

Award Sales-incentive

Bonuses:

Using SQL and Excel Viz

by

Michelle Moloney King
Data Analyst

Contents

| | |
|---------------------------------------|-----------|
| Aim: | 3 |
| Top 5 Sellers by Sales Amount: | 3 |
| Top 5 Sellers by Order Amount: | 4 |
| Order Quantity by Customers | 5 |
| My Process | 6 |
| SQL Process | 7 |
| About Michelle Moloney King | 15 |

Aim:

Title: Analysing Business Questions and Proposed Solutions: Implementing Sales Incentives for Top Performers

Summary:

This summary examines a business's objective of implementing a sales-incentive program to recognize and reward the top 5 sellers based on sales amount. The analysis also takes into consideration the top 5 sellers by quantity sold, along with a chart illustrating customer data and order amounts. By delving into these key aspects, the company aims to devise an effective sales-incentive strategy that aligns with its goals and motivates its top performers.

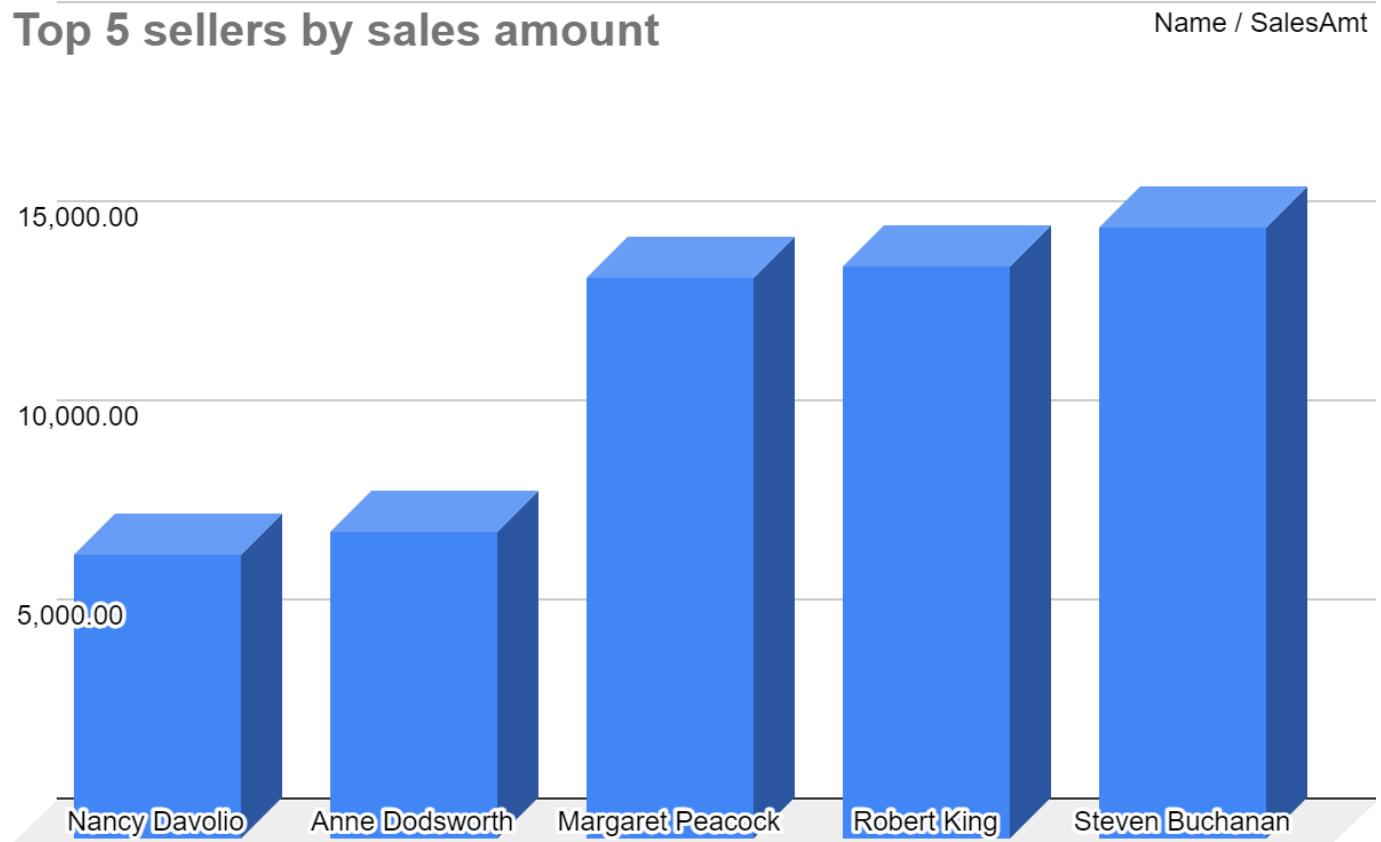
Expanded Analysis:

The business in question has recognized the significance of acknowledging and motivating its high-performing sales representatives. To achieve this, the company plans to implement a sales-incentive program that rewards the top 5 sellers based on sales amount. This approach seeks to encourage sales professionals to strive for excellence, boost overall sales performance, and foster a competitive spirit within the organisation.

To gain a comprehensive understanding of the sales landscape, it is essential to analyse both sales amount and quantity sold. By considering the top 5 sellers in terms of quantity sold, the company can identify individuals who consistently generate high sales volumes, even if their individual sales amounts may vary. This dual assessment ensures that both high-value and high-volume sales representatives are duly recognized and rewarded.

Studying the provided charts, which contains customer data and corresponding order amounts, offers valuable context for the incentive program. These charts allow for a deeper understanding of the customers' purchasing behaviours, enabling the company to tailor its sales strategies accordingly. By aligning the incentive program with customer preferences and order amounts, the business can encourage its sales team to focus on customer satisfaction, repeat purchases, and potentially upselling or cross-selling opportunities.

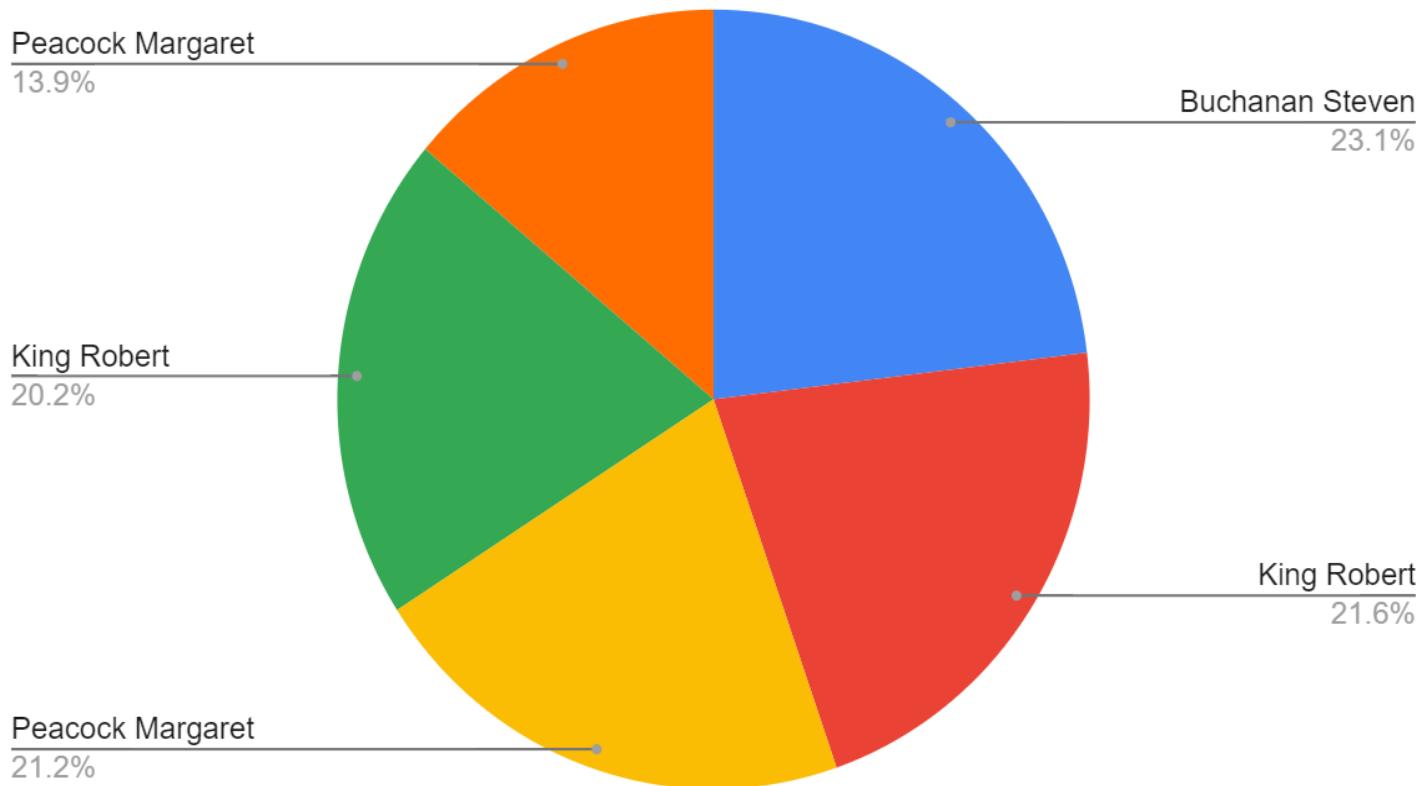
Top 5 Sellers by Sales Amount:



| Name | SalesAmt |
|------------------|-----------|
| Steven Buchanan | 15,353.60 |
| Robert King | 14,366.50 |
| Margaret Peacock | 14,104.00 |
| Anne Dodsworth | 7,698.45 |
| Nancy Davolio | 7,103.60 |

Top 5 Sellers by Order Amount:

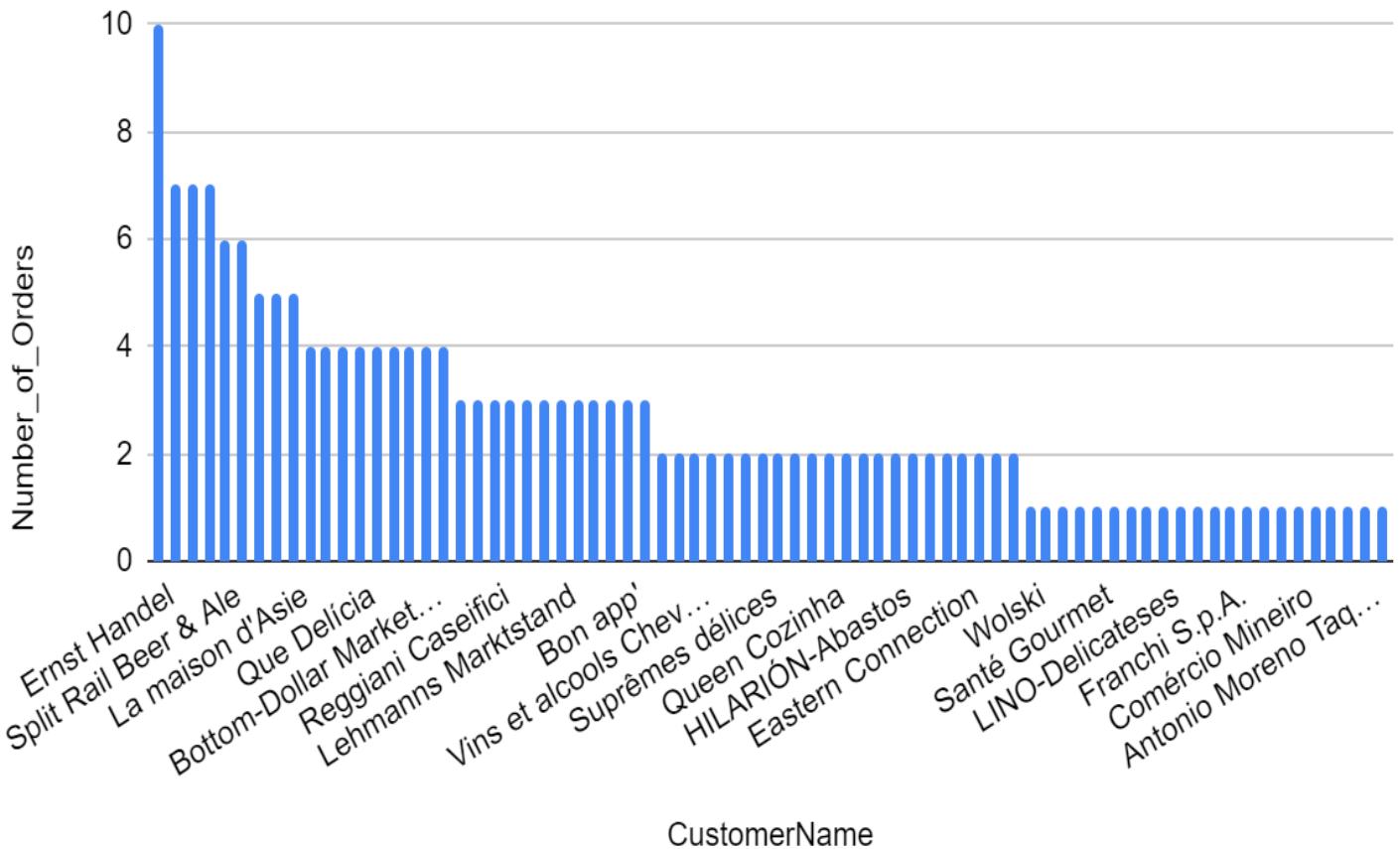
Top 5 sellers by order amount



| Name | SalesAmt | Quan. |
|------------------|-----------|-------|
| Buchanan Steven | 15,353.60 | 12 |
| King Robert | 14,366.50 | 60 |
| Peacock Margaret | 14,104.00 | 50 |
| King Robert | 13,427.00 | 12 |
| Peacock Margaret | 9,244.25 | 30 |

Order Quantity by Customers

How many orders have been placed by each customer.



My Process

Task: Utilising SQL Joins for Data Analysis

- SQL joins are employed to combine tables based on a shared column.
- The INNER JOIN command retrieves rows that have matching values in both joined tables.
- While SQL queries in the Tryit Editor are case-insensitive, proper spelling and punctuation are crucial.
- Aggregating and grouping data enhances its usefulness for decision-making purposes.
- The SUM() function, coupled with the GROUP BY clause, is utilised in SQL to aggregate data.
- Presenting the Data Solution.
- The LIMIT command in SQL enables the limitation of displayed rows in query results.
- The HAVING command in SQL filters data after aggregation.

SQL Process

To start I became familiar with the dataset. I used SELECT * to research each tables to figure out it's architecture.

To find out who deserved a pay increase... I had to look at the data in the Employees table, it gave education, name, ID but it had no sales or order data.

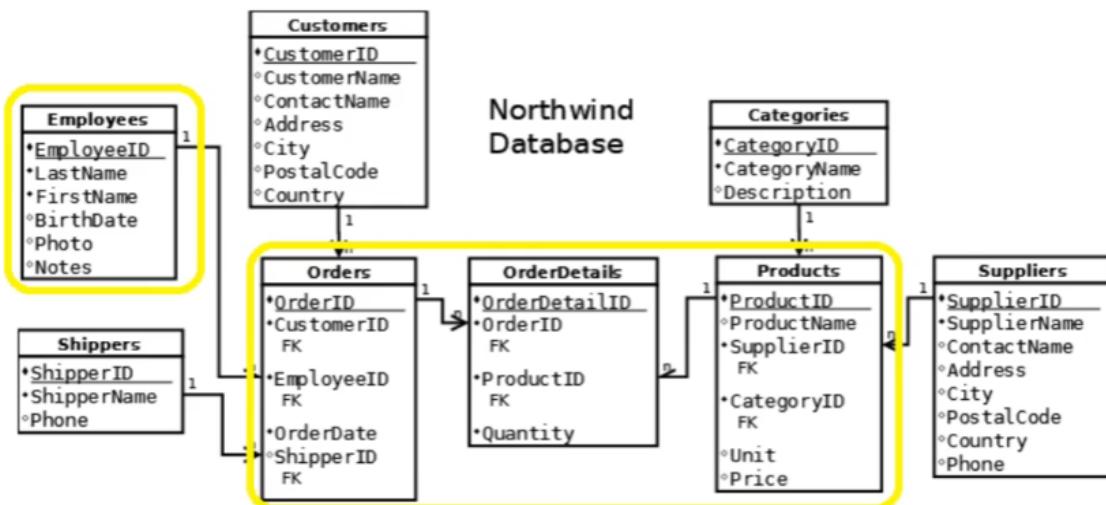
I then selected the orders table which had customerID (so I could link this to employees to get their names and orderid. But it contained no sales amount. I need the sales amount, and to see if the sales amount is also made up of several items.

In the order details table it showed me the quantity of products ordered but it did not contain the price of the items. So now I have to go to the products table where it lists the product id and the price.

Once I had the map of what tables had the data I needed and what linked them I was able to join them to create a new table with the data I needed.

Multiply the quantity from the orders details table form the price in the products table and then i can add together all the line items total prices and total sales to get the total sales for each order.

Database Tables Required

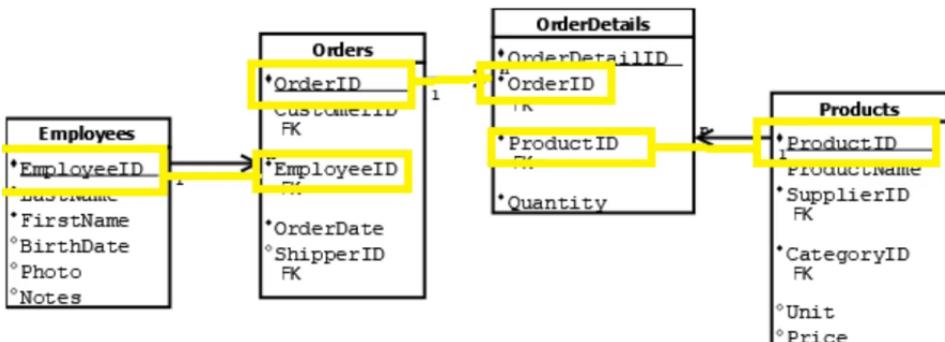


I need to link 4 tables to generate the data I need to answer the business question:

1. EMPLOYEES TABLE for first and last names.
2. ORDERS TABLE to get the orders with the highest sales.
3. ORDERDETAILS - I needed the quantity to calculate the sales value.
4. PRODUCTS TABLE - the price field to get the sales value.

Tables/Fields Needed

| Tables | Fields |
|--------------|------------------------|
| Employees | LastName and FirstName |
| Orders | OrderID |
| OrderDetails | ProductID and Quantity |
| Products | Price |



SQL code used:

```
SELECT lastname, firstname, orderid
from employees
inner join orders
on employees.employeeid = orders.employeeid
order by lastname,firstname
```

the above code generated this table -

SQL Statement: [Get your own SQL server](#)

```
SELECT lastname, firstname, orderid
from employees
inner join orders
on employees.employeeid = orders.employeeid
order by lastname,firstname
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

| LastName | FirstName | OrderID |
|----------|-----------|---------|
| Buchanan | Steven | 10248 |
| Buchanan | Steven | 10254 |
| Buchanan | Steven | 10269 |
| Buchanan | Steven | 10297 |
| Buchanan | Steven | 10320 |
| Buchanan | Steven | 10333 |
| Buchanan | Steven | 10358 |
| Buchanan | Steven | 10359 |
| Buchanan | Steven | 10372 |
| Buchanan | Steven | 10378 |
| Buchanan | Steven | 10397 |
| Callahan | Laura | 10262 |

Number of Records: 196

ENG GA 23:16 07/06/2023

Now I need to calculate the **sales amount** for the **top 5 orders**.

1. I need to join orders in **order** in **order table** and **order details tables** using the common column **order ID field**.
2. And need to join **order details** with the **products table** using their common column **products ID field**.

Codes use to execute the above needs:

```
SELECT LastName, FirstName, Orders.OrderID, Products.ProductID,
Quantity, Price
FROM employees
inner join orders
on employees.employeeID = orders.employeeid
inner join orderDetails
on orders.orderid = orderdetails.orderid
inner join products
on orderdetails.productid = products.productid
ORDER BY LastName, FirstName
```

Here is a snippet of the table generated:

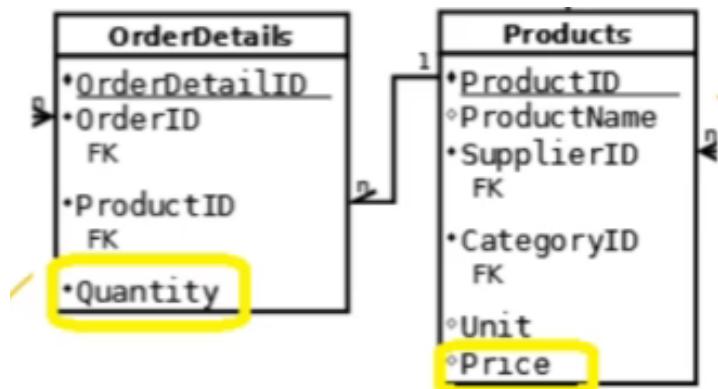
Result:

Number of Records: 518

| LastName | FirstName | OrderID | ProductID | Quantity | Price |
|----------|-----------|---------|-----------|----------|-------|
| Buchanan | Steven | 10248 | 11 | 12 | 21 |
| Buchanan | Steven | 10248 | 42 | 10 | 14 |
| Buchanan | Steven | 10248 | 72 | 5 | 34.8 |
| Buchanan | Steven | 10254 | 24 | 15 | 4.5 |
| Buchanan | Steven | 10254 | 55 | 21 | 24 |
| Buchanan | Steven | 10254 | 74 | 21 | 10 |
| Buchanan | Steven | 10269 | 33 | 60 | 2.5 |
| Buchanan | Steven | 10269 | 72 | 20 | 34.8 |
| Buchanan | Steven | 10297 | 39 | 60 | 18 |
| Buchanan | Steven | 10297 | 72 | 20 | 34.8 |
| Buchanan | Steven | 10320 | 71 | 30 | 21.5 |
| Buchanan | Steven | 10333 | 14 | 10 | 23.25 |

Let's take the first order as an example, Buchanan Steven has 3 different products within it but as we have the quantity and the price we can calculate the total amount.

I need to multiply the **Quantity field** in the orders **details table** with the **Price field** in the **Products table**.



To do this, I can add code into the code that joins the tables. At the end of select statement I put a comma after price and added **quantity * price as SalesAmount**

This generated a new field name for the calculation of quantity times price.

| LastName | FirstName | OrderID | ProductID | Quantity | Price | SalesAmt |
|----------|-----------|---------|-----------|----------|-------|----------|
| Buchanan | Steven | 10248 | 11 | 12 | 21 | 252 |
| Buchanan | Steven | 10248 | 42 | 10 | 14 | 140 |
| Buchanan | Steven | 10248 | 72 | 5 | 34.8 | 174 |
| Buchanan | Steven | 10254 | 24 | 15 | 4.5 | 67.5 |
| Buchanan | Steven | 10254 | 55 | 21 | 24 | 504 |
| Buchanan | Steven | 10254 | 74 | 21 | 10 | 210 |
| Buchanan | Steven | 10269 | 33 | 60 | 2.5 | 150 |
| Buchanan | Steven | 10269 | 72 | 20 | 34.8 | 696 |
| Buchanan | Steven | 10297 | 39 | 60 | 18 | 1080 |
| Buchanan | Steven | 10297 | 72 | 20 | 34.8 | 696 |
| Buchanan | Steven | 10320 | 71 | 30 | 21.5 | 645 |
| Buchanan | Steven | 10333 | 14 | 10 | 23.25 | 232.5 |
| Buchanan | Steven | 10333 | 21 | 10 | 10 | 100 |
| Buchanan | Steven | 10333 | 71 | 40 | 21.5 | 860 |
| Buchanan | Steven | 10358 | 24 | 10 | 4.5 | 45 |
| Buchanan | Steven | 10358 | 34 | 10 | 14 | 140 |
| Buchanan | Steven | 10358 | 36 | 20 | 19 | 380 |

Now I need to aggregate data using the sum function. I do this by changing the last line of SQL from ORDER BY LastName, FirstName to GROUP BY orders.orderid

I now need a summarised value of the sales amount. I do this by changing my select calculation with a prefix of sum and putting quantity * price into brackets.

When I run this code

```
SELECT LastName, FirstName, Orders.OrderID,
Products.ProductID,
Quantity, Price, sum(quantity * price) as SalesAmt
FROM employees
inner join orders
on employees.employeeID = orders.employeeid
inner join orderDetails
on orders.orderid = orderdetails.orderid
inner join products
on orderdetails.productid = products.productid
GROUP BY orders.orderid
```

the data goes from 518 records to 196 records. One row for each order.

SQL Statement:

```
SELECT LastName, FirstName, Orders.OrderID, Products.ProductID,
Quantity, Price, sum(quantity * price) as SalesAmt
FROM employees
inner join orders
on employees.employeeID = orders.employeeid
inner join orderDetails
on orders.orderid = orderdetails.orderid
inner join products
on orderdetails.productid = products.productid
GROUP BY orders.orderid
```

Get your own SQL server

Run SQL »

Result:

Number of Records: 196

| LastName | FirstName | OrderID | ProductID | Quantity | Price | SalesAmt |
|-----------|-----------|---------|-----------|----------|-------|----------|
| Buchanan | Steven | 10248 | 11 | 12 | 21 | 566 |
| Suyama | Michael | 10249 | 14 | 9 | 23.25 | 232.5 |
| Peacock | Margaret | 10250 | 41 | 10 | 9.65 | 2267.25 |
| Leverling | Janet | 10251 | 22 | 6 | 21 | 839.5 |
| Peacock | Margaret | 10252 | 20 | 40 | 81 | 4662.5 |
| Leverling | Janet | 10253 | 31 | 20 | 12.5 | 1806 |
| Buchanan | Steven | 10254 | 24 | 15 | 4.5 | 781.5 |
| Dodsworth | Anne | 10255 | 2 | 20 | 19 | 3115.75 |
| Leverling | Janet | 10256 | 53 | 15 | 32.8 | 648 |
| Peacock | Margaret | 10257 | 27 | 25 | 43.9 | 1400.5 |
| Davolio | Nancy | 10258 | 2 | 50 | 19 | 2529.75 |
| Peacock | Margaret | 10259 | 21 | 10 | 10 | 126 |

I now sort the data in descending order by sales amount with this code, which is placed at the end of the SQL code:
order by salesamt desc

SQL Statement:

```
inner join orderDetails
    on orders.orderid = orderdetails.orderid
inner join products
    on orderdetails.productid = products.productid
GROUP BY orders.orderid
order by salesamt desc
```

Get your own SQL server

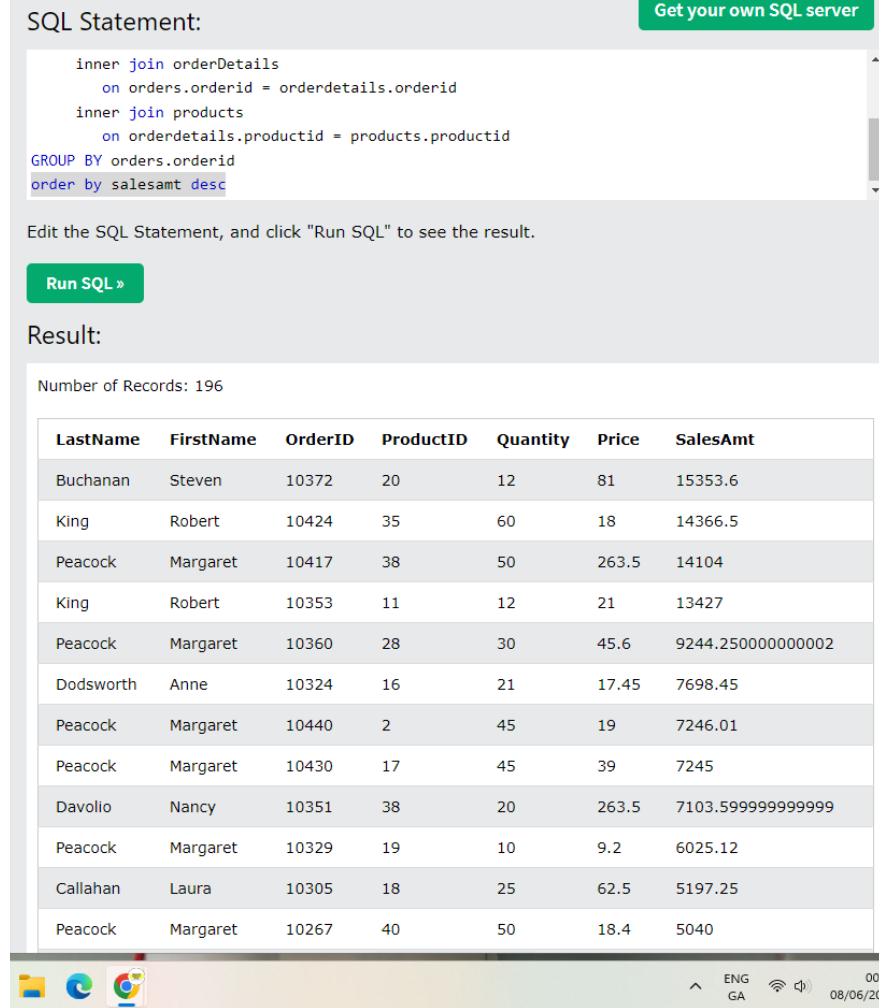
Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 196

| Last Name | First Name | Order ID | Product ID | Quantity | Price | Sales Amt |
|-----------|------------|----------|------------|----------|-------|-------------------|
| Buchanan | Steven | 10372 | 20 | 12 | 81 | 15353.6 |
| King | Robert | 10424 | 35 | 60 | 18 | 14366.5 |
| Peacock | Margaret | 10417 | 38 | 50 | 263.5 | 14104 |
| King | Robert | 10353 | 11 | 12 | 21 | 13427 |
| Peacock | Margaret | 10360 | 28 | 30 | 45.6 | 9244.250000000002 |
| Dodsworth | Anne | 10324 | 16 | 21 | 17.45 | 7698.45 |
| Peacock | Margaret | 10440 | 2 | 45 | 19 | 7246.01 |
| Peacock | Margaret | 10430 | 17 | 45 | 39 | 7245 |
| Davolio | Nancy | 10351 | 38 | 20 | 263.5 | 7103.599999999999 |
| Peacock | Margaret | 10329 | 19 | 10 | 9.2 | 6025.12 |
| Callahan | Laura | 10305 | 18 | 25 | 62.5 | 5197.25 |
| Peacock | Margaret | 10267 | 40 | 50 | 18.4 | 5040 |



I'll now limit the data to 5 rows. Using SQL - Limit 5 (at the end of the code)

Result:

Number of Records: 5

| Last Name | First Name | Order ID | Product ID | Quantity | Price | Sales Amt |
|-----------|------------|----------|------------|----------|-------|-------------------|
| Buchanan | Steven | 10372 | 20 | 12 | 81 | 15353.6 |
| King | Robert | 10424 | 35 | 60 | 18 | 14366.5 |
| Peacock | Margaret | 10417 | 38 | 50 | 263.5 | 14104 |
| King | Robert | 10353 | 11 | 12 | 21 | 13427 |
| Peacock | Margaret | 10360 | 28 | 30 | 45.6 | 9244.250000000002 |

Analysing

After analysing we see that some employees have sold more expensive orders and so that need to be in the top five. My first report gave a **list of orders with the five highest order amounts**.

And so I need to run another report with a **list of the five employees with the highest sales amounts**. Employee IDs: (10372, 10424, 10417, 10324, 10351)

I will add this code right above ORDER BY

```
having orders.orderid in (10372, 10424, 10417, 10324, 10351)
```

Result:

Number of Records: 5

| LastName | FirstName | OrderID | ProductID | Quantity | Price | SalesAmt |
|-----------|-----------|---------|-----------|----------|-------|-------------------|
| Buchanan | Steven | 10372 | 20 | 12 | 81 | 15353.6 |
| King | Robert | 10424 | 35 | 60 | 18 | 14366.5 |
| Peacock | Margaret | 10417 | 38 | 50 | 263.5 | 14104 |
| Dodsworth | Anne | 10324 | 16 | 21 | 17.45 | 7698.45 |
| Davolio | Nancy | 10351 | 38 | 20 | 263.5 | 7103.599999999999 |

How many orders have been placed by each customer.

1. To answer this I'll join **Orders table** and **Customer Table** using their common column **Customer ID**.
2. Aggrate the data by customerID
3. and use the COUNT() function to count the number of orders for each customer.

SQL used

```
SELECT CustomerName, count(OrderID) as Number_of_Orders from Customers
inner join Orders
on Customers.CustomerID = Orders.CustomerID
group by Customers.CustomerID
order by Number_of_Orders desc
```

Since OrderID is a primary key I was able to add a count function to ut as each ORDERID represents 1. I gave the new data that I was generating, a new field to make it easier to read. And grouped by number of orders, descending, to give more context.

I brought this data into excel and created charts with excel to visualise the data.

| CustomerName | Number_of_Orders |
|------------------------------|------------------|
| Ernst Handel | 10 |
| Wartian Herkku | 7 |
| Rattlesnake Canyon Grocery | 7 |
| QUICK-Stop | 7 |
| Split Rail Beer & Ale | 6 |
| Hungry Owl All-Night Grocers | 6 |
| Mère Paillardre | 5 |
| LILA-Supermercado | 5 |
| La maison d'Asie | 5 |
| Die Wandernde Kuh | 4 |
| Tortuga Restaurante | 4 |
| Save-a-lot Markets | 4 |

About Michelle Moloney King



Michelle King, based in Co. Tipperary, Ireland, is a skilled professional with a diverse background in teaching and data analytics. With a passion for leveraging data to drive strategic decision-making, Michelle is now seeking to transition from her teaching career into the field of data analytics.

Michelle's website showcases her expertise in SQL, Tableau, Python data analytics, data visualisation, pivot tables, Excel, visual communication, and design. Her strong analytical skills, problem-solving abilities, and customer-focused service make her a valuable asset in the field of data analytics.

Having a degree in Tech and having completed the Google Data Analytics Professional Certificate on Coursera, Michelle has gained in-depth knowledge in data analytics methodologies and tools. Additionally, her Google UX Design Professional Certificate demonstrates her understanding of user-centric design principles.

Now, with a strong foundation in data analytics and a proven track record of leveraging data to drive results, Michelle is eager to transition into a career in data analytics. Her diverse skill set, passion for data-driven insights, and ability to handle multiple projects make her a valuable candidate in the field of data analytics.

Links:

[Linkedin](#)

[Twitter](#)

[Website](#)