

SQL PROJECT

PIZZA SALES





Greetings!!

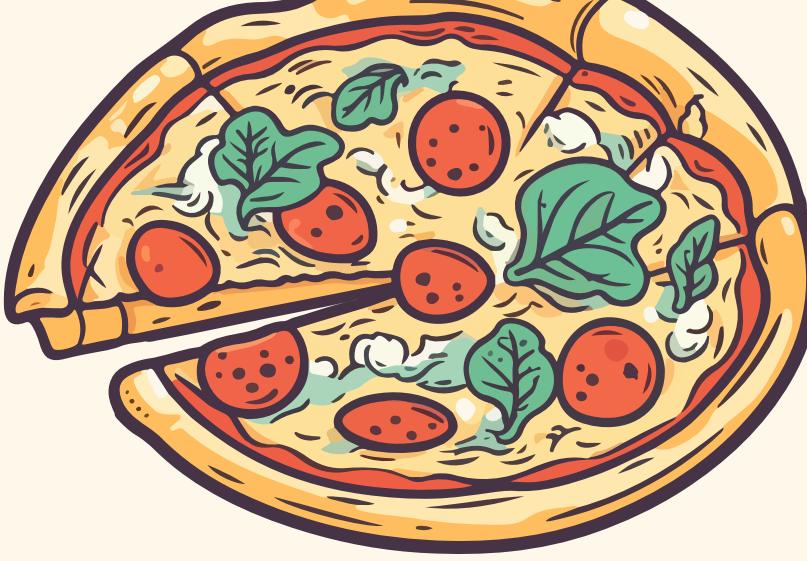
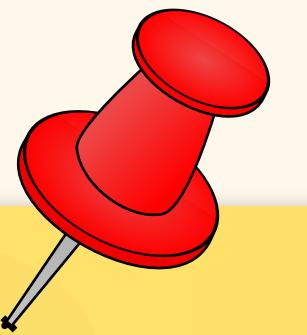
I am delighted to present to you a comprehensive SQL project aimed at evaluating the sales performance of a pizza store. In this project, I have meticulously analyzed four datasets providing various sales information about the store.

The first dataset contains order details, offering insights into order IDs, pizza IDs, quantities ordered, and corresponding pizza order IDs. The second dataset focuses on orders, including dates, order IDs, and timestamps. The third dataset delineates pizza types, showcasing categories, ingredients, names, and unique pizza type IDs. Lastly, the Pizza dataset elucidates pizza prices, sizes, orders, and pizza IDs.

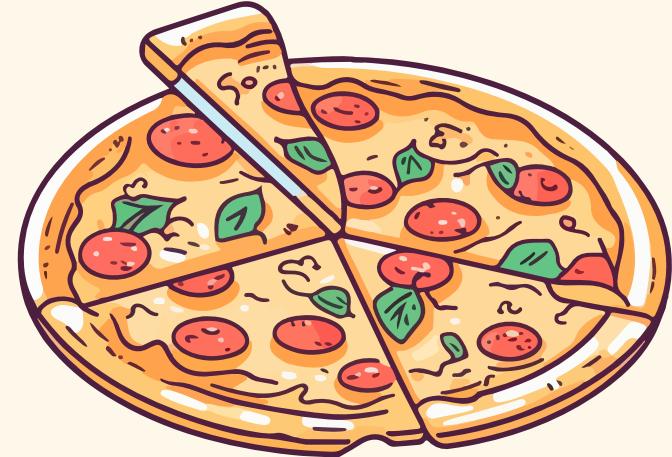
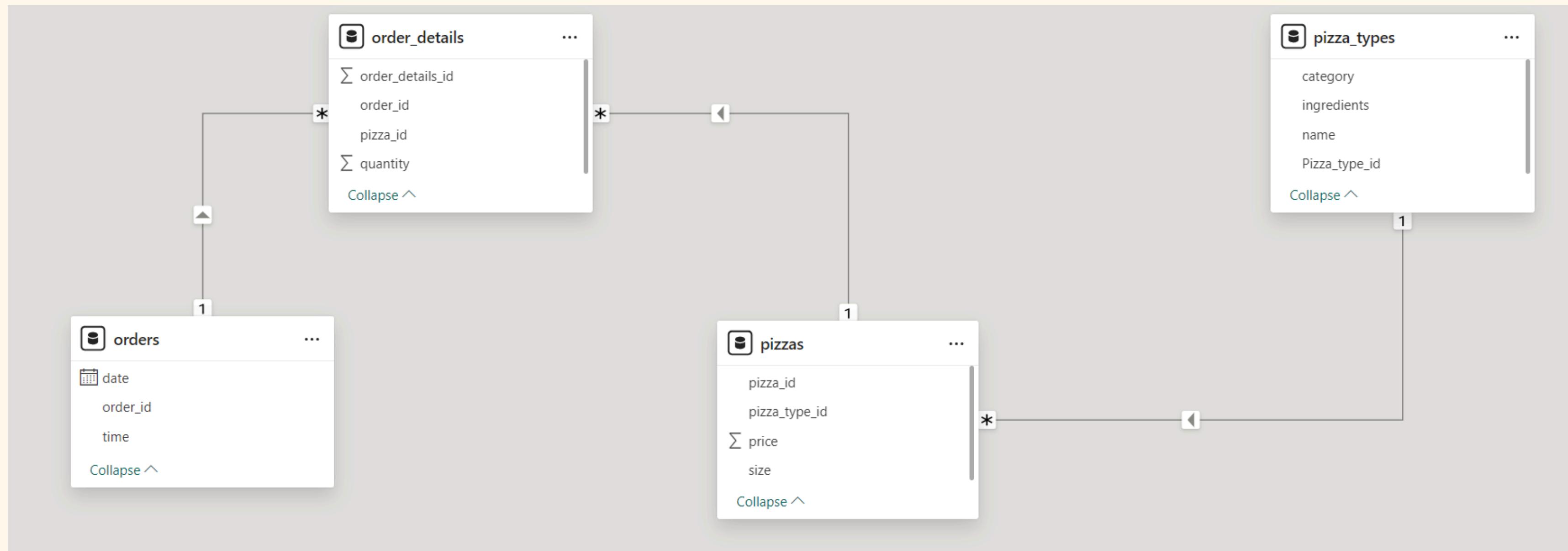
To derive meaningful insights from these datasets, I have employed a range of SQL queries, spanning from basic to advanced techniques. These include joins, group by, order by, rank, subqueries, and more.

Through this project, we aim to unveil valuable patterns and trends within the data, empowering the pizza store to make informed decisions and optimize its sales strategies.

Thank you for your attention, and I look forward to sharing the results of this analysis with you.



Database Schema





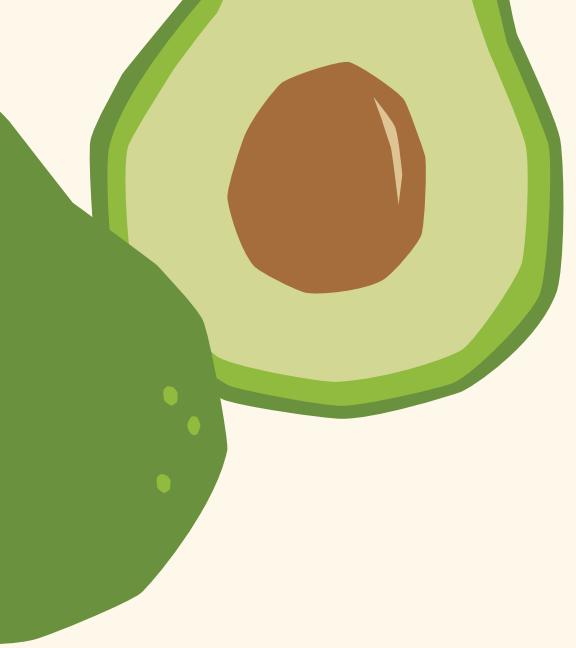
Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_order  
FROM  
    orders;
```



Result Grid	
	total_order
▶	21350

A substantial total of 21,350 orders have been integrated into the system.

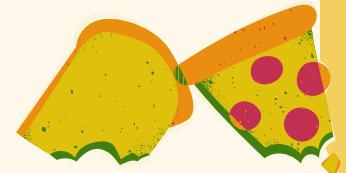


Calculate the total revenue generated from pizza sales.

- **SELECT**

```
    ROUND(SUM(order_details.quantity * pizzas.price),  
          2) AS TS  
  
FROM  
  
    order_details  
    INNER JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	TS
▶	817860.05



A significant total revenue of \$817,860.05 has been generated from pizza sales.



Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizzas  
        INNER JOIN  
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

Result Grid | Filter Rows:

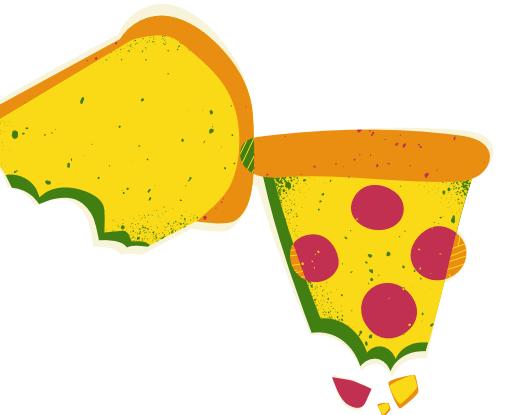
	name	price
▶	The Greek Pizza	35.95

The Greek pizza holds the highest price at \$35.95 per serving.

Identify the most common pizza size ordered

```
SELECT
    pizzas.size, COUNT(order_details.order_details_id) AS oredr
FROM
    pizzas
        INNER JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(order_details.order_details_id) DESC
LIMIT 1;
```

	size	oredr
▶	L	18526



The large-sized pizza was the most ordered by customers, totaling 18,526 orders.



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



Result Grid		Filter Rows:
	names	total
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

```
SELECT
    pizza_types.name AS names,
    SUM(order_details.quantity) AS total
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        INNER JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY names
ORDER BY total DESC
LIMIT 5;
```

Therefore, the five most popularly ordered pizzas are Classic, Barbecue, Hawaiian, Pepperoni, and Thai Chicken, each accompanied by their corresponding order numbers.

The total quantity of each pizza category ordered

Result Grid | Filter

	category	total_Q
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

```
SELECT  
    e2.category, SUM(e1.quantity)  
FROM  
    pizza_types e2  
    INNER JOIN  
    pizzas e3 ON e3.pizza_type_id = e2.pizza_type_id  
    INNER JOIN  
    order_details e1 ON e1.pizza_id = e3.pizza_id  
GROUP BY e2.category;
```

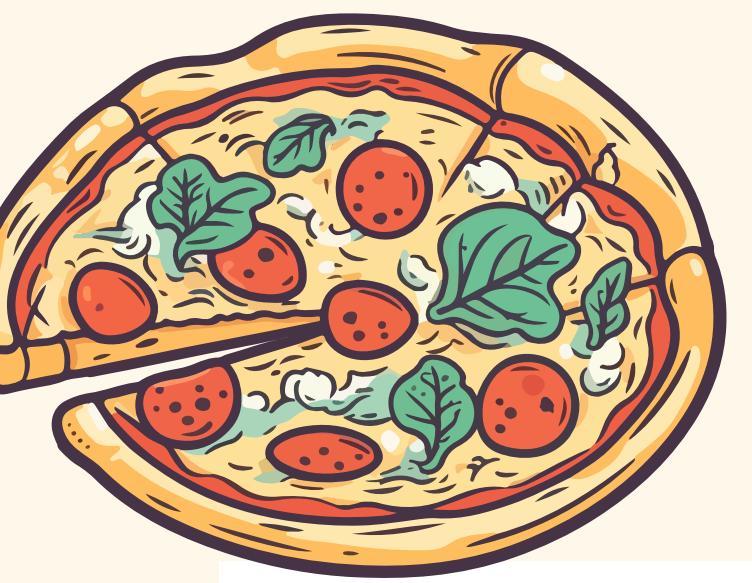
Therefore, the Classic pizza category has recorded the highest number of orders, totaling 14,888 units.

Distribution of orders by hour of the day

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS count_order  
FROM  
    orders  
GROUP BY hour;
```

	hour	count_order
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920

The distribution of orders per hour has been elucidated, noting the peak occurrence of 1231 pizza orders.



Category-wise distribution of pizzas

```
SELECT  
    category, COUNT(name) AS total_pizza  
FROM  
    pizza_types  
GROUP BY category  
ORDER BY total_pizza DESC;
```

Result Grid | Filter Rows

	category	total_pizza
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

The categorization of pizza reveals that the Supreme and Veggie categories have the highest distribution.



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

```
SELECT  
    ROUND(AVG(quantity_sum), 2) AS avg_ordered_pizzas  
FROM  
(SELECT  
    orders.order_date,  
    SUM(order_details.quantity) AS quantity_sum  
FROM  
    orders  
INNER JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS per_day_order;
```

Result Grid	
	avg_ordered_pizzas
▶	138.47



Approximately 139 pizzas are ordered per day.

Determine the top 3 most ordered pizza types based on revenue

Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

```
SELECT E1.name, SUM(E2.quantity * E3.price) AS revenue
FROM pizza_types E1
INNER JOIN pizzas E3 ON E3.pizza_type_id = E1.pizza_type_id
INNER JOIN order_details E2 ON E2.pizza_id = E3.pizza_id
GROUP BY E1.name
ORDER BY revenue DESC
LIMIT 3;
```

The most popular pizza choices based on revenue are Thai Chicken, Barbecue Chicken, and California Chicken.

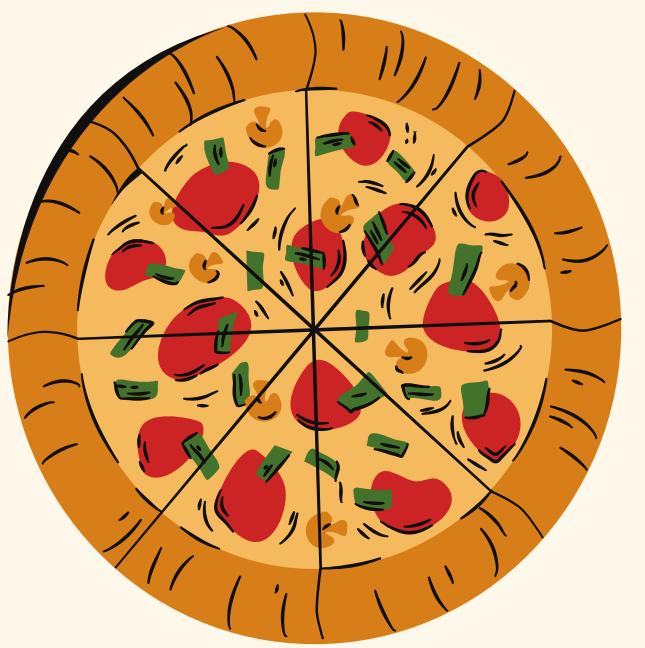
Calculate the percentage contribution of each pizza type to total revenue

Result Grid | Filter Rows:

	category	pizza_revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

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

Classic pizzas represent the largest share of total revenue, accounting for 26.91%.



Analyze the cumulative revenue generated over time

```
• SELECT order_date, ROUND(SUM(revenue) OVER(ORDER BY order_date),2) AS cumulative_revenue  
FROM  
(SELECT E2.order_date, SUM(E1.quantity*E3.price) AS revenue  
FROM order_details E1  
INNER JOIN pizzas E3  
ON E1.pizza_id=E3.pizza_id  
INNER JOIN orders E2  
ON E2.order_id=E1.order_id  
GROUP BY E2.order_date) AS date_wise_revenue  
ORDER BY cumulative_revenue DESC;
```

Result Grid		
	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
	2015-12-23	805415.9
	2015-12-22	803171.6
	2015-12-21	801288.65
	2015-12-20	799187.95
	2015-12-19	797083.05
	2015-12-18	794778.85
	2015-12-17	791892.55
	2015-12-16	790011.8
	2015-12-15	787777
	2015-12-14	785389.55
	2015-12-13	783216.95

The table reveals the cumulative revenue over time categorized by date.

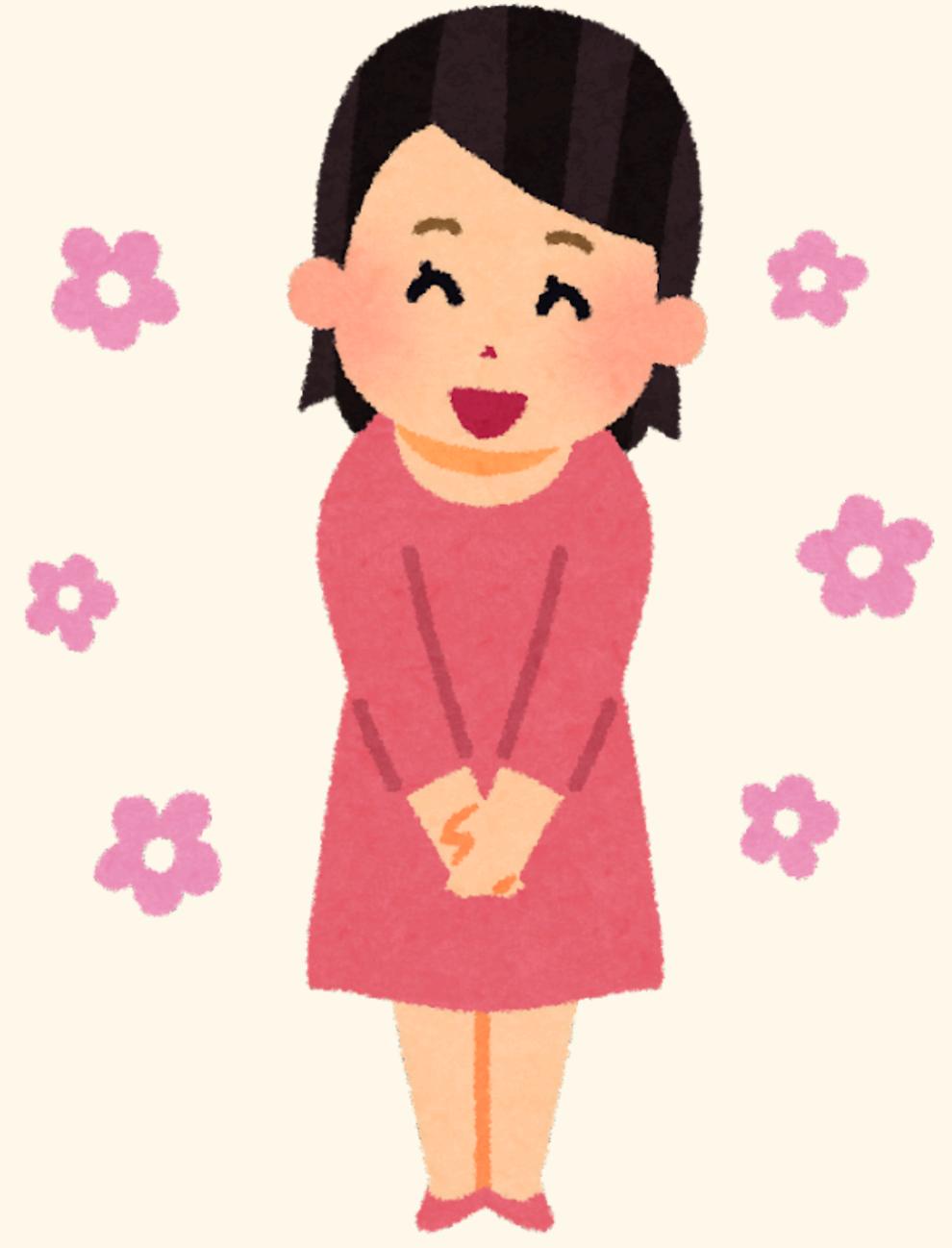
Determine the top 3 most ordered pizza types based on revenue for each pizza category

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

```
• SELECT category, name, revenue FROM
  (SELECT category, name, revenue, RANK () OVER(PARTITION BY category ORDER BY revenue DESC) AS rank_cate
  FROM
  (SELECT pizza_types.category, pizza_types.name,
  SUM(order_details.quantity*pizzas.price) AS revenue
  FROM pizza_types
  INNER JOIN pizzas
  ON pizza_types.pizza_type_id=pizzas.pizza_type_id
  INNER JOIN order_details
  ON order_details.pizza_id=pizzas.pizza_id
  GROUP BY pizza_types.category, pizza_types.name) AS category_name) AS rank_table
  WHERE rank_cate<=3;
```



Hence, the chicken category stands out as the most lucrative among the three categories, featuring the top three revenue-generating pizzas: Thai chicken, Barbecue chicken, and California chicken, in that order.



THANK YOU!