Welcome to my project !!

PIZZA SALES ANALYSIS...

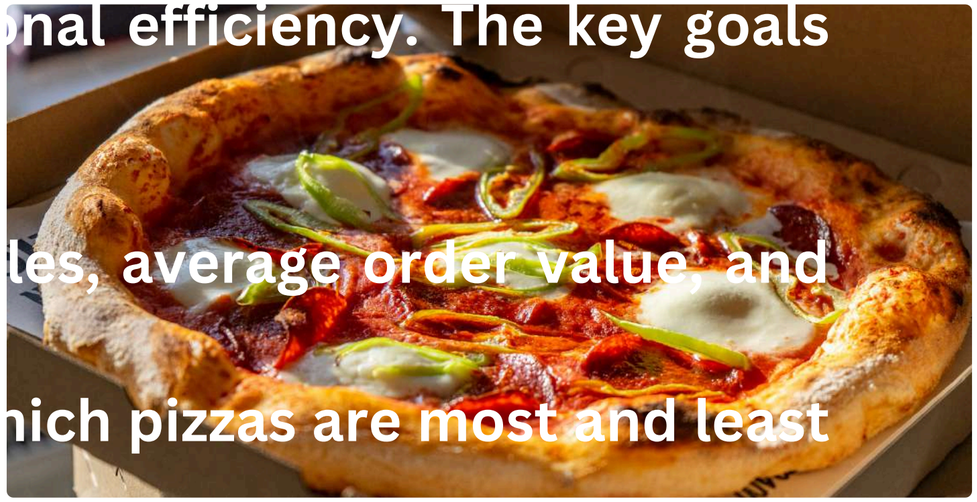
ORDER NOW

HELLO!!

My name is Raghvi Jain and I am here to uncovering valuable insights from pizza sales data. Our goal is to analyze sales trends, customer preferences, and operational efficiency to help businesses optimize their performance. Through data-driven decision-making, we aim to enhance revenue, improve inventory management, and refine marketing strategies.

OBJECTIVE

The objective of a *Pizza Sales Analysis* is to gain insights into sales performance, customer preferences, and operational efficiency. The key goals include:



1. *Revenue and Profit Analysis* – Assess total sales, average order value, and profit margins.
2. *Best-Selling & Least-Selling Items* – Identify which pizzas are most and least popular.
3. *Customer Preferences* – Understand size, crust, and topping trends.
4. *Seasonal Trends* – Analyze peak sales periods and seasonal variations.
5. *Order Patterns* – Examine order frequency, time-based trends, and high-demand hours.
6. *Store Performance* – Compare sales across different locations or channels.

TECHNOLOGY USED

Pizza sales analysis relies on a combination of data analytics, business intelligence, and machine learning technologies. Here are the key technologies used:

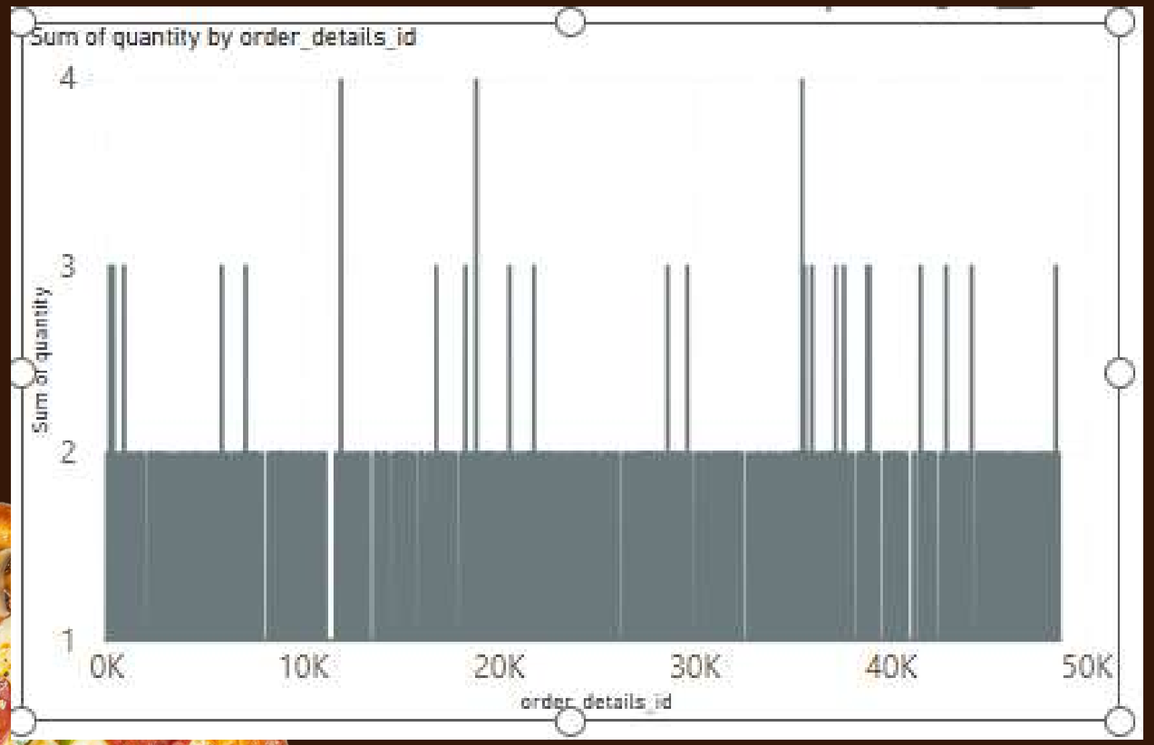
1. Data Collection & Storage
2.Data Processing.
3. Data Analytics & Visualization
- Business Intelligence Tools – Power BI, Tableau, Looker for dashboards and reporting.
- SQL & Excel – For querying and basic analysis.

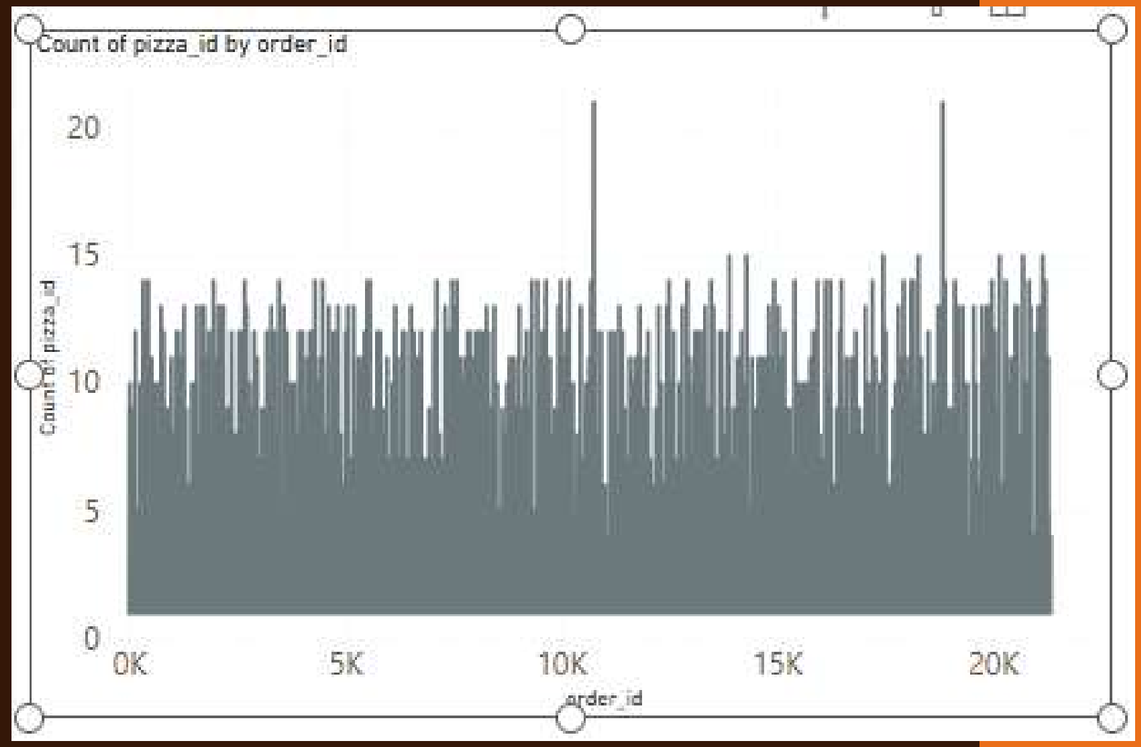
DATASET USED

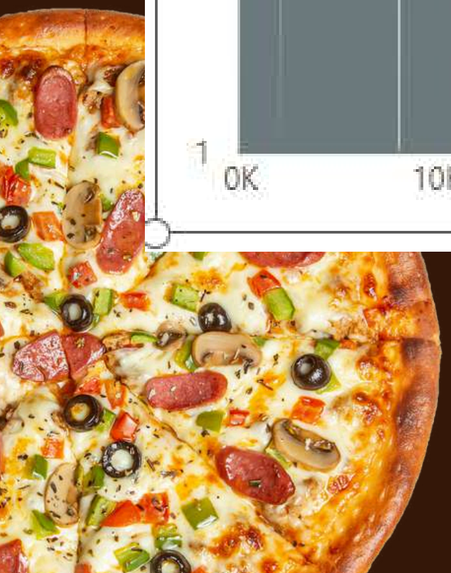
Here we have used Pizza Sales Dataset and and in this data set we have 4 tables :

1. order_details
2. orders
3. pizza_types
4. pizzas

Table 1-order_details

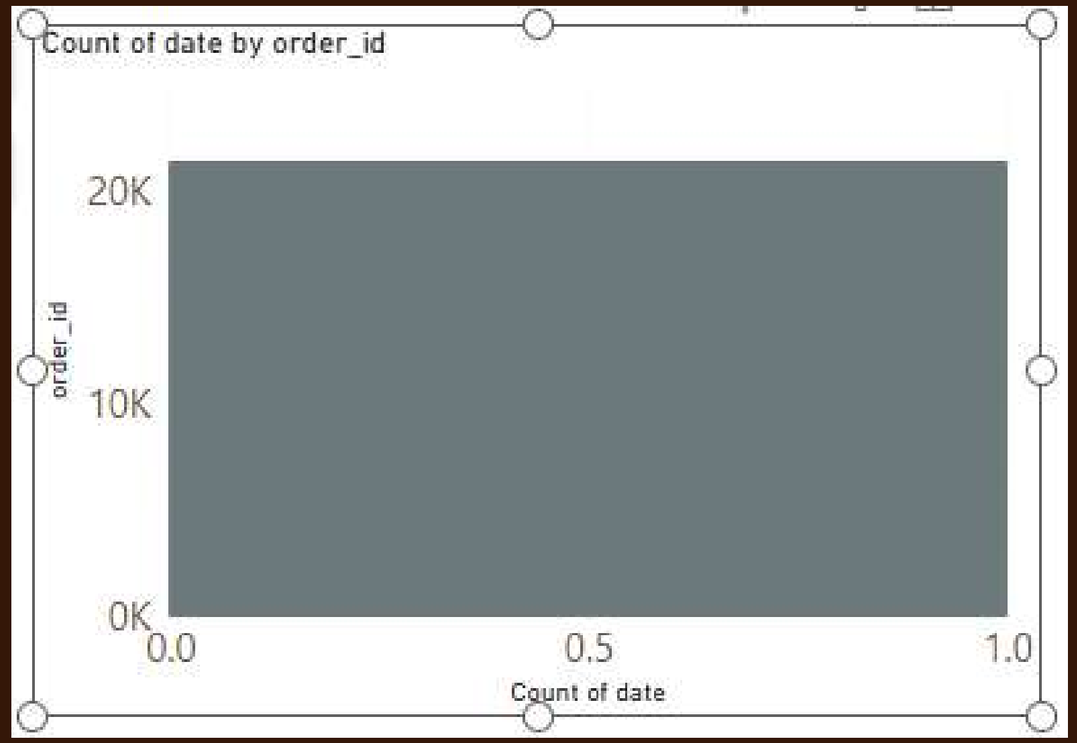






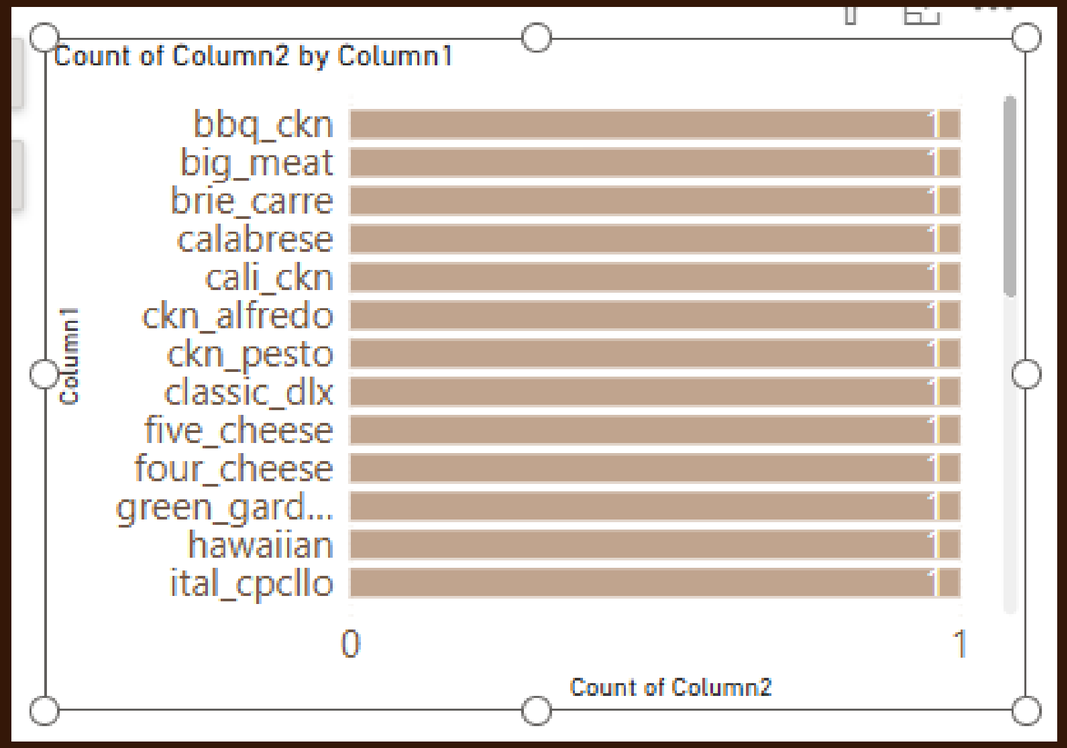
Line Graph

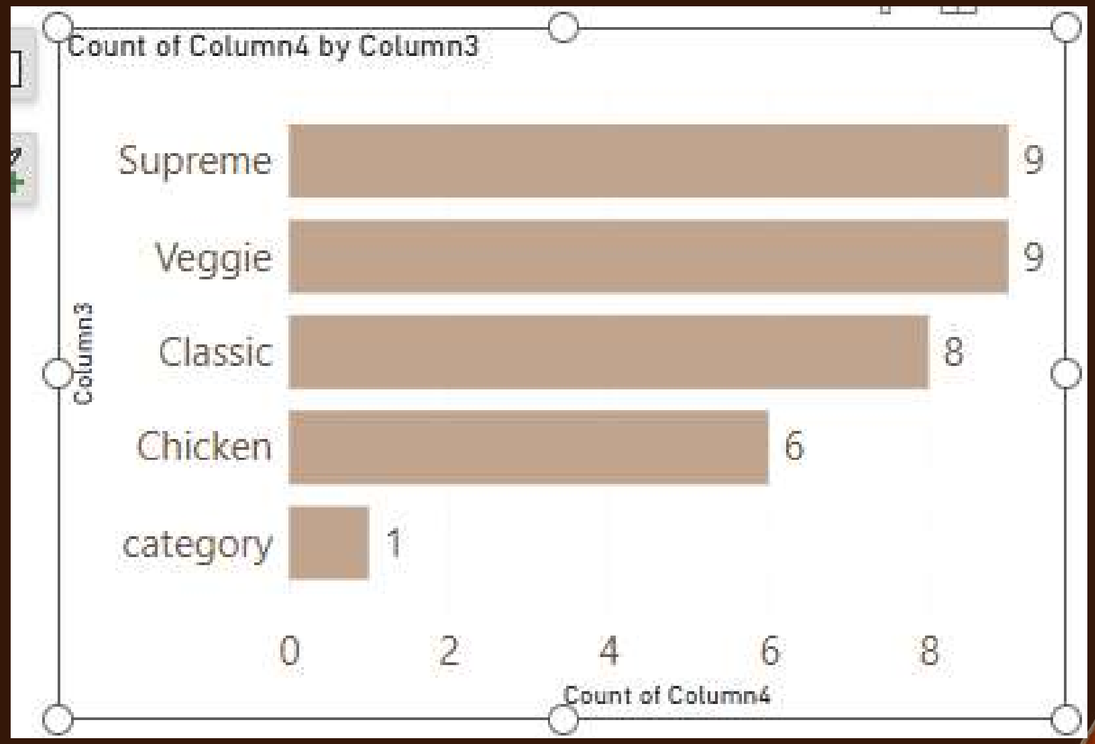
Table 2-orders



Line Graph

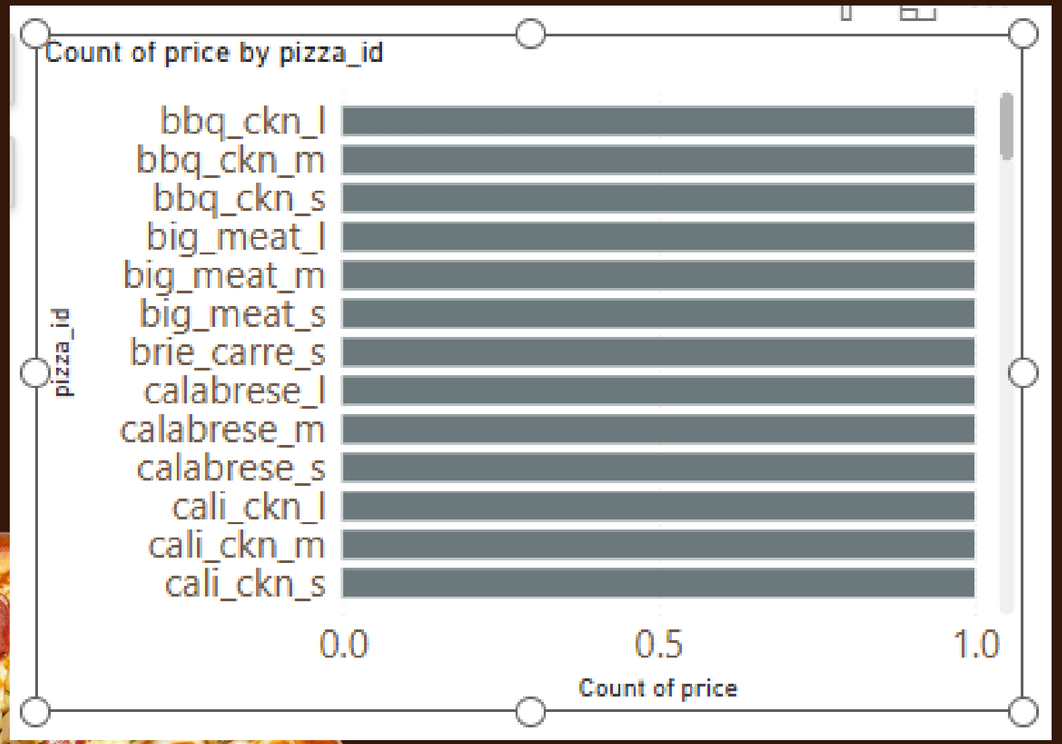
Table 3-pizza_types

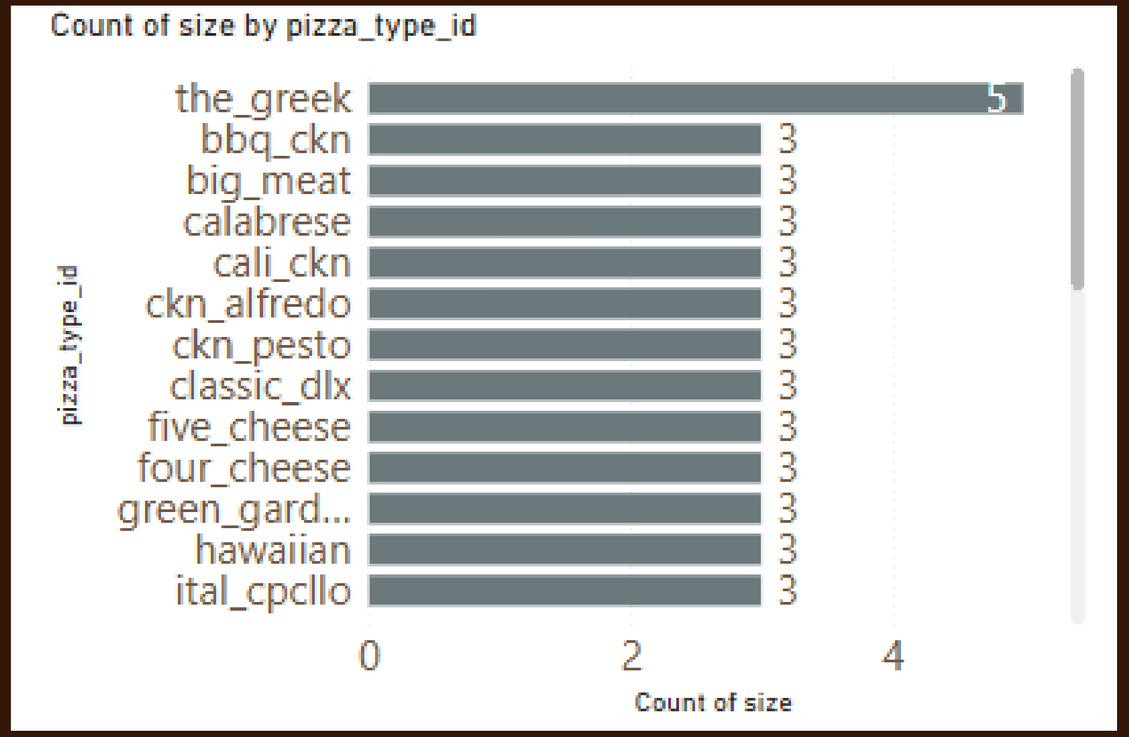


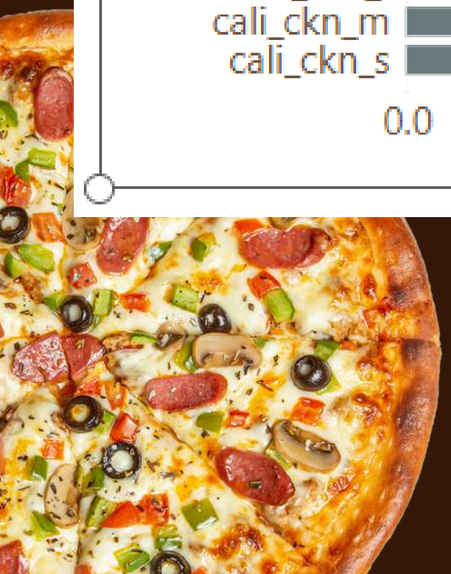


Line Graph

Table 4-pizzass







Line Graph

Now let’s perform Pizza Sales Analysis
by using SQL programming on various questions to get the details of how much pizzas have been sold and how much revenue has been generated .

-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	21350

-- CALCULATE TOTAL REVENUE GENERATED FROM PIZZA SALES.

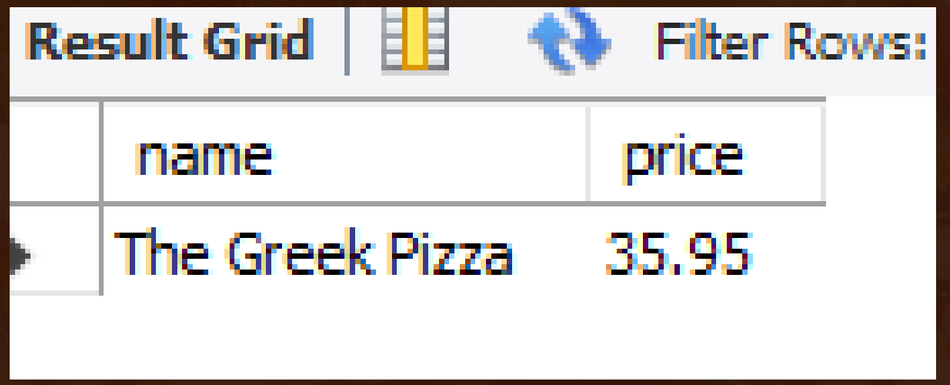
```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	total_sales
▶	817860.05

-- IDENTIFY HIGHEST-PRICED PIZZA.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1
```



-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
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;
```

Result Grid | Filter

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

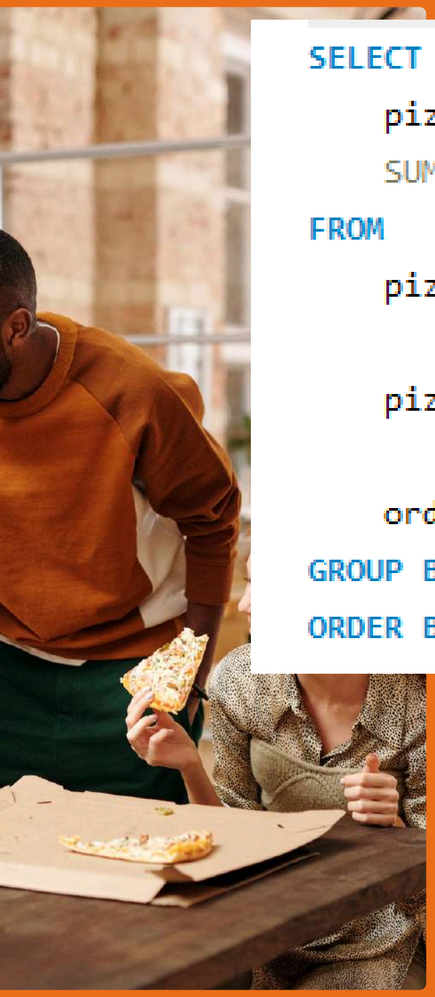
-- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITY.

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

-- JOIN THE NECCESARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter R

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	

Veggie

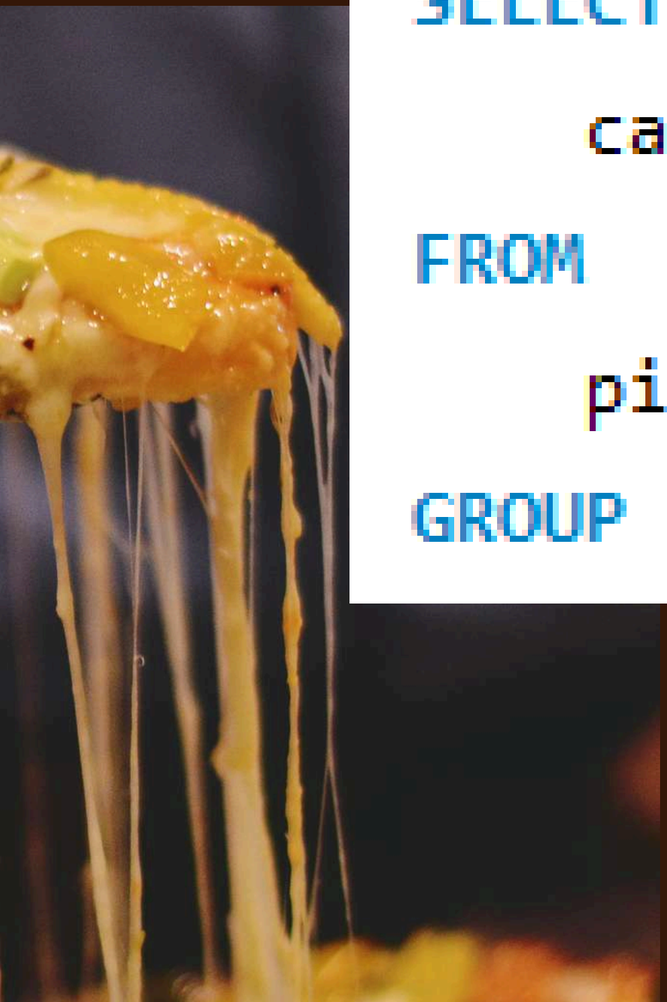
-- Determine the distribution of orders by hour of the day.

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

Result Grid | Filter Rows:

hour	count_order
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

-- Join the relevant tables to find category-wise distribution of the pizzas.

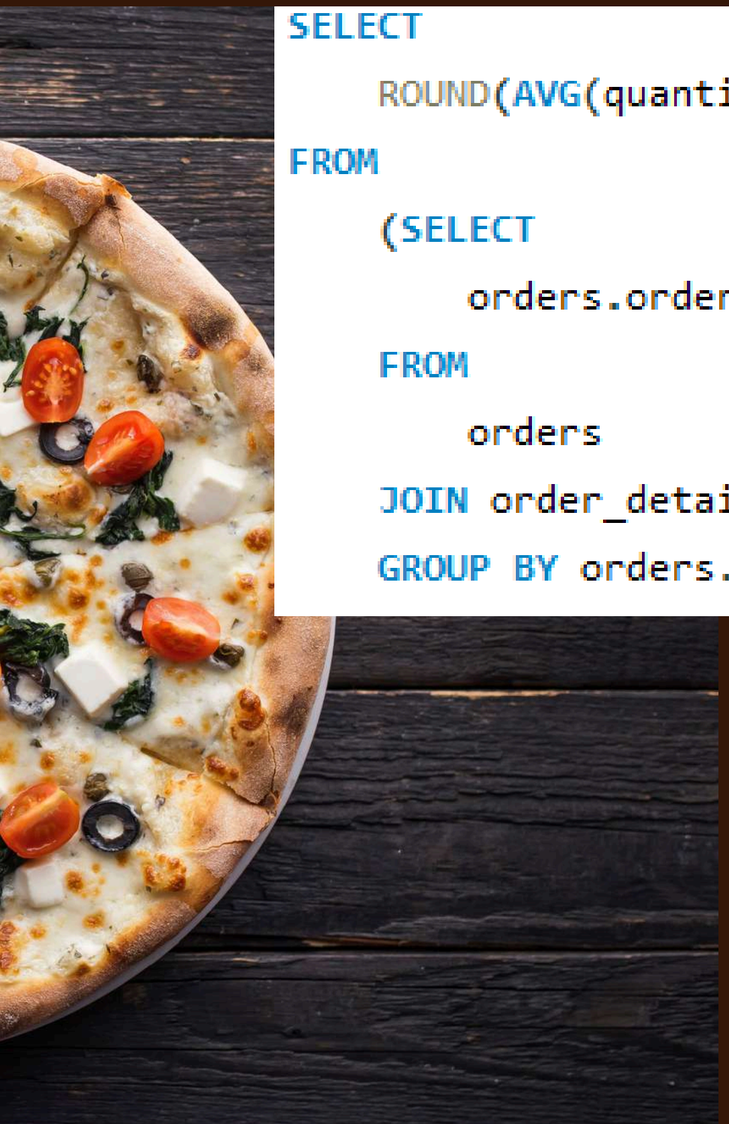


```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

-- Group the orders by date and calculate the avaerage number of pizzas ordered per day.



```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    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

	avg_pizza_ordered_per_day
▶	138

-- Determine the top 3 most ordered pizza types based on the revneue.

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

-- Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filter R

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

-- Analyze the cumlative revenue generated over time.

```sql
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 order_details.order_id=orders.order_id
group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

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

Pizza Sales Analysis
Presentation

THANK YOU FOR ATTENTION

See You Next

jainraghvi52@gmail.com