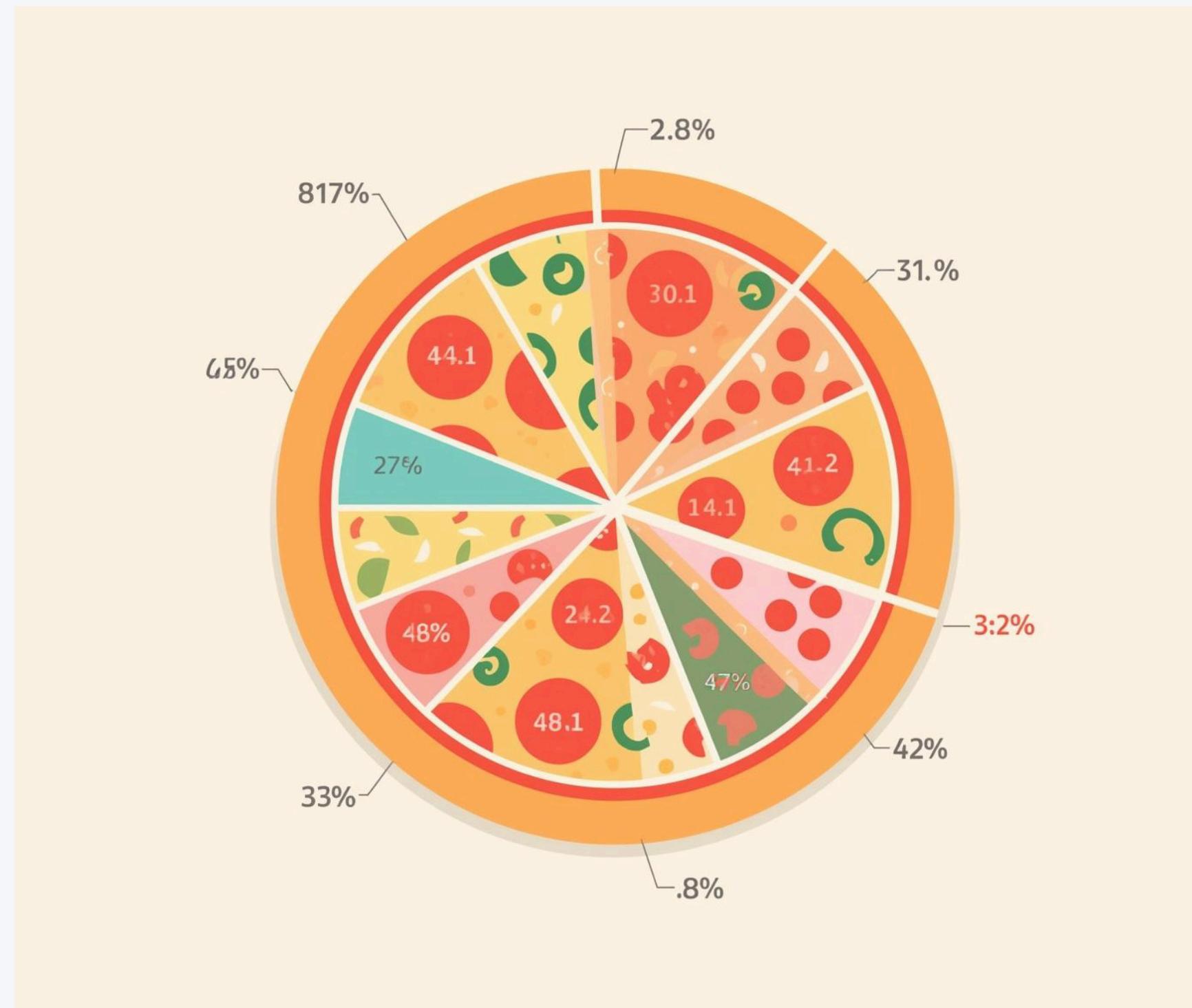


# Pizza Sales Analysis

---



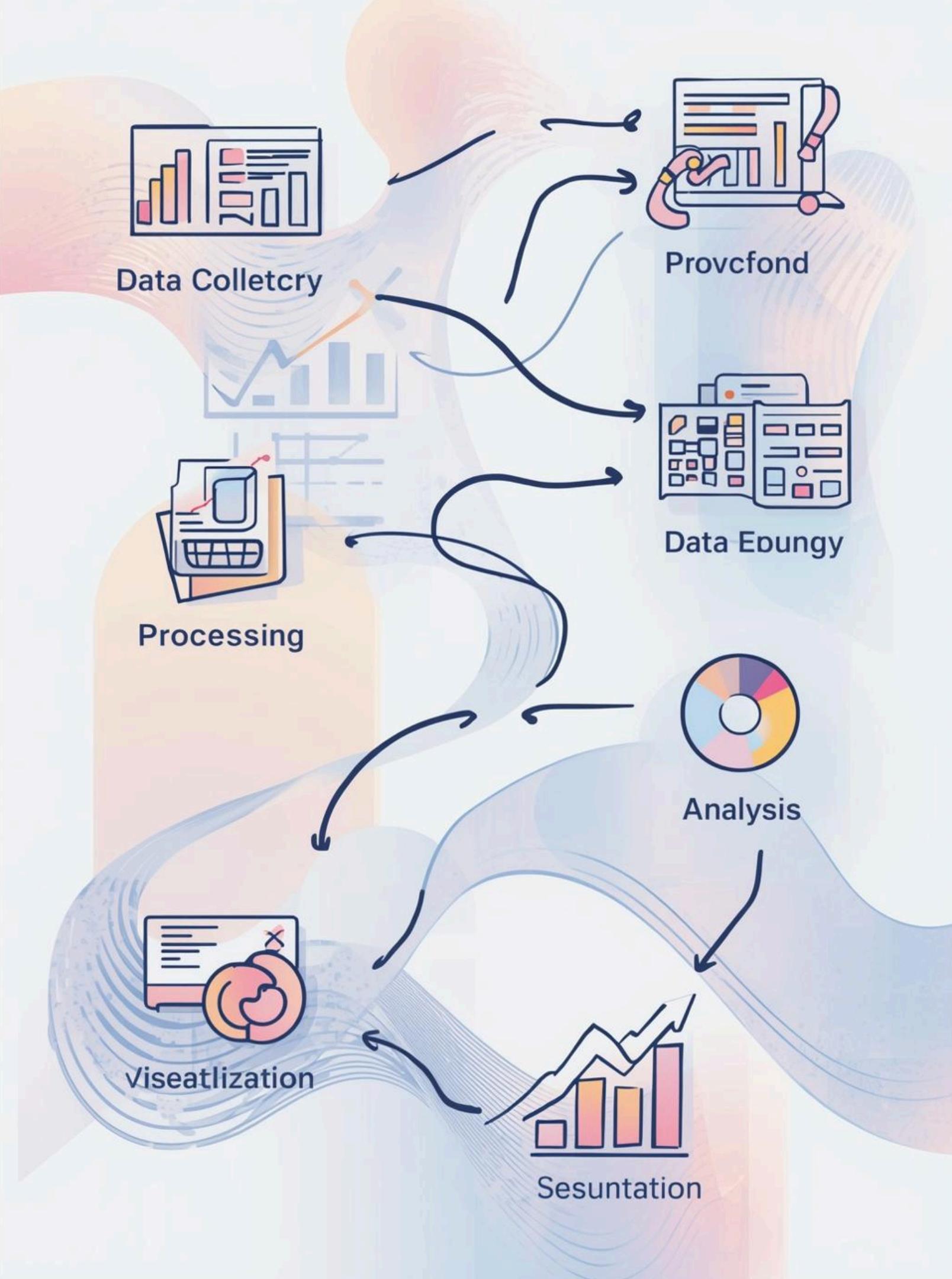
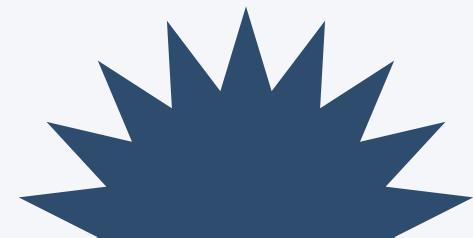
*divya joshi*

DATA ANALYST

# Project Overview

## Understanding the Purpose and Goals of Data Analysis in Business

The primary objective of this project is to **analyze pizza sales data using SQL** to extract meaningful insights that inform business decisions. By focusing on a beginner-friendly approach, this project is designed for entry-level analysts who seek to enhance their understanding of data analytics and its significance in driving informed strategies.



# Essential Tools for Data Analysis

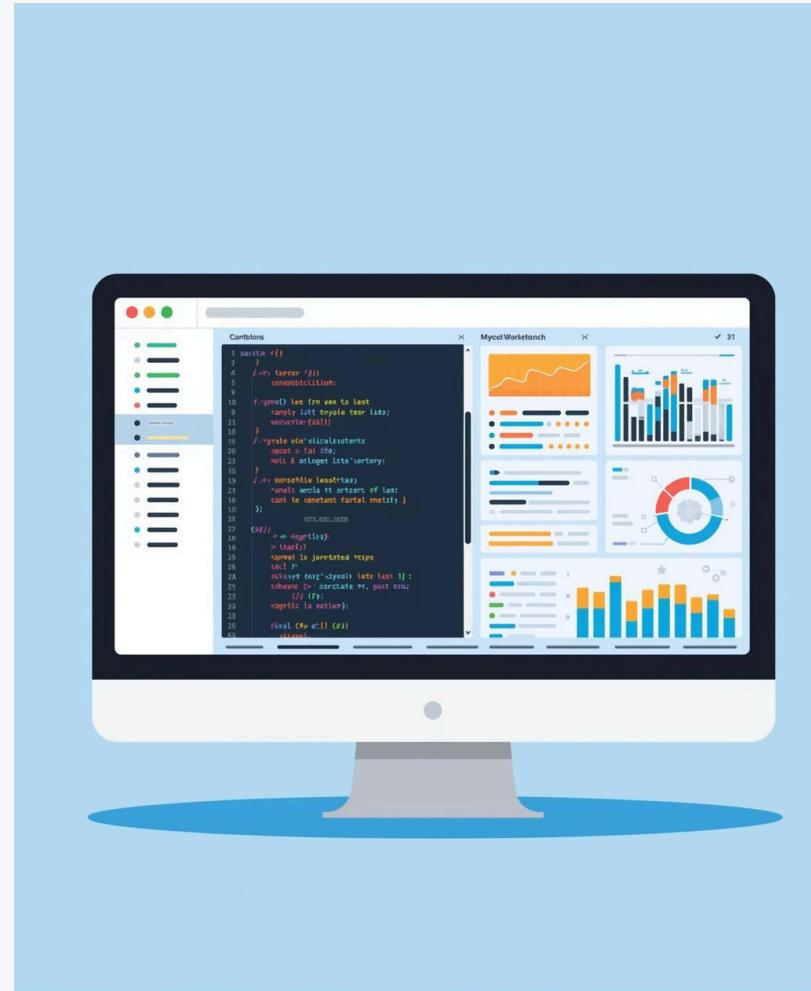
## MySQL

Open-source database for data management.



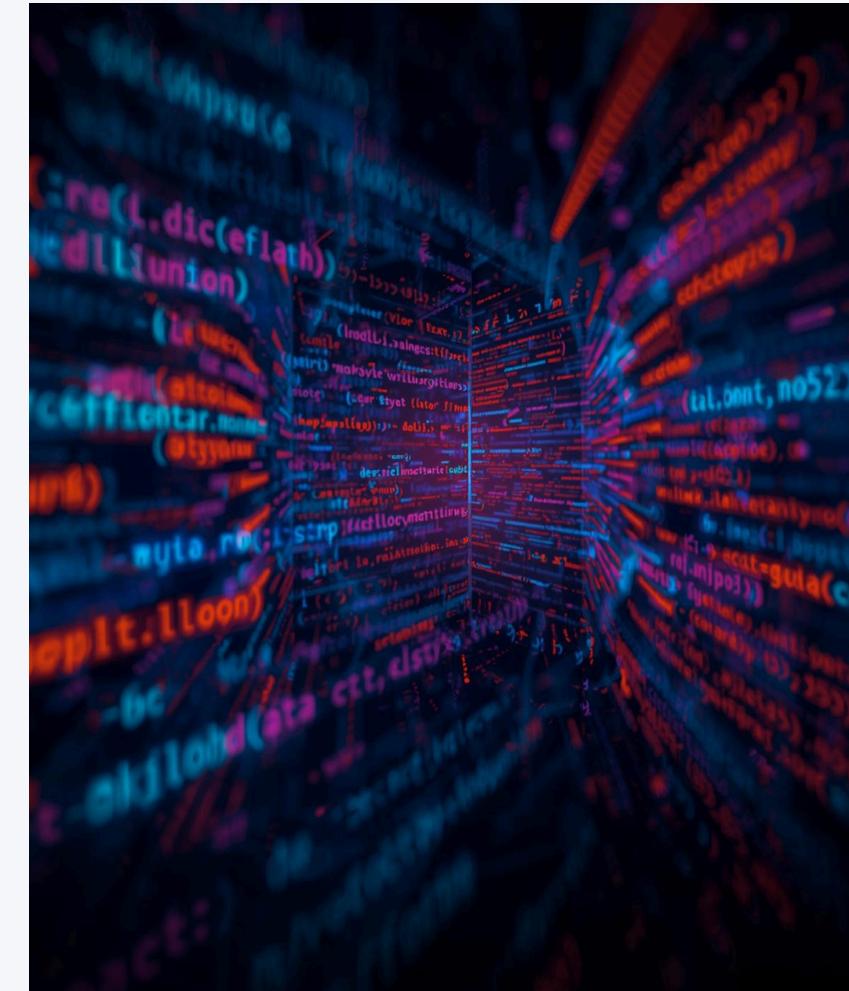
## MySQL Workbench

Tool for SQL query writing visually.



## SQL

Language for database querying and management.



## GitHub

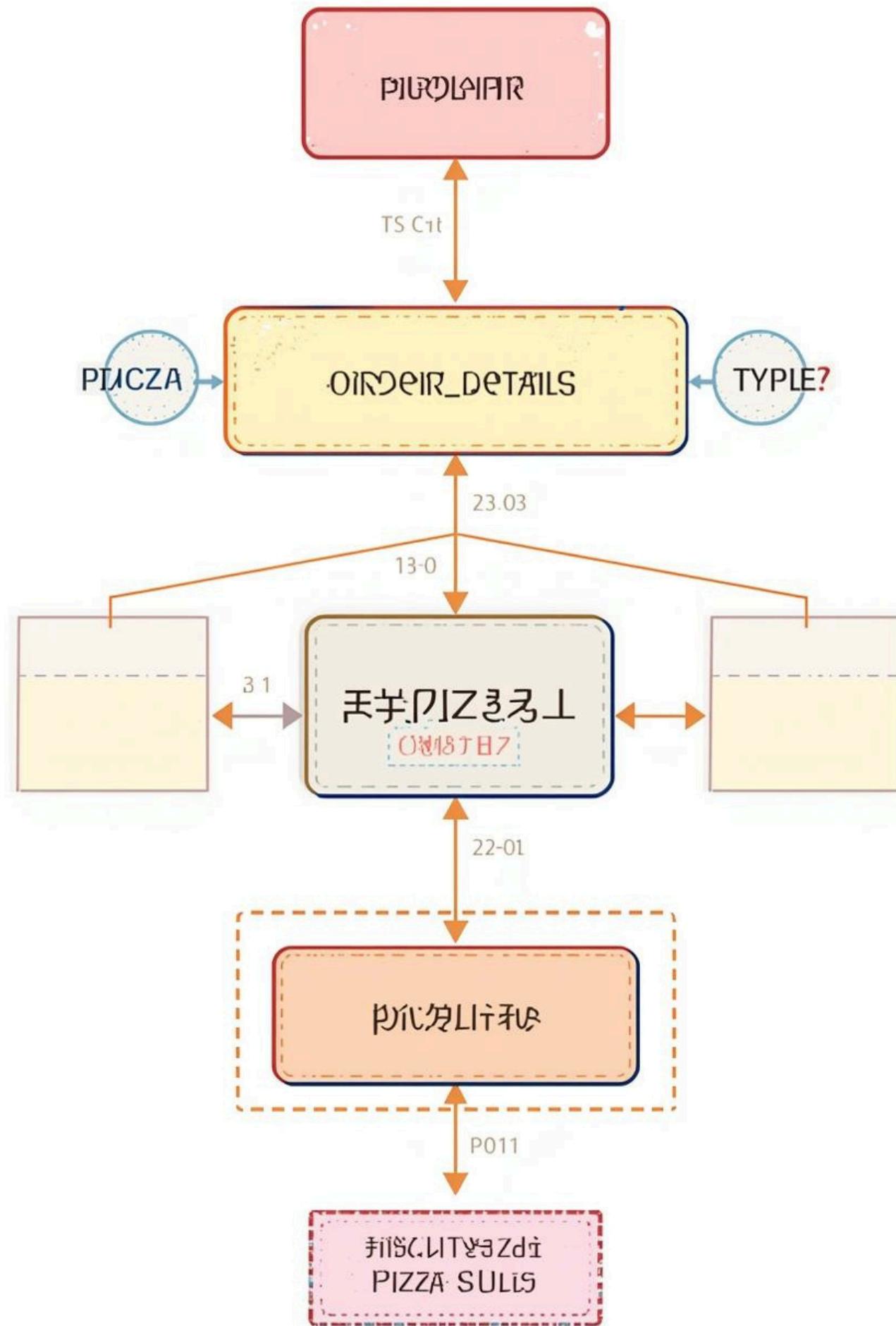
Platform for version control and sharing projects.

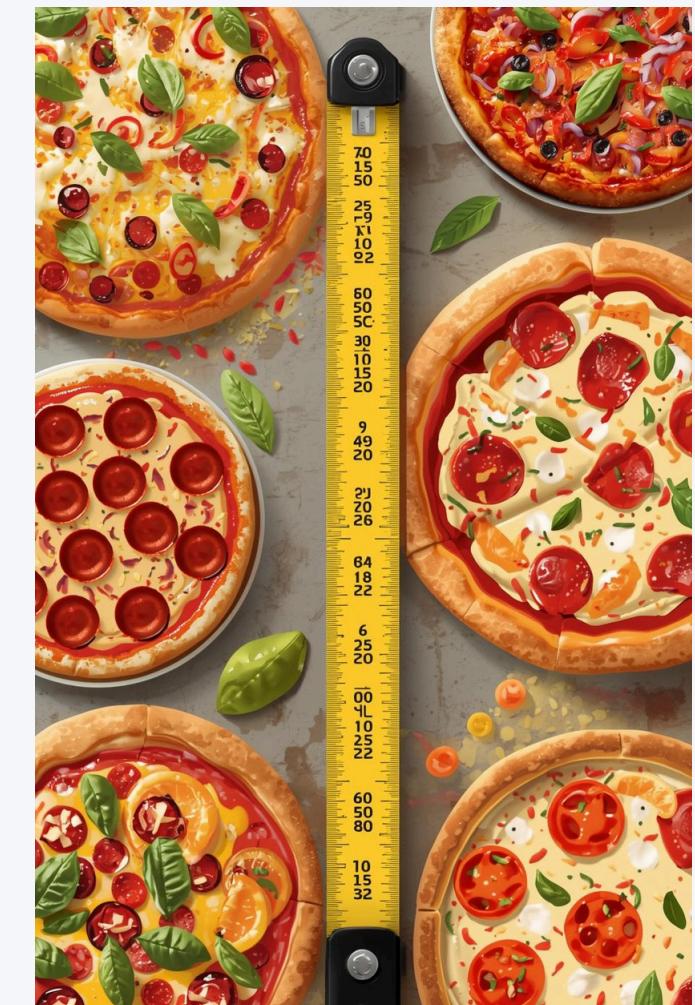
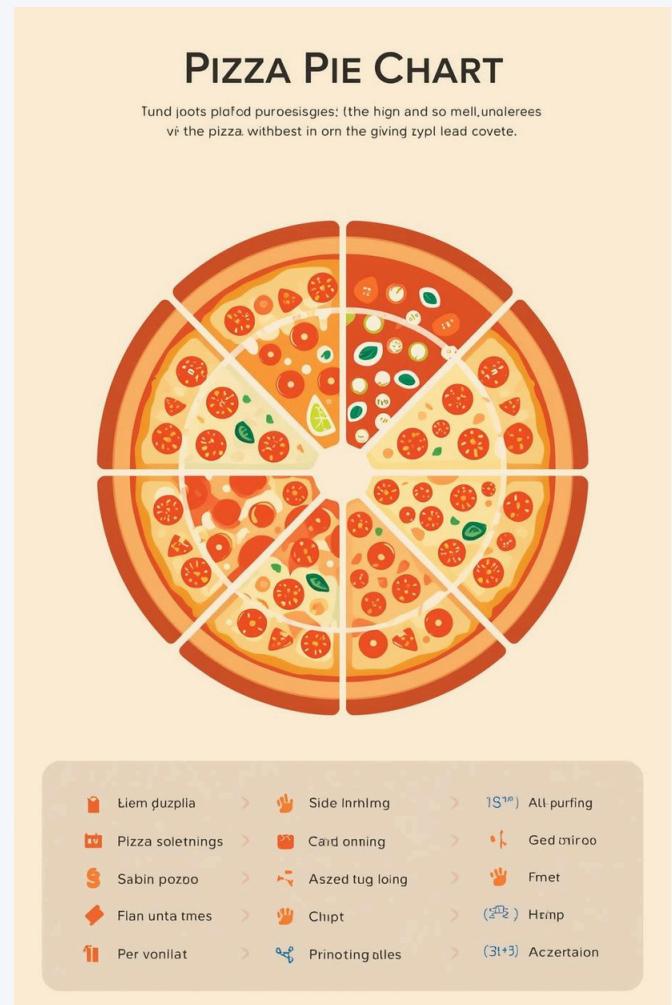


# *Dataset Description*

# Understanding the structure of the pizza sales dataset used for analysis

This project utilizes a **publicly available pizza sales dataset** that consists of several relational tables. The main tables include orders, order\_details, pizzas, and pizza\_types. Each table provides unique insights into the sales process, enabling comprehensive analysis of customer preferences and sales trends.





## Total Orders

Number of orders processed this period

## Total Revenue

Total income generated from pizza sales

## Highest Priced

Identifying the premium pizza option

## Common Size

Most frequently ordered pizza size

## Top 5 Pizzas

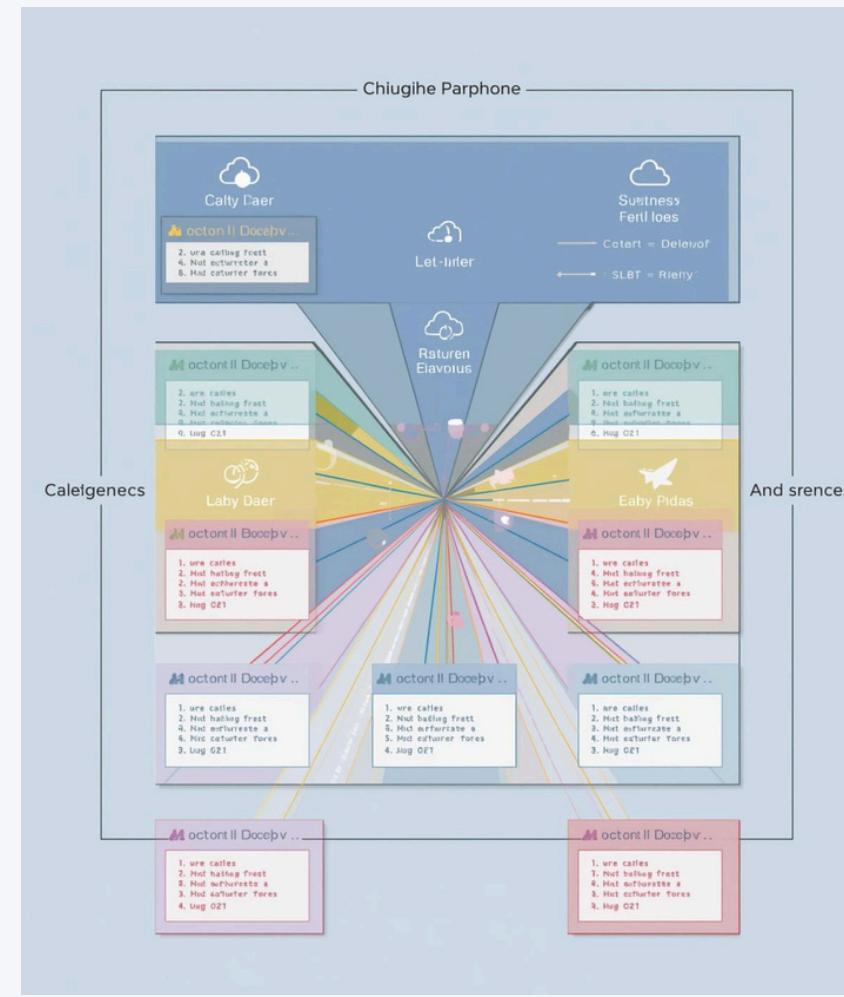
Most ordered pizzas by volume

# Key Takeaways from Basic Analysis

# Key SQL Techniques Used

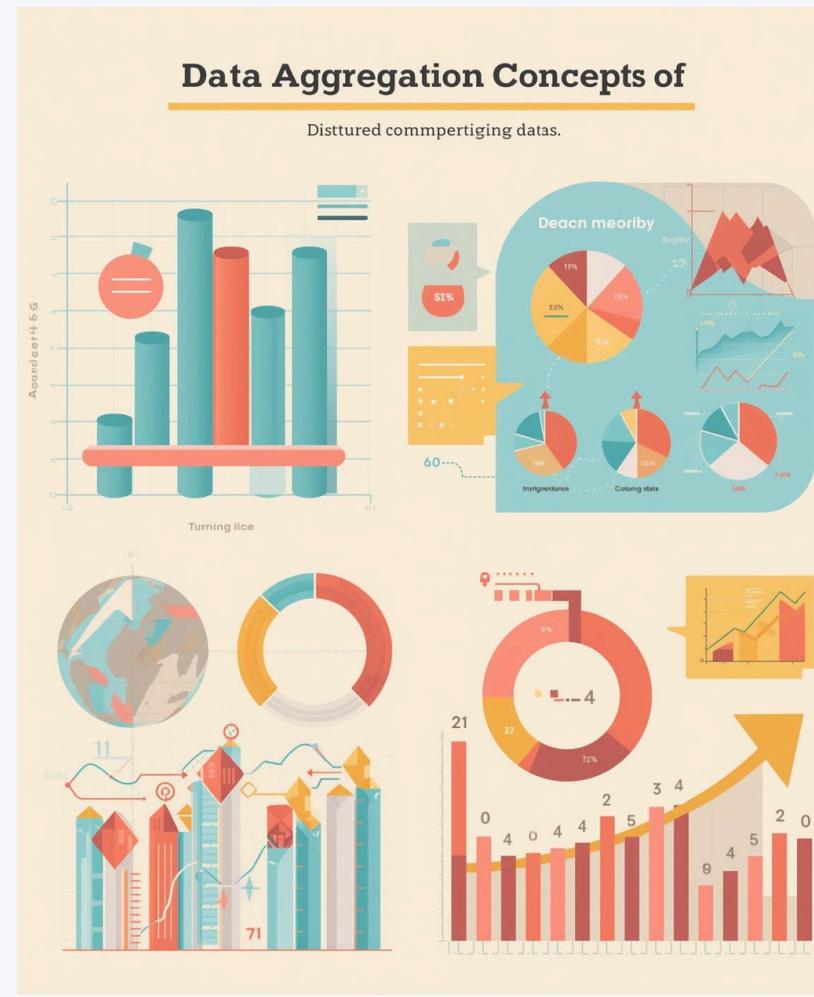
## Joins

Combining multiple tables for insights



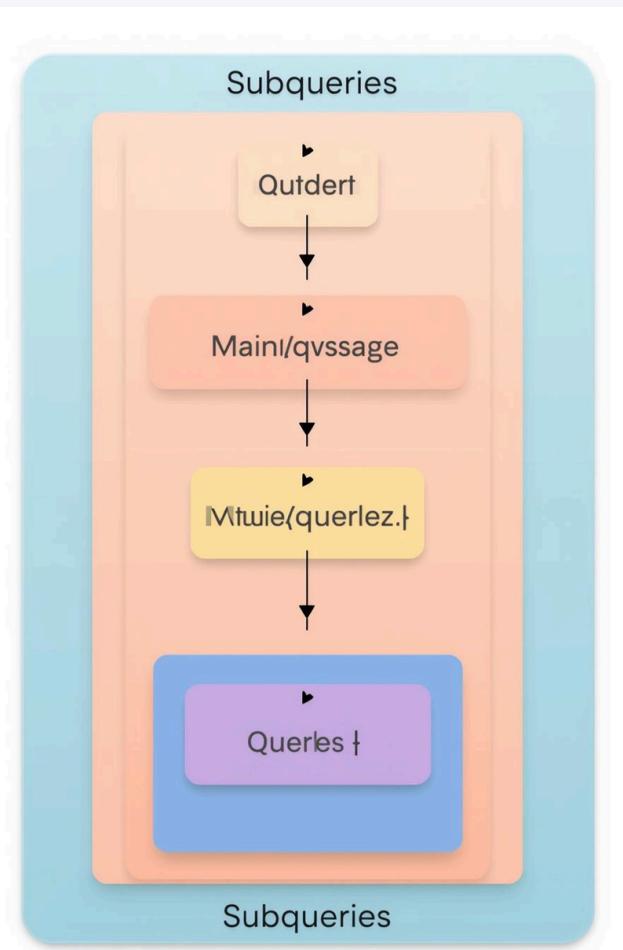
## Aggregations

Summarizing data with mathematical functions



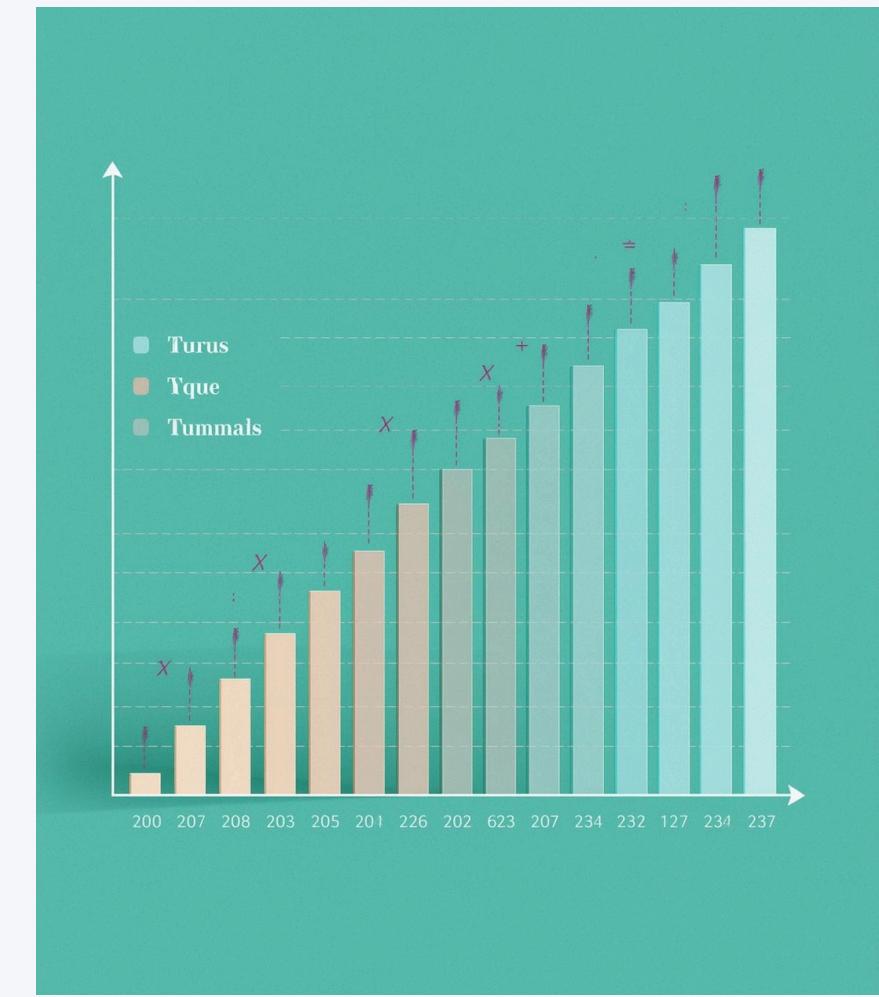
## Subqueries

Nested queries for advanced filtering



## Window Functions

Performing calculations across rows



# Total Orders Analysis

```
2      -- q1 Retrieve the total number of orders placed.  
3 •  SELECT  
4          COUNT(order_id) AS total_order  
5      FROM  
6          orders;  
7  
8
```

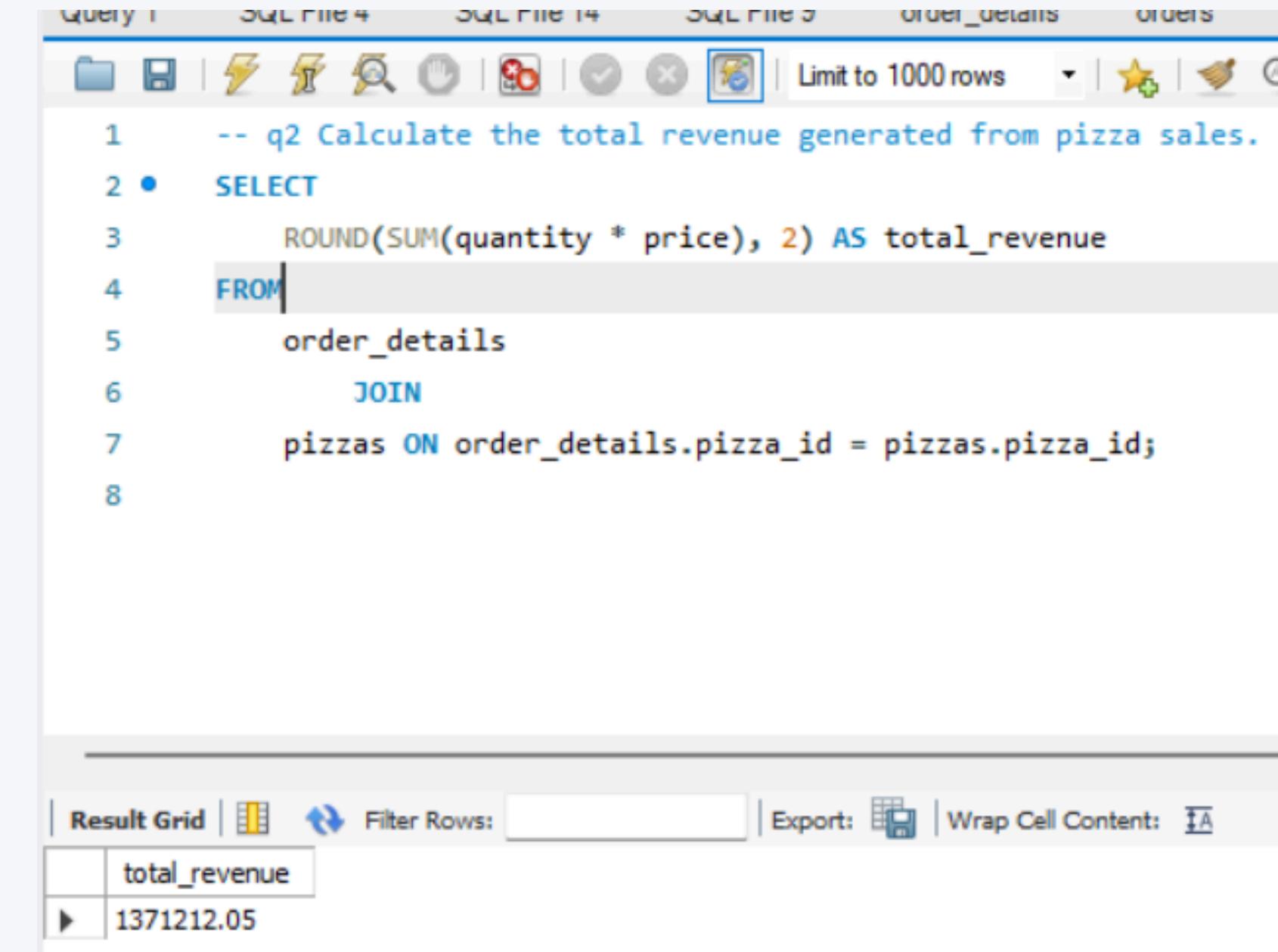
Result Grid		Filter Rows:	Export:	Wrap Cell Content	
total_order		21350			

## Total Number of Orders Placed

This query calculates the total number of unique orders placed by customers.



# Revenue Performance



The screenshot shows a MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 -- q2 Calculate the total revenue generated from pizza sales.
2 • SELECT
3     ROUND(SUM(quantity * price), 2) AS total_revenue
4 FROM
5     order_details
6     JOIN
7     pizzas ON order_details.pizza_id = pizzas.pizza_id;
8
```

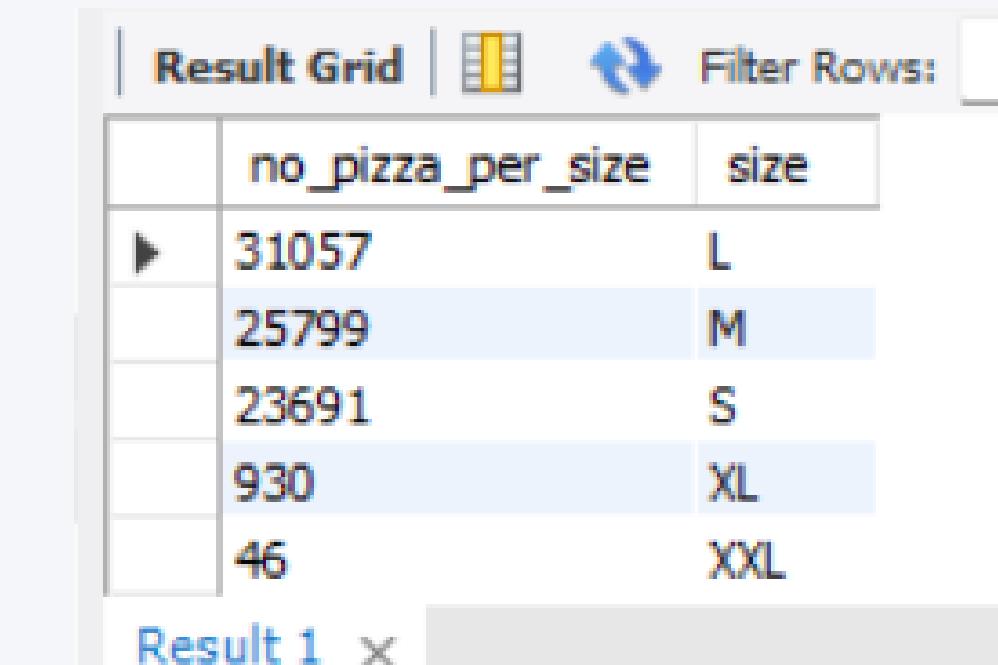
The result grid displays a single row with the column 'total\_revenue' containing the value '1371212.05'.

Total Revenue Generated from Pizza Sales  
Revenue is calculated by multiplying pizza price with quantity sold across all orders.



# Customer Preference Analysis

```
-- Identify the most common pizza sizes ordered.  
  
SELECT  
    COUNT(quantity) AS no_pizza_per_size, size  
FROM  
    order_details  
        JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY size  
ORDER BY COUNT(quantity) DESC;
```



The screenshot shows a MySQL Workbench result grid titled "Result Grid". The grid displays the results of the SQL query, which counts the number of pizzas ordered per size. The columns are labeled "no\_pizza\_per\_size" and "size". The data shows five rows: Large (L) with 31057, Medium (M) with 25799, Small (S) with 23691, Extra Large (XL) with 930, and XX Large (XXL) with 46. The "M" row is currently selected.

	no_pizza_per_size	size
▶	31057	L
	25799	M
	23691	S
	930	XL
	46	XXL

## Most Common Pizza Size Ordered

This query shows which pizza size is most frequently ordered by customers



# Product Demand Ranking

```
-- List the top 5 most ordered pizza types along with their quantities.  
SELECT  
    name, SUM(quantity) AS pizza_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 name  
ORDER BY pizza_quantity DESC  
LIMIT 5;
```

Result Grid | Filter Rows:

	name	pizza_quantity
▶	The Barbecue Chicken Pizza	4111
	The Classic Deluxe Pizza	4071
	The Pepperoni Pizza	4067
	The Hawaiian Pizza	4023
	The California Chicken Pizza	3971

Top 5 Most Ordered Pizza Types  
Pizza types are ranked based on total quantity ordered.

# Category Sales Distribution

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2 • SELECT
3     category, SUM(quantity) AS pizza_quantity
4  FROM
5      pizza_types
6      JOIN
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8      JOIN
9      order_details ON order_details.pizza_id = pizzas.pizza_id
10 GROUP BY category
11 ORDER BY pizza_quantity DESC;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	category	pizza_quantity
▶	Classic	24908
	Supreme	20111
	Veggie	19606
	Chicken	18479

## Total Quantity Sold per Pizza Category

This analysis shows how pizza sales are distributed across different categories.



# Time-Based Order Behavior

```
1      -- Determine the distribution of orders by hour of the day.  
2 •  SELECT  
3          HOUR(order_time) AS hour, COUNT(order_id)  
4      FROM  
5          orders  
6      GROUP BY hour  
7      ORDER BY COUNT(order_id) DESC limit 5;  
8
```

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch row

	hour	COUNT(order_id)
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009

## Orders Distribution by Hour of the Day

This query helps identify peak ordering hours based on customer behavior.



# Category Contribution Analysis

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.  
2 • SELECT  
3      category, COUNT(name) AS no_of_pizza_per_category  
4  FROM  
5      pizza_types  
6  GROUP BY category  
7  ORDER BY no_of_pizza_per_category DESC;  
8
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

category	no_of_pizza_per_category
Supreme	9
Veggie	9
Classic	8
Chicken	6

## Category-Wise Pizza Distribution

This query analyzes how different pizza categories contribute to overall sales.



# Daily Sales Trend

```
1      -- Group the orders by date and calculate the average number of pizzas ordered per day.
2 •  SELECT
3      ROUND(AVG(quantity), 0) AS no_of_pizza_sales_perday
4  FROM
5  (
6      SELECT
7          order_date, SUM(quantity) AS quantity
8      FROM
9          order_details
10     JOIN orders ON order_details.order_id = orders.order_id
11     GROUP BY order_date) AS data;
```

Result Grid    Filter Rows: <input type="text"/> Export:  Wrap Cell Content:	
	no_of_pizza_sales_perday
▶	241

Average Number of Pizzas Ordered per Day  
Daily total pizza sales are calculated first, and then the average is derived across all days.



# Revenue Ranking

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2 • SELECT
3      name, SUM(price * quantity) AS revenue
4  FROM
5      pizzas
6      JOIN
7      pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10     GROUP BY name
11     ORDER BY revenue DESC
12     LIMIT 5;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

name	revenue
The Barbecue Chicken Pizza	72223.25
The Thai Chicken Pizza	71882.75
The California Chicken Pizza	69394.25
The Classic Deluxe Pizza	63341
The Spicy Italian Pizza	58770.25

Top 3 Pizza Types Based on Revenue  
Pizza types are ranked based on total revenue generated.



# Total Orders Analysis

```
2      -- q1 Retrieve the total number of orders placed.  
3 •  SELECT  
4          COUNT(order_id) AS total_order  
5      FROM  
6          orders;  
7  
8
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content	
total_order		21350			

## Total Number of Orders Placed

This query calculates the total number of unique orders placed by customers.



# Revenue Contribution Analysis

```
1 -- Calculate the percentage contribution of each pizza type to total revenue
2 • select category,round(sum(quantity*price)/(SELECT
3     ROUND(SUM(quantity * price), 2) as total_sales
4
5     FROM
6         order_details
7     JOIN
8         pizzas ON order_details.pizza_id = pizzas.pizza_id)*100,2)as revenue
9     from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
10    join order_details
11      on order_details.pizza_id=pizzas.pizza_id
12      group by category order by revenue desc;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

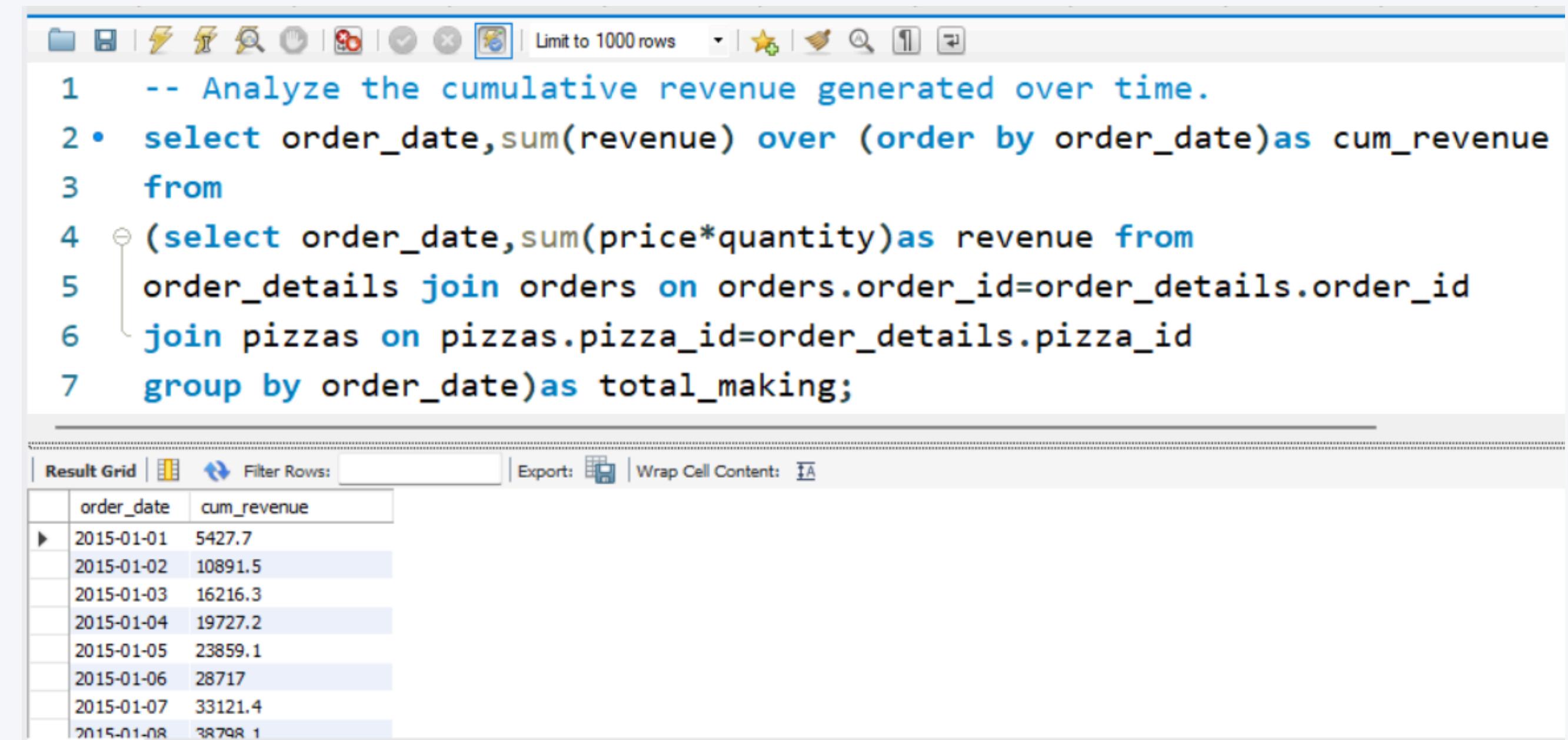
category	revenue
Classic	26.86
Supreme	25.49
Chicken	23.88
Veggie	23.77

## Percentage Contribution of Each Pizza Type to Total Revenue

This analysis shows how much each pizza type contributes to overall revenue.



# Growth Trend Analysis



The screenshot shows a database query interface with the following SQL code:

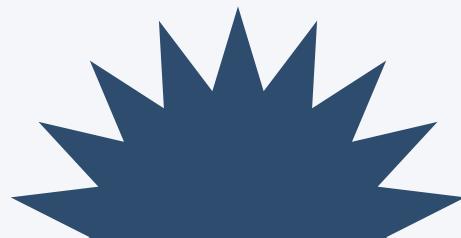
```
1 -- Analyze the cumulative revenue generated over time.
2 • select order_date,sum(revenue) over (order by order_date)as cum_revenue
3   from
4   (select order_date,sum(price*quantity)as revenue from
5   order_details join orders on orders.order_id=order_details.order_id
6   join pizzas on pizzas.pizza_id=order_details.pizza_id
7   group by order_date)as total_making;
```

The results grid displays the following data:

	order_date	cum_revenue
▶	2015-01-01	5427.7
	2015-01-02	10891.5
	2015-01-03	16216.3
	2015-01-04	19727.2
	2015-01-05	23859.1
	2015-01-06	28717
	2015-01-07	33121.4
	2015-01-08	38798.1

## Cumulative Revenue Over Time

A running total is calculated to analyze revenue growth over time.



# Category-Wise Performance Leaders

name	revenue
The Barbecue Chicken Pizza	72223.25
The Thai Chicken Pizza	71882.75
The California Chicken Pizza	69394.25
The Classic Deluxe Pizza	63341
The Spicy Italian Pizza	58770.25
The Italian Supreme Pizza	56491.25
The Four Cheese Pizza	53861.649999998335
The Hawaiian Pizza	53617
The Sicilian Pizza	51451.5
The Pepperoni Pizza	50757.75
The Mexicana Pizza	44781.75
The Five Cheese Pizza	44363

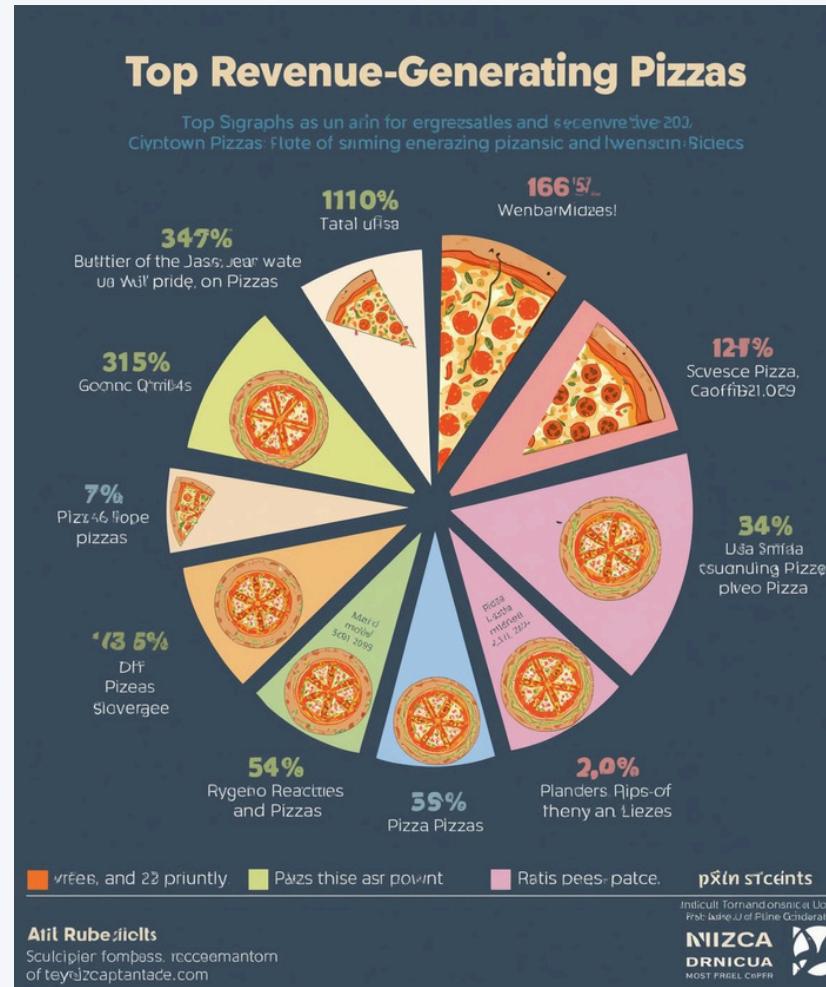
Top 3 Pizzas by Revenue for Each Category  
Window functions are used to rank pizzas within each category based on revenue.



# Key Insights from Analysis

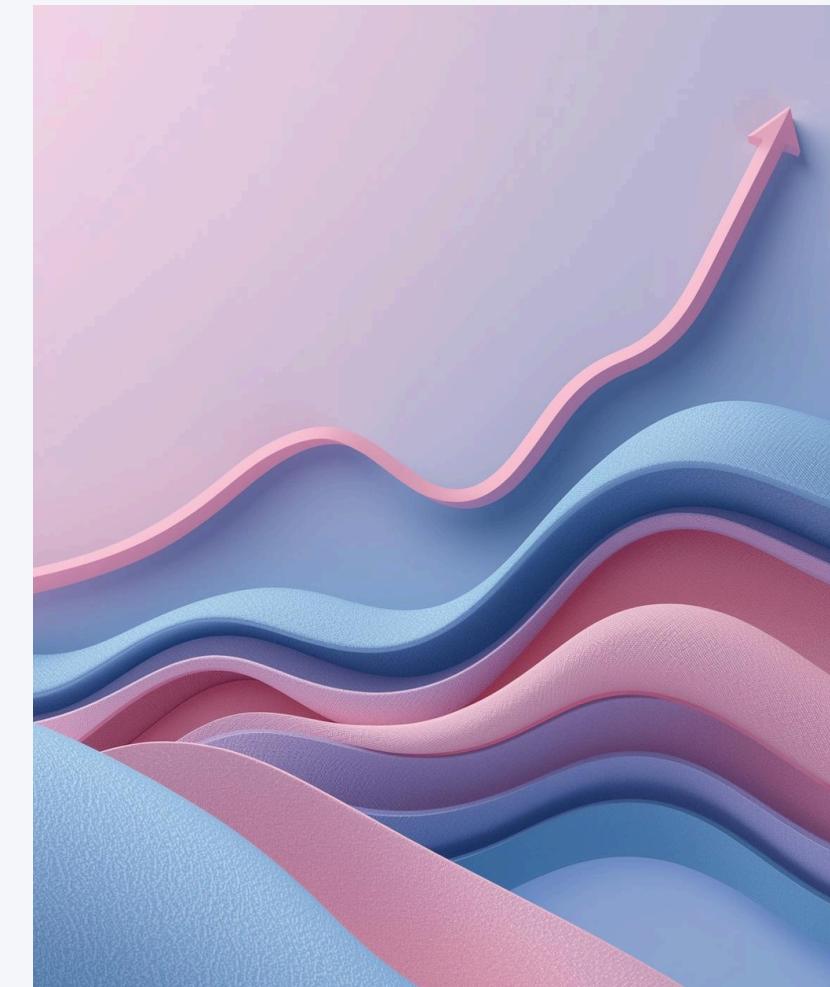
## Top Revenue Pizzas

Identifying best-selling pizzas for profit



## Sales Trends

Understanding sales patterns over time



## Marketing Insights

Strategies to enhance customer engagement



## Inventory Management

Optimizing pizza stock based on trends



# *Key Learnings*

---

## Insights Gained from Analyzing Pizza Sales Data with SQL

Throughout this project, I have **enhanced my SQL skills** significantly. By working with real-world data, I gained practical experience in data analysis, learned effective querying techniques, and discovered how data drives business decisions. These insights will be invaluable as I pursue a career in data analytics.



Thank you for your  
attention to this  
presentation

*Created by Divya Joshi.*