Project Overview

This presentation highlights my SQL project, **PIZZATALES**, where I analyzed a large pizza sales dataset using MySQL. The goal was to explore how structured queries can uncover patterns, improve decision-making, and drive business growth.

Throughout the project, I used SQL to:

Identify top-selling pizzas and revenue trends
Spot areas where operations could be more efficient
Generate insights to guide business strategy

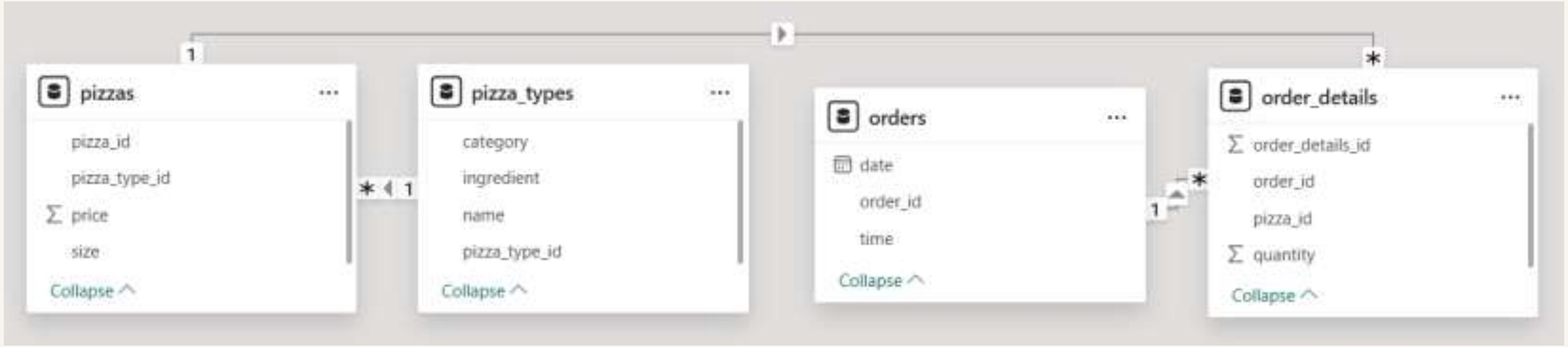
PIZZA SALES ANALYSIS

The slides include screenshots of key queries and results to show how data was transformed into meaningful insights for the business.

project by PURVA DEWANGAN

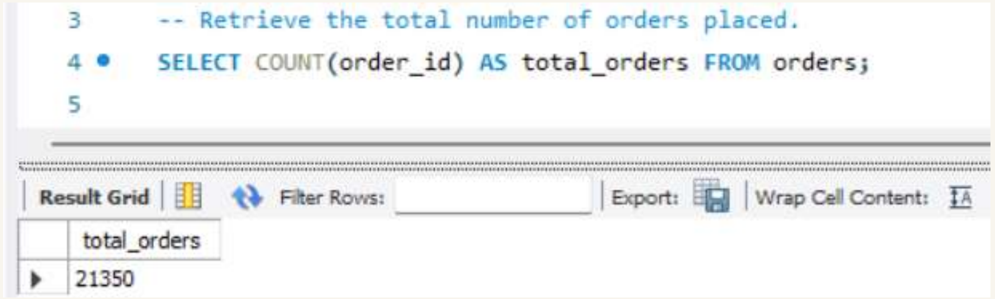
Table Overview

This is the model view of the 4 csv files used in this project:



project by PURVA DEWANGAN

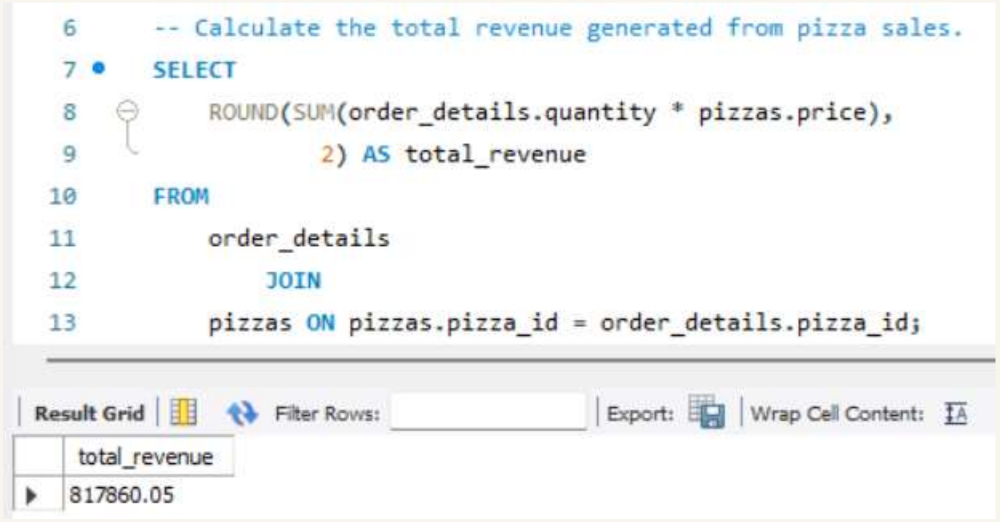
#1 Retrieve the total number of orders placed



```
-- Retrieve the total number of orders placed.
SELECT COUNT(order_id) AS total_orders FROM orders;
```

```
total_orders
21350
```

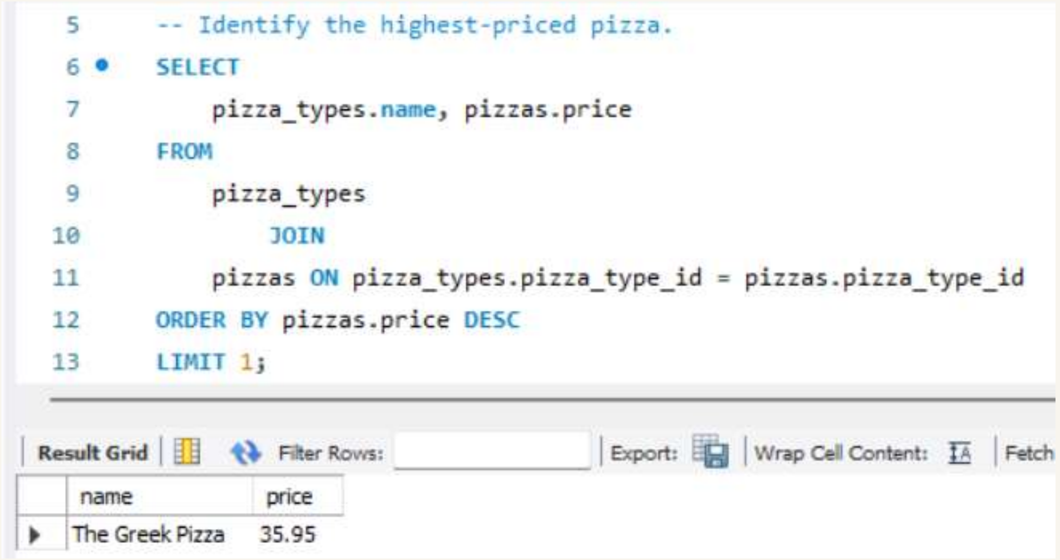
#2 Calculate the total revenue generated from pizza sales



```
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

```
total_revenue
817860.05
```

#3 Identify the highest-priced pizza

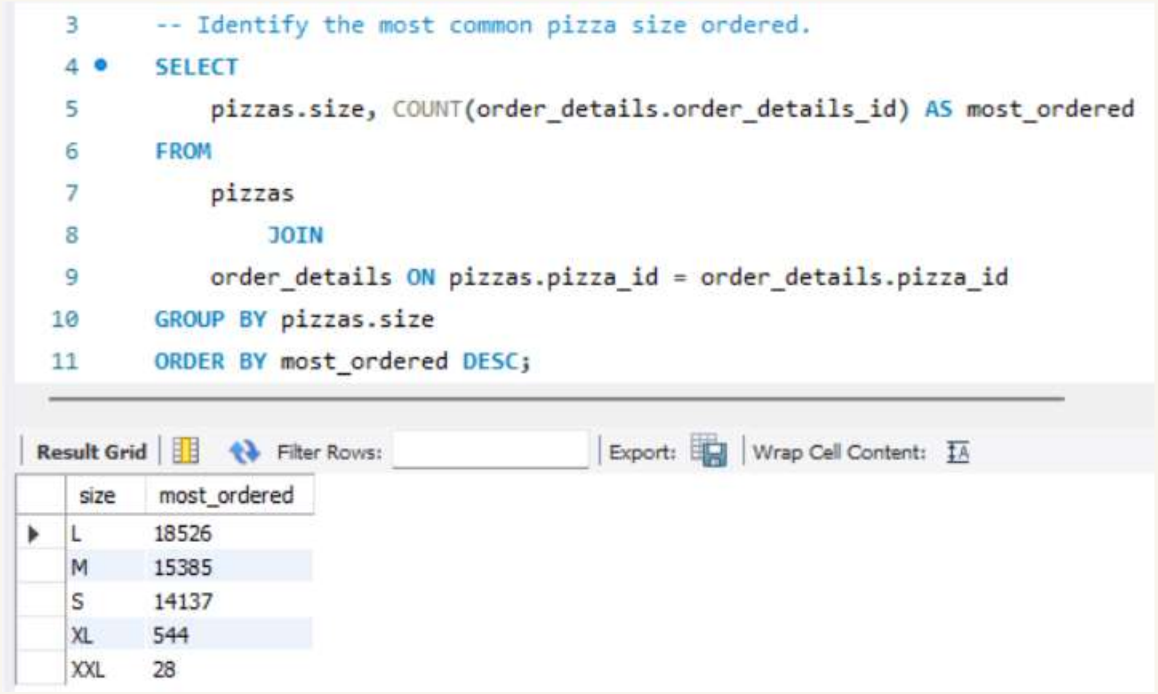


```
-- Identify the highest-priced pizza.
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch

name	price
The Greek Pizza	35.95

#4 Identify the most common pizza size ordered

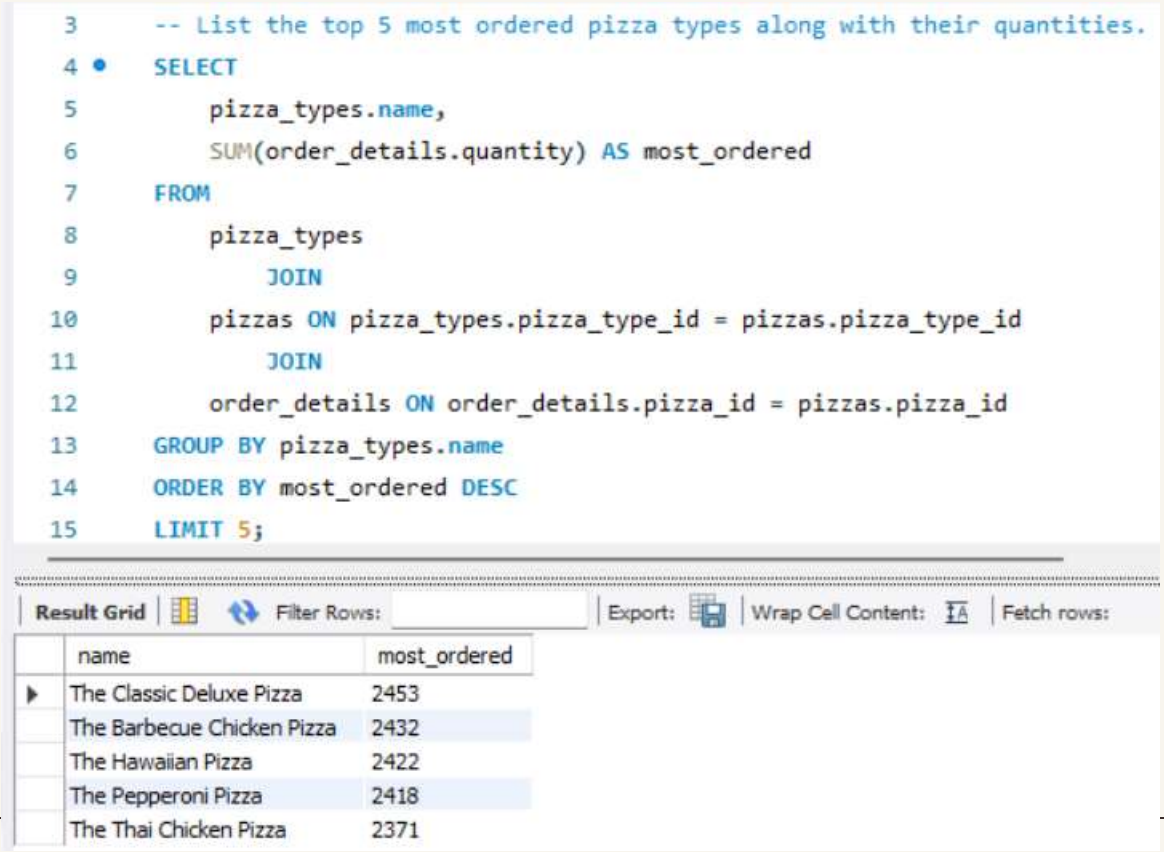


```
-- Identify the most common pizza size ordered.
SELECT
    pizzas.size, COUNT(order_details.order_details_id) AS most_ordered
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY most_ordered DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

size	most_ordered
L	18526
M	15385
S	14137
XL	544
XXL	28

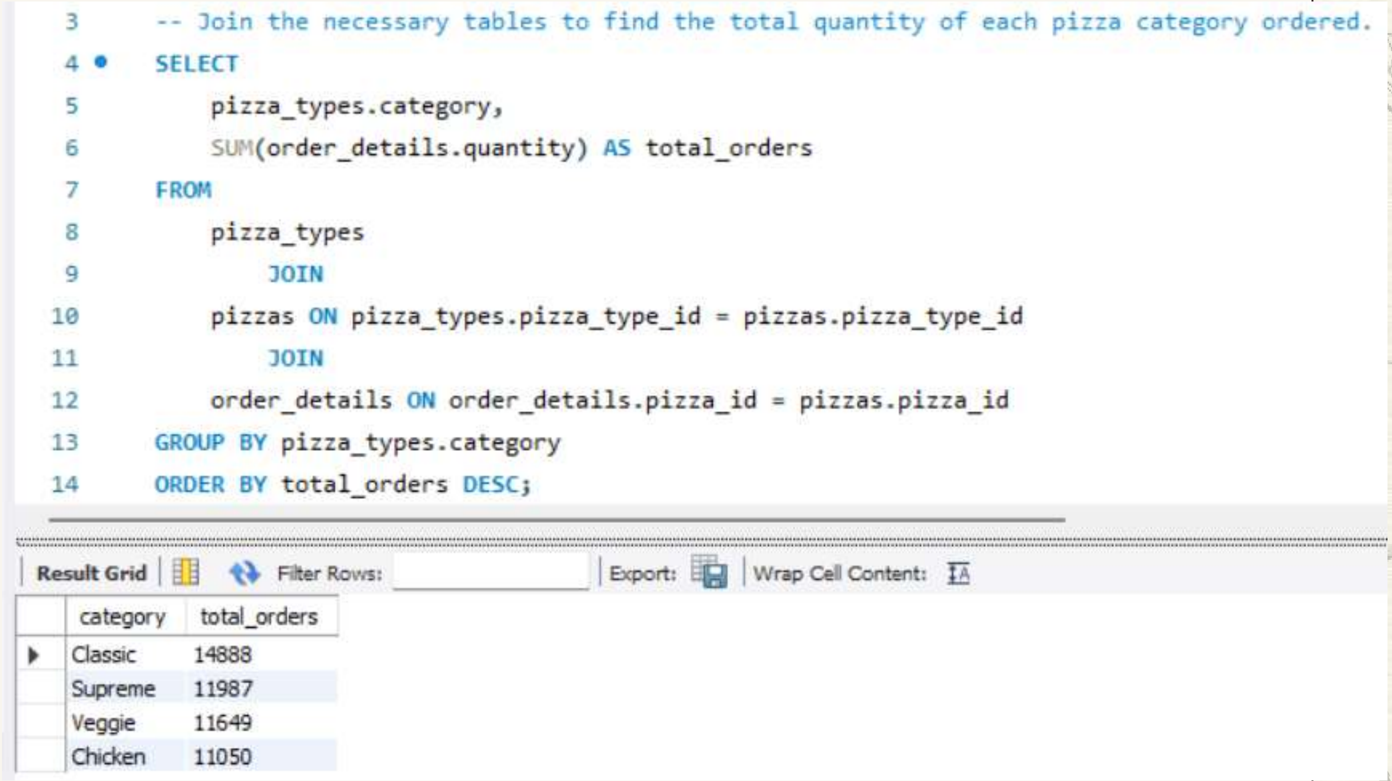
#5
List the top 5 most ordered pizza types along with their quantities



```
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS most_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY most_ordered DESC
LIMIT 5;
```

name	most_ordered
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

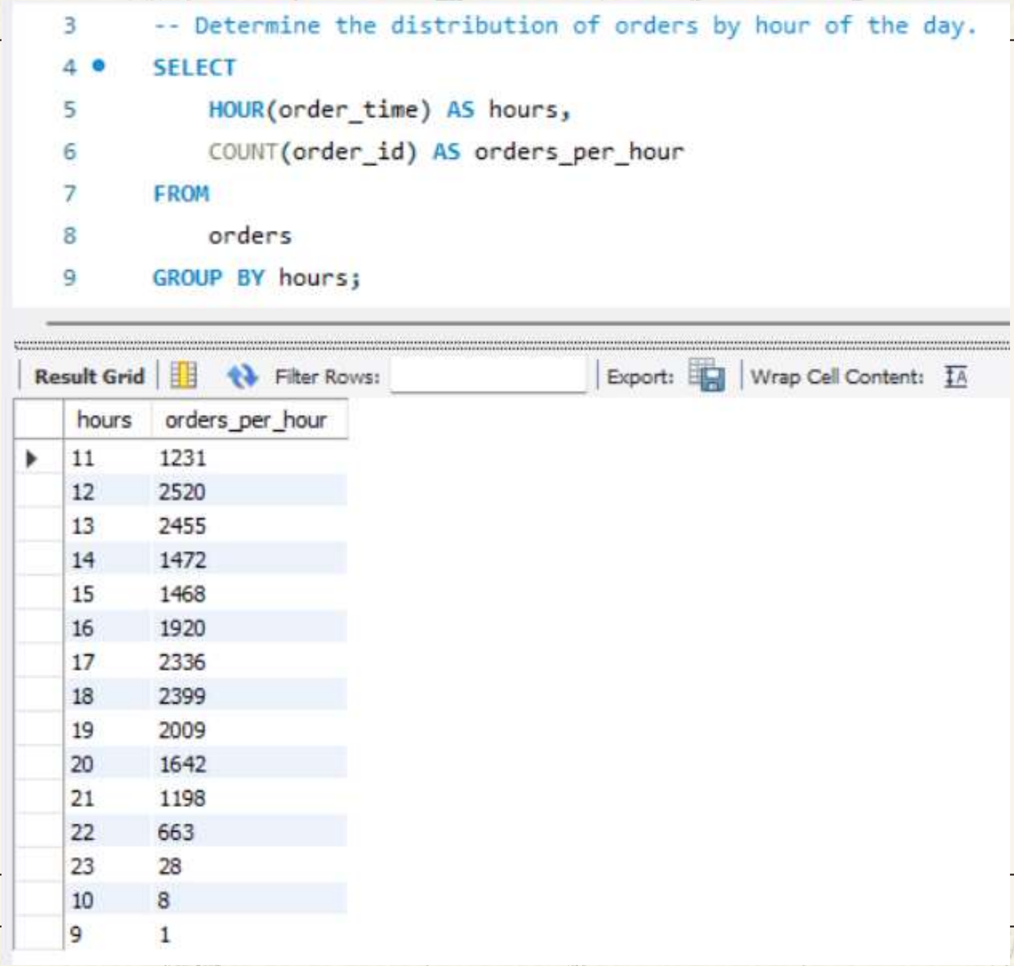


```
-- Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_orders
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_orders DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	total_orders
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

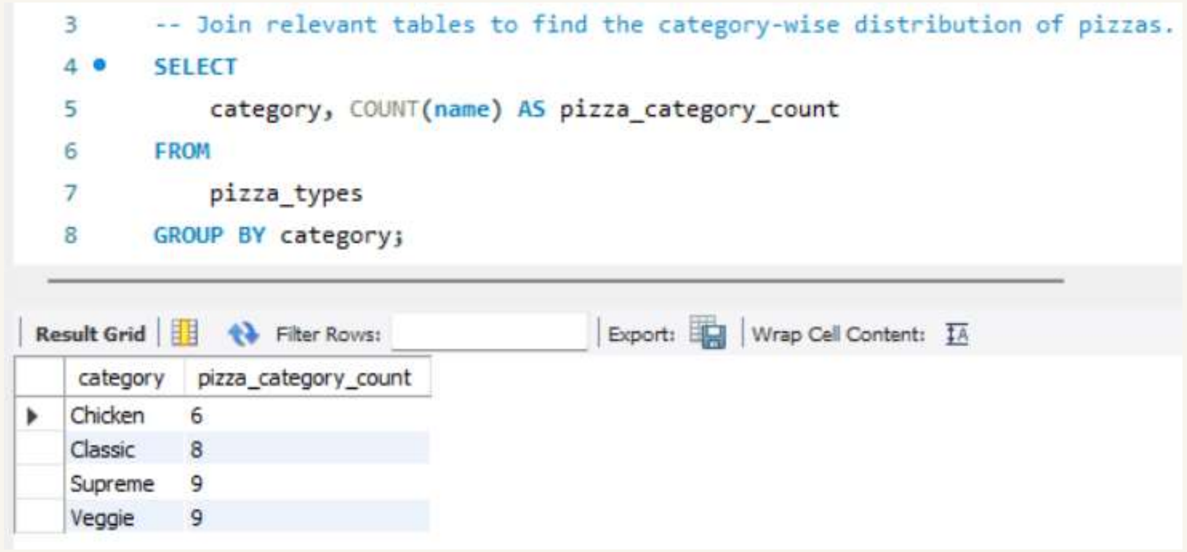
#7 Determine the distribution of orders by hour of the day



```
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hours,
    COUNT(order_id) AS orders_per_hour
FROM
    orders
GROUP BY hours;
```

hours	orders_per_hour
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

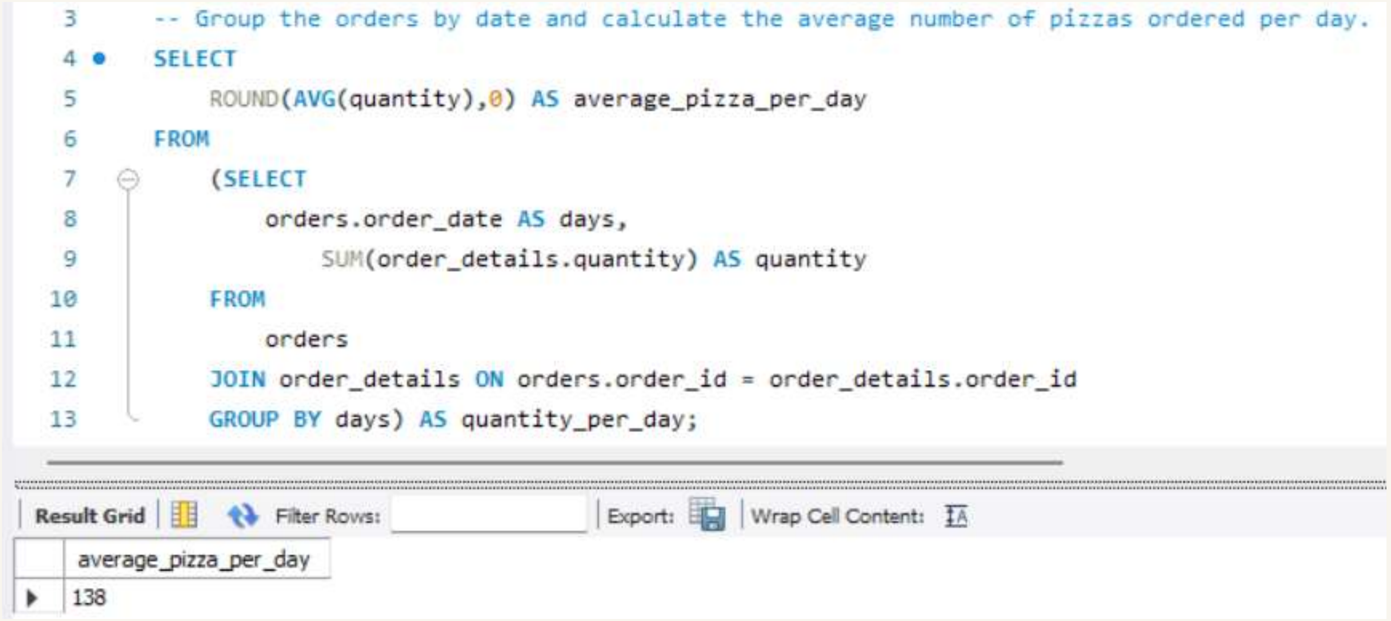
#8 Join relevant tables to find the category-wise distribution of pizzas



```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
SELECT
    category, COUNT(name) AS pizza_category_count
FROM
    pizza_types
GROUP BY category;
```

category	pizza_category_count
Chicken	6
Classic	8
Supreme	9
Veggie	9

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



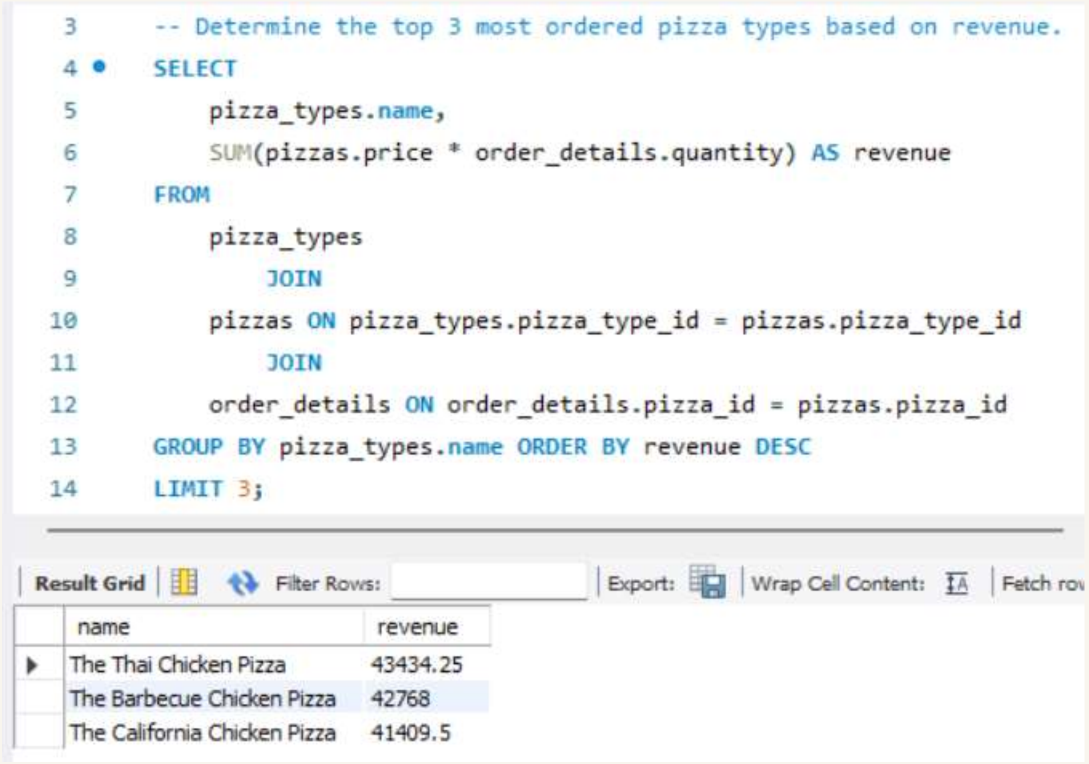
```
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity),0) AS average_pizza_per_day
FROM
    (SELECT
        orders.order_date AS days,
            SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY days) AS quantity_per_day;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
average_pizza_per_day
138
```

project by PURVA DEWANGAN

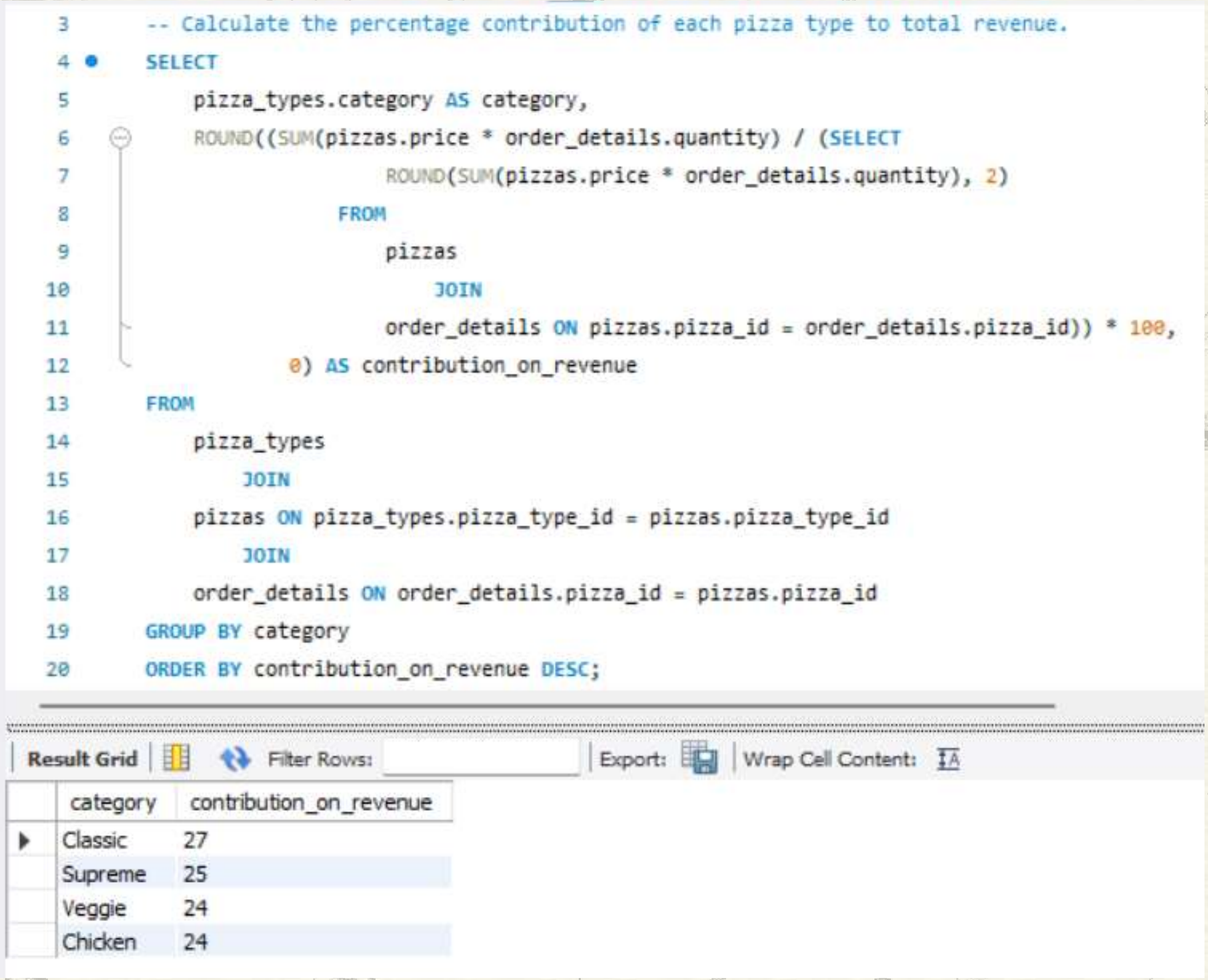
#10 Determine the top 3 most ordered pizza types based on revenue



```
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name ORDER BY revenue DESC
LIMIT 3;
```

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

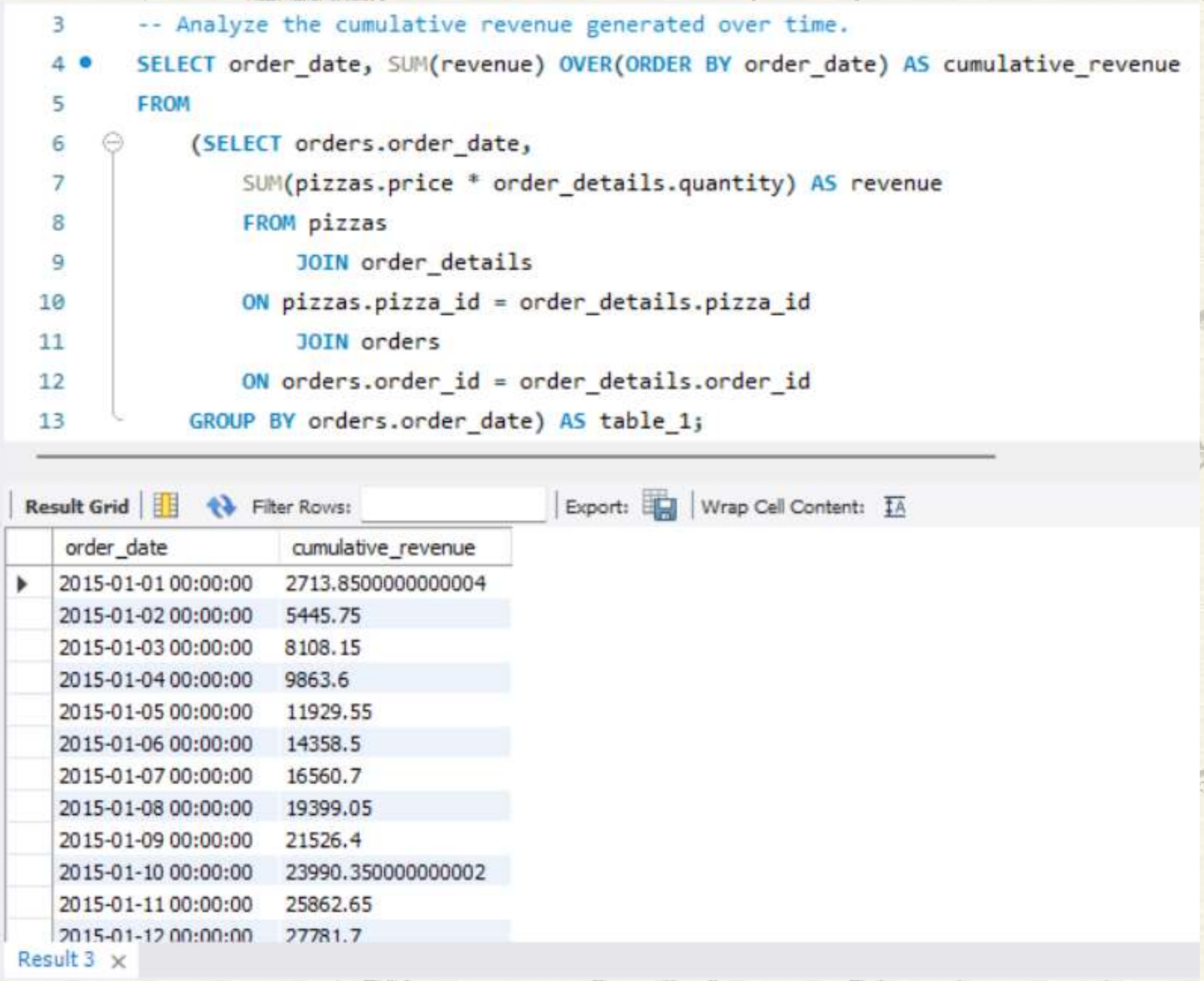
#11 Calculate the percentage contribution of each pizza type to total revenue



```
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category AS category,
    ROUND((SUM(pizzas.price * order_details.quantity) / (SELECT
                    ROUND(SUM(pizzas.price * order_details.quantity), 2)
                FROM
                    pizzas
                        JOIN
                    order_details ON pizzas.pizza_id = order_details.pizza_id)) * 100,
            0) AS contribution_on_revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY category
ORDER BY contribution_on_revenue DESC;
```

category	contribution_on_revenue
Classic	27
Supreme	25
Veggie	24
Chicken	24

#12
Analyze the cumulative revenue generated over time



```sql
-- Analyze the cumulative revenue generated over time.
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cumulative_revenue
FROM
    (SELECT orders.order_date,
        SUM(pizzas.price * order_details.quantity) AS revenue
        FROM pizzas
            JOIN order_details
        ON pizzas.pizza_id = order_details.pizza_id
            JOIN orders
        ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS table_1;
```

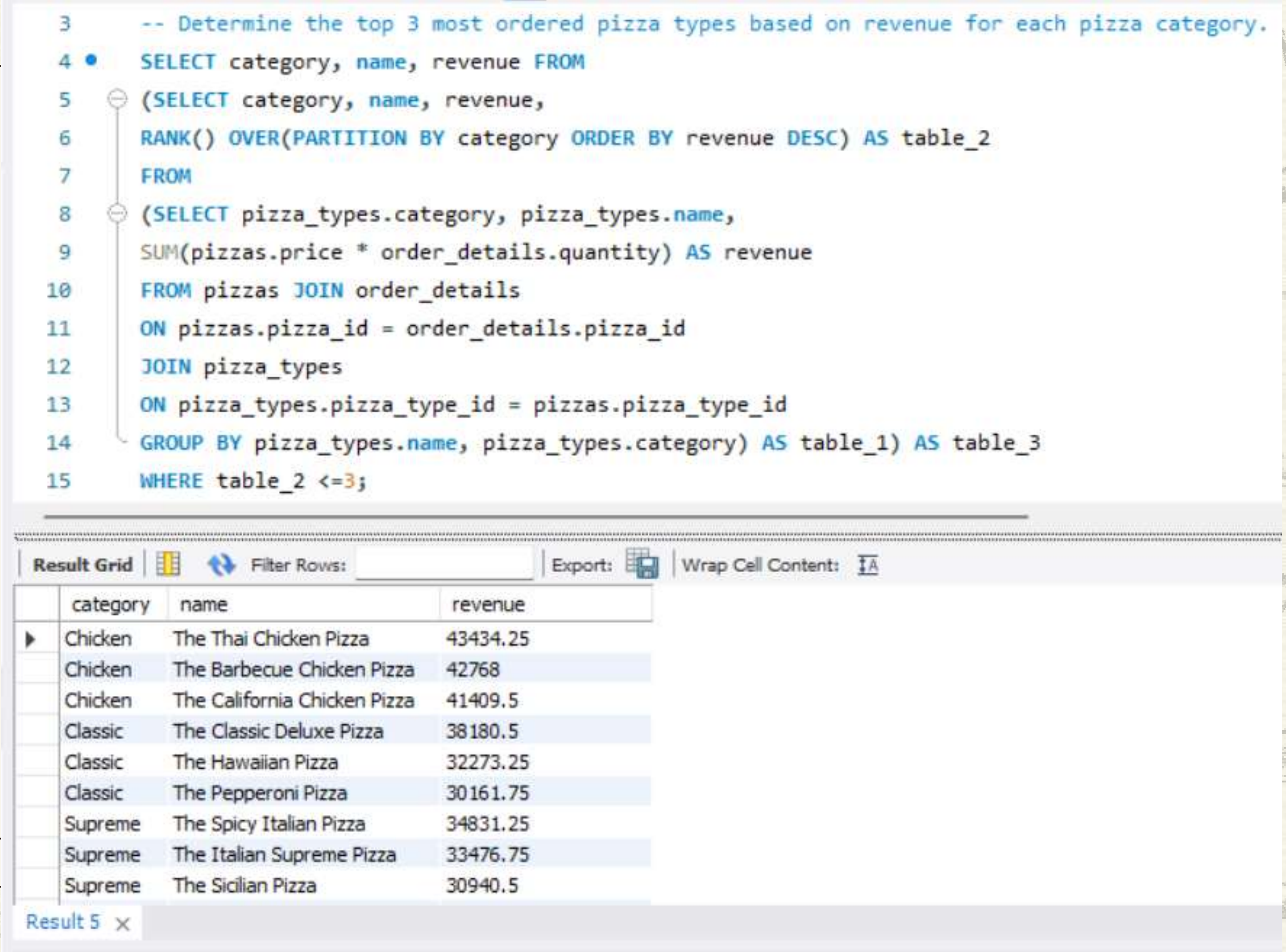
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	cumulative_revenue
2015-01-01 00:00:00	2713.8500000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55
2015-01-06 00:00:00	14358.5
2015-01-07 00:00:00	16560.7
2015-01-08 00:00:00	19399.05
2015-01-09 00:00:00	21526.4
2015-01-10 00:00:00	23990.350000000002
2015-01-11 00:00:00	25862.65
2015-01-12 00:00:00	27781.7

Result 3

project by PURVA DEWANGAN

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



```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT category, name, revenue FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS table_2
FROM
(SELECT pizza_types.category, pizza_types.name,
SUM(pizzas.price * order_details.quantity) AS revenue
FROM pizzas JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
JOIN pizza_types
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name, pizza_types.category) AS table_1) AS table_3
WHERE table_2 <=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Result 5