Pizza Shop
TASTY
HOT & SPICY PIZZA!!
Phone Orders:
1564 555 333
PIZZA | SALADS | PASTA | HOT SANDWICHES
#123 Your Street, City Name, www.websitename.com
Free Delivery

SQL Queries Project: Optimizing Data Analysis and Insights.

A Comprehensive Approach to Solving Complex SQL Queries

Understand customer preferences, analyze sales trends, and optimize business strategy

SN **by Satish Nayak**

Overview of SQL Queries Solved During the Project

1

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.

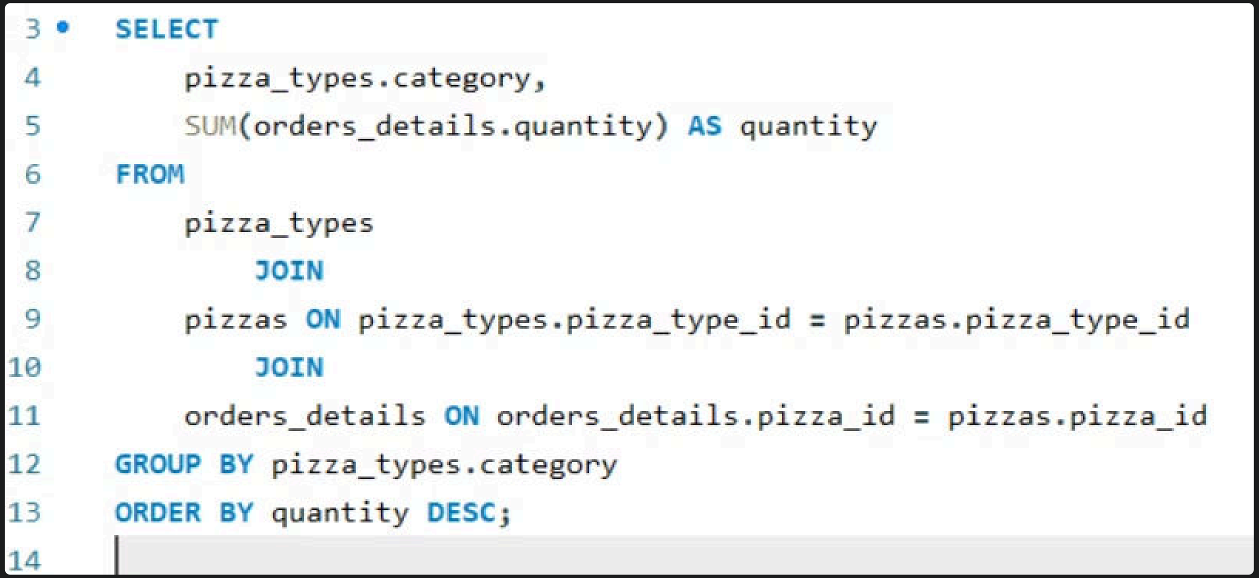
2

- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

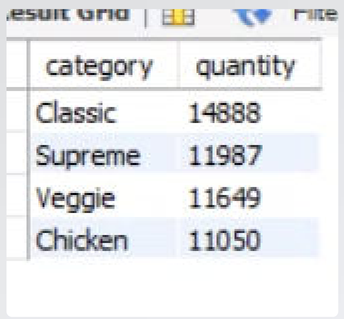
3

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Total Quantity by Pizza Category



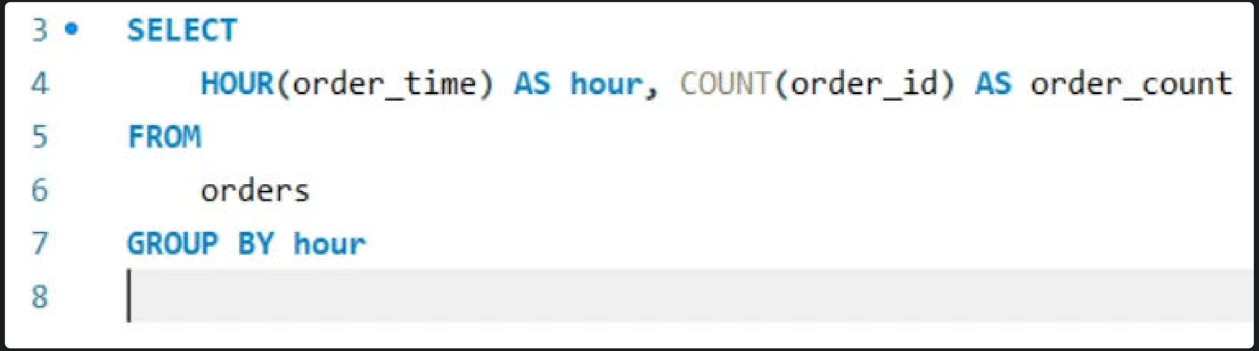
```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```



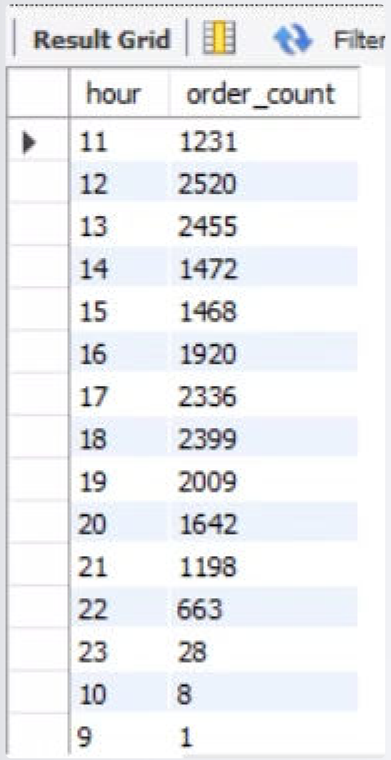
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Made with Gamma

Order Distribution by Hour

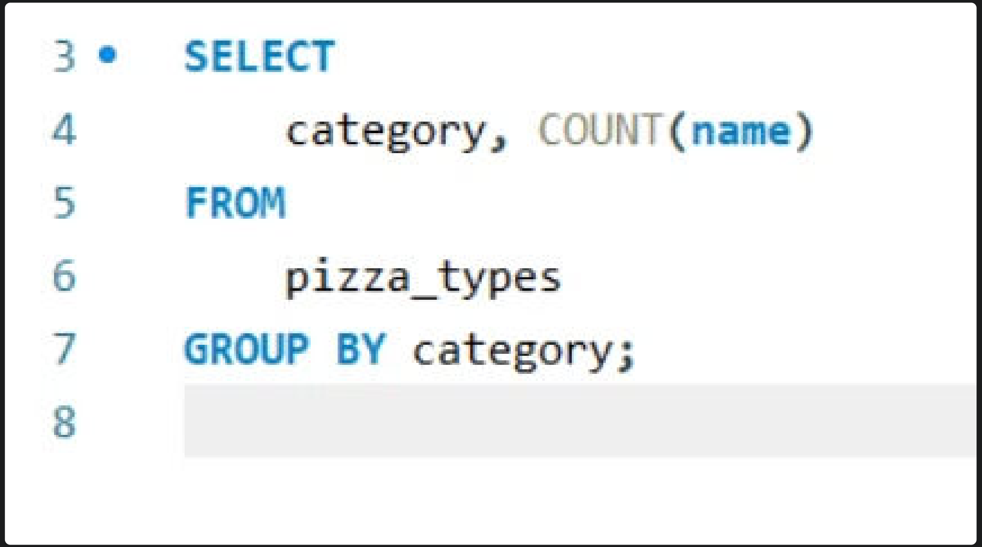


```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour
```

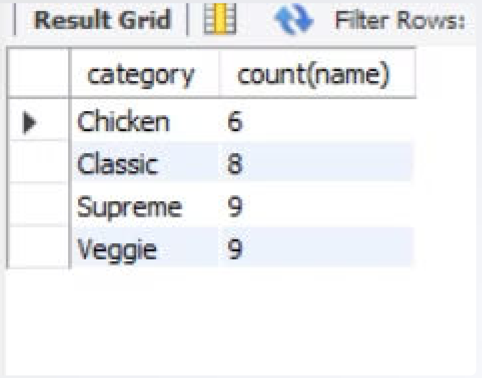

Result Grid
Filter

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

Category-Wise Pizza Distribution

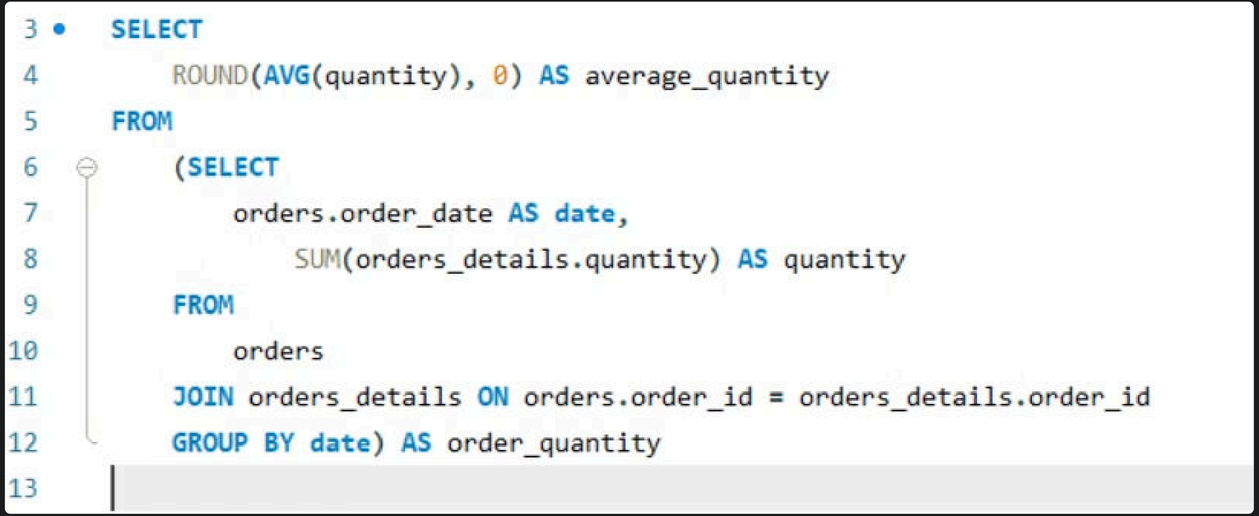


```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

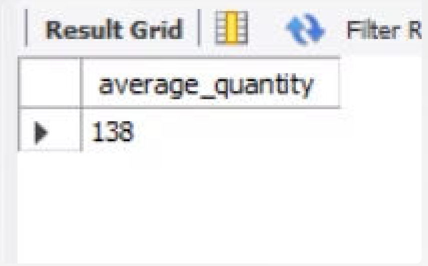


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Average Pizzas Ordered per Day

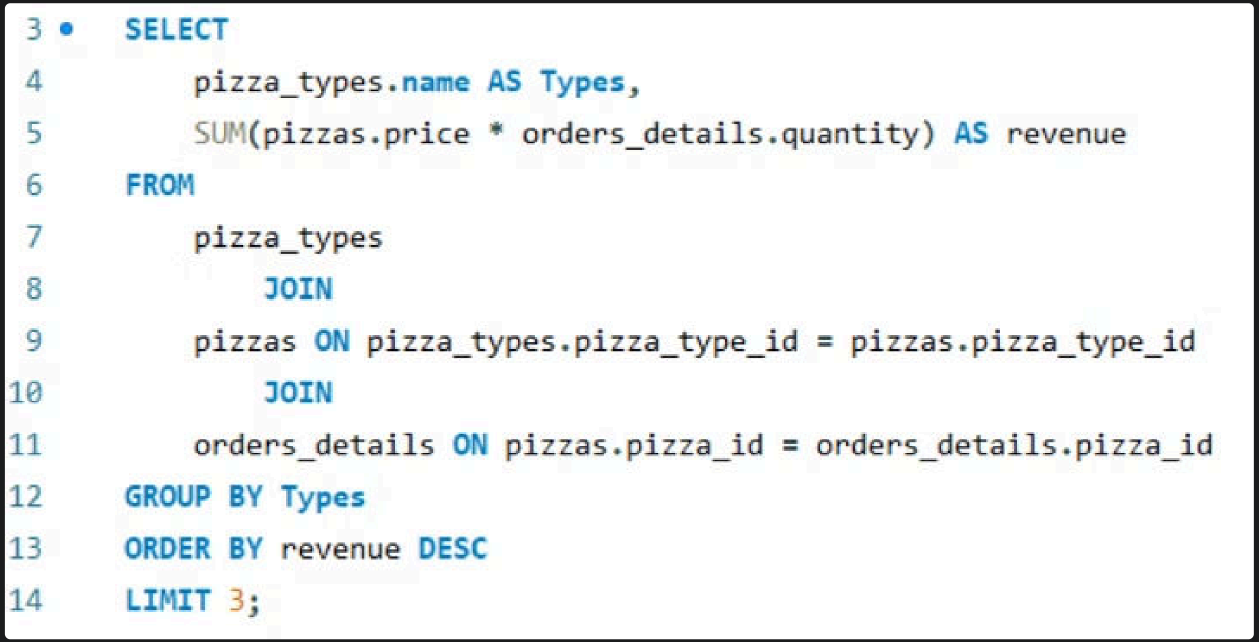


```
SELECT
    ROUND(AVG(quantity), 0) AS average_quantity
FROM
    (SELECT
        orders.order_date AS date,
            SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY date) AS order_quantity
```

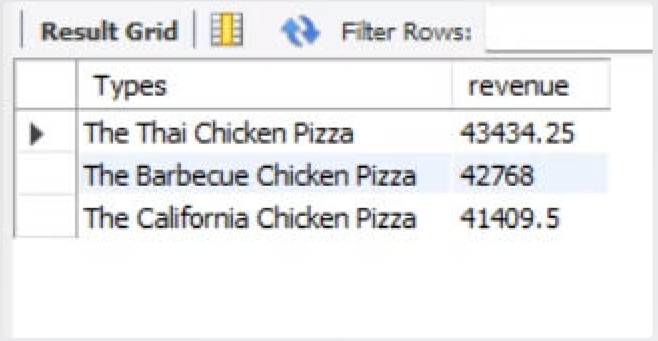


```
average_quantity
138
```

Top 3 Most Ordered Pizza Types by Revenue

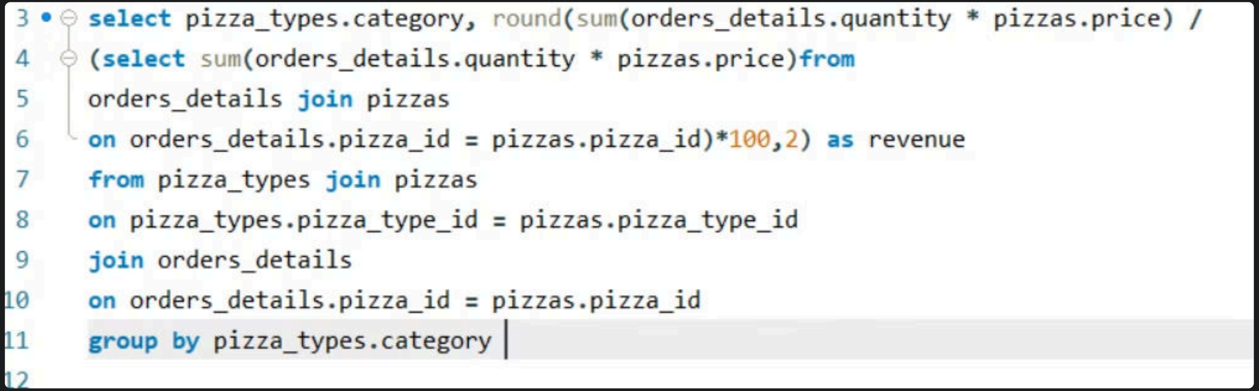


```
SELECT
    pizza_types.name AS Types,
    SUM(pizzas.price * orders_details.quantity) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY Types
ORDER BY revenue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

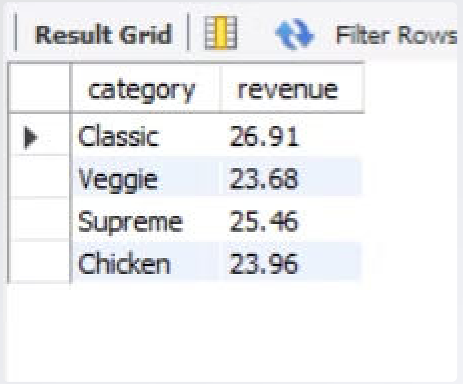


```
Types                          revenue
The Thai Chicken Pizza         43434.25
The Barbecue Chicken Pizza     42768
The California Chicken Pizza   41409.5
```

Percentage Contribution to Total Revenue



```sql
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
```

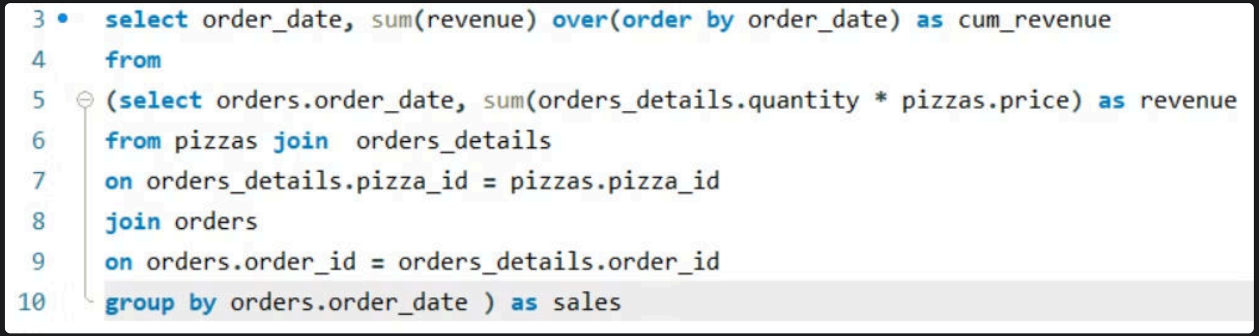


Result Grid | Filter Rows

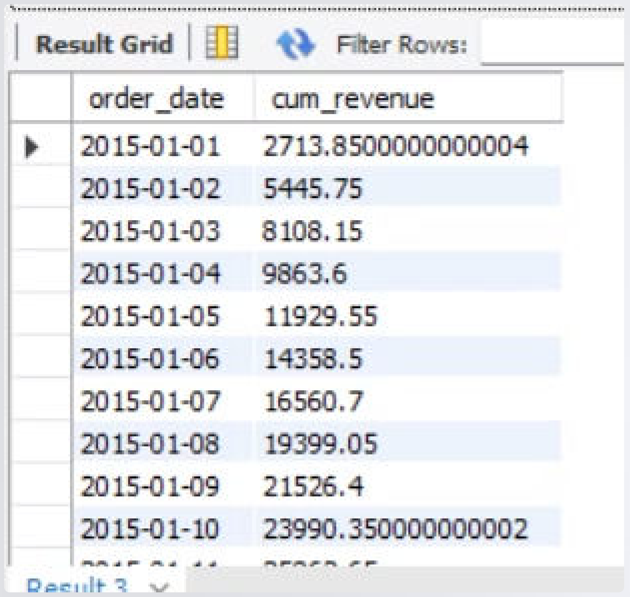
category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Made with Gamma

Cumulative Revenue Over Time



```sql
select order_date, sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date, sum(orders_details.quantity * pizzas.price) as revenue
from pizzas join  orders_details
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.order_date ) as sales
```

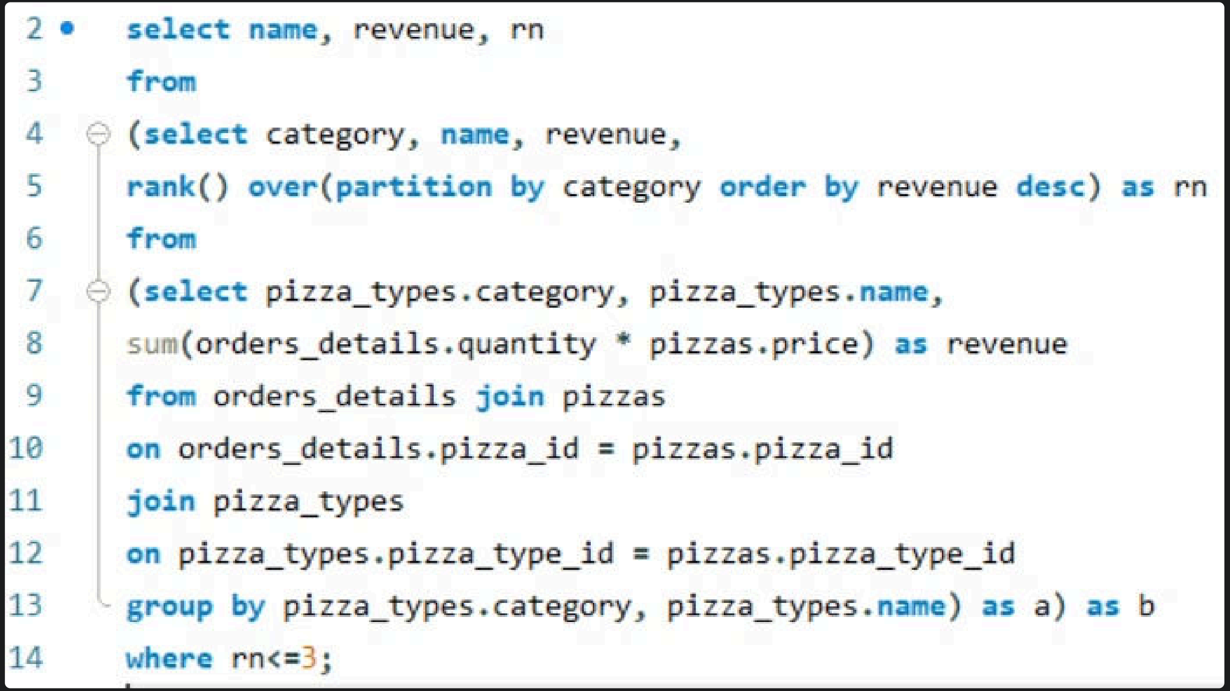


Result Grid | Filter Rows:

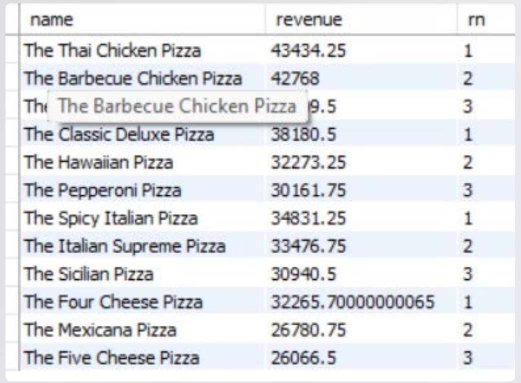
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002

Result 3

Top 3 Pizza Types by Revenue for Each Category



```sql
select 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,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<=3;
```



name	revenue	rn
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
Th[illegible]	[illegible]9.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

Made with Gamma

Project Impact:

- **Data Integration:** Enhanced my ability to join and merge multiple tables to derive meaningful insights.
- **Time-based Analysis:** Improved my skills in analyzing data trends over different time periods.
- **Aggregation and Grouping:** Learned to effectively group data to calculate averages and distributions.
- **Revenue Analysis:** Gained insights into revenue contribution and distribution across different categories.
- **Advanced SQL Techniques:** Developed a deeper understanding of advanced SQL techniques for complex queries and data analysis.

My sincere thanks to everyone who took the time to review my

SQL queries project.