

UNVEILING INSIGHTS FROM PIZZA MAX SALES DATA: AN ADVANCED SQL ANALYSIS

Excited to share my latest data analysis project where I delved into the world of pizza sales using advanced SQL techniques. From discovering the top-selling pizzas to analyzing revenue trends, here's how I turned raw data into actionable insights.





PIZZA MAX SALES DATA ANALYSIS PROJECT

- **Objective:** To analyze pizza sales data and uncover key insights using advanced SQL queries.
- **Data Source:** Four CSV files from GitHub: Order Details, Orders, Pizzas, and Pizza Types.
- **Tools Used:** MySQL for data storage and querying.

DATA SOURCES

- Data downloaded from GitHub.
- Four CSV files:
 - a. Order Details
 - b. Orders
 - c. Pizzas
 - d. Pizza Types





DATA PREPARATION

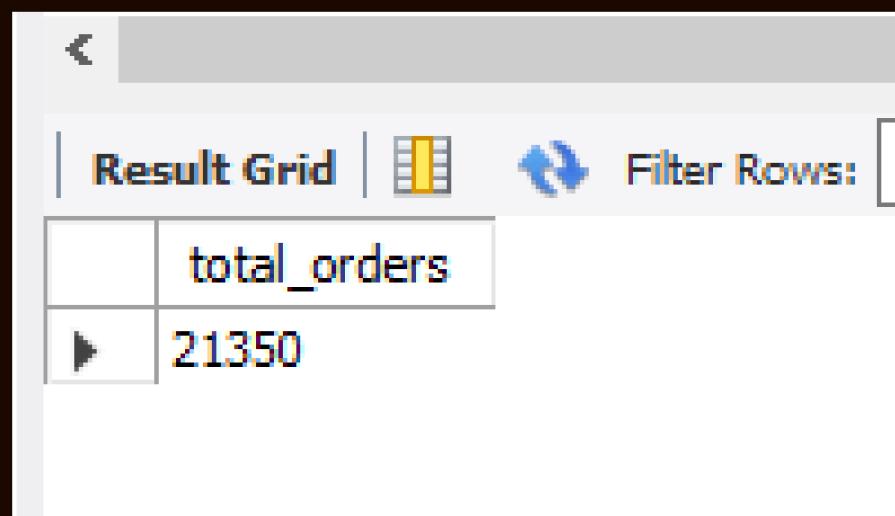
- Imported data into MySQL.
- Created tables for Order Details and Orders due to incorrect data types in CSV files.
- Ensured data consistency and accuracy before analysis.

BASIC ANALYSIS - RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

Query:

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

Result:



A screenshot of a MySQL Workbench result grid. The grid has one column labeled 'total_orders' and one row containing the value '21350'. There are navigation arrows at the bottom left of the grid.

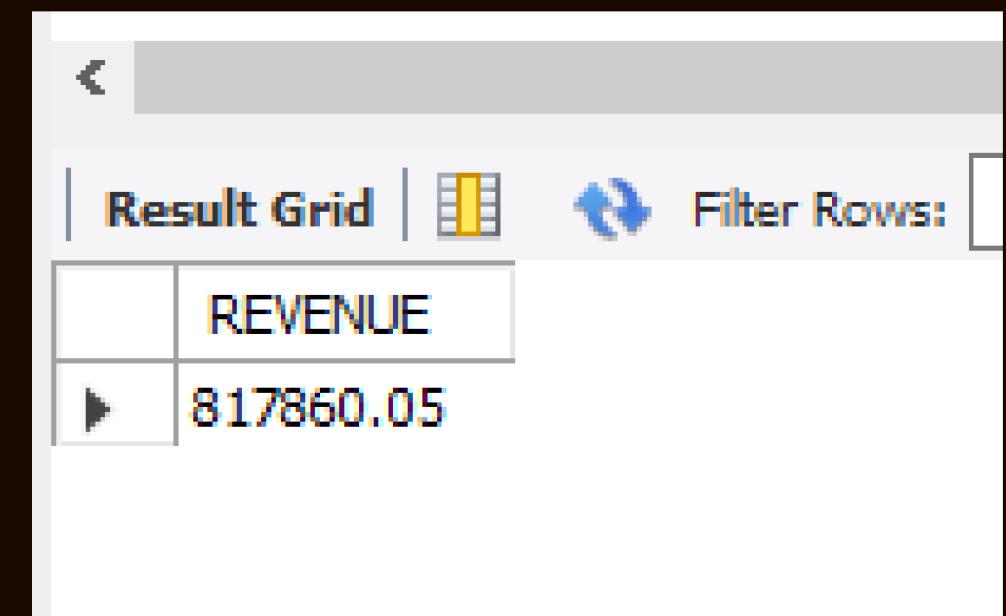
total_orders
21350

BASIC ANALYSIS - TOTAL REVENUE

Query:

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

Result:



A screenshot of a MySQL Workbench result grid. The grid has one column labeled "REVENUE". The first row is a header, and the second row contains the value "817860.05". The grid is titled "Result Grid" and has a "Filter Rows" button.

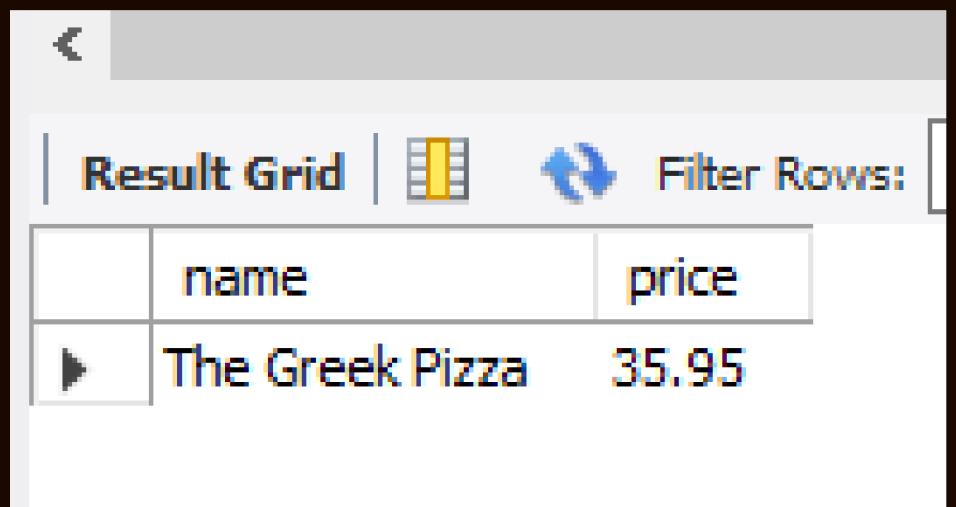
REVENUE
817860.05

BASIC ANALYSIS - HIGHEST PRICED PIZZA

Query:

```
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:



A screenshot of a MySQL Workbench result grid. The grid has two columns: 'name' and 'price'. There is one row of data: 'The Greek Pizza' with a price of '35.95'. The grid includes standard database navigation buttons like back, forward, and refresh, along with a 'Filter Rows:' button.

	name	price
▶	The Greek Pizza	35.95

BASIC ANALYSIS - MOST COMMON PIZZA SIZE ORDERED

Query:

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

Result:

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

BASIC ANALYSIS - TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

Query:

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

Result:

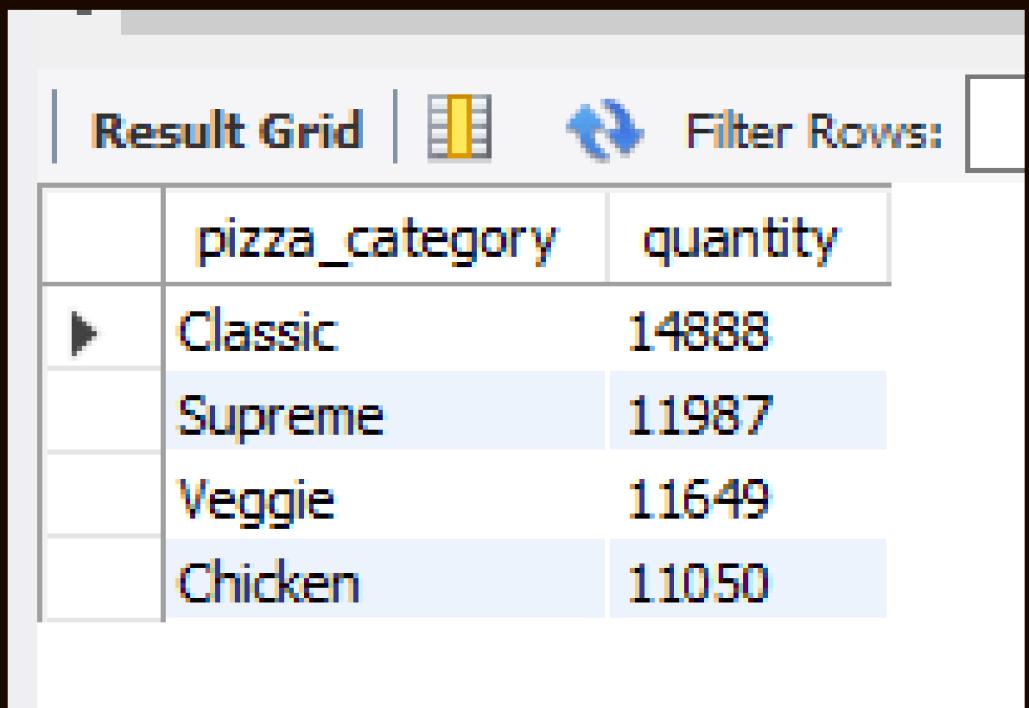
	pizza_names	quantity
▶	The Classic Deluxe Pizza	2416
▶	The Barbecue Chicken Pizza	2372
▶	The Hawaiian Pizza	2370
▶	The Pepperoni Pizza	2369
▶	The Thai Chicken Pizza	2315

INTERMEDIATE ANALYSIS - TOTAL QUANTITY BY CATEGORY

Query:

```
SELECT
    pizza_types.category AS pizza_category,
    sum(order_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_category order by quantity desc;
```

Result:



A screenshot of a database query result grid. The grid has a header row with columns labeled "pizza_category" and "quantity". There are four data rows below the header, each containing a category name and its corresponding quantity. The categories and their quantities are: Classic (14888), Supreme (11987), Veggie (11649), and Chicken (11050). The grid includes standard database interface elements like a toolbar with "Result Grid" and "Filter Rows" buttons.

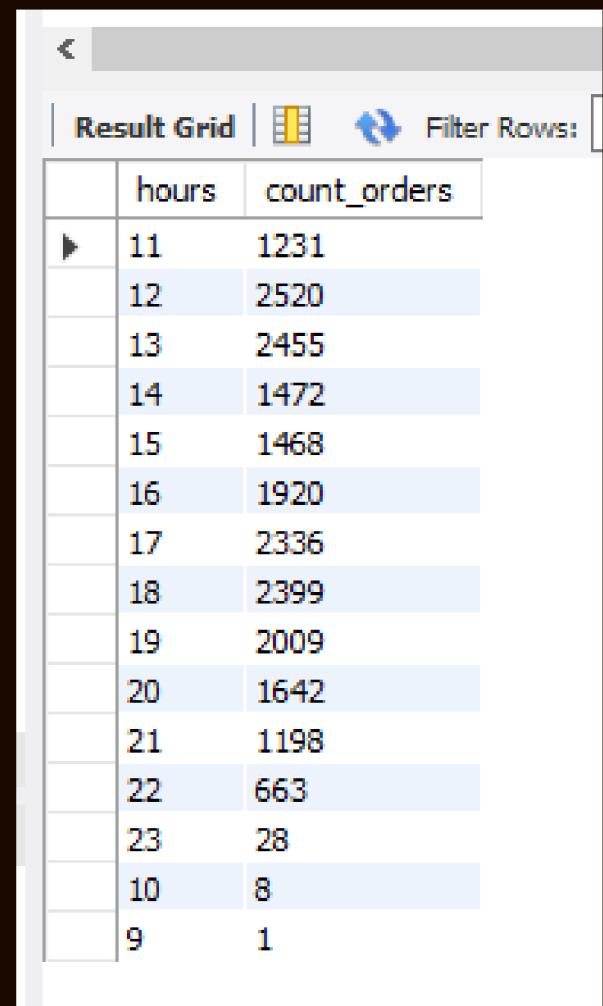
	pizza_category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

INTERMEDIATE ANALYSIS - ORDERS BY HOUR

Query:

```
SELECT
    hour(order_time) AS hours,
    COUNT(order_id) AS count_orders
FROM
    orders
GROUP BY hours;
```

Result:



A screenshot of a database query result grid titled "Result Grid". The grid has two columns: "hours" and "count_orders". The data shows the number of orders per hour from 9 to 11. The row for hour 10 is highlighted.

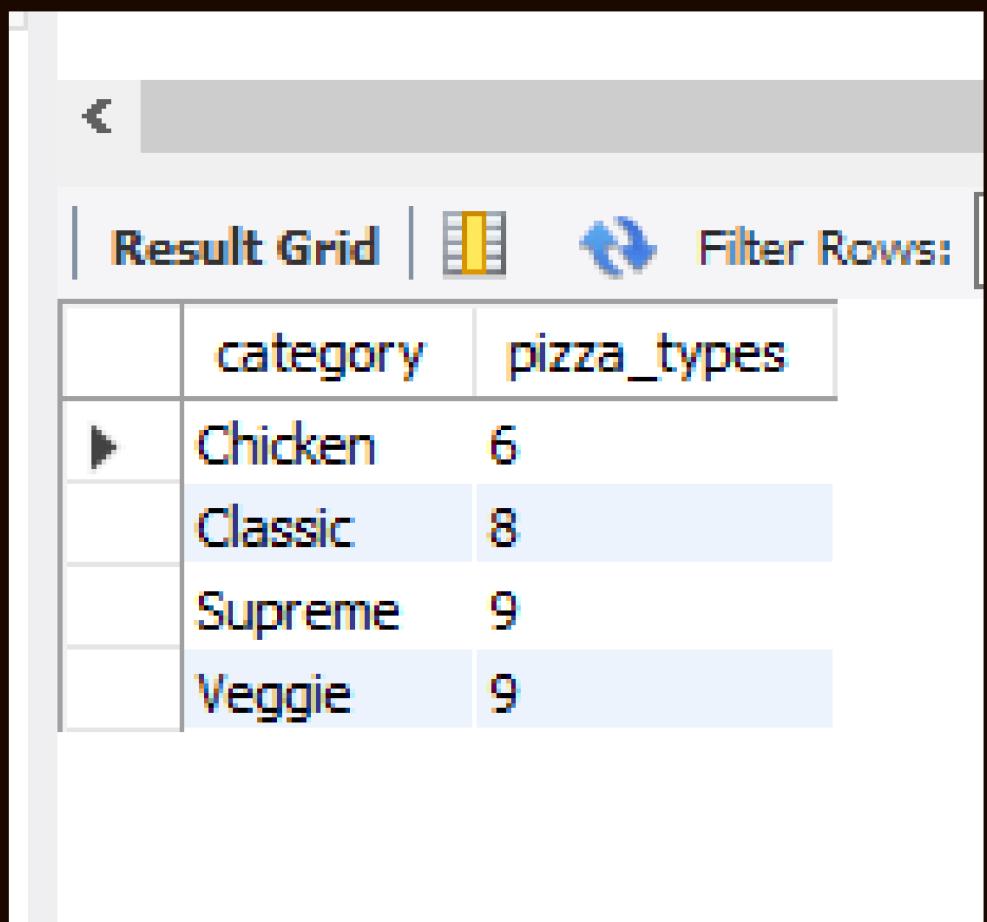
	hours	count_orders
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

INTERMEDIATE ANALYSIS - CATEGORY-WISE DISTRIBUTION

Query:

```
SELECT
    category AS category, COUNT(category) AS pizza_types
FROM
    pizza_types
GROUP BY category;
```

Result:



The screenshot shows a database query results window. At the top, there are navigation buttons (back, forward, search) and tabs for 'Result Grid' (which is selected), 'Filter Rows', and other options. The result grid displays the following data:

	category	pizza_types
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

INTERMEDIATE ANALYSIS - AVERAGE PIZZAS ORDERED PER DAY

Query:

```
SELECT
    ROUND(AVG(quantity), 0) As Avg_pizza
FROM
    (SELECT
        DATE(orders.order_date) AS date,
        ROUND(SUM(order_details.quantity), 0) AS quantity
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY date) AS pizza_quantity;
```

Result:

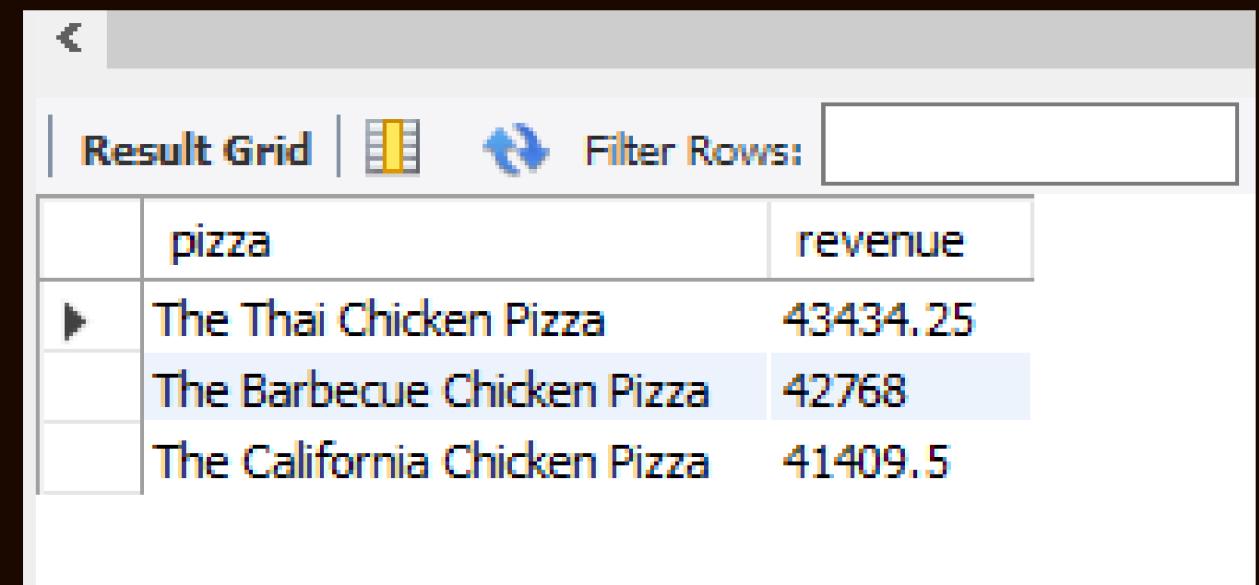
	Avg_pizza
▶	138

INTERMEDIATE ANALYSIS - TOP 3 MOST ORDERED PIZZAS BY REVENUE

Query:

```
SELECT
    pizza, SUM(price) AS revenue
FROM
    (SELECT
        pizza_types.name AS pizza,
        (order_details.quantity * pizzas.price) AS price
    FROM
        pizza_types
    JOIN pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id) AS half_data
GROUP BY pizza
ORDER BY revenue DESC limit 3;
```

Result:



The screenshot shows a MySQL Workbench result grid with three columns: 'pizza' (containing the names of the pizzas), 'revenue' (containing the total revenue for each pizza), and a third column which is partially visible. The results are as follows:

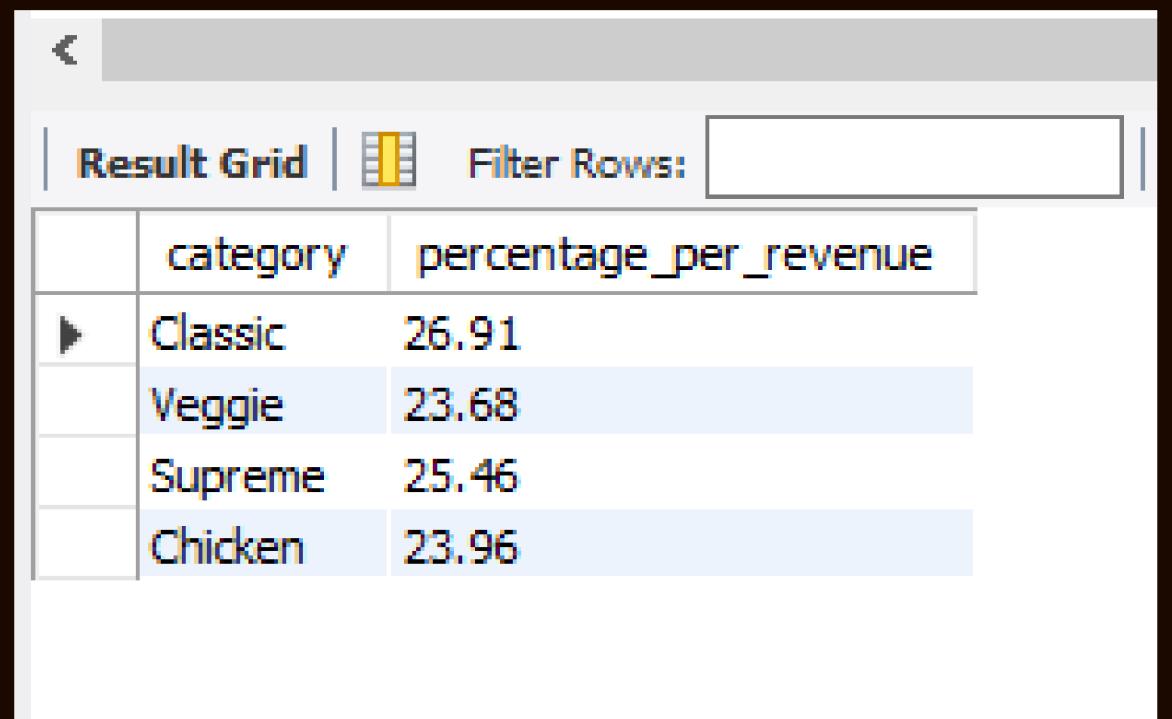
	pizza	revenue
▶	The Thai Chicken Pizza	43434.25
▶	The Barbecue Chicken Pizza	42768
▶	The California Chicken Pizza	41409.5

ADVANCED ANALYSIS - PERCENTAGE CONTRIBUTION OF PIZZA TYPE

Query:

```
with cte as (
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),2) AS REVENUE
FROM
    order_details
    JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id
)
SELECT
    pizza_types.category,
    round(SUM(order_details.quantity * pizzas.price) / (select revenue from cte ) * 100 ,2) as
percentage_per_revenue
FROM
    pizza_types
    JOIN
        pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

Result:



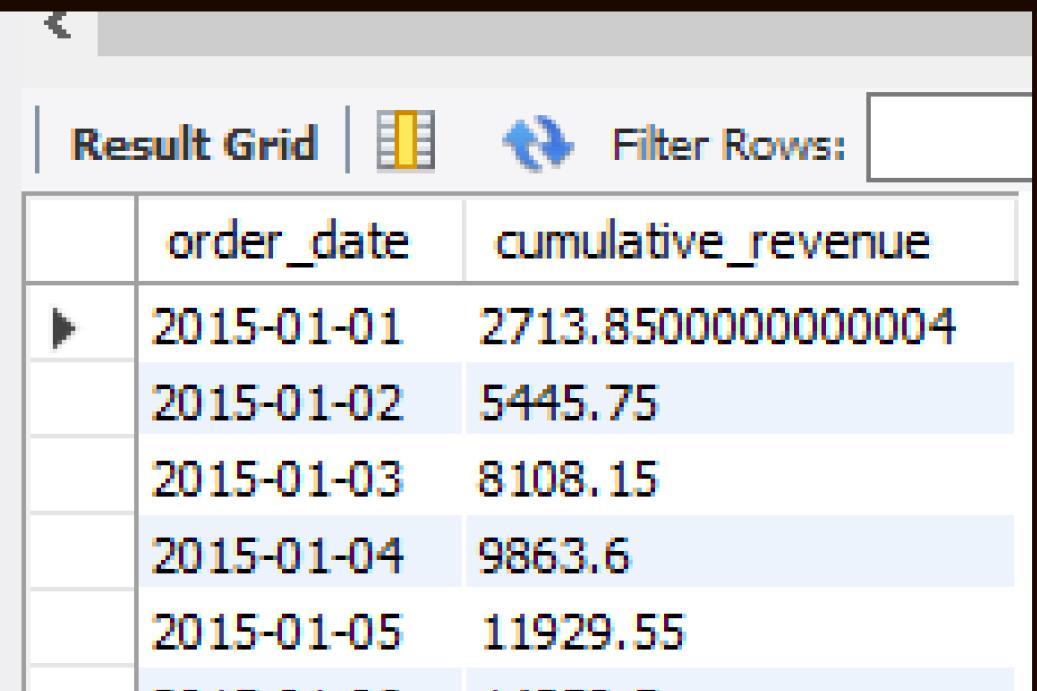
	category	percentage_per_revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

ADVANCED ANALYSIS - CUMULATIVE REVENUE

Query:

```
select order_date,  
       sum(revenue) over (order by order_date) as cumulative_revenue from  
(  
    SELECT  
        orders.order_date,  
        SUM(order_details.quantity * pizzas.price) as revenue  
    FROM  
        order_details  
        JOIN  
        pizzas ON pizzas.pizza_id = order_details.pizza_id  
        JOIN  
        orders ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) as sales;
```

Result:



The screenshot shows a software interface for viewing database query results. At the top, there are tabs for 'Result Grid' and 'Filter Rows'. The main area displays a grid of data with two columns: 'order_date' and 'cumulative_revenue'. The data shows the cumulative sum of revenue starting from January 1st, 2015.

	order_date	cumulative_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

Query:

```
select name,revenue from
(select category,name,revenue,
rank() over (partition by category order by revenue desc)as rn from
(SELECT
pizza_types.category,pizza_types.name,
SUM(order_details.quantity * pizzas.price) as revenue
FROM
pizza_types
JOIN
pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN
order_details on order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category,pizza_types.name) as a) as b
where rn <=3;
```

Result:

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

KEY INSIGHTS FROM THE ANALYSIS

- **Total Orders and Revenue:** 21,350 orders generating \$817,860.05.
- **Highest-Priced Pizza:** The Greek Pizza at \$35.95.
- **Popular Sizes and Types:** Most common size: Large.
- **Top 5 Pizzas:**
 - The Classic Deluxe Pizza
 - The Barbecue Chicken Pizza
 - The Hawaiian Pizza
 - The Pepperoni Pizza
 - The Thai Chicken Pizza
- **Average Pizzas Ordered Per Day:** 138
- **Major Contributions to Revenue:** The Thai Chicken Pizza, The Barbecue Chicken Pizza, The California Chicken Pizza
- **Order Patterns:** Orders around peak hours are 2,200; normally, orders are 1,500.

CONCLUSION

- **Summary:** This project showcased my ability to handle data analysis using advanced SQL techniques, providing valuable insights into pizza sales patterns.
- **Value:** The findings can help Pizza Max optimize their menu offerings and marketing strategies.
- **Call to Action:** Looking forward to applying my analytical skills in new challenges and opportunities. Feel free to connect with me for more insights and collaborations.

THANK YOU

Arif Ali

arif.soomro519@gmail.com

0314-1158733

