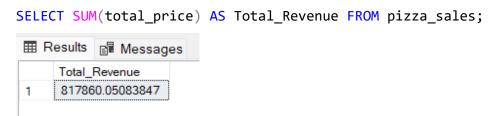
PIZZA SALES SQL QUERIES

A. KPI's

1. Total Revenue:



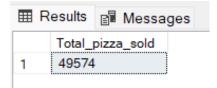
2. Average Order Value

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



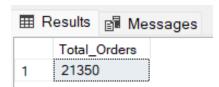
3. Total Pizzas Sold

SELECT SUM(quantity) AS Total_pizza_sold FROM pizza_sales



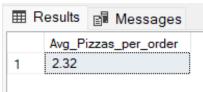
4. Total Orders

SELECT COUNT(DISTINCT order_id) AS Total_Orders FROM pizza_sales



5. 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 Avg_Pizzas_per_order
FROM pizza_sales
```



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_c	day	total_ord	ers
1	Saturd	ay	3158	
2	Wedne	sday	3024	
3	Monda	ıy	2794	
4	Sunda	у	2624	
5	Friday		3538	
6	Thursd	ay	3239	
7	Tuesda	ау	2973	

C. 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)
```

<u>Output</u>

⊞ Results			
	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	

D. % of Sales by Pizza Category

```
SELECT pizza_category, 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_category
```

<u>Output</u>

Results			
	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

E. % 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			
	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

F. Total Pizzas Sold by Pizza Category

```
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
```

Output

 	Results 📳 Mess	sages
	pizza_category	Total_Quantity_Sold
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

G. Top 5 Best Sellers by Total Pizzas Sold

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

Output

	pizza_name	Total_Pizza_Sold
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

H. Bottom 5 Best Sellers by Total Pizzas Sold

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

Output

■ Results			
pizza_name		Total_Pizza_Sold	
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	

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 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.