# SQL Mini Project –

# The Great Pizza Sales Analysis

About the Company

# About IDC Pizza

### Fast-Growing Chain

IDC Pizza is a fast-growing pizza chain

### High Volume

Handles thousands of orders every month

### Rich Dataset

Dataset contains: pizza types, sizes, prices, orders & order details

### Analysis Goal

Goal: Analyze sales performance, customer patterns & product insights

# 🎯 Objective of the Mini Project

## Core Tasks

- Clean & explore raw pizza datasets
- Solve business-driven SQL questions
- Use filtering, joins, aggregations & self joins

## Generate insights on:

- ✓ Sales performance
- ✓ Category trends
- ✓ Unordered items
- ✓ Pricing patterns

# PHASE 1 – BASIC EXPLORATION

## Topics Covered:

- Removing duplicates
- Handling NULL values
- Exploring pizza categories and attributes
- Understanding missing data

## What We Achieved:

- Identified distinct pizza categories
- Cleaned ingredient information
- Found pizzas with missing price values

# Phase 1 – Questions, Queries & Output

### Q1: List all unique pizza categories

```
SELECT DISTINCT category FROM pizza_types;
```

| | CATEGORY |
|---|---|
| 1 | Chicken |
| 2 | Classic |
| 3 | Veggie |
| 4 | Supreme |

### Q2: First 5 pizza types (replace NULL ingredients)

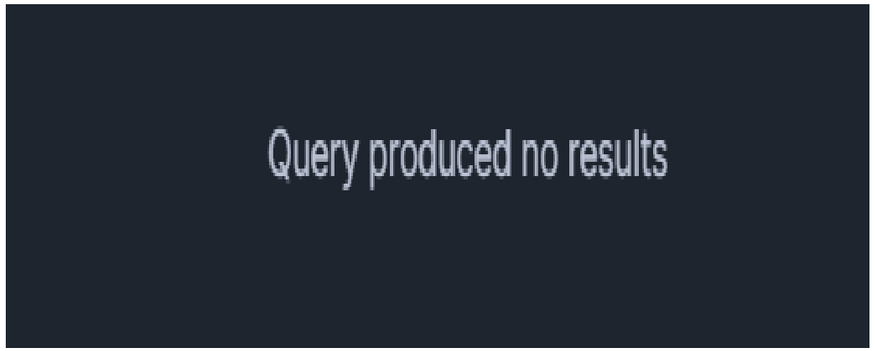
```
SELECT pizza_type_id, name,
 COALESCE(ingredients, 'Missing Data’)
 AS ingredients
FROM pizza_types
LIMIT 5;;
```

| | PIZZA_TYPE_ID | NAME | INGREDIENTS |
|---|---|---|---|
| 1 | bbq_ckn | The Barbecue Chicken Pizza | Barbecued Chicken, Red Peppers, Green Peppers, Tomatoes, Red Onions, Barbecue Sauce |
| 2 | cali_ckn | The California Chicken Pizza | Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, Fontina Cheese, Gouda Cheese |
| 3 | ckn_alfredo | The Chicken Alfredo Pizza | Chicken, Red Onions, Red Peppers, Mushrooms, Asiago Cheese, Alfredo Sauce |
| 4 | ckn_pesto | The Chicken Pesto Pizza | Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Pesto Sauce |
| 5 | southw_ckn | The Southwest Chicken Pizza | Chicken, Tomatoes, Red Peppers, Red Onions, Jalapeno Peppers, Corn, Cilantro, Chipotle Sauce |

# Phase 1 – Questions, Queries & Output

**Q3: Find pizzas missing price**

```
SELECT
  pizza_id,
  pizza_type_id,
  size,
  price
FROM pizzas
WHERE price IS NULL;
```



# PHASE 2 – FILTERING & EXPLORATION

## Topics Covered:

- Date-based filtering
- String matching (LIKE)
- Sorting & pagination
- Range filtering
- Time filtering

**Purpose:**

To understand ordering patterns and filter products/customers efficiently.

# Phase 2– Questions, Queries & Output

### Q1. Orders placed on 2015-01-01

```
SELECT * FROM orders
WHERE date = '2015-01-01';
```

| | ORDER_ID | DATE | TIME |
|---|---|---|---|
| 1 | 1 | 2015-01-01 | 11:38:36 |
| 2 | 2 | 2015-01-01 | 11:57:40 |
| 3 | 3 | 2015-01-01 | 12:12:28 |
| 4 | 4 | 2015-01-01 | 12:16:31 |
| 5 | 5 | 2015-01-01 | 12:21:30 |
| 6 | 6 | 2015-01-01 | 12:29:36 |

### Q2: List pizzas by price (high → low)

```
SELECT *FROM pizzas
ORDER BY price DESC;
```

| | PIZZA_ID | PIZZA_TYPE_ID | SIZE | PRICE |
|---|---|---|---|---|
| 1 | the_greek_xxl | the_greek | XXL | 35.95 |
| 2 | the_greek_xl | the_greek | XL | 25.50 |
| 3 | brie_carre_s | brie_carre | S | 23.65 |
| 4 | ital_veggie_l | ital_veggie | L | 21.00 |
| 5 | spicy_ital_l | spicy_ital | L | 20.75 |
| 6 | thai_ckn_l | thai_ckn | L | 20.75 |

# Phase 2– Questions, Queries & Output

### Q3. Pizzas in L or XL size

```
SELECT *FROM pizzas
WHERE size IN ('L', 'XL');
```

| | PIZZA_ID | PIZZA_TYPE_ID | SIZE | PRICE |
|---|---|---|---|---|
| 1 | bbq_ckn_l | bbq_ckn | L | 20.75 |
| 2 | cali_ckn_l | cali_ckn | L | 20.75 |
| 3 | ckn_alfredo_l | ckn_alfredo | L | 20.75 |
| 4 | ckn_pesto_l | ckn_pesto | L | 20.75 |
| 5 | southw_ckn_l | southw_ckn | L | 20.75 |
| 6 | thai_ckn_l | thai_ckn | L | 20.75 |
| 7 | big_meat_l | big_meat | L | 20.50 |
| 8 | classic_dlx_l | classic_dlx | L | 20.50 |
| 9 | hawaiian_l | hawaiian | L | 16.50 |

### Q4: Pizzas priced between 15$ and 17$

```
SELECT *FROM pizzas
WHERE price BETWEEN 15 AND 17
ORDER BY price;
```

| | PIZZA_ID | PIZZA_TYPE_ID | SIZE | PRICE |
|---|---|---|---|---|
| 1 | pepperoni_l | pepperoni | L | 15.25 |
| 2 | five_cheese_m | five_cheese | M | 15.50 |
| 3 | veggie_veg_m | veggie_veg | M | 16.00 |
| 4 | the_greek_m | the_greek | M | 16.00 |
| 5 | napolitana_m | napolitana | M | 16.00 |
| 6 | ital_cpcllo_m | ital_cpcllo | M | 16.00 |
| 7 | green_garden_m | green_garden | M | 16.00 |
| 8 | classic_dlx_m | classic_dlx | M | 16.00 |
| 9 | big_meat_m | big_meat | M | 16.00 |

# Phase 2– Questions, Queries & Output

### Q5. Pizzas containing the word “Chicken”

```
SELECT *FROM pizza_types
WHERE name LIKE '%Chicken%';
```

| | PIZZA_TYPE_ID | NAME | CATEGORY | INGREDIENTS |
|---|---|---|---|---|
| 1 | bbq_ckn | The Barbecue Chicken Pizza | Chicken | Barbecued Chicken, Red Peppers, Green Peppers, Toma |
| 2 | cali_ckn | The California Chicken Pizza | Chicken | Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, F |
| 3 | ckn_alfredo | The Chicken Alfredo Pizza | Chicken | Chicken, Red Onions, Red Peppers, Mushrooms, Asiago |
| 4 | ckn_pesto | The Chicken Pesto Pizza | Chicken | Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Pesto |
| 5 | southw_ckn | The Southwest Chicken Pizza | Chicken | Chicken, Tomatoes, Red Peppers, Red Onions, Jalapeno |
| 6 | thai_ckn | The Thai Chicken Pizza | Chicken | Chicken, Pineapple, Tomatoes, Red Peppers, Thai Sweet |

### Q6: Orders on 2015-02-15 OR placed after 8 PM

```
SELECT *FROM orders
WHERE date = '2015-02-15’
OR time > '20:00:00';
```

| | ORDER_ID | DATE | TIME |
|---|---|---|---|
| 1 | 60 | 2015-01-01 | 20:05:16 |
| 2 | 61 | 2015-01-01 | 20:08:43 |
| 3 | 62 | 2015-01-01 | 20:50:16 |
| 4 | 63 | 2015-01-01 | 20:51:42 |
| 5 | 64 | 2015-01-01 | 20:52:08 |
| 6 | 65 | 2015-01-01 | 21:16:00 |
| 7 | 66 | 2015-01-01 | 21:47:55 |
| 8 | 67 | 2015-01-01 | 22:03:40 |
| 9 | 68 | 2015-01-01 | 22:07:32 |


# PHASE 3 – SALES PERFORMANCE & JOIN ANALYSIS

## Topics Covered:

- Aggregations (SUM, AVG)
- Joining multiple tables
- Category-wise insights
- Identifying unordered pizzas
- Comparing prices using self join
- Subquery, Windows Functions, CTEs

## Goal:

To understand sales volume, revenue, category performance & pricing strategy.

# Phase 3 – Questions, Queries & Output

### Q1: Total quantity of pizzas sold

```
SELECT SUM(quantity) AS total_pizza_sold
FROM order_details;
```

| | TOTAL_PIZZA_SOLD |
|---|---|
| 1 | 49574 |

### Q2: Average pizza price

```
SELECT ROUND(AVG(price), 2)
  AS avg_pizza_price
FROM pizzas;
```

| | AVG_PIZZA_PRICE |
|---|---|
| 1 | 16.44 |

# Phase 3 – Questions, Queries & Output

## Q3: Total order value per order

```
SELECT od.order_id,
   SUM(od.quantity * p.price)
   AS total_order_value
FROM order_details od
JOIN pizzas p
   ON od.pizza_id = p.pizza_id
GROUP BY od.order_id
ORDER BY total_order_value DESC;
```

| | ORDER_ID | TOTAL_ORDER_VALUE |
|---|---|---|
| 1 | 18845 | 444.20 |
| 2 | 10760 | 417.15 |
| 3 | 1096 | 285.15 |
| 4 | 6169 | 284.00 |
| 5 | 740 | 280.95 |
| 6 | 12257 | 276.75 |
| 7 | 1685 | 272.75 |
| 8 | 4482 | 267.20 |
| 9 | 5452 | 267.10 |

## Q4: Total quantity sold per pizza category

```
SELECT pt.category,SUM(od.quantity)
   AS total_quantity_sold
FROM pizza_types pt
JOIN pizzas p
   ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od
   ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY total_quantity_sold DESC;
```

| | CATEGORY | TOTAL_QUANTITY_SOLD |
|---|---|---|
| 1 | Classic | 14888 |
| 2 | Supreme | 11987 |
| 3 | Veggie | 11649 |
| 4 | Chicken | 11050 |

# Phase 3 – Questions, Queries & Output

**Q5: Categories with more than 5000 pizzas sold**

```
SELECT pt.category,
    SUM(od.quantity)
    AS total_quantity_sold
FROM pizza_types pt
JOIN pizzas p
    ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od
    ON p.pizza_id = od.pizza_id
GROUP BY pt.category
HAVING SUM(od.quantity) > 5000;
```

| | CATEGORY | TOTAL_QUANTITY_SOLD |
|---|---|---|
| 1 | Chicken | 11050 |
| 2 | Veggie | 11649 |
| 3 | Supreme | 11987 |
| 4 | Classic | 14888 |

# Phase 3 – Questions, Queries & Output

**Q6: Pizzas never ordered**

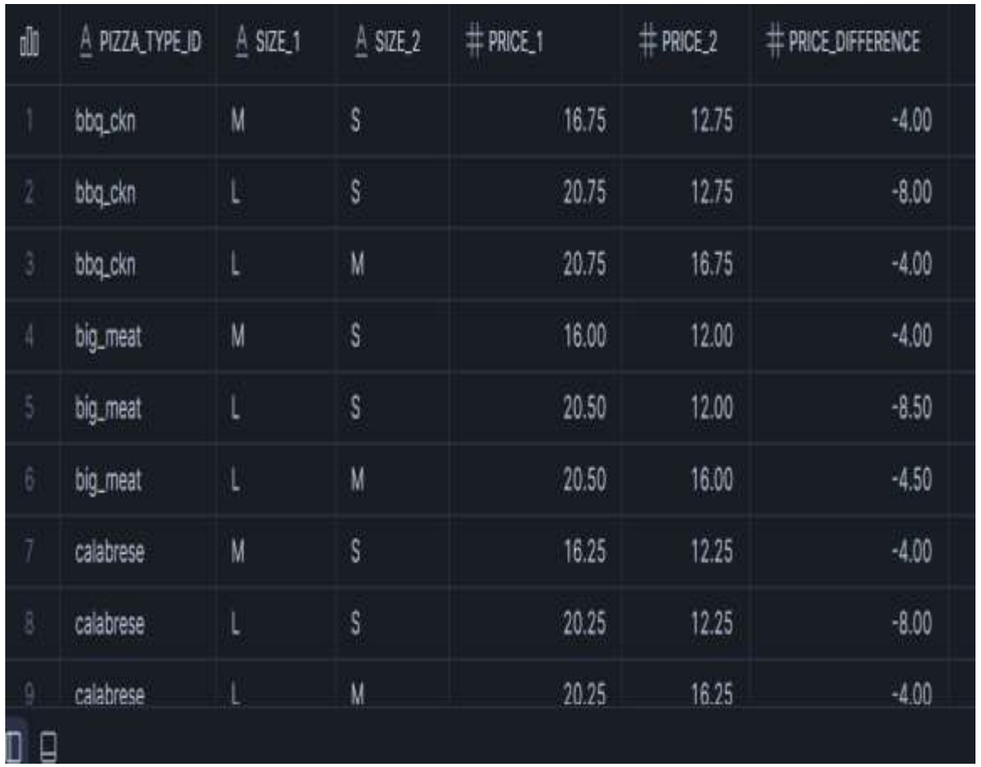
```
SELECT p.pizza_id,
    p.size,
    pt.category,
    od.order_id
FROM pizzas p
LEFT JOIN pizza_types pt
    ON p.pizza_type_id = pt.pizza_type_id
LEFT JOIN order_details od
    ON p.pizza_id = od.pizza_id
WHERE od.order_id IS NULL;
```

| | PIZZA_ID | SIZE | CATEGORY | ORDER_ID |
|---|---|---|---|---|
| 1 | big_meat_l | L | Classic | null |
| 2 | big_meat_m | M | Classic | null |
| 3 | five_cheese_m | M | Veggie | null |
| 4 | five_cheese_s | S | Veggie | null |
| 5 | four_cheese_s | S | Veggie | null |

# Phase 3 – Questions, Queries & Output

**Q7: Price difference between pizza sizes (SELF JOIN)**

```
SELECT p1.pizza_type_id,
     p1.size AS size_1,
     p2.size AS size_2,
     p1.price AS price_1,
     p2.price AS price_2,
     (p2.price - p1.price)
     AS price_difference
FROM pizzas p1
JOIN pizzas p2
     ON p1.pizza_type_id = p2.pizza_type_id
     AND p1.size < p2.size
ORDER BY p1.pizza_type_id;
```



| | PIZZA_TYPE_ID | SIZE_1 | SIZE_2 | PRICE_1 | PRICE_2 | PRICE_DIFFERENCE |
|---|---|---|---|---|---|---|
| 1 | bbq_ckn | M | S | 16.75 | 12.75 | -4.00 |
| 2 | bbq_ckn | L | S | 20.75 | 12.75 | -8.00 |
| 3 | bbq_ckn | L | M | 20.75 | 16.75 | -4.00 |
| 4 | big_meat | M | S | 16.00 | 12.00 | -4.00 |
| 5 | big_meat | L | S | 20.50 | 12.00 | -8.50 |
| 6 | big_meat | L | M | 20.50 | 16.00 | -4.50 |
| 7 | calabrese | M | S | 16.25 | 12.25 | -4.00 |
| 8 | calabrese | L | S | 20.25 | 12.25 | -8.00 |
| 9 | calabrese | L | M | 20.25 | 16.25 | -4.00 |

# Phase 3 – Questions, Queries & Output

**Q8: Most expensive pizza in each category (Correlated Subquery)**

```
SELECT
     pt.category,
     p.pizza_id,
     p.size,
     p.price
FROM pizzas p
JOIN pizza_types pt
     ON p.pizza_type_id = pt.pizza_type_id
WHERE p.price = (
     SELECT MAX(p2.price)
     FROM pizzas p2
     WHERE p2.pizza_type_id = p.pizza_type_id);
```

| | CATEGORY | PIZZA_ID | SIZE | PRICE |
|---|---|---|---|---|
| 1 | Chicken | bbq_ckn_l | L | 20.75 |
| 2 | Chicken | cali_ckn_l | L | 20.75 |
| 3 | Chicken | ckn_alfredo_l | L | 20.75 |
| 4 | Chicken | ckn_pesto_l | L | 20.75 |
| 5 | Chicken | southw_ckn_l | L | 20.75 |
| 6 | Chicken | thai_ckn_l | L | 20.75 |
| 7 | Classic | big_meat_l | L | 20.50 |
| 8 | Classic | classic_dlx_l | L | 20.50 |
| 9 | Classic | hawaiian_l | L | 16.50 |
| 10 | Classic | ital_cpcllo_l | L | 20.50 |

# Phase 3 – Questions, Queries & Output

**Q9: Pizzas priced above the average pizza price**

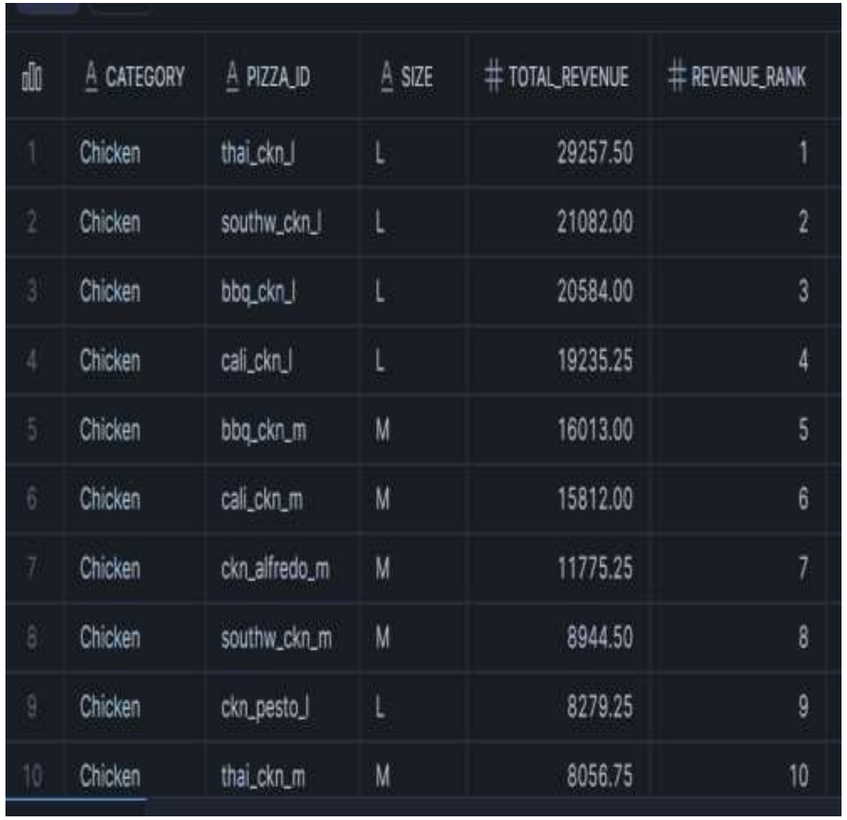
```
SELECT
    p.pizza_id,
    p.size,
    p.price
FROM pizzas p
WHERE p.price > (
   SELECT AVG(price) FROM pizzas)
ORDER BY p.price DESC;
```

| | PIZZA_ID | SIZE | PRICE |
|---|---|---|---|
| 1 | the_greek_xxl | XXL | 35.95 |
| 2 | the_greek_xl | XL | 25.50 |
| 3 | brie_carre_s | S | 23.65 |
| 4 | ital_veggie_l | L | 21.00 |
| 5 | bbq_ckn_l | L | 20.75 |
| 6 | prsc_argla_l | L | 20.75 |
| 7 | peppr_salami_l | L | 20.75 |
| 8 | ital_supr_l | L | 20.75 |
| 9 | soppressata_l | L | 20.75 |
| 10 | spicy_ital_l | L | 20.75 |

# Phase 3 – Questions, Queries & Output

**Q10: Rank pizzas by total revenue within each category**

```
SELECT
pt.category,
p.pizza_id,
p.size,
SUM(od.quantity * p.price) AS total_revenue,
    RANK() OVER (
      PARTITION BY pt.category
      ORDER BY SUM(od.quantity * p.price) DESC )
      AS revenue_rank
FROM order_details od
JOIN pizzas p
  ON od.pizza_id = p.pizza_id
JOIN pizza_types pt
  ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category, p.pizza_id, p.size;
```



| | CATEGORY | PIZZA_ID | SIZE | TOTAL_REVENUE | REVENUE_RANK |
|---|---|---|---|---|---|
| 1 | Chicken | thai_ckn_l | L | 29257.50 | 1 |
| 2 | Chicken | southw_ckn_l | L | 21082.00 | 2 |
| 3 | Chicken | bbq_ckn_l | L | 20584.00 | 3 |
| 4 | Chicken | cali_ckn_l | L | 19235.25 | 4 |
| 5 | Chicken | bbq_ckn_m | M | 16013.00 | 5 |
| 6 | Chicken | cali_ckn_m | M | 15812.00 | 6 |
| 7 | Chicken | ckn_alfredo_m | M | 11775.25 | 7 |
| 8 | Chicken | southw_ckn_m | M | 8944.50 | 8 |
| 9 | Chicken | ckn_pesto_l | L | 8279.25 | 9 |
| 10 | Chicken | thai_ckn_m | M | 8056.75 | 10 |

# Phase 3 – Questions, Queries & Output

**Q11: Top 3 best-selling pizzas in each size (ROW_NUMBER)**

```
WITH sales AS (
SELECT  p.size,p.pizza_id,
   SUM(od.quantity) AS total_qty
FROM order_details od
JOIN pizzas p
   ON od.pizza_id = p.pizza_id
GROUP BY p.size, p.pizza_id)
SELECT *FROM (
   SELECT  size,pizza_id,
     total_qty,
     ROW_NUMBER() OVER (PARTITION BY size
     ORDER BY total_qty DESC) AS rn
   FROM sales) t
 WHERE rn <= 3;
```

| | SIZE | PIZZA_ID | TOTAL_QTY | RN |
|---|---|---|---|---|
| 1 | S | big_meat_s | 1914 | 1 |
| 2 | S | hawaiian_s | 1020 | 2 |
| 3 | S | classic_dlx_s | 799 | 3 |
| 4 | XL | the_greek_xl | 552 | 1 |
| 5 | L | thai_ckn_l | 1410 | 1 |
| 6 | L | five_cheese_l | 1409 | 2 |
| 7 | L | four_cheese_l | 1316 | 3 |
| 8 | M | classic_dlx_m | 1181 | 1 |
| 9 | M | bbq_ckn_m | 956 | 2 |
| 10 | M | cali_ckn_m | 944 | 3 |

# Phase 3 – Questions, Queries & Output

**Q12: CTE to calculate revenue per pizza type & filter those above ₹40,000**

```
WITH revenue_cte AS (
    SELECT
        pt.pizza_type_id,
        pt.name,
        SUM(od.quantity * p.price)
        AS total_revenue
    FROM order_details od
    JOIN pizzas p
        ON od.pizza_id = p.pizza_id
    JOIN pizza_types pt
        ON p.pizza_type_id = pt.pizza_type_id
    GROUP BY pt.pizza_type_id, pt.name)
SELECT *FROM revenue_cte
WHERE total_revenue > 40000
ORDER BY total_revenue DESC;
```

Table | Chart

| | A PIZZA_TYPE_ID | A NAME | # TOTAL_REVENUE |
|---|---|---|---|
| 1 | thai_ckn | The Thai Chicken Pizza | 43434.25 |
| 2 | bbq_ckn | The Barbecue Chicken Pizza | 42768.00 |
| 3 | cali_ckn | The California Chicken Pizza | 41409.50 |

# Key Business Insights

**Sales Volume Milestone**

**The Stat:** 49,574 Pizzas Sold.

**The Insight:** Reaching nearly 50,000 sales confirms strong market demand and established operational stability.

**Effective Pricing Strategy**

**The Stat:** $16.44 Average Price.

**The Insight:** An average transaction of $16.44 indicates that mid-range pricing is the primary driver of sales volume.

**High-Value Opportunities**

**The Stat:** Orders #18845 & #10760.

**The Insight:** These two exceptionally large orders highlight a potential opportunity to expand into bulk sales.

**Category Growth Potential**

**The Stat:** Chicken Category Ranks Last.

**The Insight:** Classic pizzas lead in volume, while Chicken sells the least. Targeted marketing for Chicken pizzas represents a clear area for revenue growth.

**Inventory Efficiency**

**The Stat:** 5 Items with ZERO Sales.

**The Insight:** Items like 'Big Meat' (L & M) and 'Five Cheese' (M & S) and ‘Four Cheese’ (S)  have no sales history. Removing them will optimize inventory space without impacting revenue.

# Key Business Insights

**Pricing Opportunity**
**The Stat:** $4.00 Price Gap.
**The Insight:** The Large Hawaiian pizza ($16.50) is significantly cheaper than similar Classic pizzas ($20.50). Raising its price to match the category average could instantly boost profits.

**High-End Options**
**The Stat:** $35.95 for One Pizza.
**The Insight:** The XXL Greek Pizza and Small Brie Carre ($23.65) are priced well above average. These "luxury" items are crucial for driving up the total value of large orders.

**Top Sellers**
**The Stat:** 2 Clear Winners.
**The Insight:** In the Chicken category, revenue is concentrated in just two pizzas: Thai Chicken and Southwest Chicken. Future marketing should prioritize these best-sellers over lower-performing items.

**Customer Habits**
**The Stat:** Distinct Preferences by Size.
**The Insight:** Solo diners (Small size) prefer heavy meat pizzas, while groups (Large size) prefer distinct flavors like Thai Chicken. Website images should change based on the selected size to match these preferences.

**Quality Over Quantity**
**The Stat:** 3 Items > $40,000 Revenue.
**The Insight:** Even though Chicken pizzas sell fewer units than Classics, they make the most money. The Thai, BBQ, and California Chicken pizzas are the only items earning over $40k, making them the most valuable products on the menu.