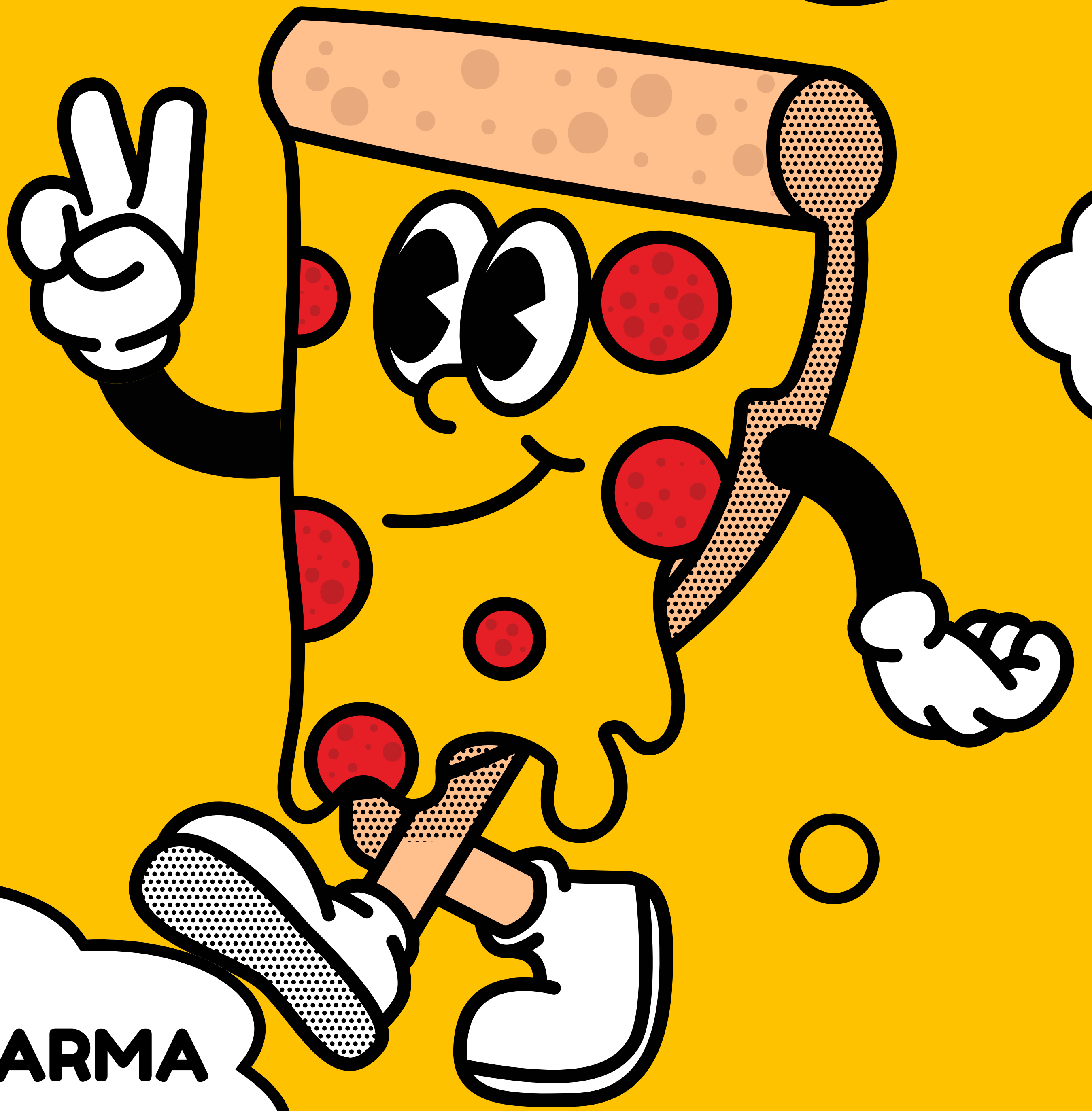
PIZZA HUT

SALES ANALYSIS

BY - AMAN SHARMA



WELCOME TO PIZZA DAY!

INTRODUCTION

IN TODAY’S FAST-PACED FOOD SERVICE INDUSTRY, LEVERAGING DATA ANALYTICS IS ESSENTIAL FOR DRIVING BUSINESS GROWTH AND OPERATIONAL EFFICIENCY. THIS PROJECT FOCUSES ON ANALYZING PIZZA HUT’S SALES DATA USING SQL TO EXTRACT MEANINGFUL INSIGHTS AND SUPPORT DATA-DRIVEN DECISION-MAKING.

THE PRIMARY OBJECTIVES OF THIS ANALYSIS ARE:

✅ UNDERSTANDING SALES TRENDS AND IDENTIFYING REVENUE PATTERNS.
✅ DETERMINING CUSTOMER PREFERENCES BY ANALYZING ORDER DATA.
✅ EVALUATING STORE PERFORMANCE BASED ON VARIOUS METRICS.
✅ UTILIZING SQL QUERIES TO GENERATE ACTIONABLE INSIGHTS THAT ENHANCE BUSINESS STRATEGIES.

TO ACHIEVE THESE OBJECTIVES, I SOLVED A SERIES OF BASIC, INTERMEDIATE, AND ADVANCED-LEVEL SQL QUERIES TO UNCOVER KEY TRENDS, OPTIMIZE SALES STRATEGIES, AND IMPROVE CUSTOMER SATISFACTION. THIS PROJECT HIGHLIGHTS THE POWER OF STRUCTURED QUERY LANGUAGE (SQL) IN TRANSFORMING RAW SALES DATA INTO VALUABLE BUSINESS INTELLIGENCE.

DATA OVERVIEW

This project utilizes Pizza Hut sales data, structured across multiple tables to facilitate an in-depth SQL-driven analysis. The dataset captures critical aspects of customer orders, pizza details, and sales performance.

Dataset Description:

- Number of Records: Extensive data covering multiple orders and sales transactions.
- Number of Attributes: Multiple columns representing key business metrics.
- Time Period Covered: Data spans a significant duration, enabling trend analysis.
- Data Source: Publicly available dataset curated for sales analytics.

Key Tables & Attributes:

✅ Orders Table – Contains order IDs, timestamps, and transactional details.
✅ Pizza Table – Includes pizza names, categories, sizes, and pricing.
✅ Order Details Table – Links orders with pizza details, tracking quantities and revenue.
✅ Customer Table (if applicable) – Stores customer demographics and purchasing patterns.

Data Cleaning & Preprocessing:

✓ Removed duplicate entries to maintain accuracy.
✓ Standardized date-time formats for consistent analysis.
✓ Addressed missing values to prevent data inconsistencies.
✓ Optimized table relationships for efficient querying.

This dataset provides a solid foundation for sales trend analysis, customer preference identification, and revenue optimization.

PROJECT OBJECTIVE

Project Objectives

This project utilizes SQL-based sales analysis to extract actionable insights from Pizza Hut's sales data, focusing on key business metrics and performance trends. The primary objectives include:

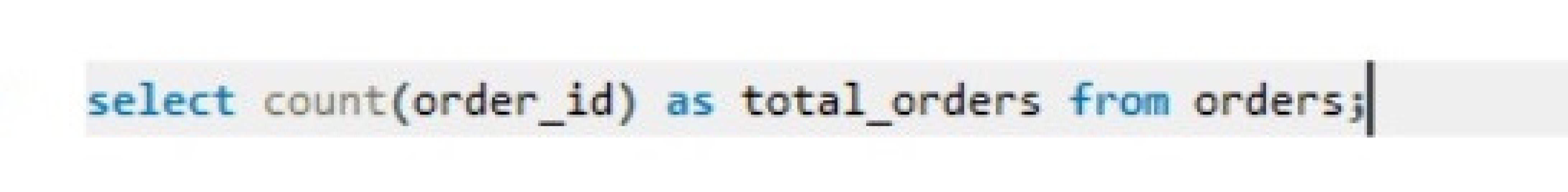
✅ Basic Analysis – Determine total orders, revenue, highest-priced pizza, most popular size, and top-selling pizzas.

✅ Intermediate Analysis – Assess category-wise sales, order distribution by time, daily order trends, and top revenue-generating pizzas.

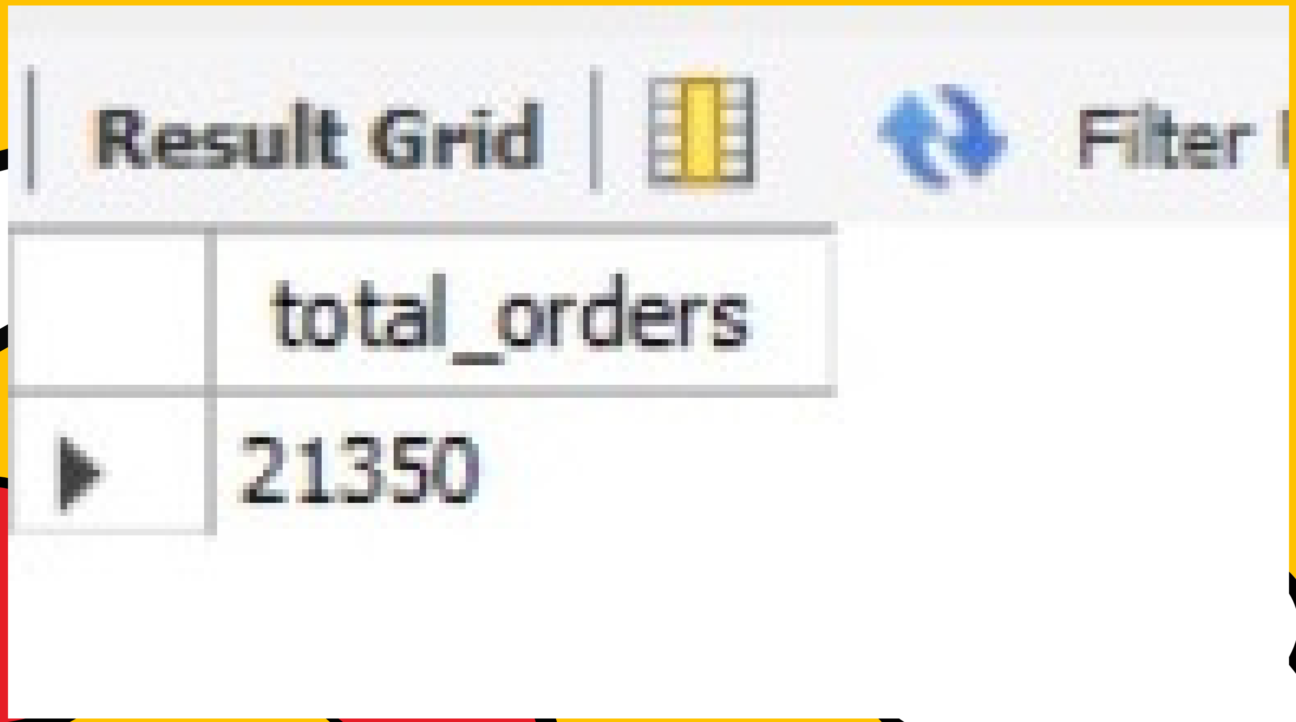
✅ Advanced Analysis – Evaluate revenue contribution by pizza type, cumulative revenue trends, and category-wise top performers.

These insights drive data-backed decision-making, helping optimize sales strategies and operational efficiency.

-RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

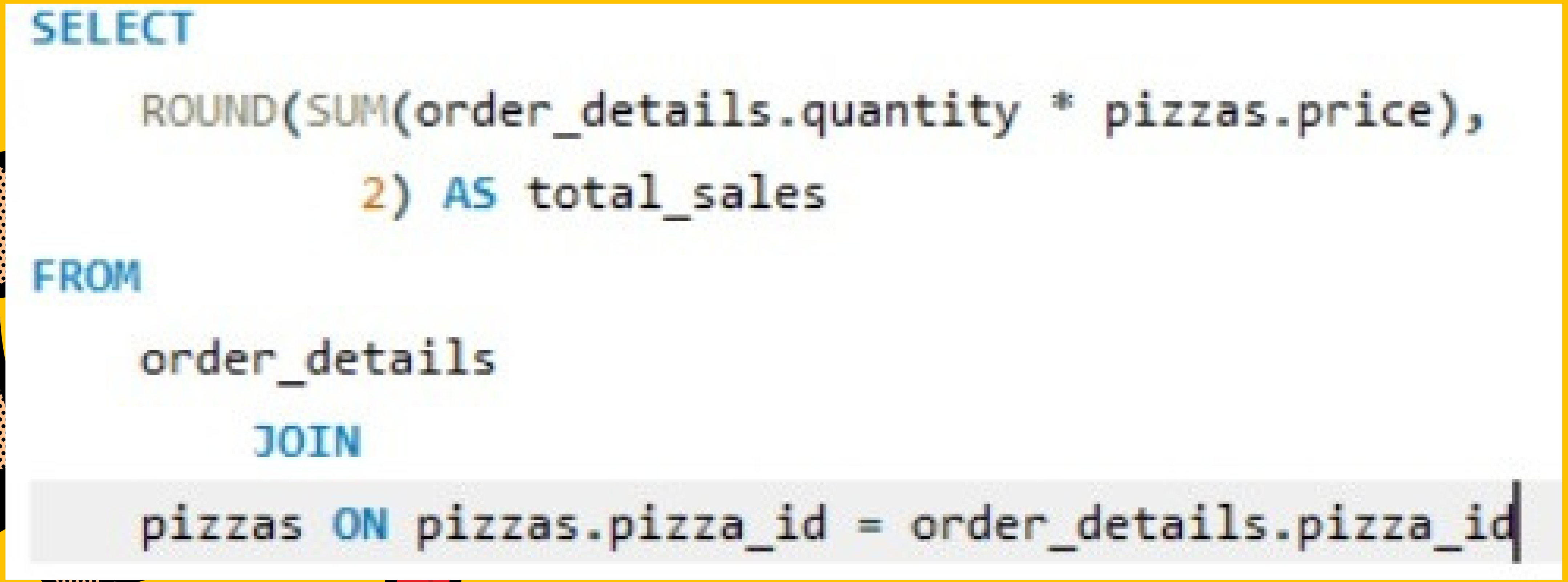


```
select count(order_id) as total_orders from orders;
```

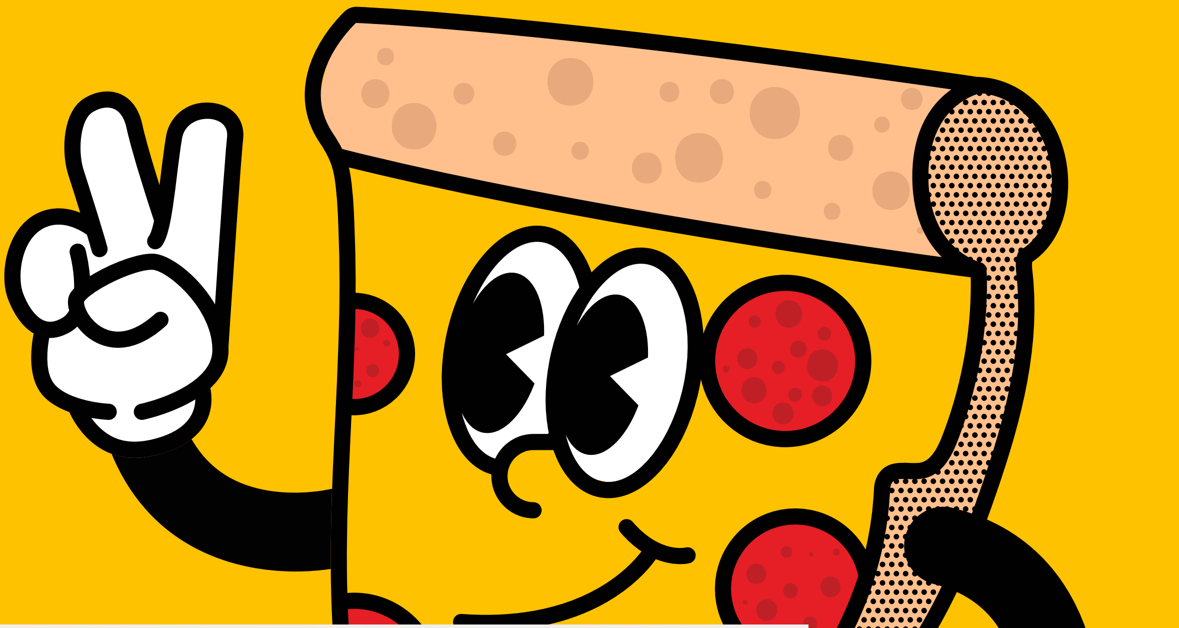


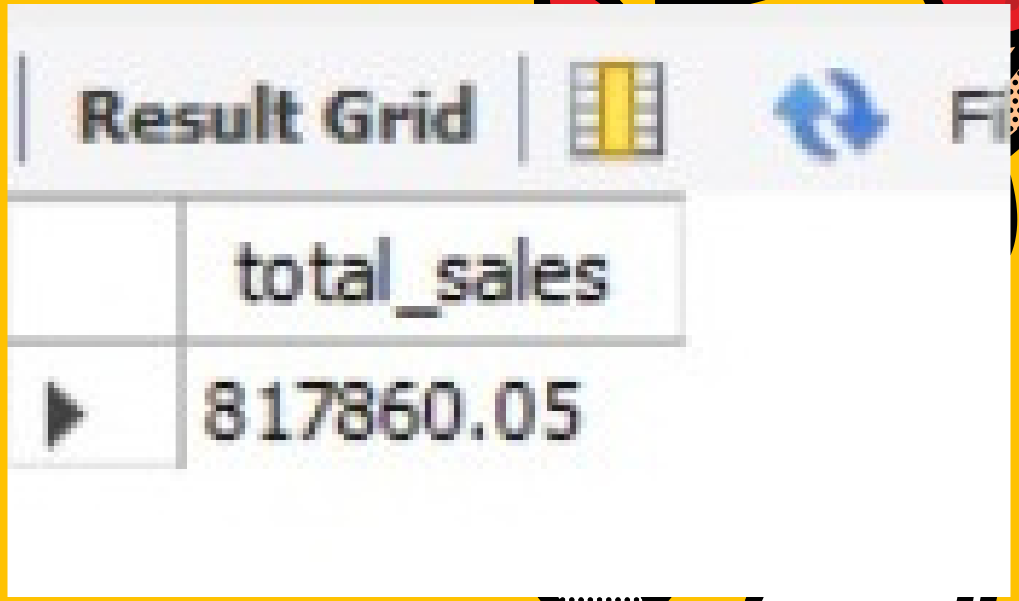
```
total_orders
21350
```

-CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



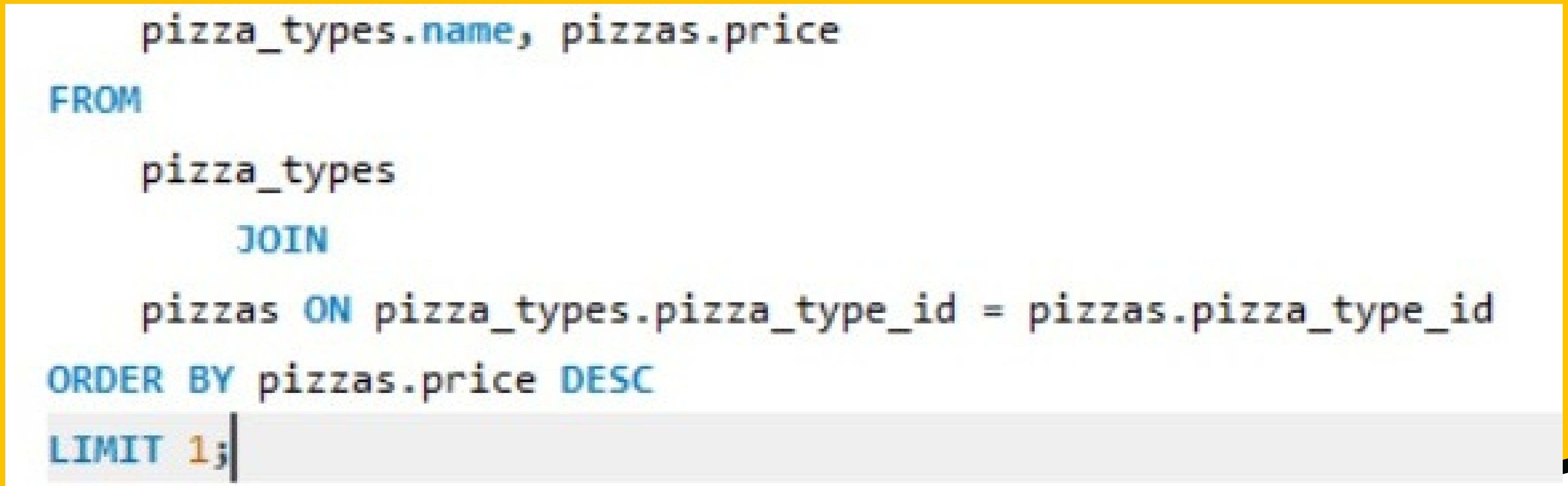
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```



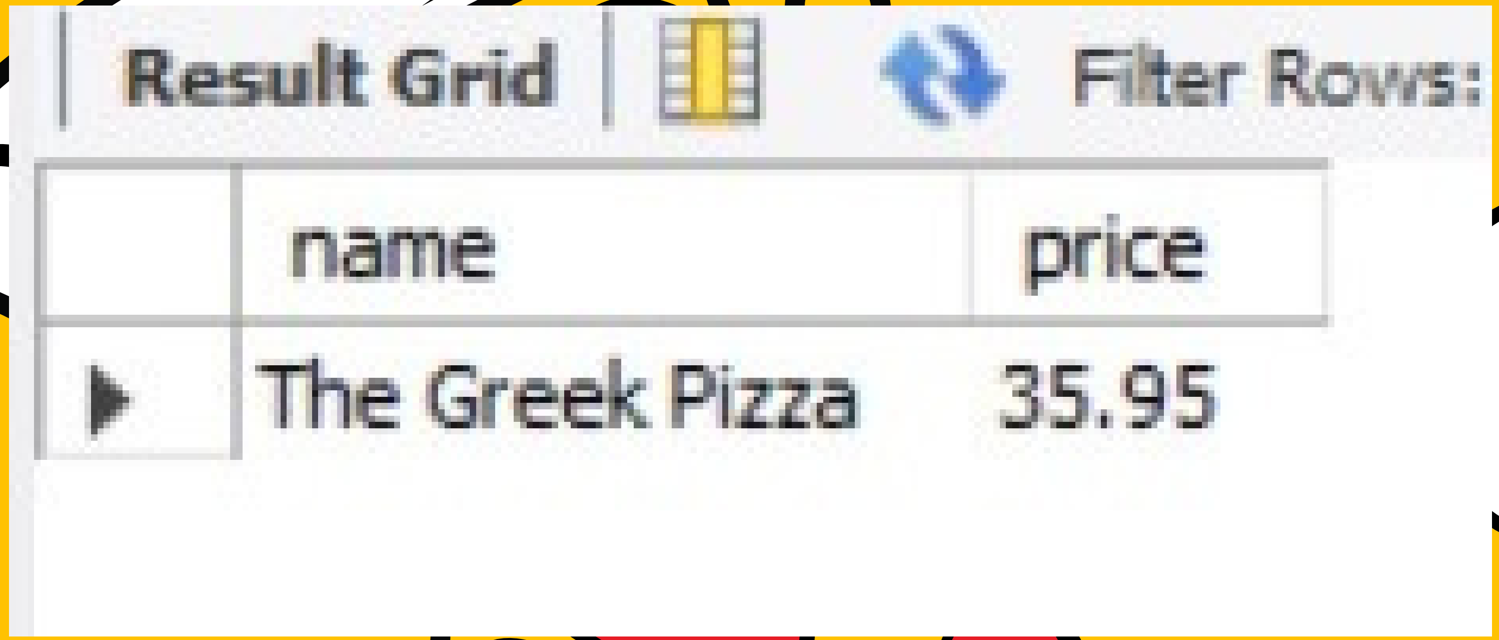


```
total_sales
817860.05
```

-IDENTIFY THE HIGHEST-PRICED PIZZA.

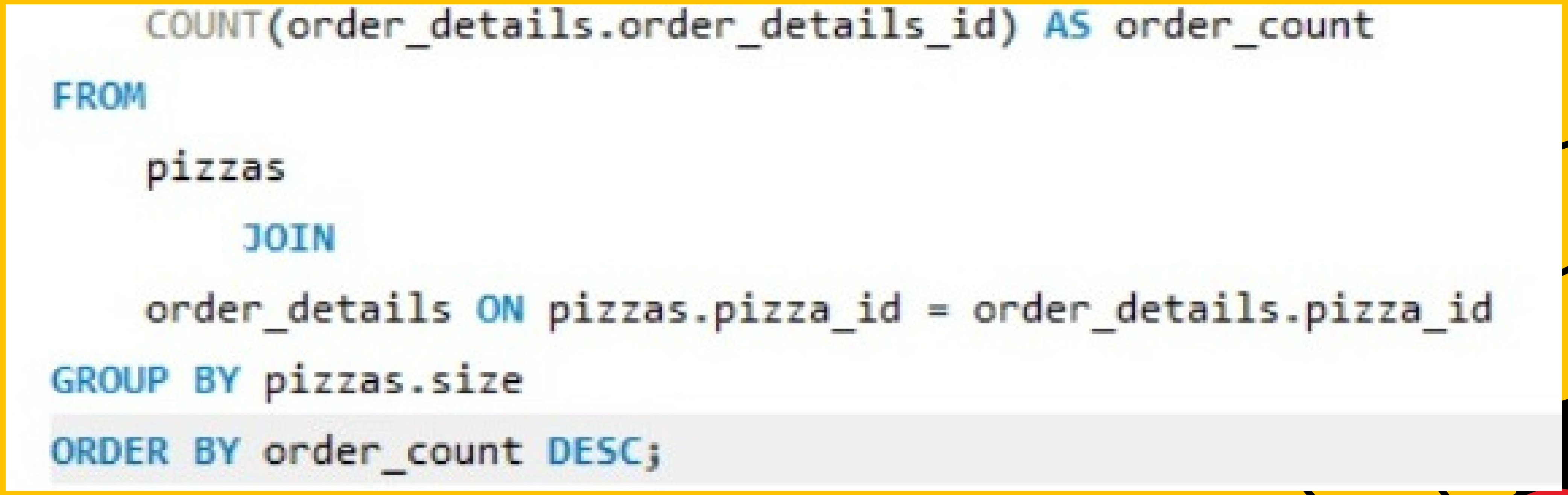


```
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

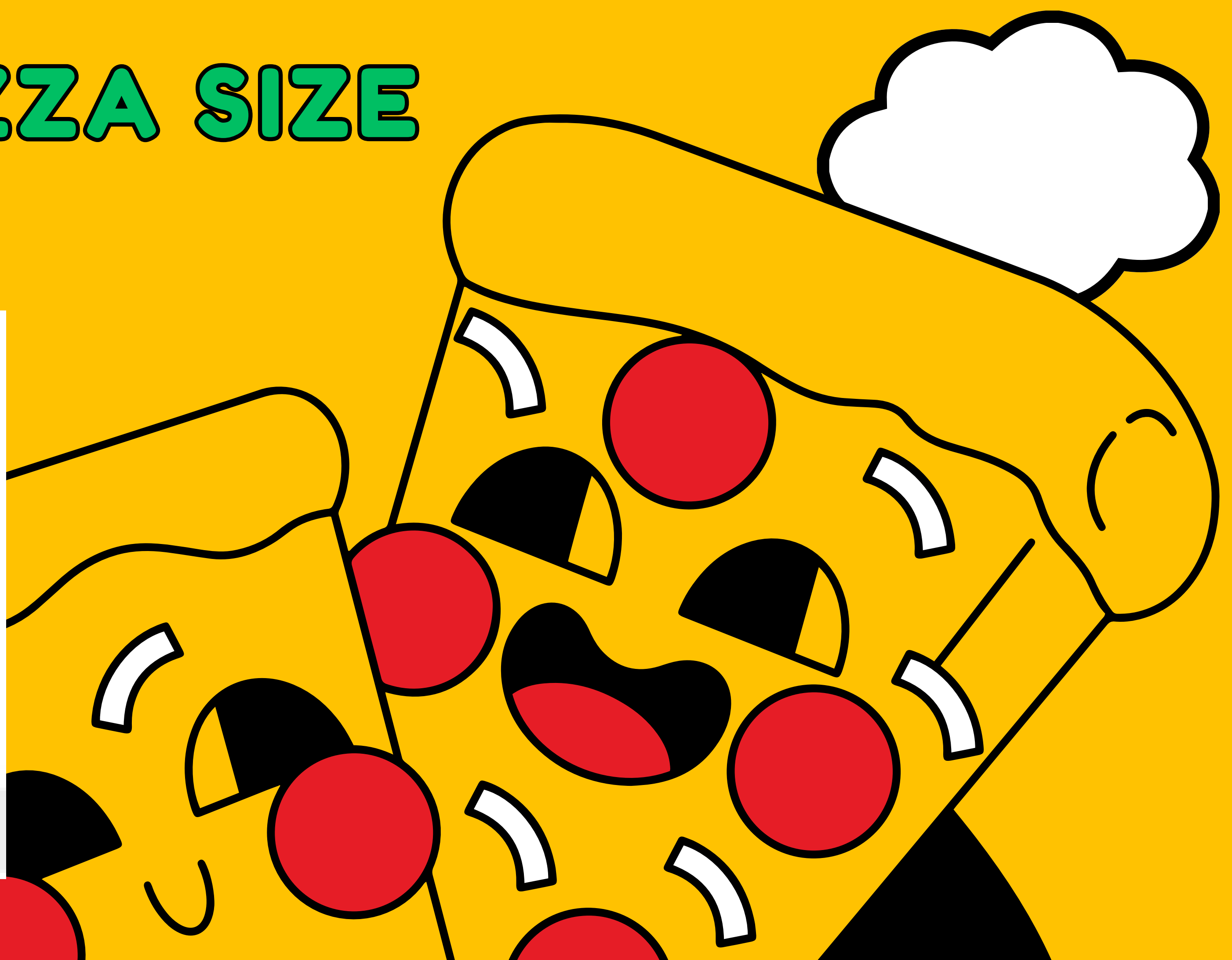


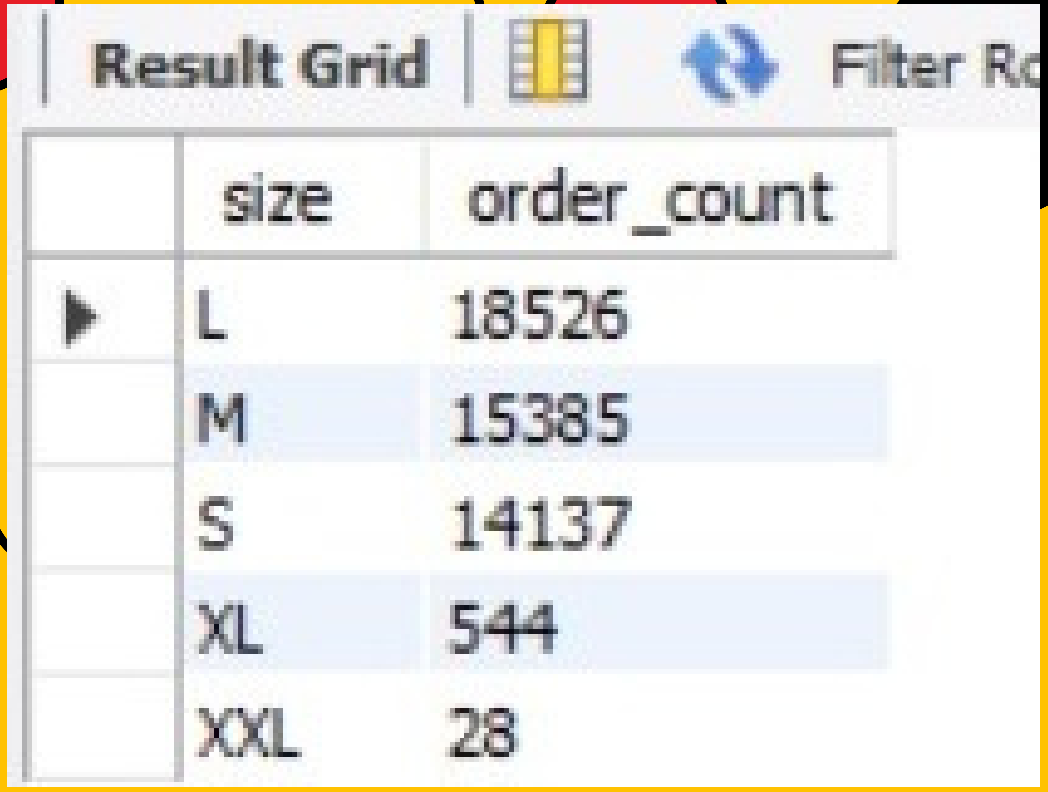
	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED



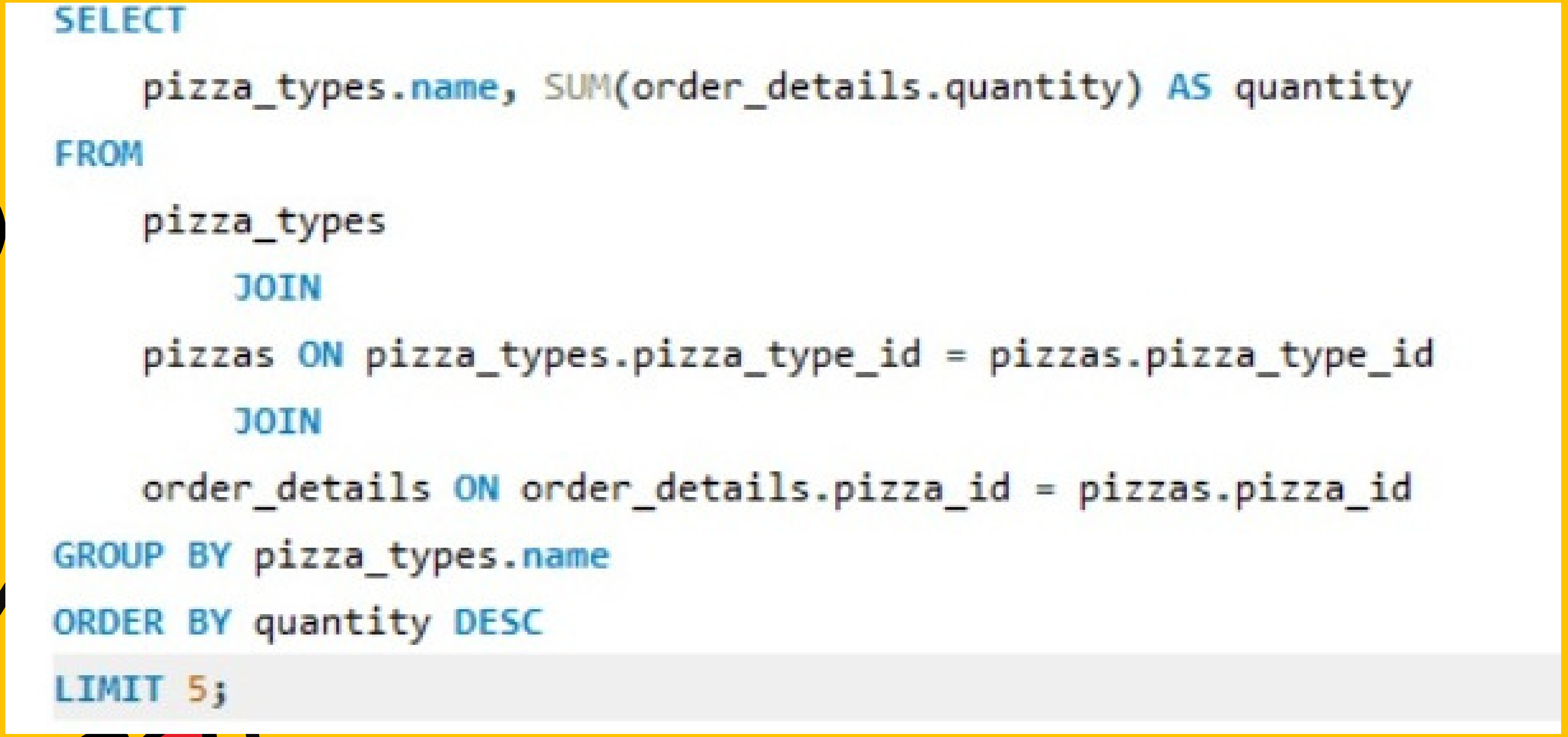
```
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



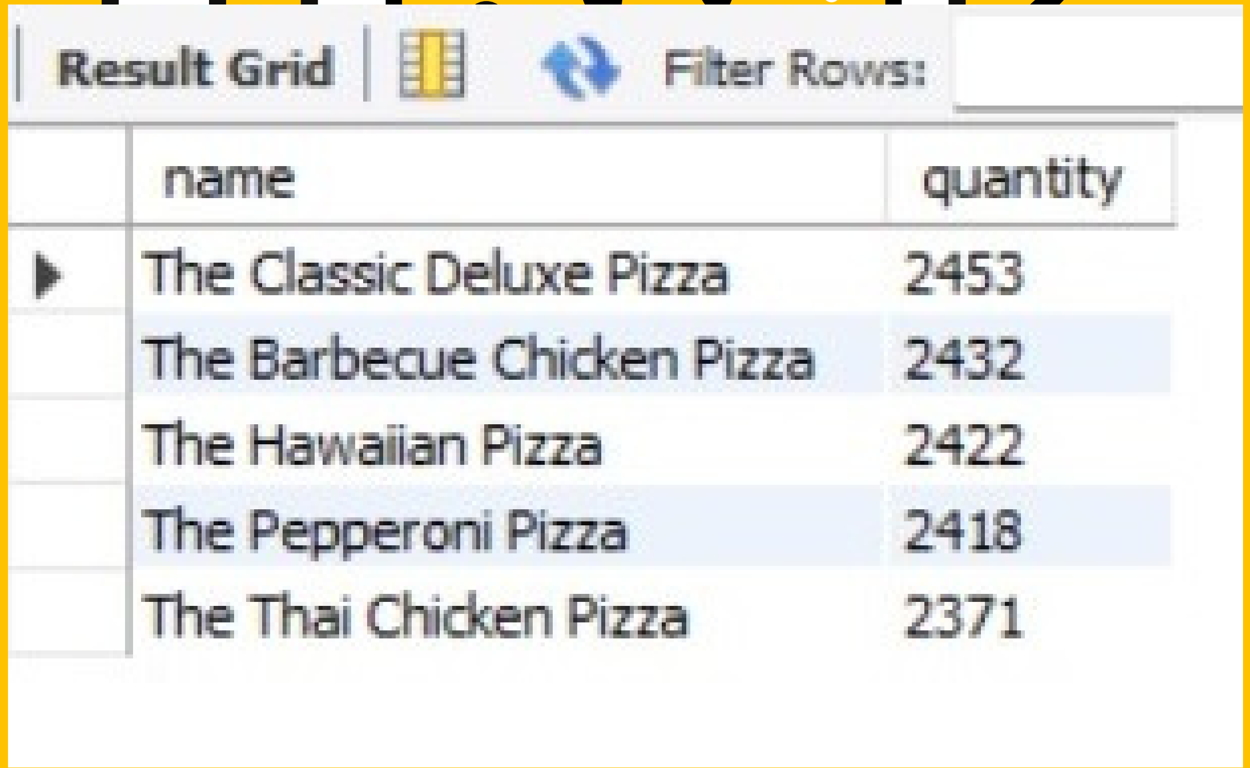


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

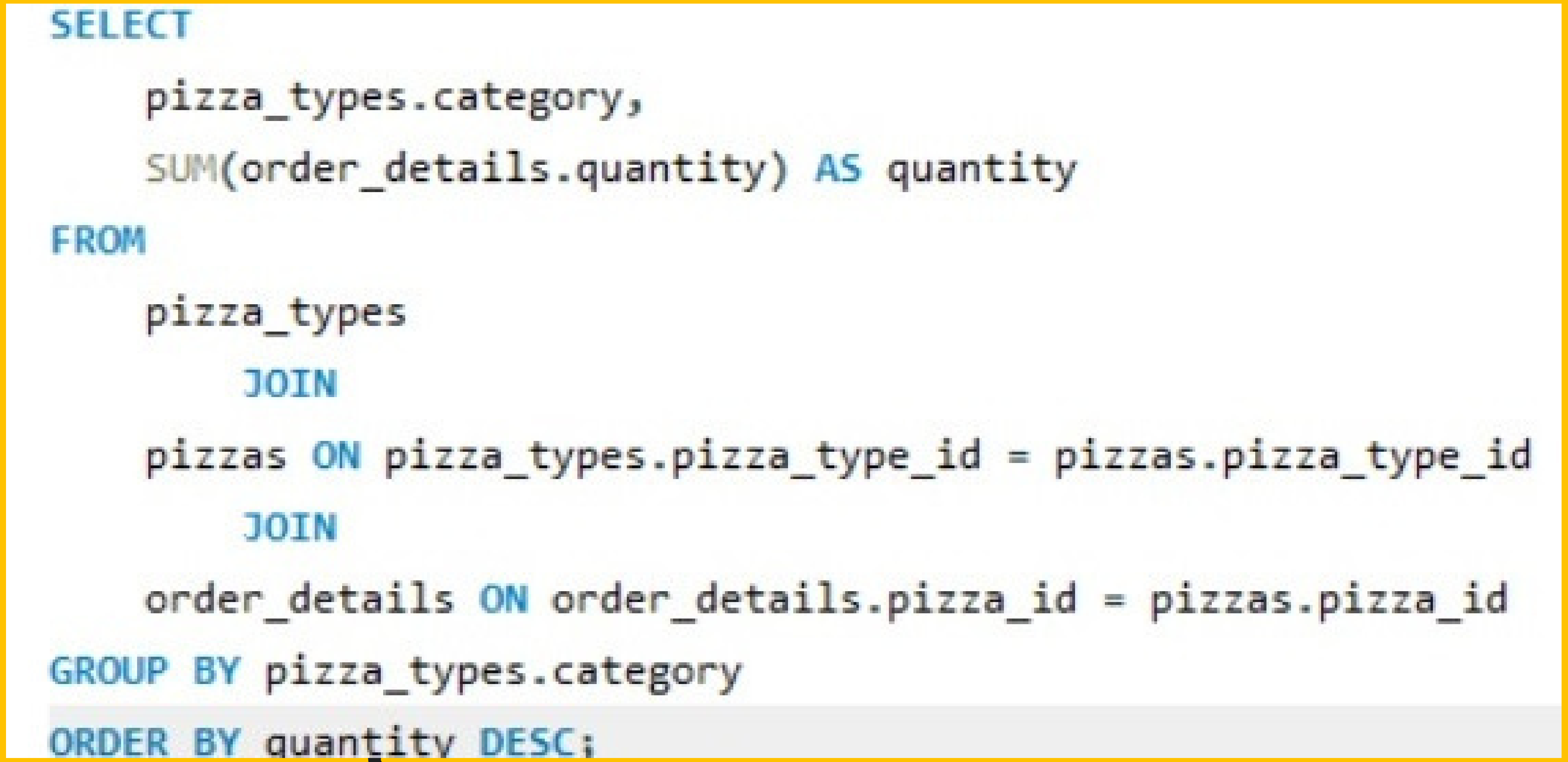


```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

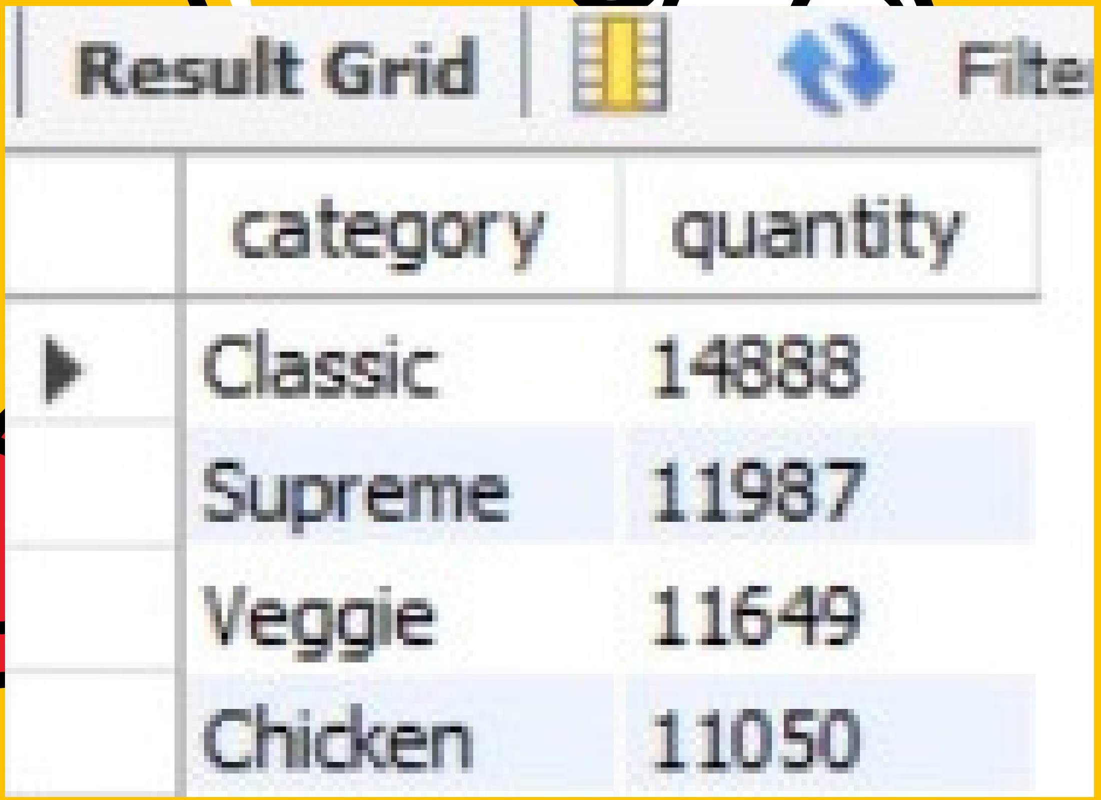


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

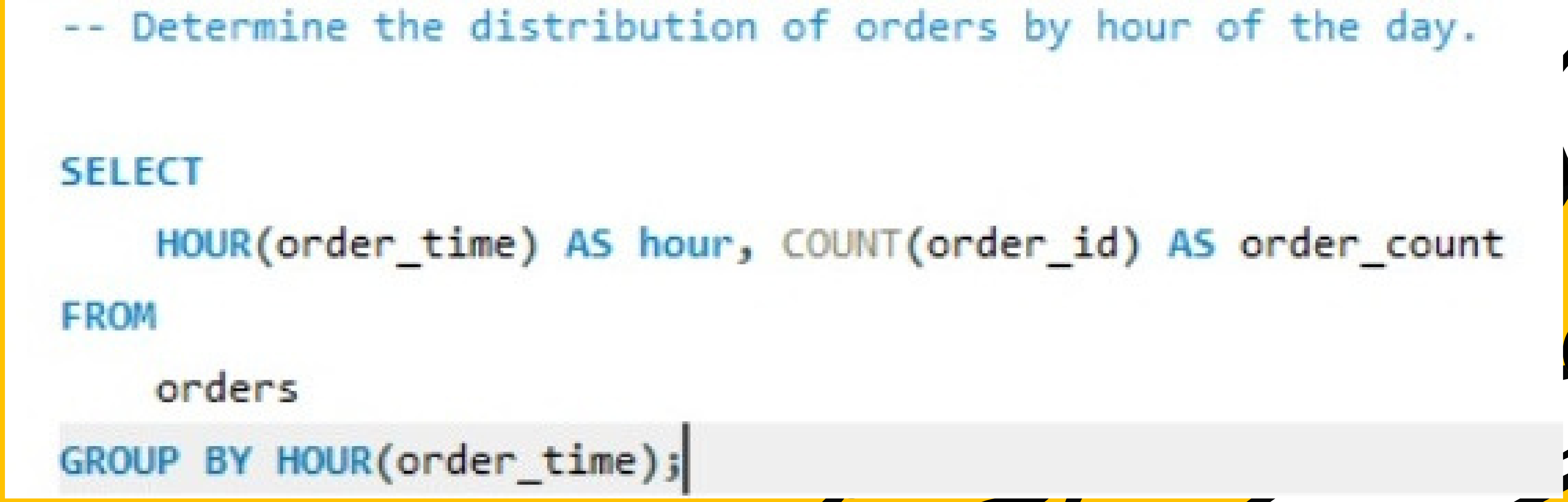


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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



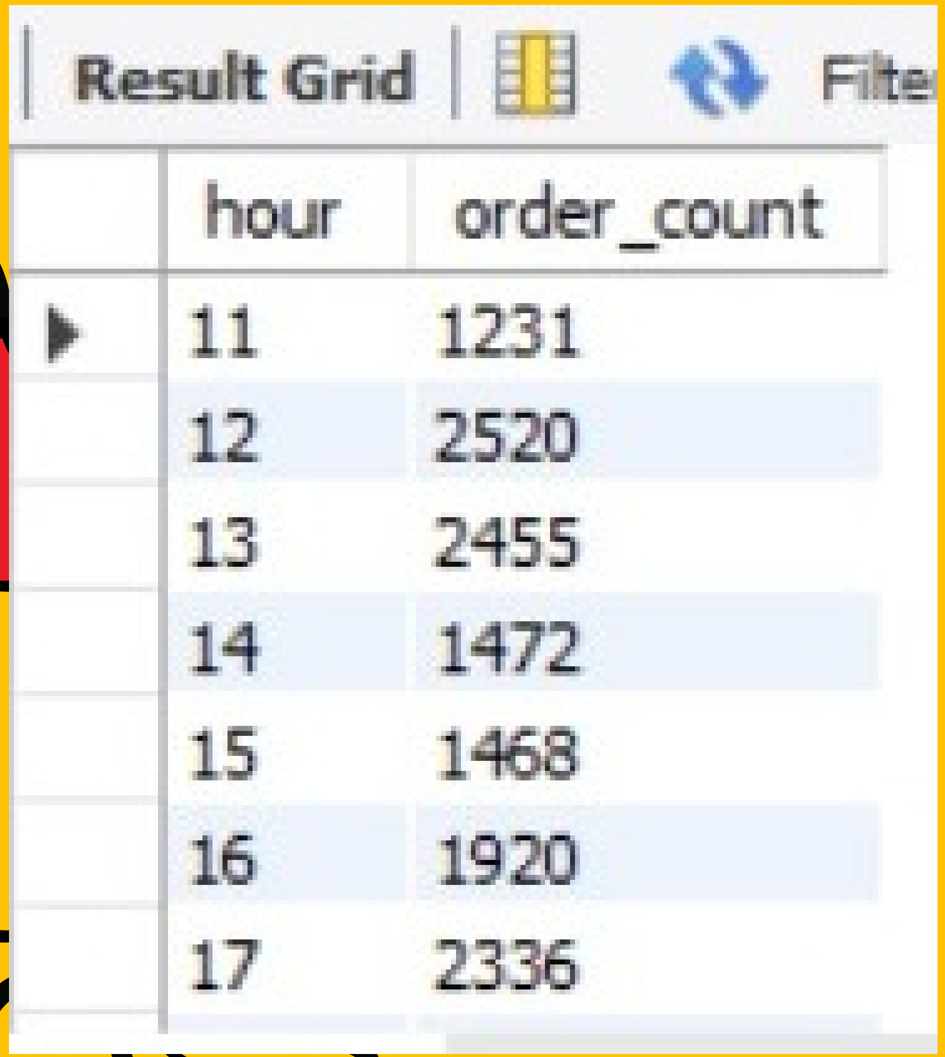
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



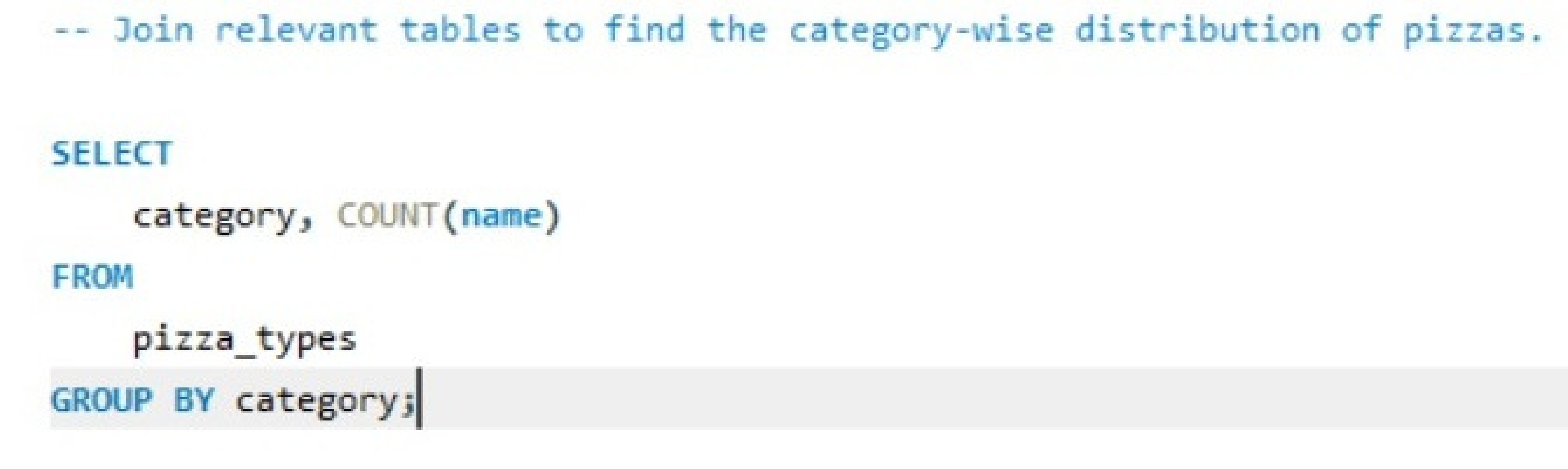
```
-- Determine the distribution of orders by hour of the day.


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



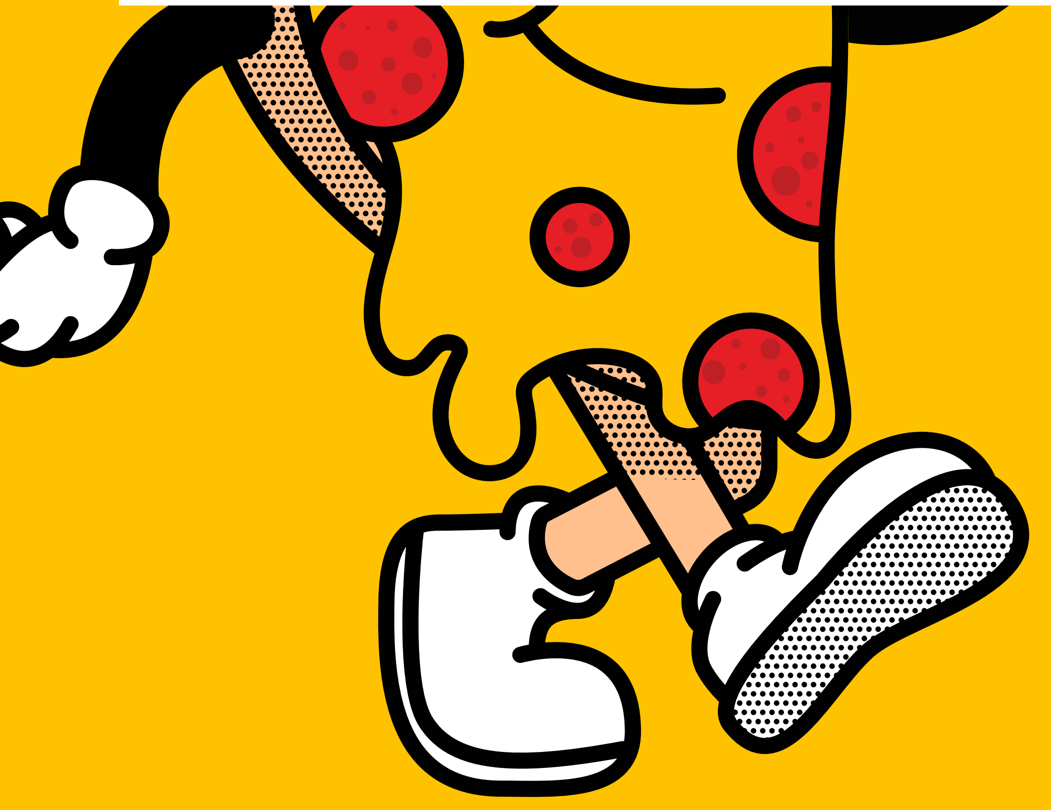
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

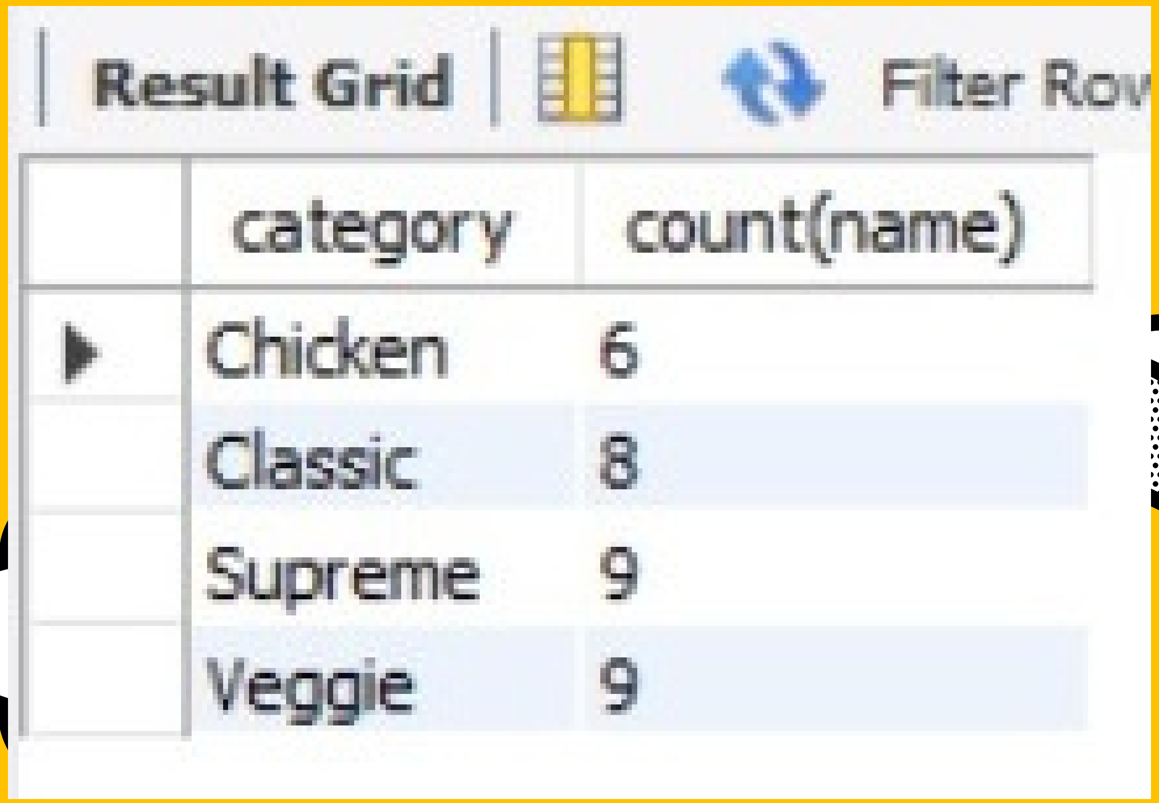
JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS



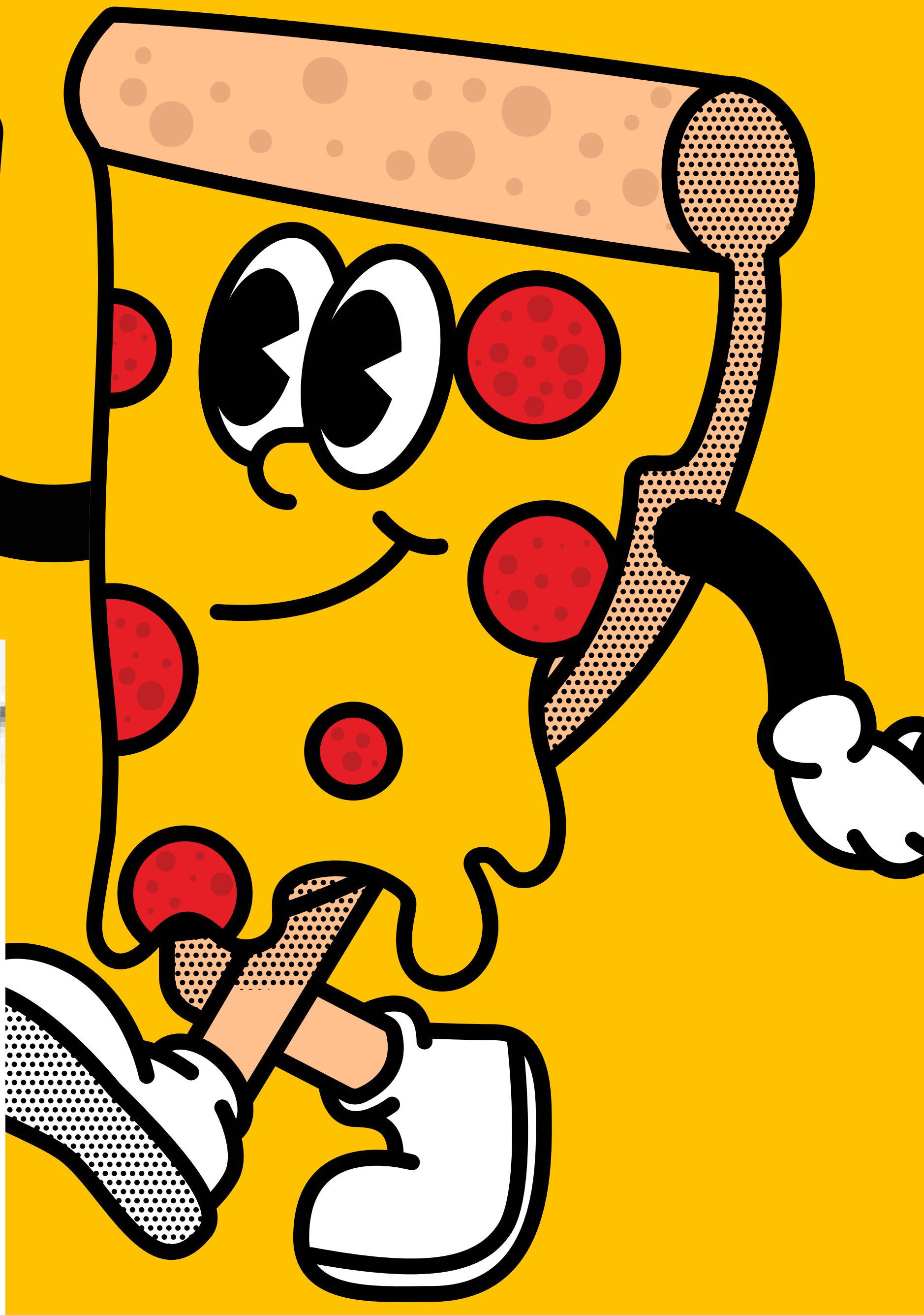
```
-- Join relevant tables to find the category-wise distribution of pizzas.


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

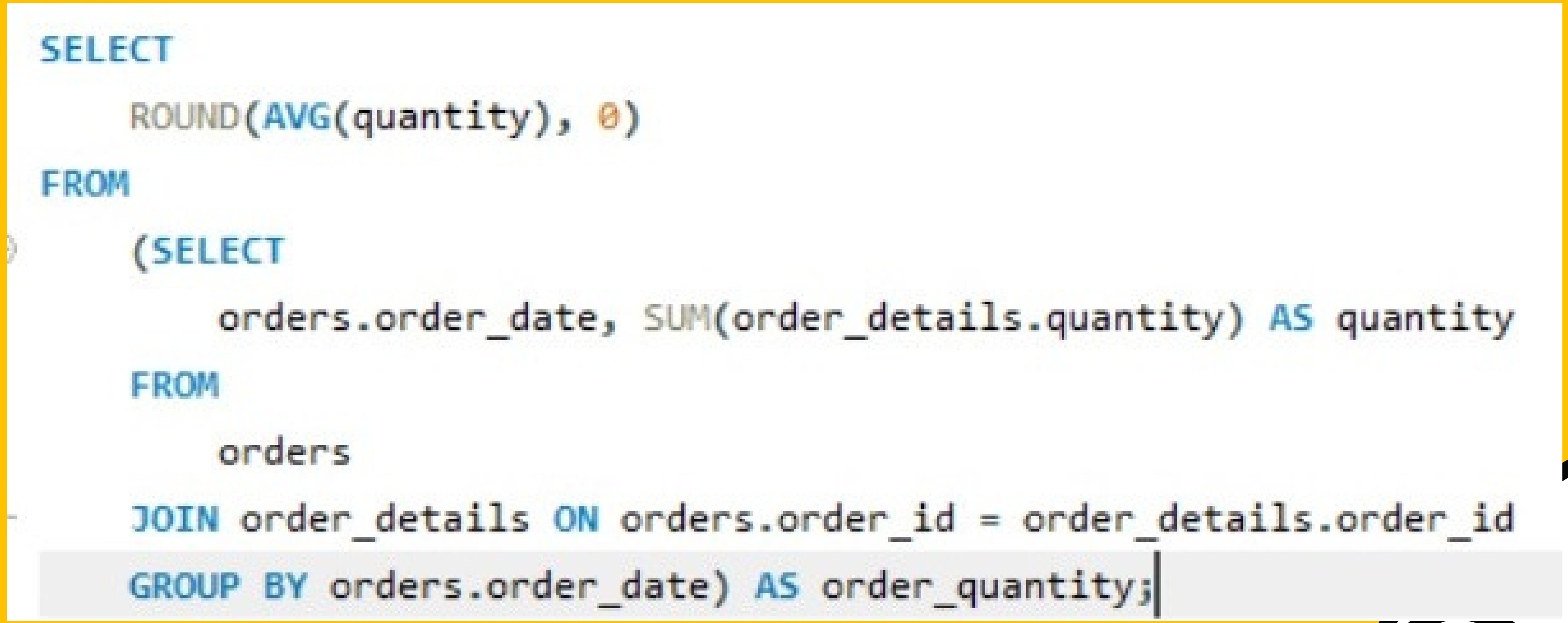




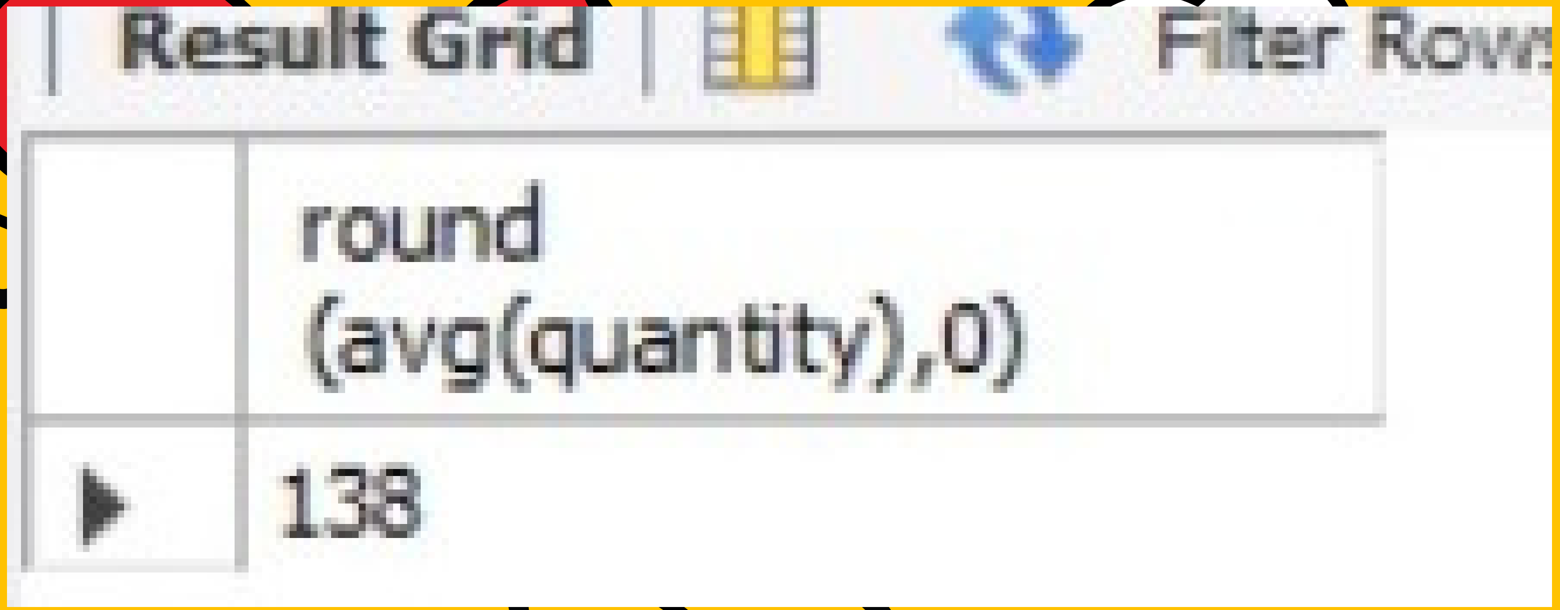
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

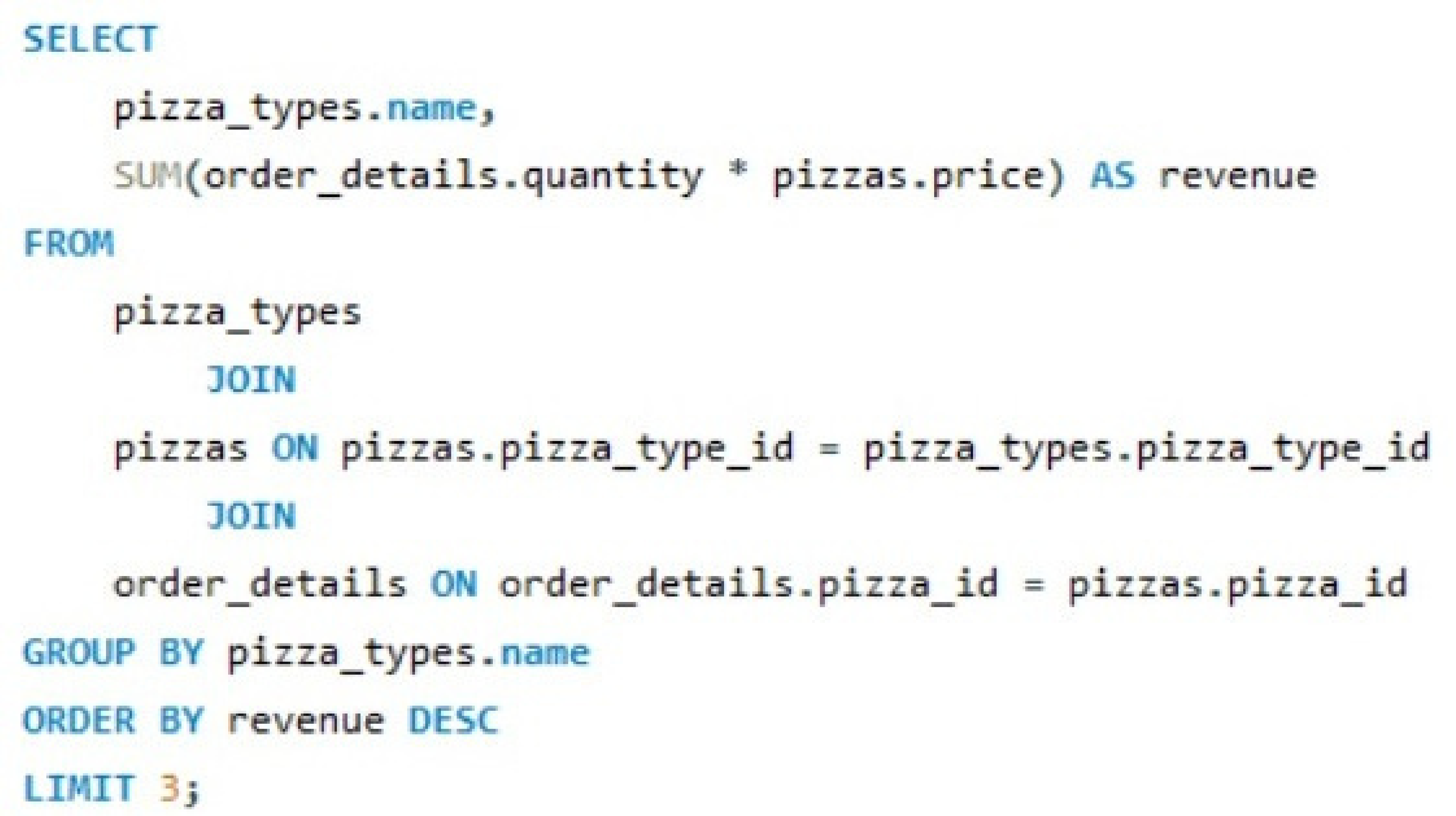


```
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

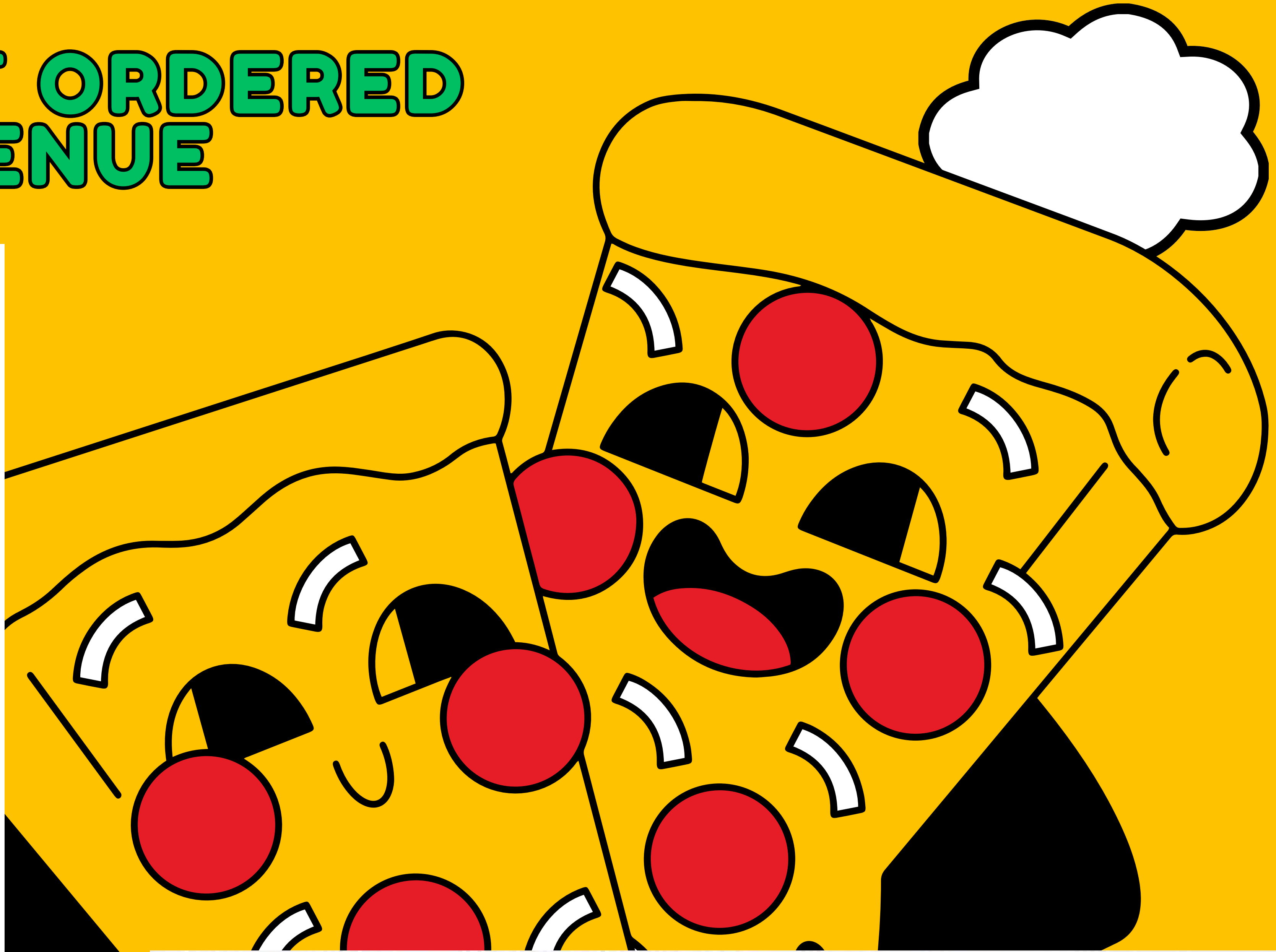


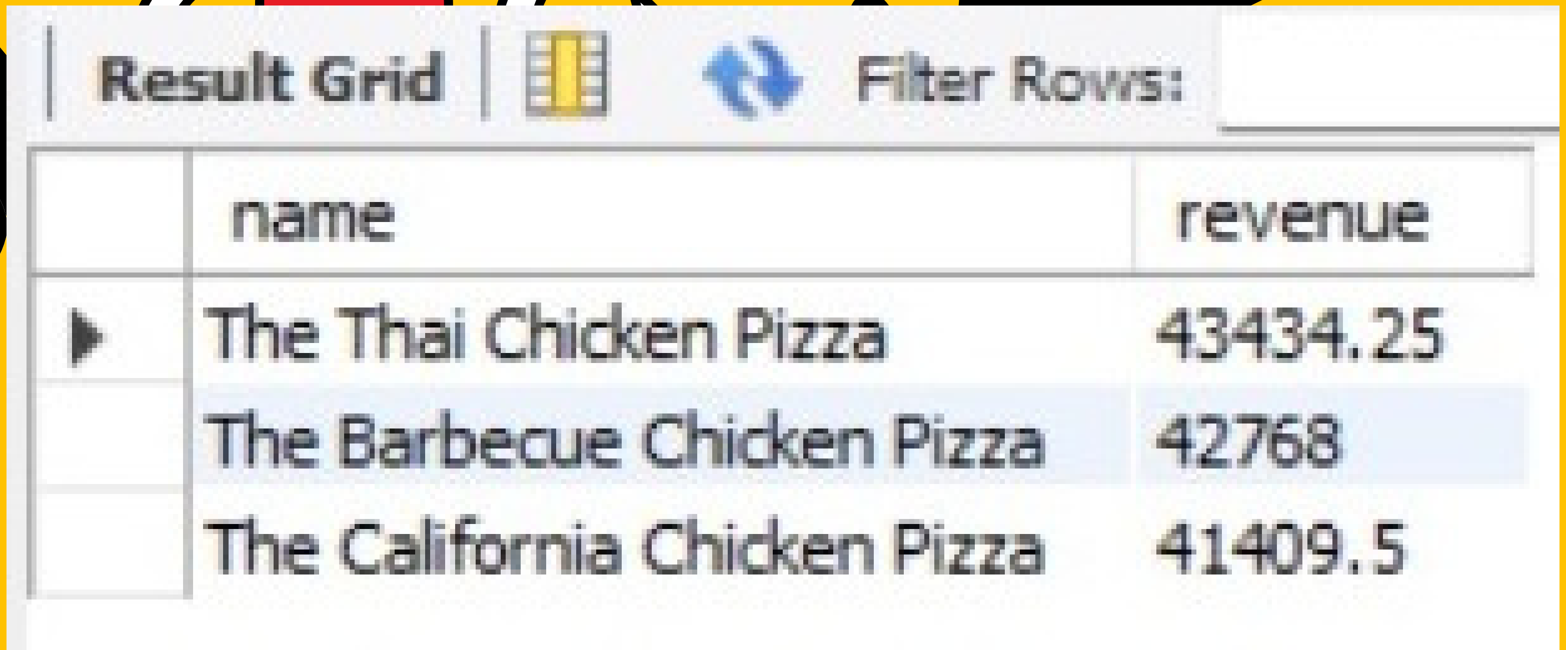
```
round
(avg(quantity),0)
138
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE



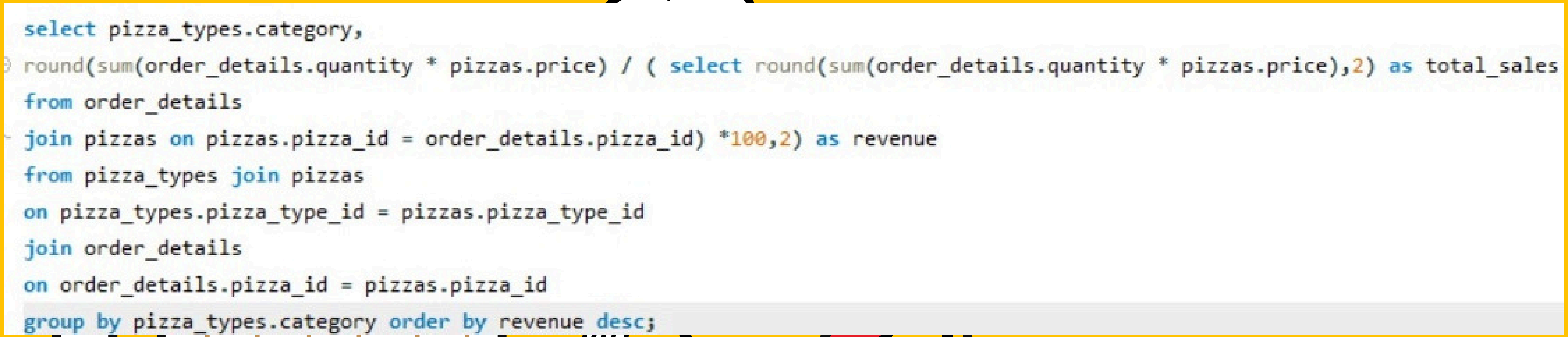
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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



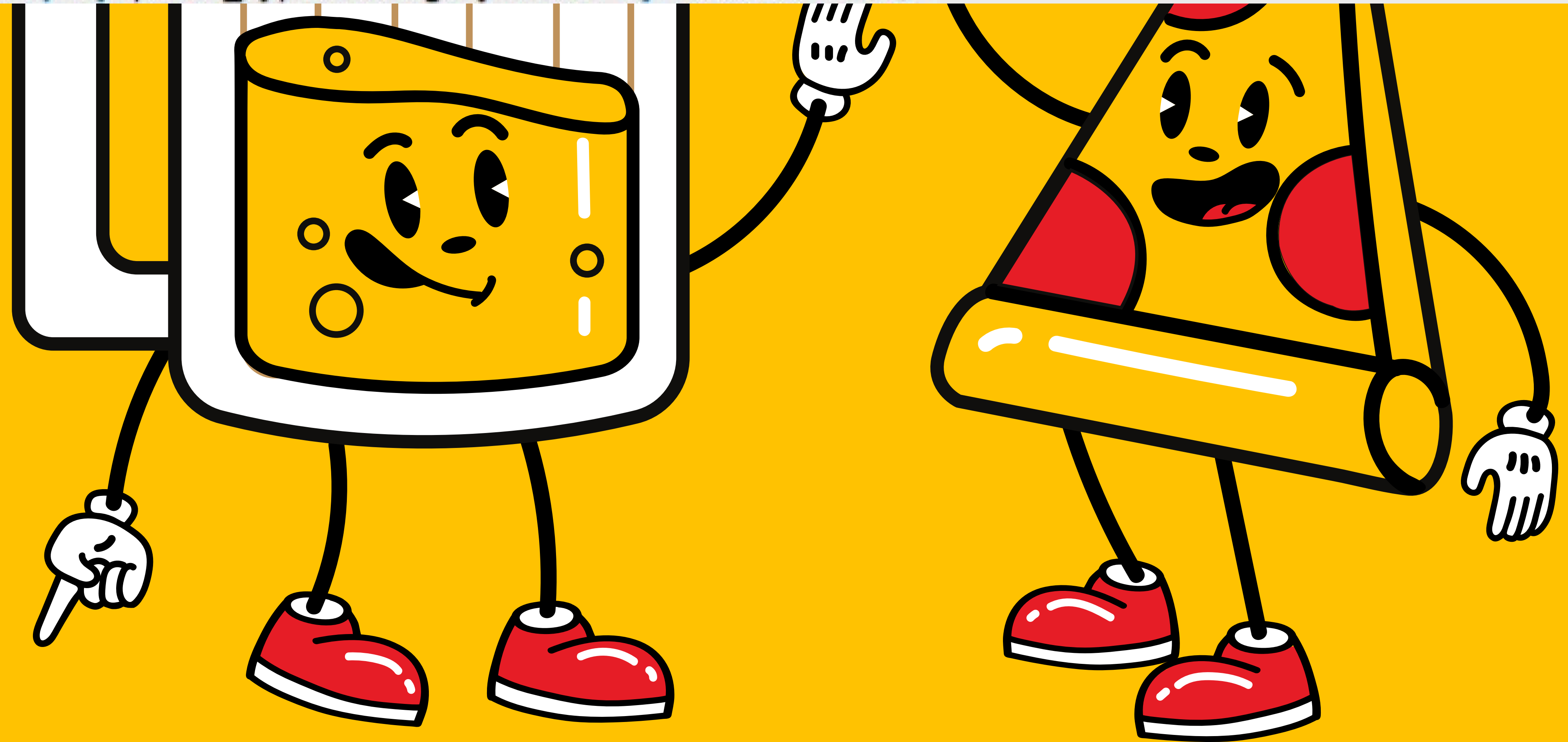


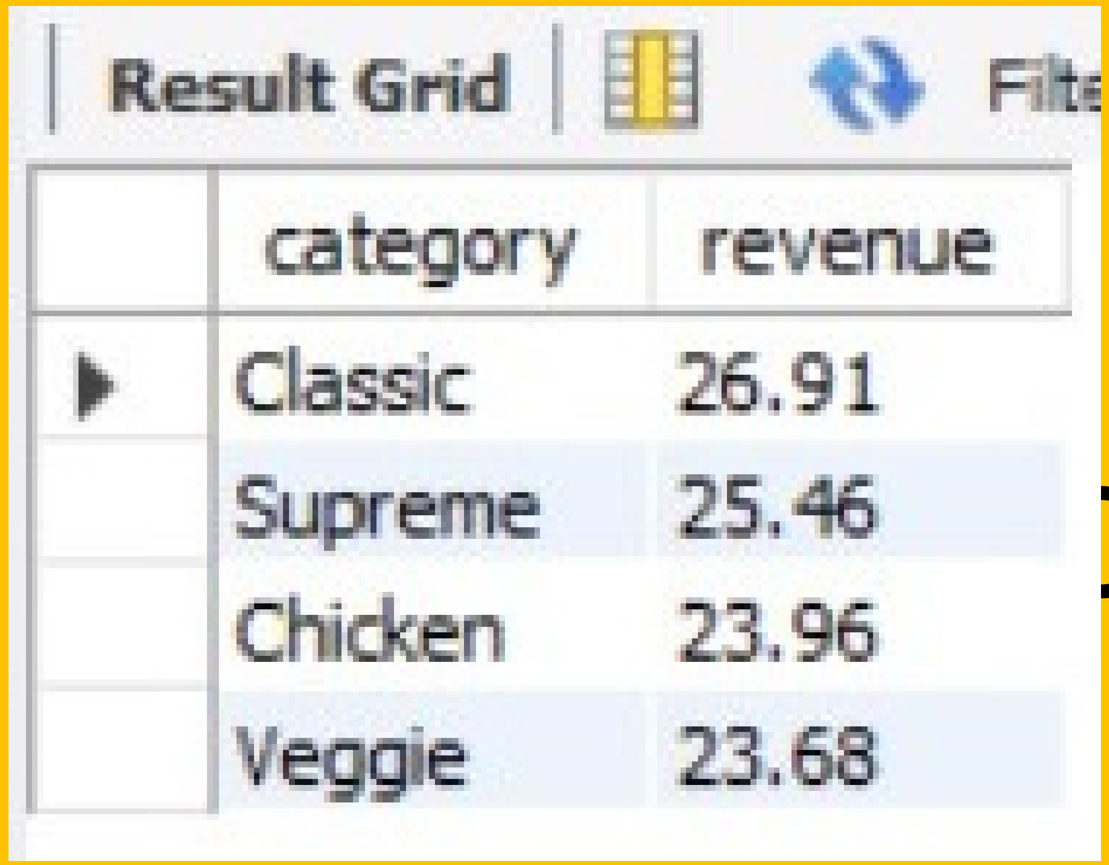
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE



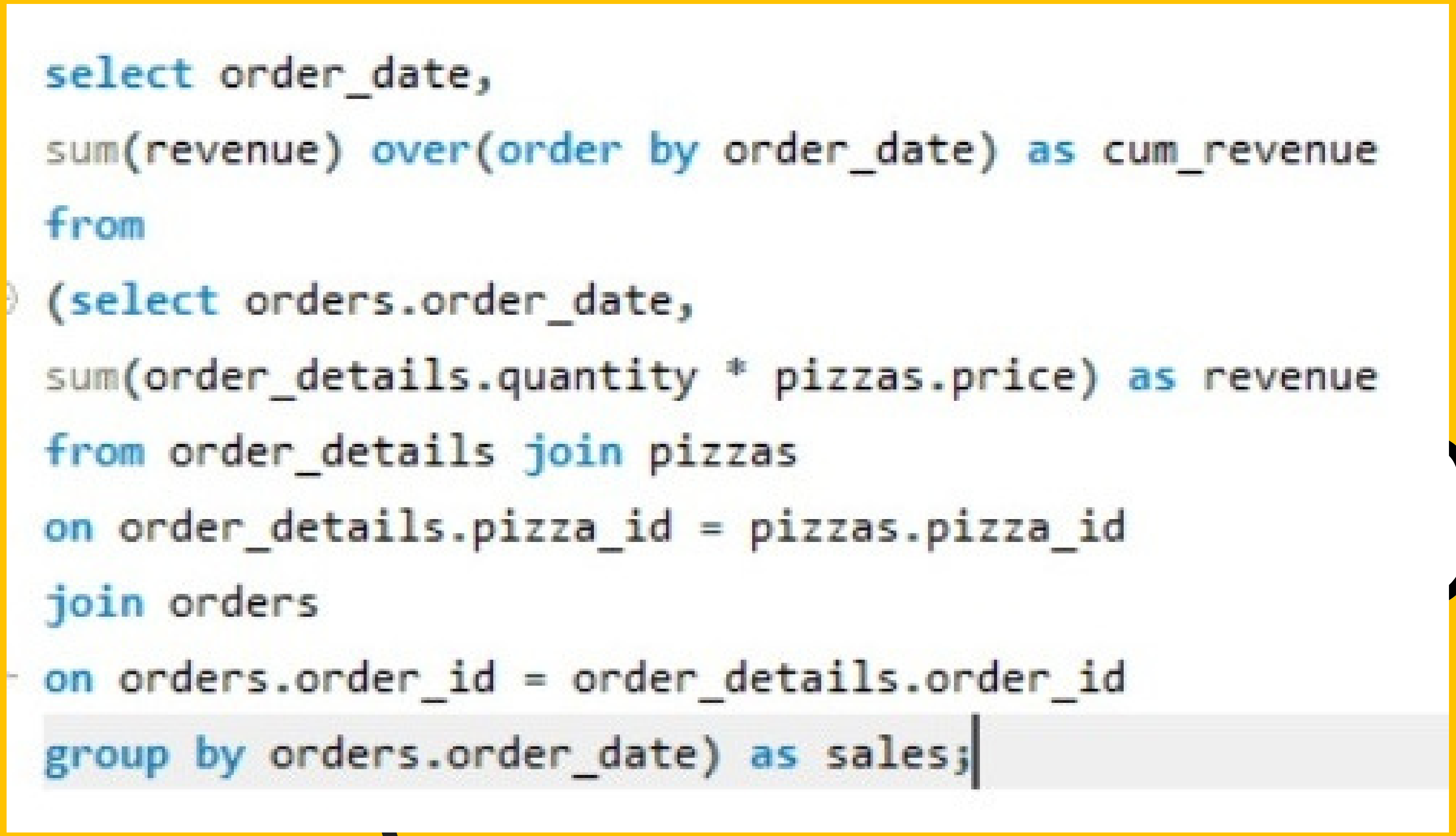
```
select pizza_types.category,
round(sum(order_details.quantity * pizzas.price) / ( select round(sum(order_details.quantity * pizzas.price),2) as total_sales
from order_details
join pizzas on pizzas.pizza_id = order_details.pizza_id) *100,2) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by revenue desc;
```



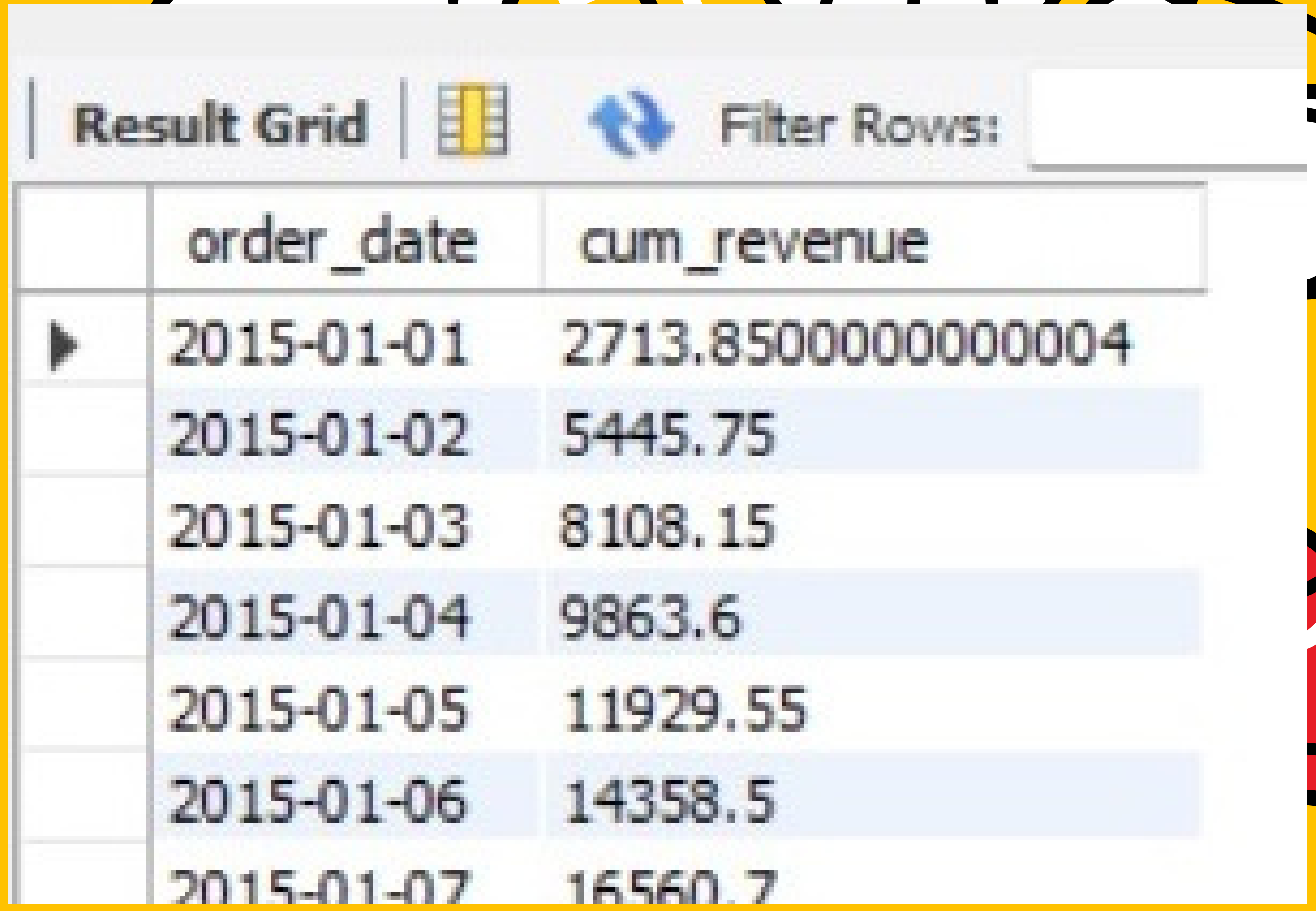


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

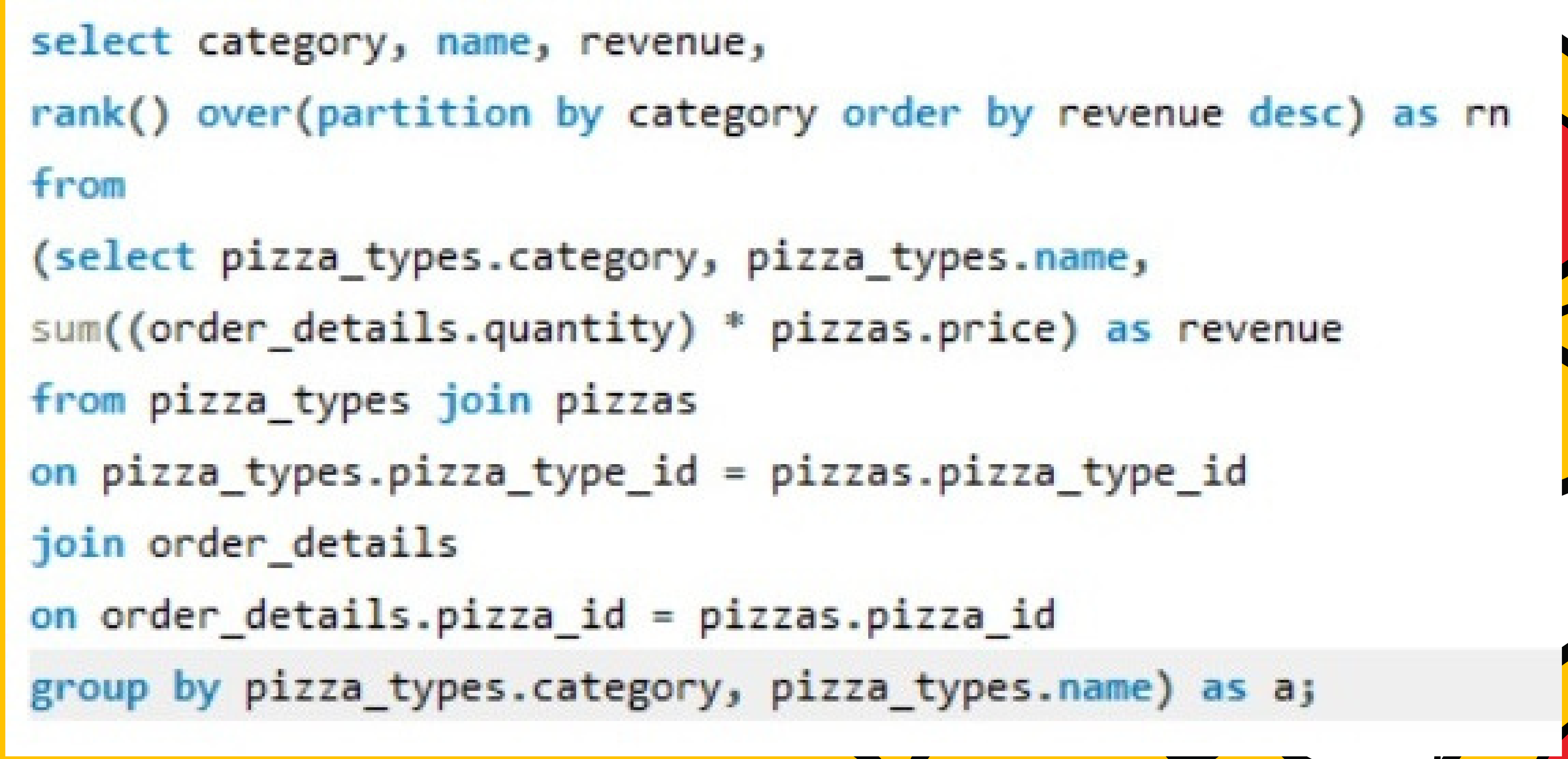


```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group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

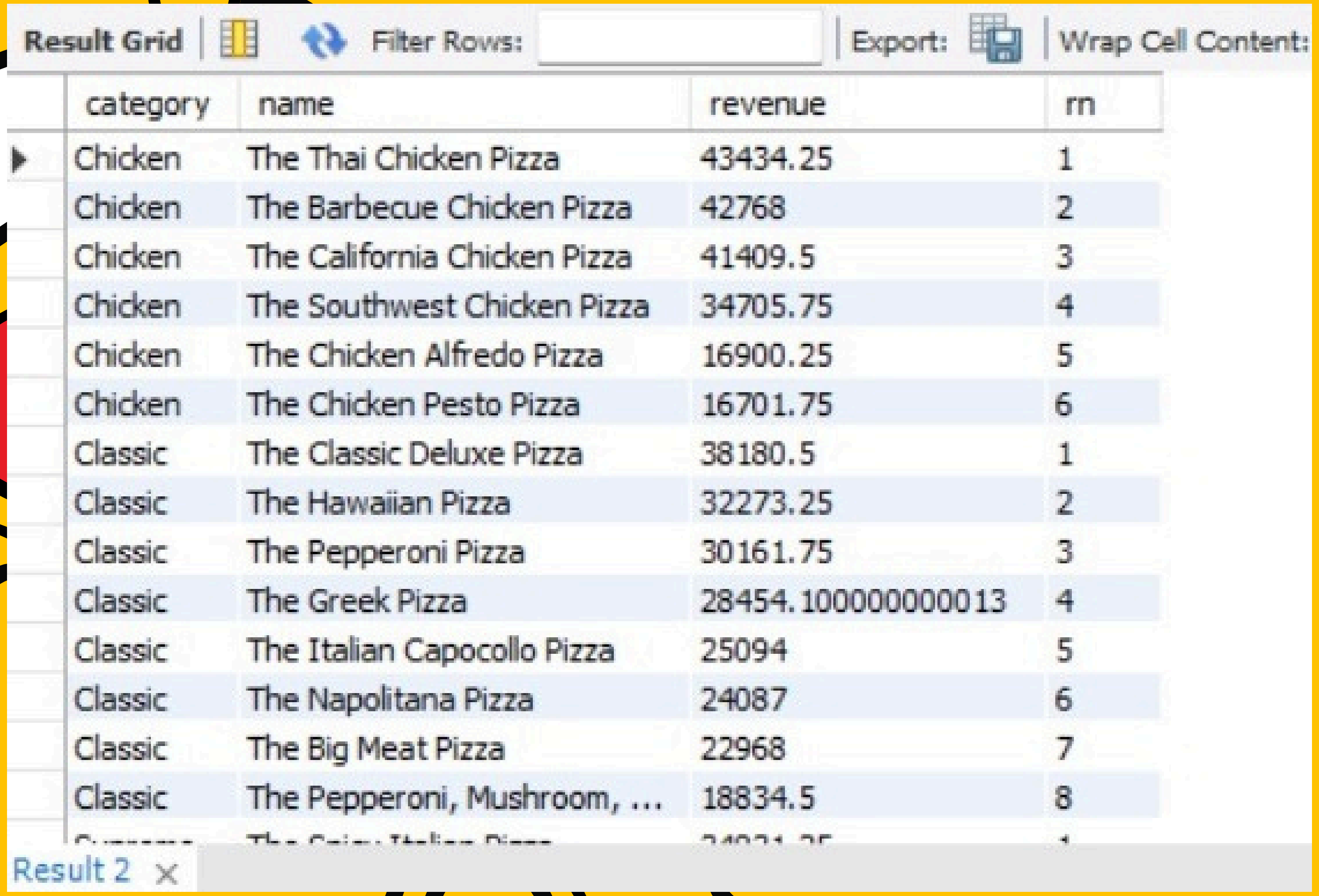


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7

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



```
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 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, pizza_types.name) as a;
```



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
Chicken	The Southwest Chicken Pizza	34705.75	4
Chicken	The Chicken Alfredo Pizza	16900.25	5
Chicken	The Chicken Pesto Pizza	16701.75	6
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Classic	The Greek Pizza	28454.100000000013	4
Classic	The Italian Capocollo Pizza	25094	5
Classic	The Napolitana Pizza	24087	6
Classic	The Big Meat Pizza	22968	7
Classic	The Pepperoni, Mushroom, ...	18834.5	8
Supreme	The Spicy Italian Pizza	34831.25	1

KEY INSIGHYS AND FINDINGS

Through an in-depth SQL-based analysis of Pizza Hut's sales data, several critical insights were uncovered:

✅ Order Volume & Revenue – A total of X orders were placed, generating $X in revenue.

✅ Top-Selling Pizzas – The top 5 most ordered pizzas were identified, with [Pizza Name] leading in sales.

✅ Pricing & Revenue Contribution – The highest-priced pizza was [Pizza Name], while the top 3 pizzas contributed X% of total revenue.

✅ Customer Preferences – The most common pizza size ordered was [Size], highlighting customer demand patterns.

✅ Sales Trends by Time – Peak ordering hours were between X AM – X PM, indicating high-demand periods.

✅ Category-Wise Distribution – The distribution of sales across pizza categories showed [Category] as the best-performing segment.

✅ Daily Order Patterns – On average, X pizzas were ordered per day, providing insights into daily demand fluctuations.

✅ Cumulative Revenue Growth – Revenue showed a steady increase over time, reflecting positive sales momentum.

These insights enable data-driven decision-making, helping optimize menu pricing, marketing strategies, and operational efficiency.

THANK YOU

I appreciate your time and interest in this project!

Through SQL-powered insights, we’ve uncovered key sales trends, customer preferences, and revenue patterns that can drive smarter business decisions. This analysis highlights the power of data in transforming operations and optimizing sales strategies.

💡 Data tells a story—let’s keep exploring it!

If you have any thoughts, questions, or ideas, I’d love to discuss them.

- Aman Kumar Sharma
- aamansharma027@gmail.com

Stay curious, stay data-driven

THANK YOU AND ENJOY A SLICE OF PIZZA!

Aman Kumar Sharma