I wanted to implement my data analysis skill and I decided to analyse a pizza sales dataset. I chose pizza because it’s a well-known Fastfood with high purchasing rate and everybody, well almost everybody loves pizza.

I got my dataset from Kaggle which also came with some vital questions for insight, and I also added mine to spice things up.

**PROBLEM STATEMENT**

KPI’s Requirement: I generated some specific metrics for calculations, this allows me to analyse key indicators for the pizza sales data to gain insight into business performance.

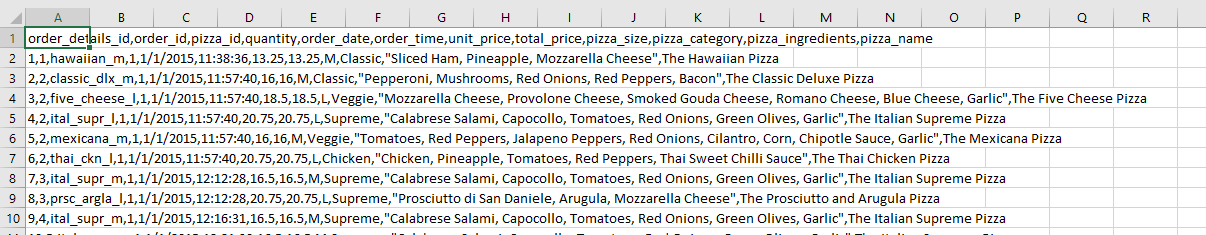
1. Total Revenue
2. Average order value
3. Total pizzas sold
4. Average pizzas per order

Charts Requirement: I would like to visualise various part of the pizza sales data to gain insight and understand key trends. I have identified the following requirements for creating charts.

1. Daily trend of total order.
2. Monthly trend of total orders.
3. Percentage of sales by pizza category.
4. Percentage of sales by pizza size.
5. Total pizza sold by pizzas category.
6. Top 5 best sellers by Revenue, Total Quantity and Total Orders.
7. Bottom 5 best sellers by Revenue, Total Quantity and Total Orders.

The Data

The dataset was in a form of a csv file (comma separated value). Here is an image of the original data (first 10 rows):



There are 11 total columns:

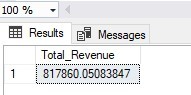
* pizza\_id
* order\_id
* quantity
* order\_date
* order\_time
* unit\_price
* total\_price
* pizza\_size
* pizza\_category
* pizza\_ingredients
* pizza\_name

KPI’s Requirement:

Query 1. Total revenue: The sum of all total price of all pizza orders.

SELECT SUM(total\_price) AS Total\_Revenue

FROM pizza\_sales



Query 2. Average Order Value: Calculated by dividing the total revenue by the total number of orders.

SELECT AVG(total\_price) AS average\_order\_value

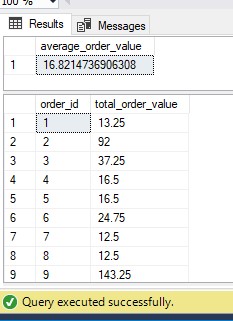
FROM pizza\_sales

SELECT order\_id, SUM(unit\_price) AS total\_order\_value

FROM pizza\_sales

GROUP BY order\_id

ORDER BY order\_id



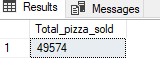
1. The inner subquery groups the rows by "order\_id" and calculates the sum of "unit\_price" for each unique order.
2. The outer query then calculates the average of the "total\_price" calculated in the subquery, giving me the AOV.

This query will give me the average order value considering all the unique order IDs, even if there are duplicate order IDs with different unit price.

Query 3. Total Pizzas Sold: This is the sum of the quantities of all pizzas sold.

SELECT SUM(quantity) AS Total\_pizza\_sold

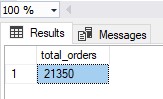
FROM pizza\_sales



Query 4. Total Orders: This is the total number of orders placed.

SELECT COUNT(DISTINCT order\_id) AS total\_orders

FROM pizza\_sales



To calculate the total number of orders when my order\_id has duplicate numbers but different purchases, I performed a COUNT of distinct order\_id values to ensure I am not double-counting orders with the same ID.

Query 5. Average Pizza sold