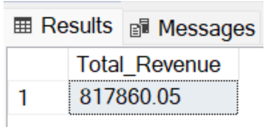
PIZZA SALES SQL QUERIES

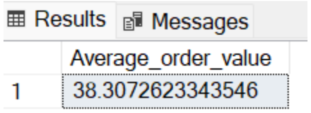
KPI’s

```
-- 1. TOTAL REVENUE
SELECT ROUND(SUM(total_price),2) AS Total_Revenue
FROM pizza_sales;
```



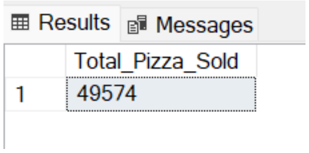
	Total_Revenue
1	817860.05

```
--2. AVERAGE ORDER VALUE
SELECT SUM(total_price)/COUNT(DISTINCT order_id) AS Average_order_value
FROM pizza_sales;
```



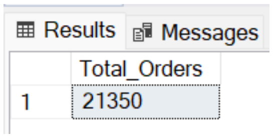
	Average_order_value
1	38.3072623343546

```
--3. TOTAL PIZZA SOLD
SELECT SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales;
```



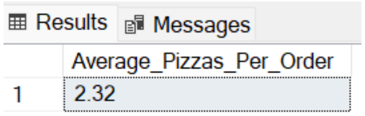
	Total_Pizza_Sold
1	49574

```
--4. TOTAL ORDERS
SELECT COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales;
```



	Total_Orders
1	21350

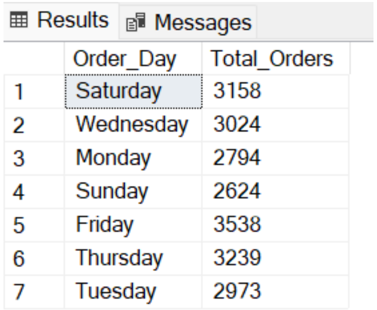
```
--5. AVERAGE PIZZA PER ORDER
SELECT CAST(CAST(SUM(quantity)AS DECIMAL(10,2)) /CAST(COUNT(DISTINCT order_id) AS
DECIMAL(10,2)) AS DECIMAL(10,2)) AS Average_Pizzas_Per_Order
FROM pizza_sales;
```



	Average_Pizzas_Per_Order
1	2.32

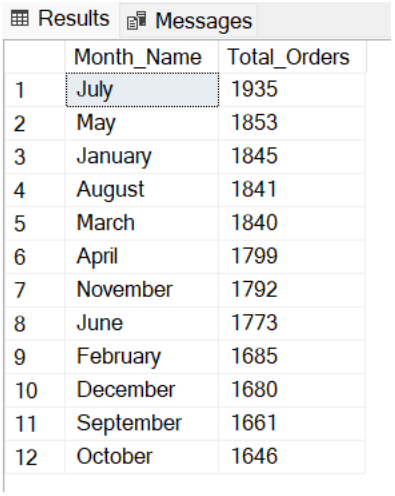
CHARTS REQUIREMENT

```
--1 DAILY TREND OF TOTAL ORDERSORDER
SELECT
       DATENAME(DW, Order_date) as Order_Day,
       COUNT(DISTINCT Order_id) AS Total_Orders
FROM pizza_sales
GROUP BY DATENAME(DW, Order_date);
```



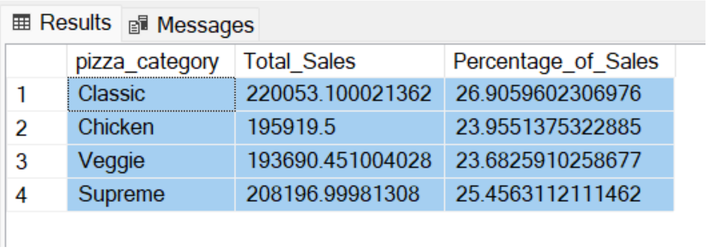
	Order_Day	Total_Orders
1	Saturday	3158
2	Wednesday	3024
3	Monday	2794
4	Sunday	2624
5	Friday	3538
6	Thursday	3239
7	Tuesday	2973

```
--2 MONTHLY TREND OF TOTAL ORDERS
SELECT
       DATENAME(MONTH, Order_date) as Month_Name,
       COUNT(DISTINCT Order_id) AS Total_Orders
FROM pizza_sales
GROUP BY DATENAME(MONTH, Order_date)
ORDER BY Total_Orders DESC;
```



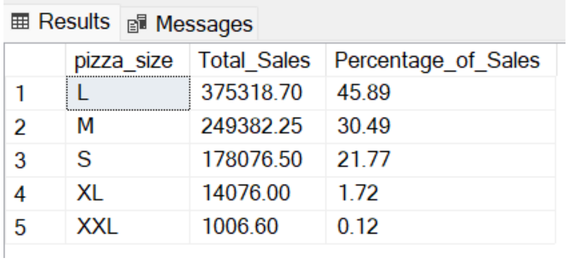
	Month_Name	Total_Orders
1	July	1935
2	May	1853
3	January	1845
4	August	1841
5	March	1840
6	April	1799
7	November	1792
8	June	1773
9	February	1685
10	December	1680
11	September	1661
12	October	1646

```
--3. PERCENTAGE OF SALES BY PIZZA CATEGORY
SELECT
       pizza_category, SUM(total_price) AS Total_Sales,
       SUM(total_price)*100/(SELECT SUM(total_price) FROM pizza_sales) AS
Percentage_of_Sales
FROM pizza_sales
GROUP BY pizza_category;
```



	pizza_category	Total_Sales	Percentage_of_Sales
1	Classic	220053.100021362	26.9059602306976
2	Chicken	195919.5	23.9551375322885
3	Veggie	193690.451004028	23.6825910258677
4	Supreme	208196.99981308	25.4563112111462

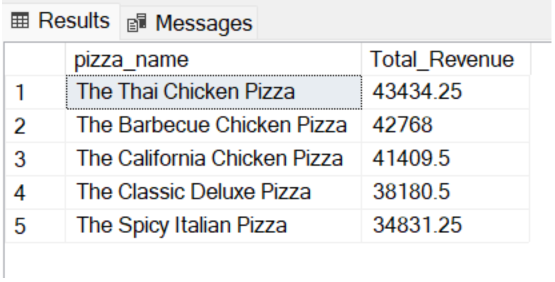
```
--4. PERCENTAGE OF SALES BY PIZZA SIZE
SELECT
       pizza_size, CAST(SUM(total_price) AS DECIMAL(10,2)) AS Total_Sales,
       CAST(SUM(total_price)*100/(SELECT SUM(total_price) FROM pizza_sales) AS
DECIMAL(10,2)) AS Percentage_of_Sales
FROM pizza_sales
GROUP BY pizza_size
ORDER BY Percentage_of_Sales DESC;
```



	pizza_size	Total_Sales	Percentage_of_Sales
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

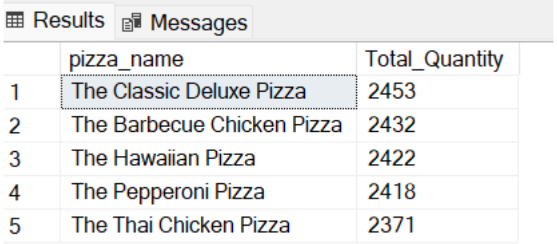
```
--5. TOP 5 BEST SELLERS BY REVENUE, TOTAL QUANTITY AND TOTAL ORDERS

TOP 5 BEST SELLERS BY REVENUE
SELECT TOP 5 pizza_name, SUM(total_price ) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue DESC;
```



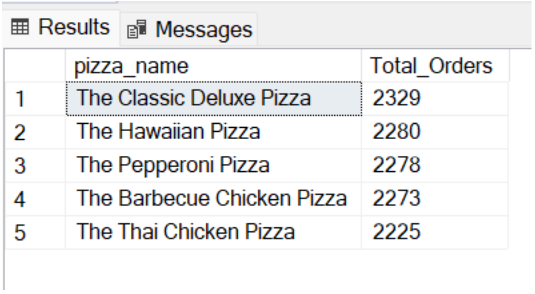
	pizza_name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Spicy Italian Pizza	34831.25

```
TOP 5 BEST SELLERS BY TOTAL QUANTITY
SELECT TOP 5 pizza_name, SUM(quantity) AS Total_Quantity
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Quantity DESC;
```



	pizza_name	Total_Quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

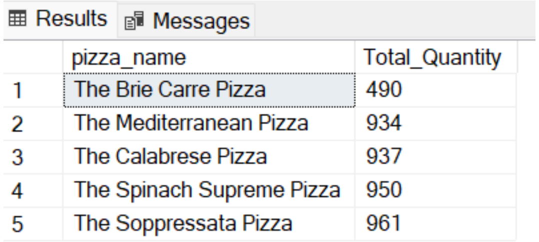
```
TOP 5 BEST SELLERS BY TOTAL ORDERS
SELECT TOP 5 pizza_name, COUNT(DISTINCT order_id ) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders DESC;
```



	pizza_name	Total_Orders
1	The Classic Deluxe Pizza	2329
2	The Hawaiian Pizza	2280
3	The Pepperoni Pizza	2278
4	The Barbecue Chicken Pizza	2273
5	The Thai Chicken Pizza	2225

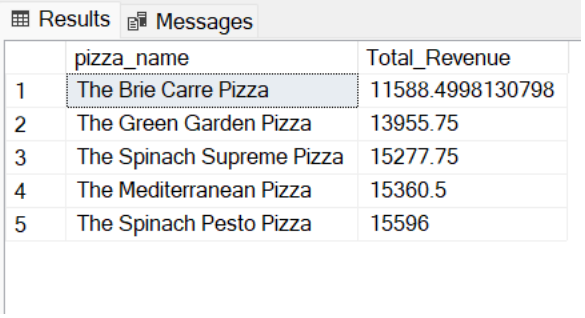
```
--5. TOP 5 WORST SELLERS BY REVENUE, TOTAL QUANTITY AND TOTAL ORDERS

TOP 5 WORST SELLERS BY TOTAL QUANTITY
SELECT TOP 5 pizza_name, SUM(quantity) AS Total_Quantity
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Quantity ASC;
```



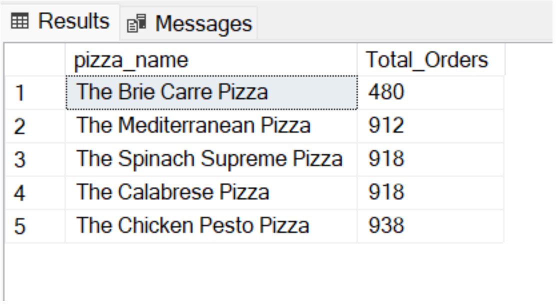
	pizza_name	Total_Quantity
1	The Brie Carre Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppressata Pizza	961

```
TOP 5 WORST SELLERS BY REVENUE
SELECT TOP 5 pizza_name, SUM(total_price ) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue ASC;
```



	pizza_name	Total_Revenue
1	The Brie Carre Pizza	11588.4998130798
2	The Green Garden Pizza	13955.75
3	The Spinach Supreme Pizza	15277.75
4	The Mediterranean Pizza	15360.5
5	The Spinach Pesto Pizza	15596

```
TOP 5 WORST SELLERS BY TOTAL ORDERS
SELECT TOP 5 pizza_name, COUNT(DISTINCT order_id ) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders ASC;
```



	pizza_name	Total_Orders
1	The Brie Carre Pizza	480
2	The Mediterranean Pizza	912
3	The Spinach Supreme Pizza	918
4	The Calabrese Pizza	918
5	The Chicken Pesto Pizza	938

```
--NOTE
/*If we want to apply the Month, Quarter, Week filters to the above queries we
can use*/
WHERE clause. Follow some of below examples
SELECT DATENAME(DW, order date) AS order day, COUNT(DISTINCT order_id) AS
total orders
FROM pizza sales
WHERE MONTH(order date) = 1
GROUP BY DATENAME(DW, order date);
/*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.*/

SELECT DATENAME(DW, order date) AS order day, COUNT(DISTINCT order id) AS
total orders
FROM pizza sales
WHERE DATEPART(QUARTER, order date) = 1
GROUP BY DATENAME(DW, order date)
/*Here DATEPART(QUARTER, order date) = 1 indicates that the output is for
the Quarter 1. MONTH(order date) = 3 indicates output for Quarter 3.

SELECT pizza_category, sum(total_price) as Total_Sales, sum(total_price) * 100 /
```

```
(SELECT sum(total_price) from pizza_sales WHERE MONTH(order_date) = 1) AS PCT
from pizza_sales
WHERE MONTH(order_date) = 1
GROUP BY pizza_category;
/*Here WHERE MONTH(order_date) = 1 filters the output for the Month of January. Also, if
it is applied to the main query, it should also be used in the subquery to get an
accurate result.
```