PIZZA SALES QUIRES

**PROBLEM STATEMENT**

**KPI's REQUIREMENT**

We need to analyze key indicators for our pizza sales data to gain insights into our business performance. Specifically, we want to calculate the following metrics:

1. **Total Revenue:** The sum of the total price of all pizza orders.
2. **Average Order Value:** The average amount spent per order, calculated by dividing the total revenue by the total number of orders.
3. **Total Pizzas Sold:** The sum of the quantities of all pizzas sold.
4. **Total Orders:** The total number of orders placed.
5. **Average Pizzas Per Order:** The average number of pizzas sold per order, calculated by dividing the total number of pizzas sold by the total number of orders.
6. KPI’s
7. Total\_Revenue:

SELECT SUM(total\_price) AS Total\_Revenue FROM pizza\_sales

A screenshot of a computer

AI-generated content may be incorrect.

1. Average\_Order\_Value:

SELECT SUM(total\_price) / COUNT(DISTINCT order\_id) AS Average\_Order\_Value FROM pizza\_sales;

A screenshot of a computer

AI-generated content may be incorrect.

1. Total\_Pizza\_Sold:

SELECT SUM(quantity) AS Total\_Pizza\_Sold FROM pizza\_sales;

A screenshot of a menu

AI-generated content may be incorrect.

1. Total\_Orders:

SELECT COUNT(DISTINCT order\_id) AS Total\_Orders FROM pizza\_sales;

A screenshot of a phone

AI-generated content may be incorrect.

1. Average\_Pizza\_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 pizza\_sales;

A screenshot of a computer

AI-generated content may be incorrect.

**PROBLEM STATEMENT**

**CHARTS REQUIREMENT**

We would like to visualize various aspects of our pizza sales data to gain insights and understand key trends. We have identified the following requirements for creating charts:

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.
2. **Hourly Trend for Total Orders**:  
   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 order activity.
3. **Percentage of Sales by Pizza Category**:  
   Create a pie chart that shows the distribution of sales across different pizza categories. This chart will provide insights into the popularity of various pizza categories and their contribution to overall sales.
4. CHARTS REQUIREMENT
5. Daily Trend for Total Orders:

-- Daily Trend

-- Count distinct orders by day of week

SELECT

DAYNAME(order\_date) AS day\_of\_week,

COUNT(DISTINCT order\_id) AS total\_orders

FROM

pizza\_sales

GROUP BY

DAYNAME(order\_date)

ORDER BY

FIELD(day\_of\_week, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday');

A black and white table with numbers

AI-generated content may be incorrect.

1. Hourly Trend for Total Orders:

-- Hourly Trend

SELECT

HOUR(order\_time) AS hour\_of\_day,

COUNT(DISTINCT order\_id) AS total\_orders

FROM

pizza\_sales

GROUP BY

HOUR(order\_time)

ORDER BY

HOUR(order\_time);

A black and white rectangular table with numbers

AI-generated content may be incorrect.

1. Percentage of Sales by Pizza Category:

SELECT pizza\_category, sum(total\_price) \* 100 / (SELECT sum(total\_price) from pizza\_sales) as Total\_Sales

from pizza\_sales

GROUP BY pizza\_category;

A screenshot of a black and white screen

AI-generated content may be incorrect.

# NOTE : If you want to apply the Month, Quarter, Week filters to the above queries you can use WHERE clause. Follow some of below examples

SELECT pizza\_category, sum(total\_price) \* 100 / (SELECT sum(total\_price) from pizza\_sales) as Total\_Sales

from pizza\_sales

WHERE month(order\_date)= 1

GROUP BY pizza\_category;

A screenshot of a computer

AI-generated content may be incorrect.

# Here MONTH(order\_date) = 1 indicates that the output is for the month of January.

MONTH(order\_date) = 4 indicates output for Month of April.

PROBLEM STATEMENT

CHARTS REQUIREMENT

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.

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.

Top 5 Best Sellers by Total Pizzas Sold:

Create a bar chart highlighting the top 5 best-selling pizzas based on the total number of pizzas sold. This chart will help us identify the most popular pizza options.

Bottom 5 Worst Sellers by Total Pizzas Sold:

Create a bar chart showcasing the bottom 5 worst-selling pizzas based on the total number of pizzas sold. This chart will enable us to identify underperforming or less popular pizza options.

1. Percentage of Sales by Pizza Size:

SELECT

pizza\_size,

ROUND(SUM(total\_price), 2) AS total\_sales,

ROUND(SUM(total\_price) \* 100 / (SELECT SUM(total\_price) FROM pizza\_sales), 2) AS percentage

FROM

pizza\_sales

GROUP BY

pizza\_size

ORDER BY

percentage DESC;

A screenshot of a graph

AI-generated content may be incorrect.

1. Total Pizza Sold by Pizza Category:

SELECT pizza\_category, sum(quantity) as Total\_Pizzas\_Sold from pizza\_sales

Group by pizza\_category;

A screenshot of a black and white screen

AI-generated content may be incorrect.

1. Top 5 Best Sellers by Total Pizzas Sold:

SELECT

pizza\_name,

SUM(quantity) AS total\_pizzas\_sold

FROM

pizza\_sales

GROUP BY

pizza\_name

ORDER BY

total\_pizzas\_sold DESC

LIMIT 5;

A black and white screen with white text

AI-generated content may be incorrect.

1. Bottom 5 Best Sellers by Total Pizzas Sold:

SELECT

pizza\_name,

SUM(quantity) AS total\_pizzas\_sold

FROM

pizza\_sales

GROUP BY

pizza\_name

ORDER BY

total\_pizzas\_sold ASC -- Changed to ASC for bottom values

LIMIT 5;

A black and white screen with white text

AI-generated content may be incorrect.

Total SQL Quire :

-- Create the PizzaDB database

CREATE DATABASE PizzaDB;

-- Set PizzaDB as the active database

USE PizzaDB;

-- View all raw data from pizza\_sales table

SELECT \* FROM pizza\_sales;

-- Key Performance Indicators (KPIs)

-- 1. Calculate total revenue from all pizza sales

SELECT SUM(total\_price) AS Total\_Revenue FROM pizza\_sales;

-- 2. Calculate average order value (total revenue divided by number of orders)

SELECT SUM(total\_price) / COUNT(DISTINCT order\_id) AS Average\_Order\_Value FROM pizza\_sales;

-- 3. Calculate total quantity of pizzas sold

SELECT SUM(quantity) AS Total\_Pizza\_Sold FROM pizza\_sales;

-- 4. Count total number of orders placed

SELECT COUNT(DISTINCT order\_id) AS Total\_Orders FROM pizza\_sales;

-- 5. Calculate average number of 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 pizza\_sales;

-- Sales Trends Analysis

-- Daily trend: Orders by day of week

SELECT

DAYNAME(order\_date) AS day\_of\_week,

COUNT(DISTINCT order\_id) AS total\_orders

FROM

pizza\_sales

GROUP BY

DAYNAME(order\_date)

ORDER BY

FIELD(day\_of\_week, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday');

-- Hourly trend: Orders by hour of day

SELECT

HOUR(order\_time) AS hour\_of\_day,

COUNT(DISTINCT order\_id) AS total\_orders

FROM

pizza\_sales

GROUP BY

HOUR(order\_time)

ORDER BY

HOUR(order\_time);

-- Sales Breakdown by Category

-- Percentage of sales by pizza category (for January only)

SELECT pizza\_category, sum(total\_price) \* 100 / (SELECT sum(total\_price) from pizza\_sales) as Total\_Sales

from pizza\_sales

WHERE month(order\_date)= 1

GROUP BY pizza\_category;

-- Percentage of sales by pizza size (with rounded values)

SELECT

pizza\_size,

ROUND(SUM(total\_price), 2) AS total\_sales,

ROUND(SUM(total\_price) \* 100 / (SELECT SUM(total\_price) FROM pizza\_sales), 2) AS percentage

FROM

pizza\_sales

GROUP BY

pizza\_size

ORDER BY

percentage DESC;

-- Total pizzas sold by category

SELECT pizza\_category, sum(quantity) as Total\_Pizzas\_Sold from pizza\_sales

Group by pizza\_category;

-- Product Performance Analysis

-- Top 5 best-selling pizzas by quantity

SELECT

pizza\_name,

SUM(quantity) AS total\_pizzas\_sold

FROM

pizza\_sales

GROUP BY

pizza\_name

ORDER BY

total\_pizzas\_sold DESC

LIMIT 5;

-- Bottom 5 worst-selling pizzas by quantity

SELECT

pizza\_name,

SUM(quantity) AS total\_pizzas\_sold

FROM

pizza\_sales

GROUP BY

pizza\_name

ORDER BY

total\_pizzas\_sold ASC

LIMIT 5;