Home About Contact

SQL DATA ANALYSIS PROJECT - PIZZA SALES DATASET

- WHERE EVERY SLICE TELLS A STORY

Home About Contact

Business insights generated using SQL by analyzing pizza sales data,
including orders, revenue, category performance, and sales trends.

Tools Used:

- SQL (MySQL) – Data analysis
- GitHub – Project sharing & version control
- PowerPoint – Insight presentation

Name: Apurva Chavan
Role: Aspiring Data Analyst
Dataset: Pizza Sales Database

SHODWE
Pizza Resto

Home About Contact

This project analyzes a Pizza Sales database using SQL to understand day-to-day business performance.

- Data includes customer orders, pizzas sold, prices, and categories
- Multiple tables were joined to analyze sales patterns and revenue
- SQL queries were used to answer real business questions
- Analysis focuses on product performance and revenue trends
- Raw data is transformed into clear, actionable insights

Retrieve the total no of orders placed

```sql
-- Retrieve the total no of orders placed
select count(order_id) as total_order from orders;
```

	total_order
▶	21350

Calculate the total revenue generated from the pizza sales.

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

total_sales
779752.3

Identify the 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;
```

	name	price
▶	The Greek Pizza	35.95

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;
```

size	order_count
L	17664
M	14668
S	13488
XL	517
XXL	27

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

```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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

name	quantity
The Classic Deluxe Pizza	2332
The Barbecue Chicken Pizza	2329
The Pepperoni Pizza	2311
The Hawaiian Pizza	2309
The California Chicken Pizza	2248

Join the necessary 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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

category	quantity
Classic	14177
Supreme	11459
Veggie	11099
Chicken	10533

Determine the distribution of orders by hour of the day.

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

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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.

```sql
SELECT ROUND(AVG(quantity), 0) as average_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;
```

	average_pizza_ordered_per_day
▶	139

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

```sql
SELECT pizza_types.name,
       SUM(order_details.quantity * pizzas.price) AS 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 revenue DESC
LIMIT 3;
```

name	revenue
The Thai Chicken Pizza	41129.25
The Barbecue Chicken Pizza	40918.75
The California Chicken Pizza	39274

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

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

	category	revenue
▶	Classic	26.87
	Supreme	25.52
	Chicken	23.94
	Veggie	23.67

Analyze the cumulative 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 orders.order_id = order_details.order_id
    GROUP BY orders.order_date
) AS sales;
```

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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006

order_date	cum_revenue
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001
2015-01-27	61043.85000000001
2015-01-28	63059.85000000001
2015-01-29	65105.150000000016
2015-01-30	67375.45000000001
2015-01-31	69793.30000000002
2015-02-01	72982.50000000001
2015-02-02	75311.10000000002
2015-02-03	77925.90000000002
2015-02-04	80159.80000000002
2015-02-05	82375.60000000002

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
SELECT name, revenue
FROM (
    SELECT
        category,
        name,
        revenue,
        RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn
    FROM (
        SELECT
            pizza_types.category,
            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 order_details.pizza_id = pizzas.pizza_id
        GROUP BY pizza_types.category, pizza_types.name
    ) AS a
) AS b
WHERE rn <= 3;
```

name	revenue
The Thai Chicken Pizza	41129.25
The Barbecue Chicken Pizza	40918.75
The California Chicken Pizza	39274
The Classic Deluxe Pizza	36284.5
The Hawaiian Pizza	30756.25
The Pepperoni Pizza	28816
The Spicy Italian Pizza	33408.25
The Italian Supreme Pizza	32023.25
The Sicilian Pizza	29490.5
The Four Cheese Pizza	30811.600000000606
The Mexicana Pizza	25732.5
The Five Cheese Pizza	25123