Pizaa Sales Analysis Using SQL

By: Priya Berad

Project Overview

This project entails a comprehensive analysis of pizza sales data using SQL to uncover actionable business insights. It simulates a real-world scenario where a pizzeria wants to better understand customer behavior, optimize menu offerings, and maximize revenue.

By writing and executing structured queries, the project explores key metrics such as total orders, revenue, top-selling pizzas, time-based order patterns, and category-wise performance.

SQL Techniques Employed

- Joins – Combine data from multiple tables to get complete order details like pizza name, size, and price.
- Window Functions – Used to rank or analyze rows (e.g., top-selling pizzas) without collapsing the dataset.
- CTEs – Temporary result sets that simplify complex queries and improve readability.
- GROUP BY and HAVING - Grouped data by attributes (e.g., pizza type or size) to summarize metrics. HAVING filters groups based on conditions.

SQL Techniques Employed

- Aggregations - Used functions like SUM(), COUNT(), AVG() to calculate total revenue, number of orders, or average order value.
- Date Functions - Functions like EXTRACT(), DATE_TRUNC(), or TO_CHAR() used to analyze order trends by day, month, or hour.
- ORDER BY and LIMIT - Sorted results to rank pizzas or sales, and used LIMIT to show top performers (e.g., top 5 pizzas by revenue).
- Subqueries - Nested queries inside SELECT or WHERE clauses to fetch intermediate results for filtering or comparison.

Analytical Insights

The project uncovers several key business insights, including:

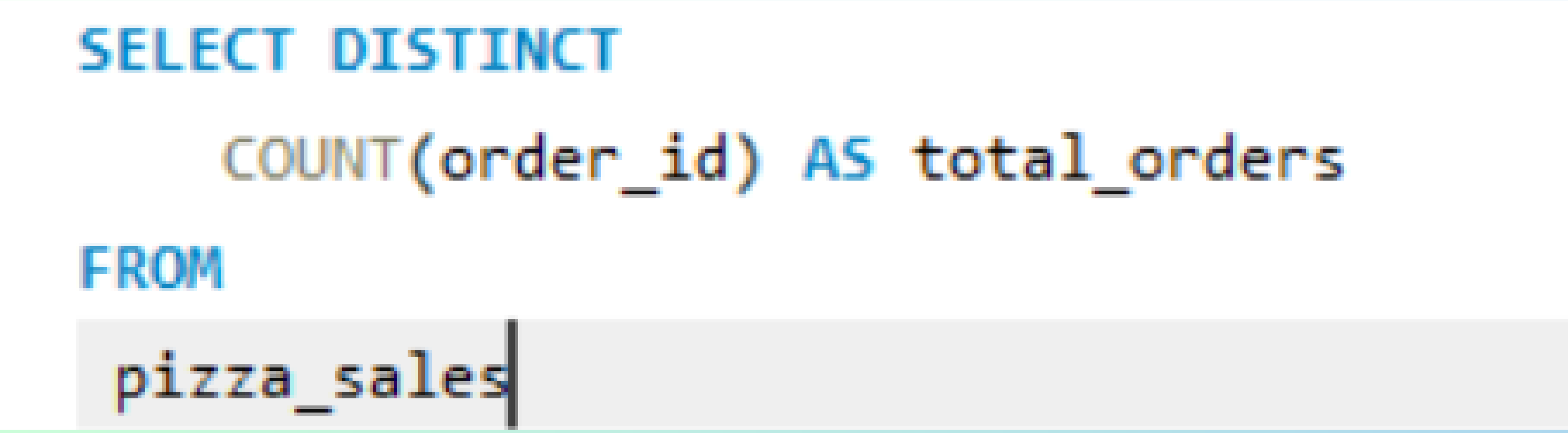
- Top-Selling Pizzas - Identify the most frequently ordered pizzas based on quantity sold.
- Revenue Analysis - Calculate total and average revenue by pizza type, size, and category.
- Order Trends - Analyze daily, weekly, and monthly order volumes to spot peak periods and slowdowns.
- Size Preferences - Determine which pizza sizes (S/M/L/XL) are most popular among customers.

Analytical Insights

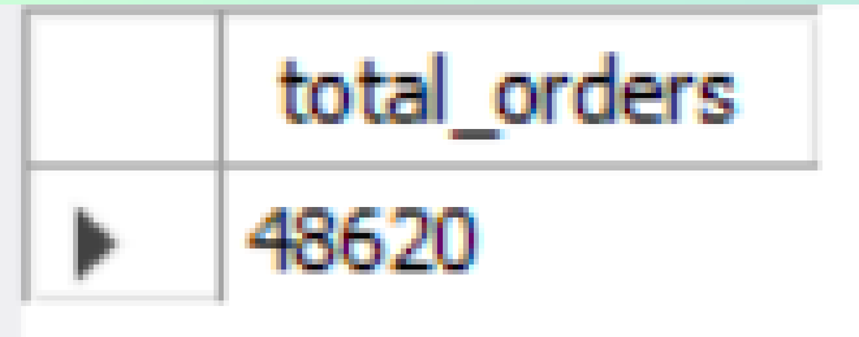
- Category Performance - Compare revenue and order count across categories like Classic, Supreme, and Veggie.
- High-Value Orders - Detect large or premium orders by ranking total order value.
- Order Frequency by Hour - Identify busy hours during the day to optimize staffing and prep.

Basic Query

Total number of orders placed



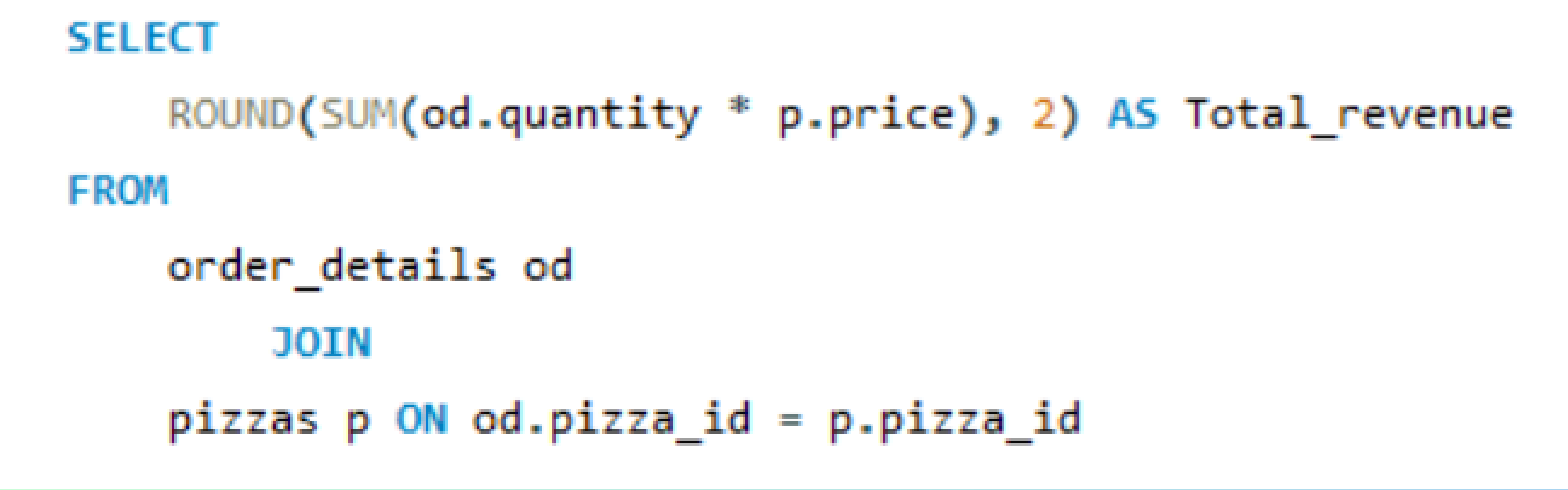
```
SELECT DISTINCT
    COUNT(order_id) AS total_orders
FROM
 pizza_sales
```



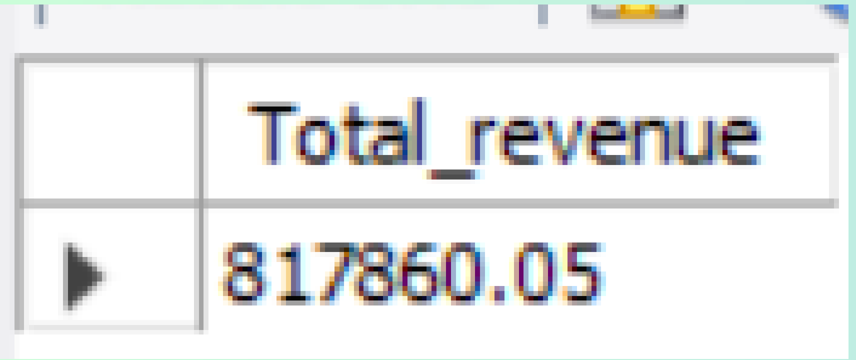
	total_orders
▶	48620

Basic Query

Total revenue generated



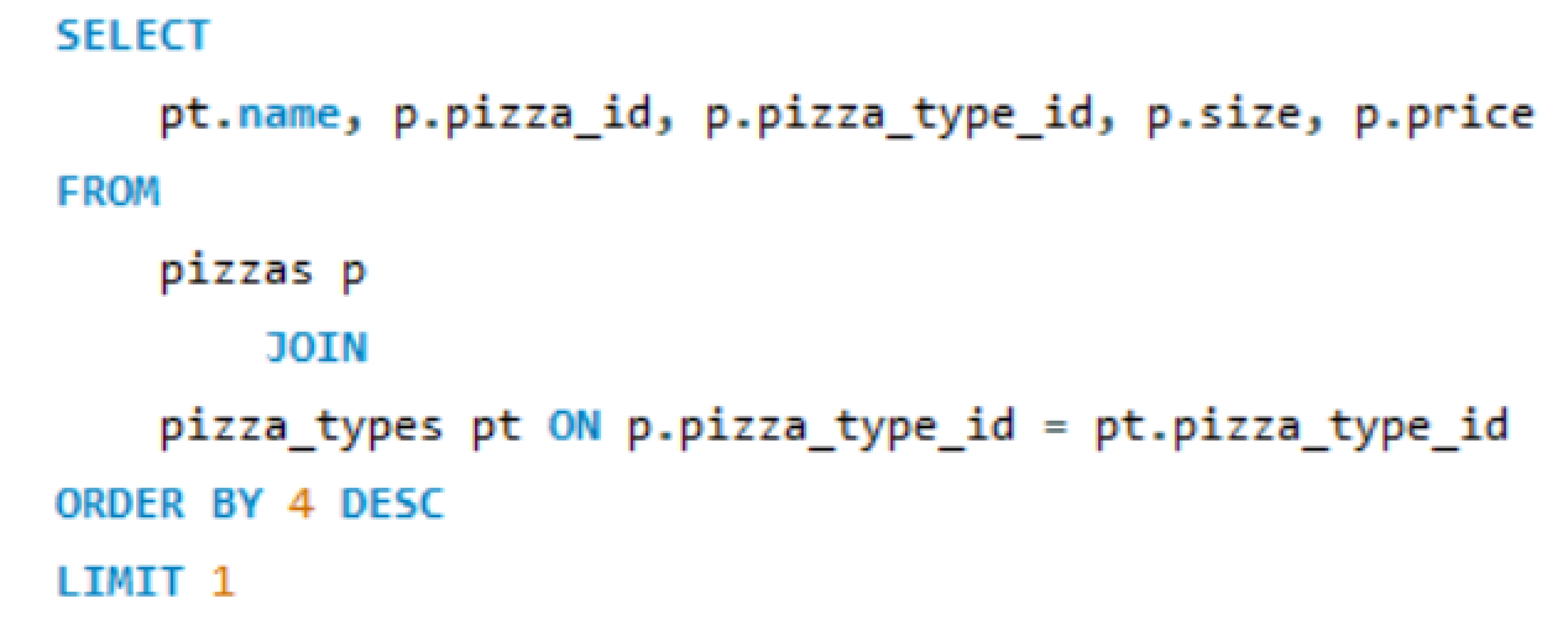
```
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS Total_revenue
FROM
    order_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
```



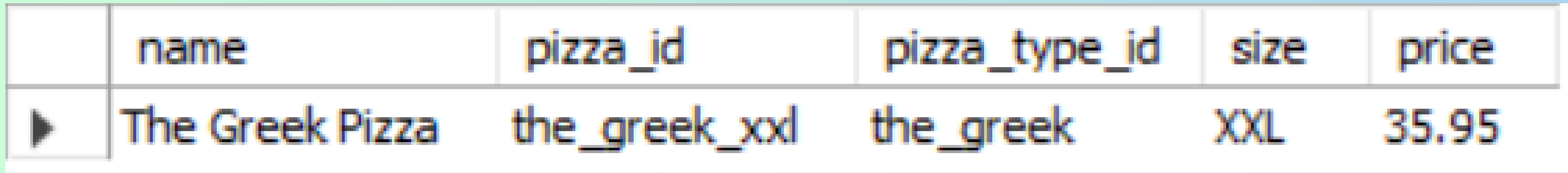
	Total_revenue
▶	817860.05

Basic Query

Highest priced pizza



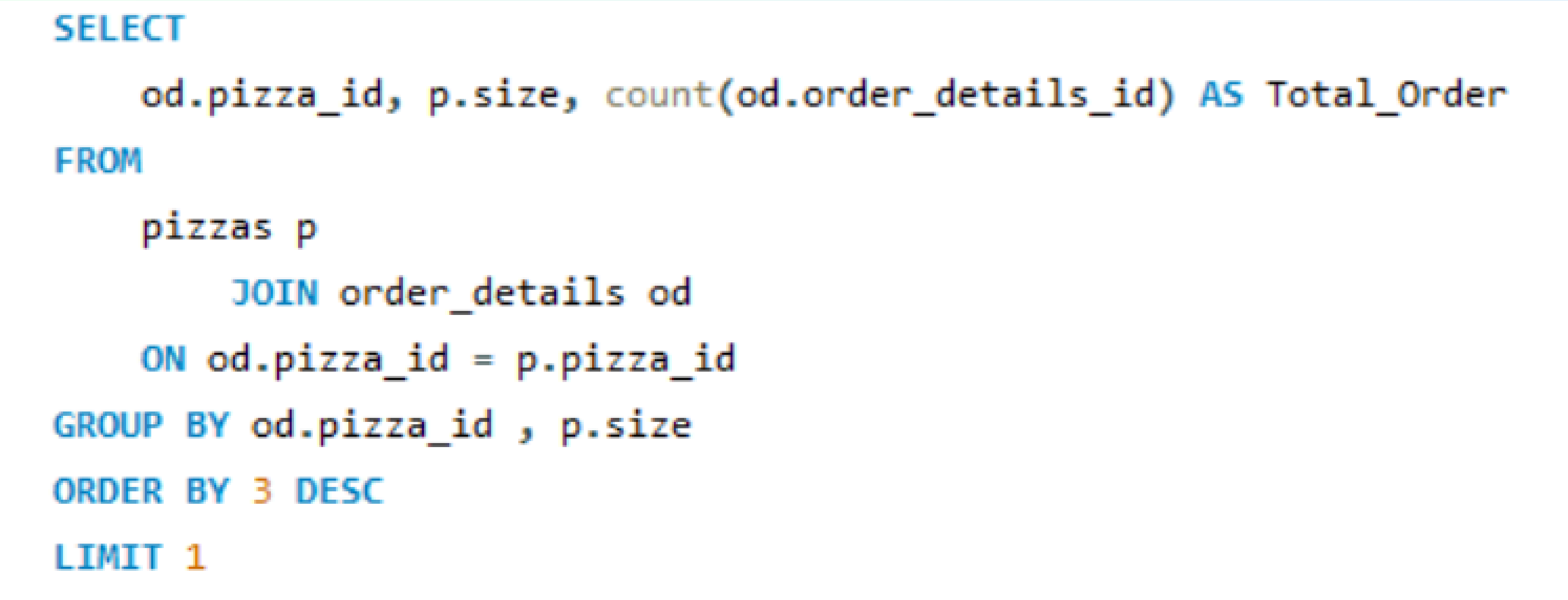
```
SELECT
    pt.name, p.pizza_id, p.pizza_type_id, p.size, p.price
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY 4 DESC
LIMIT 1
```



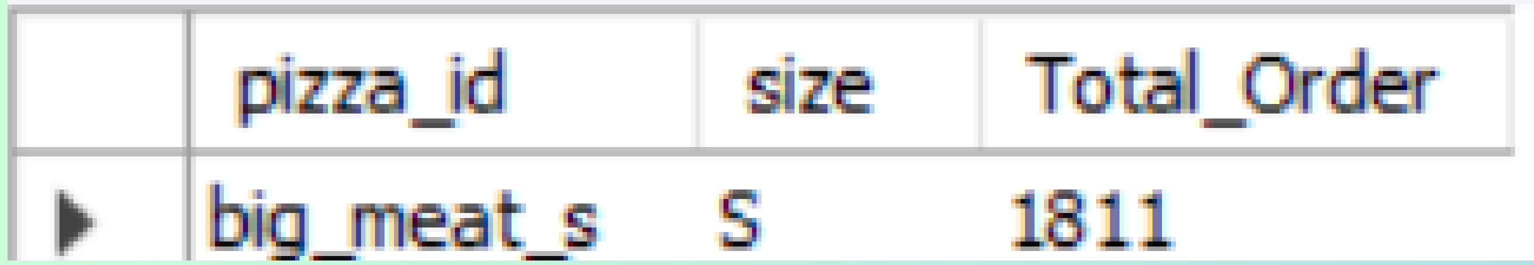
	name	pizza_id	pizza_type_id	size	price
▶	The Greek Pizza	the_greek_xxl	the_greek	XXL	35.95

Basic Query

Most common ordered pizza size



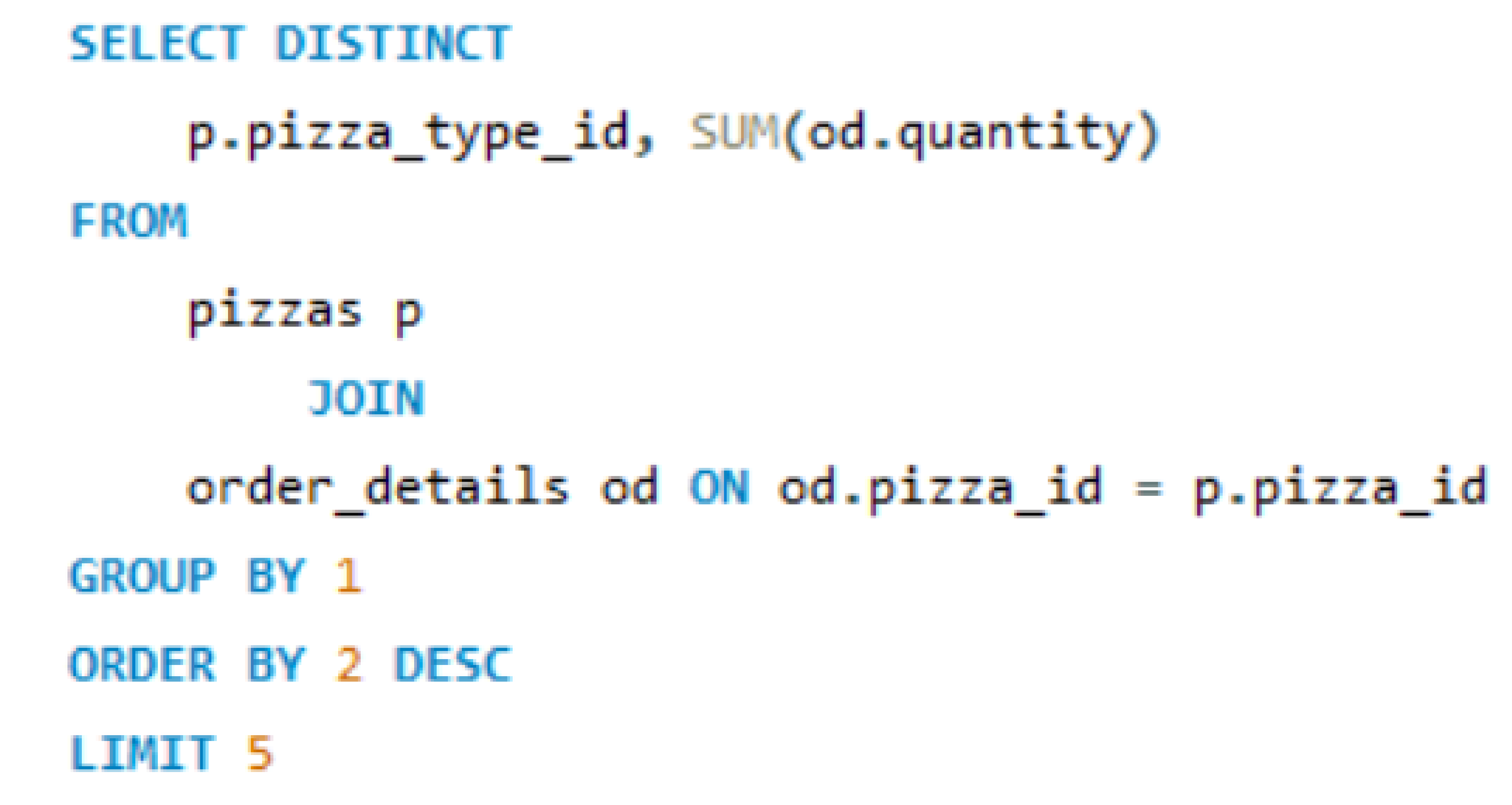
```
SELECT
    od.pizza_id, p.size, count(od.order_details_id) AS Total_Order
FROM
    pizzas p
        JOIN order_details od
    ON od.pizza_id = p.pizza_id
GROUP BY od.pizza_id , p.size
ORDER BY 3 DESC
LIMIT 1
```



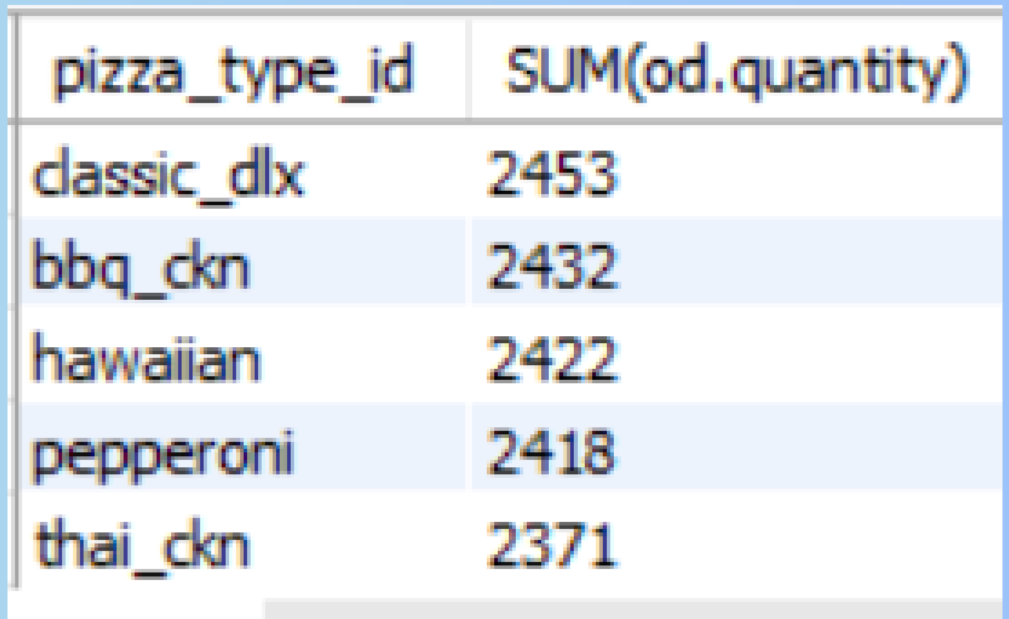
	pizza_id	size	Total_Order
▶	big_meat_s	S	1811

Basic Query

Top 5 most ordered pizza types



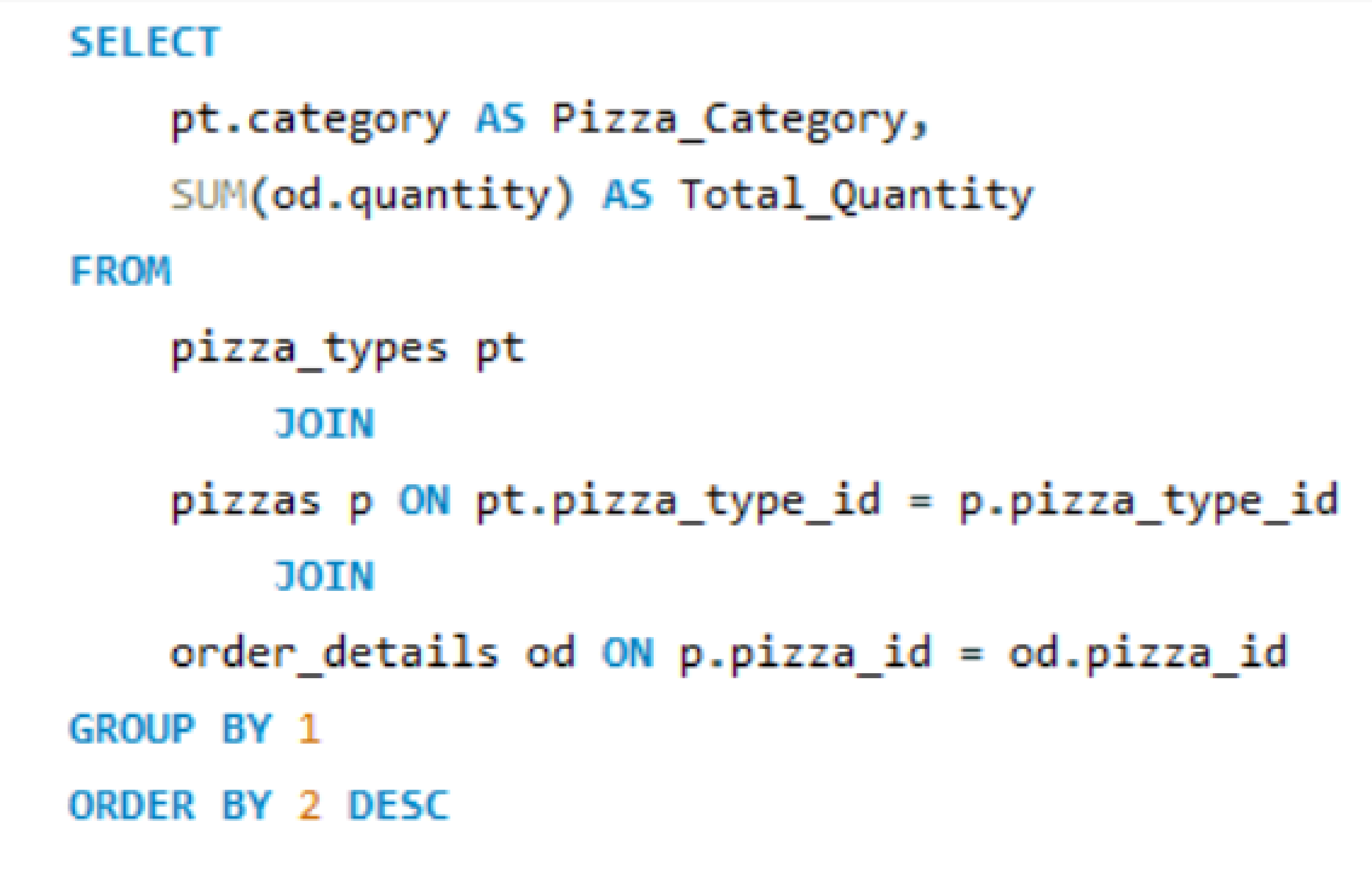
```
SELECT DISTINCT
    p.pizza_type_id, SUM(od.quantity)
FROM
    pizzas p
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5
```



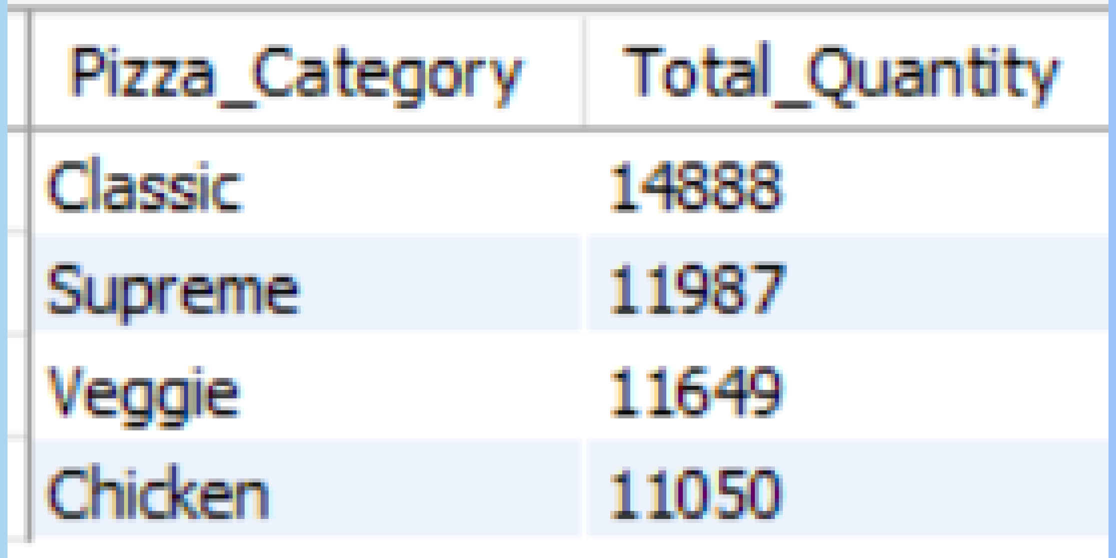
pizza_type_id	SUM(od.quantity)
classic_dlx	2453
bbq_ckn	2432
hawaiian	2422
pepperoni	2418
thai_ckn	2371

Intermediate Query

Total quantity of each pizza category ordered



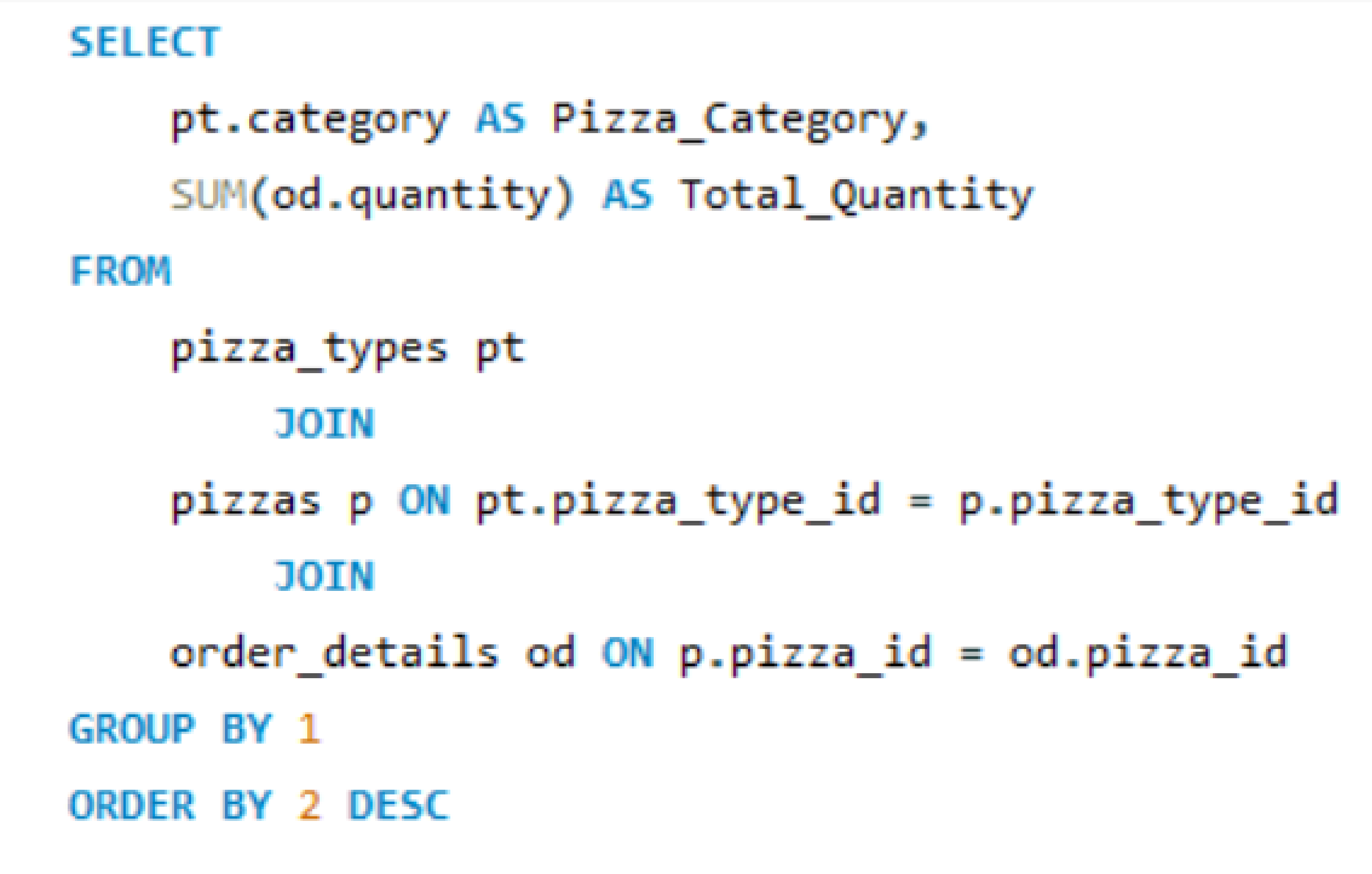
```sql
SELECT
    pt.category AS Pizza_Category,
    SUM(od.quantity) AS Total_Quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY 1
ORDER BY 2 DESC
```



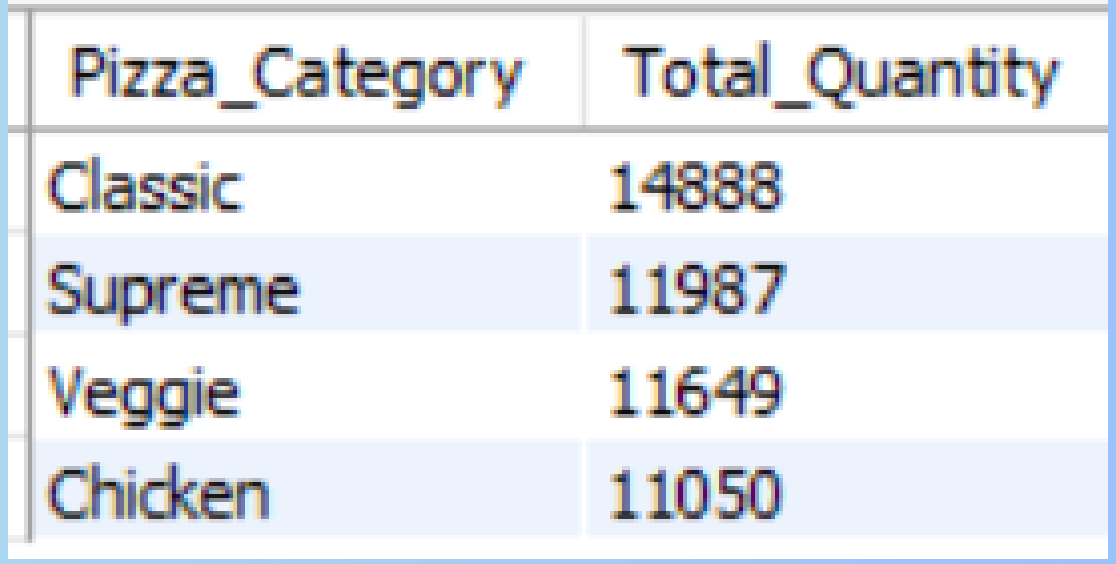
Pizza_Category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Intermediate Query

Total quantity of each pizza category ordered



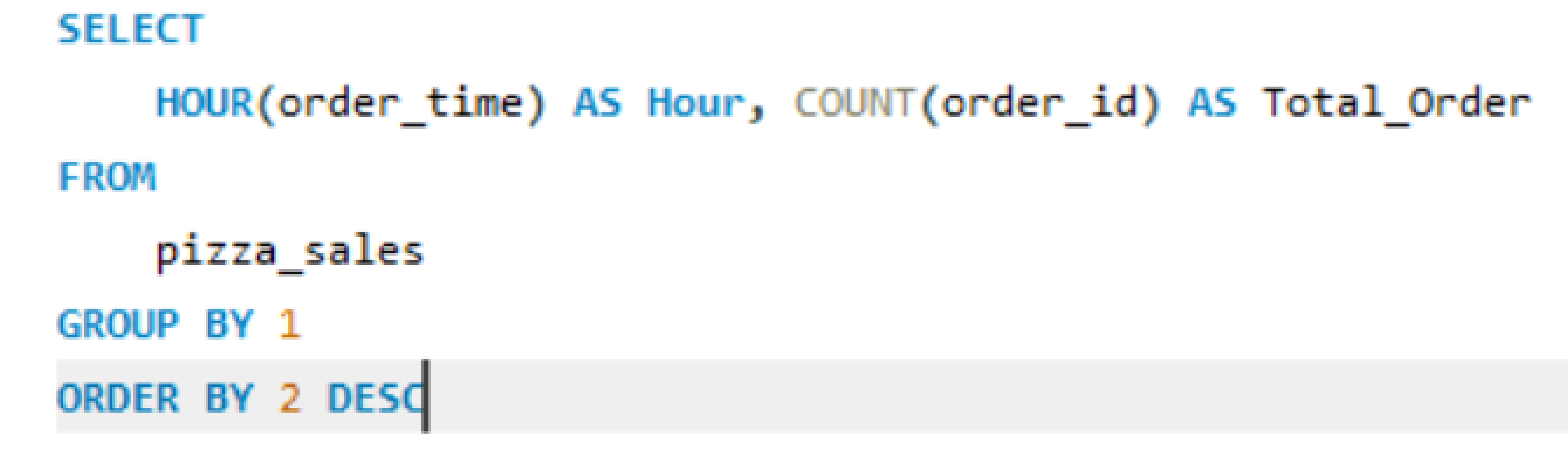
```
SELECT
    pt.category AS Pizza_Category,
    SUM(od.quantity) AS Total_Quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY 1
ORDER BY 2 DESC
```



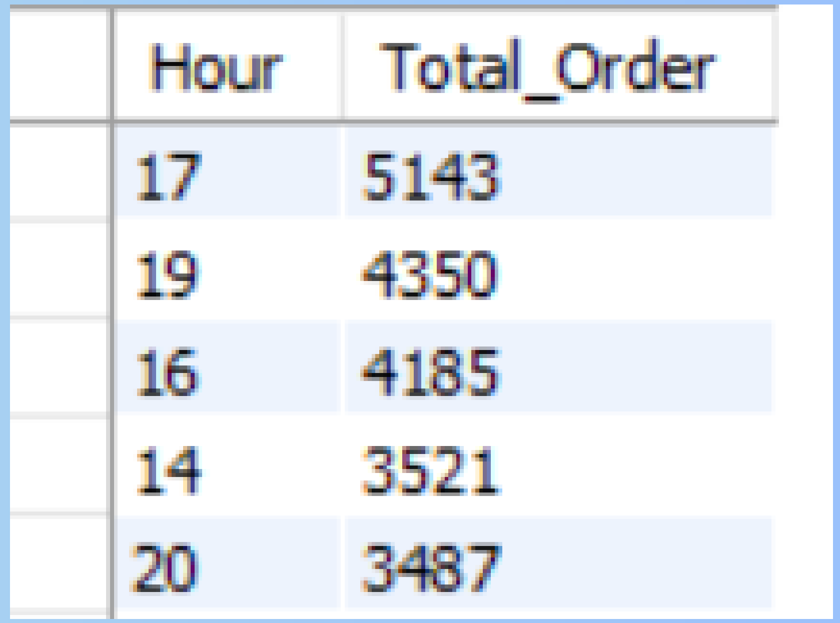
Pizza_Category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Intermediate Query

Distribution of orders by hour of the day



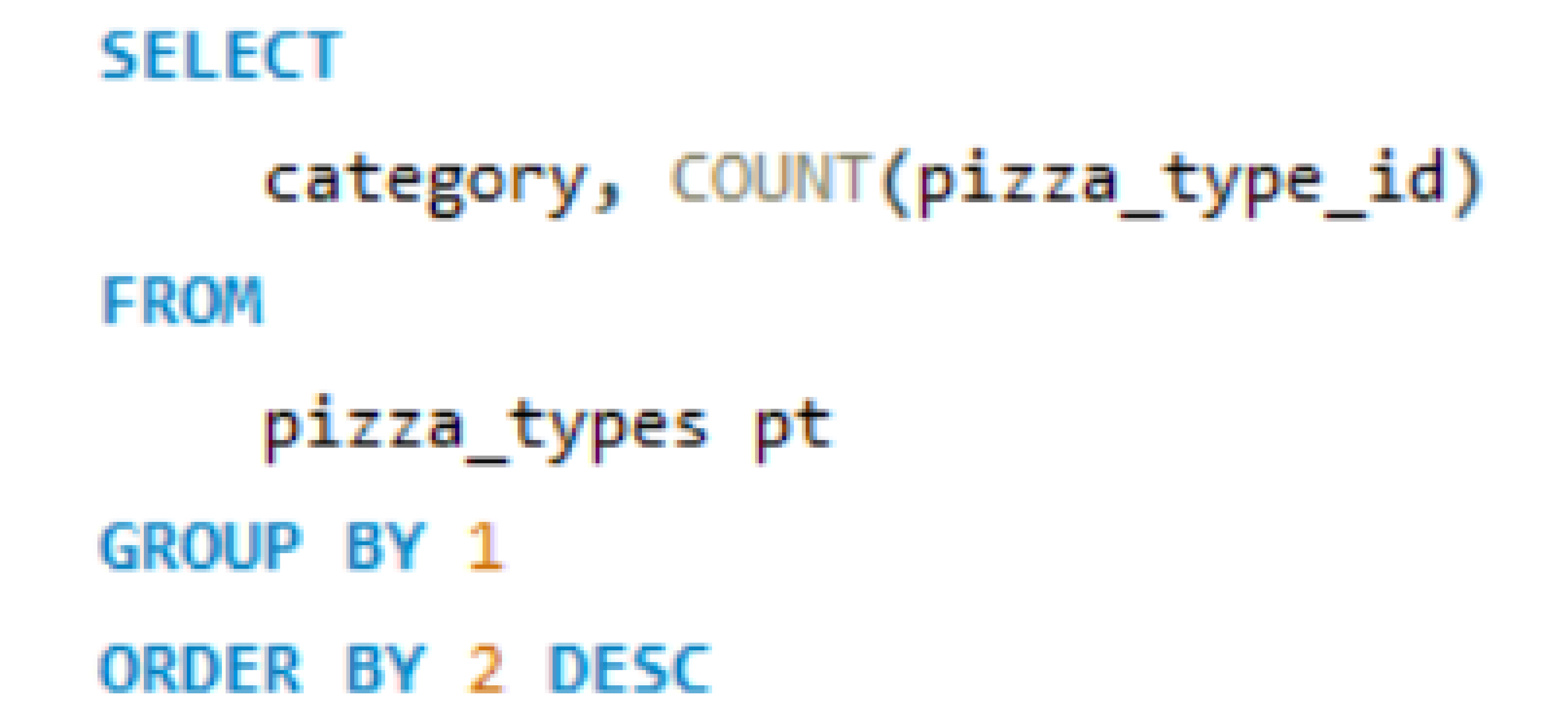
```
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS Total_Order
FROM
    pizza_sales
GROUP BY 1
ORDER BY 2 DESC
```



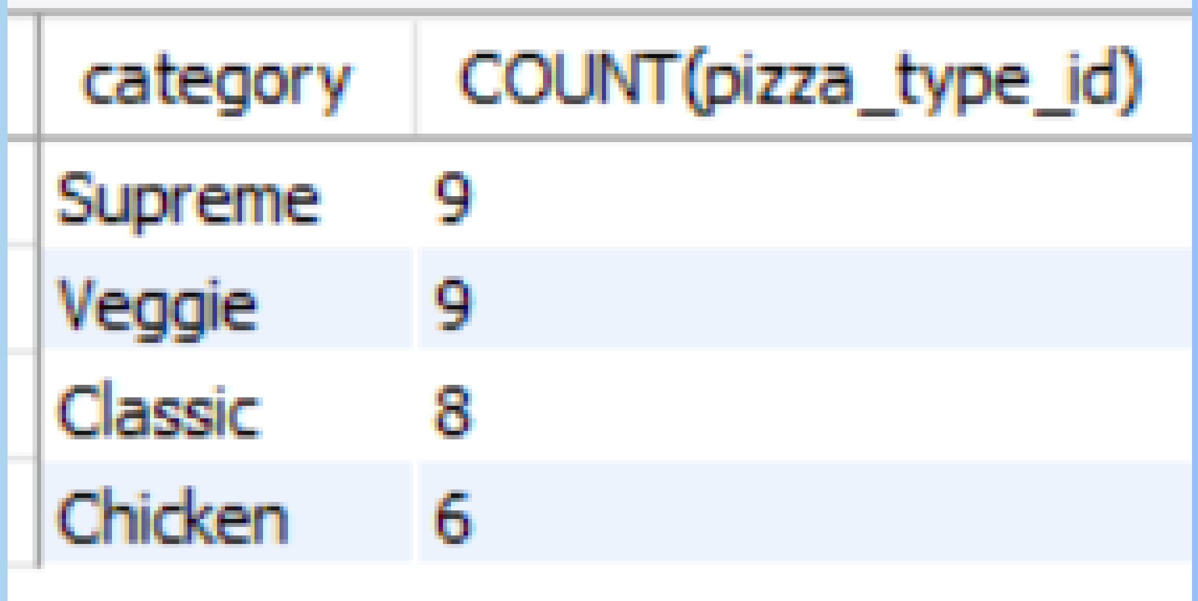
Hour	Total_Order
17	5143
19	4350
16	4185
14	3521
20	3487

Intermediate Query

Find the category-wise distribution of pizzas



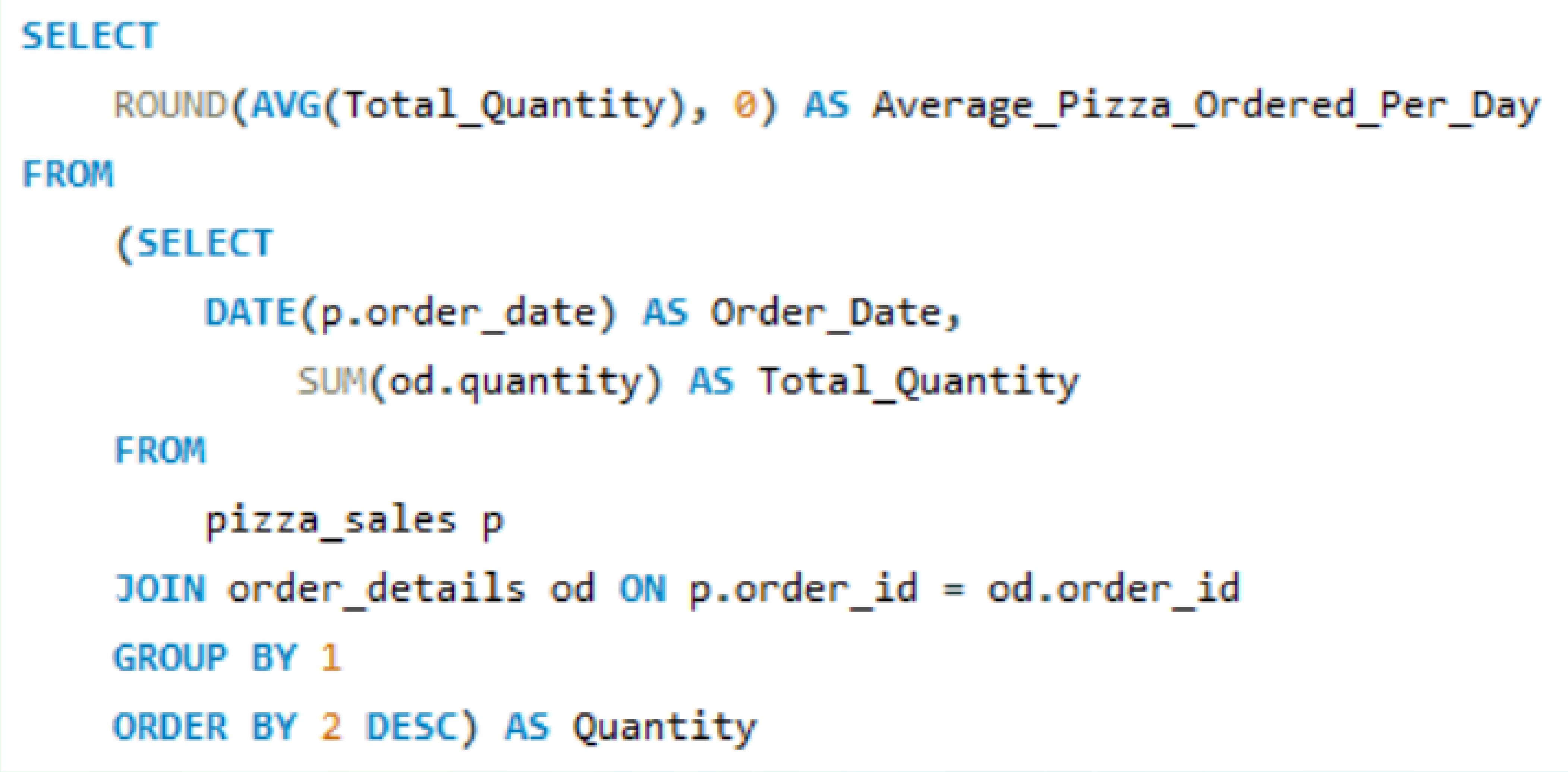
```
SELECT
    category, COUNT(pizza_type_id)
FROM
    pizza_types pt
GROUP BY 1
ORDER BY 2 DESC
```



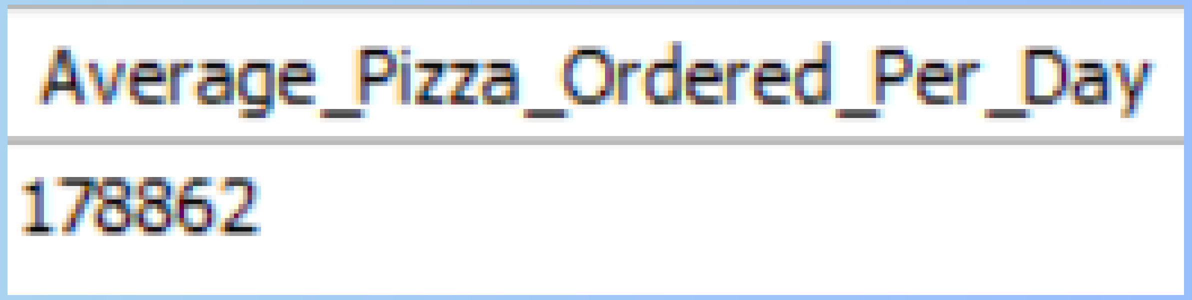
category	COUNT(pizza_type_id)
Supreme	9
Veggie	9
Classic	8
Chicken	6

Intermediate Query

Calculate the average number of pizzas ordered per day



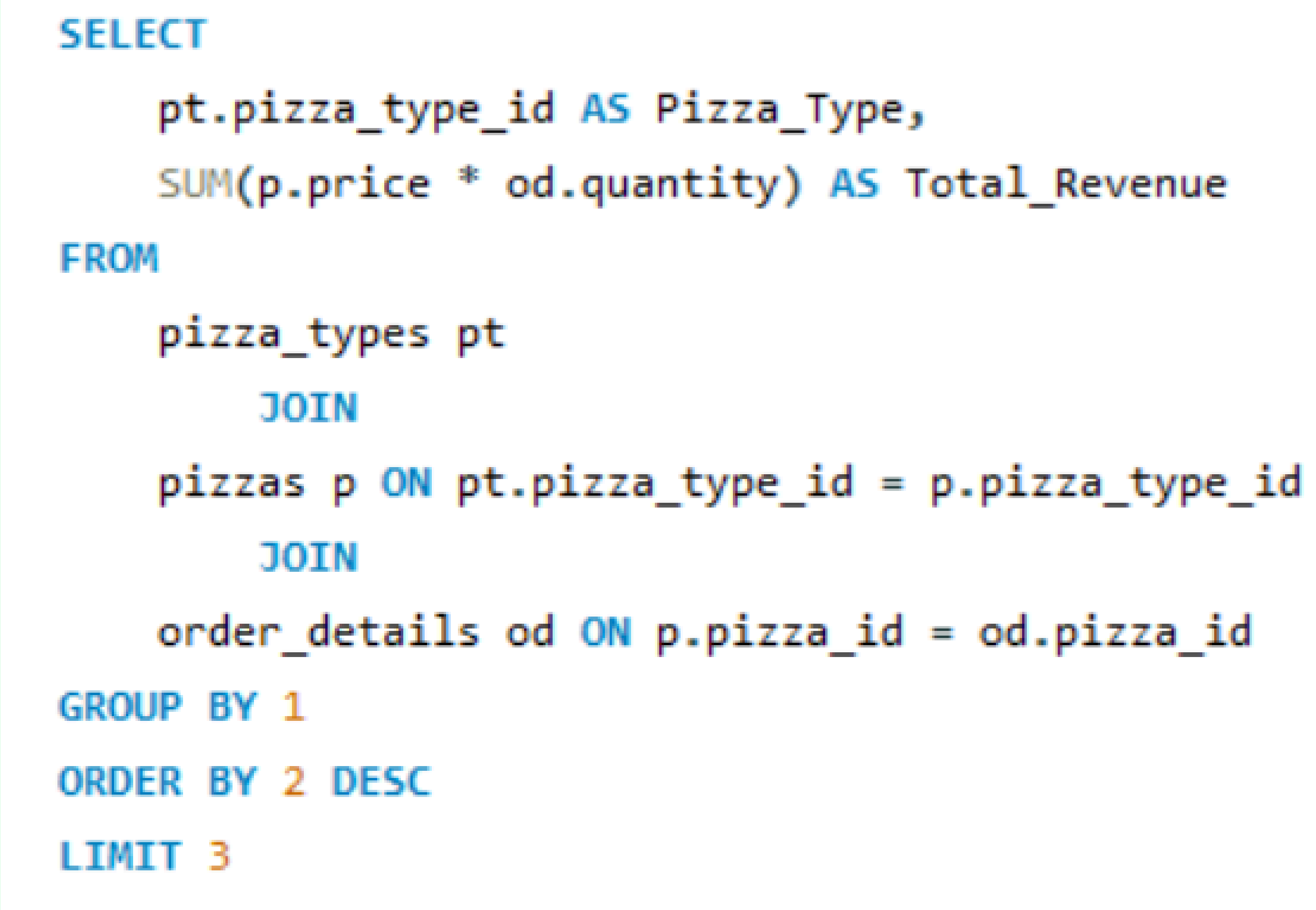
```
SELECT
    ROUND(AVG(Total_Quantity), 0) AS Average_Pizza_Ordered_Per_Day
FROM
    (SELECT
        DATE(p.order_date) AS Order_Date,
            SUM(od.quantity) AS Total_Quantity
    FROM
        pizza_sales p
    JOIN order_details od ON p.order_id = od.order_id
    GROUP BY 1
    ORDER BY 2 DESC) AS Quantity
```



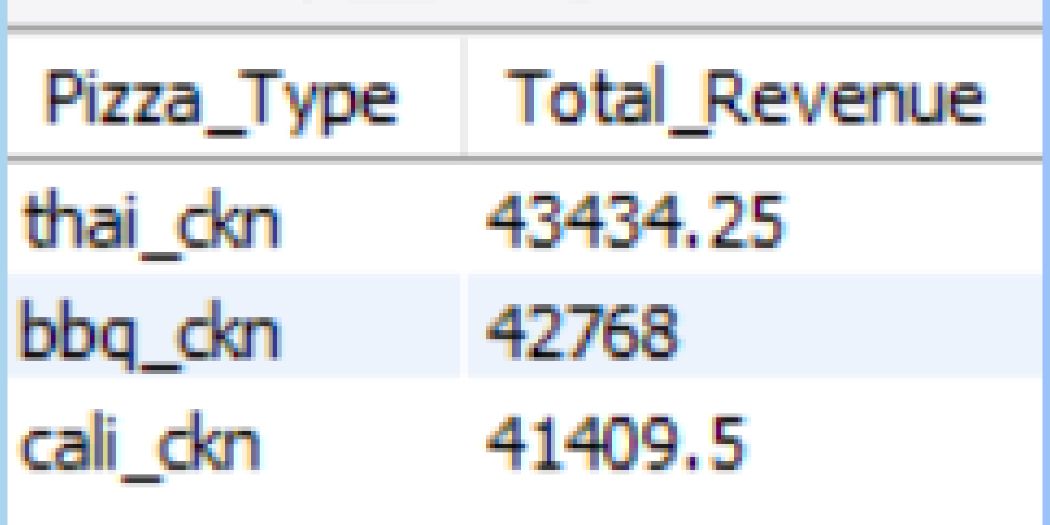
Average_Pizza_Ordered_Per_Day
178862

Intermediate Query

Top 3 most ordered pizza types based on revenue



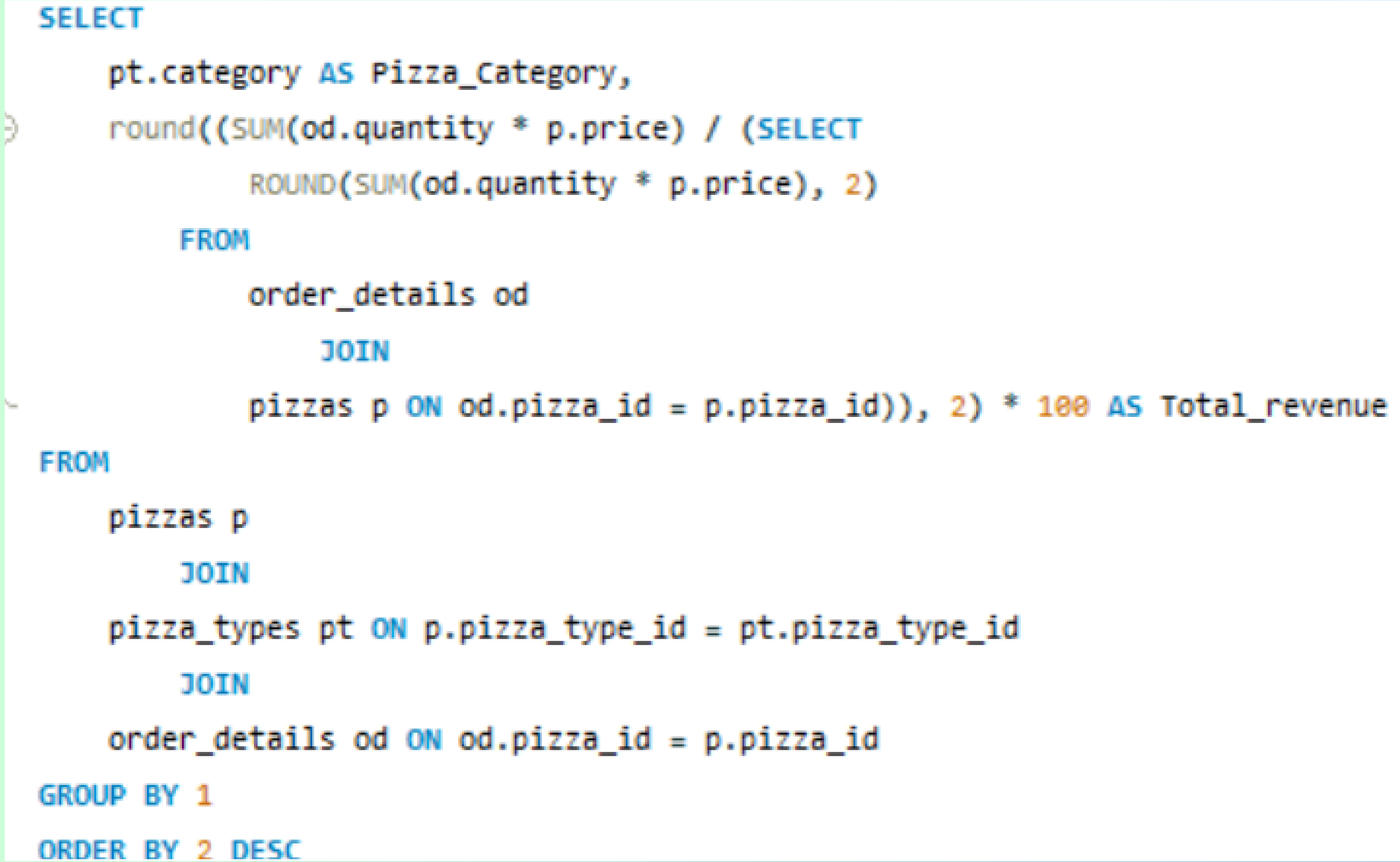
```
SELECT
    pt.pizza_type_id AS Pizza_Type,
    SUM(p.price * od.quantity) AS Total_Revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3
```



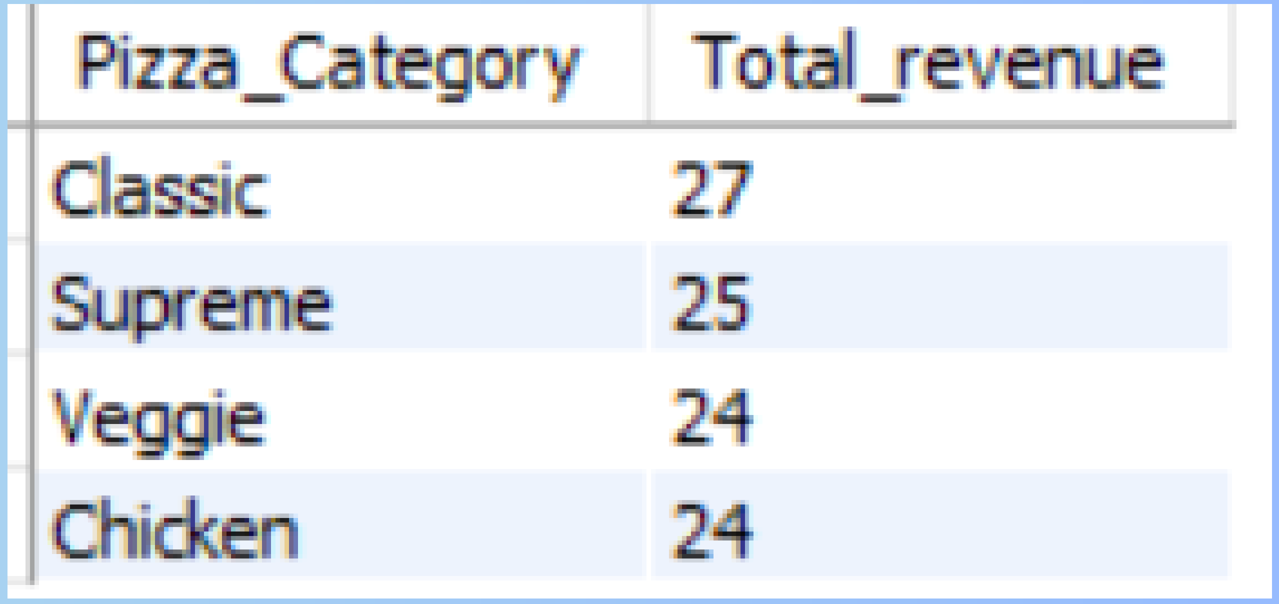
Pizza_Type	Total_Revenue
thai_ckn	43434.25
bbq_ckn	42768
cali_ckn	41409.5

Advanced Query

Percentage contribution of each pizza type to total revenue



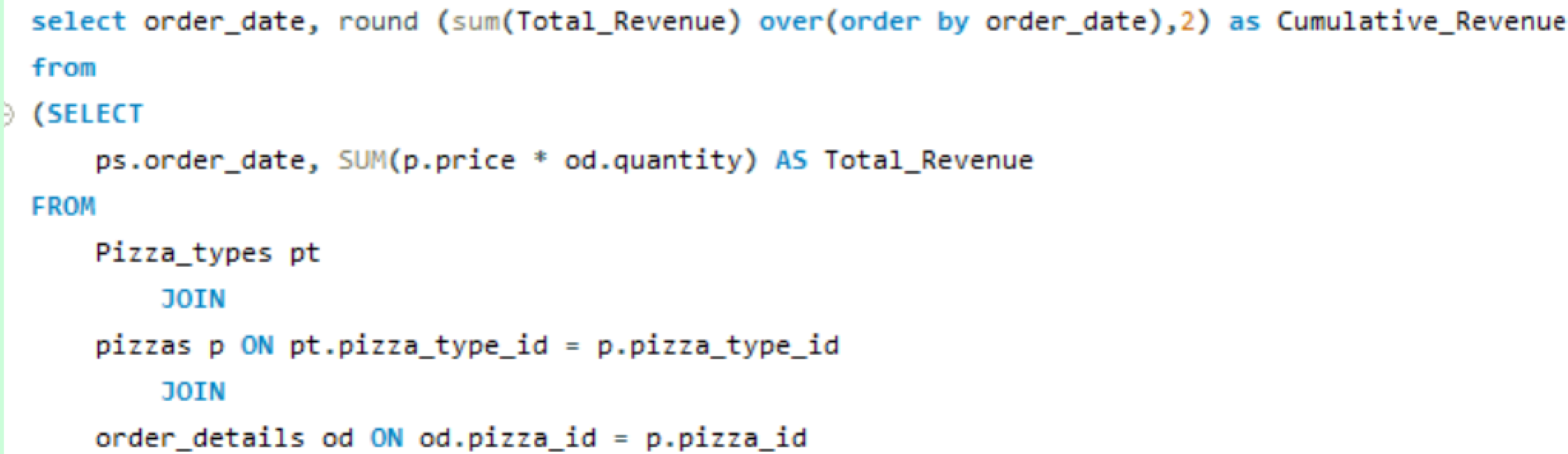
```
SELECT
    pt.category AS Pizza_Category,
    round((SUM(od.quantity * p.price) / (SELECT
            ROUND(SUM(od.quantity * p.price), 2)
        FROM
            order_details od
                JOIN
            pizzas p ON od.pizza_id = p.pizza_id)), 2) * 100 AS Total_revenue
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY 1
ORDER BY 2 DESC
```



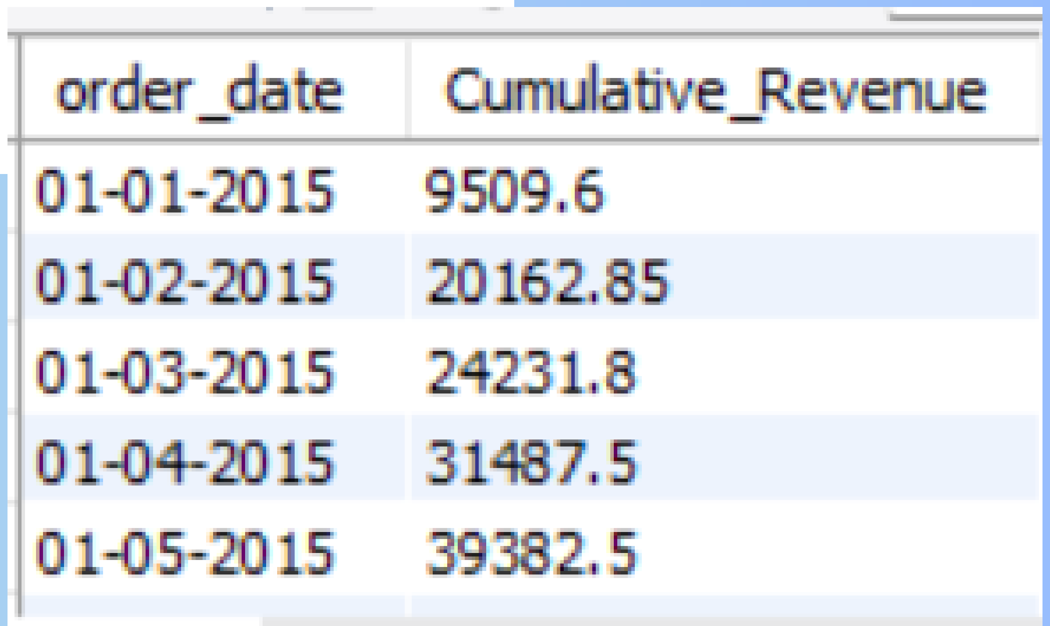
Pizza_Category	Total_revenue
Classic	27
Supreme	25
Veggie	24
Chicken	24

Advanced Query

Analyze the cumulative revenue generated over time



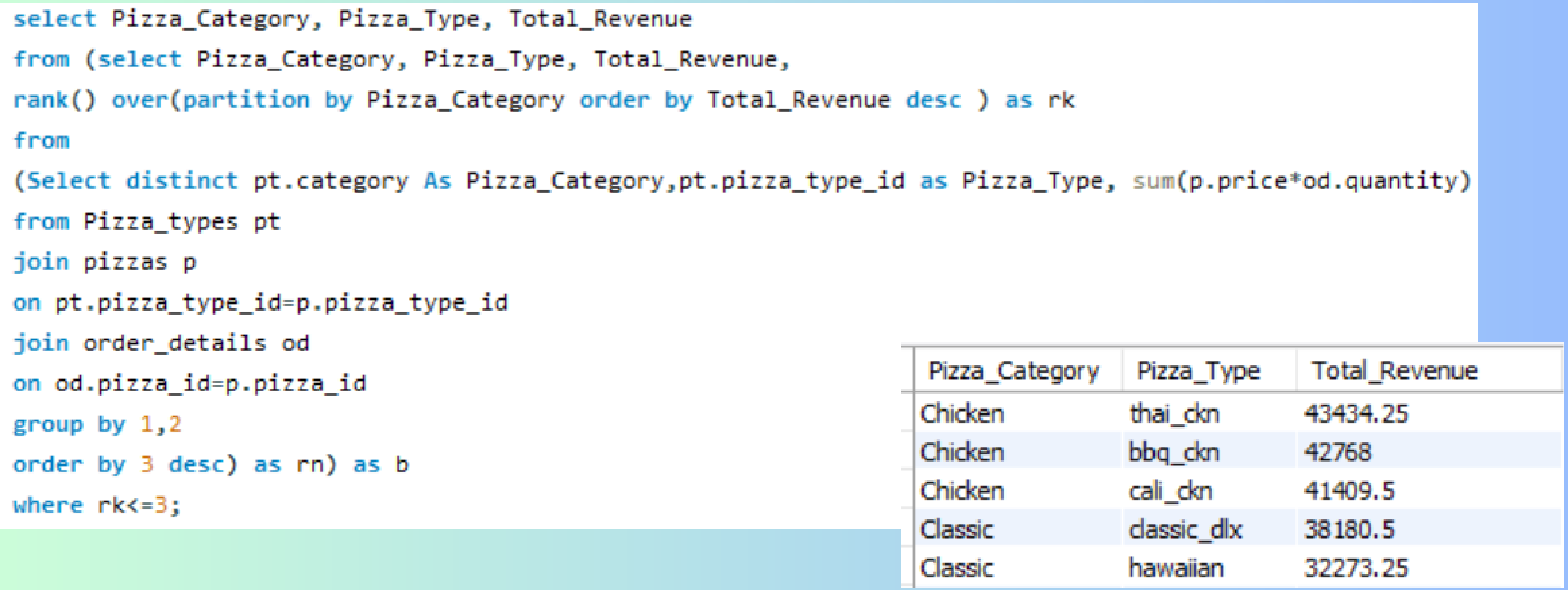
```
select order_date, round (sum(Total_Revenue) over(order by order_date),2) as Cumulative_Revenue
from
(SELECT
    ps.order_date, SUM(p.price * od.quantity) AS Total_Revenue
FROM
    Pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
        JOIN
    pizza_sales ps ON od.order_id = ps.order_id
```



order_date	Cumulative_Revenue
01-01-2015	9509.6
01-02-2015	20162.85
01-03-2015	24231.8
01-04-2015	31487.5
01-05-2015	39382.5

Advanced Query

Top 3 most ordered pizza types based on revenue for each pizza category



```
select Pizza_Category, Pizza_Type, Total_Revenue
from (select Pizza_Category, Pizza_Type, Total_Revenue,
rank() over(partition by Pizza_Category order by Total_Revenue desc ) as rk
from
(Select distinct pt.category As Pizza_Category,pt.pizza_type_id as Pizza_Type, sum(p.price*od.quantity)
from Pizza_types pt
join pizzas p
on pt.pizza_type_id=p.pizza_type_id
join order_details od
on od.pizza_id=p.pizza_id
group by 1,2
order by 3 desc) as rn) as b
where rk<=3;
```

Pizza_Category	Pizza_Type	Total_Revenue
Chicken	thai_ckn	43434.25
Chicken	bbq_ckn	42768
Chicken	cali_ckn	41409.5
Classic	classic_dlx	38180.5
Classic	hawaiian	32273.25

Thank You!