PIZZA SALES SQL QUERIES

1. KPI’S
2. Total Revenue

Select SUM(total\_price) As Total\_Revenue from [Data Model - Pizza Sales]

Answer

A screenshot of a computer screen

Description automatically generated

1. Average order value

Select SUM(total\_price) / COUNT(Distinct order\_id) As Avg\_Order\_Value from [Data Model - Pizza Sales]

Answer

A screenshot of a computer

Description automatically generated

1. Total pizzas sold

Select SUM(quantity) As Total\_pizzas\_Sold from [Data Model - Pizza Sales]

Answer

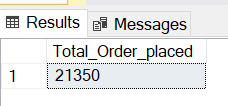
A screenshot of a computer

Description automatically generated

1. Total orders placed

Select COUNT(Distinct order\_id) As Total\_Order\_placed from [Data Model - Pizza Sales]

Answer



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 Average\_Pizza\_per\_Order from [Data Model - Pizza Sales]

Answer

A screenshot of a computer

Description automatically generated

CHARTS REQUIREMENT

1. Daily trend for total orders: Create a bar chart that displays the daily trend of total orders over a specific period of time. This chart will help us identify any patterns or fluctuations in order volumes on a daily basis.

Select DATENAME(DW, order\_date) As Order\_day, COUNT(DISTINCT order\_id) AS Total\_orders

from [Data Model - Pizza Sales]

GROUP by DATENAME(DW, order\_date)

Answer

A screenshot of a computer

Description automatically generated

1. Hourly Trend for total orders: Create a line chart that illustrates the hourly trend of hourly orders throughout the day. This chart will allow us to identify peak hours or periods of high order activity.

SELECT DATEPART(HOUR, order\_time) AS Order\_Hour, COUNT(DISTINCT order\_id) AS Total\_Orders

FROM [Data Model - Pizza Sales]

GROUP BY DATEPART(HOUR, order\_time)

ORDER BY DATEPART(HOUR, order\_time)

Answer

A screenshot of a table

Description automatically generated

1. Percentage of sales by pizza category: Create a pie chart that shows distribution of sales across different pizza category. 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, SUM(total\_price) \* 100 / (SELECT SUM(total\_price) from [Data Model - Pizza Sales]) AS PCT\_Total\_Sales

from [Data Model - Pizza Sales]

GROUP BY pizza\_category

ANSWER

A screenshot of a computer

Description automatically generated

1. Percentage of sales by pizza size: Generates a pie chart that represent 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, CAST(SUM(total\_price) AS DECIMAL (10,2)) As Total\_Sales, CAST(SUM(total\_price) \* 100 /

(SELECT SUM(total\_price) from [Data Model - Pizza Sales]) As DECIMAL (10,2)) AS PCT\_Total\_Sales

from [Data Model - Pizza Sales]

GROUP BY pizza\_size

ORDER BY PCT\_Total\_Sales DESC

ANSWER

A screenshot of a computer

Description automatically generated

--Percentage of sales by pizza size for first quarter

SELECT pizza\_size, CAST(SUM(total\_price) AS DECIMAL (10,2)) As Total\_Sales, CAST(SUM(total\_price) \* 100 /

(SELECT SUM(total\_price) from [Data Model - Pizza Sales] Where DATEPART(quarter, order\_date) = 1) As DECIMAL (10,2)) AS PCT\_Total\_Sales

from [Data Model - Pizza Sales]

Where DATEPART(quarter, order\_date) = 1

GROUP BY pizza\_size

ORDER BY PCT\_Total\_Sales DESC

Answer

A screenshot of a computer

Description automatically generated

1. Total pizza 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 category.

SELECT pizza\_category, SUM(quantity) As Total\_Pizzas\_sold

from [Data Model - Pizza Sales]

GROUP BY pizza\_category

ANSWER

A screenshot of a computer

Description automatically generated

1. Top 5 best sellers of by total pizzas sold: Create a bar chart highlighting the top 5 bestselling pizzas based on the total number of pizzas sold. This chart will help us identify the most popular pizza options.

SELECT TOP 5 pizza\_name, SUM(quantity) AS Total\_Pizzas\_sold

from [Data Model - Pizza Sales]

GROUP BY pizza\_name

ORDER BY Total\_Pizzas\_sold DESC

ANSWER

A screenshot of a menu

Description automatically generated

1. Bottom 5 worst sellers by pizzas sold: Create a bar chart showcasing the bottom 5 worst selling pizzas based on total number of pizzas sold. This chart will enable us identify underperforming or less pizza options.

SELECT TOP 5 pizza\_name, SUM(quantity) AS Total\_Pizzas\_sold

from [Data Model - Pizza Sales]

GROUP BY pizza\_name

ORDER BY Total\_Pizzas\_sold ASC

ANSWER

A screenshot of a menu

Description automatically generated

NOTES

If you want to apply the month, quarter, week filters to the above queries, you can use where clause. Follow the below examples.

SELECT pizza\_category, SUM(total\_price) As Total\_Sales, SUM(total\_price) \* 100 /

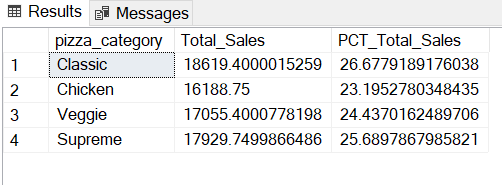
(SELECT SUM(total\_price) from [Data Model - Pizza Sales] Where MONTH(order\_date) = 1) AS PCT\_Total\_Sales

from [Data Model - Pizza Sales]

Where MONTH(order\_date) = 1

GROUP BY pizza\_category

ANSWER



* Where Month(order\_date) = 1 indicates that the output is for the month of January, Month(order\_date) = 4 indicates the month of April. Also why the where function is added at the sub query is to give the right answer because it is taking the total sales

Select DATENAME(DW, order\_date) As Order\_day, COUNT(DISTINCT order\_id) AS Total\_orders

from [Data Model - Pizza Sales]

Where DATEPART(Quarter, order\_date) = 1

GROUP by DATENAME(DW, order\_date)

* Where DARTPART(Quarter, order\_date) = 1 indicates that the output is for quarter 1, Month(order\_date) = 3 indicates quarter 3