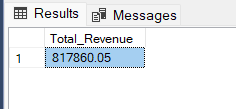
KPI's REQUIREMENT

1. ﻿﻿﻿Total Revenue:

SELECT ROUND(SUM(total\_price), 2) AS Total\_Revenue

FROM pizza\_sales;

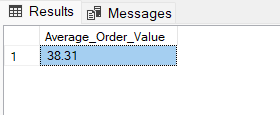
Total\_Revenue - 817’860.05



1. ﻿﻿﻿Average Order Value:

SELECT ROUND(ROUND(SUM(total\_price), 2) / COUNT(DISTINCT order\_id), 2) AS Average\_Order\_Value

FROM pizza\_sales;

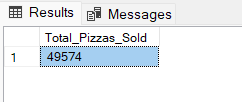
Average\_Order\_Value - 38.31  


1. Total Pizzas Sold:

SELECT SUM(quantity) AS Total\_Pizzas\_Sold

FROM pizza\_sales;

Total\_Pizzas\_Sold - 49’574



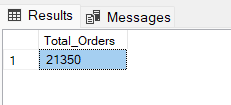
﻿﻿﻿

1. Total Orders:

SELECT COUNT(DISTINCT order\_id) AS Total\_Orders

FROM pizza\_sales;

Total\_Orders - 21’350

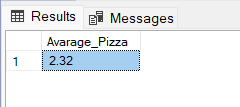


1. Average Pizzas Per Order:

SELECT CAST(CAST(SUM(quantity) AS decimal(10,2))/ CAST(COUNT(DISTINCT order\_id) AS decimal(10,2)) AS decimal(10,2)) AS Avarage\_Pizza

FROM pizza\_sales;

Avarage\_Pizza - 2.32



CHARTS REQUIREMENT

1. Daily Trend for Total Orders:

SET DATEFIRST 1;

SELECT DATENAME(DW, order\_date) AS Order\_day

,COUNT(DISTINCT order\_id) AS Total\_orders\_per\_day

FROM pizza\_sales

GROUP BY DATENAME(DW, order\_date), DATEPART(DW, order\_date)

ORDER BY DATEPART(DW, order\_date);

Order\_day Total\_orders\_per\_day

Monday 2’794

Tuesday 2’973

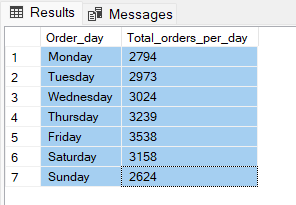
Wednesday 3’024

Thursday 3’239

Friday 3’538

Saturday 3’158

Sunday 2’624



1. Monthly Trend for Total Orders:

SELECT DATENAME(MONTH, order\_date) AS Month

,COUNT(DISTINCT order\_id) AS Total\_orders\_per\_month

FROM pizza\_sales

GROUP BY DATENAME(MONTH, order\_date), DATEPART(MONTH, order\_date)

ORDER BY DATEPART(MONTH, order\_date);

Month Total\_orders\_per\_month

January 1’845

February 1’685

March 1’840

April 1’799

May 1’853

June 1’773

July 1’935

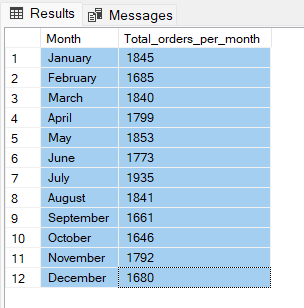
August 1’841

September 1’661

October 1’646

November 1’792

December 1’680



SELECT DATENAME(MONTH, order\_date) AS Month

,COUNT(DISTINCT order\_id) AS Total\_orders\_per\_month

FROM pizza\_sales

GROUP BY DATENAME(MONTH, order\_date)

ORDER BY Total\_orders\_per\_month DESC;

Month Total\_orders\_per\_month

July 1’935

May 1’853

January 1’845

August 1’841

March 1’840

April 1’799

November 1’792

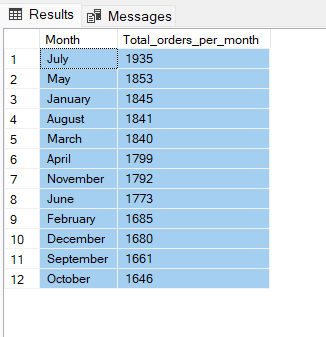
June 1’773

February 1’685

December 1’680

September 1’661

October 1’646



1. Percentage of Sales by Pizza Category:

SELECT pizza\_category

,ROUND(SUM(total\_price) \* 100.0 / (SELECT SUM(total\_price) FROM pizza\_sales), 3) AS '%\_Pizza\_Category'

FROM pizza\_sales

GROUP BY pizza\_category;

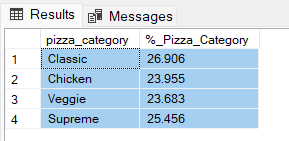
pizza\_category %\_Pizza\_Category

Classic 26.906

Chicken 23.955

Veggie 23.683

Supreme 25.456



SELECT pizza\_category

,ROUND(SUM(total\_price), 3) AS Total\_Sales

,ROUND(SUM(total\_price) \* 100.0 / (SELECT SUM(total\_price) FROM pizza\_sales), 3) AS '%\_Pizza\_Category'

FROM pizza\_sales

GROUP BY pizza\_category;

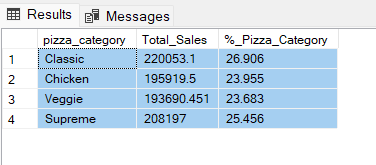
Pizza\_category Total\_Sales %\_Pizza\_Category

Classic 220’053.1 26.906

Chicken 195’919.5 23.955

Veggie 193’690.451 23.683

Supreme 208’197 25.456



1. Percentage of Sales by Pizza Size: 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.
2. Total Pizzas Sold by Pizza Category: 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.
3. [Top](https://6.top) 5 Best Sellers by Revenue, Total Quantity and Total Orders 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.
4. Bottom 5 Best Sellers by Revenue, Total Quantity and Total Orders Create a bar chart showcasing the bottom 5 worst-selling pizzas based on the Revenue, Total Quantity. Total Orders. This chart will enable us to identify underperforming or less popular pizza options.