Project report
Data Science DW&DV

group # 3/35
Linghua Lai
Muhammad Waheed ud din Siddiqui

This document contains .jpg images of the work and results that were done in Project for this course. There is also a description of each step for easy comprehension and grading purposes.

For the project, we used "Sales developer dataset". This data is about a company that offers sales of products to customer. The data is a record of the transaction of sales made through this company between July 2006 to July 2009.

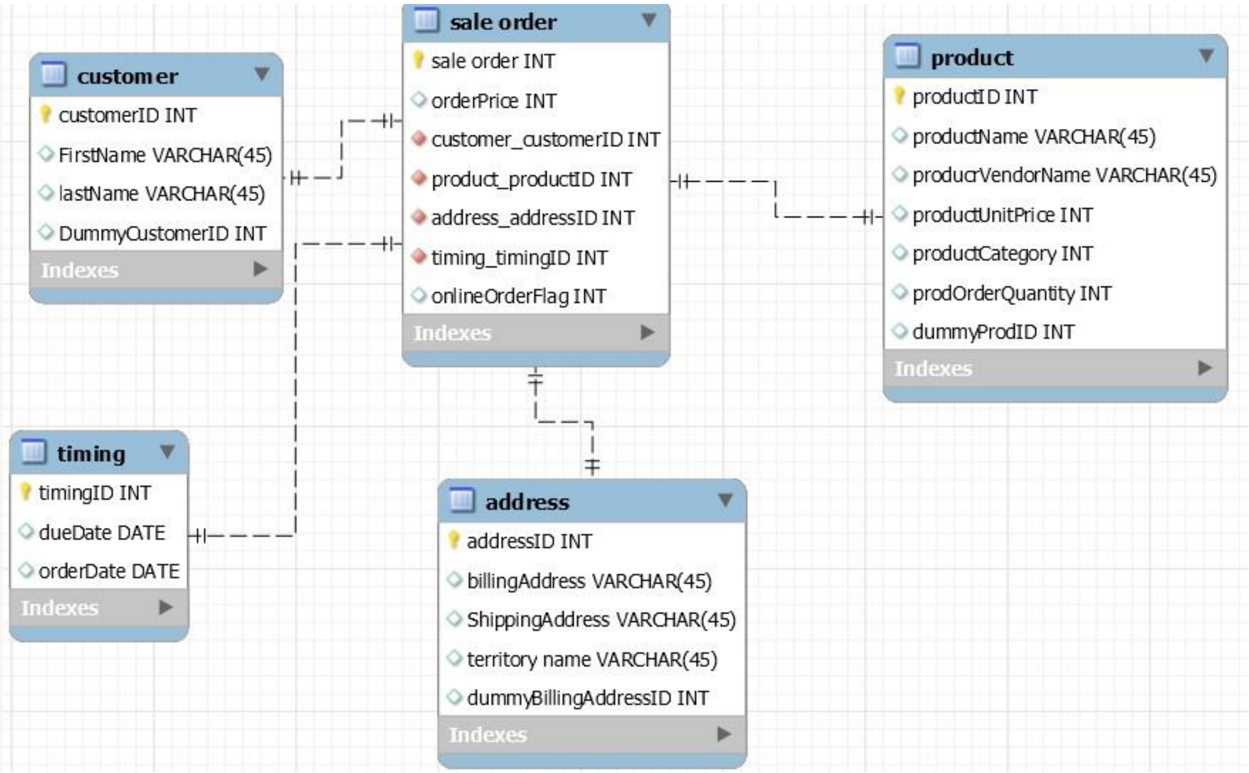
Business questions:

After looking at the data in Microsoft's Access program, we came up with following business questions:

- Which vendor is able to make most sales through the sale services of the company?
- What is the average delivery time of sale orders that were made online?
- Which product category is most famous in different territories in the world?
- The stats of online sales vs. offline sales made by the company

In order to answer these questions, we moved on to make a relational database model using MySQL workbench.

Relational database:



sale order
- sale order INT
- orderPrice INT
- customer_customerID INT
- product_productID INT
- address_addressID INT
- timing_timingID INT
- onlineOrderFlag INT

customer
- customerID INT
- FirstName VARCHAR(45)
- lastName VARCHAR(45)
- DummyCustomerID INT

product
- productID INT
- productName VARCHAR(45)
- producrVendorName VARCHAR(45)
- productUnitPrice INT
- productCategory INT
- prodOrderQuantity INT
- dummyProdID INT

timing
- timingID INT
- dueDate DATE
- orderDate DATE

address
- addressID INT
- billingAddress VARCHAR(45)
- ShippingAddress VARCHAR(45)
- territory name VARCHAR(45)
- dummyBillingAddressID INT

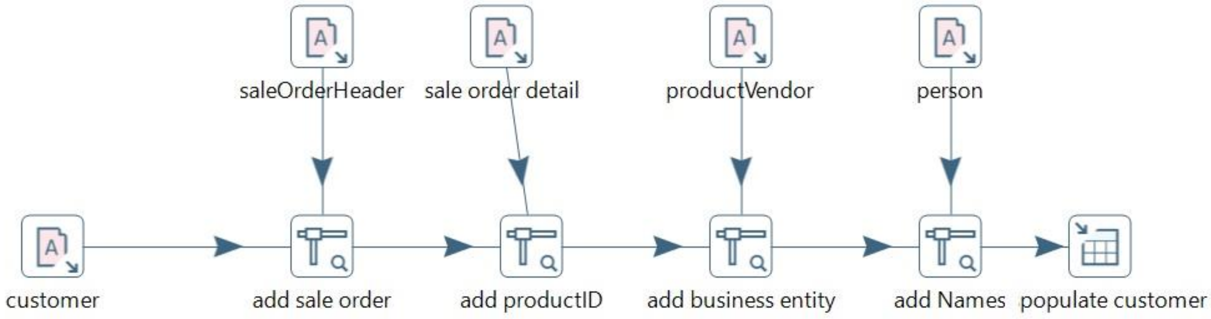
This design is a star schema. The fact table is the “sale order” table. Rest of the tables are dimension tables.

We moved on to populate the tables from the database by making transformations.

ETL:

Customer:

First the customer table was filled. Following is the transformation that made it possible:

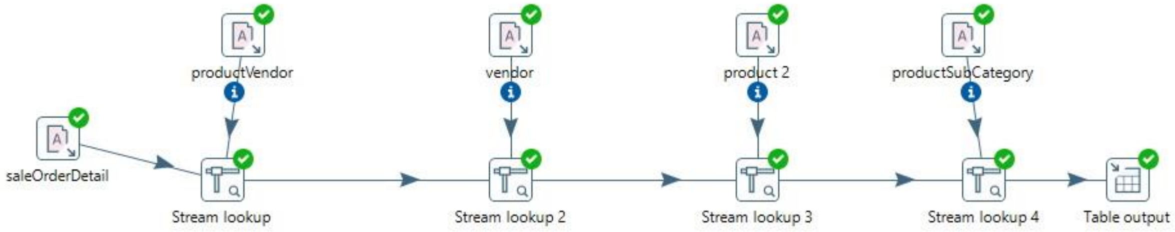


In the data the customer table had a field “person ID” and there was another table, Person table that had names of the persons. It was understood that the names correspond to the customers so we wanted to link every customer with its name. For this purpose 4 lookups were executed because there was no direct link from customer to person table.

Firstly the customerID in customer table was matched with customerID in SaleOrderHeader and the sale order for every customer was streamed out from saleOrderHeader table. This made sure that we are only looking into orders made by unique customers. Secondly, the productID were added by looking up the product sold in each order. That productID was matched with productID in ProductVendor table to add businessEntityID with every customer. Now every customer could be looked up in the person table using business entity ID. That is how we added person names to each customer. In the RDB Customer's table primary key "customerID" was set to auto-increment. The actual customerID from every order in the database was also recorded in the field "dummy customer ID". This will be used later in order to refer every customer to its sale order.

A key problem was observed after this transformation was carried out and data was observed. The customer table has no link to the person table so the person's names were all NULL.

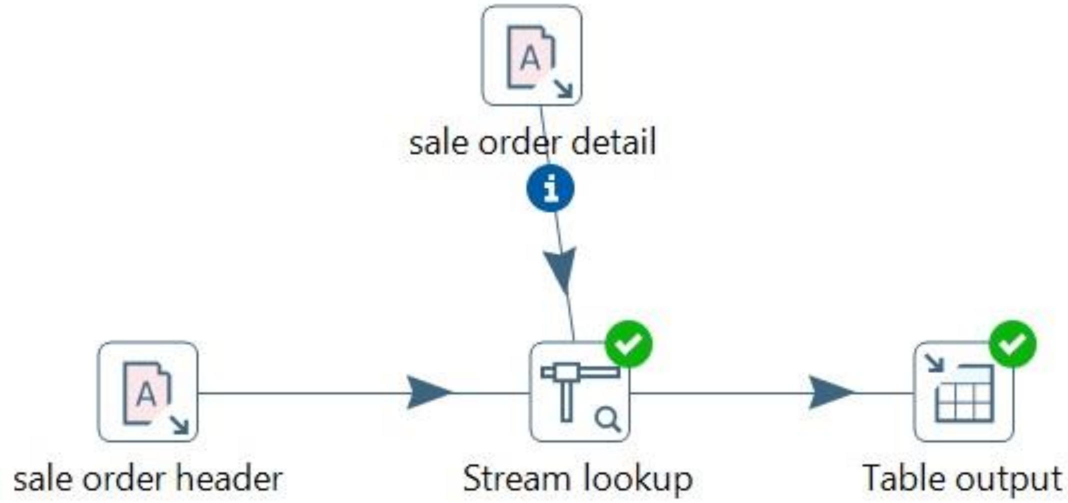
Product:



To populate products we needed product's vendor name and product's name among other details for sale order like order quantity and product price. Firstly the BusinessEntityID for every product was streamed by looking up productID in saleOrderDetail and prodcutVendor. On the bases of this businessEntityID the vendor's name was added to the flow. Products name was then added to the stream from the product table. Also the product subcategoryID was recorded so that it can be used to lookup category names from the productSubCategory table. Lastly the subcategory name was added by looking up subcategoryID and the product table in the RDB was populated.

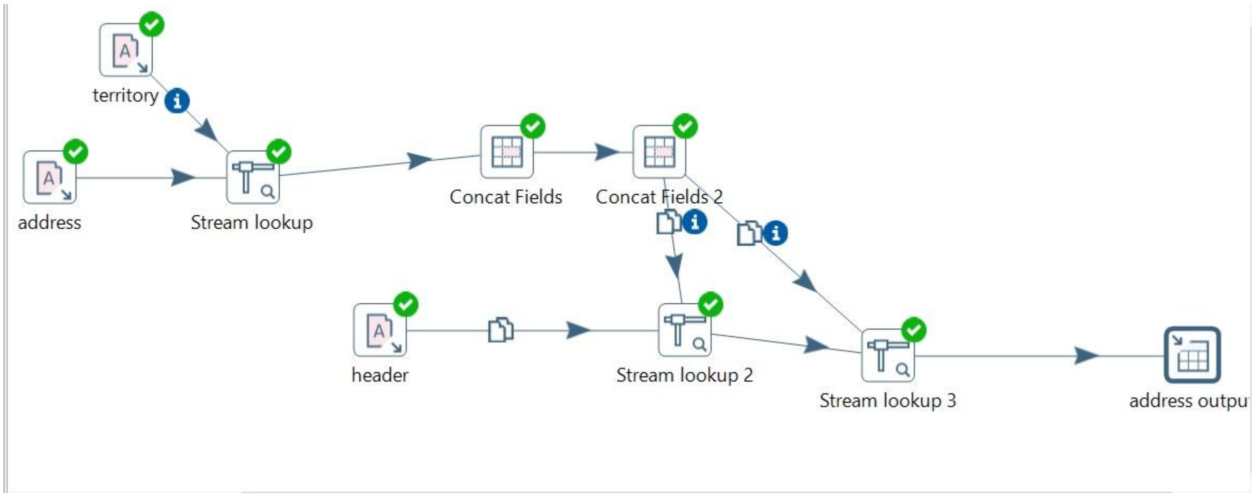
The productID in the RDB was set to auto-increment and the record of original productIDs from data was recorded in "dummyProdID" for syncing each value at the end. Another important point to notice is that we had to record product of every sale order. This means there is repetition of a lot of products in the product table. This happened because we kept the field "prodOrderQuantity" in the dimension table. This field should have been in the fact table in order to avoid any repetition in product entries in the product table.

Timing:



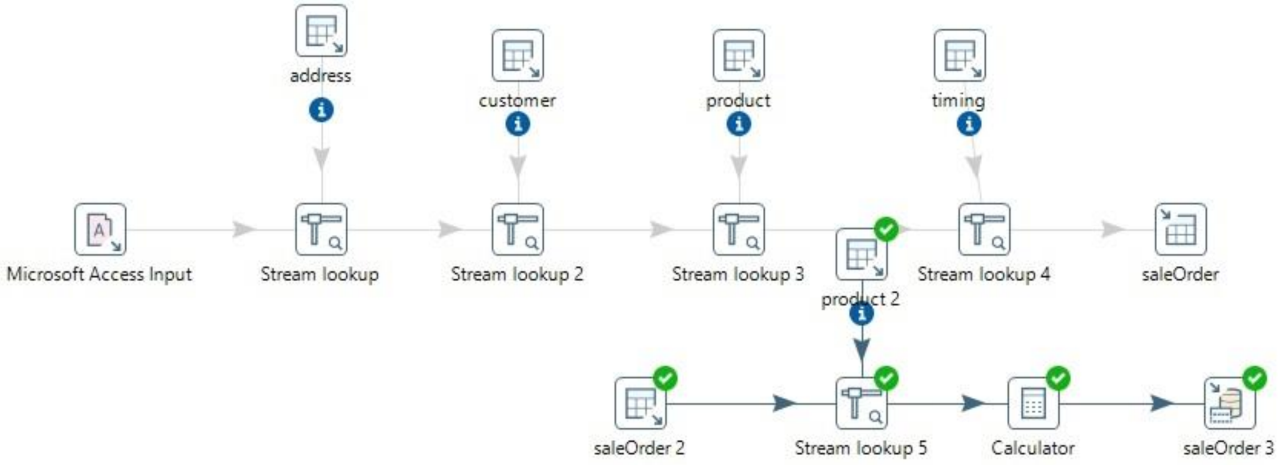
For every order the due date was in the sale order detail table while the order date was in sale order header table. The two tables were lookedup on the basis of sale order ID and the due dates were streamed in the data. Finally the timing table was populated. The record of sale order ID was also kept in the timing table so that it becomes easy to populate the fact table later.

Address:



The address table in the data had a field territoryID which was looked up with territory table and the name of territory was added to each address. After that the various fields of address (city, postal code, state etc.) were concatenated to make one long address with datatype: string. The address ID was then lookedup with the shipping addressID and billing addressID of each order in the sale header and the addresses of these IDs were streamed with the main flow. billingAddressId of the orders was also recorded for making the population of fact table easy.

Fact table:

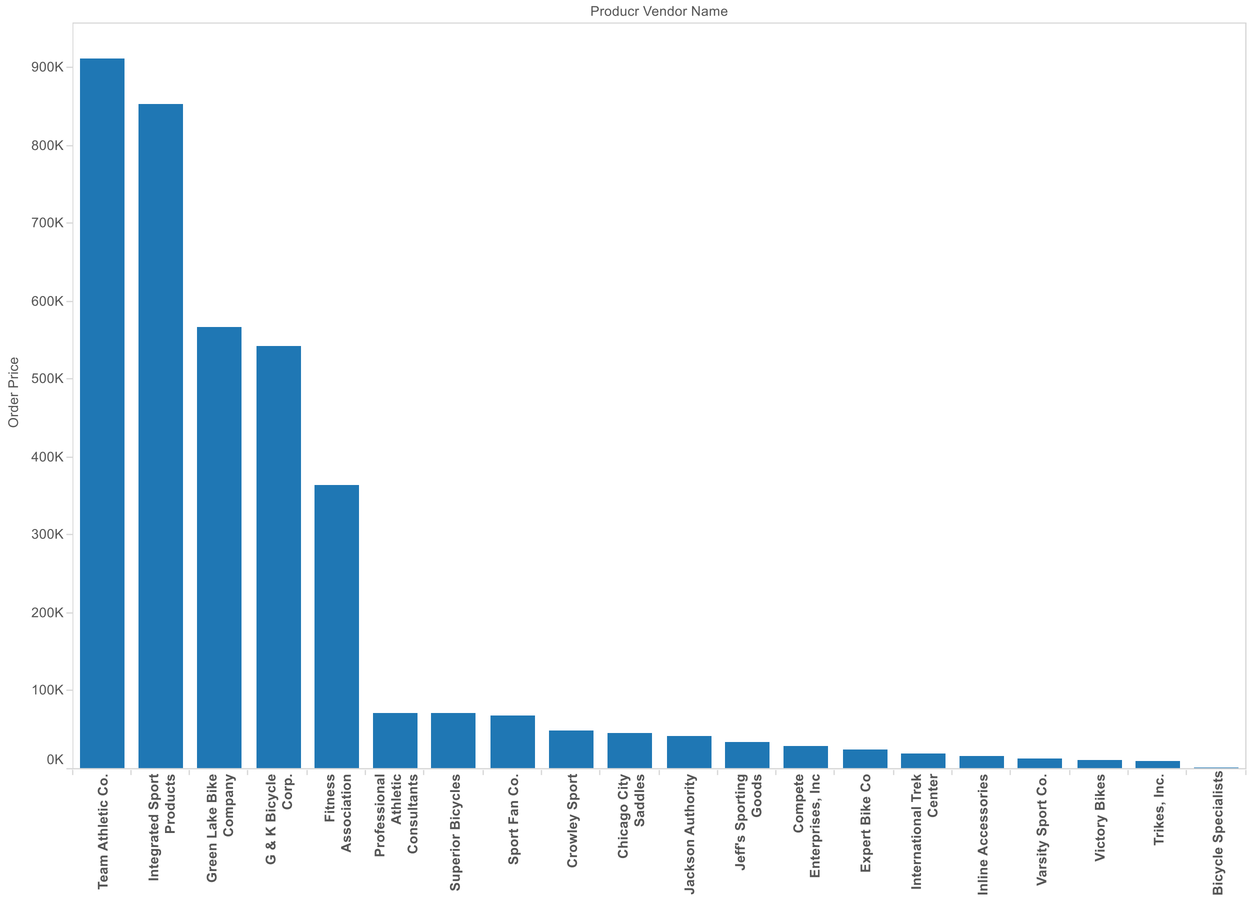


To populate the fact table, the sale order detail table was taken from the data and primary keys of the four dimension tables (address, customer, product and timing) were added to it. In the first lookup, billingAddressID was used to map every order with the addressID of the dimension table address. For syncing with customer table, CustomerID and dummyCustomerID fields were lookedup. Similarly productID and dummyProductID for Product table while no lookup was made for Timing table. The flag onlineOrderFlag was also taken from the sale order table in the original data. Wherever the value was “0” it was considered as an offline order and if there was any other value, it was considered to be online order. The PrimaryKeys of the dimension tables were added to the main fact table “sale Order”. In order to give value to “order price” field in the fact table, a calculation was made by multiplying productOrderQuantity and prodUnitPrice in the product dimension table and the value was filled in sale order with respect to productID key.

Business Intelligence:

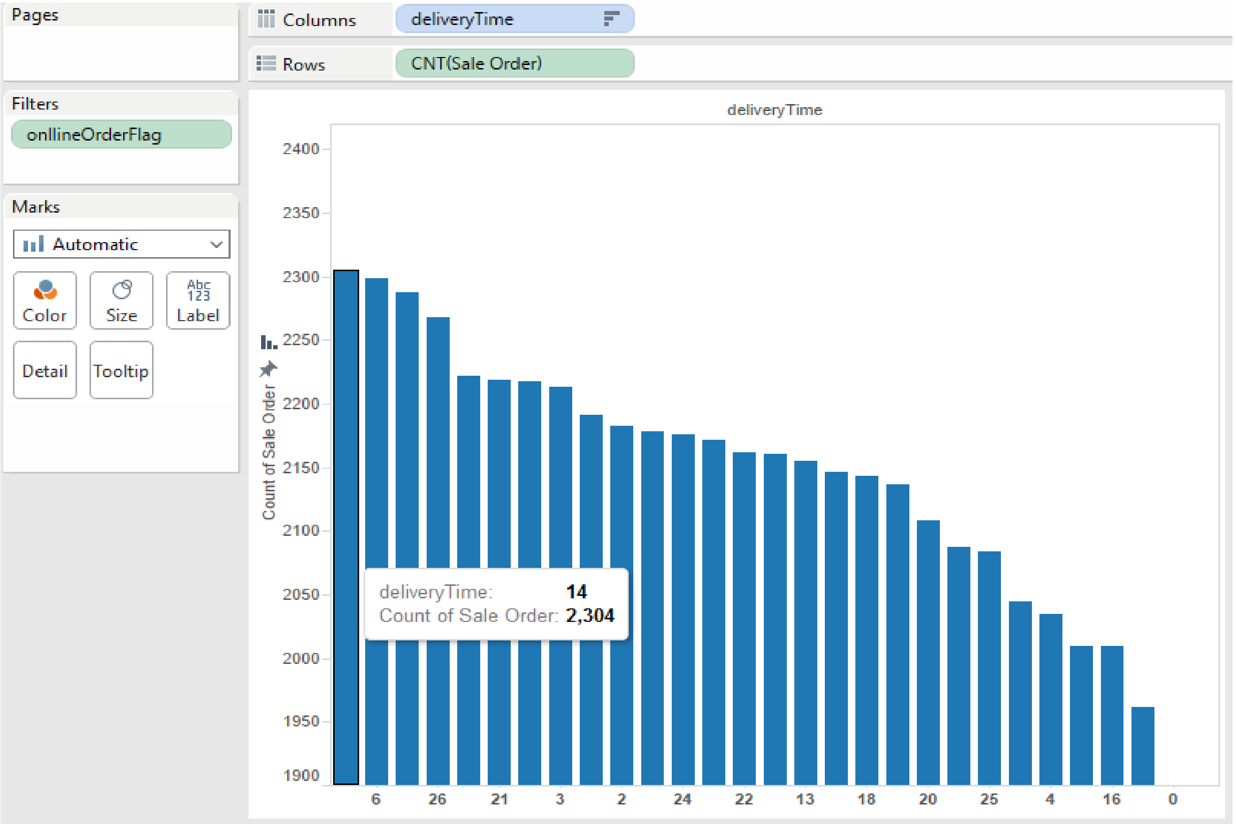
The first question was: Which vendor is able to make most sales through the sale services of the company? The result was obtained by summing up all order prices for every product vendor name throughout the data.

Sheet 1



Sum of Order Price for each Producr Vendor Name. The view is filtered on Producr Vendor Name, which excludes Null.

The second question was what is the average delivery time of sale orders that were made online? The overview of the delivery time is shown in the graph below:



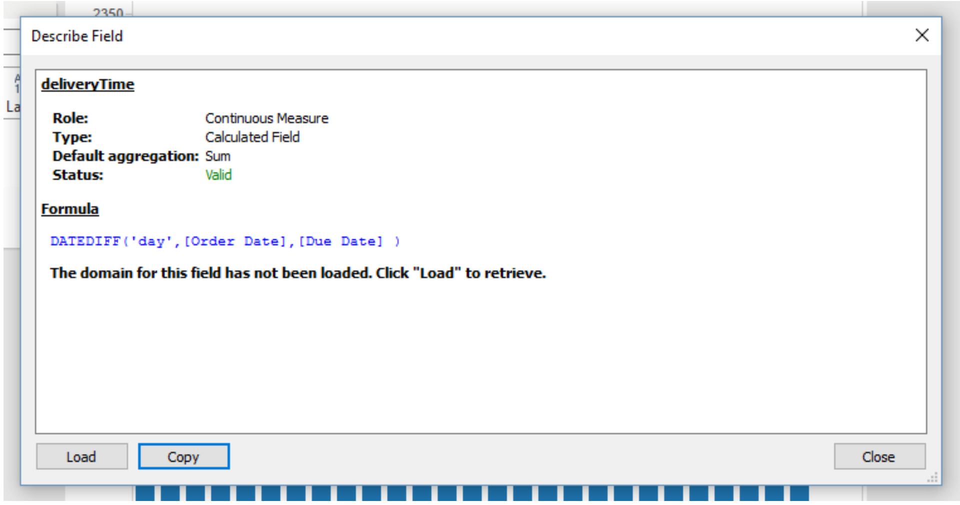
Columns: deliveryTime

Rows: CNT(Sale Order)

Filters: onllineOrderFlag

deliveryTime: 14
Count of Sale Order: 2,304

This was obtained by counting all the orders that have same delivery time in days. Delivery time is a calculation described in tableau as:



deliveryTime

Role: Continuous Measure
Type: Calculated Field
Default aggregation: Sum
Status: Valid

Formula

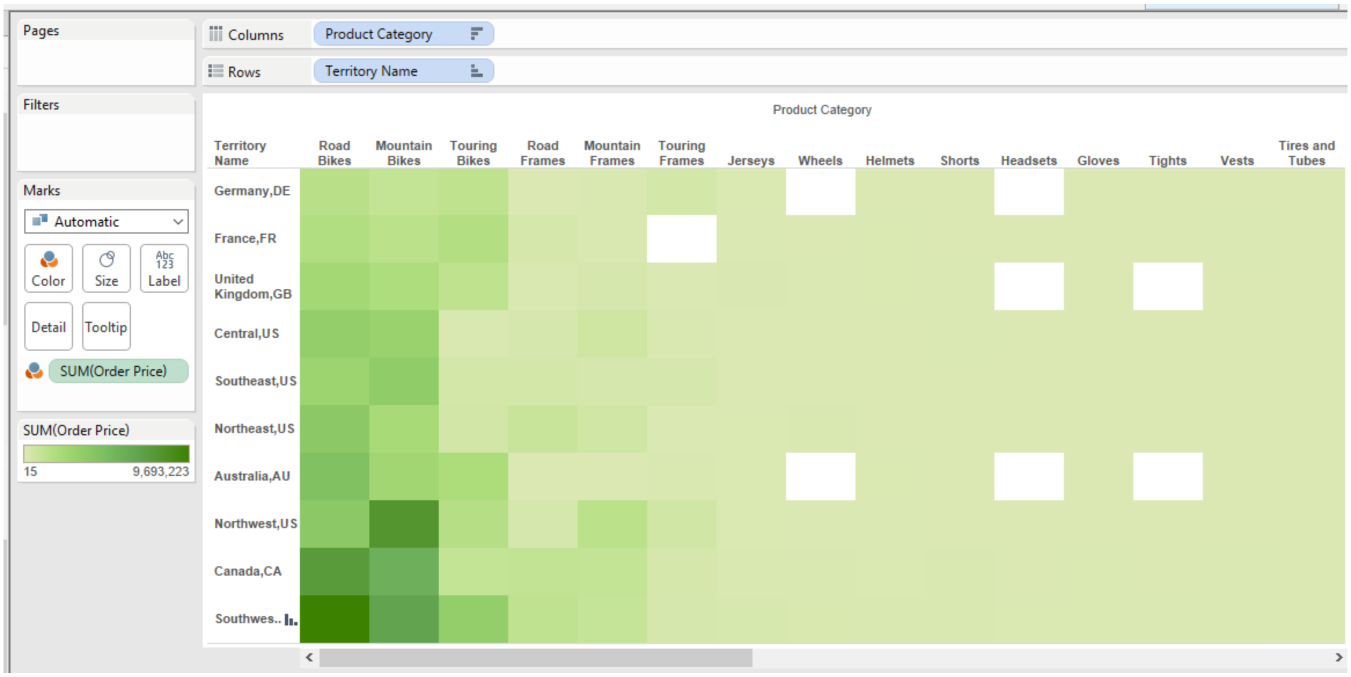
```
DATEDIFF('day',[Order Date],[Due Date] )
```

The domain for this field has not been loaded. Click "Load" to retrieve.

The average delivery time of all orders was found out to be “14.00” days.

Our third focus was to find out which product category is most famous in different territories in the world?

For this purpose, following infographic was generated:



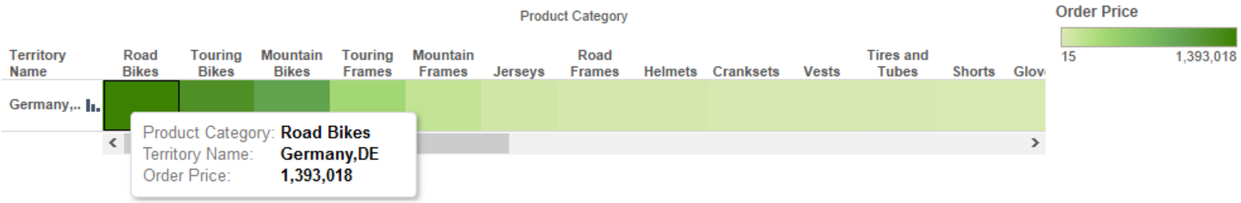
Columns: Product Category

Rows: Territory Name

Marks: Automatic

SUM(Order Price)

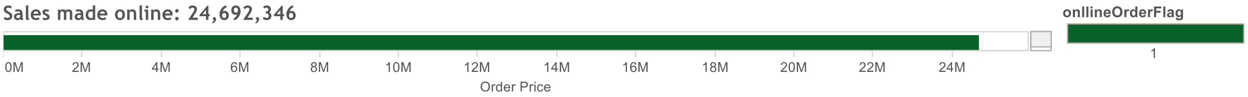
It tells us that in the “Southwest, US” territory the Road Bikes are making most sales. The table is arranged as per the ascending order of product category from the southwest US territory. The color strength tells us how one product is doing in one territory in comparison to other territories. If we want to see famous category in one territory we just arrange the data into ascending order for that territory. For example, In Germany, DE the amount of road bikes sold is too less in comparison to other territories but it is still the best sold product.



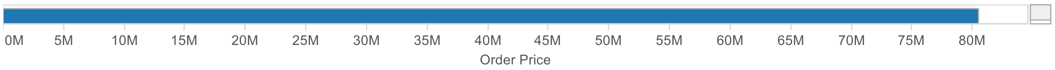
Product Category: Road Bikes
Territory Name: Germany,DE
Order Price: 1,393,018

Lastly, The stats of online sales vs. offline sales made by the company

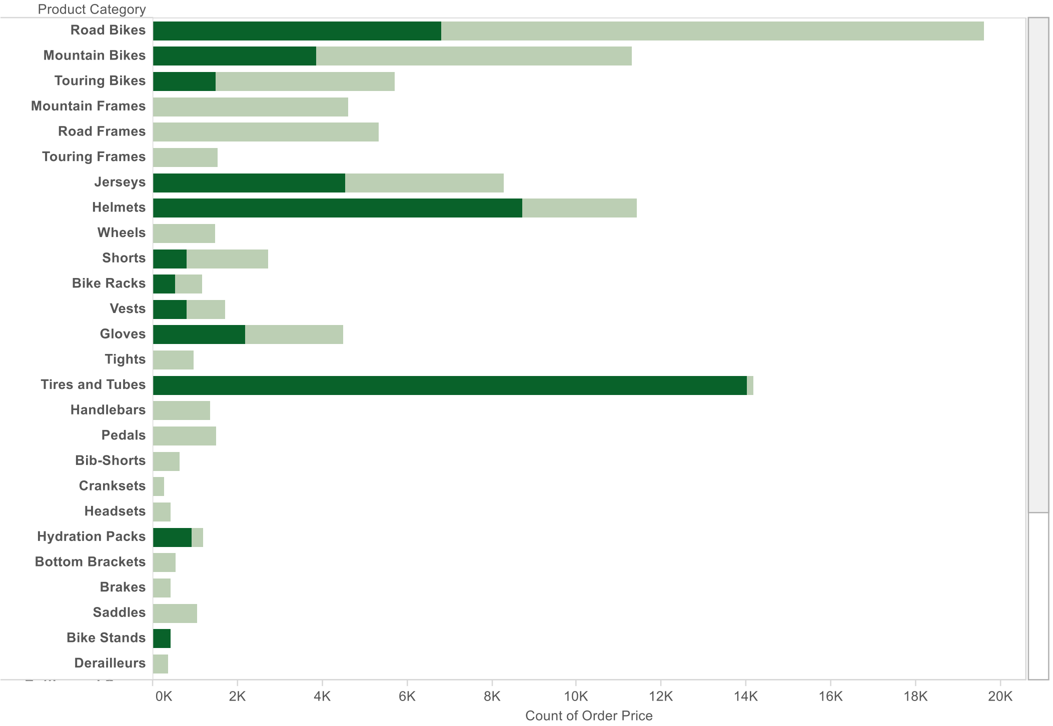
Sales made online: 24,692,346



total sales made offline: 80,555,585



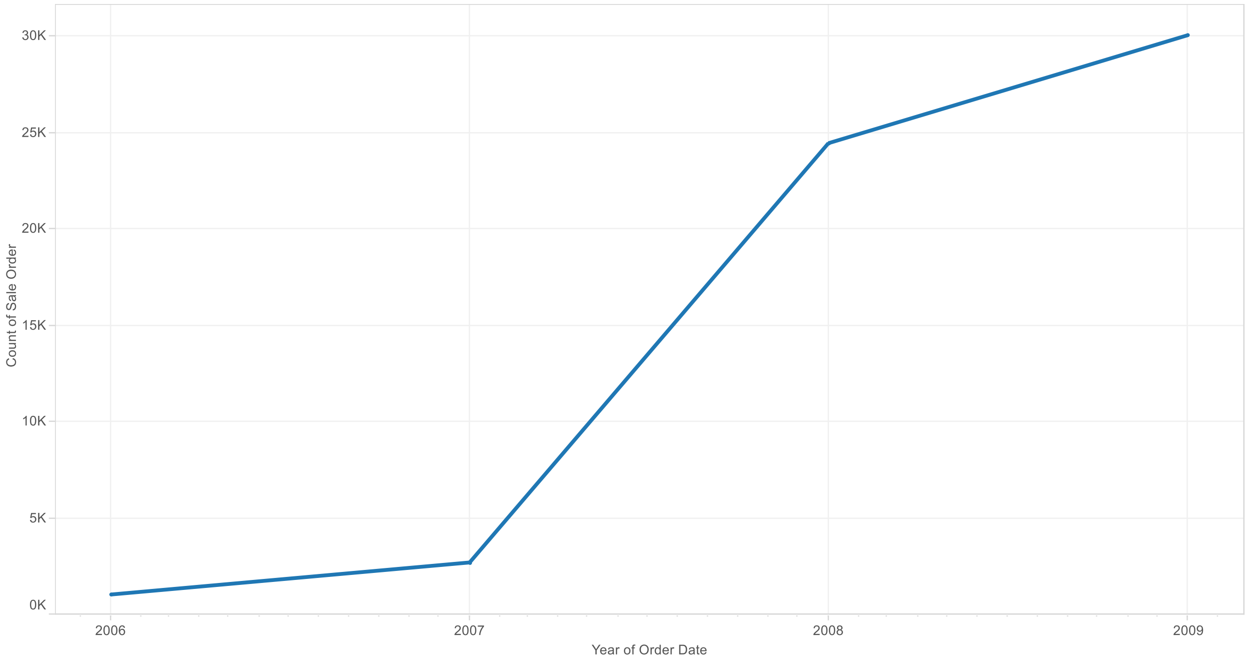
proportion of each product category sold online (green) vs. offline sales (colourless)



The ifnographs tell us that the company makes more sales offline than online. It also shows that even though the “road bikes” are most popularly sold offline, the online market sales are dominated by “tires and Tubes” category.

If we look at order dates of all the orders and compare them with online order flag, we see intereting results as well.

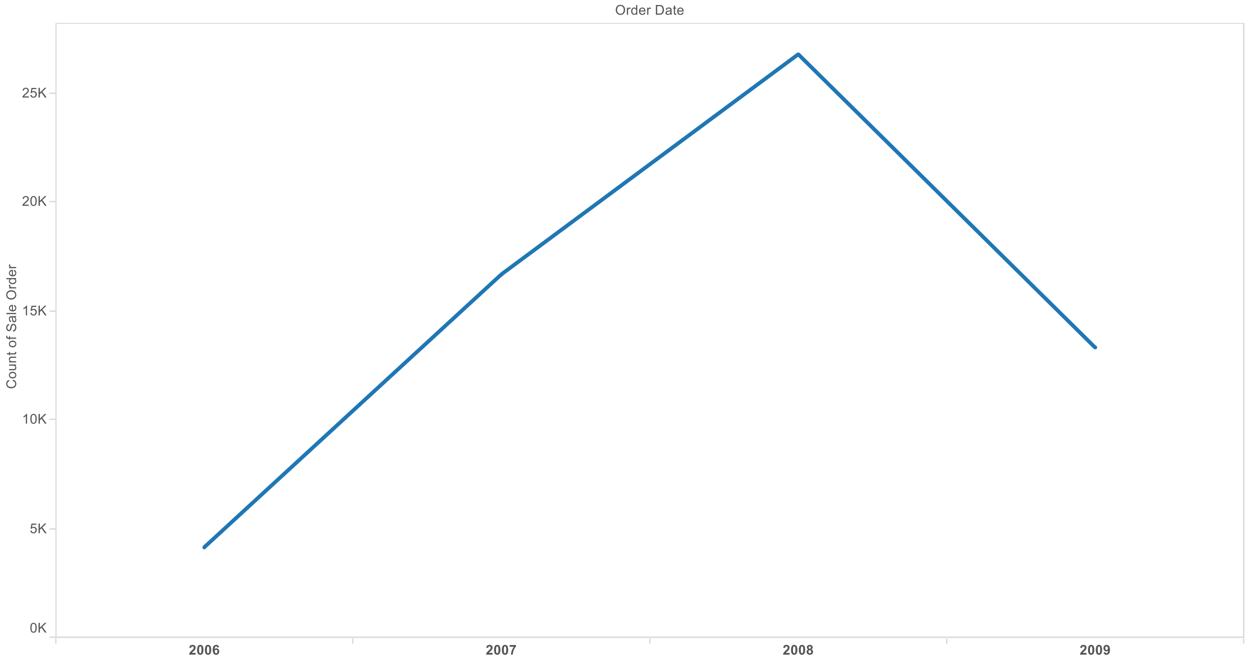
Sheet 8



The trend of count of Sale Order for Order Date Year. The data is filtered on onllineOrderFlag, which includes values greater than or equal to 1.

Number of orders made online

Sheet 9



The trend of count of Sale Order for Order Date Year. The data is filtered on onllineOrderFlag, which includes values less than or equal to 0.

Number of orders made offline

These two graphs show that in the later years the company shows declining trend in orders that it makes offline while the number of orders made online are increasing every year.