



LARANA PIZZA

LARANA PIZZA



SALES ANALYSIS



LARANA PIZZA

WELCOME TO LARANA PIZZA

Welcome to my SQL-powered Sales Analysis Dashboard, where I dive deep into pizza sales data to uncover key business insights. This project demonstrates the use of Structured Query Language (SQL) to extract, aggregate, and interpret transactional data from a relational database – turning raw sales data into meaningful strategic knowledge.

By Nitin Raj



VISSION & MISSION



VISSION

To empower data-driven decision-making in the food and retail industry by transforming raw transactional data into actionable insights using SQL — with a focus on clarity, accuracy, and business impact.

MISSION

My mission is to analyze and interpret pizza sales data using efficient SQL queries, uncovering key trends that support data-driven decisions. Through this project, I aim to demonstrate the practical impact of SQL in real-world business analytics.





● CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT

```
ROUND(SUM(order_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

order_details

JOIN

```
pizzas ON pizzas.pizza_id = order_details.pizza_id
```

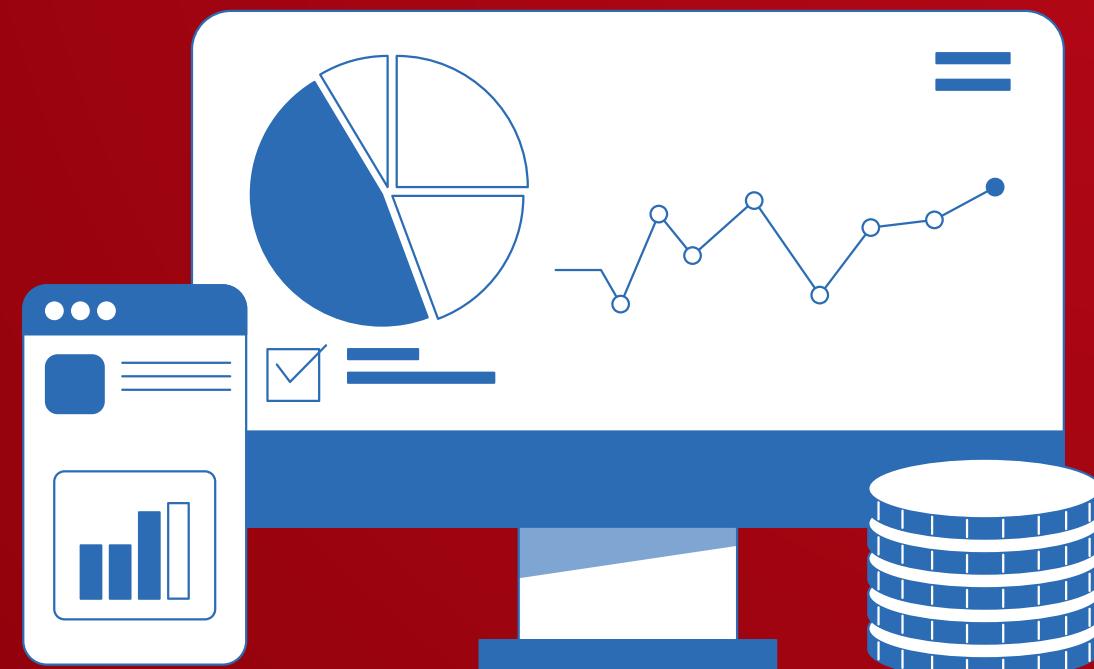
Result Grid	
	total_sales
▶	817860.05



LARANA PIZZA

● IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size, count(order_details.order_details_id) as order_count  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size order by order_count desc ;
```



	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



LARANA PIZZA

DETERMINE
THE
DISTRIBUTION
OF ORDERS
BY HOUR OF
THE DAY.



```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```



hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920



GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT  
    ROUND(AVG(quantity_avg_pizzas_perday), 0)  
FROM  
    (SELECT  
        orders.order_date,  
        SUM(order_details.quantity) AS quantity_avg_pizzas_perday  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid



Filter Rows:

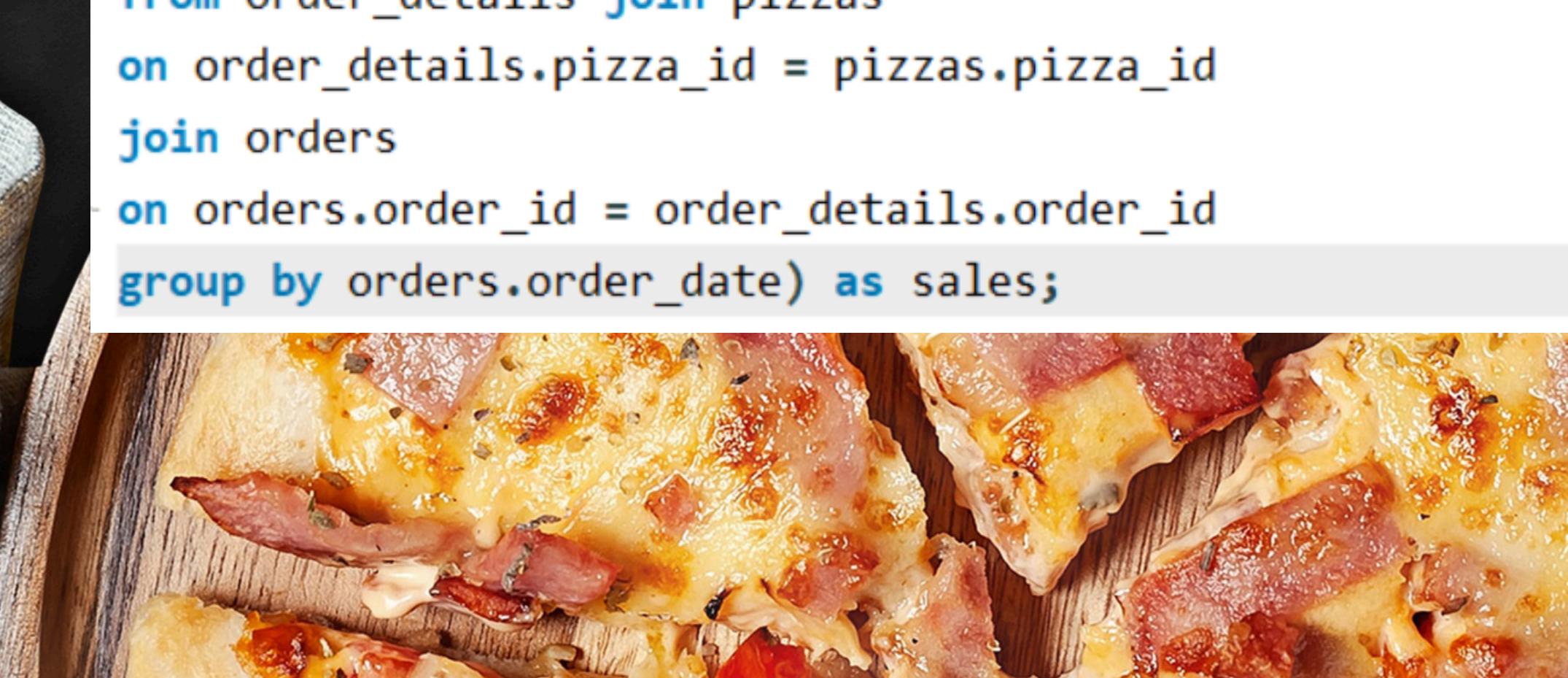
ROUND(AVG(quantity_avg_pizzas_perday), 0)

138

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
select order_date,  
sum(revenue) over(order by order_date) as cum_revenue  
from  
(select orders.order_date,  
sum(order_details.quantity * pizzas.price) as revenue  
from order_details join pizzas  
on order_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales;
```



	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4



FREE DELIVERY SERVICE



Through this project, I demonstrated how SQL can be effectively used to extract meaningful insights from sales data. From identifying best-selling products to analyzing revenue trends and customer behavior, each query served a real-world business objective. This analysis not only highlights the importance of structured data but also shows how powerful, yet simple, SQL can be in driving smarter decisions in the retail and food industry.





LARANA PIZZA

OUR CONTACT

📞 8539 *****

🌐 nitinraj*****@gmail.com

📍 123 Anywhere St., Any City, ST 12345



LARANA PIZZA

THANK YOU!

Like and
Comment

