

SQL PROJECT

SAKLAİN'S PIZZA

A PROJECT ON PIZZAHUT SALES ANALYSIS



[HTTPS://WWW.LINKEDIN.COM/IN/MD-SAKLAİN-RAZA-1A6A9A287/](https://www.linkedin.com/in/md-saklain-raza-1a6a9a287/)



WELCOME TO Saklain's Pizza

Hi, I'm Md Saklain Raza. In this project, I utilized SQL queries to analyze and solve problems related to pizza sales data.



PROJECT SUMMARY

01 OBJECTIVE:

To analyze pizza sales to:

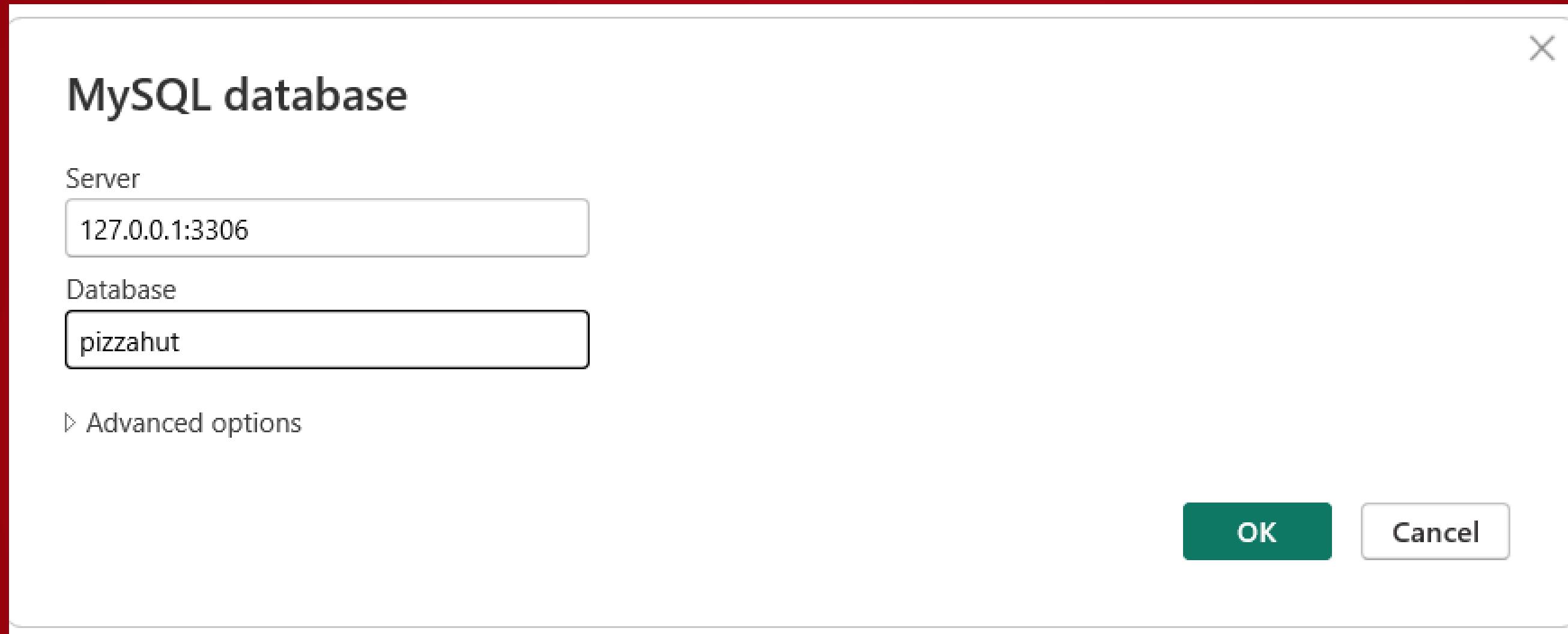
- Provide actionable insights on sales performance
- Revenue generation

02 FEATURES OF THE TOPIC:

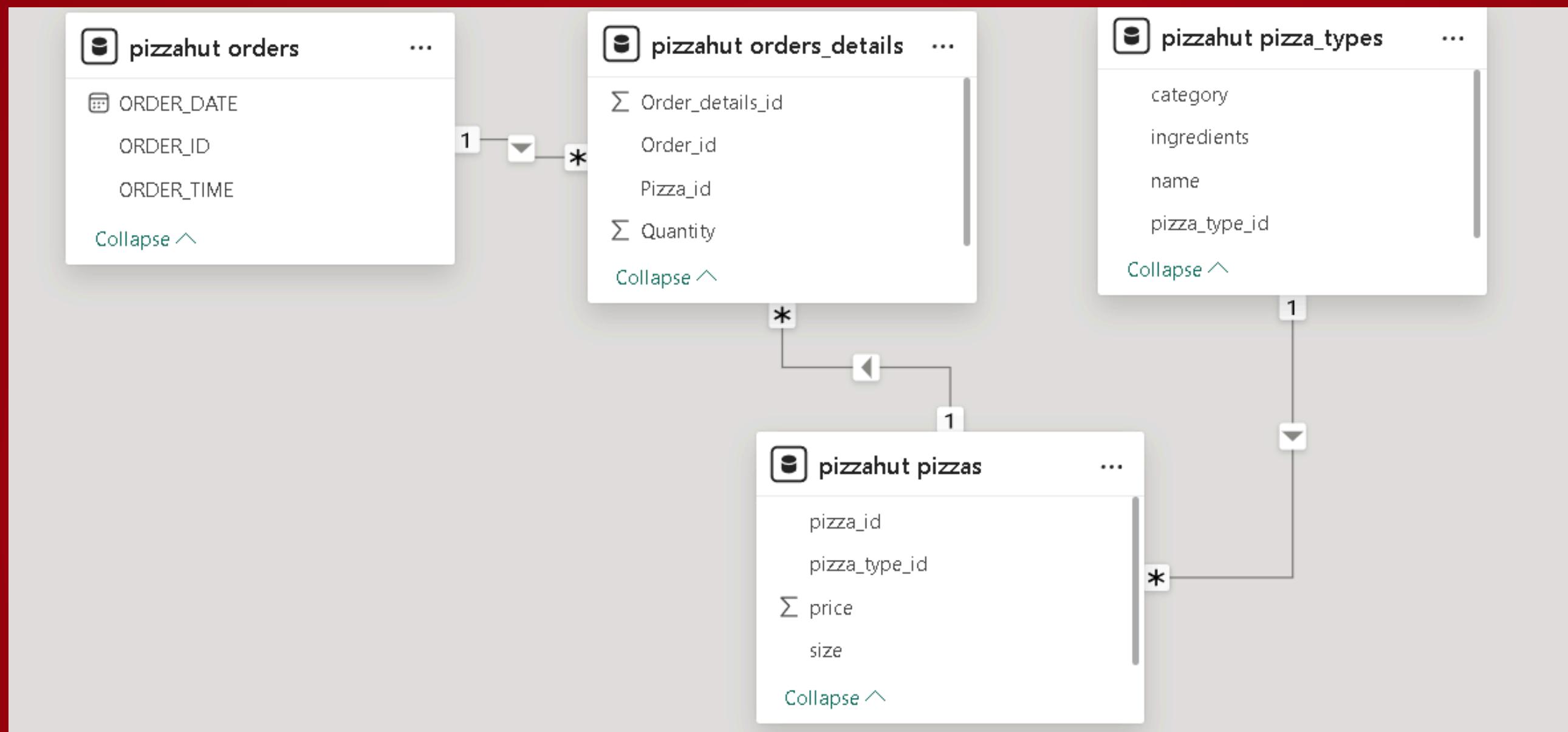
- Sub Query
- Group by & Order by
- Cumulative sum
- Rank
- Visualizations



CONNECTING TO PIZZAHUT DATABASE



DATA SCHEMA



BASIC QUESTIONS



Q1. Retrieve the total number of orders placed.

```
SELECT  
    COUNT(ORDER_ID) AS TOTAL_ORDERS  
FROM  
    orders;
```

Result Grid		Filter Rows:
	TOTAL_ORDERS	
▶	21350	

Q2. Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(orders_details.Quantity * pizzas.price),  
          2) AS TOTAL_REVENUE  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.Pizza_id;
```

Result Grid		Filter Rows:
TOTAL_REVENUE		
▶	817860.05	

Q3. Identify the highest-priced pizza.

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

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

Q4. Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(orders_details.Order_details_id) AS ORDER_COUNT
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.Pizza_id
GROUP BY pizzas.SIZE
ORDER BY ORDER_COUNT DESC
LIMIT 1;
```

Result Grid | Filter Row

	size	ORDER_COUNT
▶	L	18526

Q5. List the top 5 most ordered pizza types along with their quantities.

```
• 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 orders_details.Pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY TOTAL_Quantity DESC  
LIMIT 5;
```

	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

INTERMEDIATE QUESTIONS

Q6. Join the necessary tables to find the total quantity of each pizza category ordered.

- **SELECT**

```
    pizza_types.category,  
    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 orders_details.Pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY TOTAL_QUANTITY DESC;
```

Result Grid | Filter Rows:

	category	TOTAL_QUANTITY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Q7. Determine the 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  
ORDER BY COUNT_ORDER DESC;
```

Result Grid | Filter Rows:

	HOUR	COUNT_ORDER
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28

Q8.Join relevant tables to find the category-wise distribution of pizzas.

- **SELECT**
category, COUNT(name) **AS** COUNT_NAME
FROM
pizza_types
GROUP BY category
ORDER BY COUNT_NAME **DESC**;

Result Grid | Filter Rows:

	category	COUNT_NAME
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

- **SELECT**

```
    ROUND(AVG(TOTAL_QUANTITY), 0) AS AVG_PIZZAS_ORDER_PER_DAY
```

```
FROM
```

```
(SELECT
```

```
    orders.ORDER_DATE,
```

```
        SUM(orders_details.Quantity) AS TOTAL_QUANTITY
```

```
FROM
```

```
    orders
```

```
JOIN orders_details ON orders.ORDER_ID = orders_details.Order_id
```

```
GROUP BY orders.ORDER_DATE) AS ORDER_QUANTITY;
```

Result Grid		Filter Rows:
	Avg_Pizzas_Order_Per_Day	
▶	138	

Q10. Determine the top 3 most ordered pizza types based on revenue.

- **SELECT**

```
    pizza_types.name,  
    SUM(orders_details.Quantity * pizzas.price) 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.name  
ORDER BY REVENUE DESC  
LIMIT 3;
```

Result Grid | Filter Rows:

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

ADVANCED QUESTIONS

Q11. Calculate the percentage contribution of each pizza type to total revenue.

```
• SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.Quantity * pizzas.price) / (SELECT
        SUM(orders_details.Quantity * pizzas.price)
    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;
```

Result Grid | Filter Rows:

	category	REVENUE
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q12. Analyze the cumulative revenue generated over time.

```
• SELECT ORDER_DATE,REVENUE,
ROUND(SUM(REVENUE) OVER(ORDER BY ORDER_DATE),2) AS CUM_REVENUE
FROM
(SELECT orders.ORDER_DATE,
ROUND(SUM(orders_details.Quantity*pizzas.price),2) AS REVENUE
FROM orders_details JOIN pizzas
ON orders_details.Pizza_id=pizzas.pizza_id
JOIN orders
ON orders_details.Order_id=orders.ORDER_ID
GROUP BY orders.ORDER_DATE) AS SALES;
```

	ORDER_DATE	REVENUE	CUM_REVENUE
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.5
	2015-01-07	2202.2	16560.7
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.4
	2015-01-10	2463.95	23990.35
	2015-01-11	1872.3	25862.65
	2015-01-12	1919.05	27781.7
	2015-01-13	2049.6	29831.3

Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
• SELECT category, name, REVENUE, RN  
  FROM  
    (SELECT category, name, REVENUE,  
     RANK() OVER(PARTITION BY category ORDER BY REVENUE DESC) AS RN  
    FROM  
    (SELECT pizza_types.category, pizza_types.name,  
     ROUND(SUM(orders_details.Quantity*pizzas.price),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, pizza_types.name) AS a) AS b  
  WHERE RN<=3;
```

	category	name	REVENUE	RN
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.7	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3

THANK YOU!

