PIZZA STORE ANALYSIS -

SQL QUERIES FOR OBTAINING

THE KPIS

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.

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. **Monthly 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. **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.

5. **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.

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, and total orders. This chart will help us identify the most popular pizza options.

7. **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, and total orders. This chart will enable us

to identify underperforming or less popular pizza options.

SQL QUERIES FOR THE SAME

A. Total Revenue

```
SELECT SUM(total_price) AS Total_Revenue
FROM pizza_sales;
```

B. Average Order Value

```
SELECT (SUM(total_price) / COUNT(DISTINCT order_id)) AS
Avg_order_Value
FROM pizza_sales;
```

C. Total Pizzas Sold

```
SELECT SUM(quantity) AS Total_pizza_sold
FROM pizza_sales;
```

D. Total Orders

```
SELECT COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales;
```

E. Average Pizzas Per Order

```
SELECT
  ROUND(SUM(quantity) / COUNT(DISTINCT order_id), 2) AS
Avg_Pizzas_per_order
FROM pizza_sales;
```

F. Daily Trend for Total Orders

```
SELECT
  DAYNAME(order_date) AS order_day,
  COUNT(DISTINCT order_id) AS total_orders
FROM pizza_sales
GROUP BY DAYNAME(order_date);
```

G. Monthly Trend for Orders

```
SELECT
  MONTHNAME(order_date) AS Month_Name,
  COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY MONTHNAME(order_date);
```

H. % of Sales by Pizza Category

```
SELECT
  pizza_category,
  ROUND(SUM(total_price), 2) AS total_revenue,
  ROUND(SUM(total_price) * 100 / (SELECT SUM(total_price) FROM
pizza_sales), 2) AS PCT
FROM pizza_sales
GROUP BY pizza_category;
```

I. % of Sales by Pizza Size

```
SELECT
  pizza_size,
```

```
  ROUND(SUM(total_price), 2) AS total_revenue,
  ROUND(SUM(total_price) * 100 / (SELECT SUM(total_price) FROM
pizza_sales), 2) AS PCT
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size;
```

J. Total Pizzas Sold by Pizza Category (for February)

```
SELECT
  pizza_category,
  SUM(quantity) AS Total_Quantity_Sold
FROM pizza_sales
WHERE MONTH(order_date) = 2
GROUP BY pizza_category
ORDER BY Total_Quantity_Sold DESC;
```

K. Top 5 Pizzas by Revenue

```
SELECT
  pizza_name,
  SUM(total_price) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue DESC
LIMIT 5;
```

L. Bottom 5 Pizzas by Revenue

```
SELECT
  pizza_name,
  SUM(total_price) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue ASC
LIMIT 5;
```

M. Top 5 Pizzas by Quantity Sold

```
SELECT
  pizza_name,
  SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold DESC
LIMIT 5;
```

N. Bottom 5 Pizzas by Quantity Sold

```
SELECT
  pizza_name,
  SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold ASC
LIMIT 5;
```

O. Top 5 Pizzas by Total Orders

```
SELECT
  pizza_name,
  COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders DESC
LIMIT 5;
```

P. Bottom 5 Pizzas by Total Orders

```
SELECT
```

```
  pizza_name,
  COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders ASC
LIMIT 5;
```

DASHBOARD - POWER BI