INTRODUCTION-

Welcome to the Pizza Sales Analysis project! Here, we delve into Pizza sales data to extract insights and trends vital for optimizing sales strategies, understanding customer behaviour, and improving business operations. Through advanced SQL techniques, we'll analyze data to uncover correlations, identify emerging trends, and provide predictive insights. Our focus areas encompass data analysis, sales strategies, customer behaviour, SQL techniques, optimization, insights, trends, and business operations, aiming to empower stakeholders with actionable intelligence for informed decision-making and sustained growth.

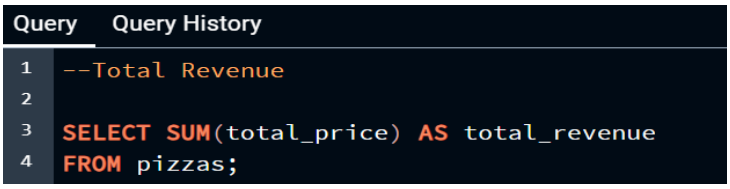
DATASET OVERVIEW-

The dataset used in this project consists of Approximately 22,000 rows of data, representing Pizza sales transactions. Before analysis, the dataset underwent preprocessing to handle missing values and ensure data quality, a crucial step in data analysis workflows. This preprocessing stage ensures the integrity and reliability of our findings, enabling us to draw accurate insights and make informed decisions based on the data.

PIZZAS SQL QUERIES

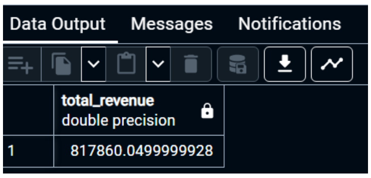
A. KPs

1. Total Revenue



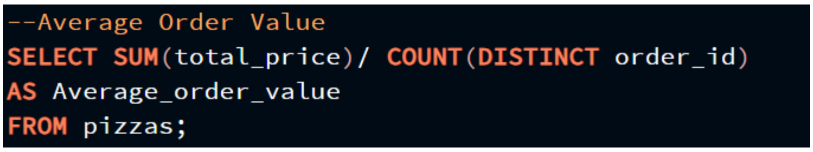
```sql
--Total Revenue

SELECT SUM(total_price) AS total_revenue
FROM pizzas;
```

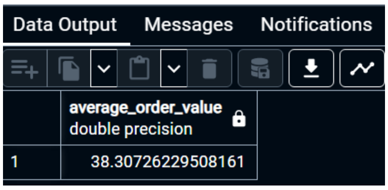


	total_revenue double precision
1	817860.0499999928

2. Average Order Value

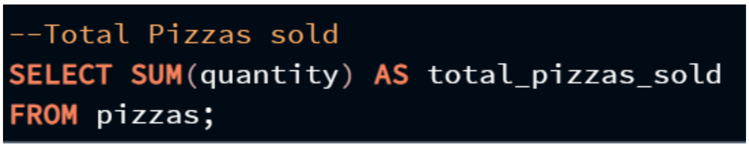


```sql
--Average Order Value
SELECT SUM(total_price)/ COUNT(DISTINCT order_id)
AS Average_order_value
FROM pizzas;
```

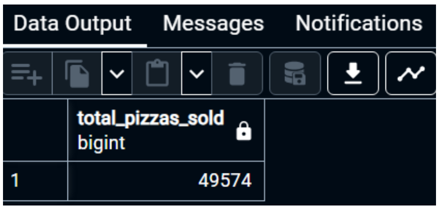


	average_order_value double precision
1	38.30726229508161

3. Total Pizzas Sold

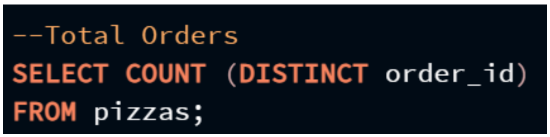


```sql
--Total Pizzas sold
SELECT SUM(quantity) AS total_pizzas_sold
FROM pizzas;
```

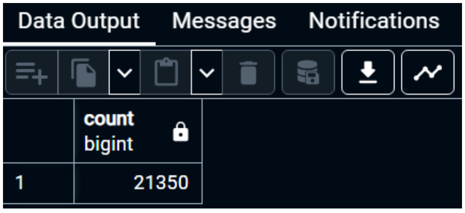


	total_pizzas_sold bigint
1	49574

4. Total Orders

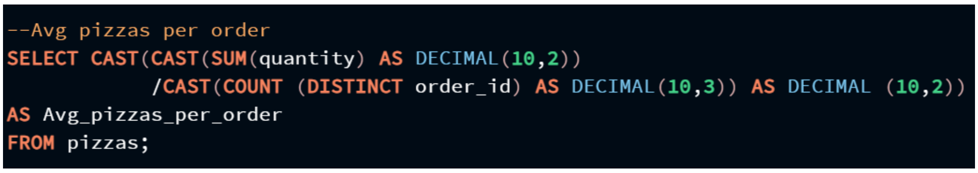


```sql
--Total Orders
SELECT COUNT (DISTINCT order_id)
FROM pizzas;
```

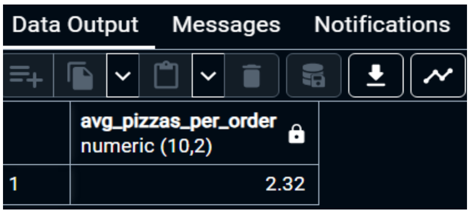


	count bigint
1	21350

5. Average Pizzas Per Order

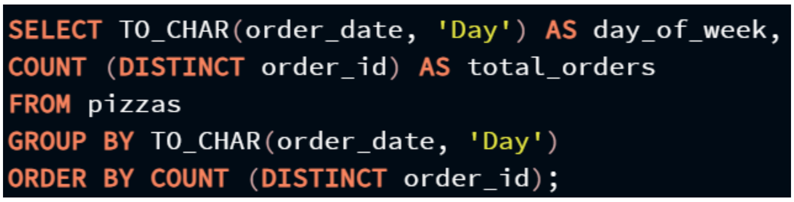


```sql
--Avg pizzas per order
SELECT CAST(CAST(SUM(quantity) AS DECIMAL(10,2))
            /CAST(COUNT (DISTINCT order_id) AS DECIMAL(10,3)) AS DECIMAL (10,2))
AS Avg_pizzas_per_order
FROM pizzas;
```

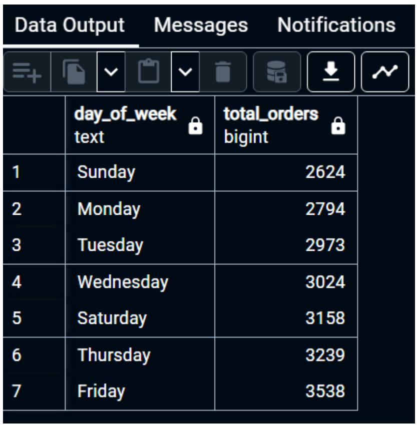


	avg_pizzas_per_order numeric (10,2)
1	2.32

6. Daily Trend

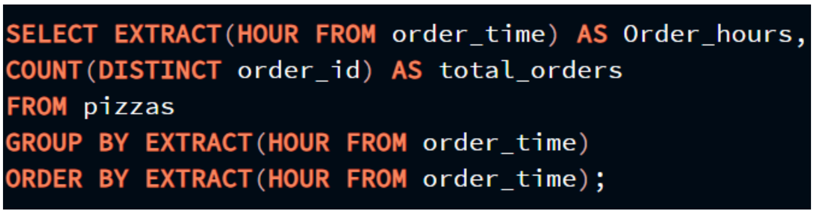


```sql
SELECT TO_CHAR(order_date, 'Day') AS day_of_week,
COUNT (DISTINCT order_id) AS total_orders
FROM pizzas
GROUP BY TO_CHAR(order_date, 'Day')
ORDER BY COUNT (DISTINCT order_id);
```

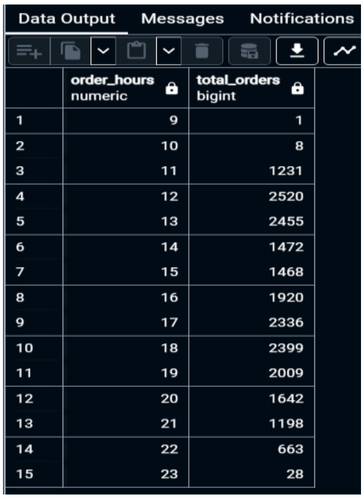


	day_of_week text	total_orders bigint
1	Sunday	2624
2	Monday	2794
3	Tuesday	2973
4	Wednesday	3024
5	Saturday	3158
6	Thursday	3239
7	Friday	3538

7. Hourly Trend

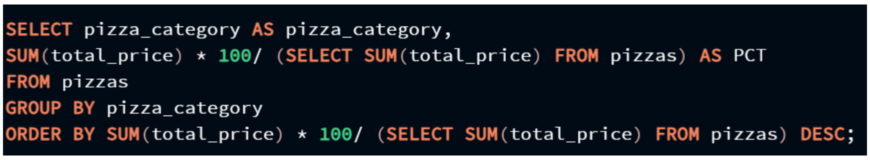


```sql
SELECT EXTRACT(HOUR FROM order_time) AS Order_hours,
COUNT(DISTINCT order_id) AS total_orders
FROM pizzas
GROUP BY EXTRACT(HOUR FROM order_time)
ORDER BY EXTRACT(HOUR FROM order_time);
```

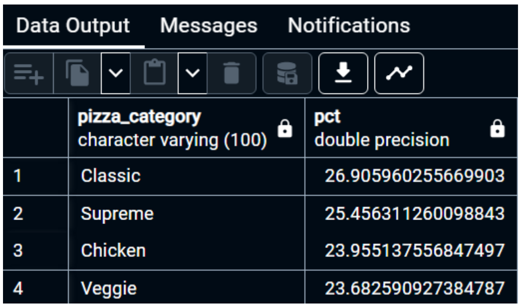


	order_hours numeric	total_orders bigint
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

8. Percentage Of Sales By Pizza Category

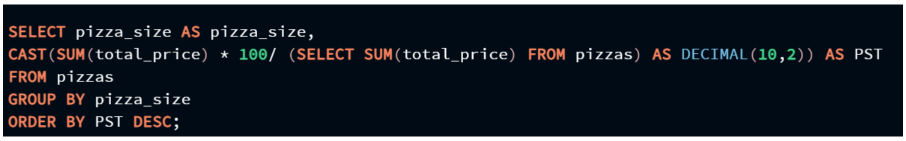


```sql
SELECT pizza_category AS pizza_category,
SUM(total_price) * 100/ (SELECT SUM(total_price) FROM pizzas) AS PCT
FROM pizzas
GROUP BY pizza_category
ORDER BY SUM(total_price) * 100/ (SELECT SUM(total_price) FROM pizzas) DESC;
```

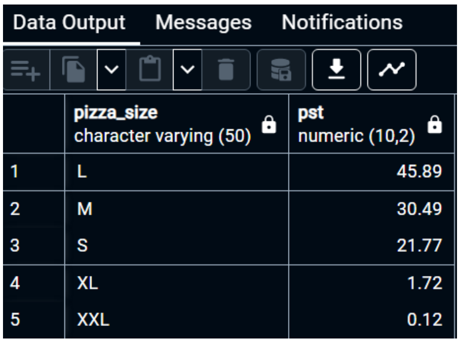


	pizza_category character varying (100)	pct double precision
1	Classic	26.905960255669903
2	Supreme	25.456311260098843
3	Chicken	23.955137556847497
4	Veggie	23.682590927384787

9. Percentage of Sales by Pizza Size

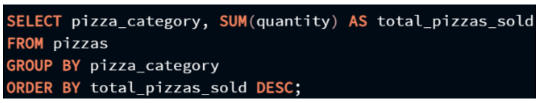


```
SELECT pizza_size AS pizza_size,
CAST(SUM(total_price) * 100/ (SELECT SUM(total_price) FROM pizzas) AS DECIMAL(10,2)) AS PST
FROM pizzas
GROUP BY pizza_size
ORDER BY PST DESC;
```

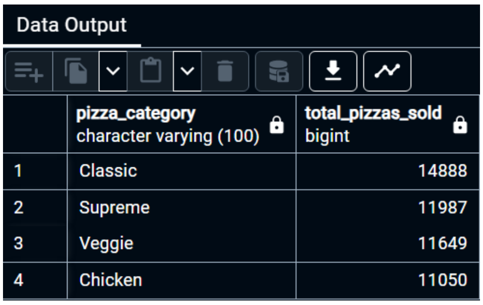


	pizza_size character varying (50)	pst numeric (10,2)
1	L	45.89
2	M	30.49
3	S	21.77
4	XL	1.72
5	XXL	0.12

10. Total Pizzas sold by Pizzas Category

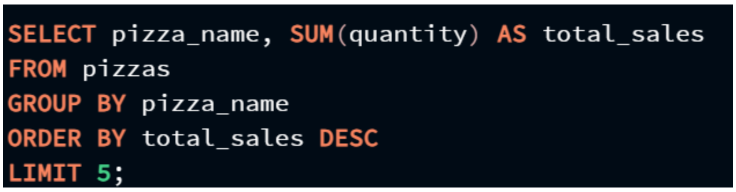


```
SELECT pizza_category, SUM(quantity) AS total_pizzas_sold
FROM pizzas
GROUP BY pizza_category
ORDER BY total_pizzas_sold DESC;
```

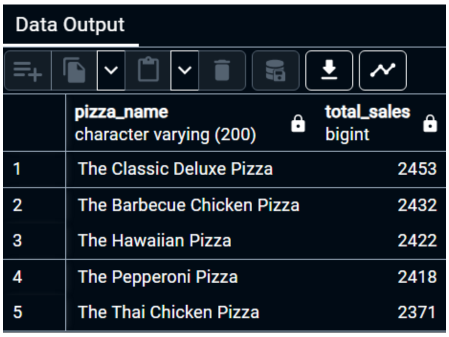


	pizza_category character varying (100)	total_pizzas_sold bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

11. Top 5 Best Sellers By Total Pizzas Sold



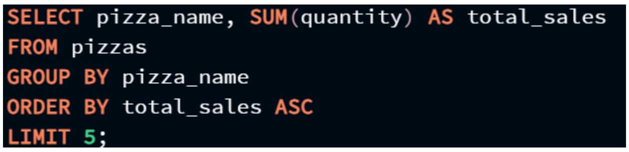
```
SELECT pizza_name, SUM(quantity) AS total_sales
FROM pizzas
GROUP BY pizza_name
ORDER BY total_sales DESC
LIMIT 5;
```



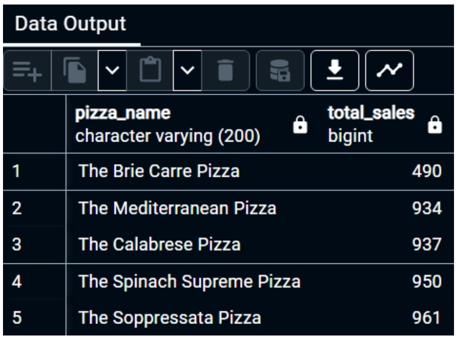
Data Output

	pizza_name character varying (200)	total_sales bigint
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

12. Bottom 5 Worst Sellers By Total Pizzas Sold



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



Data Output

	pizza_name character varying (200)	total_sales bigint
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

CONCLUSION-

In wrapping up this personal project, I've delved deep into pizza sales data, extracting invaluable insights into customer behaviours, market trends, and the intricate dynamics of the pizza industry. With a meticulous analysis under my belt, I've armed myself with actionable intelligence to refine sales strategies and seize emerging opportunities. This journey has not only broadened my understanding of the pizza market but also empowered me to make informed decisions in my pursuit of culinary entrepreneurship.