

# PIZZA SALES ANALYSIS

SQL  
PROJECT

BY - Susmita Manna



# INTRODUCTION



Hi , I am Susmita Manna!

In this project ,  
I have utilized MySQL to solve queries related to Pizza Sales.  
Please go through these slides.

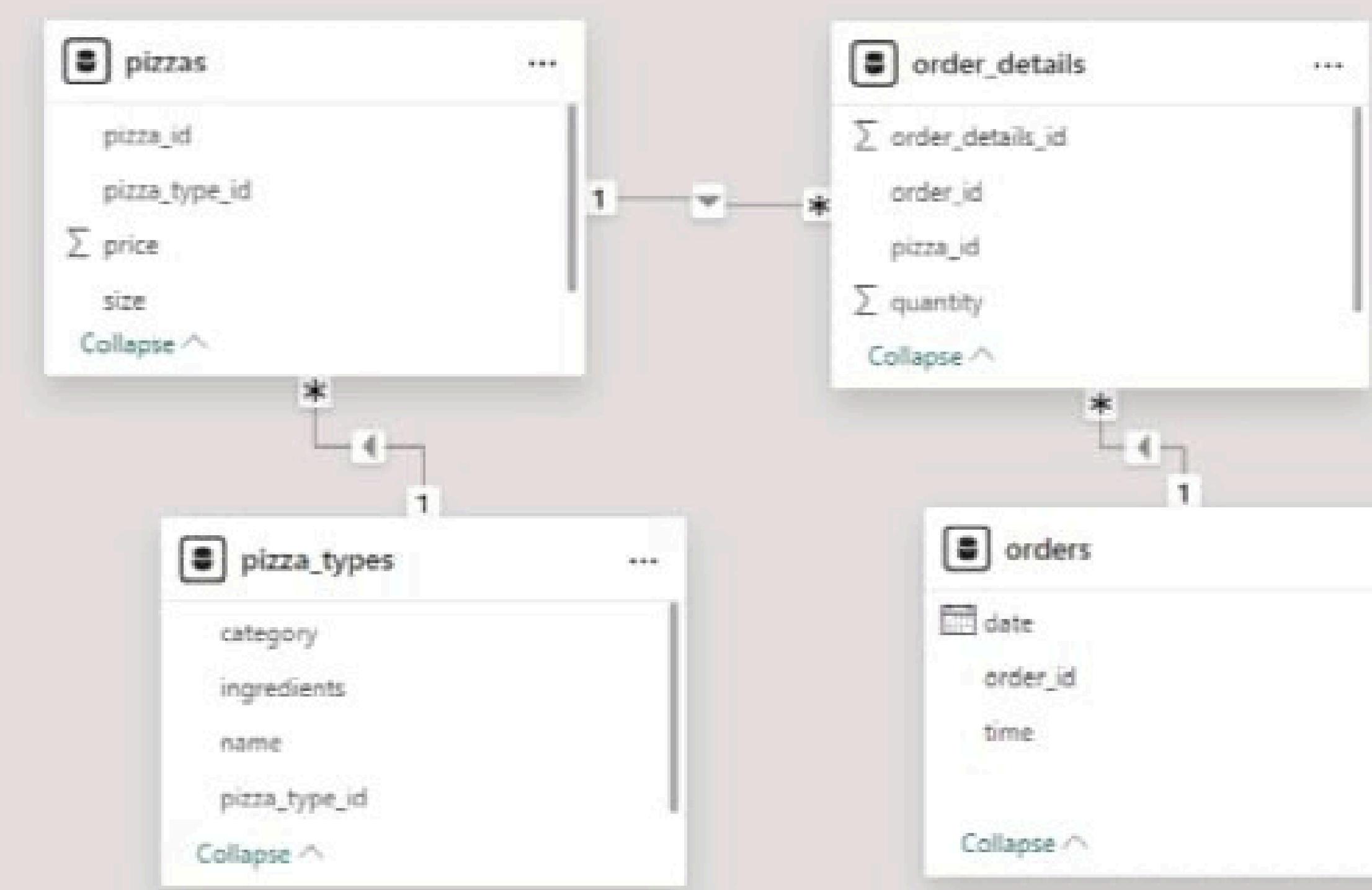
# OBJECTIVE

**The objective of the project is to analyze pizza sales data to identify trends and provide actionable insights that can help to increase sales and aim to uncover key metrics and patterns within the sales data by leveraging SQL queries in MYSQL**



# MODEL VIEW

Representing relationships between Tables:



# PROBLEM STATEMENTS

- 1** Retrieve the total no of ordered placed.
- 2** Calculate the total revenue generated from pizza sales
- 3** Identify the highest priced pizza
- 4** Identify the most common pizza size ordered.
- 5** List the top 5 ordered pizza types along with their quantities
- 6** Join the necessary tables to find the total quantity of each pizza category ordered.
- 7** Determine the distribution of orders by hour of the day
- 8** Join relevant tables to find the category wise distribution of pizzas.
- 9** Group the orders by date and calculate the average number of pizzas ordered per day.
- 10** Determine the top 3 most ordered pizza types based on revenue
- 11** Calculate the percentage contribution of each pizza type to total revenue
- 12** Analyze the cumulative revenue generated over time
- 13** Determine top 3 most ordered pizza types based on revenue for each pizza category

**1**

# Retrieve the total no of ordered placed.

Query

```
2 • SELECT * FROM orders;  
3 • SELECT COUNT(*) AS total_no_of_orders  
4 FROM orders;
```

Output

Result Grid	
	total_no_of_orders
▶	21350

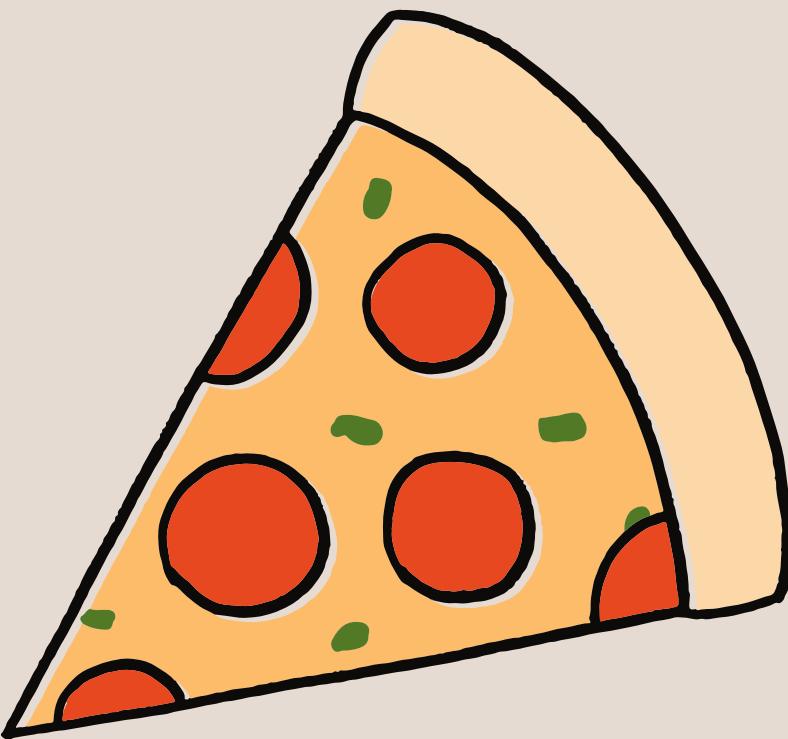
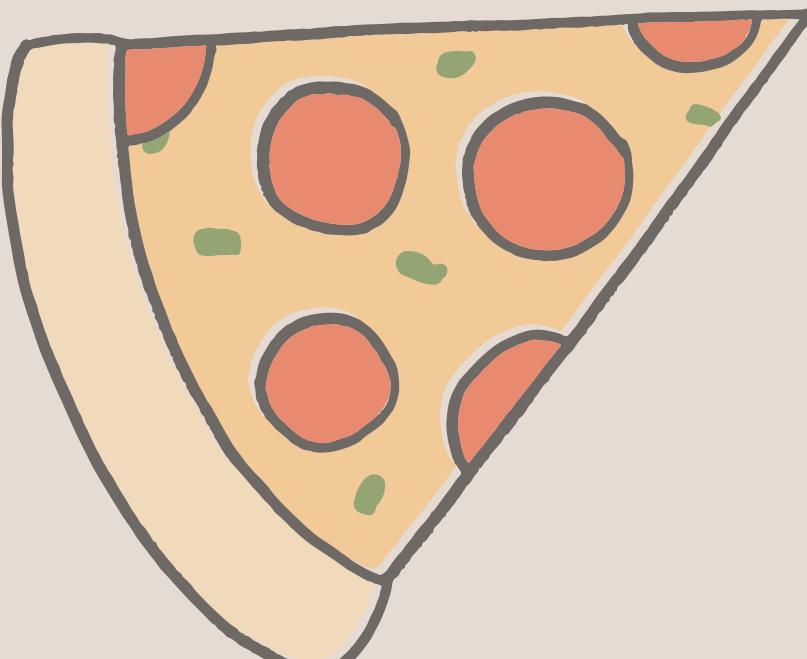


2

# Calculate the total revenue generated from pizza sales

Query

```
3 • SELECT
4     ROUND(SUM(ord.quantity * pz.price), 2) AS total_revenue_by_pizza_sales
5 FROM
6     orders_details AS ord
7     JOIN
8         pizzas AS pz ON ord.pizza_id = pz.pizza_id
9
```



Output

Result Grid	Filter Rows:
total_revenue_by_pizza_sales	
817860.05	

3

# Identify the highest priced pizza

Query

```
3 • SELECT * FROM pizza_types;  
4  
5 • SELECT  
6     pizza_types.name, pizzas.price  
7  FROM  
8      pizzas  
9      JOIN  
10     pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
11 ORDER BY pizzas.price DESC  
12 LIMIT 1;
```



Output

Result Grid		Filter Rows:
	name	price
	The Greek Pizza	35.95

# 4 Identify the most common pizza size ordered.

## Query

```
2 • SELECT * FROM pizzas;  
3 • SELECT * FROM orders_details;  
4  
5 • SELECT  
6   pizzas.size  
7   ,COUNT(orders_details.order_details_id) AS total_order_count  
8   FROM pizzas  
9   JOIN orders_details  
10  ON  
11    pizzas.pizza_id = orders_details.pizza_id  
12  GROUP BY pizzas.size  
13  ORDER BY total_order_count DESC;
```



## Output

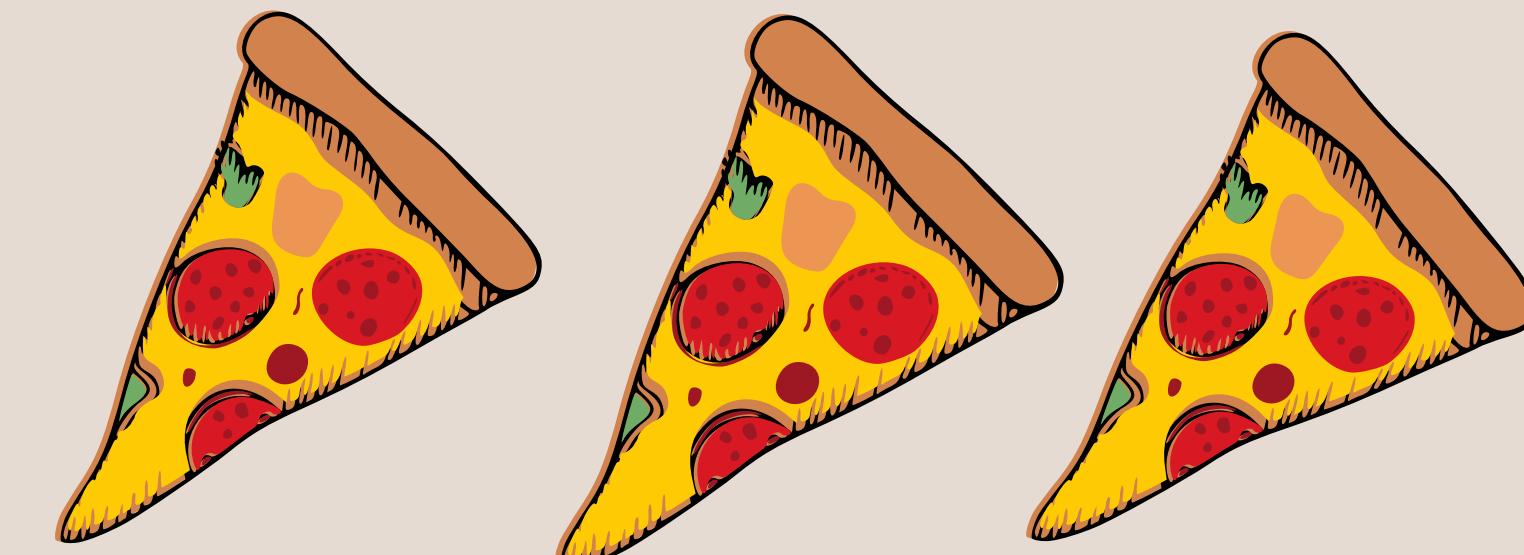
Result Grid		Filter Rows:
	size	total_order_count
	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

5

# List the top 5 ordered pizza types along with their quantities

## Query

```
4 • SELECT pizza_types.name  
      , SUM(orders_details.quantity) AS total_quantity  
  FROM pizza_types  
 JOIN pizzas  
   ON  
pizza_types.pizza_type_id = pizzas.pizza_type_id  
JOIN orders_details  
   ON  
pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY total_quantity DESC  
Limit 5;
```



## Output

Result Grid		Filter Rows:
	name	total_quantity
	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

6

Join the necessary tables to find the total quantity of each pizza category ordered.

## Query

```
2 •  SELECT * FROM pizza_types;
3 •  SELECT * FROM orders_details;
4 •  SELECT * FROM pizzas;
5
6 •  SELECT pizza_types.category
7      , sum(orders_details.quantity) AS quantity
8  FROM pizza_types
9  JOIN pizzas
10 ON
11 pizza_types.pizza_type_id = pizzas.pizza_type_id
12 JOIN orders_details
13 ON
14 pizzas.pizza_id = orders_details.pizza_id
15 GROUP BY pizza_types.category
16 ORDER BY quantity DESC;
```



## Output

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



# Determine the distribution of orders by hour of the day

Query

```
2 • SELECT HOUR(order_time) AS hour  
3      , COUNT(order_id) AS order_count  
4  FROM orders  
5 GROUP BY hour;
```



Output

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

8

# Join relevant tables to find the category wise distribution of pizzas.

## Query

```
2 • SELECT * FROM pizza_types;
3 • SELECT * FROM orders_details;
4 • SELECT * FROM pizzas;
5
6 • SELECT pizza_types.category
    , sum(orders_details.quantity) AS quantity
8 FROM pizza_types
9 JOIN pizzas
10 ON
11 pizza_types.pizza_type_id = pizzas.pizza_type_id
12 JOIN orders_details
13 ON
14 pizzas.pizza_id = orders_details.pizza_id
15 GROUP BY pizza_types.category
16 ORDER BY quantity DESC;
```



## Output

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

9 Group the orders by date and calculate the average number of pizzas ordered per day.

## Query

```
3 • SELECT ROUND(avg(quantity),0) FROM
4 (
5   SELECT orders.order_date
6       , SUM(orders_details.quantity) AS quantity
7   FROM orders
8   JOIN orders_details
9   ON
10  orders.order_id = orders_details.order_id
11  GROUP BY orders.order_date
12 ) AS table_1
```



## Output

Result Grid	Filter Rows:
ROUND(avg(quantity),0)	
138	

# 10 Determine the top 3 most ordered pizza types based on revenue

## Query

```
3     , ROUND(SUM(orders_details.quantity * pizzas.price),2)AS total_revenue
4 FROM pizza_types
5 JOIN pizzas
6 ON |
7 pizza_types.pizza_type_id = pizzas.pizza_type_id
8 JOIN orders_details
9 ON
10 pizzas.pizza_id = orders_details.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY total_revenue DESC
13 LIMIT 3;
```



## Output

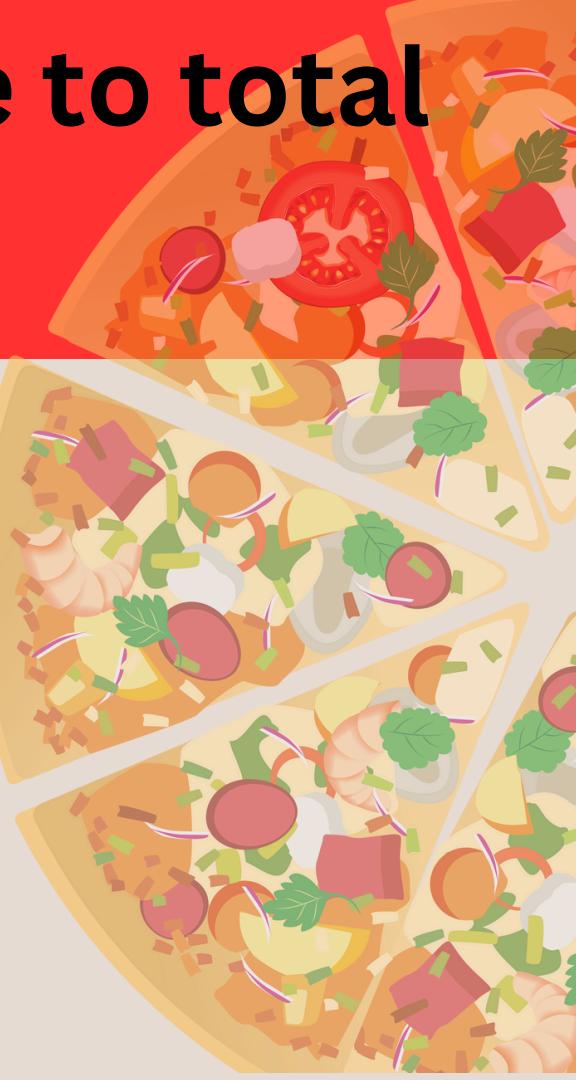
	name	total_revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

11

# Calculate the percentage contribution of each pizza type to total revenue

## Query

```
SELECT
    pizza_types.category
    ,ROUND(SUM(orders_details.quantity * pizzas.price)
        / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price), 2) AS total_sales
        FROM
            orders_details
        JOIN
            pizzas ON orders_details.pizza_id = pizzas.pizza_id)*100 ,2 ) AS revenue
FROM pizza_types
JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN orders_details
ON
    orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```



## Output

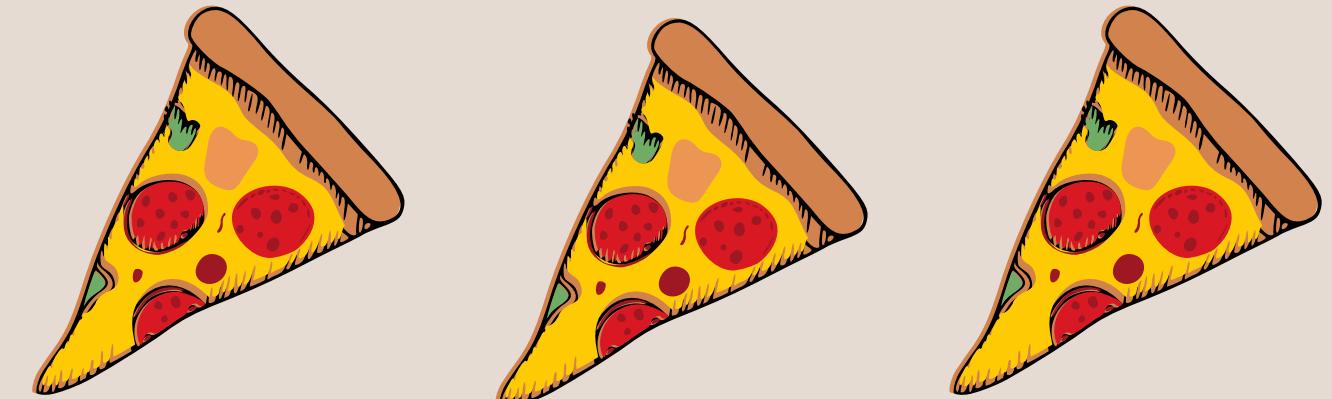
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12

# Analyze the cumulative revenue generated over time

## Query

```
• SELECT order_date  
      , SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue  
    FROM  
    (   
        SELECT orders.order_date  
              ,SUM(orders_details.quantity * pizzas.price) AS revenue  
        FROM orders_details  
      JOIN orders  
      ON  
        orders_details.order_id = orders.order_id  
      JOIN pizzas  
      ON  
        orders_details.pizza_id = pizzas.pizza_id  
      GROUP BY orders.order_date  
    ) AS table_1  
  ORDER BY cumulative_revenue DESC;
```



## Output

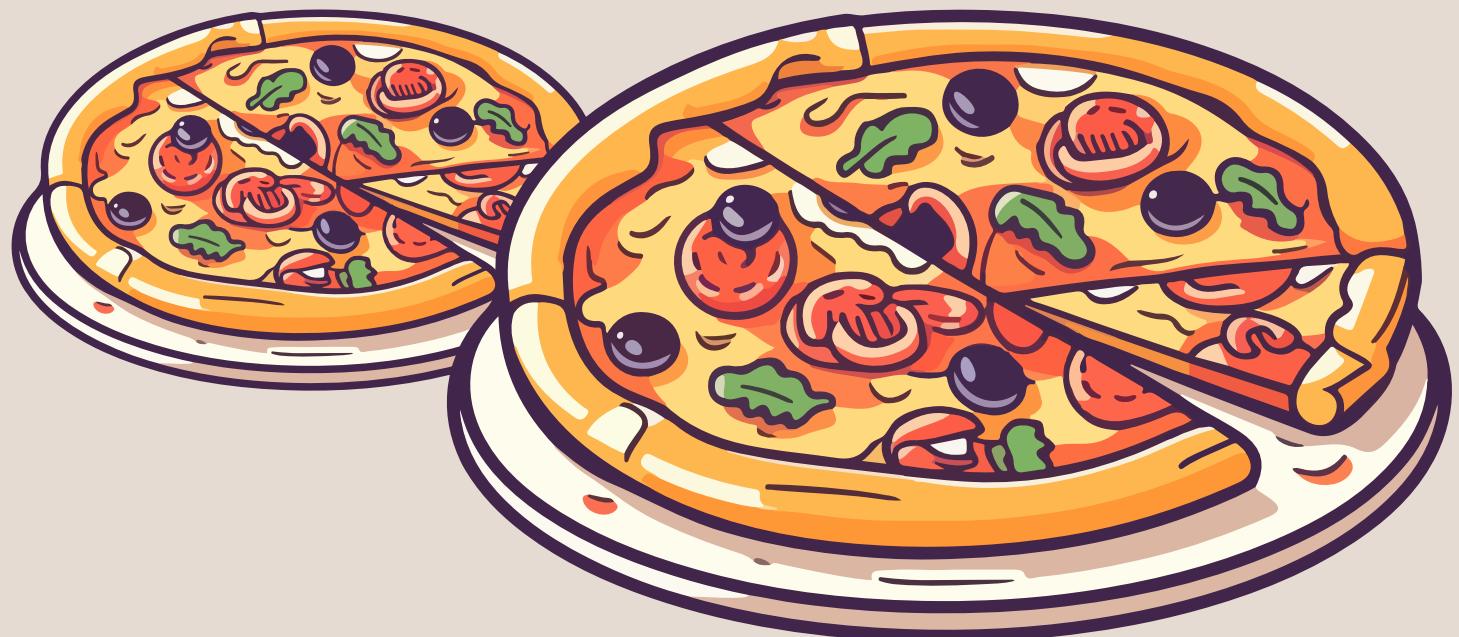
	order_date	cumulative_revenue
▶	2015-12-31	817860.05
	2015-12-30	814944.05
	2015-12-29	813606.25
	2015-12-28	812253
	2015-12-27	810615.8
	2015-12-26	809196.8
	2015-12-24	807553.75

13

# Determine top 3 most ordered pizza types based on revenue for each pizza category

## Query

```
3 •   SELECT name, revenue
4 -   FROM (
5     SELECT name, category, revenue,
6       RANK() OVER(PARTITION BY category
7                   ORDER BY revenue) AS pizza_rank
8   FROM
9   (
10    SELECT
11      pizza_types.name
12      , pizza_types.category
13      ,SUM((orders_details.quantity) * pizzas.price) AS revenue
14  FROM
15    pizza_types
16      JOIN
17      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
18      JOIN
19      orders_details ON pizzas.pizza_id = orders_details.pizza_id
20  GROUP BY pizza_types.name , pizza_types.category
21 ) AS table_1
22 ) AS table_2
23 LIMIT 3;
```



## Output

	name	revenue
▶	The Chicken Pesto Pizza	16701.75
▶	The Chicken Alfredo Pizza	16900.25
▶	The Southwest Chicken Pizza	34705.75

# THANK YOU

