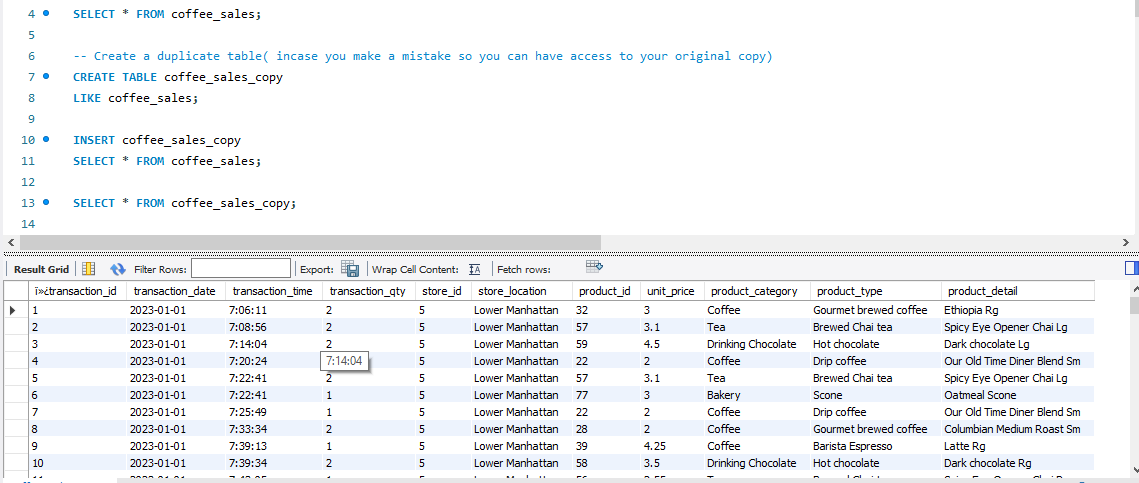
I will be performing a data cleaning and exploratory data analysis (EDA) in this project. I start off by

* Creating the database
* Perform a data cleaning on the dataset
* Conduct an EDA

First of all, let’s create the database;

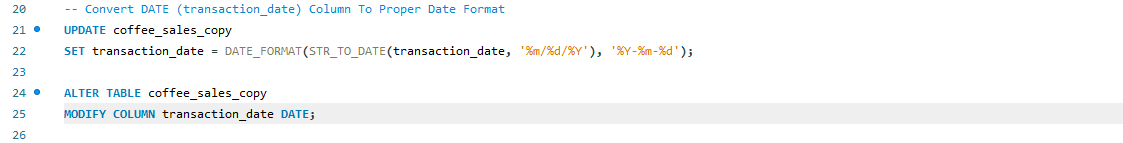


Then I import the data in the database. Now I’m set to view a snapshot of the data, and also create a duplicate table in the database so I can refer to the original one in case of any mistakes I commit. That means I will be using the duplicate for cleaning and analysis.



From here, I will begin cleaning the data.

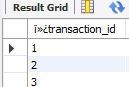
I will continue by converting the DATE (transaction\_date) column into a proper date format



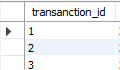
I will do the same for TIME (transaction\_time) column.



Then I will now change the column name `ï»¿transaction\_id` to transaction\_id

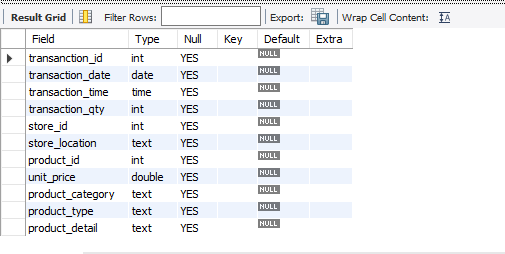




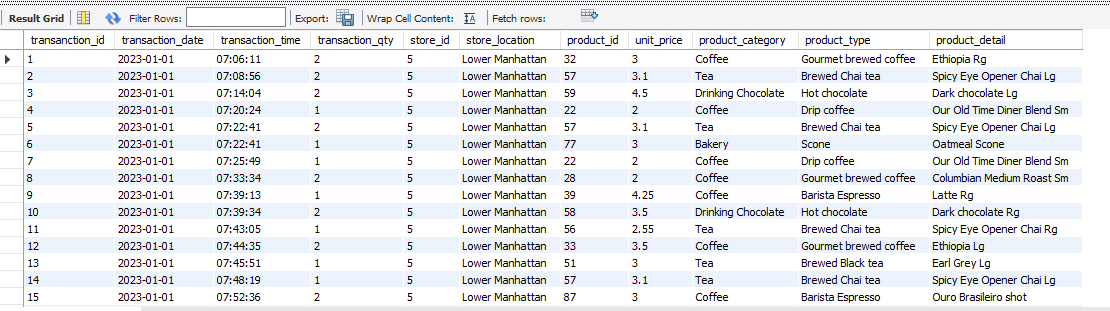


This is after changing the column name.

This is a description of the table

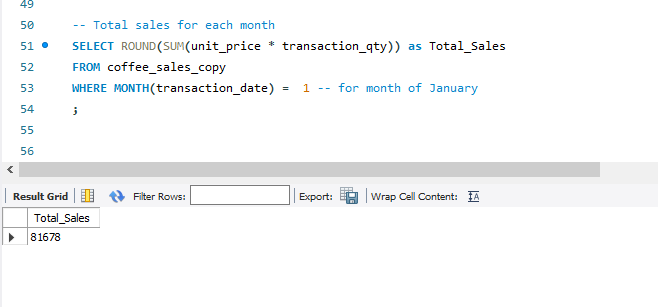


I will now preview a snapshot of the data after cleaning;



In the next step, I will begin the exploratory data analysis using the client’s KPI requirements.

1. **The client wants to find the total sales per month and also the month on month increase and decrease in sales.**



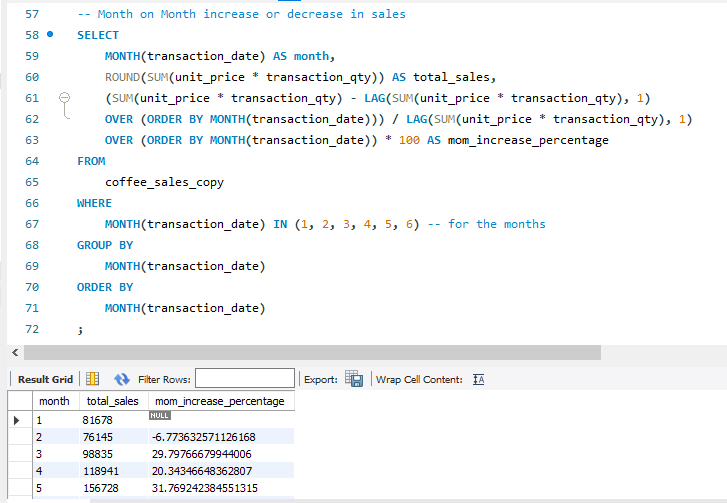
I used the WHERE function to filter out the particular month I want at that time, so anytime I want to find out the total sales for other months, I will just have to change the number ‘1’ to say ‘2’, that is if I want to know the total sales for February.

Note that: 1 = January 4= April

2= February 5= May

3= March 6= June

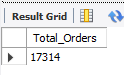
I used this function multiple times, it saves a lot of time.



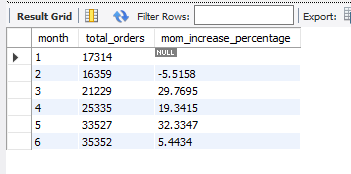
I used the LAG function here because I want to compare the differences in the sales for the current row value to that of the previous row value.

1. **The client is interested in knowing the total number of orders in each respective month and also the difference in the number of orders between the current month and previous month.**

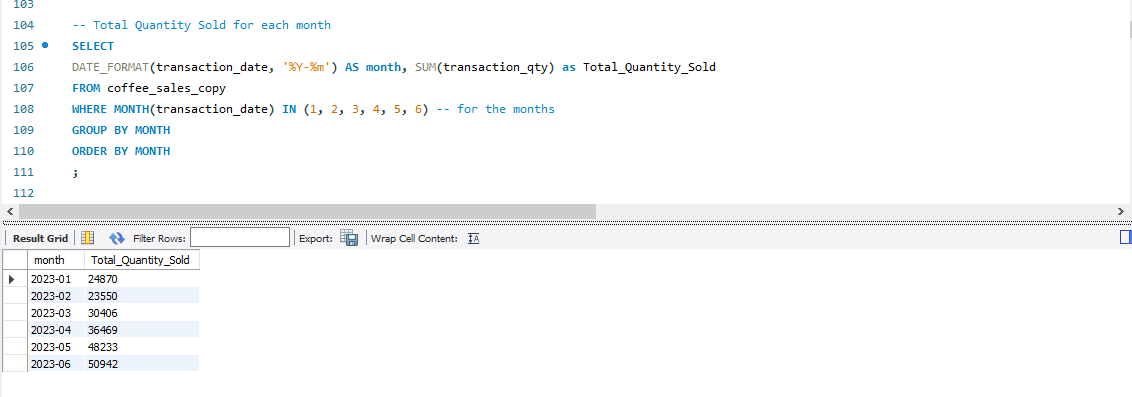
I used the same sql query I used for the first requirement but I selected the Count of (transaction\_id) instead of Total Sales( unit\_price \* transaction\_qty) to find total orders.



Same for the difference in number of orders

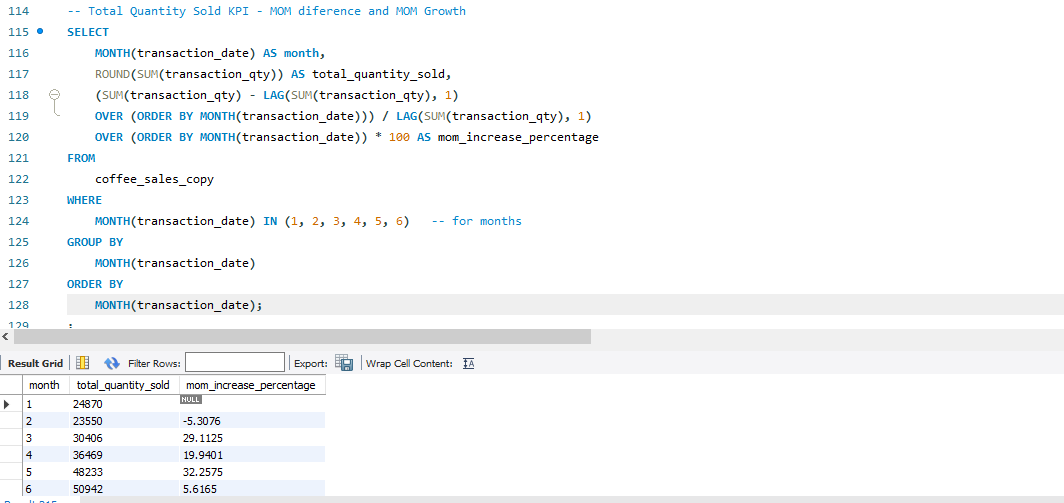


1. **The client wants to know the Total Quantity Sold for each month as well as the month to month increase and decrease in quantity sold.**

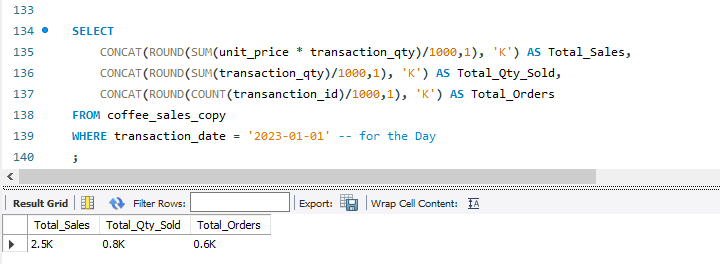


Here we have the total quantity that was sold per month.

This is a snapshot of the month on month increase in quantity sold, I used the LAG function here too for the difference of values between previous month and current month.



1. **Now we find the daily total sales, daily quantity sold and total orders as per the client’s demand.**

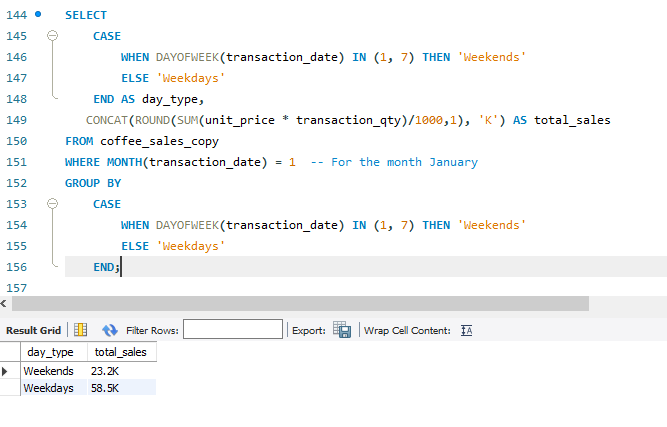


In this query, I used the CONCAT function to combine two strings or text together. As you can see from the snapshot, the output for Total sales is 2.5K, this was possible by using the CONCAT function to combine 2,500 and K together.

The WHERE function was also used to filter out the selected day. I can use any other date that’s in the database if I want to analyse other days as well.

1. **Weekend and Weekday Sales Analysis as per client’s demand**

In this query, I use the CASE function to make a decision in the query, kind of like saying "if this happens, do this; otherwise, do that." So in this query, my case was that ‘WHEN DAYOFWEEK(transaction\_date)’ falls in 1 and 7 then it’s a Weekend and therefore that falls between 2 and 6 is a Weekday. This is a snapshot of the query and its results:

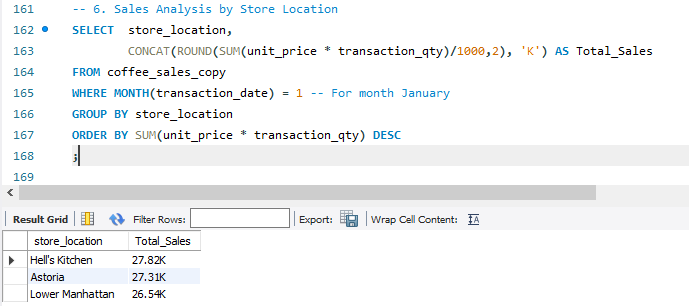


The WHERE function was used to filter out the particular month I wanted to see.

NOTE: As per my logic, the day of week starts on Sunday and ends on Saturday so that explains why I used 1 and 7 for weekends and 5-6 for weekdays.

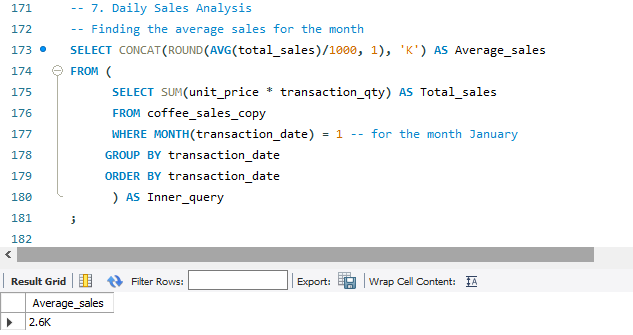
1. **Sales Analysis by Store Location**

We find the total sales for each store location.



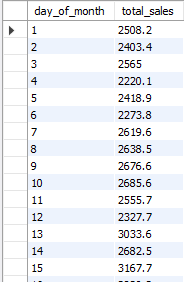
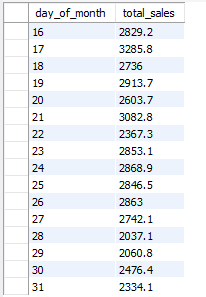
1. **We are now looking for the total sales in the days in a particular month and then we compare daily sales and average sales; whether above, below or equal to daily sales as per client’s demand.**

Before we find the daily sales for the month, we need first find the average sales for the month.

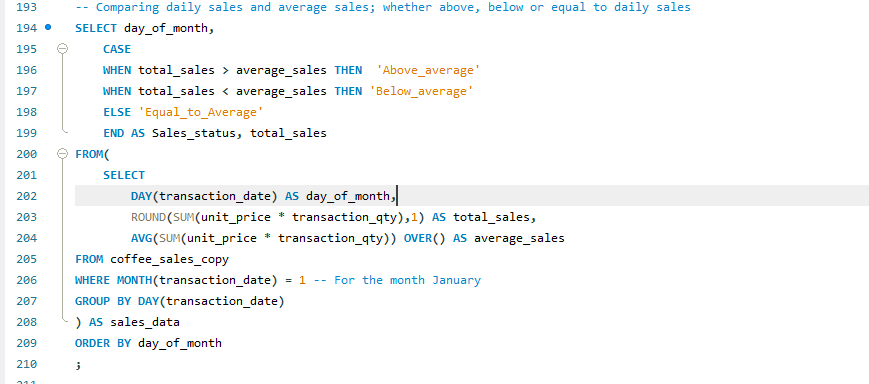


In this query, I used an inner query or a subquery because I had to first get some data then use that data to get more results. In this case, I had to first find the total sales from the coffee shop sales using the WHERE function to filter out the selected month, then I now used the data I just fetched to calculate the average sales for the selected month.

I proceed to find the daily sales in the selected month (January);

Now I have to compare the daily sales and average sales to indicate whether the daily sales is above, below and (or) equal to the average sales.

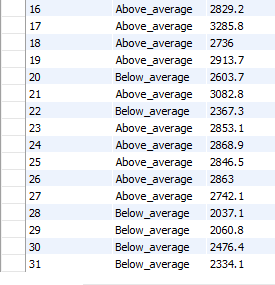
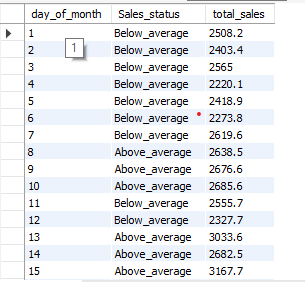


In this query, I used an inner query and an outer query. The CASE function was used to make a decision that if total sales is greater than average sales then status is above average.

The WHERE function was used to filter out the selected month.

The inner query was used to calculate the total sales per day and average sales across the days.

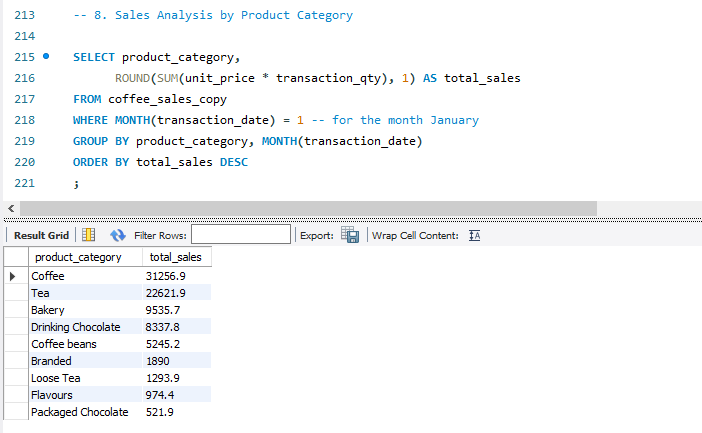
The outer query was used to compare and label the sales status.



This was how the results looked after.

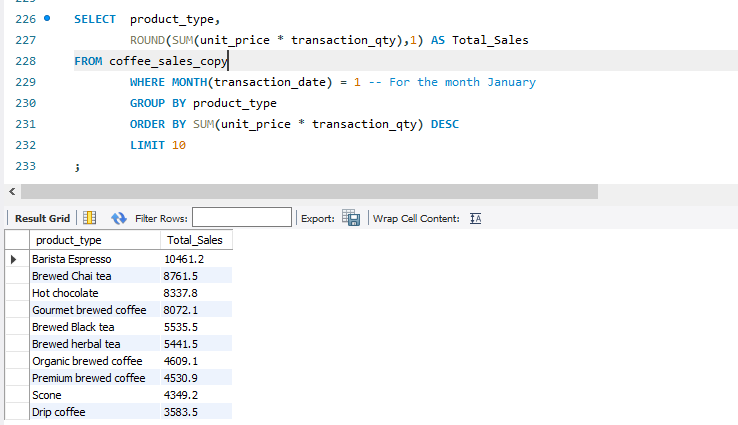
1. **Sales Analysis by Product Category**

In this query, I find the total sales for each product category



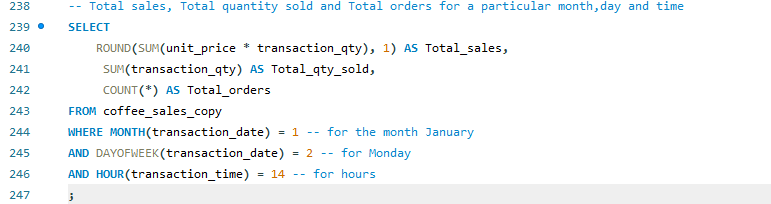
1. **Client wants to know the top 10 products by sales**

In this query, we find the total sales of the product type and then limit the results to 10 using the DESC function.

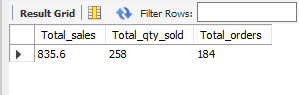


1. **Sales Analysis by Days and Hours as per client demand**

First of all, I look for the total sales, quantity and orders in a selected month, day and time.



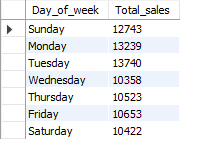
I used WHERE function to filter the selected month which is January, also I filtered the day which is Monday and the time which is in the 14th hour. This is the result or output for my sql query;



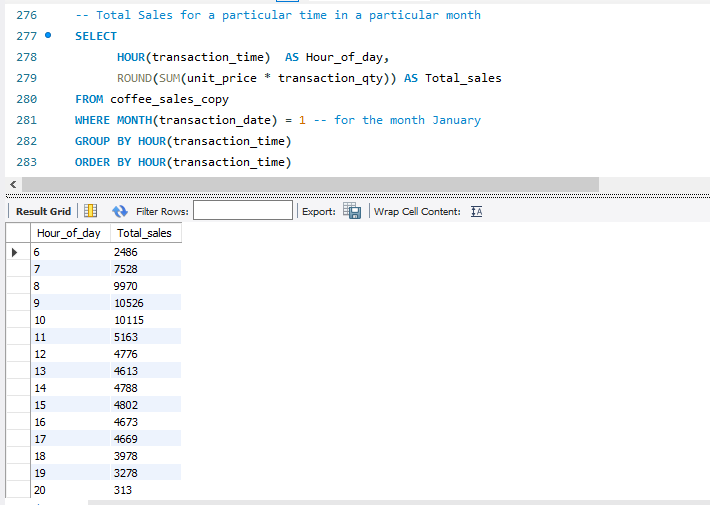
Now, I want to find how much coffee was sold from Monday to Sunday in a selected month (January).



This right here is the sql query I used to perform this analysis, I used the CASE statement to turn the numbers into actual day names, Monday, Tuesday up to Sunday. This helps the client to know which days of the week had the most sales or least sales.



Now I find the total sales per hour for a selected month.



The SQL queries documented above provide valuable insights into sales performance by day, helping to identify patterns and trends over time. By using functions like CASE, GROUP BY, WHERE and subqueries, we can transform raw transaction data into meaningful reports that support data-driven decision-making. These queries form a strong foundation for further analysis and can be adapted to suit a variety of business needs.