SQL PROJECT ON PIZZA SALES

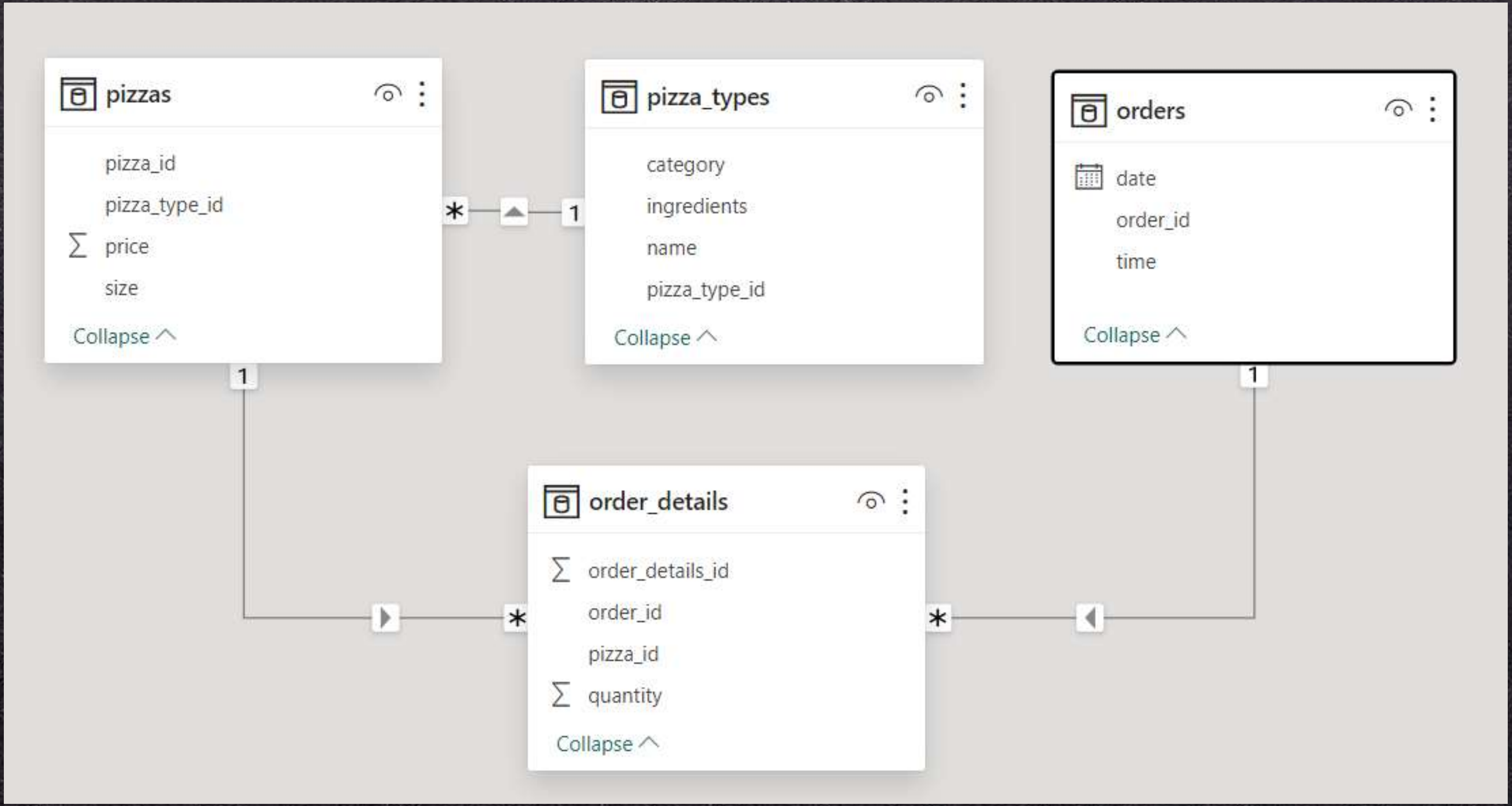
VIVEK KUMAR

Project Overview

This project aims to analyze pizza sales data using SQL to uncover valuable insights and improve business operations. This analysis will help in making data-driven decisions to enhance marketing strategies, optimize inventory, and improve customer satisfaction.

VIVEK KUMAR

DATABASE SCHEMA



VIVEK KUMAR

Basic Questions

1. Retrieve the total number of orders placed.

2. Calculate the total revenue generated from pizza sales.

3. Identify the highest-priced pizza.

4. Identify the most common pizza size ordered.

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

Intermediate Questions

1. Join the necessary tables to find the total quantity of each pizza category ordered.

2. Join relevant tables to find the category-wise distribution of pizzas (category-wise orders).

3. Group the orders by date and calculate the average number of pizzas ordered per day.

4. Determine the top 3 most ordered pizza types based on revenue.

VIVEK KUMAR

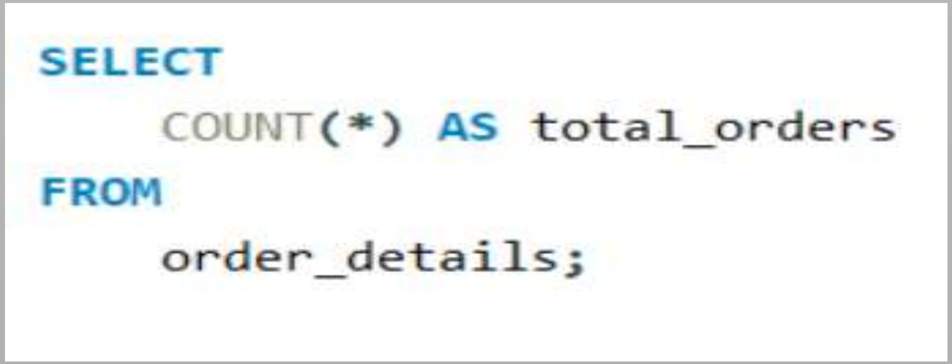
Advanced Questions

1. Calculate the percentage contribution of each pizza category type to total revenue.

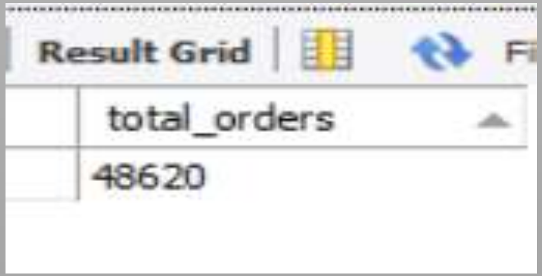
2. Analyse the cumulative revenue generated over time.

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

Retrieve the total number of orders placed.

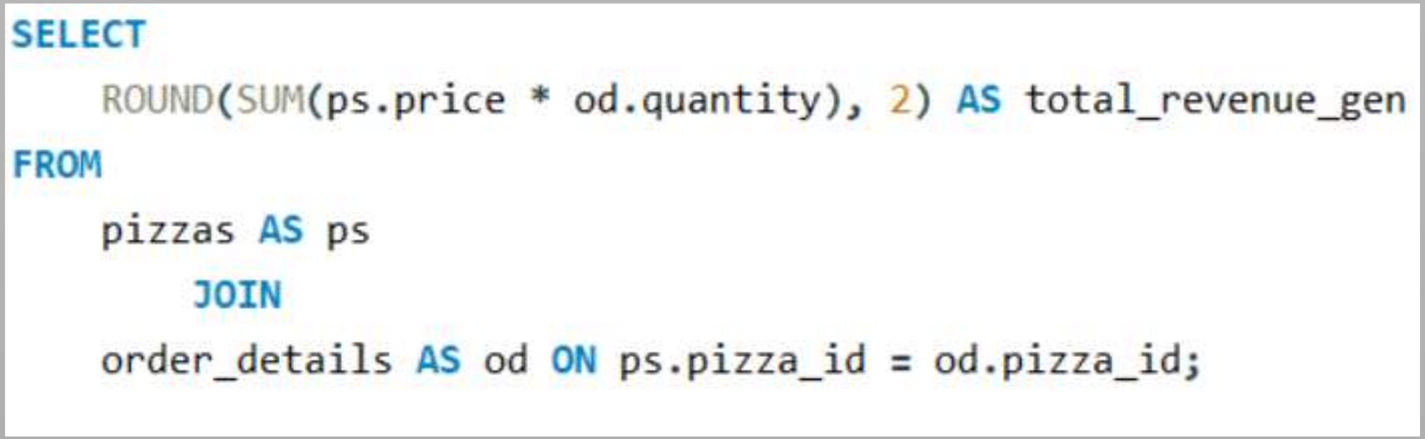


```
SELECT
    COUNT(*) AS total_orders
FROM
    order_details;
```

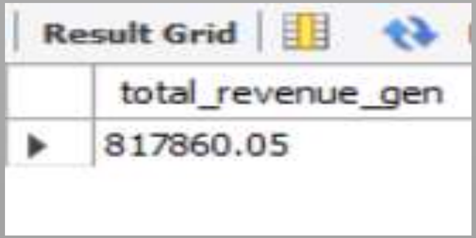


```
total_orders
48620
```

Calculate the total revenue generated from pizza sales.



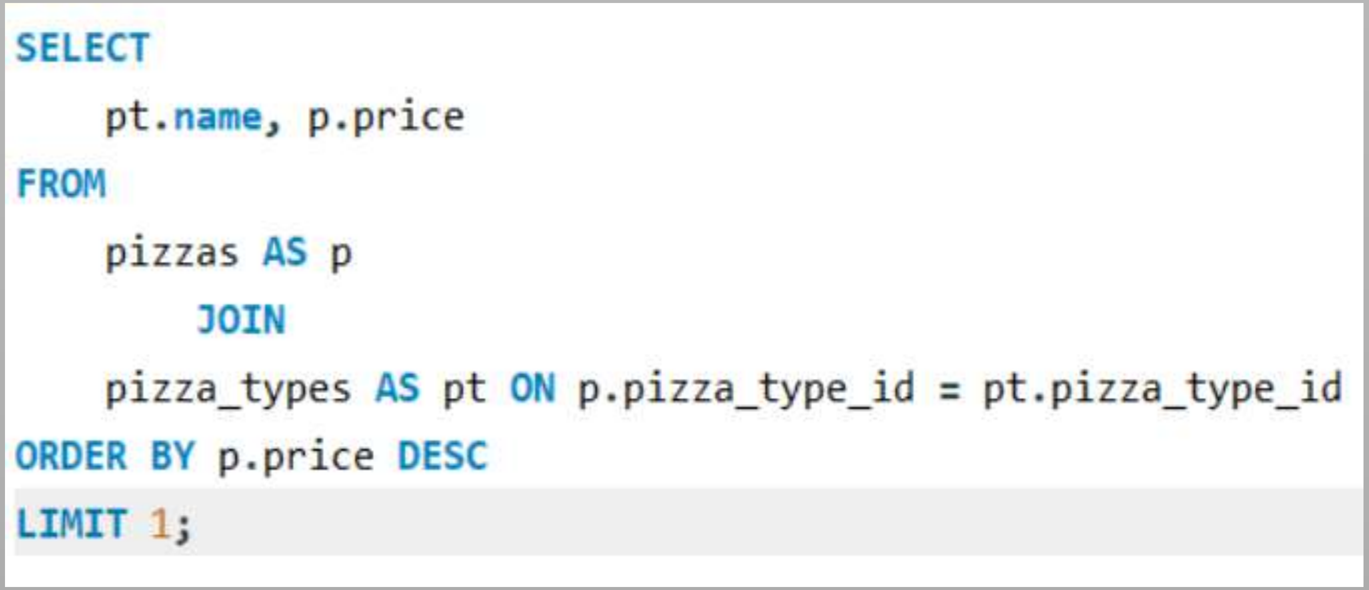
```
SELECT
    ROUND(SUM(ps.price * od.quantity), 2) AS total_revenue_gen
FROM
    pizzas AS ps
        JOIN
    order_details AS od ON ps.pizza_id = od.pizza_id;
```



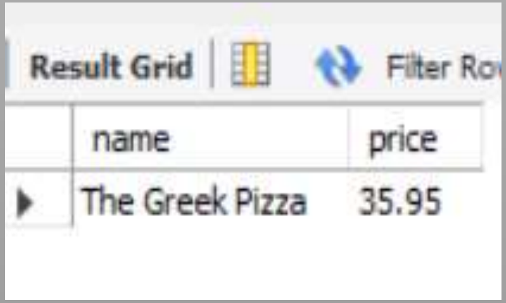
```
total_revenue_gen
817860.05
```

VIVEK KUMAR

Identify the highest-priced pizza.

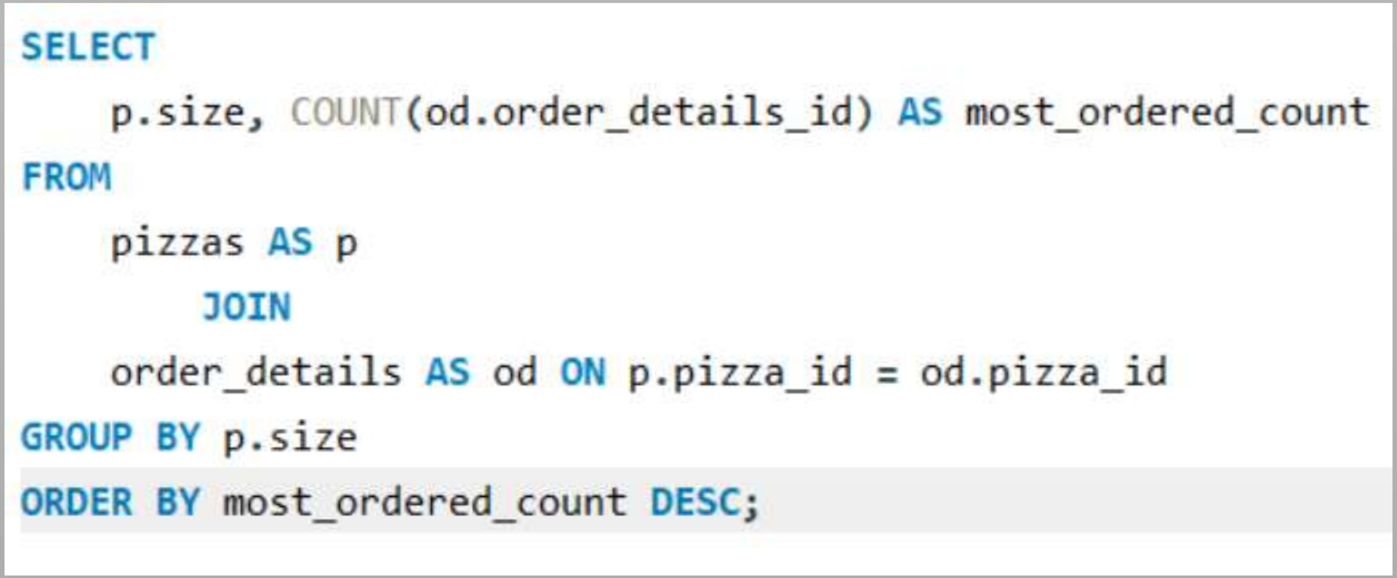


```sql
SELECT
    pt.name, p.price
FROM
    pizzas AS p
        JOIN
    pizza_types AS pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```

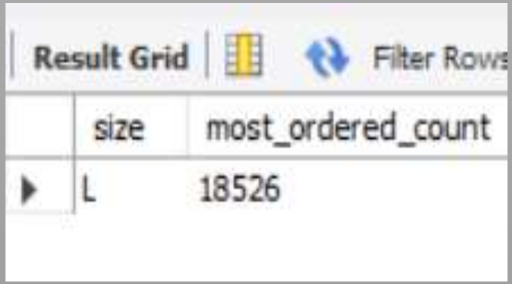


```
name             price
The Greek Pizza  35.95
```

Identify the most common pizza size ordered.



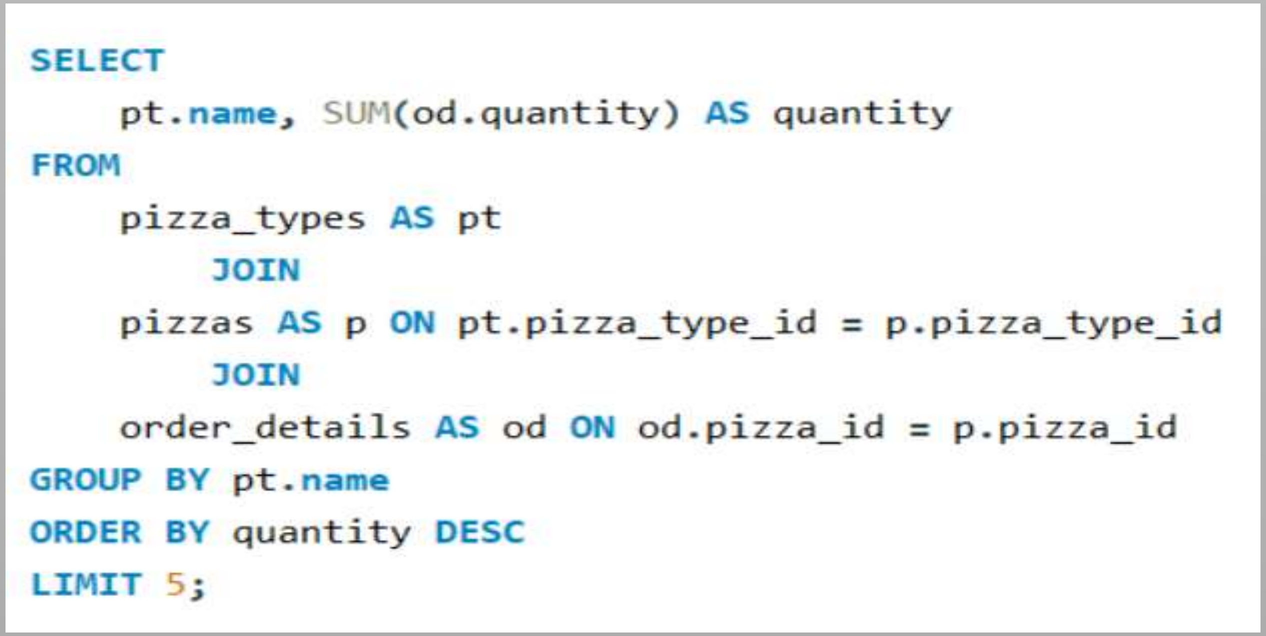
```
SELECT
    p.size, COUNT(od.order_details_id) AS most_ordered_count
FROM
    pizzas AS p
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY most_ordered_count DESC;
```



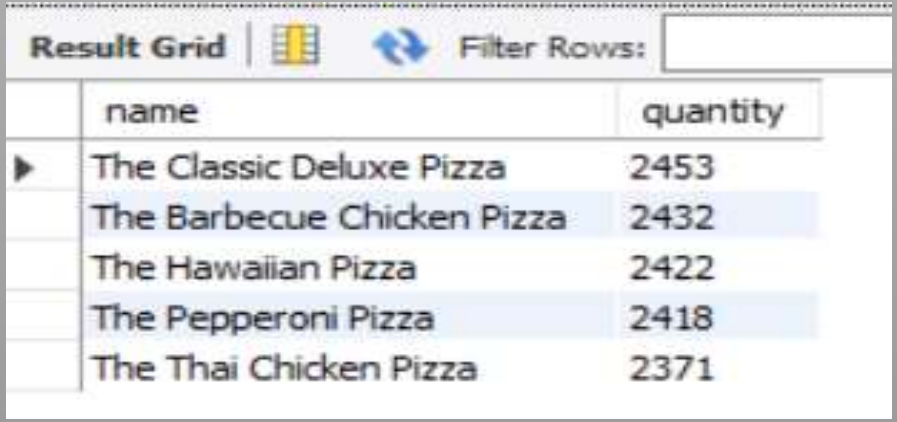
```
size  most_ordered_count
L     18526
```

VIVEK KUMAR

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



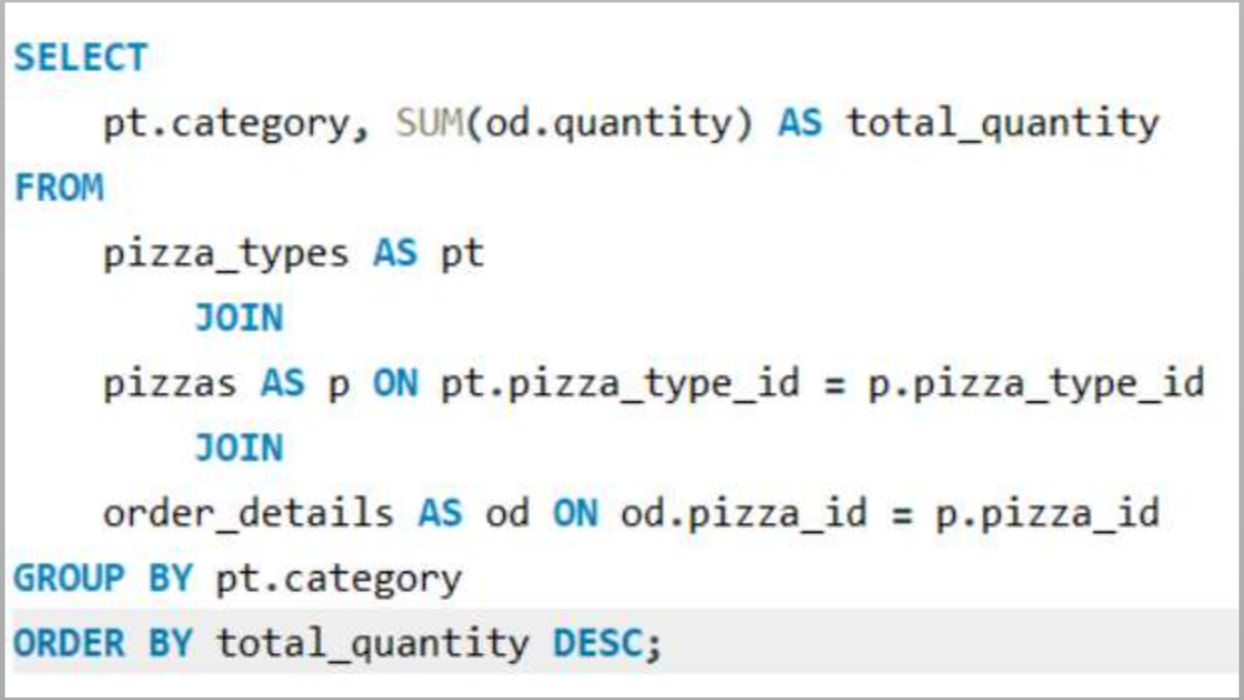
```
SELECT
    pt.name, SUM(od.quantity) AS quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5;
```



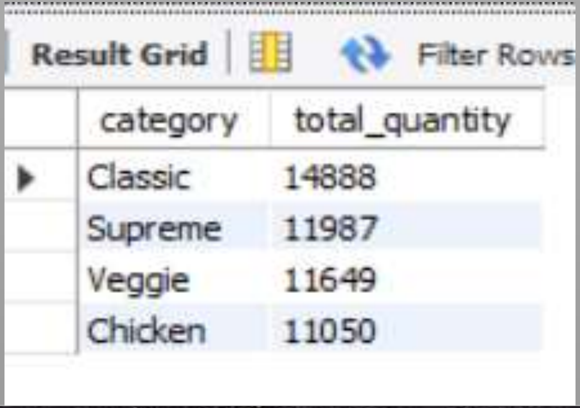
```
name                          quantity
The Classic Deluxe Pizza      2453
The Barbecue Chicken Pizza    2432
The Hawaiian Pizza            2422
The Pepperoni Pizza           2418
The Thai Chicken Pizza        2371
```

VIVEK KUMAR

Join the necessary tables to find the total quantity of each pizza category ordered.



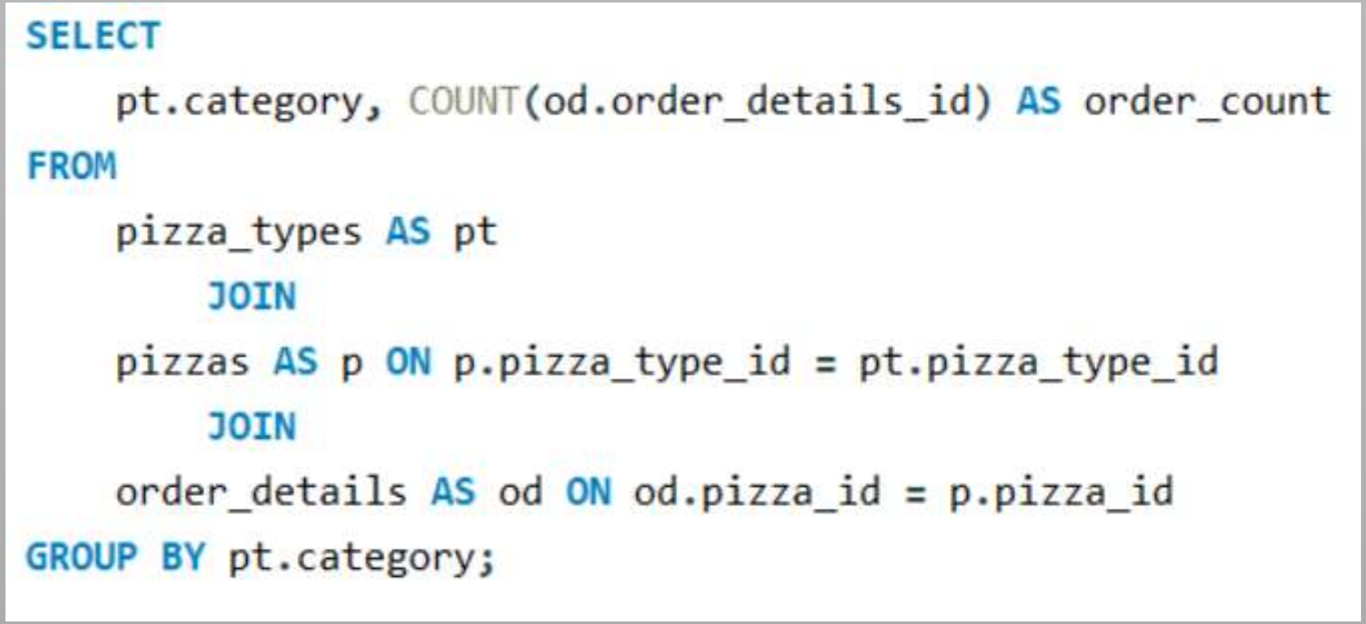
```
SELECT
    pt.category, SUM(od.quantity) AS total_quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY total_quantity DESC;
```



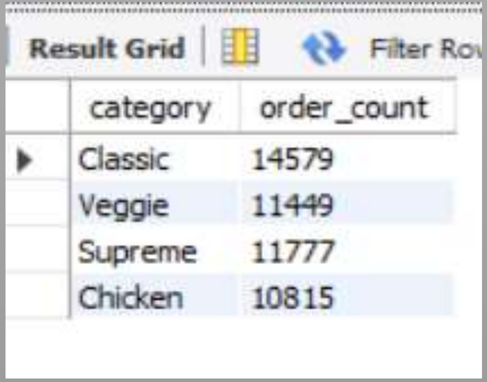
```
category    total_quantity
Classic     14888
Supreme     11987
Veggie      11649
Chicken     11050
```

VIVEK KUMAR

Join relevant tables to find the category-wise distribution of pizzas (category-wise orders).



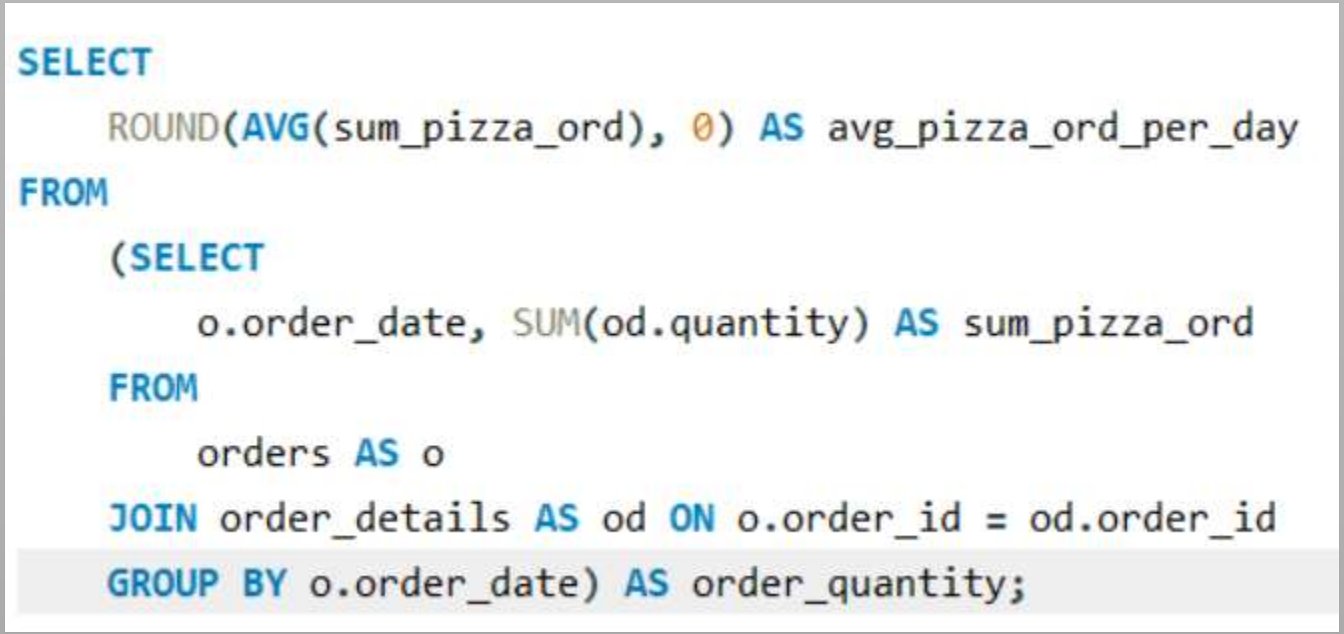
```
SELECT
    pt.category, COUNT(od.order_details_id) AS order_count
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.category;
```



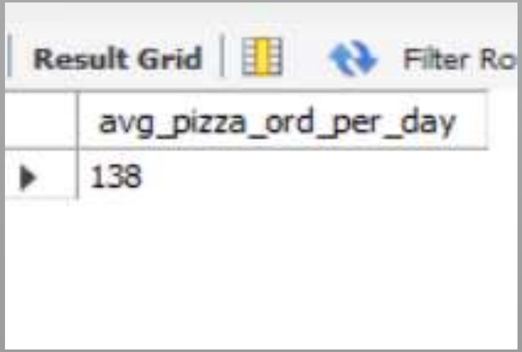
category	order_count
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

VIVEK KUMAR

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



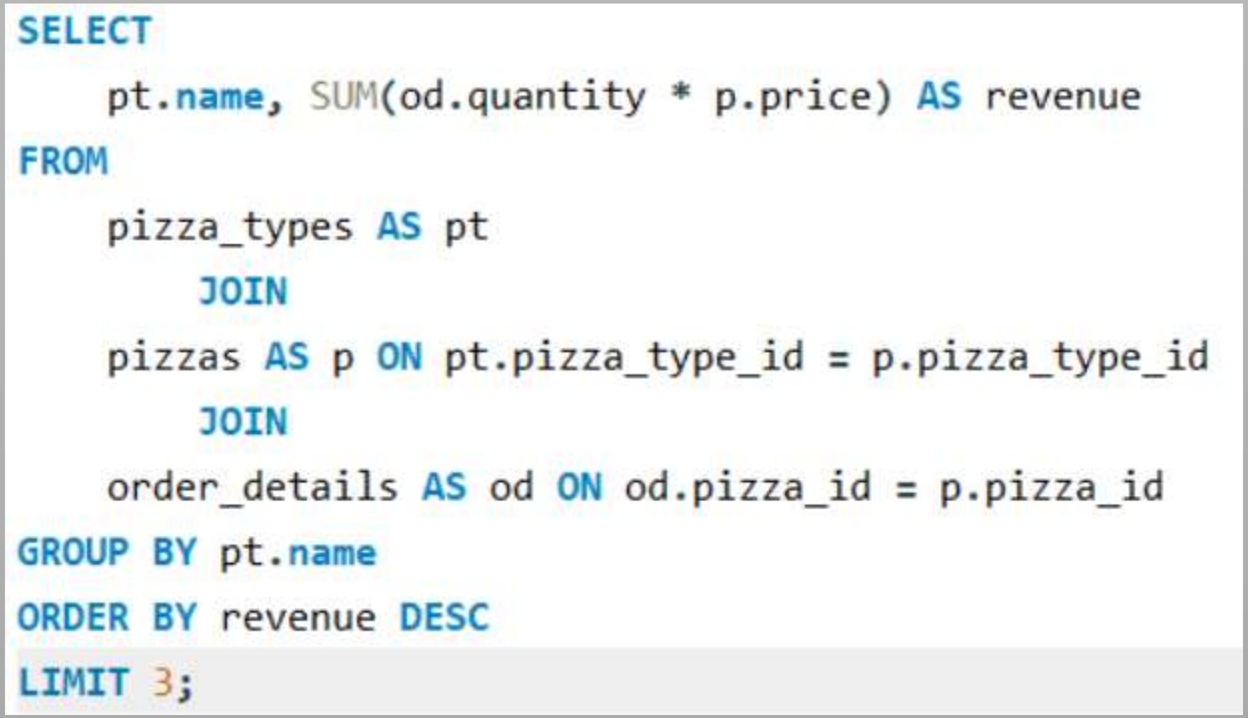
```
SELECT
    ROUND(AVG(sum_pizza_ord), 0) AS avg_pizza_ord_per_day
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS sum_pizza_ord
    FROM
        orders AS o
    JOIN order_details AS od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```



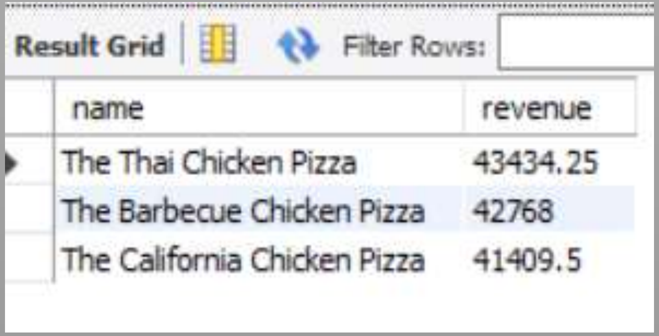
```
Result Grid | Filter Ro
   avg_pizza_ord_per_day
▶  138
```

VIVEK KUMAR

Determine the top 3 most ordered pizza types based on revenue.



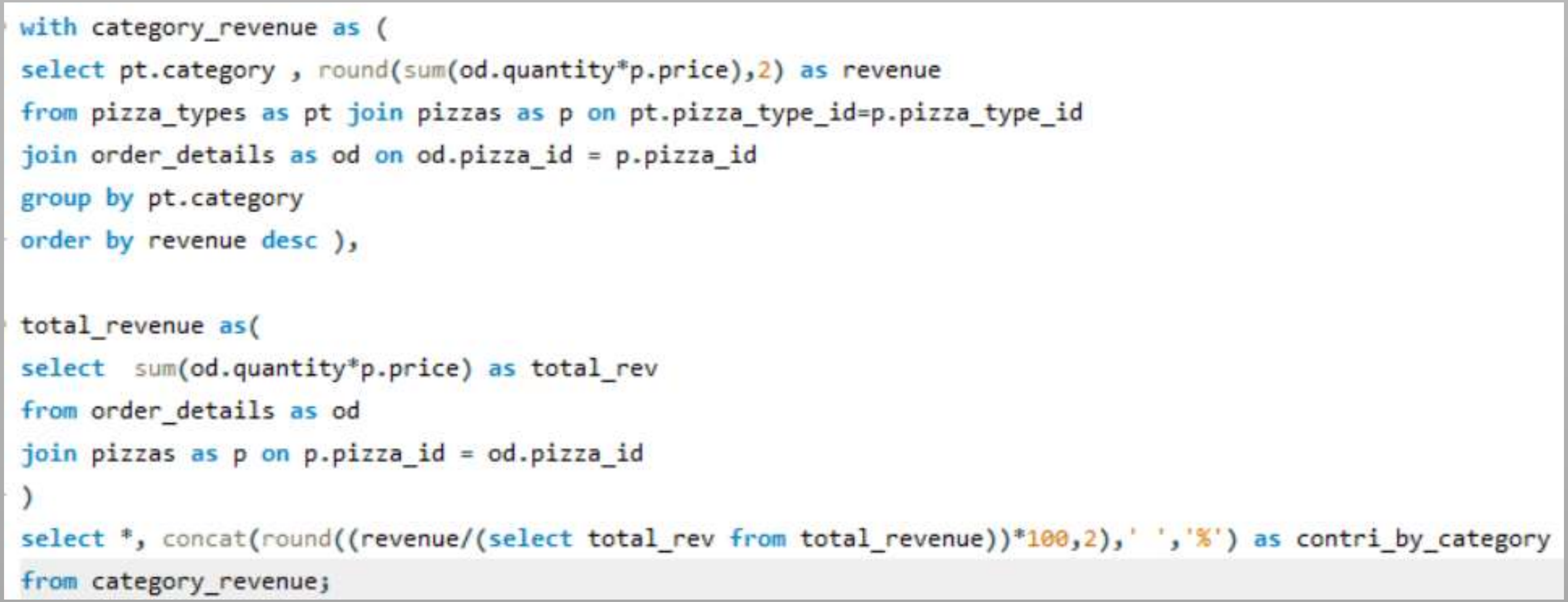
```
SELECT
    pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```



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

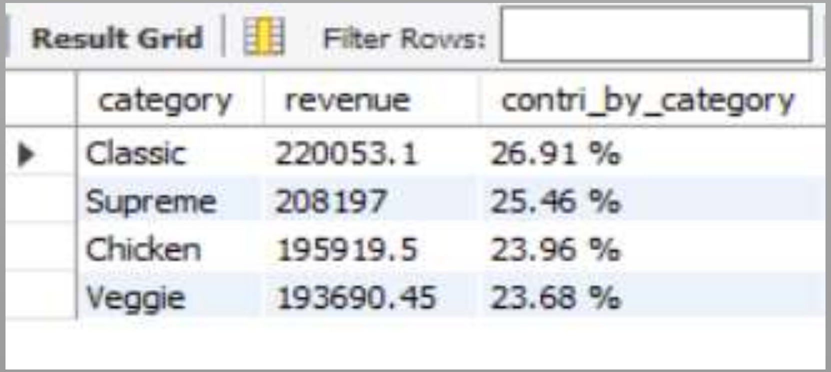
VIVEK KUMAR

Calculate the percentage contribution of each pizza category type to total revenue.



```
with category_revenue as (
select pt.category , round(sum(od.quantity*p.price),2) as revenue
from pizza_types as pt join pizzas as p on pt.pizza_type_id=p.pizza_type_id
join order_details as od on od.pizza_id = p.pizza_id
group by pt.category
order by revenue desc ),

total_revenue as(
select  sum(od.quantity*p.price) as total_rev
from order_details as od
join pizzas as p on p.pizza_id = od.pizza_id
)
select *, concat(round((revenue/(select total_rev from total_revenue))*100,2),' ','%') as contri_by_category
from category_revenue;
```

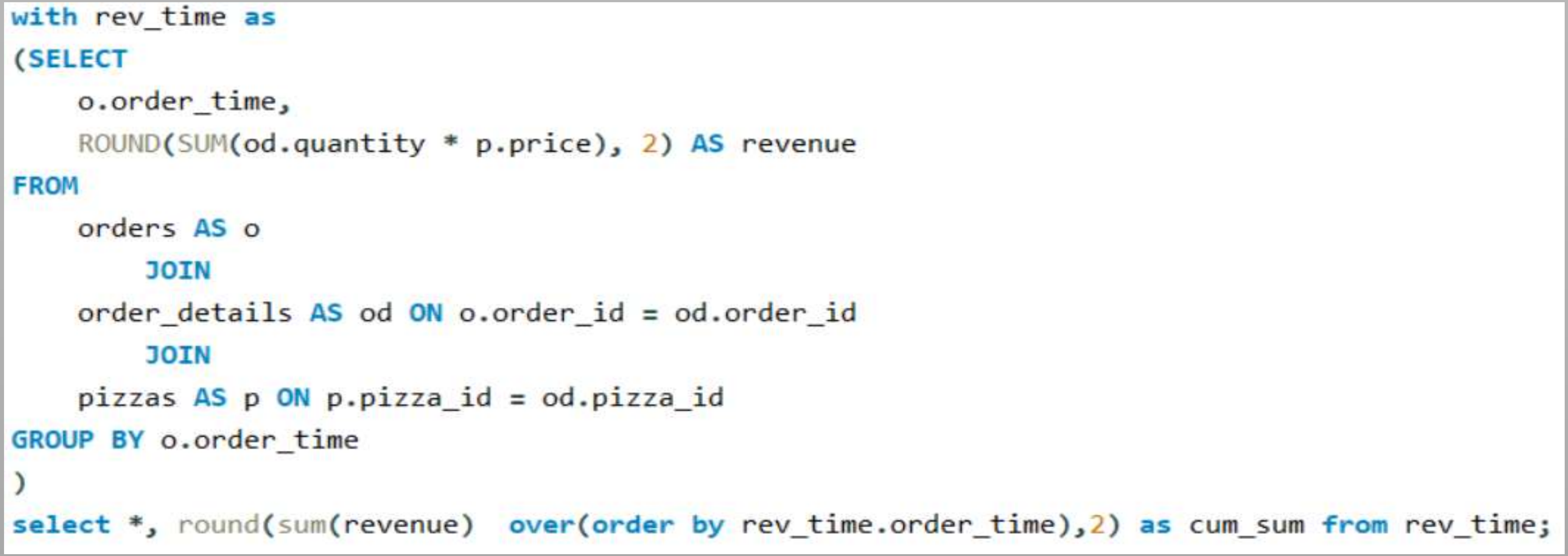


Result Grid
Filter Rows:

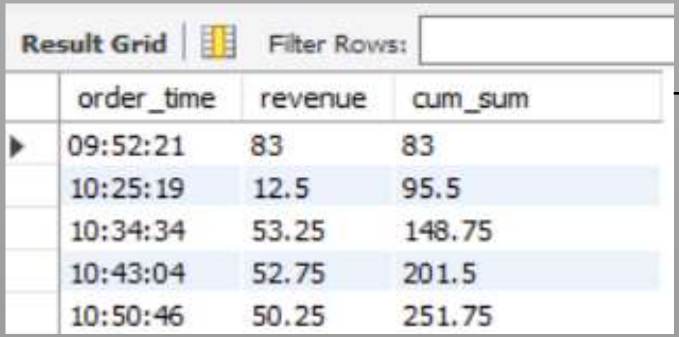
category	revenue	contri_by_category
Classic	220053.1	26.91 %
Supreme	208197	25.46 %
Chicken	195919.5	23.96 %
Veggie	193690.45	23.68 %

VIVEK KUMAR

Analyse the cumulative revenue generated over time.

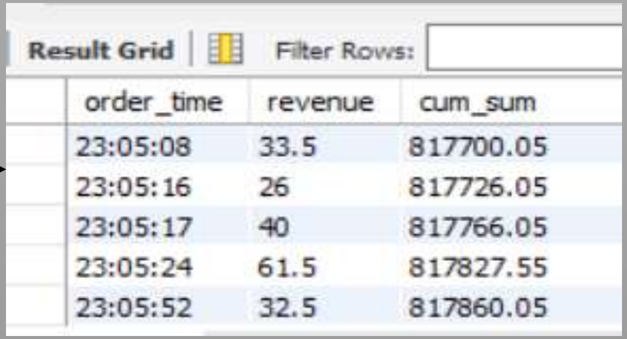


```
with rev_time as
(SELECT
    o.order_time,
    ROUND(SUM(od.quantity * p.price), 2) AS revenue
FROM
    orders AS o
        JOIN
    order_details AS od ON o.order_id = od.order_id
        JOIN
    pizzas AS p ON p.pizza_id = od.pizza_id
GROUP BY o.order_time
)
select *, round(sum(revenue)  over(order by rev_time.order_time),2) as cum_sum from rev_time;
```



Result Grid | Filter Rows:

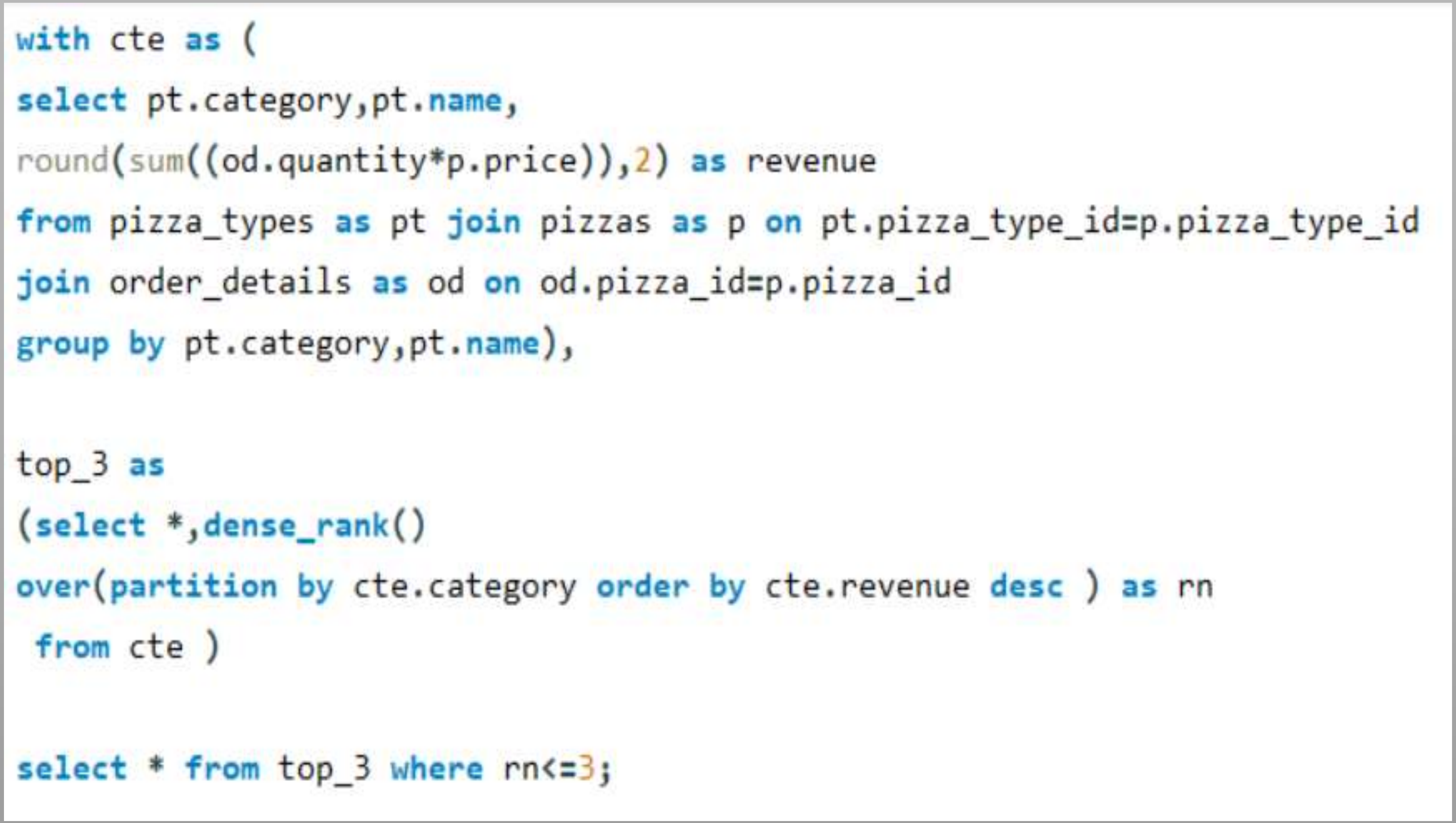
order_time	revenue	cum_sum
09:52:21	83	83
10:25:19	12.5	95.5
10:34:34	53.25	148.75
10:43:04	52.75	201.5
10:50:46	50.25	251.75



Result Grid | Filter Rows:

order_time	revenue	cum_sum
23:05:08	33.5	817700.05
23:05:16	26	817726.05
23:05:17	40	817766.05
23:05:24	61.5	817827.55
23:05:52	32.5	817860.05

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



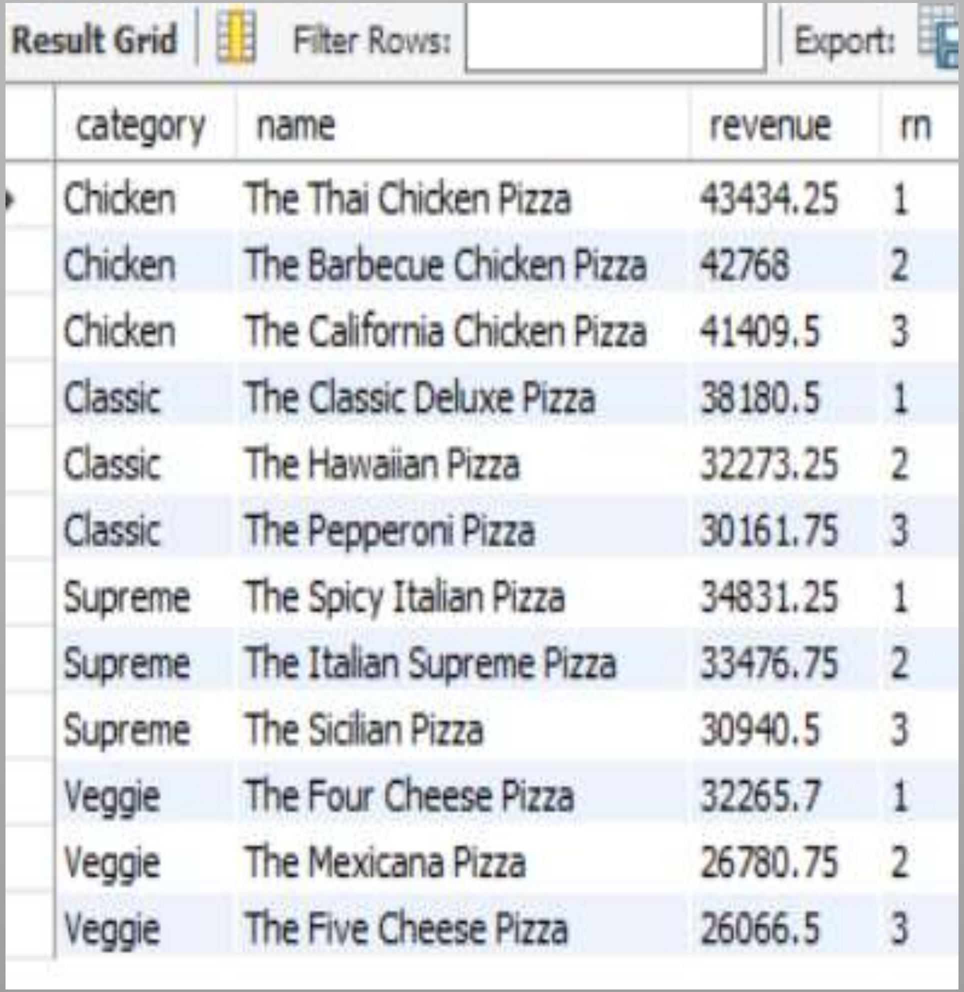
```
with cte as (
select pt.category,pt.name,
round(sum((od.quantity*p.price)),2) as revenue
from pizza_types as pt join pizzas as p on pt.pizza_type_id=p.pizza_type_id
join order_details as od on od.pizza_id=p.pizza_id
group by pt.category,pt.name),

top_3 as
(select *,dense_rank()
over(partition by cte.category order by cte.revenue desc ) as rn
 from cte )

select * from top_3 where rn<=3;
```

VIVEK KUMAR

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



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
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.7	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

VIVEK KUMAR

Conclusion

- This project gave us clear insights into what sells best, when sales peak, and what customers like.

- This analysis demonstrates the power of SQL in delivering actionable business intelligence, emphasizing the importance of data-driven decision-making in the competitive food industry.

VIVEK KUMAR

THANK YOU

VIVEK KUMAR