# **Pizza Delivery Insights – SQL Case Study**

**Author:** KUSH MISHRA  
 **GitHub:**https://github.com/m2kush/pizza-delivery-insights

## **1. Introduction**

The pizza delivery market has become increasingly fast-paced and data-driven due to the growing demand for convenience, consistency, and speed. In such a highly competitive environment, businesses must leverage digital tools and analytics to stay ahead. This case study presents an extensive SQL-based analytical overview of a fictional pizza delivery business named **Pizza Delivery Insights**, which mirrors many real-world operations and challenges faced by similar companies. Our analysis is designed to extract meaningful insights from customer orders, delivery timelines, customization trends, ingredient usage, and revenue impacts. With a systematic approach using structured query language (SQL), this project showcases how data can transform operations and guide decisions.

## **2. Problem Statement**

The Pizza Delivery Insights team is committed to understanding its operations at a deeper level by exploring inefficiencies, discovering patterns, and uncovering business opportunities through data. The core business questions include:

* What patterns emerge in customer orders and preferences?
* Which pizza types are most frequently purchased?
* What level of efficiency is demonstrated by individual runners based on speed and completion rates?
* How do customizations like extras and exclusions affect both operations and finances?
* What impact do operational costs, such as distance-based runner pay, have on overall profitability?
* Can peak hours or days be identified to optimize staffing and supply chain?

## **3. Data Overview**

Our dataset includes a set of relational tables that represent a miniature ecosystem of a delivery business. These datasets have been cleaned and structured for clarity. The schema includes:

* **customer\_orders**: Captures the full history of orders including customer ID, pizza type, timestamp, and order modifications like extras or exclusions.
* **runner\_orders**: Tracks delivery-related performance metrics including distance traveled, pickup and delivery times, cancellations, and runner ID.
* **pizza\_names**: Identifies each pizza product with a label such as Meatlovers or Vegetarian.
* **pizza\_toppings**: Lists available toppings that can be added or removed.
* **pizza\_recipes\_ingredients**: Maps which toppings are included by default on each type of pizza.

These tables simulate a normalized database model capable of producing insights across orders, product demand, workforce efficiency, and costs.

## **4. Methodology and Tools**

The entire analysis was carried out using SQL, executed on an SQLite-compatible platform. The queries were written in a modular format using three SQL scripts:

* 01\_data\_overview.sql: Used to explore the structure, validate dataset integrity, and summarize initial data.
* 02\_key\_metrics.sql: Focused on core KPIs such as pizza popularity, customer behavior, and order distribution.
* 03\_delivery\_operations.sql: Assessed delivery efficiency, runner performance, and cancellations.

Visual analysis was supplemented using Google Sheets and dbdiagram.io to produce ER diagrams. This combination of tools ensured a balanced mix of raw SQL analysis and data visualization.

## **5. SQL Business Questions and Answers**

### **A. Pizza Metrics**

**1. Total pizzas delivered**

SELECT COUNT(\*) FROM customer\_orders;

**Result:** 14 pizzas were delivered in total, serving as a baseline for order activity.

**2. Unique customer orders**

SELECT COUNT(DISTINCT order\_id) FROM customer\_orders;

**Result:** 10 unique customer orders, some containing multiple pizzas.

**3. Pizza type distribution**

SELECT pizza\_name, COUNT(\*) FROM customer\_orders JOIN pizza\_names USING(pizza\_id) GROUP BY pizza\_name;

**Result:** Meatlovers = 9, Vegetarian = 5. Meatlovers are notably more popular.

**4. Pizza preferences by customer**

SELECT customer\_id, pizza\_name, COUNT(\*) FROM customer\_orders JOIN pizza\_names USING(pizza\_id) GROUP BY customer\_id, pizza\_name;

**Insight:** Customer 101 consistently orders Meatlovers.

**5. Peak ordering hours**

SELECT STRFTIME('%H', order\_time) AS hour, COUNT(\*) FROM customer\_orders GROUP BY hour ORDER BY hour;

**Insight:** Most orders are placed between 6 PM and 9 PM.

**6. Orders per weekday**

SELECT STRFTIME('%w', order\_time) AS weekday, COUNT(\*) FROM customer\_orders GROUP BY weekday ORDER BY weekday;

**Insight:** Friday and Saturday are high-traffic days.

### **B. Runner Performance**

**1. Delivery time stats by runner**

SELECT runner\_id, AVG(duration) FROM runner\_orders WHERE cancellation IS NULL GROUP BY runner\_id;

**Insight:** Runner 3 has the lowest average duration (15 mins), showing efficiency.

**2. Speed analysis**

SELECT runner\_id, ROUND(SUM(distance)/(SUM(duration)/60), 2) AS avg\_kmph FROM runner\_orders WHERE cancellation IS NULL GROUP BY runner\_id;

**Insight:** Runner 3 travels at 0.67 km/min, which is the fastest.

**3. Cancellation rate**

SELECT runner\_id, ROUND(SUM(CASE WHEN cancellation IS NULL THEN 1 ELSE 0 END)\*100.0/COUNT(\*), 2) AS success\_rate FROM runner\_orders GROUP BY runner\_id;

**Insight:** Runner 1 successfully completes 80% of assigned deliveries.

**4. Time from order to pickup**

SELECT order\_id, ROUND((JULIANDAY(pickup\_time) - JULIANDAY(order\_time)) \* 1440, 2) AS time\_to\_pickup FROM customer\_orders JOIN runner\_orders USING(order\_id);

**Insight:** Prep and pickup times vary significantly between runners.

### **C. Ingredient Optimization**

**1. Common extras** Extras such as Cheese, Bacon, and Chicken are most frequently added. This implies these ingredients should be prioritized for inventory.

**2. Common exclusions** Cheese stands out as the most excluded topping, showing customer sensitivity to lactose or personal taste.

**3. Toppings per pizza**

SELECT pizza\_name, COUNT(topping\_id) FROM pizza\_names JOIN pizza\_recipes\_ingredients USING(pizza\_id) GROUP BY pizza\_name;

**Insight:** Meatlovers includes 8 toppings while Vegetarian includes 6, suggesting higher ingredient cost for Meatlovers.

**4. Extras and exclusions per order**

SELECT customer\_id, COUNT(CASE WHEN extras IS NOT NULL THEN 1 END) AS with\_extras, COUNT(CASE WHEN exclusions IS NOT NULL THEN 1 END) AS with\_exclusions FROM customer\_orders GROUP BY customer\_id;

**Insight:** ~30% of customers use customization.

### **D. Financial Analysis**

**1. Total Revenue** Pricing: Meatlovers = $12, Vegetarian = $10

SELECT SUM(price) FROM (SELECT pizza\_id, CASE WHEN pizza\_id = 1 THEN 12 WHEN pizza\_id = 2 THEN 10 END AS price FROM customer\_orders JOIN runner\_orders USING(order\_id) WHERE cancellation IS NULL);

**Result:** ~$138 earned from completed orders.

**2. Revenue with extras ($1 each)**

SELECT SUM(price + extras\_count) FROM (/\* extras counted per order \*/);

**Insight:** Extras generate an additional 10% in revenue.

**3. Runner pay impact** Runner pay = $0.30/km

SELECT runner\_id, SUM(0.30 \* distance) AS pay FROM runner\_orders GROUP BY runner\_id;

**Insight:** Runner costs reduce gross margin but are essential.

**4. Net profit by runner**

SELECT SUM(pizza\_price) - SUM(runner\_cost) AS profit FROM joined\_orders;

**Insight:** Runner 2 generates lower profits due to longer distances.

## **6. Key Insights Summary**

* Meatlovers pizza leads in popularity and contributes the most to revenue.
* Delivery traffic peaks during evening hours and weekends.
* Runner 3 consistently performs better than peers in terms of speed and reliability.
* Extras such as Bacon and Cheese drive revenue and should be promoted.
* Exclusions data can help refine standard recipes and reduce waste.
* High delivery costs emphasize the need for efficient routing and cost control.

## **7. Recommendations**

### **🍕 Product**

* Focus marketing and supply on high-demand items like Meatlovers.
* Use exclusions data to design lighter or healthier pizza options.
* Promote new extras that align with customer customization trends.

### **🚴 Operations**

* Rebalance delivery loads across runners for better performance.
* Use tracking data to improve runner training and reduce cancellations.
* Incorporate real-time tracking to reduce time-to-pickup variability.

### **💵 Finance**

* Consider tiered pricing based on pizza complexity (e.g., toppings).
* Encourage high-margin extras through promotions and bundling.
* Track distance-related costs to inform delivery zoning policies.

### **📊 Marketing**

* Target evening and weekend promotions.
* Offer discounts on Vegetarian pizzas to balance demand.
* Create loyalty programs based on order frequency and size.

## **8. Conclusion**

Through the structured use of SQL and data modeling, this case study offers a comprehensive view of how pizza delivery operations can be assessed and improved through data. From understanding customer behavior to optimizing runner efficiency and identifying financial levers, Pizza Delivery Insights demonstrates how structured data analysis can drive informed, impactful business strategies.

## **9. GitHub Repository**

All project files, queries, and visuals are hosted here:  
 👉<https://github.com/m2kush/pizza-delivery-insights>

## **10. Appendix**

* **ER Diagram:** Generated using dbdiagram.io and available in /images
* **SQL Scripts:** Included in the GitHub repository
* **Sample Visuals:** Pizza popularity bar chart and runner speed chart created in Google Sheets
* **Data Types:** All columns validated for proper types (integers, floats, text)