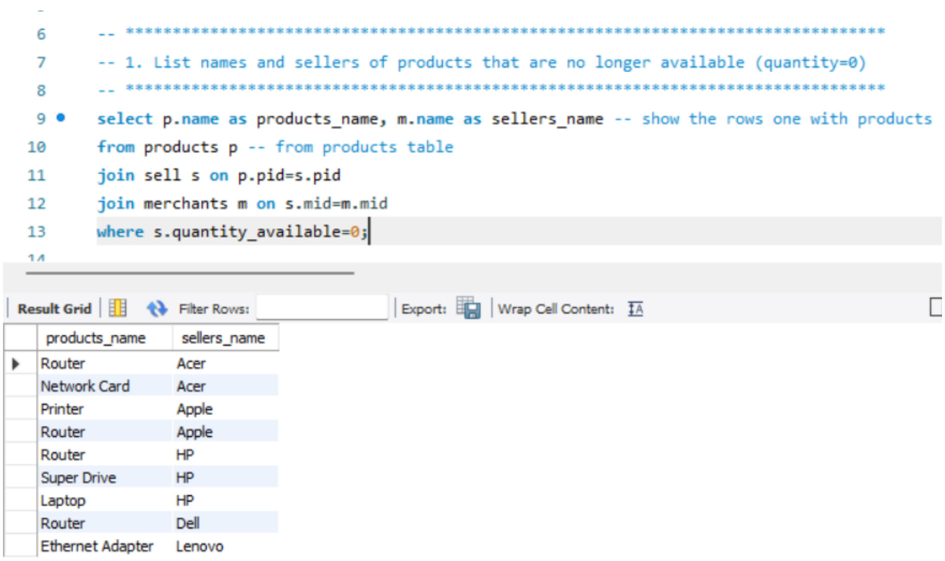
1. List names and sellers of products that are no longer available (quantity=0)

Explanation of Each SQL Component:

SQL Component	Role in Query	Explanation
select p.name as product_name, m.name AS sellers_name	Select Statement	Retrieves the names of products and merchants, labeling them as products_name and sellers_name in the output. This clarifies what each column represents in the results.
from products p	From Clause	Specifies the primary table (products) from which to start gathering data. The alias p simplifies referencing the table in other parts of the query.
join sell s on p.pid=s.pid	Join Clause	Performs an inner join between products (p) and sell (s) based on the product ID (pid). This ensures that only products that are linked to a sale are considered.
join merchants m on s.mid=m.mid	Join Clause	Connects the sell records (s) with their corresponding merchants (m) using the merchant ID (mid). This join retrieves the merchant details for each sale.
where s.quantity_available=0;	Where Clause	Filters the results to show only those products whose quantity_available is zero, effectively showing products that are out of stock.



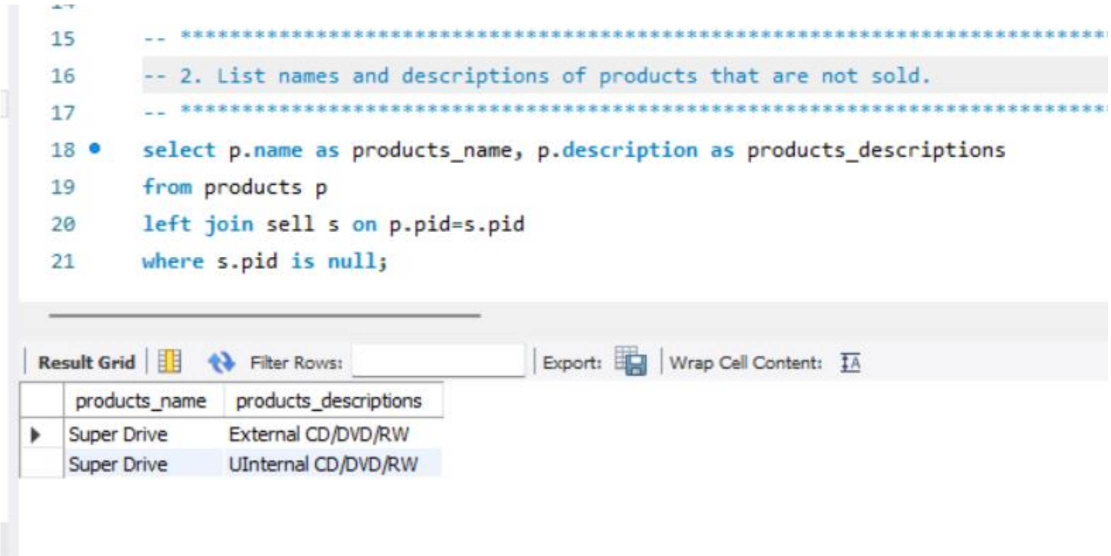
```
-- **********************************************************************************
-- 1. List names and sellers of products that are no longer available (quantity=0)
-- **********************************************************************************
select p.name as products_name, m.name as sellers_name -- show the rows one with products
from products p -- from products table
join sell s on p.pid=s.pid
join merchants m on s.mid=m.mid
where s.quantity_available=0;
```

products_name	sellers_name
Router	Acer
Network Card	Acer
Printer	Apple
Router	Apple
Router	HP
Super Drive	HP
Laptop	HP
Router	Dell
Ethernet Adapter	Lenovo

2. List names and descriptions of products that are not sold.

Explanation of Each SQL Component:

SQL Component	Role in Query	Explanation
select p.name as products_name, p.description as products_descriptions	Select Statement	Retrieves the names and descriptions of products. It labels them as products_name and products_description in the output, clearly identifying the contents of each column for easy understanding.
from products p	From Clause	Specifies the primary table, products, from which to retrieve data. The alias p is used to simplify references to this table in other parts of the query.
left join sell s on p.pid=s.pid	Join Clause	Performs a left join between products (p) and sell (s). This join type ensures all products are shown, including those without matching records in sell. The join is based on the product ID (pid), ensuring that products are attempted to be matched with sales records.
where s.pid is null;	Where Clause	Filters the results to include only those records where there is no corresponding entry in the sell table, as indicated by s.pid being NULL. This condition effectively selects products that have not been sold.

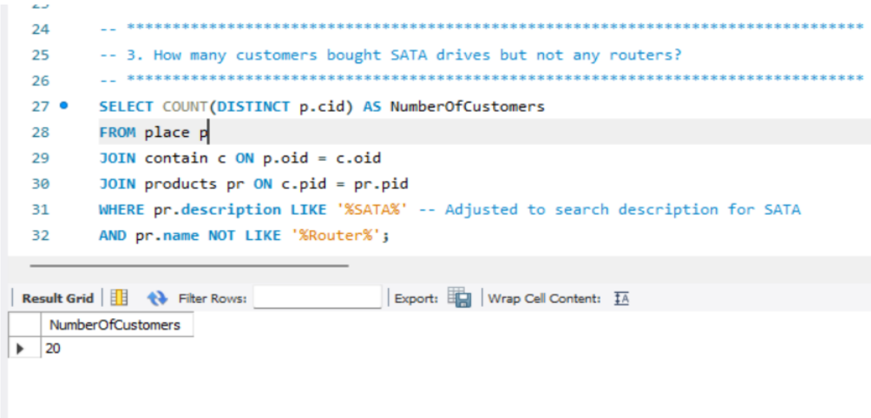


```
-- ***********************************************************************
-- 2. List names and descriptions of products that are not sold.
-- ***********************************************************************
select p.name as products_name, p.description as products_descriptions
from products p
left join sell s on p.pid=s.pid
where s.pid is null;
```

products_name	products_descriptions
Super Drive	External CD/DVD/RW
Super Drive	UInternal CD/DVD/RW

3. How many customers bought SATA drives but not any routers?

SQL Component	Role in Query	Explanation
SELECT COUNT(DISTINCT p.cid) AS NumberOfCustomers	Selection and Aggregation	This command selects and counts the distinct customer IDs (cid) from the place table. The result is aliased as NumberOfCustomers, providing the total number of unique customers who match the conditions specified in the query.
FROM place p	Base Table for Join	Specifies the place table as the starting point for the join operations. The table is aliased as p to simplify references to it in the rest of the query.
JOIN contain c ON p.oid = c.oid	Join Operation	Joins the place table (p) with the contain table (c). The join is made on the oid column, linking orders in place with their corresponding entries in contain, which details what products are included in each order.
JOIN products pr ON c.pid = pr.pid	Join Operation	Further joins the contain table (c) to the products table (pr). This join is based on the pid column, connecting the products listed in contain to their detailed descriptions in products.
WHERE pr.description LIKE '%SATA%'	Filtering Condition	Adds a condition to filter the products based on their descriptions. Only products whose description contains 'SATA' are considered for the aggregation. This targets only those entries relevant to products associated with SATA technology.
AND pr.name NOT LIKE '%Router%'	Additional Filter	Further refines the selection by excluding any products whose names include the word 'Router'. This ensures that the count excludes customers who have purchased any routers, focusing on a specific subset of products.

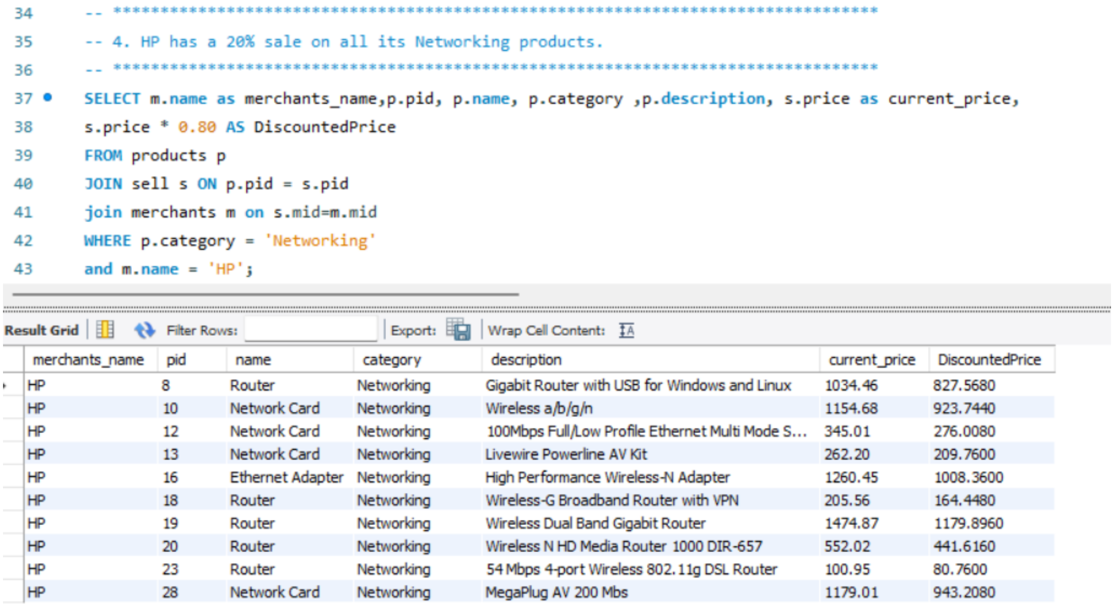


```
-- ********************************************************************************
-- 3. How many customers bought SATA drives but not any routers?
-- ********************************************************************************
SELECT COUNT(DISTINCT p.cid) AS NumberOfCustomers
FROM place p
JOIN contain c ON p.oid = c.oid
JOIN products pr ON c.pid = pr.pid
WHERE pr.description LIKE '%SATA%' -- Adjusted to search description for SATA
AND pr.name NOT LIKE '%Router%';
```

NumberOfCustomers
20

4. HP has a 20% sale on all its Networking products.

SQL Component	Role in Query	Explanation
SELECT m.name as merchants_name	Selection and Projection	Retrieves the merchant's name from the merchant table and renames it as merchants_name for the output.
p.pid	Selection	Selects the product ID from the products table.
p.name	Selection	Selects the product name from the products table.
p.category	Selection	Selects the product category from the products table.
p.description	Selection	Selects the product description from the products table.
s.price as current_price	Selection and Projection	Retrieves the current price from the sell table and renames it as current_price for the output.
s.price * 0.80 AS DiscountedPrice	Computation and Projection	Calculates 80% of the current price to derive the discounted price, displayed as DiscountedPrice.
FROM products p	Table Specification	Specifies the products table as the source of product data, and aliases it as p for reference in the query.
JOIN sell s ON p.pid = s.pid	Join Operation	Joins the products table with the sell table based on the matching product ID (pid).
JOIN merchants m ON s.mid = m.mid	Join Operation	Joins the sell table with the merchants table based on the matching merchant ID (mid).
WHERE p.category = 'Networking'	Filter	Filters the results to include only those products that belong to the 'Networking' category.
AND m.name = 'HP'	Additional Filter	Further filters the results to include only products sold by merchants named 'HP'.



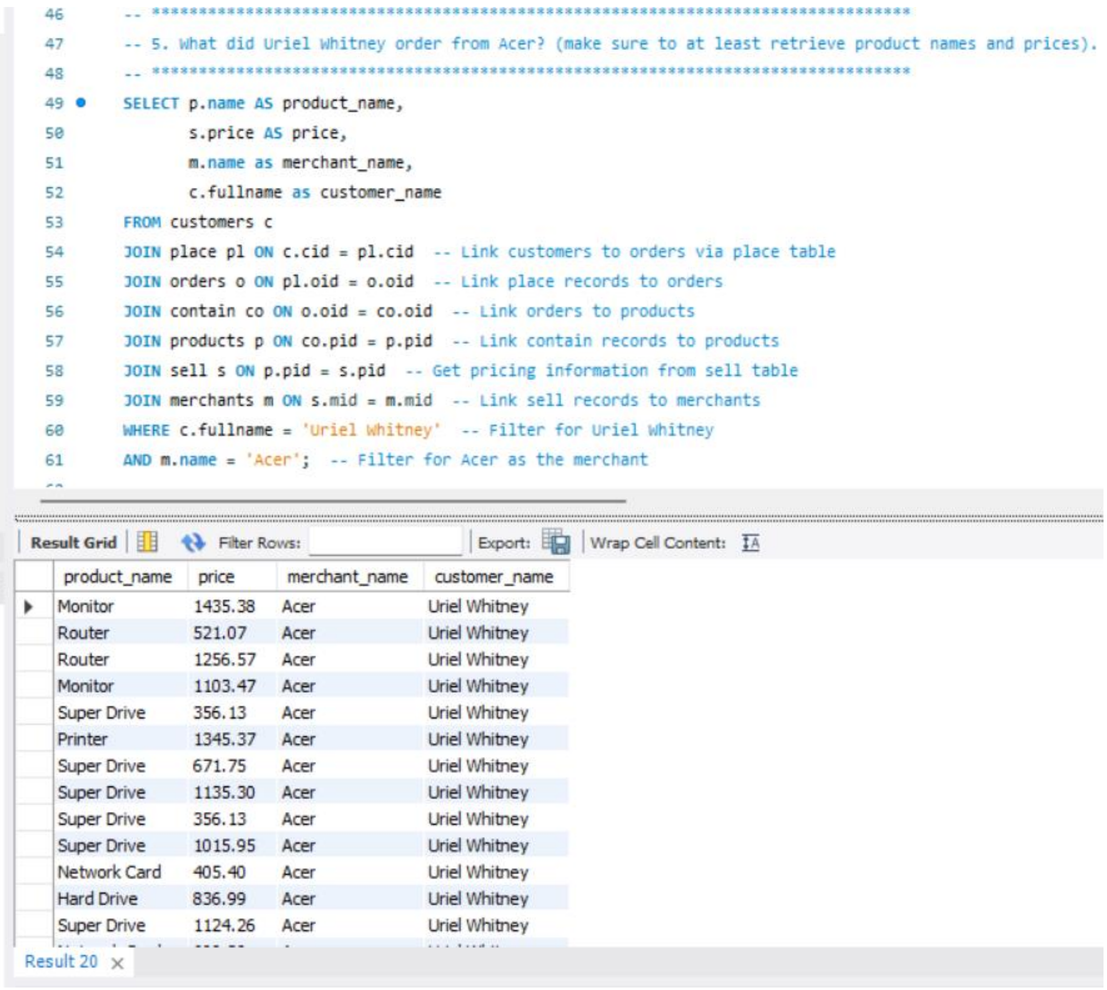
```sql
-- **********************************************************************************
-- 4. HP has a 20% sale on all its Networking products.
-- **********************************************************************************
SELECT m.name as merchants_name,p.pid, p.name, p.category ,p.description, s.price as current_price,
s.price * 0.80 AS DiscountedPrice
FROM products p
JOIN sell s ON p.pid = s.pid
join merchants m on s.mid=m.mid
WHERE p.category = 'Networking'
and m.name = 'HP';
```

merchants_name	pid	name	category	description	current_price	DiscountedPrice
HP	8	Router	Networking	Gigabit Router with USB for Windows and Linux	1034.46	827.5680
HP	10	Network Card	Networking	Wireless a/b/g/n	1154.68	923.7440
HP	12	Network Card	Networking	100Mbps Full/Low Profile Ethernet Multi Mode S...	345.01	276.0080
HP	13	Network Card	Networking	Livewire Powerline AV Kit	262.20	209.7600
HP	16	Ethernet Adapter	Networking	High Performance Wireless-N Adapter	1260.45	1008.3600
HP	18	Router	Networking	Wireless-G Broadband Router with VPN	205.56	164.4480
HP	19	Router	Networking	Wireless Dual Band Gigabit Router	1474.87	1179.8960
HP	20	Router	Networking	Wireless N HD Media Router 1000 DIR-657	552.02	441.6160
HP	23	Router	Networking	54 Mbps 4-port Wireless 802.11g DSL Router	100.95	80.7600
HP	28	Network Card	Networking	MegaPlug AV 200 Mbs	1179.01	943.2080

5. What did Uriel Whitney order from Acer? (make sure to at least retrieve product names and prices).

SQL Component	Role in Query	Explanation
SELECT p.name AS product_name	Selection and Projection	Retrieves the name of the product from the products table and aliases it as product_name in the output.
s.price AS price	Selection	Selects the price of the product from the sell table.
m.name as merchant_name	Selection and Projection	Retrieves the name of the merchant from the merchants table and aliases it as merchant_name in the output.
c.fullname as customer_name	Selection and Projection	Retrieves the full name of the customer from the customers table and aliases it as customer_name in the output.
FROM customers c	Table Specification	Specifies the customers table as the source of customer data, with an alias c for use in the query.
JOIN place pl ON c.cid = pl.cid	Join Operation	Joins the customers table to the place table, linking customers to their orders via the place table using the customer ID (cid).
JOIN orders o ON pl.oid = o.oid	Join Operation	Joins the place table to the orders table, linking the place records to specific orders using the order ID (oid).
JOIN contain co ON o.oid = co.oid	Join Operation	Joins the orders table to the contain table, linking orders to the products included in those orders using the order ID (oid).

JOIN products p ON co.pid = p.pid	Join Operation	Joins the contain table to the products table, linking the products contained in orders to their details in the products table using the product ID (pid).
JOIN sell s ON p.pid = s.pid	Join Operation	Joins the products table to the sell table, linking the products to their pricing information using the product ID (pid).
JOIN merchants m ON s.mid = m.mid	Join Operation	Joins the sell table to the merchants table, linking the sales records to the merchants using the merchant ID (mid).
WHERE c.fullname = 'Uriel Whitney'	Filter	Filters the results to include only records associated with the customer named "Uriel Whitney".
AND m.name = 'Acer'	Additional Filter	Further filters the results to include only those transactions involving the merchant named "Acer".

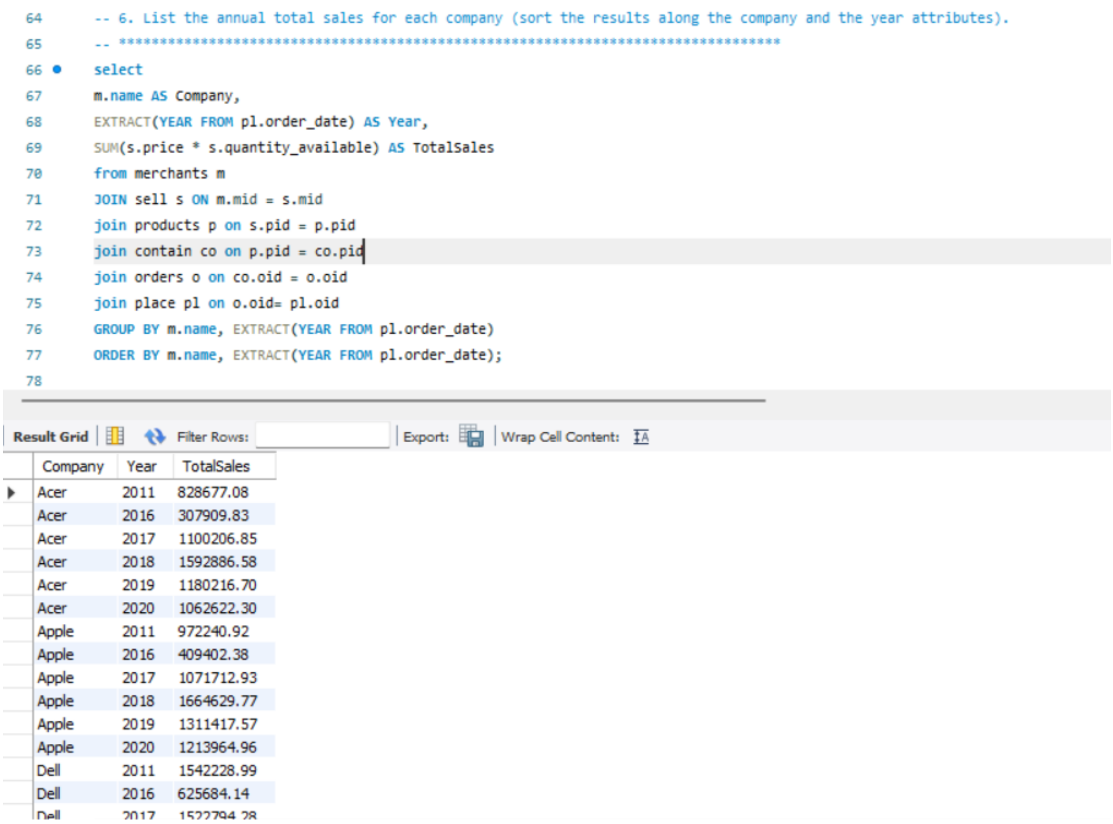


```
-- ********************************************************************************
-- 5. What did Uriel Whitney order from Acer? (make sure to at least retrieve product names and prices).
-- ********************************************************************************
SELECT p.name AS product_name,
       s.price AS price,
       m.name as merchant_name,
       c.fullname as customer_name
FROM customers c
JOIN place pl ON c.cid = pl.cid  -- Link customers to orders via place table
JOIN orders o ON pl.oid = o.oid  -- Link place records to orders
JOIN contain co ON o.oid = co.oid  -- Link orders to products
JOIN products p ON co.pid = p.pid  -- Link contain records to products
JOIN sell s ON p.pid = s.pid  -- Get pricing information from sell table
JOIN merchants m ON s.mid = m.mid  -- Link sell records to merchants
WHERE c.fullname = 'Uriel Whitney'  -- Filter for Uriel Whitney
AND m.name = 'Acer';  -- Filter for Acer as the merchant
```

product_name	price	merchant_name	customer_name
Monitor	1435.38	Acer	Uriel Whitney
Router	521.07	Acer	Uriel Whitney
Router	1256.57	Acer	Uriel Whitney
Monitor	1103.47	Acer	Uriel Whitney
Super Drive	356.13	Acer	Uriel Whitney
Printer	1345.37	Acer	Uriel Whitney
Super Drive	671.75	Acer	Uriel Whitney
Super Drive	1135.30	Acer	Uriel Whitney
Super Drive	356.13	Acer	Uriel Whitney
Super Drive	1015.95	Acer	Uriel Whitney
Network Card	405.40	Acer	Uriel Whitney
Hard Drive	836.99	Acer	Uriel Whitney
Super Drive	1124.26	Acer	Uriel Whitney

6. List the annual total sales for each company (sort the results along the company and the year attributes).

SQL Component	Role in Query	Explanation
SELECT	Clause	Begins the query and specifies the columns to be displayed in the result set.
m.name AS Company	Selection and Projection	Retrieves the name of the company from the merchants table and aliases it as Company in the output.
EXTRACT(YEAR FROM pl.order_date) AS Year	Function and Projection	Extracts the year part from the order_date column of the place table and aliases it as Year.
SUM(s.price * s.quantity_available) AS TotalSales	Aggregate Function	Calculates the total sales by multiplying the price by the quantity available from the sell table, then summing up these values for each group.
FROM merchants m	Table Specification	Specifies the merchants table as the starting point of the join sequence, with an alias m for use in the query.
JOIN sell s ON m.mid = s.mid	Join Operation	Joins the merchants table to the sell table, linking records based on the merchant ID (mid).
join products p on s.pid = p.pid	Join Operation	Joins the sell table to the products table, linking products to their sales information using the product ID (pid).
join contain co on p.pid = co.pid	Join Operation	Joins the products table to the contain table, linking products to orders they are contained in.
join orders o on co.oid = o.oid	Join Operation	Joins the contain table to the orders table, establishing a connection between product containers and their respective orders.
join place pl on o.oid= pl.oid	Join Operation	Joins the orders table to the place table, completing the link from the original order placement to the actual order details.
GROUP BY m.name, EXTRACT(YEAR FROM pl.order_date)	Grouping	Groups the results by company name and year, for aggregate calculations.
ORDER BY m.name, EXTRACT(YEAR FROM pl.order_date)	Sorting	Orders the result set by company name and year, for orderly presentation.



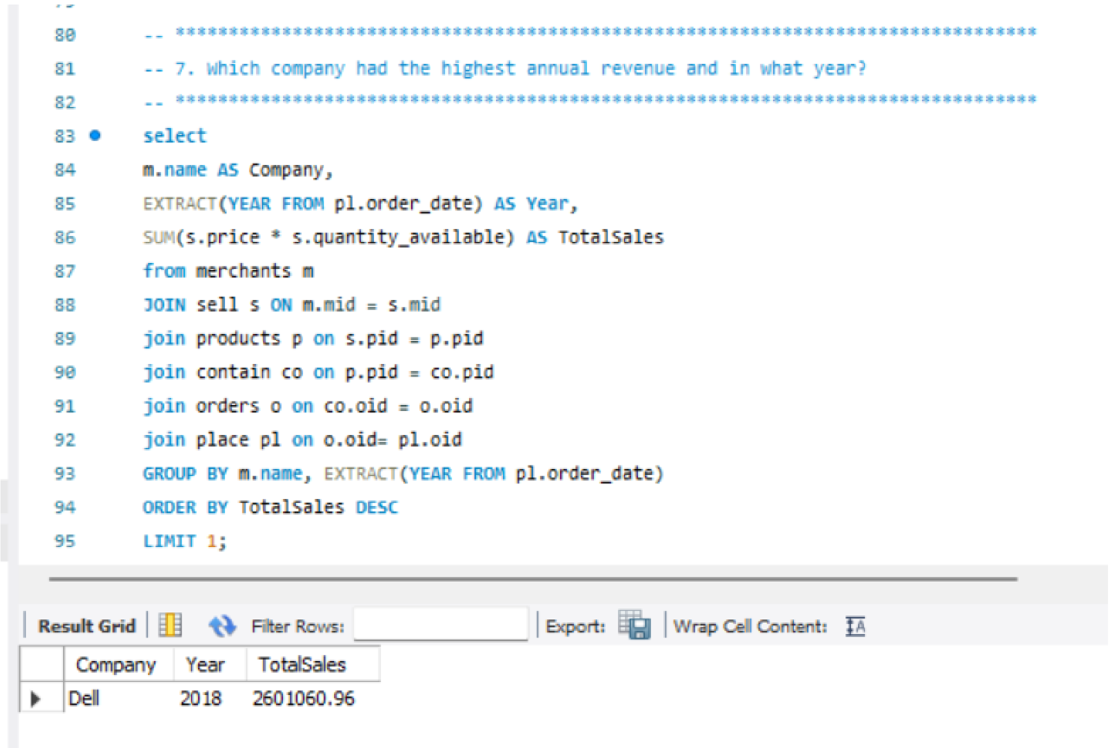
```sql
-- 6. List the annual total sales for each company (sort the results along the company and the year attributes).
-- ***************************************************************************
select
m.name AS Company,
EXTRACT(YEAR FROM pl.order_date) AS Year,
SUM(s.price * s.quantity_available) AS TotalSales
from merchants m
JOIN sell s ON m.mid = s.mid
join products p on s.pid = p.pid
join contain co on p.pid = co.pid
join orders o on co.oid = o.oid
join place pl on o.oid= pl.oid
GROUP BY m.name, EXTRACT(YEAR FROM pl.order_date)
ORDER BY m.name, EXTRACT(YEAR FROM pl.order_date);
```

Company	Year	TotalSales
Acer	2011	828677.08
Acer	2016	307909.83
Acer	2017	1100206.85
Acer	2018	1592886.58
Acer	2019	1180216.70
Acer	2020	1062622.30
Apple	2011	972240.92
Apple	2016	409402.38
Apple	2017	1071712.93
Apple	2018	1664629.77
Apple	2019	1311417.57
Apple	2020	1213964.96
Dell	2011	1542228.99
Dell	2016	625684.14
Dell	2017	1522794.28

7. Which company had the highest annual revenue and in what year?

SQL Component	Role in Query	Explanation
SELECT	Clause	Specifies the columns to be displayed in the query results.
m.name AS Company	Column Alias	Displays the merchant's name as "Company".
EXTRACT(YEAR FROM pl.order_date) AS Year	Function with Alias	Extracts the year from the order date and labels it as "Year".
SUM(s.price * s.quantity_available) AS TotalSales	Aggregated Column	Calculates the total sales by multiplying the price by the quantity available and labels it as "TotalSales".
FROM merchants m	Table Source	Identifies the table merchants and uses alias m for referencing in the query.
JOIN sell s ON m.mid = s.mid	JOIN Operation	Joins the sell table to merchants based on the merchant ID (mid).
JOIN products p ON s.pid = p.pid	JOIN Operation	Joins the products table to sell based on the product ID (pid).
JOIN contain co ON p.pid = co.pid	JOIN Operation	Joins the contain table to products based on the product ID (pid).
JOIN orders o ON co.oid = o.oid	JOIN Operation	Joins the orders table to contain based on the order ID (oid).
JOIN place pl ON o.oid = pl.oid	JOIN Operation	Joins the place table to orders based on the order ID (oid).

GROUP BY m.name, EXTRACT(YEAR FROM pl.order_date)	Grouping	Groups the results by company name and the year of the order date.
ORDER BY TotalSales desc	Ordering	Orders the results by the TotalSales in descending order.
LIMIT 1	Limit	Restricts the output to just one row, showing the company with the highest total sales for any year.

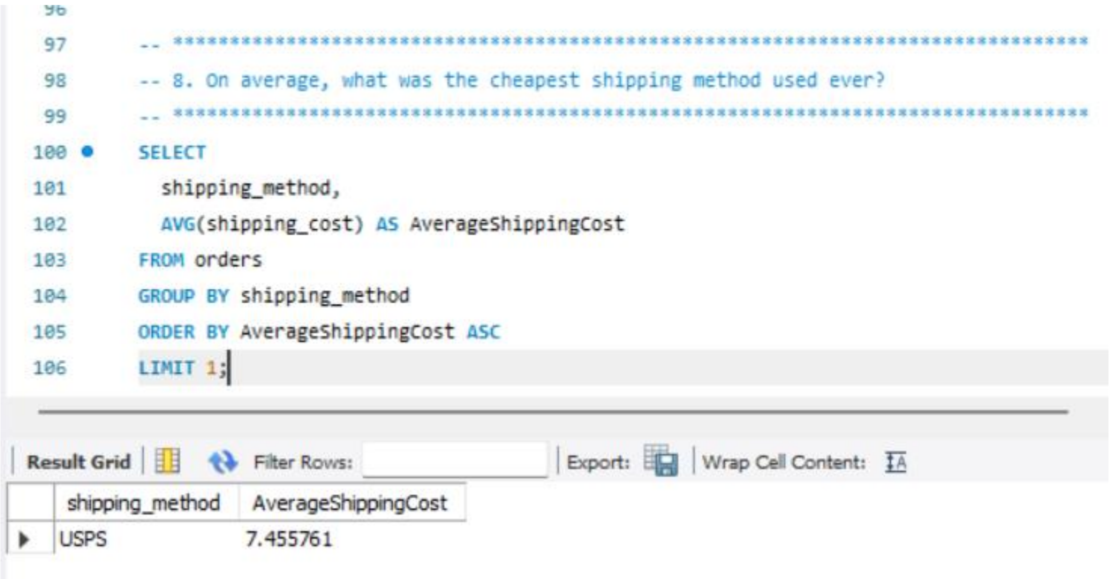


```
-- ********************************************************************************
-- 7. Which company had the highest annual revenue and in what year?
-- ********************************************************************************
select
m.name AS Company,
EXTRACT(YEAR FROM pl.order_date) AS Year,
SUM(s.price * s.quantity_available) AS TotalSales
from merchants m
JOIN sell s ON m.mid = s.mid
join products p on s.pid = p.pid
join contain co on p.pid = co.pid
join orders o on co.oid = o.oid
join place pl on o.oid= pl.oid
GROUP BY m.name, EXTRACT(YEAR FROM pl.order_date)
ORDER BY TotalSales DESC
LIMIT 1;
```

Company	Year	TotalSales
Dell	2018	2601060.96

8. On average, what was the cheapest shipping method used ever?

SQL Component	Role in Query	Explanation
SELECT	Clause	Specifies the columns to be displayed in the query results.
shipping_method	Column Selection	Specifies the shipping method to be included in the result set.
AVG(shipping_cost) AS AverageShippingCost	Aggregated Column with Alias	Calculates the average shipping cost and labels it as "AverageShippingCost".
FROM orders	Table Source	Identifies the source table orders from which to retrieve data.
GROUP BY shipping_method	Grouping	Groups the results by the shipping method.
ORDER BY AverageShippingCost ASC	Ordering	Orders the results by the average shipping cost in ascending order.
LIMIT 1	Limit	Restricts the output to just the lowest average shipping cost.

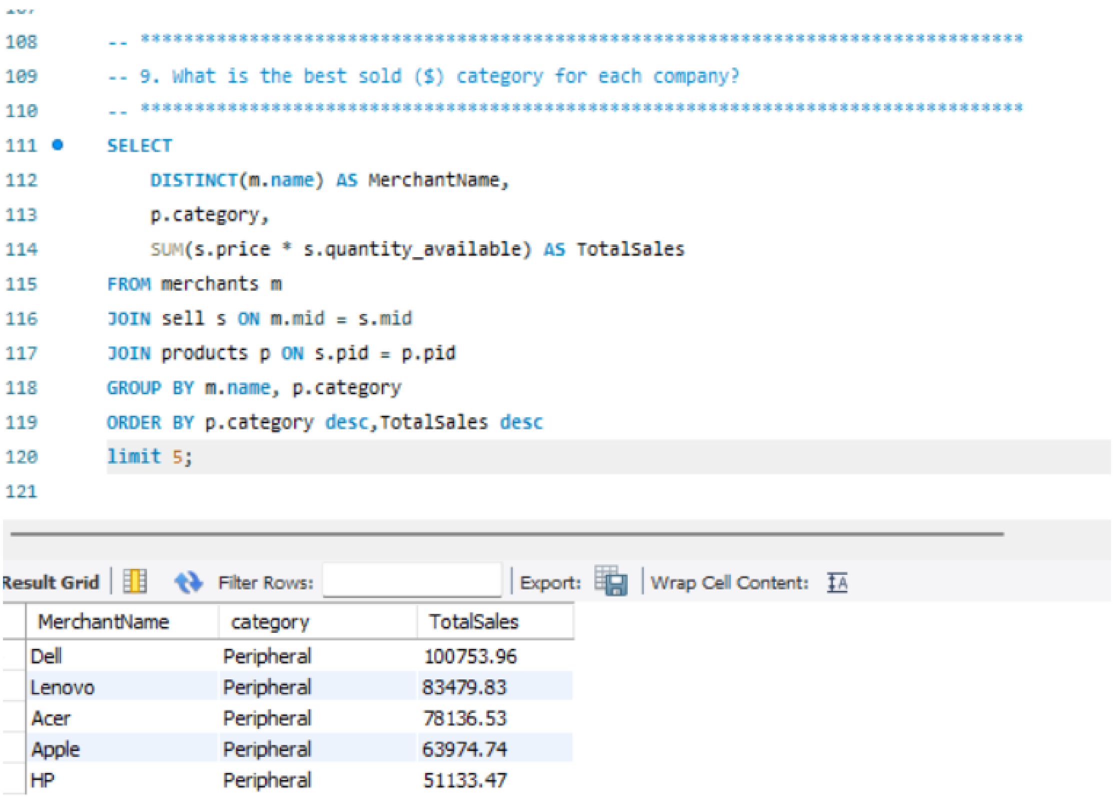


```
-- ********************************************************************************
-- 8. On average, what was the cheapest shipping method used ever?
-- ********************************************************************************
SELECT
  shipping_method,
  AVG(shipping_cost) AS AverageShippingCost
FROM orders
GROUP BY shipping_method
ORDER BY AverageShippingCost ASC
LIMIT 1;
```

	shipping_method	AverageShippingCost
▶	USPS	7.455761

9. What is the best sold ($) category for each company?

SQL Component	Role in Query	Explanation
SELECT	Clause	Specifies the columns to be displayed in the query results.
DISTINCT(m.name) AS MerchantName	Column Selection with Alias	Selects unique merchant names, labeled as "MerchantName".
p.category	Column Selection	Specifies the product category to be included in the results.
SUM(s.price * s.quantity_available) AS TotalSales	Aggregated Column with Alias	Calculates the total sales by multiplying the price by the quantity available, labeled as "TotalSales".
FROM merchants m	Table Source	Identifies 'merchants' as the source table, aliased as 'm'.
JOIN sell s ON m.mid = s.mid	Join	Joins the 'sell' table to 'merchants' on matching merchant IDs.
JOIN products p ON s.pid = p.pid	Join	Joins the 'products' table to 'sell' on matching product IDs.
GROUP BY m.name, p.category	Grouping	Groups the results by merchant name and product category.
ORDER BY p.category desc, TotalSales desc	Ordering	Orders the results first by product category in descending order, then by total sales also in descending order.
LIMIT 5	Limit	Limits the output to the top 5 results based on the specified order.



```
-- *********************************************************************************
-- 9. What is the best sold ($) category for each company?
-- *********************************************************************************
SELECT
    DISTINCT(m.name) AS MerchantName,
    p.category,
    SUM(s.price * s.quantity_available) AS TotalSales
FROM merchants m
JOIN sell s ON m.mid = s.mid
JOIN products p ON s.pid = p.pid
GROUP BY m.name, p.category
ORDER BY p.category desc,TotalSales desc
limit 5;
```

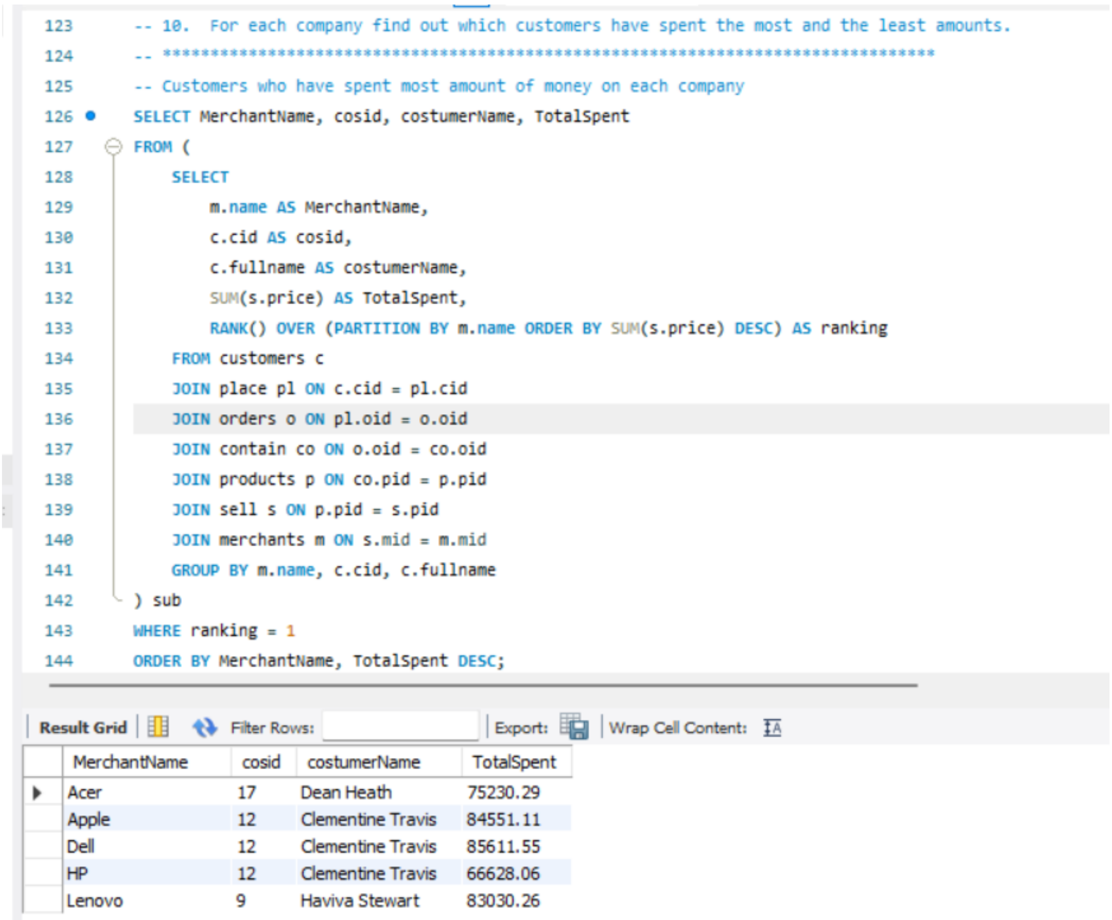
MerchantName	category	TotalSales
Dell	Peripheral	100753.96
Lenovo	Peripheral	83479.83
Acer	Peripheral	78136.53
Apple	Peripheral	63974.74
HP	Peripheral	51133.47

10. For each company find out which customers have spent the most and the least amounts.

SQL Query Breakdown

SQL Component	Role in Query	Explanation
SELECT	Clause	Specifies the columns to display in the final result.
MerchantName, cosid, costumerName, TotalSpent	Output Columns	Lists the columns to be included in the output: merchant name, customer ID, customer name, and total spent.
FROM (...) sub	Subquery with Alias	Uses a subquery to perform calculations, aliased as sub.
SELECT within subquery	Subquery Clause	Begins selection within the subquery.
m.name AS MerchantName, ...	Column Selection and Aliasing	In the subquery, selects and renames columns to identify the merchant, customer ID, customer name, and calculate total spent.
SUM(s.price) AS TotalSpent	Aggregated Column with Alias	Calculates total amount spent by each customer per merchant.
RANK() OVER (...) AS ranking	Window Function with Alias	Assigns a rank to each customer based on total spent, partitioned by merchant.

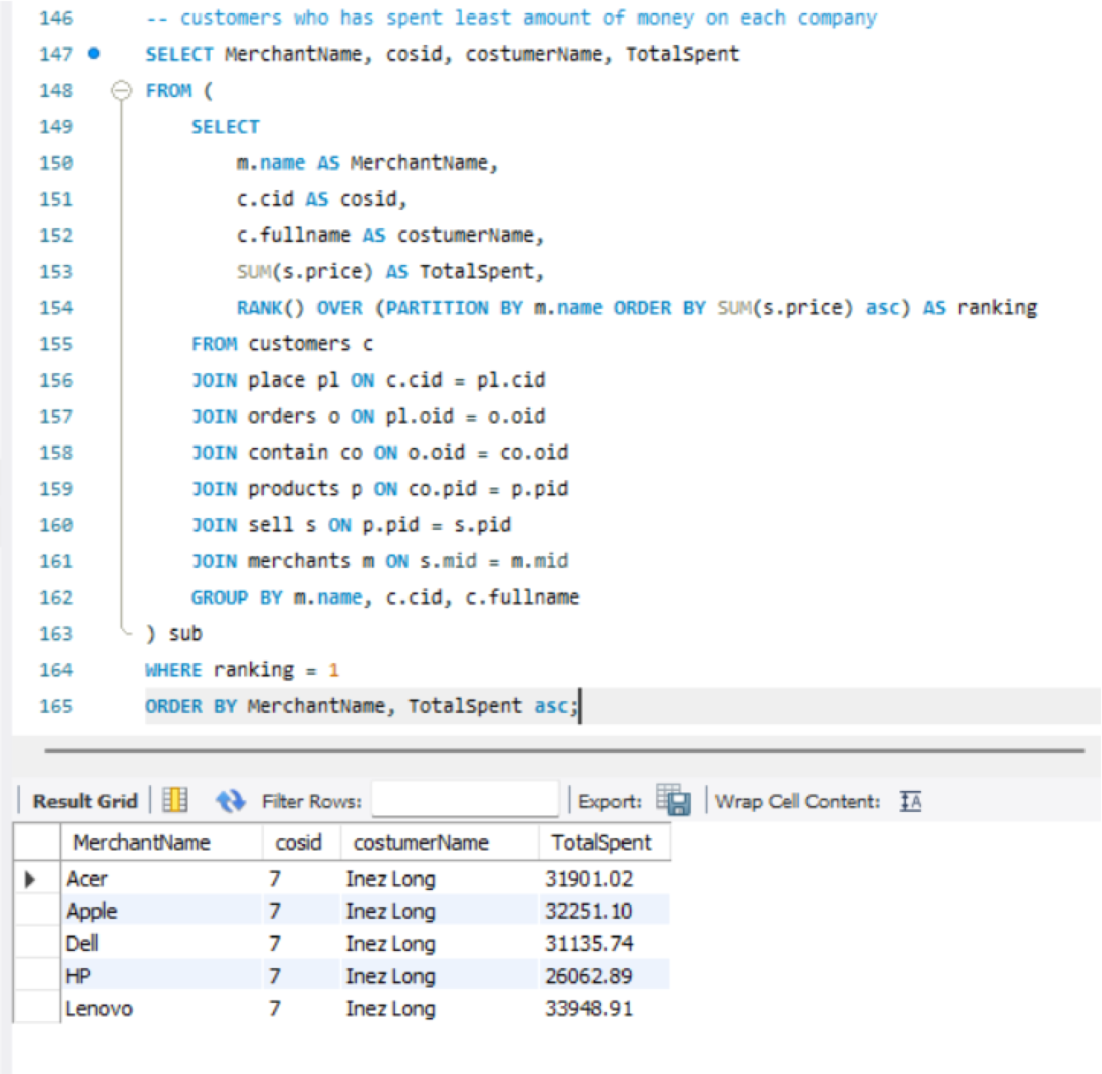
FROM customers c JOIN ...	Data Source and Joins	Specifies the tables and joins needed to gather all necessary information.
GROUP BY m.name, c.cid, c.fullname	Group By Clause	Groups the results by merchant name, customer ID, and customer name to prepare for the SUM operation.
WHERE ranking = 1	Filter Condition	Filters the subquery results to include only the top-ranked (highest spending) customers per merchant.
ORDER BY MerchantName, TotalSpent DESC	Ordering	Orders the final results by merchant name and the total spent in descending order.



```
-- 10.  For each company find out which customers have spent the most and the least amounts.
-- ********************************************************************************
-- Customers who have spent most amount of money on each company
SELECT MerchantName, cosid, costumerName, TotalSpent
FROM (
    SELECT
        m.name AS MerchantName,
        c.cid AS cosid,
        c.fullname AS costumerName,
        SUM(s.price) AS TotalSpent,
        RANK() OVER (PARTITION BY m.name ORDER BY SUM(s.price) DESC) AS ranking
    FROM customers c
    JOIN place pl ON c.cid = pl.cid
    JOIN orders o ON pl.oid = o.oid
    JOIN contain co ON o.oid = co.oid
    JOIN products p ON co.pid = p.pid
    JOIN sell s ON p.pid = s.pid
    JOIN merchants m ON s.mid = m.mid
    GROUP BY m.name, c.cid, c.fullname
) sub
WHERE ranking = 1
ORDER BY MerchantName, TotalSpent DESC;
```

MerchantName	cosid	costumerName	TotalSpent
Acer	17	Dean Heath	75230.29
Apple	12	Clementine Travis	84551.11
Dell	12	Clementine Travis	85611.55
HP	12	Clementine Travis	66628.06
Lenovo	9	Haviva Stewart	83030.26

10-2：Customer who has spent least amount of money on each company:
Similar to the find costomer who has spent most money on each companythe only diffrence is Totalspent in ascending order.



```sql
-- customers who has spent least amount of money on each company
SELECT MerchantName, cosid, costumerName, TotalSpent
FROM (
    SELECT
        m.name AS MerchantName,
        c.cid AS cosid,
        c.fullname AS costumerName,
        SUM(s.price) AS TotalSpent,
        RANK() OVER (PARTITION BY m.name ORDER BY SUM(s.price) asc) AS ranking
    FROM customers c
    JOIN place pl ON c.cid = pl.cid
    JOIN orders o ON pl.oid = o.oid
    JOIN contain co ON o.oid = co.oid
    JOIN products p ON co.pid = p.pid
    JOIN sell s ON p.pid = s.pid
    JOIN merchants m ON s.mid = m.mid
    GROUP BY m.name, c.cid, c.fullname
) sub
WHERE ranking = 1
ORDER BY MerchantName, TotalSpent asc;
```

MerchantName	cosid	costumerName	TotalSpent
Acer	7	Inez Long	31901.02
Apple	7	Inez Long	32251.10
Dell	7	Inez Long	31135.74
HP	7	Inez Long	26062.89
Lenovo	7	Inez Long	33948.91