

Hells

SQL PROJECT ON PIZZA SALES ANALYSIS

DATA INSIGHTS AND QUERY IMPLEMENTATION

PIJUSH PAL

5TH MARCH, 2025



INTRODUCTION

PROJECT OBJECTIVE

The main goal is to analyze pizza sales data using SQL queries to extract meaningful insights.

SCOPE OF ANALYSIS

- Order trends (total orders, peak hours)
- Revenue generation (total revenue, top-selling pizzas)
- Customer preferences (most ordered pizza types, sizes, and categories)
- Business insights for improving sales strategy





Dataset Overview

Dataset Structure

- **Orders** – Contains order IDs, timestamps, and order details.
- **Order Details** – Links each order to specific pizzas, including pizza_id, quantity, and price.
- **Pizzas** – Includes pizza_id, pizza_name, size, price, and category.
- **Pizza Types** – Describes different types of pizzas, such as their ingredients and classification.

Relationships Between Tables

- Orders and Order Details are linked via order_id.
- Order Details and Pizzas are linked via pizza_id.
- Pizzas and Pizza Types share a common category.
- These relationships help in performing JOIN queries for insights.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT  
    COUNT(*) AS total_order  
FROM  
    orders;
```

	total_order
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

SELECT

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

FROM

```
order_details
```

JOIN

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

	total_revenue
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
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;
```

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT

pizzas.size, COUNT(pizzas.size) **AS** count_pizza

FROM

pizzas

JOIN

order_details **ON** pizzas.pizza_id = order_details.pizza_id

GROUP BY pizzas.size

ORDER BY count_pizza **DESC;**

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

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS sum_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 sum_quantity
LIMIT 5;
```

	name	sum_quantity
▶	The Brie Carre Pizza	490
	The Mediterranean Pizza	934
	The Calabrese Pizza	937
	The Spinach Supreme Pizza	950
	The Soppressata Pizza	961

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

SELECT

```
    pizza_types.category,  
    SUM(order_details.quantity) AS quantity
```

FROM

```
    pizza_types
```

JOIN

```
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

JOIN

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

GROUP BY pizza_types.category

ORDER BY quantity;

	category	quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

	order_hour	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

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

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

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

SELECT

round(AVG(quantity),2)

FROM

(SELECT

orders.oredr_date, SUM(order_details.quantity) AS quantity

FROM

orders

JOIN order_details ON orders.order_id = order_details.order_id

GROUP BY orders.oredr_date) AS order_quantity;

	round(AVG(quantity),2)
▶	138.47

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

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

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

SELECT

```
    orders.oredr_date,  
    round(SUM(order_details.quantity* pizzas.price),2) AS revenue,  
    round(sum(SUM(order_details.quantity* pizzas.price))  
    over (order by orders.oredr_date),2)  
    as cumulative_revenue  
FROM  
    orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    join pizzas on pizzas.pizza_id = order_details.pizza_id  
GROUP BY orders.oredr_date;
```

	oredr_date	revenue	cumulative_revenue
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
select
    category, name, revenue, rank_revenue
from
    (select
        category, name, revenue,
        rank() over(partition by category order by revenue desc) as rank_revenue
    from
        (select pizza_types.category, pizza_types.name,
            sum(pizzas.price * order_details.quantity) as revenue
        from pizza_types
        join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id
        join order_details on pizzas.pizza_id = order_details.pizza_id
        group by pizza_types.category, pizza_types.name) as a_table) as b_table
where rank_revenue<=3;
```

	category	name	revenue	rank_revenue
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3

PIJUSH PAL

THANK YOU!

