US REGIONAL SALES REPORT

A SQL PROJECT BY SANDRA ASAGADE

INTRODUCTION

Over the past 3 weeks, I've been focused on learning basic SQL; how to query a data, how to use joins, window functions, how to create stored procedure and so many more. This is the last week of my SQL Track (*from my 3 months data challenge*), and I'm ending it with this particular project. In this project, I'm going to be sharing the steps I took in analyzing the data, and I'll also be transparent and sincere with my report.

ABOUT THE DATA

This project is about using SQL (Structured Query Language) to perform exploratory data analysis. The US Regional Sales data was gotten from Kaggle. It is a fictitious (imaginary) sales data for a certain company across the US regions. It's a csv file broken down into 6 tables; customers, location, region, products, sales_team, and sales_order tables

TOOL USED

I made use of MySQL Workbench because it is what I'm more familiar with. In case you're wondering what MySQL is, I mentioned it in my recent blogpost.

TABLES

customer	
_CustomerID	int
CustomerNames	text

location	
_StoreID	int
CityName	text
County	text
StateCode	text
State	text
Type	text
Latitude	double
Longitude	double
AreaCode	int
Population	int
Household Income	int
Median Income	int
Land Area	int
Water Area	int
Time Zone	text

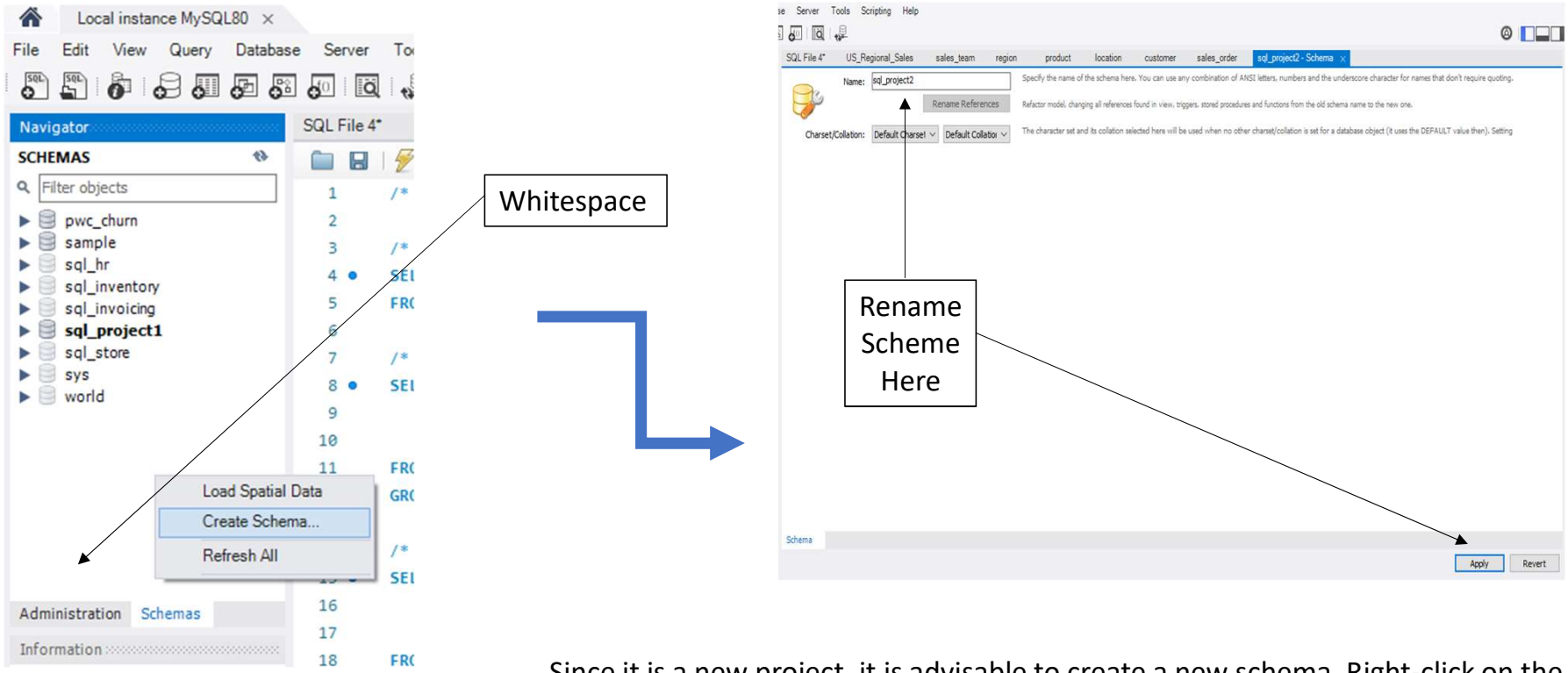
sales_order	
OrderNumber	text
SalesChannel	text
WarehouseCode	text
ProcuredDate	text
OrderDate	text
ShipDate	text
DeliveryDate	text
CurrencyCode	text
_SalesTeamID	int
_CustomerID	int
_StoreID	int
_ProductID	int
OrderQuantity	int
DiscountApplied	double
UnitPrice	double
UnitCost	double

product	
_ProductID	int
ProductName	text

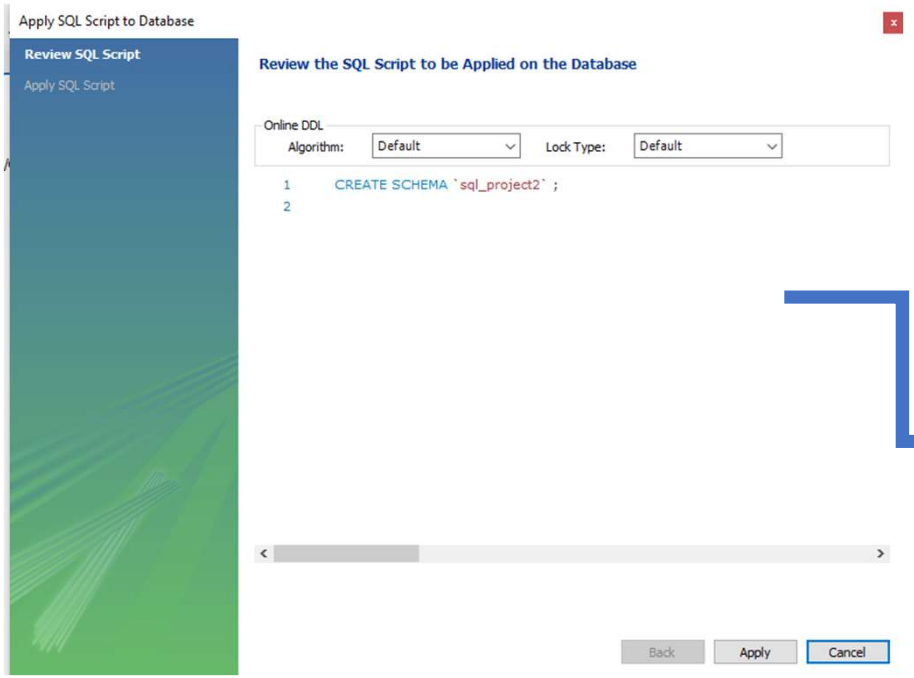
sales_team	
_SalesTeamID	int
SalesTeam	text
Region	text

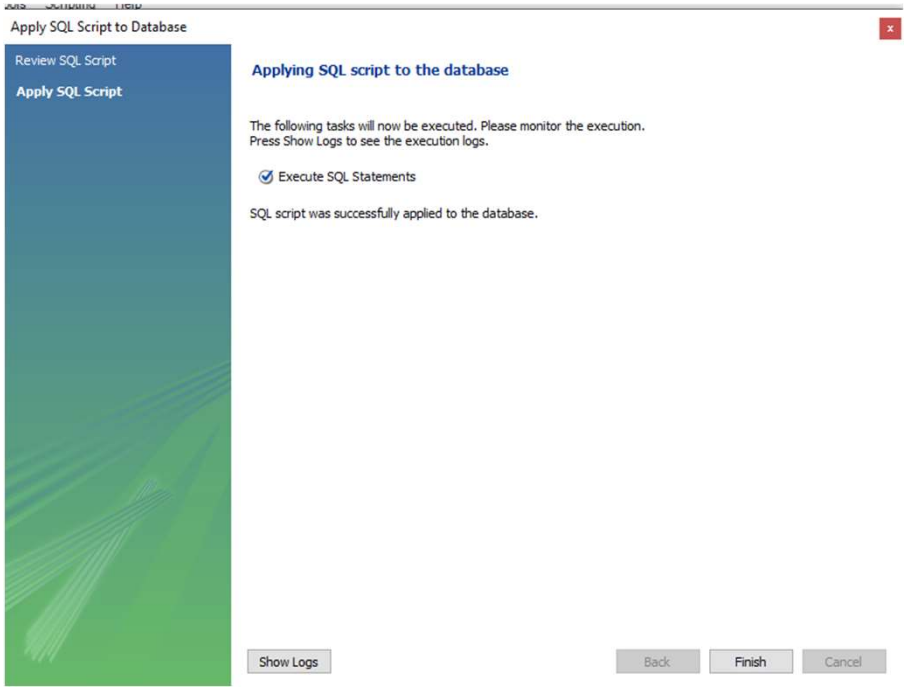
region	
StateCode	text
State	text
Region	text

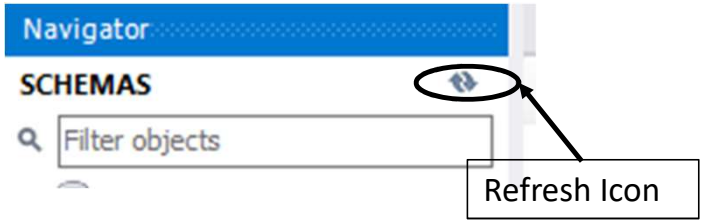
LOADING THE DATA – Create Schema



Since it is a new project, it is advisable to create a new schema. Right-click on the whitespace in the schema pane and click on 'Create Schema'. A new page will be displayed, there you can rename your schema *(I named mine 'sql_project1')* and then click on 'Apply'.

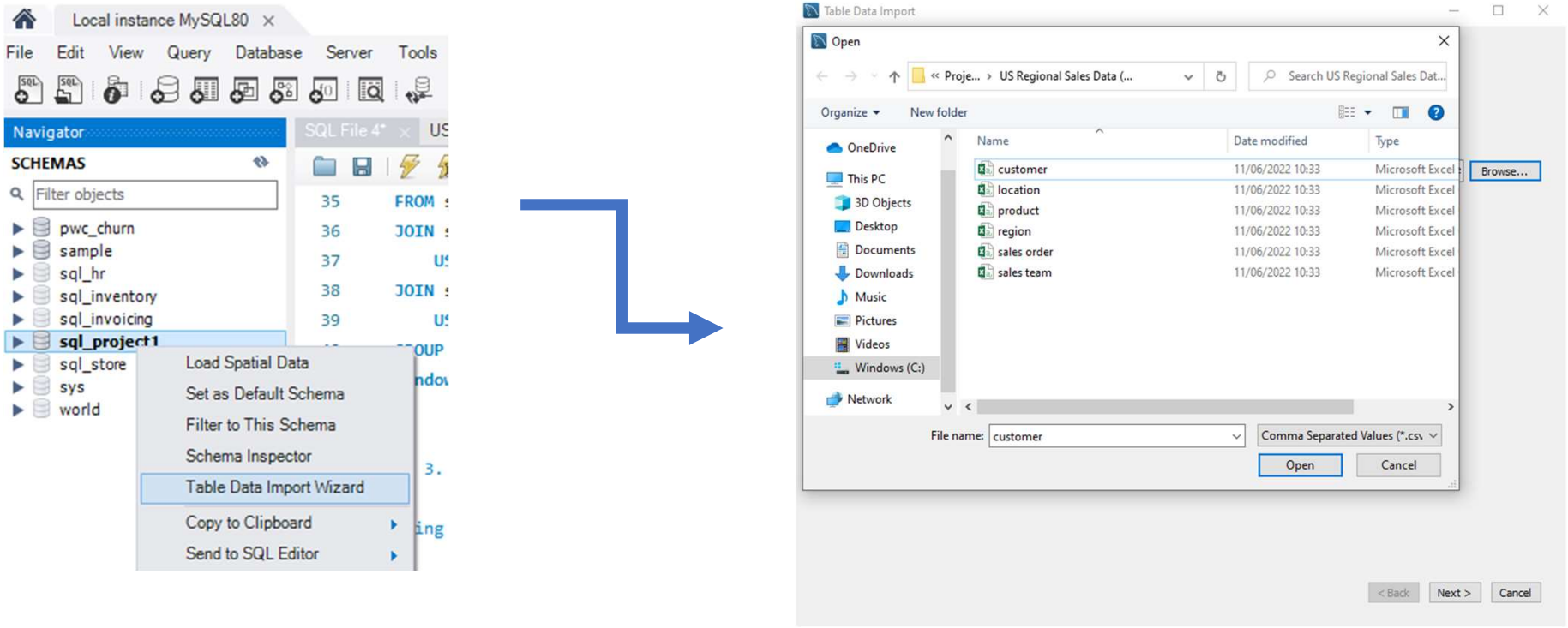




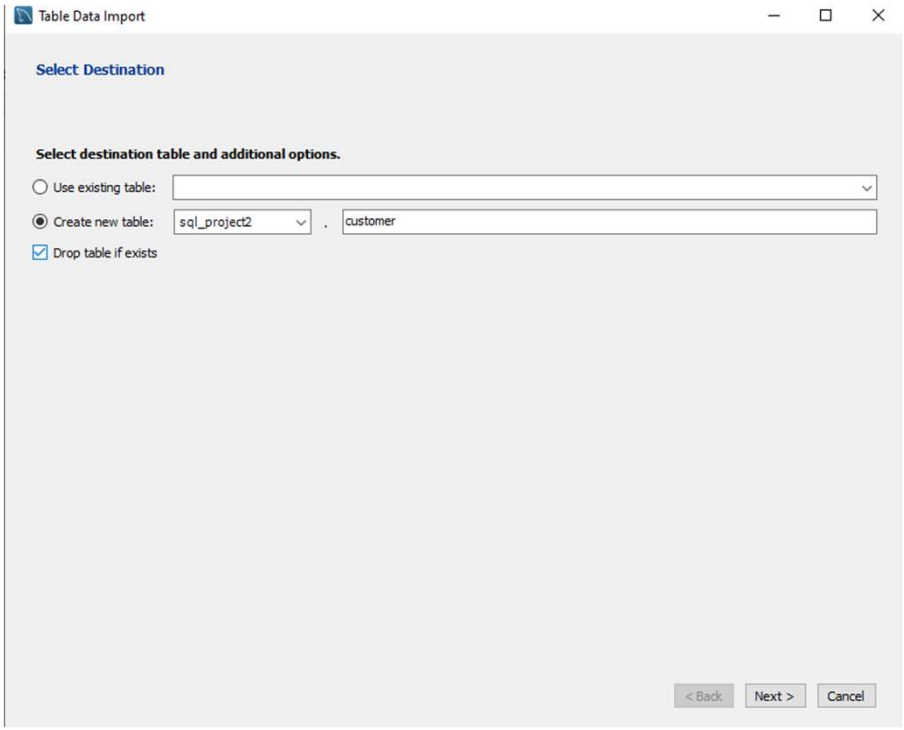


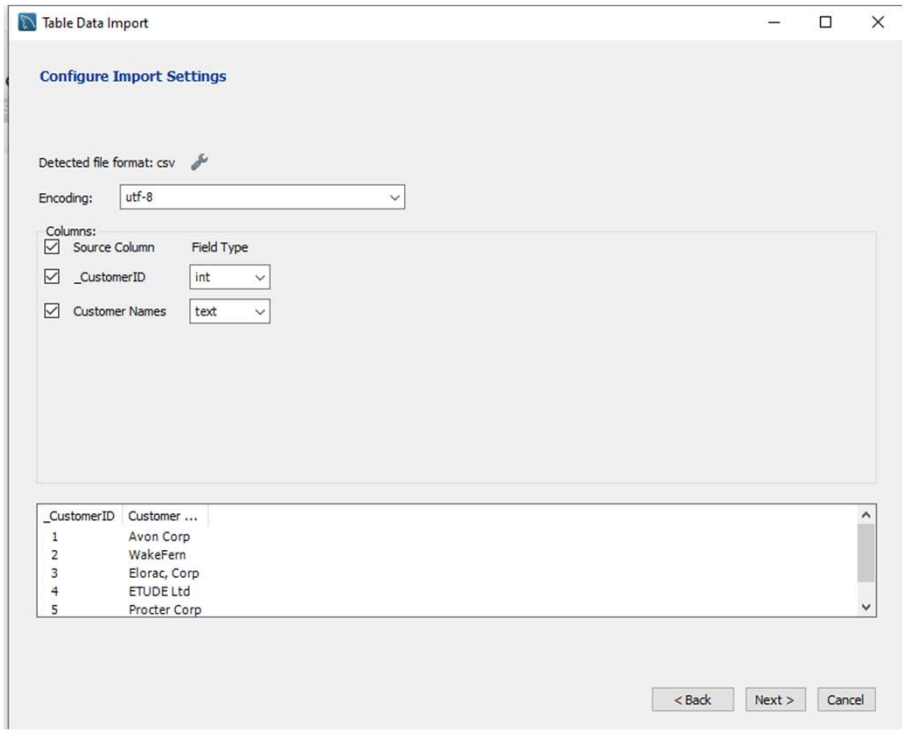
A pop-up page will be displayed asking you to review the SQL script (*You can also create your schema this way in the SQL editor as shown in the pop-up*). After reviewing the script and you are sure that it is correct, click on 'Apply' and then click 'Finish'. On your schema pane, you would see a refresh icon, click on it to refresh your schema so it appears in the database

LOADING THE DATA – Importing tables (data)

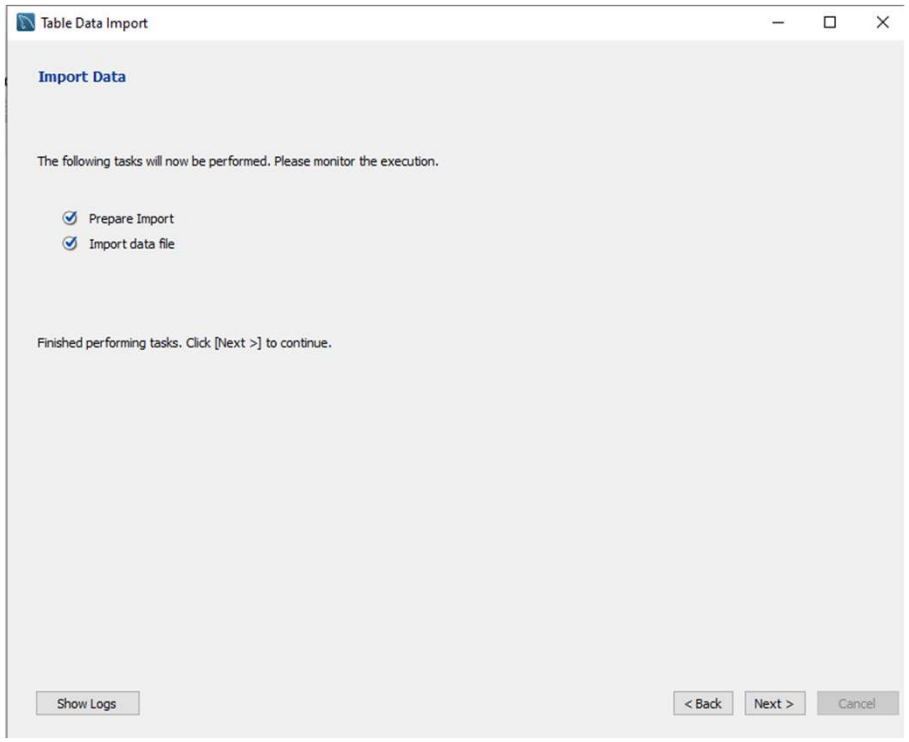


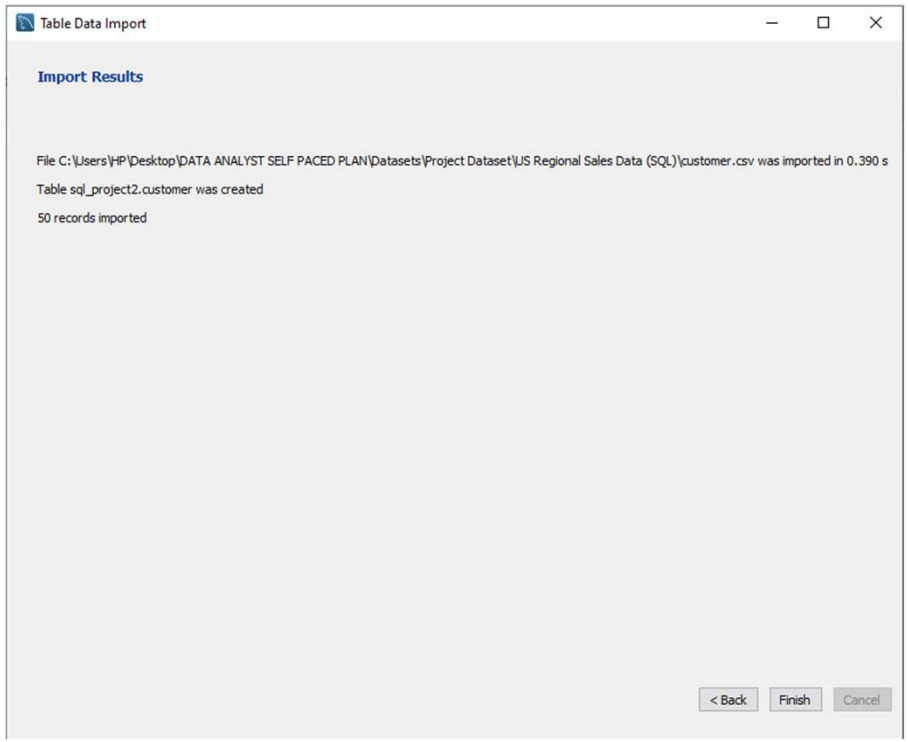
Right-click on your new schema and click on 'Table Data Import Wizard'. A pop-up will be displayed asking for the file path to the data, you can either copy and paste the file path or browse and select manually. In my own case, I browsed manually by clicking on 'Browse' and selecting the data, and then clicking on 'Next >'.





In the next pop-up, ensure to check the box that says ‘Drop table if exists’ as it would prevent duplicate tables. After doing all that, click on ‘Next >’. Another pop-up will be displayed showing you the columns in the data and their respective data types, you can choose to select the columns you want to import manually or select all and click on ‘Next >’.





In the next pop-up that says 'Import Data', wait for the data to be imported and click on 'Next >' and finally in the next and final pop-up that says 'Import Results', click on 'Finish' and the refresh your schema to apply changes. I imported 6 tables; the customer, location, region, sales_team and sales_order tables. (*NOTE: They have to be in csv format.*)

EXPLORATORY DATA ANALYSIS

Now that the tables have been imported, it's time for the interesting part, which is analyzing the data. Before I began analyzing the data, I listed out a couple of questions that would help me in writing the right queries to get insights for the analysis.

1. Are there duplicates in the data?
2. Are the data types correct?
3. What is the total number of transactions made, total quantity of products ordered, total revenue and total profit?
4. What is the total number of transactions made, total quantity of products ordered, total revenue and total profit per order date?
5. What is the total transaction per sales channel?
6. What is the total number of customers and sales teams?
7. What is the total number of transaction for each customer?
8. What is the total number of stores in each region?
9. What is the total number of states in each region?
10. What is the total transaction per location type?
11. What is the total revenue and profit in each region?
12. What is the date difference between the order date and the delivery date?
13. How did each sales team perform based on the total transaction, total quantity sold and revenue made?
14. What are the most expensive and least expensive products ordered by customers?
15. How can I view the sales order of any sales team by just entering their name?

1. Are there duplicates in the data?

```
4 •  SELECT
5        OrderNumber,
6        COUNT(OrderNumber)
7    FROM sql_project1.sales_order
8    GROUP BY OrderNumber
9    HAVING COUNT(OrderNumber) > 1;
```

Result Grid | Filter Rows: | Exp

OrderNumber	COUNT(OrderNumber)

There were no duplicate entries in the data, and that is why the result is empty

2. Are the data types correct?

After importing the tables, I discovered that the date columns where of type TEXT, and in order for me to be able to use the date functions, I had to change the type to DATE type

```
174 •  UPDATE sql_project1.sales_order
175    SET ProcuredDate = str_to_date(ProcuredDate, "%d/%m/%Y");
176 •  UPDATE sql_project1.sales_order
177    SET OrderDate = str_to_date(OrderDate, "%d/%m/%Y");
178 •  UPDATE sql_project1.sales_order
179    SET ShipDate = str_to_date(ShipDate, "%d/%m/%Y");
180 •  UPDATE sql_project1.sales_order
181    SET DeliveryDate = str_to_date(DeliveryDate, "%d/%m/%Y");
182 •  SELECT
183        ProcuredDate,
184        OrderDate,
185        ShipDate,
186        DeliveryDate
187    FROM sql_project1.sales_order;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProcuredDate	OrderDate	ShipDate	DeliveryDate
2017-12-31	2018-05-31	2018-06-14	2018-06-19
2017-12-31	2018-05-31	2018-06-22	2018-07-02
2017-12-31	2018-05-31	2018-06-21	2018-07-01
2017-12-31	2018-05-31	2018-06-02	2018-06-07

3. What is the total number of transactions made, total quantity of products ordered, total revenue and total profit?

```
12 •  SELECT
13        COUNT(*) AS Total_orders_made,
14        SUM(OrderQuantity) AS Total_Quantity,
15        ROUND(SUM(UnitPrice),2) AS Revenue,
16        ROUND(SUM(Profit),2) AS Profit
17    FROM sql_project1.sales_order;
18
```

Result Grid | Filter Rows: | Export:

Total_orders_made	Total_Quantity	Revenue	Profit
7991	36162	18255731.2	6820482.09

4. What is the total number of transactions made, total quantity of products ordered, total revenue and total profit per order date?

```
20 •  SELECT
21        DATE_FORMAT(OrderDate, '%Y') AS Year,
22        COUNT(*) AS Total_Transaction,
23        SUM(OrderQuantity) AS Total_Quantity,
24        ROUND(SUM(UnitPrice),2) AS Revenue,
25        ROUND(SUM(Profit),2) AS Profit
26    FROM sql_project1.sales_order
27    GROUP BY Year
28    ORDER BY Revenue DESC;
```

Result Grid | Filter Rows: | Export: | Wrap

Year	Total_Transaction	Total_Quantity	Revenue	Profit
2020	3125	14043	7114689.8	2656782.22
2019	3030	13637	6957950	2614611.37
2018	1836	8482	4183091.4	1549088.5

Company's revenue and profit increased continuously from 2018 to 2020

5. What is the total transaction per sales channel?

```
30 •  SELECT
31        SalesChannel,
32        COUNT(*) AS Total_Transaction
33    FROM sql_project1.sales_order
34    GROUP BY SalesChannel;
```

Result Grid | Filter Rows: | Exp

SalesChannel	Total_Transaction
In-Store	3298
Online	2425
Distributor	1375
Wholesale	893

6. What is the total number of customers and sales teams?

```
38 •  SELECT
39        COUNT(DISTINCT _CustomerID) AS Total_Customers,
40        COUNT(DISTINCT _SalesTeamID) AS Total_Sales_Team
41    FROM sql_project1.sales_order s1;
42
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Total_Customers	Total_Sales_Team
50	28

7. What is the total number of transaction for each customer?

```
43 •  SELECT
44        c._CustomerID,
45        CustomerNames,
46        COUNT(*) AS Total_Transaction
47    FROM sql_project1.sales_order s1
48    JOIN sql_project1.customer c
49        USING(_CustomerID)
50    GROUP BY _CustomerID
51    ORDER BY Total_Transaction DESC;
```

Result Grid | Filter Rows: | Expo

_CustomerID	CustomerNames	Total_Transaction
12	Medline	210
18	Eminence Corp	186
3	Elorac, Corp	181
29	Apotheca, Ltd	179
11	Apollo Ltd	178
34	OUR Ltd	176

Out of the 50 customers who have been ordering, these are the top 6, with Medline being the customer who ordered the most.

8. What is the total number of stores in each region?

```
54 •  SELECT
55        Region,
56        COUNT(_StoreID) AS Stores_Count
57    FROM sql_project1.location
58    JOIN sql_project1.region
59        USING(StateCode)
60    GROUP BY Region
61    ORDER BY Stores_Count DESC;
```

Result Grid | Filter Rows: | Export

Region	Stores_Count
West	127
South	116
Midwest	78
Northeast	46

9. What is the total number of states in each region?

```
64 •  SELECT
65        Region,
66        COUNT(State) AS Total_States
67    FROM sql_project1.region
68    GROUP BY Region
69    ORDER BY Total_States DESC;
```

Result Grid | Filter Rows:

Region	Total_States
South	16
West	12
Midwest	12
Northeast	8

10. What is the total transaction per location type?

```
135 •  SELECT
136        Type,
137        COUNT(*) AS Total_Transaction
138    FROM sql_project1.sales_order
139    JOIN sql_project1.location
140        USING(_StoreID)
141    GROUP BY Type;
142
```

Result Grid | Filter Rows:

Type	Total_Transaction
Town	364
City	6382
Borough	105
Township	696
CDP	316
Consolidated Government	25
Urban County	24
Metropolitan Government	23
Other	32
Unified Government	24

Customers who are based in the city ordered more than customers in other location type.

11. What is the total revenue and profit in each region?

```sql
WITH cte AS (
    SELECT
        Region,
        DATE_FORMAT(OrderDate, '%Y') AS Year,
        ROUND(SUM(UnitPrice),2) AS Revenue,
        ROUND(SUM(Profit),2) AS Profit
    FROM sql_project1.sales_order
    JOIN sql_project1.location
        USING(_StoreID)
    JOIN sql_project1.region
        USING(StateCode)
    GROUP BY Region, Year
    ORDER BY Region, Year, Revenue)
SELECT
    Region,
    Year,
    Revenue,
    Profit,
    ROUND(SUM(Revenue) OVER(PARTITION BY Region),2) AS Revenue_Per_Region,
    ROUND(((Revenue/18255731.2)*100),2)AS Percent_Of_Total_Revenue
FROM cte
GROUP BY Region, Year
ORDER BY Region;
```

Results

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Region	Year	Revenue	Profit	Revenue_Per_Region	Percent_Of_Total_Revenue
Midwest	2018	961610.8	356029.59	3851280.6	5.27
Midwest	2019	1497195.4	561080.58	3851280.6	8.2
Midwest	2020	1392474.4	528713.84	3851280.6	7.63
Northeast	2018	422488.6	153792.78	2182806.4	2.31
Northeast	2019	846076	315264	2182806.4	4.63
Northeast	2020	914241.8	350588.12	2182806.4	5.01
South	2018	1288564.1	475469.59	5810367.3	7.06
South	2019	2233110	833696.99	5810367.3	12.23
South	2020	2288693.2	856739.07	5810367.3	12.54
West	2018	1510427.9	563796.54	6411276.9	8.27
West	2019	2381568.6	904569.8	6411276.9	13.05
West	2020	2519280.4	920741.19	6411276.9	13.8

Region: Shows the 4 different regions in US

Year: Distinct years of sales for each region

Revenue: Total revenue for each region in their respective years

Profit: Total profit for each region in their respective years

Revenue_Per_Region: Total revenue for each region

Percent_Of_Total_Revenue: Percentage of total revenue made each region in their respective years

12. What is the date difference between the order date and the delivery date?

```
195 •  SELECT
196        DATEDIFF(DeliveryDate,OrderDate) AS Delivery_Duration
197    FROM sql_project1.sales_order
198    GROUP BY Delivery_Duration
199    ORDER BY Delivery_Duration DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Delivery_Duration
38
37
36
35
34
33

So here, I wanted to know how long it takes products to get to the customers from the day it was ordered to the day it was delivered.

13. How did each sales team perform based on the total transaction, total quantity sold and revenue made?

```
144 •  SELECT
145        s2._SalesTeamID,
146        s2.SalesTeam,
147        COUNT(*) AS Total_Transaction,
148        SUM(OrderQuantity) AS Total_Quantity_Sold,
149        ROUND(SUM(UnitPrice),2) AS Revenue
150    FROM sql_project1.sales_order s1
151    JOIN sql_project1.sales_team s2
152        USING(_SalesTeamID)
153    GROUP BY SalesTeam
154    ORDER BY Total_Transaction DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

_SalesTeamID	SalesTeam	Total_Transaction	Total_Quantity_Sold	Revenue
13	Todd Roberts	340	1474	739639.8
18	Shawn Wallace	316	1476	692197.1
8	George Lewis	315	1409	729341.9
12	Carl Nguyen	314	1393	737395.3
7	Shawn Cook	303	1306	703278.9
1	Adam Hernandez	302	1366	703245.4

14. What are the most expensive and least expensive products ordered by customers?

```
SELECT
    p.ProductName,
    s1.UnitPrice AS Price,
    first_value(p.ProductName) over
        (order by UnitPrice Desc) AS Most_Expensive_Product,
   last_value(p.ProductName) over
    (order by UnitPrice Desc
        range between unbounded preceding
            and unbounded following) AS Least_Expensive_Product
FROM sql_project1.sales_order s1
JOIN sql_project1.product p
    USING(_ProductID)
GROUP BY p.ProductName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductName	Price	Most_Expensive_Product	Least_Expensive_Product
Vanities	6324.8	Vanities	Wardrobes
Rugs	6277.9	Vanities	Wardrobes
Platters	5976.4	Vanities	Wardrobes
Mirrors	5735.2	Vanities	Wardrobes
Bar Tools	5581.1	Vanities	Wardrobes

Total number of products

```
SELECT
    COUNT(DISTINCT ProductName) AS Total_Products
FROM sql_project1.product;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

Total_Products
47

There are 47 products all together. The most expensive is Vanites, while the least expensive is Wardrobes.

15. How can I view the sales order of any sales team by just entering their name?

```
157 •  USE sql_project1;
158 •  DROP procedure IF EXISTS sales_team_stats;
159    DELIMITER $$
160 •  USE sql_project1 $$
161 •  CREATE PROCEDURE sales_team_stats (IN TeamName VARCHAR(30))
162    BEGIN
163    SELECT
164        OrderNumber, SalesChannel, WarehouseCode, ProcuredDate, OrderDate,ShipDate, DeliveryDate,
165        CurrencyCode, _CustomerID, _StoreID, ProductName, OrderQuantity, DiscountApplied, UnitPrice, UnitCost,
166        ROUND((UnitPrice - UnitCost),2) AS Profit
167    FROM sql_project1.sales_order s1
168    JOIN sql_project1.sales_team s2
169        USING(_SalesTeamID)
170    JOIN sql_project1.product p
171        USING(_ProductID)
172    WHERE SalesTeam = TeamName;
173    END $$
174    DELIMITER ;
175 •  CALL sales_team_stats("Joshua Ryan");
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

OrderNumber	SalesChannel	WarehouseCode	ProcuredDate	OrderDate	ShipDate	DeliveryDate	CurrencyCode	_CustomerID	_StoreID	ProductName	OrderQuantity	DiscountApplied	Uni
SO - 000115	In-Store	WARE-NMK1003	2017-12-31	2018-06-01	2018-06-15	2018-06-20	USD	32	138	Computers	6	0.15	393
SO - 000117	In-Store	WARE-PUJ1005	2018-04-10	2018-06-01	2018-06-19	2018-06-27	USD	10	320	Computers	3	0.075	123
SO - 000128	In-Store	WARE-UHY1004	2017-12-31	2018-06-02	2018-06-20	2018-06-29	USD	32	238	Clocks	4	0.05	399
SO - 000140	In-Store	WARE-NMK1003	2017-12-31	2018-06-03	2018-06-18	2018-06-28	USD	26	196	TV and video	4	0.05	113
SO - 000169	In-Store	WARE-XYS1001	2017-12-31	2018-06-08	2018-06-11	2018-06-17	USD	1	30	Table Linens	2	0.15	105

I wanted to know the sales order of individual sales team, so that when I enter or input a particular sales team name, I get to see how customers ordered, what they ordered for, the quantity, price, etc.

CONCLUSION

- I've heard people talk about their love for the GROUP BY clause, I didn't quite understand why, but doing this project, I made use of it 90% of the queries.
- I made use of inner join all through because I was focusing majorly on the sales_order table.
- Importing the sales_order table took a long time because it contained 7991 entries. I might have to consider using another another DBMS in my next project.
- SQL might seem weird for starters, but it is really a straight-forward language if we understand the problem or question we are trying to answer.

Thanks for reading and going through my project's report. I am rooting for everyone who is still struggling with SQL queries, don't stop practicing!

THE END