📊 Pizza Sales SQL Project – Detailed Analysis & Findings

This section presents a structured explanation of the **SQL analyses performed**, the **queries used**, and the **business insights derived** from the pizza sales dataset. The analysis progresses from **basic exploration** to **advanced analytical techniques**, reflecting a real-world analytics workflow.

1️⃣ Dataset Exploration (Initial Queries)

Queries Used

```
SELECT * FROM pizzas;
SELECT * FROM pizza_types;
SELECT * FROM orders;
SELECT * FROM order_details;
```

Purpose

- Understand the structure of each table
- Verify columns, data types, and table relationships

Findings

- The dataset consists of **four relational tables** connected through primary and foreign keys.
- `orders` stores order date and time.
- `order_details` stores pizza quantity per order.
- `pizzas` stores pizza size and price.
- `pizza_types` stores pizza names, categories, and ingredients.

✅ **Conclusion:** The dataset is well-structured and suitable for relational SQL analysis.

2️⃣ Total Number of Orders Placed

Query

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

Output

- **Total Orders:** `21,350`

Findings

- The business processed **21,350 orders**, indicating strong customer demand.
- This metric forms the basis for revenue, average order value, and demand trend analysis.

📌 **Business Insight:**
High order volume reflects strong customer reach and consistent sales activity.

3️⃣ Total Revenue Generated from Pizza Sales

Query

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

Output

- **Total Revenue:** `$817,860.05`

Findings

- The business generated approximately **$817K in total revenue**.
- Revenue is calculated using quantity sold and unit price.

📌 **Business Insight:**
This metric represents overall business performance and profitability.

4️⃣ Highest-Priced Pizza

Query

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

Output

- **The Greek Pizza – $35.95**

Findings

- The Greek Pizza is the most expensive item on the menu.
- Premium pricing reflects specialty ingredients.

📌 **Business Insight:**
High-priced pizzas can improve margins but may have lower order frequency.

5️⃣ Most Common Pizza Size Ordered

Query

```
SELECT pizzas.size,
COUNT(order_details.order_details_id) AS Sales_by_Size
FROM pizzas
JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Sales_by_Size DESC;
```

Output Summary

Size	Orders
L	18,526
M	15,385
S	14,137
XL	544
XXL	28

Findings

- **Large pizzas** are the most frequently ordered.
- Medium and Small sizes also show strong demand.
- XL and XXL pizzas have very low demand.

📌 **Business Insight:**
Inventory and promotions should focus on **Large and Medium sizes** to maximize efficiency.

6️⃣ Top 5 Most Ordered Pizza Types

Query

```
SELECT pizza_types.name,
SUM(order_details.quantity) AS Sales_by_Pizza
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 Sales_by_Pizza DESC
LIMIT 5;
```

Output

Pizza Name	Quantity Sold
Classic Deluxe Pizza	2,453
Barbecue Chicken Pizza	2,432
Hawaiian Pizza	2,422
Pepperoni Pizza	2,418
Thai Chicken Pizza	2,371

Findings

- A small number of pizzas account for a large share of total sales.
- Classic and Chicken-based pizzas dominate customer preference.

📌 **Business Insight:**
These pizzas should be prioritized in promotions, combos, and inventory planning.

7️⃣ Category-wise Total Quantity Ordered

Query

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

Output Summary

Category	Quantity
Classic	14,888
Supreme	11,987

Category	Quantity
Veggie	11,649
Chicken	11,050

Findings

- Classic pizzas are the most ordered category.
- Demand is well distributed across all categories.

📌 **Business Insight:**
Classic pizzas should remain the core offering, while targeted promotions can boost other categories.

8️⃣ Order Distribution by Hour of the Day

Query

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

Peak Hours

Hour	Orders
12	2,520
13	2,455
18	2,399
17	2,336
19	2,009

Findings

- Peak demand occurs during **lunch (12–1 PM)** and **dinner (6–7 PM)**.
- Very low demand during early mornings and late nights.

📌 **Business Insight:**
Staffing, preparation, and delivery capacity should align with peak hours.

9️⃣ Category-wise Pizza Variety Count

Query

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

Output

Category	Pizza Types
Classic	8
Supreme	9
Veggie	9
Chicken	6

Findings

- Supreme and Veggie categories offer the widest variety.
- Chicken category has fewer options.

📌 **Business Insight:**
Expanding chicken pizza varieties may increase category revenue.

🔟 Average Number of Orders Per Day

Query

```
SELECT AVG(num_of_orders)
FROM (
    SELECT date, COUNT(order_id) AS num_of_orders
    FROM orders
    GROUP BY date
) AS avgorders;
```

Output

- **≈ 59.64 orders/day**

📌 **Business Insight:**
Supports daily staffing and inventory planning.

1️⃣1️⃣ Average Number of Pizzas Ordered Per Day

Query

```
SELECT AVG(numpizzaordered)
FROM (
    SELECT orders.date,
           SUM(order_details.quantity) AS numpizzaordered
    FROM orders
    JOIN order_details
        ON order_details.order_id = orders.order_id
    GROUP BY orders.date
) AS numofpizzaordered;
```

Output

- **≈ 138.47 pizzas/day**

📌 **Business Insight:**
Customers often place multi-pizza orders—bundle offers are effective.

1️⃣2️⃣ Top 3 Pizza Types by Revenue

Query

```
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;
```

Output

Pizza	Revenue
Thai Chicken Pizza	$43,434.25
Barbecue Chicken Pizza	$42,768.00
California Chicken Pizza	$41,409.50

📌 **Business Insight:**
Chicken pizzas combine high demand with premium pricing.

🔹 Advanced SQL Analysis – Queries, Outputs & Business Insights

1️⃣3️⃣ Revenue Contribution by Category

Query

```
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_percentage
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.category
ORDER BY revenue_percentage DESC;
```

Output Summary

Category	Revenue %
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

📌 **Business Insight:**
Balanced revenue distribution reduces dependency on any single category.

1️⃣4️⃣ Cumulative Revenue Generated Over Time

Query

```
SELECT
    date,
    SUM(revenue) OVER (ORDER BY date) AS cumulative_revenue
FROM (
    SELECT
        orders.date,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM orders
    JOIN order_details
        ON order_details.order_id = orders.order_id
    JOIN pizzas
        ON pizzas.pizza_id = order_details.pizza_id
    GROUP BY orders.date
) AS daily_sales;
```

Findings

- Revenue increases steadily over time.
- No sharp volatility observed.

📌 **Business Insight:**
Indicates stable demand and predictable cash flow.

1️⃣5️⃣ Top 3 Revenue-Generating Pizza Types by Category

(Using Ranking & Window Functions)

Query

```
SELECT category, name, revenue, rankbysales
FROM (
    SELECT
        pizza_types.category,
        pizza_types.name,
        SUM(order_details.quantity * pizzas.price) AS revenue,
        RANK() OVER (
            PARTITION BY pizza_types.category
            ORDER BY SUM(order_details.quantity * pizzas.price) DESC
        ) AS rankbysales
    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.category, pizza_types.name
) ranked_pizzas
WHERE rankbysales <= 3
ORDER BY category, rankbysales;
```

Findings

- Each category has distinct revenue leaders.
- High-revenue pizzas are not always the most ordered.
- Premium pricing significantly impacts revenue.

📌 **Business Insight:**
Category-specific bestsellers should be featured in menus and promotions.

▩ Final Conclusion

This project demonstrates how SQL can transform raw transactional data into actionable business insights. Through **basic, intermediate, and advanced analysis**—using joins, aggregations, subqueries, and window functions—the project uncovers customer behavior patterns, product performance trends, and revenue dynamics, showcasing strong real-world SQL proficiency.