

SQL QUERIES FOR DATA **VALIDATION**

Suraj Kumar

ASPIRING DATA ANALYST

Download dataset(in csv Format) - [Pizza-Sales-Project/pizza_sales \(1\).csv at main · Sooraj1411/Pizza-Sales-Project \(github.com\)](#)

For this report, I have used a single file and the overview of the file is :

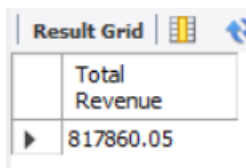
pizza_sales (1)	
order_date	
order_id	
order_time	
pizza_category	
pizza_id	
pizza_ingredients	
pizza_name	
pizza_name_id	
pizza_size	
quantity	
total_price	
unit_price	
Collapse ^	

Pizza Sales SQL Queries

1.KPI's:

- Total Revenue –

```
SELECT  
ROUND(SUM(Total_price), 2) AS 'Total Revenue'  
FROM pizza_sales;
```

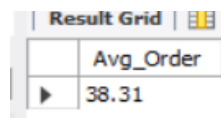


A screenshot of a SQL query result grid. The grid has two columns: the first column is empty, and the second column is labeled 'Total Revenue'. The value '817860.05' is displayed in the first row of the second column.

	Total Revenue
▶	817860.05

- Average Order Value –

```
SELECT  
ROUND(SUM(Total_price)/COUNT(DISTINCT order_id),  
2) AS 'Avg_Order'  
FROM pizza_sales;
```

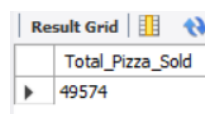


A screenshot of a SQL query result grid. The grid has two columns: the first column is empty, and the second column is labeled 'Avg_Order'. The value '38.31' is displayed in the first row of the second column.

	Avg_Order
▶	38.31

- Total Pizza Sold –

```
SELECT  
SUM(quantity) AS 'Total_Pizza_Sold'  
FROM pizza_sales;
```



A screenshot of a SQL query result grid. The grid has two columns: the first column is empty, and the second column is labeled 'Total_Pizza_Sold'. The value '49574' is displayed in the first row of the second column.

	Total_Pizza_Sold
▶	49574

- Total Orders –

```
SELECT  
COUNT(DISTINCT order_id) as Total_Orders
```

FROM pizza_sales;

Result Grid	
Total_Orders	
▶	21350

- Average Pizzas Per Order –

```
SELECT  
SUM(quantity)/COUNT(DISTINCT order_id) AS  
Avg_Pizza_Per_Order  
FROM  
pizza_sales;
```

Result Grid	
Avg_Pizza_Per_Order	
▶	2.3220

2.Daily Trends For Total Orders :

```
SELECT  
    DAYNAME(order_date) AS 'Day',  
    COUNT(DISTINCT order_id) AS 'Total_Orders'  
FROM  
    pizza_sales  
GROUP BY DAYNAME(order_date)  
ORDER BY Total_Orders DESC;
```

Result Grid		
Filter Row		
	Day	Total_Orders
▶	Friday	3538
	Thursday	3239
	Saturday	3158
	Wednesday	3024
	Tuesday	2973
	Monday	2794
	Sunday	2624

3.Monthly Trends For Total Orders :

```
SELECT  
    MONTHNAME(order_date) AS 'Month',  
    COUNT(DISTINCT order_id) AS 'Total_Orders'  
FROM pizza_sales
```

GROUP BY Month ORDER BY Total_Orders DESC;

	Month	Total_Orders
▶	July	1935
	May	1853
	January	1845
	August	1841
	March	1840
	April	1799
	November	1792
	June	1773
	February	1685
	December	1680
	September	1661
	October	1646

4.Percentage of Sales By Pizza Category :

```
SELECT
pizza_category AS category,
ROUND(SUM(total_price) * 100 / (SELECT
SUM(total_price)
FROM
pizza_sales),
2) AS '% Sales'
FROM
pizza_sales
GROUP BY pizza_category;
```

Result Grid		
	category	% Sales
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

5.Percentage of Sales By Pizza Size :

```
SELECT
pizza_size AS size,
ROUND(SUM(total_price) * 100 / (SELECT SUM(total_price)
FROM
pizza_sales), 2) AS '% Sales'
```

FROM pizza_sales GROUP BY size;

	size	% Sales
▶	M	30.49
	L	45.89
	S	21.77
	XL	1.72
	XXL	0.12

6.Total Pizza Sold in Pizza Category:

```
SELECT
pizza_category, SUM(quantity) as Total_Pizza_Sold
FROM
pizza_sales
GROUP BY pizza_category
```

	pizza_category	Total_Pizza_Sold
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050


7.Top 5 Pizzas By Revenue :

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

	pizza_name	Total_Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Spicy Italian Pizza	34831.25


8. Bottom 5 Pizzas By Revenue :

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

Result Grid  Filter Rows: <input type="text"/>		
	pizza_name	Total_Revenue
▶	The Brie Carre Pizza	11588.499999999999
	The Green Garden Pizza	13955.75
	The Spinach Supreme Pizza	15277.75
	The Mediterranean Pizza	15360.5
	The Spinach Pesto Pizza	15596

9. Top 5 Pizzas By Qty :

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

Result Grid  Filter Rows: <input type="text"/>		
	pizza_name	Total_Quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



10. Bottom 5 Pizzas By Qty :

```
SELECT  
pizza_name, SUM(quantity) as Total_Quantity  
FROM  
pizza_sales
```

```

GROUP BY pizza_name
ORDER BY Total_Quantity ASC
LIMIT 5;

```


Result Grid   Filter Rows: <input type="text"/>		
	pizza_name	Total_Quantity
▶	The Brie Carre Pizza	490
	The Mediterranean Pizza	934
	The Calabrese Pizza	937
	The Spinach Supreme Pizza	950
	The Soppressata Pizza	961

11.Top 5 Pizzas By 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;

```

Result Grid   Filter Rows: <input type="text"/>		
	pizza_name	Total_Orders
▶	The Classic Deluxe Pizza	2329
	The Hawaiian Pizza	2280
	The Pepperoni Pizza	2278
	The Barbecue Chicken Pizza	2273
	The Thai Chicken Pizza	2225

12.Bottom 5 Pizzas By Orders :

```

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

```


Result Grid  Filter Rows:		
	pizza_name	Total_Orders
▶	The Brie Carre Pizza	480
	The Mediterranean Pizza	912
	The Calabrese Pizza	918
	The Spinach Supreme Pizza	918
	The Chicken Pesto Pizza	938

NOTE :

If you want to apply the *pizza_category* or *pizza_size* filters to the above queries you can use *WHERE* clause. Follow some of below examples

```
SELECT
pizza_name, COUNT(DISTINCT order_id) AS Total_Orders
FROM
pizza_sales
WHERE
pizza_category = 'Classic'
GROUP BY pizza_name
ORDER BY Total_Orders ASC
LIMIT 5;
```

Remember to use *where* clause in subquery too if there is any in SQL query....us the *where* clause as same as it is in main query...

Key Findings Through Reports :

- Despite the Thai Chicken Pizza being at no. 5 in Top Pizzas by Qty and Orders, it still is the no. 1 Pizza in terms of generating Revenues.
- Despite the Classic Deluxe being at no. 1 in Top Pizzas by Qty and Orders, it still is the no. 4 Pizza in terms of generating Revenues.

- **Thursday, Friday, Saturday remains the best day of the week in terms of getting most orders.**
- **Orders of XXL size Pizzas are almost negligible. It is around 0.72% only with only around 1000 pizzas.**