



SQl Project

Pizza Sales

By Hrithik

Background

This project is all about using what trends are involved in the buying and what the customers like. The idea is that we can make more money if people buy more because they know what's good for them. This data-driven insight will help us know our clients better to offer them these services.





Dataset Description

This analysis dataset came from Kaggle, an order of datasets and data science competitions platform. The information is extensive and covers everything about orders, types of pizzas, and in-depth details of the order.

Here are the main tables:

- **order_details:** Contains details of each order, including pizza ID and quantity.
- **orders:** Includes information about each order, such as order ID, date, and time.
- **pizza_types:** Provides information about the types of pizzas, including names and categories.
- **pizzas:** Contains details about individual pizzas, including their prices and sizes.

Analytical Approach

The analysis employs SQL queries to extract, transform, and analyze the data. Different SQL methods like joins, aggregations, and filtering help in coming up with major statistics and insights. This approach enables a comprehensive understanding of the sales dynamics and customer behavior.



1. Retrieve the total number of orders placed.

Input

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

Output

	total_orders
▶	21350

2. Calculate the total revenue from pizza sales.

Input

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

Output

	total_revenue
▶	817860.05

3. Identify the highest-priced pizza.

Input

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

Output

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

Input

```
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
LIMIT 1;
```

Output

	size	order_count
▶	L	18526

5. Top 5 most ordered pizzas with quantities.

Input

```
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;
```

Output

	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

6. Calculate the total quantity of each pizza category ordered.

Input

```
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;
```

Output

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

Input

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

Output

	hour	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

8. Find top 3 category-wise distribution of pizzas.

Input

```
SELECT
    category, COUNT(name) AS pizza_type_count
FROM
    pizza_types
GROUP BY category
ORDER BY pizza_type_count DESC
LIMIT 3;
```

Output

	category	pizza_type_count
▶	Supreme	9
	Veggie	9
	Classic	8

9. Group orders by date and find the average daily pizza orders.

Input

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

Output

	Avg_pizzas_ordered
▶	138

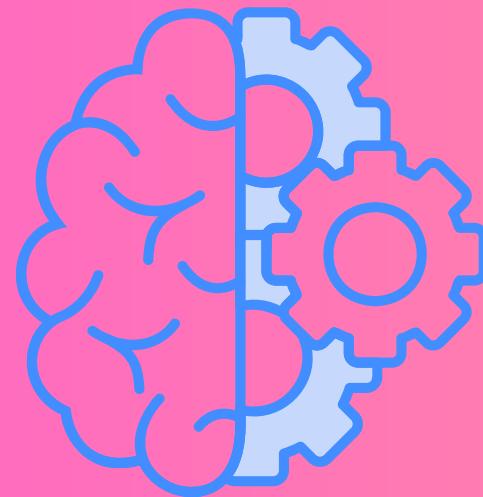
10. Identify the top 3 highest revenue-generating pizza types.

Input

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

Output

	name	total_revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



Summary of Key Findings

- **Total number of orders and revenue:** The analysis showed that there were **21,350** orders placed, generating **\$817,860.05** in total sales.
- **Most popular pizzas:** **The Greek Pizza** was the most expensive pizza available while **Large(L)** was the most ordered size.
- **Top 5 orders:** **The Classic Deluxe Pizza, The BBQ Chicken Pizza, The Hawaiian Pizza, The Pepperoni Pizza, and The Thai Chicken Pizza** were the top 5 most ordered pizzas; these contributed a lot to our sales.
- **When orders came in:** Most orders were made around **noon** which means this is our busiest time for making sales.



Final Thoughts

This analysis provides valuable insights into pizza sales patterns and customer preferences. By leveraging these findings, the business can make informed decisions to optimize operations and boost sales.

Thank you for your attention, and I welcome any questions or discussions.

Thank You!

