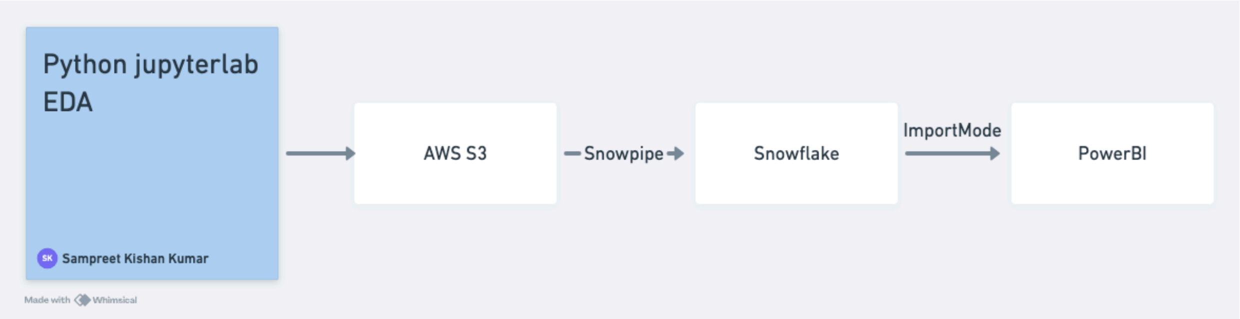
PEI data analyst assessment work

Submitted by: Sampreet Kishan
Modified: Aug 7th, 2024
sampy.prithvi@gmail.com

Task in hand:
Link to the question

Proposed Solution:



Brief Overview:

EDA:

1) Performed some simple EDA on a jupyterlab instance to make sure the datasets were somewhat good to work with.
2) Made sure all 3 datasets had no null values.
3) Made sure that the customers table had unique Customer_ID values which could act as the primary key for the foreign keys present in the shipping and orders datasets.
4) Made sure there was 100% match for all the foreign keys.
5) Made sure the numerical values in all tables were in acceptable ranges.
6) Noticed that the Shipping table wasn't connected to the Orders table when in most cases it should be. The question being: How can a customer have a shipment out/delivered without ever having placed an order. (Please look at the github repo's jupyter notebook for more information).

AWS S3 (input data source):

1) Uploaded files to S3 bucket.
2) Customers file is in excel format. Since downstream snowflake warehouse can't read excel file, wrote a lambda function (along with s3 notifications) to convert Customer excel(files) to csv which Snowflake can easily read.

SnowFlake data warehousing:

1) Created database, schemas, virtual warehouse, role for this project.
2) Created 3 tables (customers, orders, shipping) and set up primary key -> foreign key relationship between the tables.
3) Created stages and file formats to stage the datasets on S3.

4) Pulled the staged datasets into the tables on SnowFlake.
5) Set up snowpipe to ensure any additional data on AWS S3 is also ingested into SnowFlake.

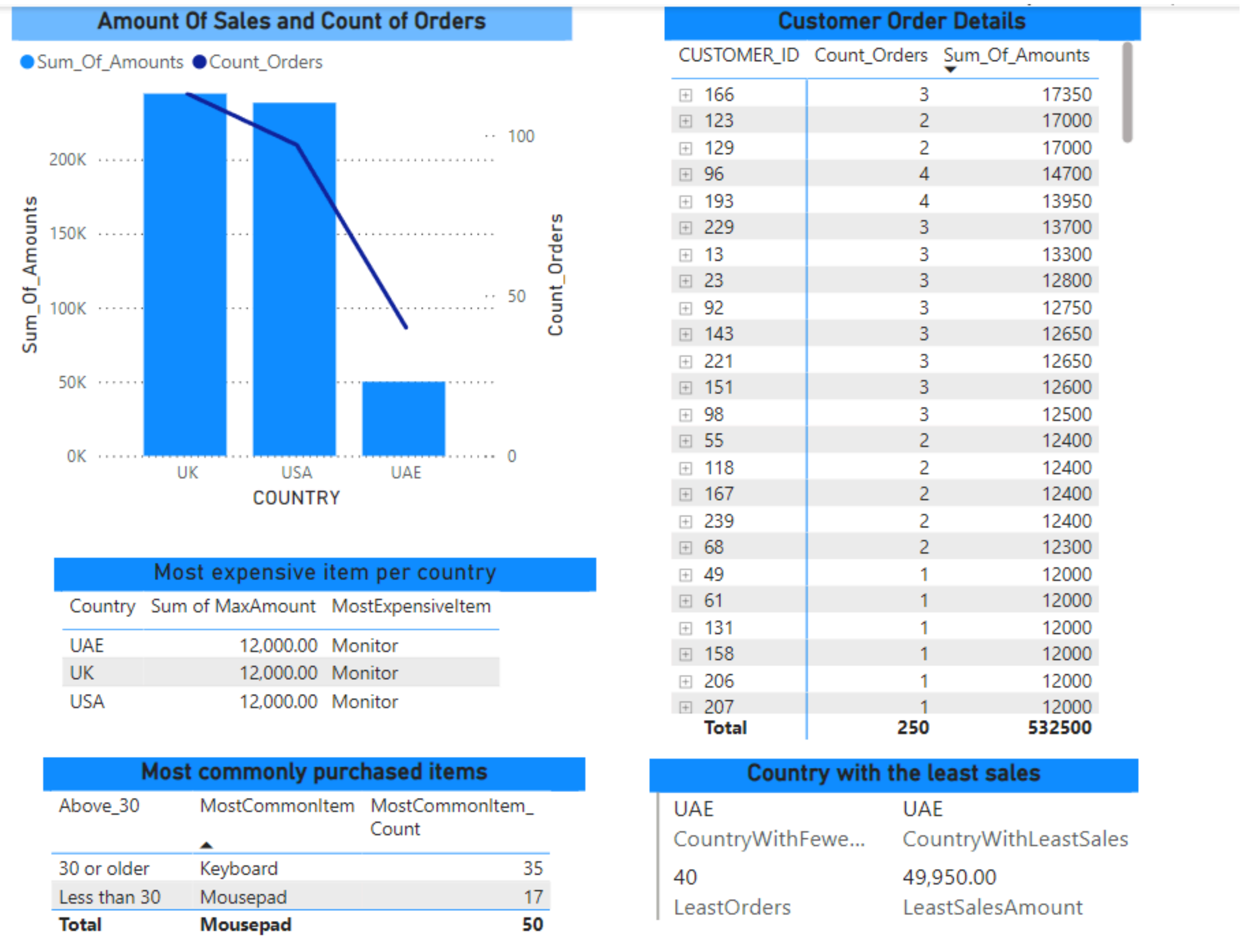
PowerBI (data modellig and data viz):

1) Connected PowerBI to Snowflake account.
2) Pulled the data and made minimal data type transformations using PowerQuery.
3) Used the modeling tab, to create dimension table(customers) > fact table (shipping, orders) 1:many relationships.
4) Wrote DAX queries to obtain most expensive items per country, Most Common item and count per age group, countries with the least sales items and amounts etc.

Some comments:

1) It would be necessary for the upstream data engineering team to figure out a way to incorporate an order_ID (of type integer) column in the shipping json file.
2) The data engineering team could potentially work on cleaning the names of customers with non-alphanumeric values.
3) The data engineering team could either push the data into the SnowFlake tables created or load it into AWS S3 bucket since I have already created somewhat of an ETL pipeline to bring it into PowerBI in a somewhat clean, seamless and automated manner.

The report:



Amount Of Sales and Count of Orders

Customer Order Details

CUSTOMER_ID	Count_Orders	Sum_Of_Amounts
⊞ 166	3	17350
⊞ 123	2	17000
⊞ 129	2	17000
⊞ 96	4	14700
⊞ 193	4	13950
⊞ 229	3	13700
⊞ 13	3	13300
⊞ 23	3	12800
⊞ 92	3	12750
⊞ 143	3	12650
⊞ 221	3	12650
⊞ 151	3	12600
⊞ 98	3	12500
⊞ 55	2	12400
⊞ 118	2	12400
⊞ 167	2	12400
⊞ 239	2	12400
⊞ 68	2	12300
⊞ 49	1	12000
⊞ 61	1	12000
⊞ 131	1	12000
⊞ 158	1	12000
⊞ 206	1	12000
⊞ 207	1	12000
Total	**250**	**532500**

Most expensive item per country

Country	Sum of MaxAmount	MostExpensiveItem
UAE	12,000.00	Monitor
UK	12,000.00	Monitor
USA	12,000.00	Monitor

Most commonly purchased items

Above_30	MostCommonItem	MostCommonItem_Count
30 or older	Keyboard	35
Less than 30	Mousepad	17
Total	**Mousepad**	**50**

Country with the least sales

UAE CountryWithFewe...	UAE CountryWithLeastSales
40 LeastOrders	49,950.00 LeastSalesAmount

Insights:

1) It looks like UK and US bring in the most revenue.
2) The most expensive sale in each country is Monitor.
3) Younger people largely purchase mousepads and others a keyboard.
4) Customers that drive the largest Sales amounts usually buy multiple items.
5) UAE is the country with the least amount of orders and sales.