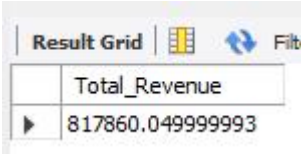
PIZZA SALES SQL QUERIES

A. KPI's

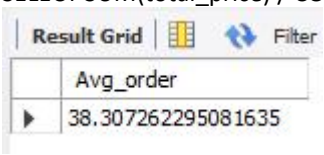
1. Total Revenue:

SELECT SUM(total_price) AS Total_Revenue FROM pizza_sales_excel_file;



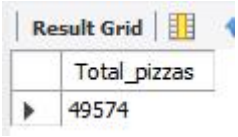
2. AVERAGE ORDER VALUE:

SELECT SUM(total_price) / COUNT(DISTINCT order_id) AS Avg_order FROM pizza_sales_excel_file;



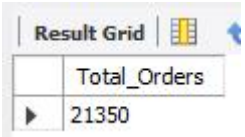
3. Total Pizzas sold:

SELECT SUM(quantity) AS Total_pizzas FROM pizza_sales_excel_file;



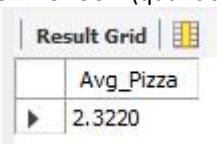
4. Total Orders:

SELECT COUNT(DISTINCT order_id) AS Total_Orders FROM pizza_sales_excel_file;



5. AVERAGE Pizzas per order:

SELECT SUM(quantity) / COUNT(DISTINCT order_id) AS Avg_Pizza FROM pizza_sales_excel_file;



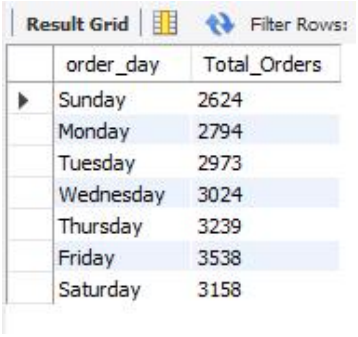
Then comes the next part:

CHARTS REQUIREMENT

1. Daily Trend for Total Orders: Create a bar chart that displays the daily trend of total orders over a specific time period. This chart will help us identify any patterns or fluctuations in order volumes on a daily basis. For this we use the query which is this:

```
SELECT
   DAYNAME(order_date) AS order_day,
   COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales_excel_file
WHERE (order_date) IS NOT NULL
GROUP BY
   DAYOFWEEK(order_date),
   DAYNAME(order_date)
```

ORDER BY DAYOFWEEK(order_date);



Result Grid | Filter Rows:

order_day	Total_Orders
Sunday	2624
Monday	2794
Tuesday	2973
Wednesday	3024
Thursday	3239
Friday	3538
Saturday	3158

2. <u>Monthly Trend for Total Orders:</u> Create a line chart that illustrates the hourly trend of total orders throughout the day. This chart will allow us to identify peak hours or periods of high orders activity. To make this we use this query:

```
SELECT
   m.month_name,
   IFNULL(t.Total_orders, 0) AS Total_orders
FROM (
   SELECT 1 AS month_num,  'January' AS month_name UNION
   SELECT 2, 'February' UNION
   SELECT 3, 'March' UNION
   SELECT 4, 'April' UNION
   SELECT 5, 'May' UNION
   SELECT 6, 'June' UNION
   SELECT 7, 'July' UNION
   SELECT 8, 'August' UNION
   SELECT 9, 'September' UNION
   SELECT 10, 'October' UNION
   SELECT 11, 'November' UNION
   SELECT 12, 'December'
) AS m
LEFT JOIN (
   SELECT
      MONTH(order_date) AS month_num,
      COUNT(DISTINCT order_id) AS Total_orders
   FROM pizza_sales_excel_file
   WHERE (order_date) IS NOT NULL
   GROUP BY MONTH(order_date)
) AS t
ON m.month_num = t.month_num
ORDER BY m.month_num;
```

Or the alternative query is

```
SELECT
   MONTHNAME(order_date) AS Month_name,
   COUNT(DISTINCT order_id) AS Total_orders
FROM pizza_sales_excel_file
WHERE order_date IS NOT NULL
GROUP BY MONTH(order_date),
      MONTHNAME(order_date)
ORDER BY MONTH(order_date);
```

Result Grid | Filter Rows:

month_name	Total_orders
January	1845
February	1685
March	1840
April	1799
May	1853
June	1773
July	1935
August	1841
September	1661
October	1646
November	1792
December	1680

3. <u>Percentage of Sales by Pizza Category :</u> Create a pie chart that shows the distribution of sales across different pizza categories. This chart will provide insight into the popularity of various pizza categories and their contribution to overall sales.

```
SELECT
   pizza_category,
   SUM(total_price) AS Total_Sales,
   ROUND(SUM(total_price) * 100 /
        (SELECT SUM(total_price)
         FROM pizza_sales_excel_file
         WHERE MONTH(order_date) = 1), 2) AS PCT
FROM pizza_sales_excel_file
WHERE MONTH(order_date) = 1
GROUP BY pizza_category;
```

Result Grid | Filter Rows:

pizza_category	Total_Sales	PCT
Classic	18619.4	26.68
Veggie	17055.400000000027	24.44
Supreme	17929.749999999996	25.69
Chicken	16188.75	23.2

4. <u>Percentage of Sales by Pizza SIZE :</u> Generate a pie chart that represents the percentage of sales attributed to different pizza sizes. This chart will help us understand customer preferences for pizza sizes and their impact on sales.

```
SELECT
   pizza_size,
   ROUND(SUM(total_price), 2) AS Total_Sales,
   ROUND(SUM(total_price) * 100 /
        (SELECT SUM(total_price)
         FROM pizza_sales_excel_file
         WHERE QUARTER(order_date) = 1), 2) AS PCT
FROM pizza_sales_excel_file
WHERE QUARTER(order_date) = 1
GROUP BY pizza_size
ORDER BY PCT DESC;
```

Result Grid | Filter Rows:

pizza_size	Total_Sales	PCT
L	95229.65	46.37
M	61159	29.78
S	45384.25	22.1
XL	3289.5	1.6
XXL	287.6	0.14

5. <u>Total Pizzas Sold by Pizza Category:</u> Create a funnel chart that presents the total number of pizzas sold for each pizza category. This chart will allow us to compare the sales performance of different pizza categories.

```
SELECT
   pizza_category,
   SUM(quantity) AS Total_Pizzas_Sold
FROM pizza_sales_excel_file
WHERE pizza_category IS NOT NULL
GROUP BY pizza_category
ORDER BY Total_Pizzas_Sold DESC;
```

Result Grid | Filter Rows:

pizza_category	Total_Pizzas_Sold
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

6. <u>Top 5 Best Sellers by Revenue, Total Quantity and Total Orders:</u> Create a bar chart highlighting the top 5 best-selling pizzas based on the Revenue, Total Quantity, Total Orders. This chart will help us identify the most popular pizza options.

```
SELECT
   pizza_name,
   SUM(total_price) AS Total_Revenue,
   SUM(quantity) AS Total_Quantity,
   COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales_excel_file
GROUP BY pizza_name
ORDER BY Total_Revenue DESC, Total_Quantity DESC, Total_Orders DESC
LIMIT 5;
```

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

pizza_name	Total_Revenue	Total_Quantity	Total_Orders
The Thai Chicken Pizza	43434.25	2371	2225
The Barbecue Chicken Pizza	42768	2432	2273
The California Chicken Pizza	41409.5	2370	2197
The Classic Deluxe Pizza	38180.5	2453	2329
The Spicy Italian Pizza	34831.25	1924	1822

And if we want individual columns into descending order then break the query separately into 3 parts with 3 columns.

Note: These are all the queries done, when the DATE and TIME column is in the original DATE and TIME format, if it is not and rather in text or VARCHAR then some query would be slightly different.