 WHERE
8     discounted_price LIKE '%9';
```

## Purpose:

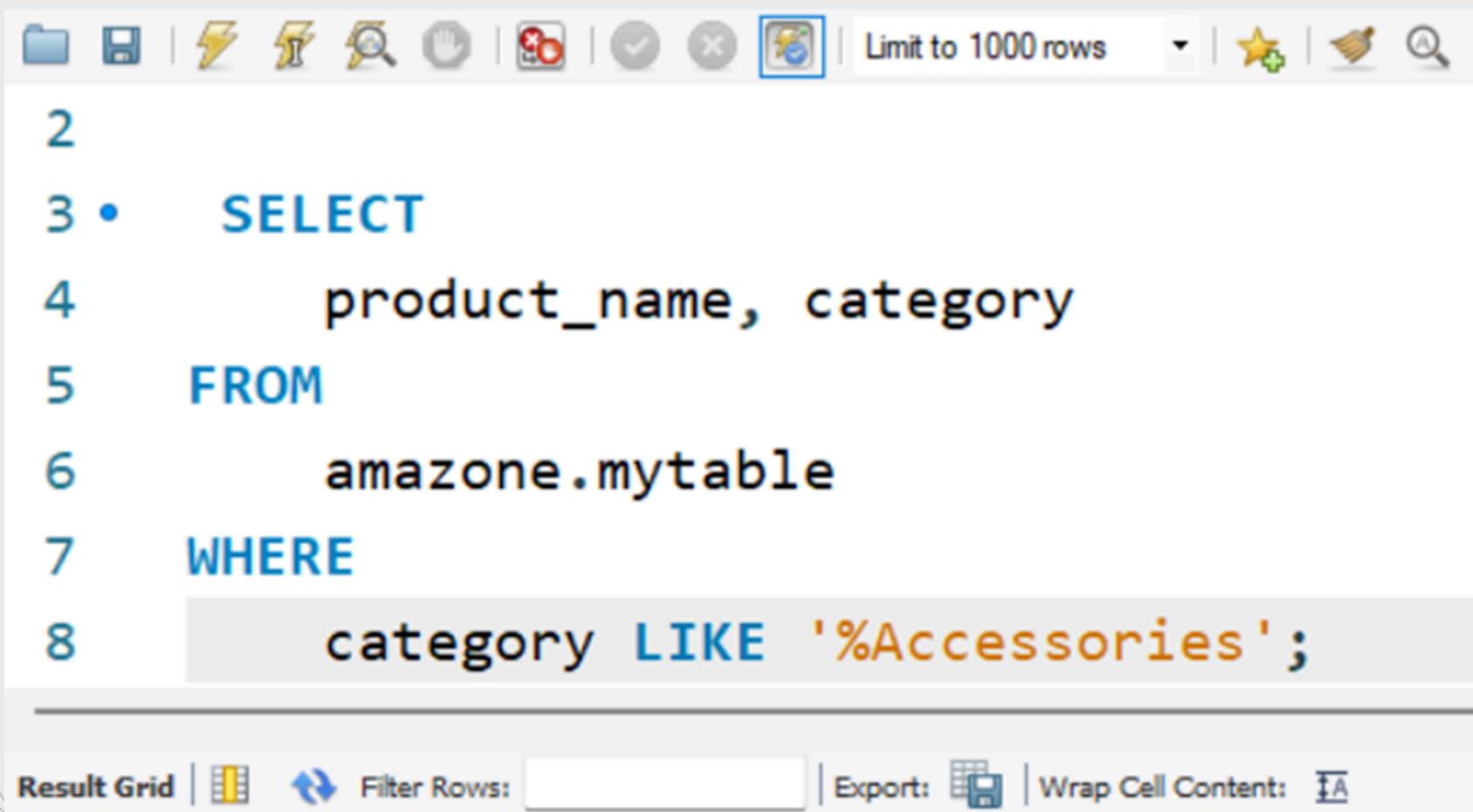
It filters products based on a pricing pattern commonly used in marketing.

**Display review contents that contains words like worst,  
waste, poor, or not good.**

```
SELECT  
    review_content  
FROM  
    amazone.mytable  
WHERE  
    review_content LIKE '%worst%'  
        OR '%waste%'  
        OR '%poor%'  
        OR '%not good%';
```

**Purpose:**  
**To extract reviews  
with negative  
sentiment.**

# List all products where the category includes "Accessories."



A screenshot of a MySQL Workbench query editor. The query window contains the following SQL code:

```
2
3 •  SELECT
4      product_name, category
5  FROM
6      amazone.mytable
7 WHERE
8      category LIKE '%Accessories';
```

The code is numbered 1 through 8. The WHERE clause is highlighted with a gray background. The status bar at the bottom shows "Result Grid" and other export options.

**Purpose:**  
**To display all accessory-related products.**

**Thank you!**