Retail Sales Analytics

Phase 1: Excel – Data Cleaning & Preparation

Task 1

Standardize Column Headers

Done.

Task 2

Remove Duplicates

Delete the duplicates by using remove duplicates.

Task 3

Handle Missing Values

Find the empty values by using filter. Then fill the blanks with “NA”.

Task 4

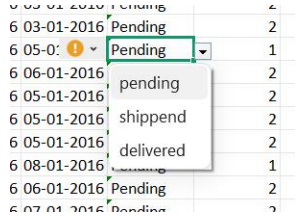
Data Type Conversion

Change the column data type to suitable data type. Example for change the order date to date. Qty to number.

Task 5

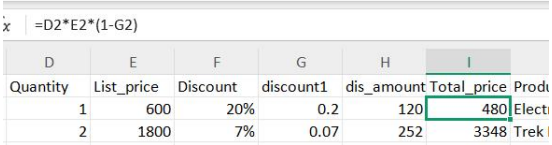
Data Validation

Use dropdowns for fields like order_status (Pending, Shipped, Delivered)
to ensure consistency



Task 6

Create New Derived Columns

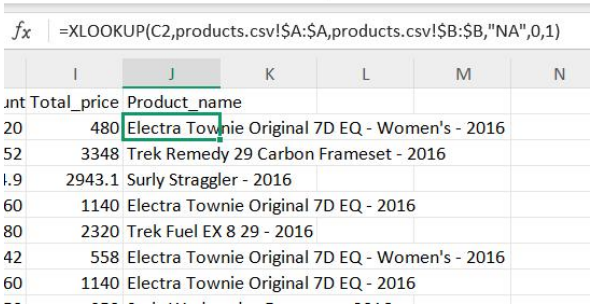


total_price = (list_price * quantity) - discount

Task 7

Merge Lookup Data

XLOOKUP to merge product names into order_items using product_id.



	I	J	K	L	M	N
ınt	Total_price	Product_name				
20	480	Electra Townie Original 7D EQ - Women's - 2016				
52	3348	Trek Remedy 29 Carbon Frameset - 2016				
1.9	2943.1	Surly Straggler - 2016				
60	1140	Electra Townie Original 7D EQ - 2016				
80	2320	Trek Fuel EX 8 29 - 2016				
42	558	Electra Townie Original 7D EQ - Women's - 2016				
60	1140	Electra Townie Original 7D EQ - 2016				

Task 8

pivot table summarizing total sales by product category

Category_name	Sum of Total_price
Children Bicycles	292149
Comfort Bicycles	394016
Cruisers Bicycles	995007
Cyclocross Bicycles	711026
Electric Bikes	916689
Mountain Bikes	2715064
Road Bikes	1665095
Grand Total	**7689046**

Task 9

Sort and Filter for Outliers

	Product Name	Price
higest value	Trek Domane SLR 9 Disc - 2018	12000
lowest value	Strider Classic 12 Balance Bike - 2018	90

Task 10

Prepare Final CSVs

Done

Phase 2: SQL – Database Management and Querying

Task 1

Create Tables Based on ERD

```
create table customer (
   Customer_id int ,
   First_name varchar(50),
   Last_name varchar(50),
   Email varchar(50),
   Email_website varchar(50),
   Phone varchar(50),
   Street varchar(50),
   City varchar(50),
   State varchar(50),
   Zip_code varchar(50)
);
```

Task 2

Import CSVs into SQL

Done

Task 3

Inner Join for Order Details

```
select orders.order_id,orders.customer_id,products.product_id,products.product_name,
order_items.Quantity,order_items.list_price,order_items.discount1 from orders join order_items on
orders.order_id= order_items.order_id join products on order_items.product_id=
products.product_id
order by orders.order_id,products.product_name;
```

Task 4

Total Sales by Store

```
select orders.store_id,stores.store_name,sum(order_items.quantity*order_items.list_price*(1-
order_items.discount1)) as Total_sales from orders
join order_items on orders.order_id=order_items.order_id
join stores on orders.store_id=stores.store_id
group by orders.store_id,stores.store_name;
```

Task 5

Top 5 Selling Products

```
select product_id,product_name,sum(quantity)as Total_qty from order_items group by
product_id,product_name order by Total_qty desc limit 5;
```

Task 6

Customer Purchase Summary

```
select customers.customer_id,customers.first_name as customer_name, count( distinct
orders.order_id) as total_order,
sum(order_items.quantity) as Total_Qty,
sum(order_items.quantity*order_items.list_price*(1-order_items.discount1)) as Total_sale from
customers
join orders on customers.customer_id=orders.customer_id
join order_items on orders.order_id=order_items.order_id group by
customers.customer_id,first_name
order by total_sale desc;
```

Task 7

Segment Customers by Total Spend

```
select customers.customer_id,customers.first_name as customer_name, count( distinct
orders.order_id) as total_order,
sum(order_items.quantity*order_items.list_price*(1-order_items.discount1)) as total_spent,
case
when sum(order_items.quantity*order_items.list_price*(1-order_items.discount1))<5000
then 'Low' when sum(order_items.quantity*order_items.list_price*(1-order_items.discount1))
between 5000 and 15000
then 'mid' else 'high' end as spending_brackets from customers
join orders on customers.customer_id=orders.customer_id
join order_items on orders.order_id=order_items.order_id
group by customer_id,first_name order by total_spent desc;
```

Task 8

Staff Performance Analysis

```
select orders.store_id,orders.staff_id,staffs.first_name as
staff_name,sum(order_items.quantity*order_items.list_price*(1-order_items.discount1)) as sales
from orders join order_items on orders.order_id=order_items.order_id
```

```
join staffs on orders.staff_id=staffs.staff_id
group by orders.staff_id,orders.store_id order by staff_id;
```

Task 9

Stock Alert Query

```
select stocks.store_id,stocks.product_id,products.product_name,stocks.quantity from stocks
join products on products.product_id=stocks.product_id
where stocks.quantity<10 order by stocks.store_id;
```

Task 10

Create Final Segmentation Table

```
create table customer_segments (
 customer_id int,
 customer_name varchar(50),
 recency int,
 frequency int,
 monetary double
 );
```

Phase 3: Python and ML Tasks Mandatory Tasks

Task 1

Load Data from SQL

```
engine = create_engine("mysql+pymysql://root:312712@localhost:3306/retail")
pd.read_sql("select * from orders;",engine)
```

Task 2

Basic EDA (Exploratory Data Analysis)

```
4]: df.describe()
```

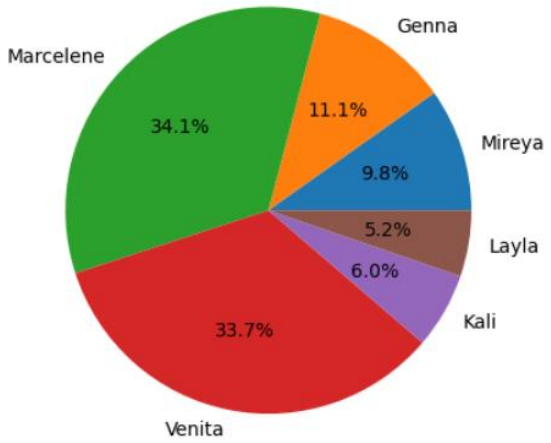
4]:

	Order_id	Customer_id	Order_status	Store_id	Staff_id
count	1615.000000	1615.000000	1615.000000	1615.000000	1615.000000
mean	808.000000	654.171517	3.778947	1.892260	5.855108
std	466.354658	443.229967	0.693430	0.558395	1.913899
min	1.000000	1.000000	1.000000	1.000000	2.000000
25%	404.500000	237.000000	4.000000	2.000000	6.000000
50%	808.000000	638.000000	4.000000	2.000000	6.000000
75%	1211.500000	1041.500000	4.000000	2.000000	7.000000
max	1615.000000	1445.000000	4.000000	3.000000	9.000000

```
pc
```

	store_id	staff_id	staff_name	sales
0	1	2	Mireya	752541.36
1	1	3	Genna	853294.29
2	2	6	Marcelene	2624140.52
3	2	7	Venita	2591649.90
4	3	8	Kali	463921.74
5	3	9	Layla	403627.01

```
plt.pie(pc['sales'], labels=pc['staff_name'], autopct='%1.1f%%')
plt.Category("pie chart Example")
plt.show()
```



Task 3

Calculate RFM Features for Customers

```
Customers=pd.read_sql("select * from retail.customers;",engine)
```

```
order_items=pd.read_sql("select * from retail.order_items;",engine)
```

```
orders =pd.read_sql("select * from retail.orders;",engine)
```

```
df1= orders.merge(order_items, on='Order_id', how='inner')
```

```
df2 = Customers.merge(df1, on='Customer_id', how='inner')
```

```
df2['Order_date'] = pd.to_datetime(df2['Order_date'])
```

```
rfm = df2.groupby('Customer_id').agg(
    recency=('Order_date', lambda x: (pd.to_datetime("today").normalize() - x.max()).days),
    frequency=('Total_price', 'nunique'),
    monetary=('Total_price', 'sum')
).reset_index()
```

```
rfm
```

	Customer_id	recency	frequency	monetary
0	1	2561	11	27888.30
1	2	2784	10	19329.22
2	3	2589	13	24051.70
3	4	2775	9	21151.00
4	5	2776	8	17520.40
...	...	...	...	...
1440	1441	2857	3	9173.40
1441	1442	3037	5	6987.30
1442	1443	3294	5	10372.14
1443	1444	3259	2	1400.00
1444	1445	2817	1	8000.00

1445 rows × 4 columns

Task 4

Export Segmentation Results to SQL

rfm.to_sql('customer_segments',con=engine,if_exists='append',index=False)

Phase 4: Power BI – Visualization & Dashboarding Mandatory Tasks

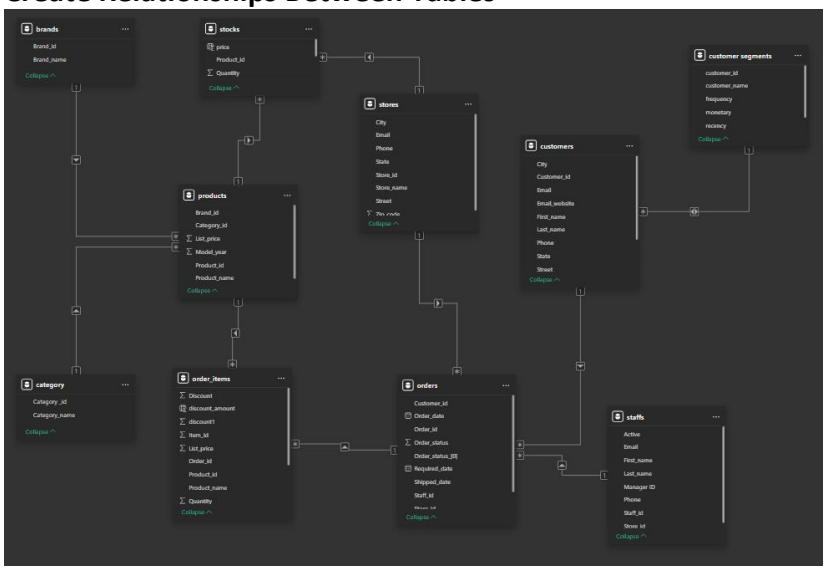
Task 1

Connect Power BI to SQL

I can’t connect mysql server to power bi for some unknown reason I save the table as csv file then import to power bi.

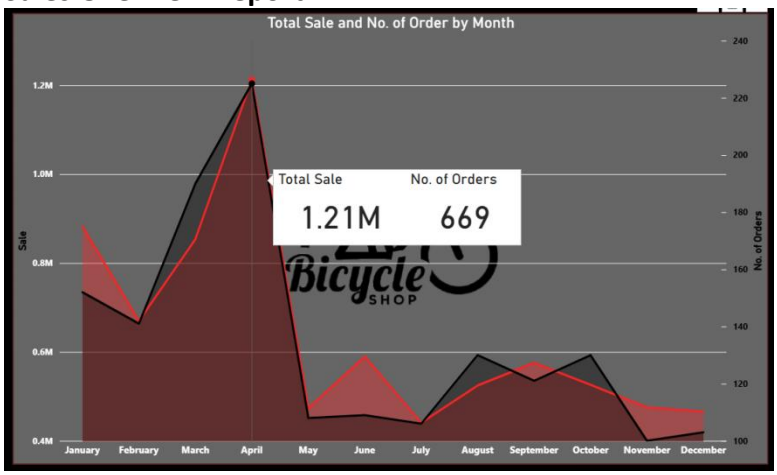
Task 2

Create Relationships Between Tables



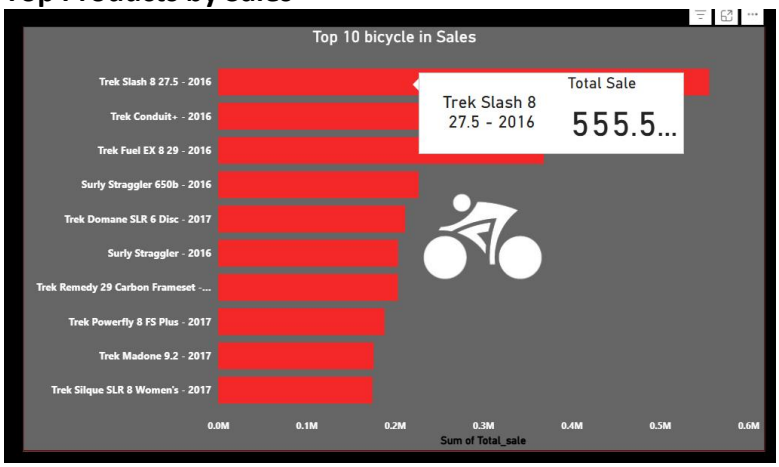
Task 3

Sales Overview Report



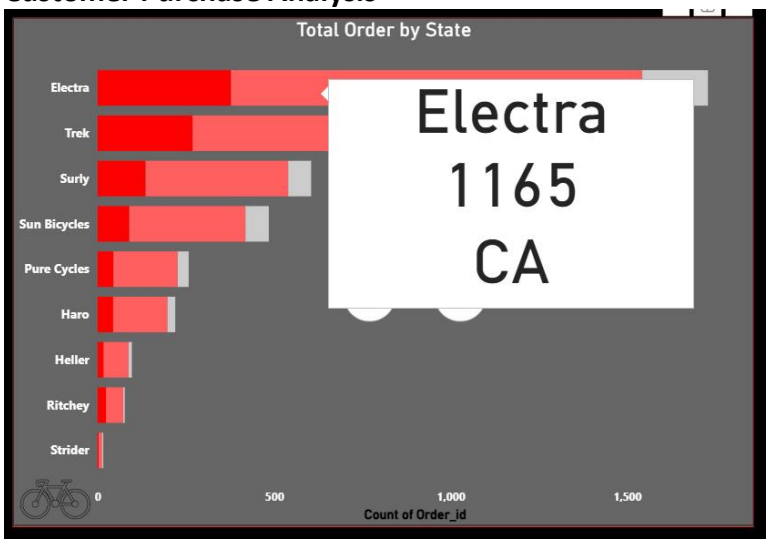
Task 4

Top Products by Sales



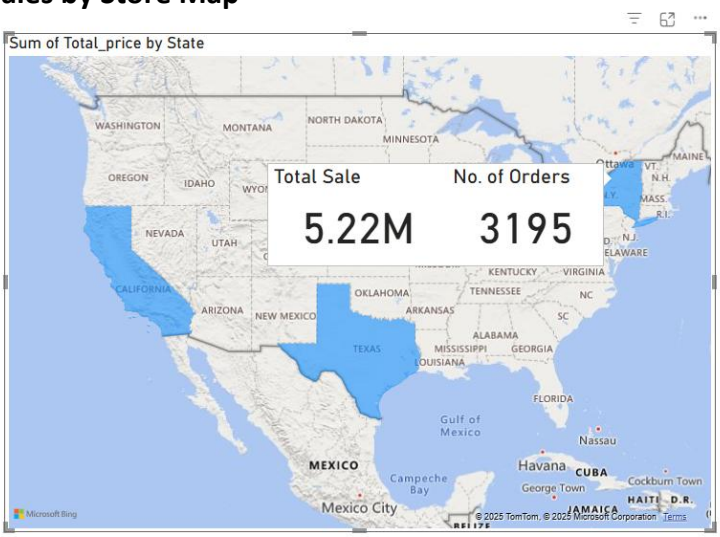
Task 5

Customer Purchase Analysis



Task 6

Sales by Store Map

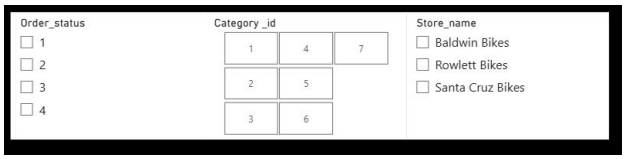


Task 7

Low Stock Alert Dashboard

Task 8

Interactive Filters and Slicers



Task 9

Staff Performance Report

First_name	Sum of Total_sale	Count of Order_id
Genna	8,53,294.29	184
Kali	4,63,921.74	88
Layla	4,03,627.01	86
Marcelene	26,24,140.52	553
Mireya	7,52,541.36	164
Venita	25,91,649.90	540
Total	**76,89,174.82**	**1615**

Task 10

Task 11

Import customer_segments Table

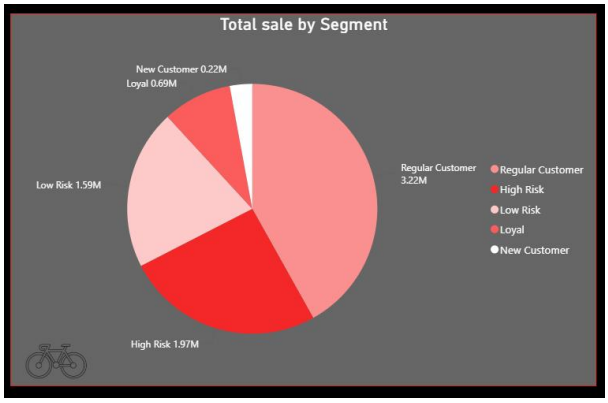
I can’t connect mysql server to power bi for some unknown reason I save the table as csv file then import to power bi.

Task 12

Visualize Customer Segments

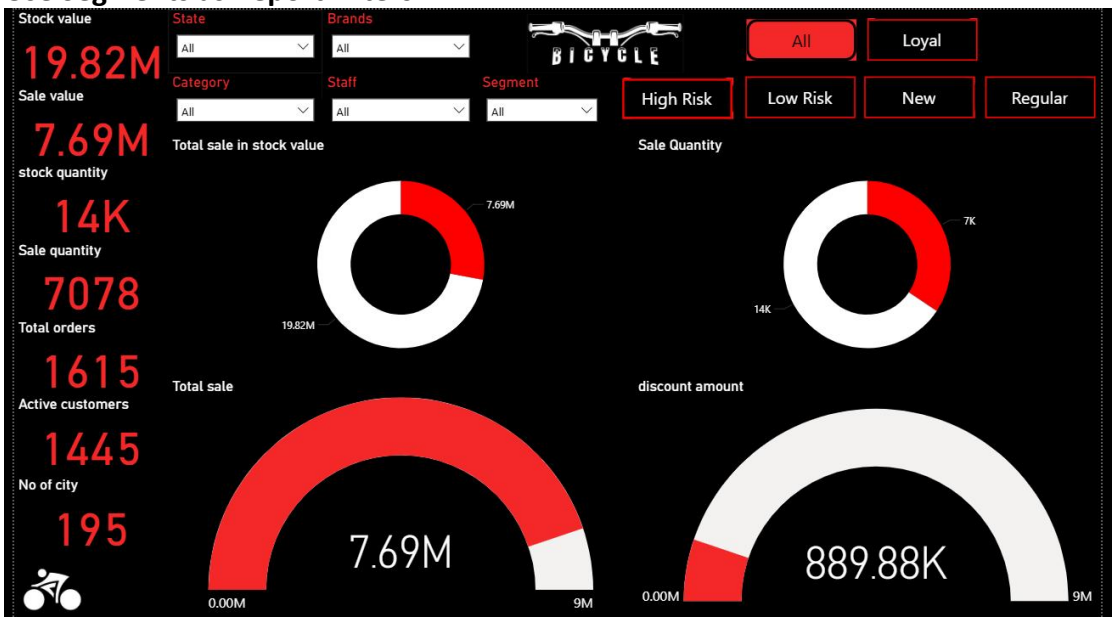
Task 13

Segment-Level Revenue Breakdown



Task 14

Use Segments as Report Filters



Final report

***Our best selling store**

Baldwin bikes in New York - 5.2M

***Best employees**

Venita is best seller in loyal customer

Marcelene is best in low risk and new customers.

***Best selling season**

*New Year

*March & April (School reopen)

*Halloween & Christmas

***Best selling**

Brand is Trek - 4.6M in 7.7M

Product is Trek Domane SLR 9 Disc - 2018 - 43k

***Target(2018) -Min(4M)**

Previous Sales

2016 - 2.42M

2017 - 3.44M

In 2018 after 4 months of sales - 1.74

We have to archive million in 8 months we still have year end sales and Christmas.

If we focus on new customers and call our lost customer(high risk).