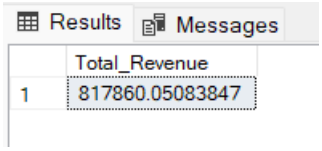
PIZZA SALES SQL QUERIES

A. KPI's

1. **Total Revenue**

   SELECT SUM(total_price) AS Total_Revenue FROM pizza_sales;

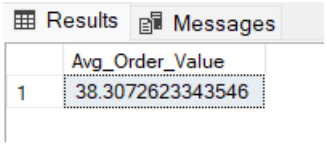
   Output:



2. **Average Order Value**

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

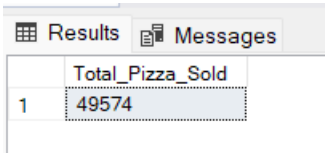
   Output:



3. **Total Pizzas Sold**

   SELECT SUM(quantity) AS Total_Pizza_Sold
   FROM pizza_sales

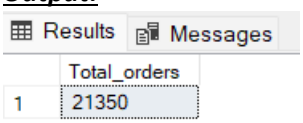
   Output:



4. **Total Orders**

   SELECT COUNT(DISTINCT order_id) AS Total_orders
   FROM pizza_sales

   Output:



5. **Average Pizzas Per Order**

// we have to convert it in decimal format so we are using cast method

```
SELECT CAST(CAST(SUM(quantity) AS DECIMAL(10,2)) /
CAST(COUNT(DISTINCT order_id) AS DECIMAL(10,2)) AS DECIMAL(10,2))
AS Avg_Pizzas_per_order
FROM pizza_sales;
```

Output:

Results | Messages

	Avg_Pizzas_per_order
1	2.32

B. Daily Trend for Total Orders

```
SELECT DATENAME(DW, order_date) AS order_day,  COUNT(DISTINCT order_id) AS
Total_orders
FROM pizza_sales
GROUP BY DATENAME(DW, order_date)
```

Output:

Results | Messages

	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

C. Monthly Trend for 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;
```

Output:

Results Messages

	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

D. Hourly Trend for Orders

SELECT DATEPART(HOUR, order_time) AS order_hours, COUNT(DISTINCT order_id) AS Total_orders
FROM pizza_sales
GROUP BY DATEPART(HOUR, order_time)
ORDER BY DATEPART(HOUR, order_time)

Output:

Results Messages

	order_hours	Total_orders
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

E. % 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 where month(order_date) = 1)  AS
PCT
FROM pizza_sales
where MONTH(order_date) = 1
GROUP BY pizza_category;
```

Output:

	pizza_category	Total_Sales	PCT
1	Classic	18619.4000015259	26.6779189176038
2	Chicken	16188.75	23.1952780348435
3	Veggie	17055.4000778198	24.4370162489706
4	Supreme	17929.7499866486	25.6897867985821

Results Messages

	pizza_category	Total_Revenue	PCT
1	Classic	220053.10	26.91
2	Chicken	195919.50	23.96
3	Veggie	193690.45	23.68
4	Supreme	208197.00	25.46

F. % of Sales by Pizza Size

```
SELECT pizza_size , CAST(SUM(total_price) AS DECIMAL(10,2)) AS Total_Revenue ,
CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) FROM pizza_sales) AS
DECIMAL(10,2))AS PCT
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size;
```

Output:

Results Messages

	pizza_size	Total_Revenue	PCT
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

Or

```
SELECT pizza_size , SUM(total_price) AS Total_Sales, SUM(total_price) * 100 /
(SELECT SUM(total_price) FROM pizza_sales)  AS PCT
FROM pizza_sales
GROUP BY pizza_size
Order by pct desc;
```

Output:

	pizza_size	Total_Sales	PCT
1	L	375318.701004028	45.8903330244889
2	M	249382.25	30.492044420599
3	S	178076.49981308	21.7734684107037
4	XL	14076	1.72107684995364
5	XXL	1006.6000213623	0.123077294254...

G. Total Pizzas Sold by Pizza Category

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

Output:

Results | Messages

	pizza_category	Total_Quantity_Sold
1	Classic	1178
2	Supreme	964
3	Veggie	944
4	Chicken	875

H. Top 5 Pizzas 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;
```

Output:

Results | Messages

	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

I. Bottom5 Pizzas 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;
```

Output:

Results | Messages

	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

J. Top 5 Pizzas by Quantity

```
SELECT TOP 5 pizza_name , SUM(quantity) AS Total_quantity FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_quantity DESC
```

Results | Messages

	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

K. Bottom 5 Pizzas by Quantity

```
SELECT TOP 5 pizza_name , SUM(quantity) AS Total_quantity FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_quantity ASC
```

Results | Messages

	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

L. Top 5 Pizzas Sellers with respect to 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
```

Results | Messages

	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

// bottom 5 pizza sellers with respect to 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;
```