

SQL Case Study: Pizza Sales Analysis

By Priyanshi Garg

Introduction

This case study explores pizza order data using SQL in Google BigQuery, with the goal of answering key business questions related to sales, revenue, customer behavior, and product performance.

The dataset is structured into multiple tables, including orders, order details, pizzas, and pizza types.

Throughout this analysis, I applied SQL techniques ranging from basic queries to more complex joins and aggregations. The queries are grouped into Basic and Advanced levels to reflect the progression of my learning and ability to handle real-world data analysis tasks.

Dataset Source:

This project is based on a dataset shared in a YouTube tutorial by Data With Harshit:
https://youtu.be/zZpMvAedh_E?si=E_0gHuNzQFmIPLuP

Basic:

1. Retrieve the total number of orders placed.

```
SELECT count(order_id) FROM `priyanshi-garg.pizza.orders`
```

| Query results | | |
|-----------------|--------------------|---------|
| Job information | | Results |
| Row | total_no_of_orders | |
| 1 | 21350 | |

2. Calculate the total revenue generated from pizza sales.

```
SELECT round(sum(pizzas.price*order_detail.quantity)) as total_revenue_generated  
FROM `priyanshi-garg.pizza.order_details` as order_detail
```

```
left join `priyanshi-garg.pizza.pizzas` as pizzas
on order_detail.pizza_id=pizzas.pizza_id
```

| Query results | | |
|-----------------|-------------------------|---------|
| Job information | | Results |
| Row | total_revenue_generated | |
| 1 | 817860.0 | |

3. Identify the highest-priced pizza .

```
SELECT *
FROM `priyanshi-garg.pizza.pizzas`
order by price desc
limit 1
```

| Query results | | | | | | Save results | Open in |
|-----------------|---------------|---------------|-------|-------|-------------------|------------------------------|-------------------------|
| Job information | | Results | Chart | JSON | Execution details | Execution graph | |
| Row | pizza_id | pizza_type_id | size | price | | | |
| 1 | the_greek_xxl | the_greek | XXL | 35.95 | | | |

4. Identify the most common pizza size ordered.

```
select p.size,count(od.order_details_id) as orders_no,
from `priyanshi-garg.pizza.order_details` as od
full join `priyanshi-garg.pizza.pizzas` as p
on od.pizza_id=p.pizza_id
group by p.size
order by orders_no desc
Limit 1
```

| Query results | | |
|-----------------|------|-----------|
| Job information | | Results |
| Row | size | orders_no |
| 1 | L | 18526 |

5. List the top 5 most ordered pizza types(name) along with their quantities.

```

select sum(od.quantity) as quantity_ordered ,pt.name
from `priyanshi-garg.pizza.order_details` as od
left join `priyanshi-garg.pizza.pizzas` as p
on od.pizza_id=p.pizza_id
left join `priyanshi-garg.pizza.pizza_type` as pt
on p.pizza_type_id=pt.pizza_type_id
group by pt.name
order by quantity_ordered desc
limit 5

```

| Query results | | | |
|-----------------|------------------|----------------------------|-------|
| Job information | | Results | Chart |
| Row | quantity_ordered | name | JSON |
| 1 | 2453 | The Classic Deluxe Pizza | |
| 2 | 2432 | The Barbecue Chicken Pizza | |
| 3 | 2422 | The Hawaiian Pizza | |
| 4 | 2418 | The Pepperoni Pizza | |
| 5 | 2371 | The Thai Chicken Pizza | |

Advanced:

1. Join the necessary tables to find the total quantity of each pizza category ordered.

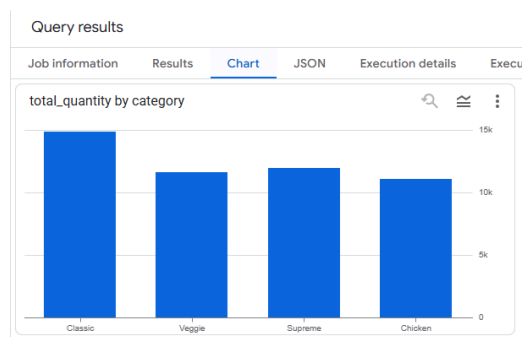
```

select pizza_type.category, sum(order_details.quantity) as total_quantity
from `priyanshi-garg.pizza.pizzas` as pizza
left join `priyanshi-garg.pizza.order_details` as order_details
on order_details.pizza_id=pizza.pizza_id
left join `priyanshi-garg.pizza.pizza_type` as pizza_type
on pizza_type.pizza_type_id=pizza.pizza_type_id
group by pizza_type.category

```

Query results

| Job information | Results | Chart | JSON | Exec |
|-----------------|----------|----------------|------|------|
| Row | category | total_quantity | | |
| 1 | Classic | 14888 | | |
| 2 | Veggie | 11649 | | |
| 3 | Supreme | 11987 | | |
| 4 | Chicken | 11050 | | |



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

```
select extract(hour FROM time) as hour, count(order_id) as distribution_of_orders
from `priyanshi-garg.pizza.orders`
group by hour
order by hour
```

Query results

| Job information | | Results | Chart | JS |
|-----------------|--------|-----------------------|-------|----|
| Row | hour ▼ | distribution_of_order | | |
| 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 | | |

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

```
select category, count(name) as distribution_of_pizzas
from `priyanshi-garg.pizza.pizza_type`
group by category
```

Query results

| Job information | | Results | Chart | JSON | Execution |
|-----------------|------------|--------------------------|-------|------|-----------|
| Row | category ▼ | distribution_of_pizzas ▼ | | | |
| 1 | Supreme | 9 | | | |
| 2 | Veggie | 9 | | | |
| 3 | Classic | 8 | | | |
| 4 | Chicken | 6 | | | |

4. Group the orders by date and calculate the average number of pizzas ordered per day.

```
select avg(no_of_orders)
```

From

```
(select o.date,sum(od.quantity) as no_of_orders
from `priyanshi-garg.pizza.orders` as o
join `priyanshi-garg.pizza.order_details` as od
on od.order_id=o.order_id
group by o.date)
```

Query results

| Job information | | Results |
|-----------------|-------------------|---------|
| Row | f0_ | |
| 1 | 138.4748603351... | |

5. Determine the top 3 most ordered pizza types based on revenue.

```
select pt.name, round(sum(od.quantity*p.price),2) as revenue,
from `priyanshi-garg.pizza.order_details` as od
join `priyanshi-garg.pizza.pizzas` as p
on od.pizza_id=p.pizza_id
join`priyanshi-garg.pizza.pizza_type` as pt
on pt.pizza_type_id=p.pizza_type_id
group by pt.name
order by revenue desc
limit 3
```

Query results

| Job information | | Results | Chart | JSON | Ex |
|-----------------|------------------------------|----------|-------|------|----|
| Row | name | revenue | | | |
| 1 | The Thai Chicken Pizza | 43434.25 | | | |
| 2 | The Barbecue Chicken Pizza | 42768.0 | | | |
| 3 | The California Chicken Pizza | 41409.5 | | | |

Conclusion & Summary

This project helped reinforce my understanding of SQL by applying it to a realistic business scenario.

I learned how to:

- Join multiple tables effectively
- Calculate key metrics like revenue, order counts, and product popularity
- Use functions like GROUP BY, JOIN, EXTRACT, and aggregation techniques to extract insights

o

These queries could help a pizza business make decisions about best-selling products, ideal inventory levels, and customer order patterns throughout the day.

I look forward to expanding this analysis further by exploring advanced metrics and visualizing the results using BI tools like Tableau or Google Data Studio.