PIZZA SALES ANALYSIS USING SQL & POWER BI

Analyzing Data to Uncover Business Insights

2025

Presented by Surbhi Gupta

TABLE OF CONTENTS

PROJECT OVERVIEW

This project analyzes a year’s sales data from a fictional pizza restaurant, focusing on key metrics like product popularity, ingredient demand, peak sales periods, and ordering patterns. Interactive dashboards are utilized to uncover insights that can refine sales strategies, streamline operations, and enhance the customer experience.

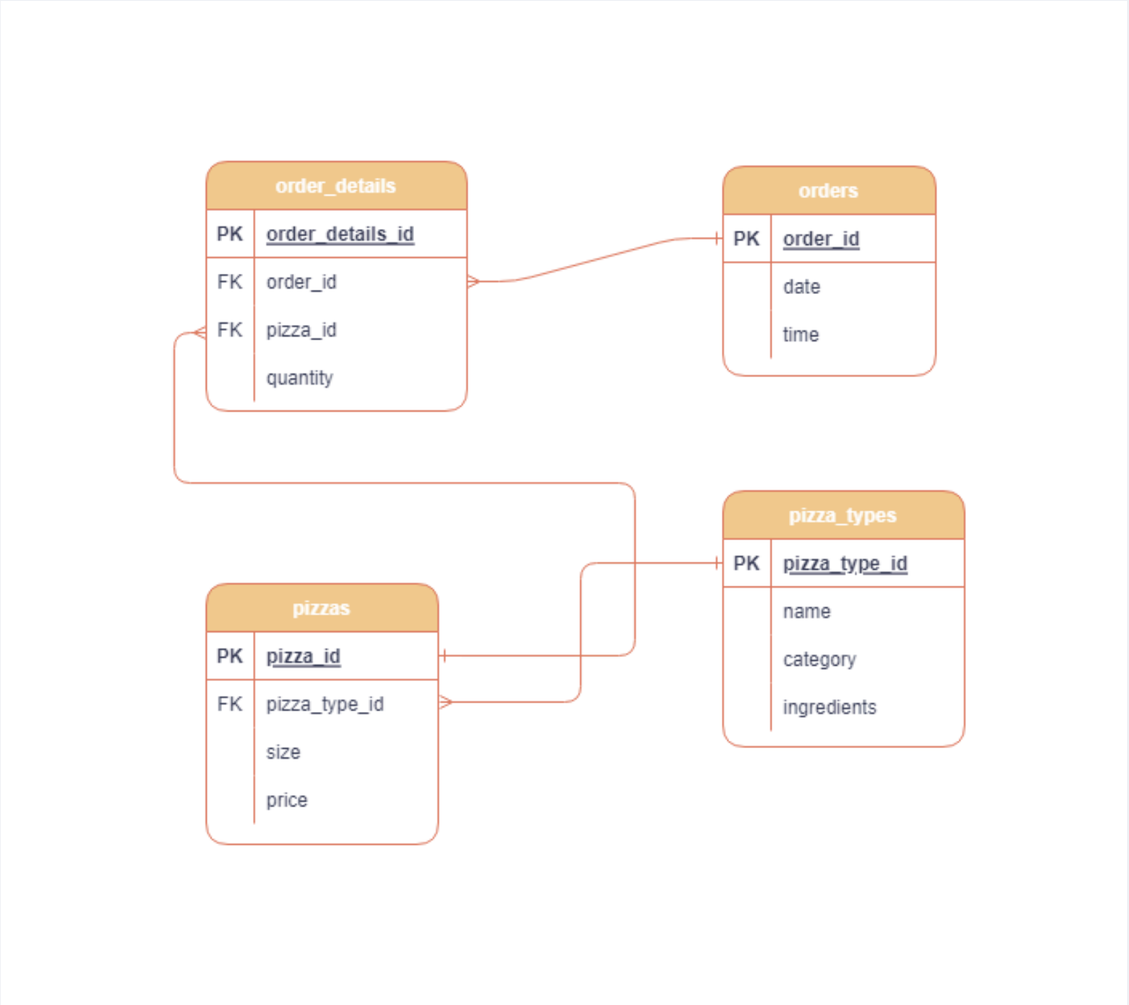
Key Focus Areas:

- **Pizza Performance:** Identifying bestsellers and underperformers to optimize the menu.
- **Category & Size Preferences:** Analyzing trends across vegetarian/non-vegetarian options and pizza sizes.
- **Ingredient Popularity:** Highlighting high-demand ingredients for better inventory and menu planning.
- **Peak Sales Times:** Pinpointing peak hours, days, and high-demand periods for operational efficiency.
- **Monthly Trends:** Identifying seasonal sales trends to drive targeted promotions.

This concise analysis offers actionable recommendations to boost performance and customer satisfaction.

DATA STRUCTURE

Pizza Sale's database structure as seen below consists of four tables: Order Details, Orders, Pizza Types and Pizzas with 48,620 records.



Let’s Start

```sql
--Retrieve the total number of orders placed.

SELECT COUNT(order_id) as "Total_number_orders_place" FROM orders;
```

OUTPUT

	Total_number_orders_place bigint
1	21350

```sql
--Calculate the total revenue generated from pizza sales.

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

OUTPUT

	Revenue numeric
1	817860.05

```sql
--Identify the highest-priced pizza.

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

OUTPUT

	Pizza_Name text	Price double precision
1	The Greek Pizza	35.95

Insights : The Greek pizza, priced at 35.95, is identified as the highest-priced option among the 96 varieties. This insight highlights a premium menu item that could appeal to customers seeking higher-end, specialty pizzas.

```sql
--Identify the most common pizza size ordered.

SELECT pizzas.size AS "Pizza_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;
```

Insights : Large (L) pizzas are the most ordered size (18,526 orders), showing strong customer preference for standard portions, followed by Medium and Small. The very low demand for XL and XXL sizes suggests a need to review pricing, marketing, or menu positioning.

OUTPUT

	Pizza_Size text	order_count bigint
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

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

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
```

Insights : The Classic Deluxe Pizza leads with 2,453 orders, followed closely by Barbecue Chicken, Hawaiian, Pepperoni, and Thai Chicken pizzas, each with similar popularity. This indicates balanced customer preferences for both classic and unique flavors.

	name text	quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

```sql
--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 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;
```

OUTPUT

	category text	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

```sql
--Determine the distribution of orders by hour of the day.

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

	hour numeric	order_count bigint
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
15	23	28

INSIGHTS: Most orders are placed between 12 PM and 7 PM, with the highest order count at 12 PM (2,520 orders). Early morning (9-10 AM) and late night (10-11 PM) see minimal orders, indicating low demand during these times.

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

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

OUTPUT

	category text	count bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9

```sql
--Group the orders by date and calculate the average number of pizzas ordered per day.

SELECT ROUND (AVG( quantity ),0) AS Average_number_of_pizzas_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
```

OUTPUT

	average_number_of_pizzas_ordered_per_day numeric
1	138

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

SELECT pizza_types.name,
SUM(order_details.quantity * Pizzas.price::numeric) 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.name ORDER BY Revenue DESC LIMIT 3
```

OUTPUT

	name text	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

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

SELECT pizza_types.category ,
(sum(order_details.quantity * pizzas.price) /
(SELECT ROUND (SUM(order_details.quantity * Pizzas.price)::numeric,2 )AS total_sales
FROM order_details JOIN Pizzas
ON order_details.pizza_id = pizzas.pizza_id)*100) as percentage_contribution
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 percentage_contribution DESC;
```

OUTPUT

	category text	percentage_contribution double precision
1	Classic	26.90596025566967
2	Supreme	25.45631126009862
3	Chicken	23.955137556847287
4	Veggie	23.682590927384577

```sql
--Analyze the cumulative revenue generated over time.

SELECT order_date, SUM(revenue) over(ORDER BY order_date) as cumulative_revenue
FROM
(
SELECT orders.order_date,
(SUM(order_details.quantity * Pizzas.price)::numeric )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;
```

OUTPUT

	order_date date	cumulative_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60
5	2015-01-05	11929.55
6	2015-01-06	14358.50
7	2015-01-07	16560.70
8	2015-01-08	19399.05
9	2015-01-09	21526.40
10	2015-01-10	23990.35
11	2015-01-11	25862.65
12	2015-01-12	27781.70
13	2015-01-13	29831.30
14	2015-01-14	32358.70
15	2015-01-15	34343.50
16	2015-01-16	36937.65
17	2015-01-17	39001.75
18	2015-01-18	40978.60
19	2015-01-19	43365.75
20	2015-01-20	45763.65
21	2015-01-21	47804.20

Insights: The cumulative revenue shows consistent daily growth over time, starting from ₹2,713.85 on 2015-01-01 and reaching ₹817860.05 by 2015-12-31, reflecting steady customer demand. The increasing trend indicates successful sales performance, likely driven by popular products and effective operations. To maintain this momentum, consider analyzing daily sales drivers (e.g., promotions, top-selling products) and replicating successful strategies. Additionally, identify any seasonal patterns or peak days to optimize inventory and marketing efforts for future growth. Regular tracking of cumulative revenue can help spot anomalies or declining trends early.

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

SELECT category, name, revenue from
(
SELECT category, name,revenue, rank() over (partition by category order by revenue) as rn
FROM
(
SELECT pizza_types.category ,pizza_types.name ,
(SUM(order_details.quantity * Pizzas.price)::numeric )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;
```

OUTPUT

	category text	name text	revenue numeric
1	Chicken	The Chicken Pesto Pizza	16701.75
2	Chicken	The Chicken Alfredo Pizza	16900.25
3	Chicken	The Southwest Chicken Pizza	34705.75
4	Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
5	Classic	The Big Meat Pizza	22968
6	Classic	The Napolitana Pizza	24087
7	Supreme	The Brie Carre Pizza	11588.4999999999
8	Supreme	The Spinach Supreme Pizza	15277.75
9	Supreme	The Calabrese Pizza	15934.25
10	Veggie	The Green Garden Pizza	13955.75
11	Veggie	The Mediterranean Pizza	15360.5
12	Veggie	The Spinach Pesto Pizza	15596

Insights: The Southwest Chicken Pizza dominates with ₹34,705.75 in revenue, indicating strong customer preference, while the Chicken and Classic categories have clear leaders with significant revenue gaps. Supreme and Veggie categories show balanced demand, requiring targeted promotions to boost sales. The Brie Carre Pizza underperforms at ₹11,588.5, suggesting a need for better positioning or pricing adjustments. Focus on marketing top performers like the Southwest Chicken and Napolitana pizzas, bundle them with sides, and analyze low performers to identify and address gaps in demand or customer appeal. Regular monitoring and adjustments will help optimize revenue across all categories.

Recommendations

- Based on the uncovered insights, the following recommendations have been provided :
- Store credit or points can be applied as discounts when purchasing pizzas. Additionally, by occasionally multiplying these points on random days, sales may experience an uplift.
- To enhance Sunday revenue, think about reducing working hours to support employee well-being while also driving pizza sales through limited-time B1G1 flash sales.
- Let customers design custom pizzas with up to 5 toppings. The best ones chosen by the top chef can be added to the menu, and the customer who created it can name the pizza, allowing for regular menu updates based on customer choices.
- At the close of the year, gather feedback from customers to identify areas for improvement in the restaurant.
- Due to the lower demand for XL and XXL-sized pizzas, consider introducing a half-and-half pizza option, allowing customers to enjoy two different pizza varieties on a single pie.