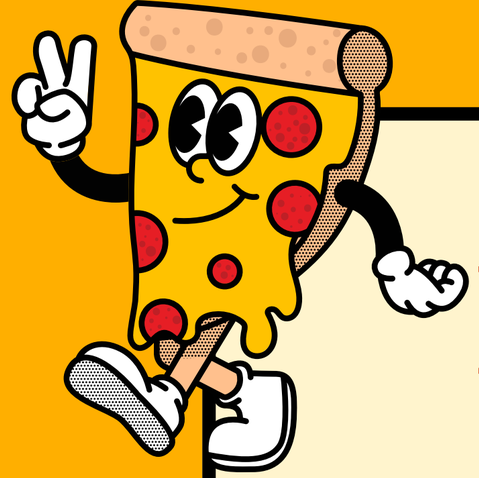
Pizza Sales

ANALYSIS USING SQL

Siddharth Pandey



Introduction

In this project, we embarked on an extensive data analysis journey using Structured Query Language (SQL) to extract, manipulate, and derive insights from multiple data files. Our objective was to answer a series of key questions that would help us understand patterns, trends, and relationships within the data. The data files we used spanned various domains, each contributing unique and valuable information to our analysis.

Table

Order Table

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31

Pizzas Table

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75

Order Details Table

order_detail_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1

Pizza Types Table

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chick
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichо
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Or
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomato

Questions

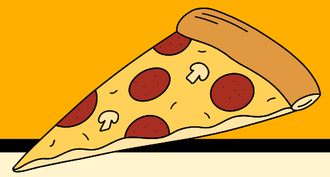
Q1. Retrieve the total number of orders placed.

Query

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

Output

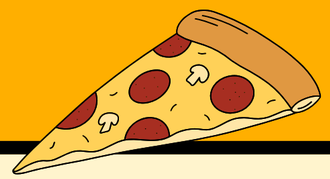
total_orders
21350



Q2. Calculate the total revenue generated from pizza sales.(in 2 decimal value)

Query

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



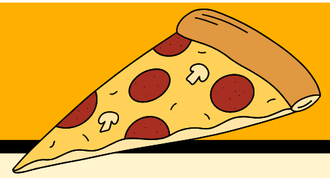
Output

total_revenue
817860.05

Q3. Identify the highest-priced pizza

Query

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



Output

name	price
The Greek Pizza	35.95

Q4. Identify the most common pizza size ordered

Query

```
SELECT
    p.size, COUNT(order_detail_id) AS order_count
FROM
    order_details AS o
        JOIN
    pizzas AS p ON o.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY order_count DESC
LIMIT 1;
```

Output

size	order_count
L	18526

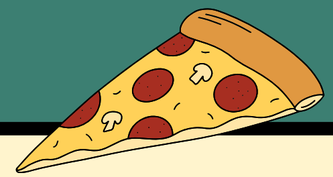
Q5. List the top 5 most ordered pizza types along with their quantities

Query

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

Output

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



Q6. Find the total quantity of each pizza category ordered

Query

```
SELECT
    pt.category, SUM(o.quantity) as total_quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id
GROUP BY pt.category
```

Output

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Q7. Determine the distribution of orders by hour of the day

Query

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

Output

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

Q8. Find the category-wise distribution of pizzas

Query

```
SELECT
    category, COUNT(category) AS total_count
FROM
    pizza_types
GROUP BY category
```

Output

category	total_count
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q9. Calculate the average number of pizzas ordered per day

Query

```
SELECT
    ROUND(AVG(total_quantity), 2) AS avg_quantity_perday
FROM
    (SELECT
        od.order_date AS date, SUM(o.quantity) AS total_quantity
    FROM
        orders AS od
    JOIN order_details AS o ON od.order_id = o.order_id
    GROUP BY od.order_date) AS order_per_day
```

Output

avg_quantity_perday
138.47

Q10. Determine the top 3 most ordered pizza types based on revenue

Query

```sql
SELECT
    pt.name, ROUND(SUM(p.price * o.quantity), 2) AS revenue
FROM
    pizzas AS p
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types AS pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3
```

Output

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

Q11. Calculate the percentage contribution of each pizza type to total revenue

Query

```
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) * 100 / (SELECT
                    SUM(o.quantity * p.price)
                FROM
                    order_details AS o
                        JOIN
                    pizzas AS p ON o.pizza_id = p.pizza_id), 2) AS percenatage_share
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
```

Output

category	percenatage_share
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Q12. Analyze the cumulative revenue generated over time

Query

```
SELECT order_date, round(sum(revenue)
OVER (ORDER BY order_date),2) AS cum_revenue
FROM (SELECT o.order_date, round(sum(p.price*od.quantity),2)
AS revenue
FROM orders AS o
JOIN order_details AS od ON o.order_id=od.order_id
JOIN pizzas AS p ON p.pizza_id=od.pizza_id
GROUP BY o.order_date) AS sales
```

Output

order_date	cum_revenue
2015-01-01	2713.85
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

Conclusion

By conducting the analysis using SQL we came up with various solution of question which was important to answer and which will help the business reshape and structure as per the analysis.

Thank You