Pizza Sales

SQL Project

Let’s Go

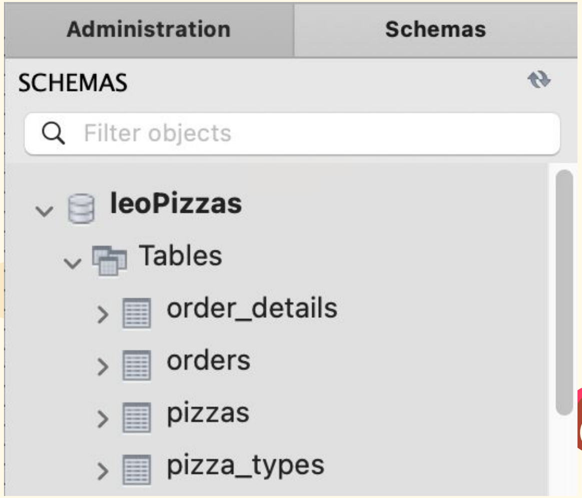
ABOUT THE Project

Hello, my name is Frenny. In this project, I have utilized SQL queries to solve various questions related to pizza sales analysis.

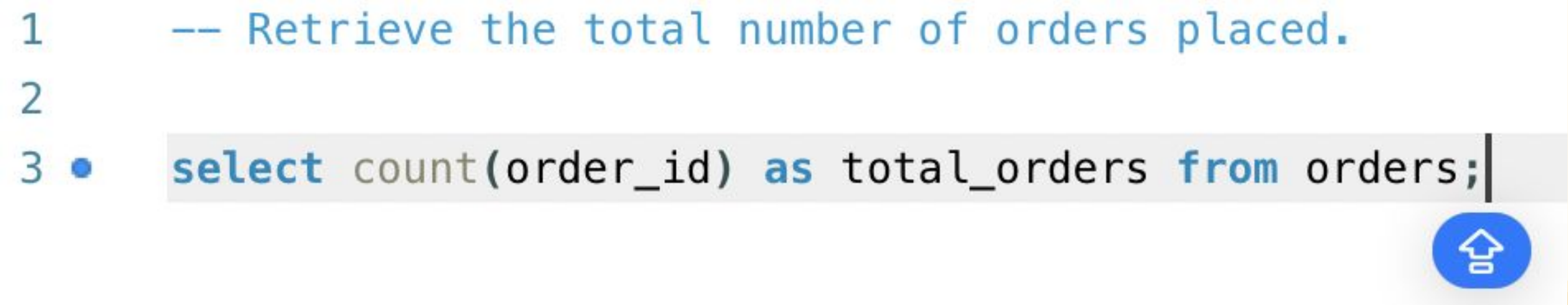
Questions?

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.
6. Join the necessary tables to find the total quantity of each pizza category.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyse the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

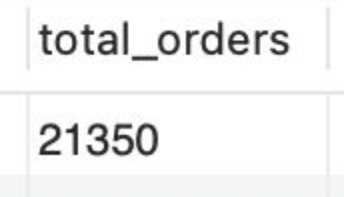
DataBase



1. Retrieve the total number of orders placed.

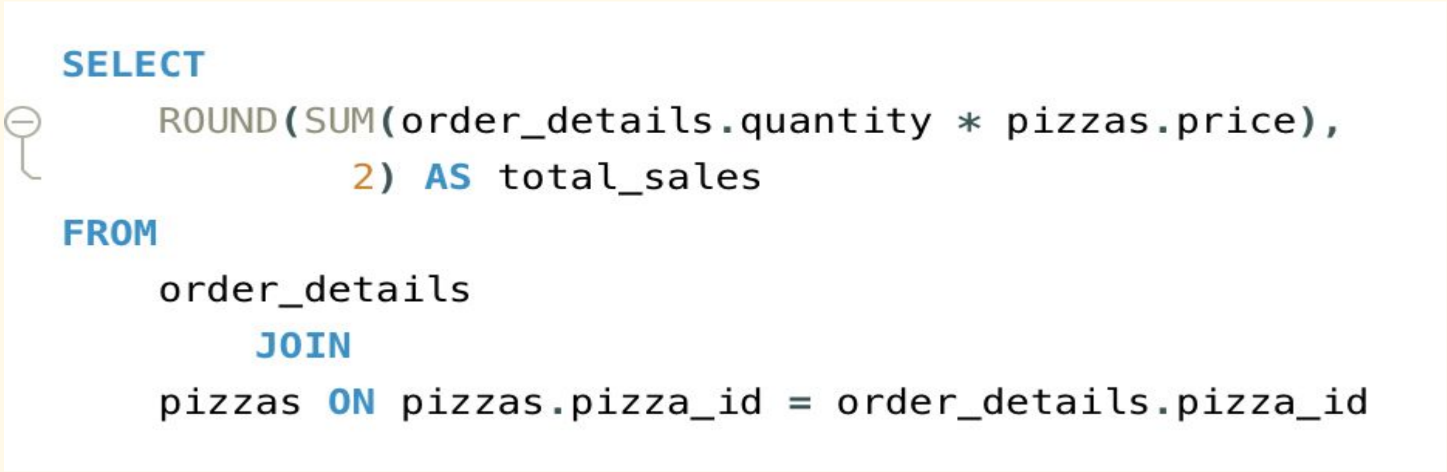


```sql
-- Retrieve the total number of orders placed.

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

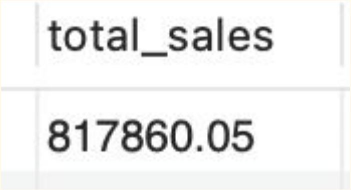


total_orders
21350

2. Calculate the total revenue generated from pizza sales.

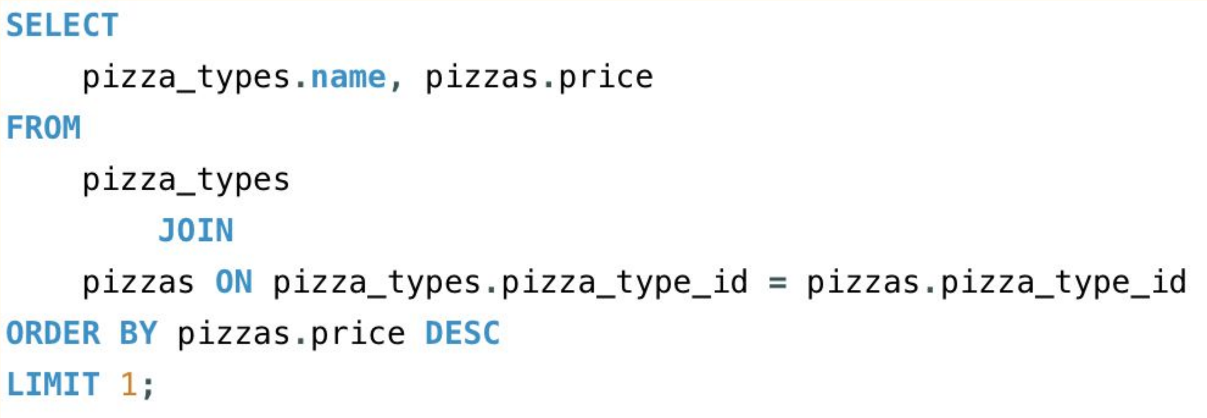


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

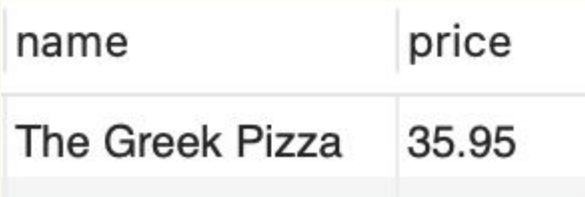


total_sales
817860.05

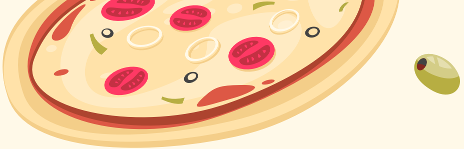
3. Identify the highest-priced pizza.



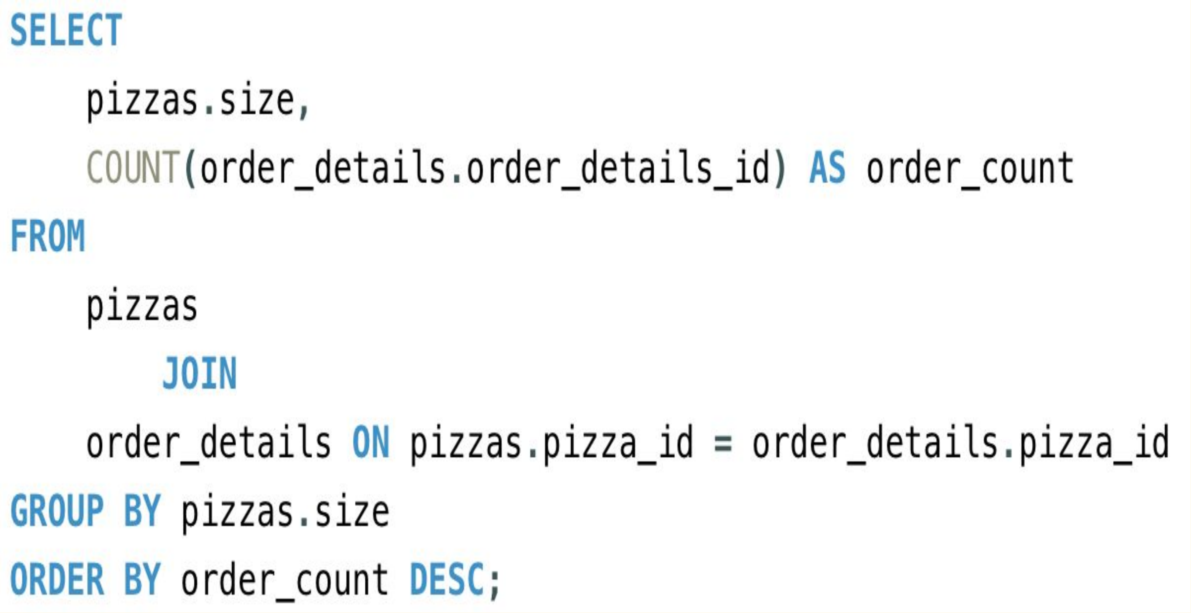
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



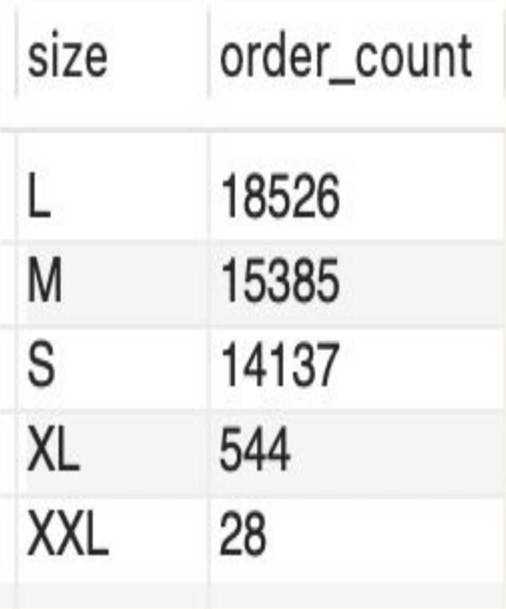
name	price
The Greek Pizza	35.95



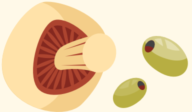
4. Identify the most common pizza size ordered.

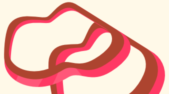


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

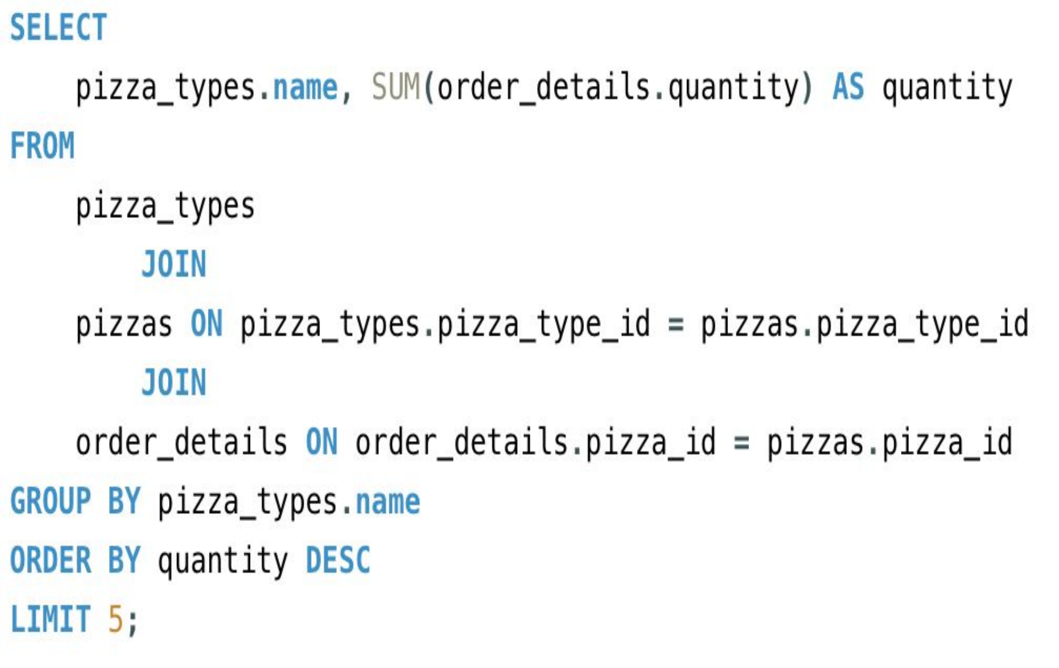


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

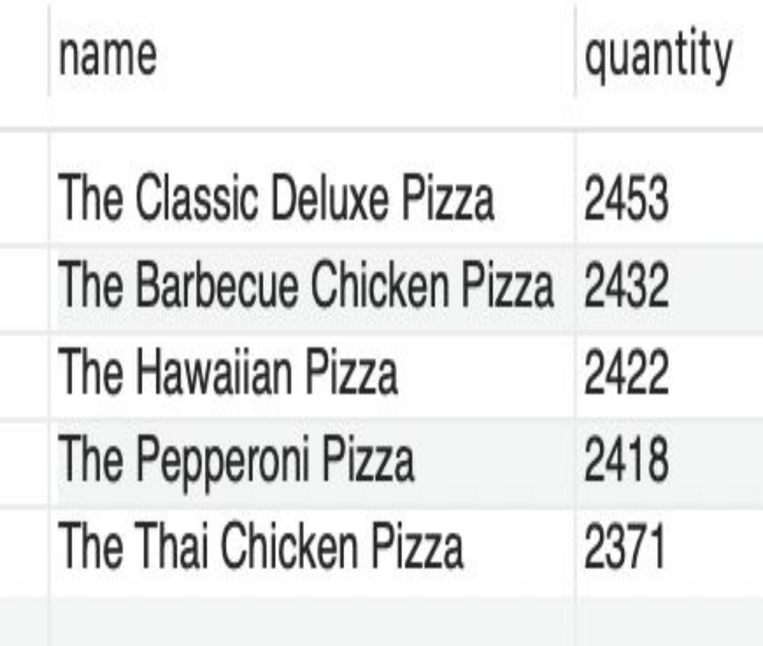




5. List the top 5 most ordered pizza types along with their quantities.

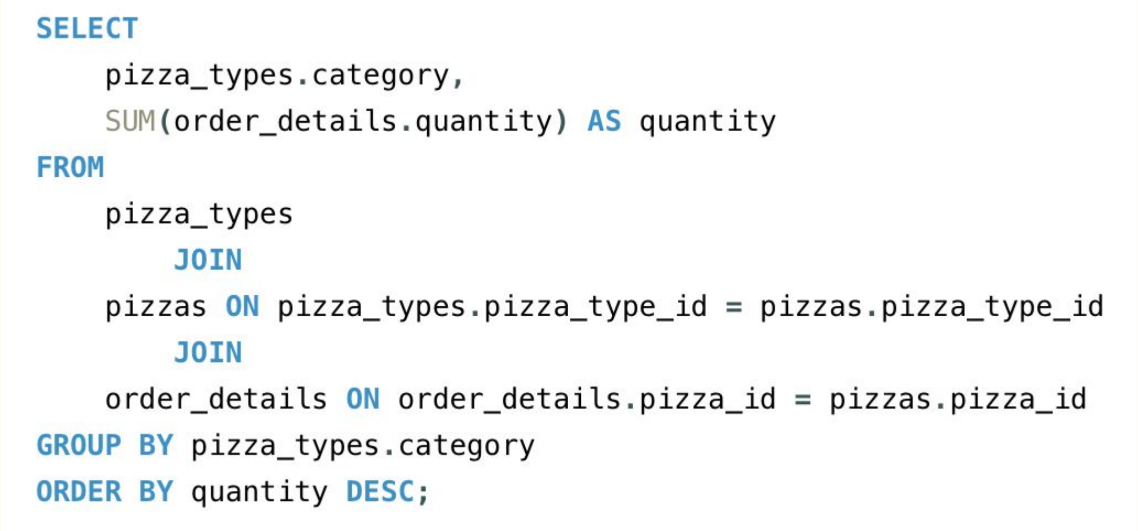


```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

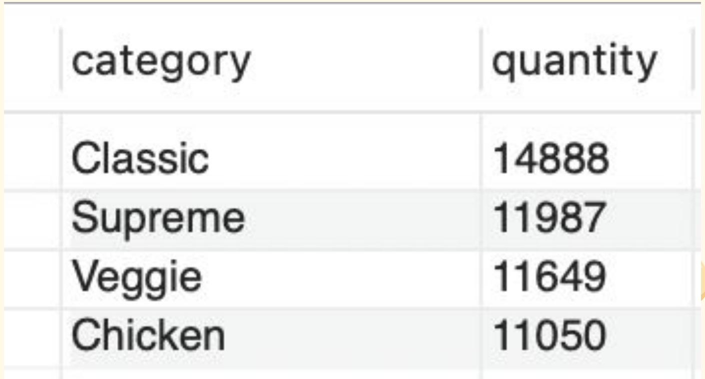


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

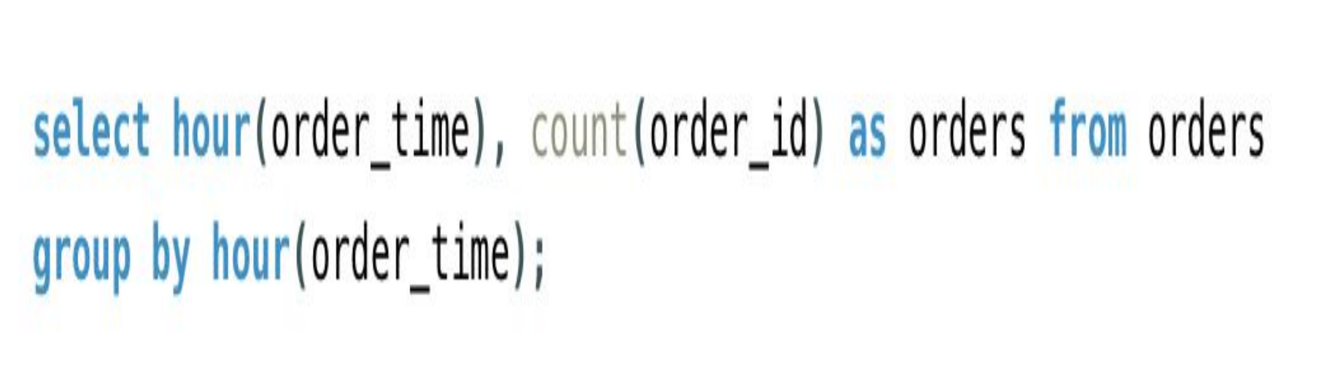
6. Join the necessary tables to find the total quantity of each pizza category.



```
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;
```

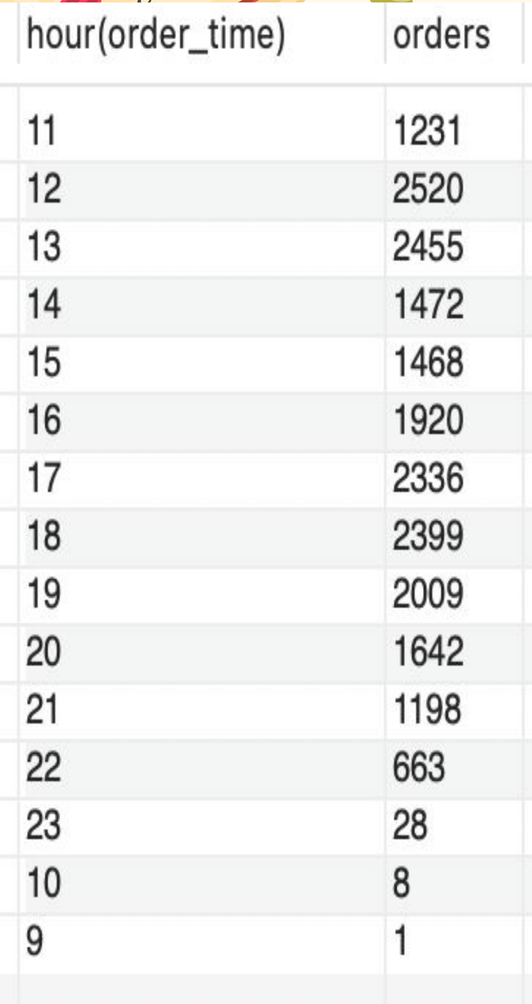


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



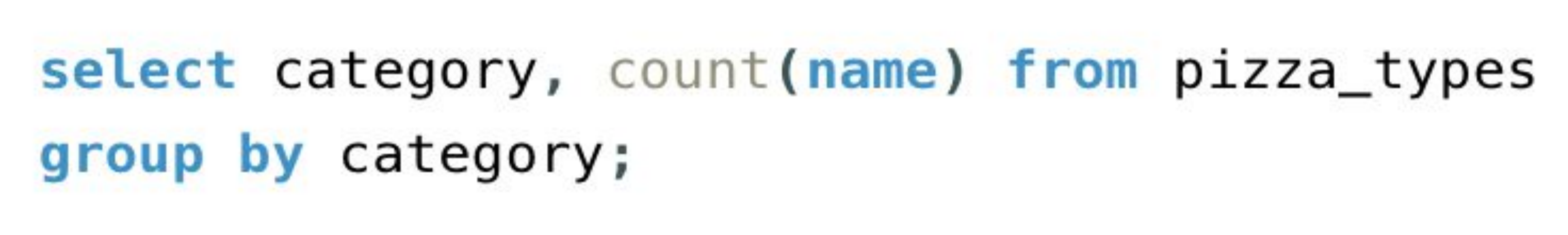
```
select hour(order_time), count(order_id) as orders from orders
group by hour(order_time);
```

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

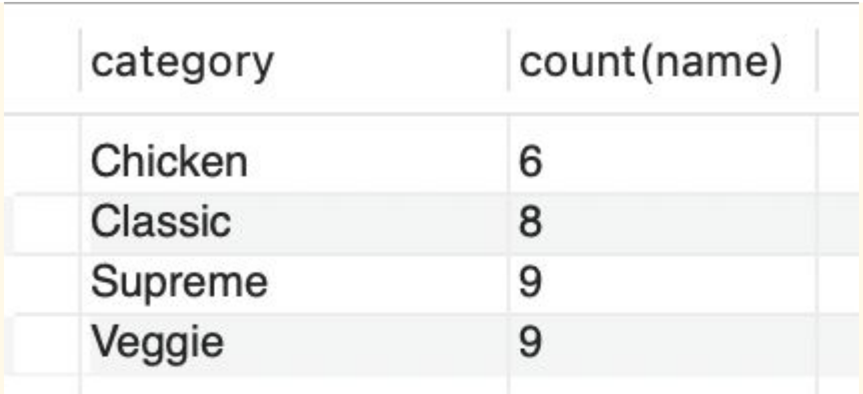


hour(order_time)	orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

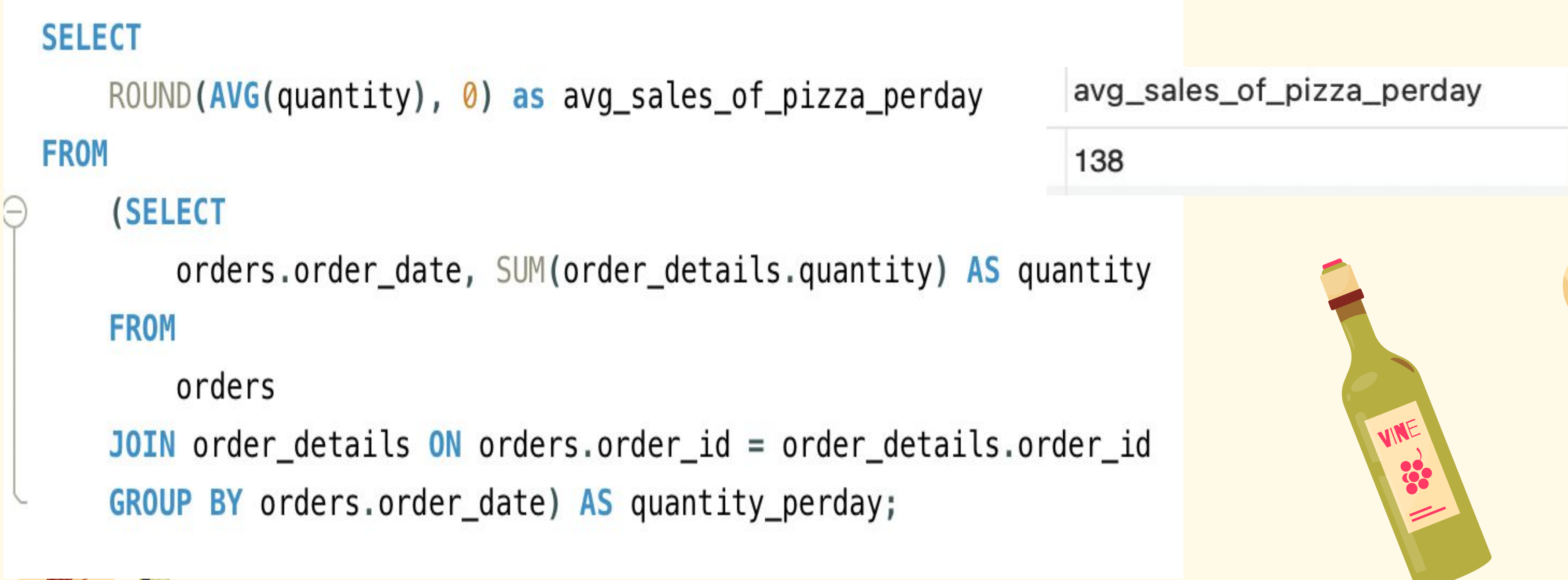


```
select category, count(name) from pizza_types
group by category;
```



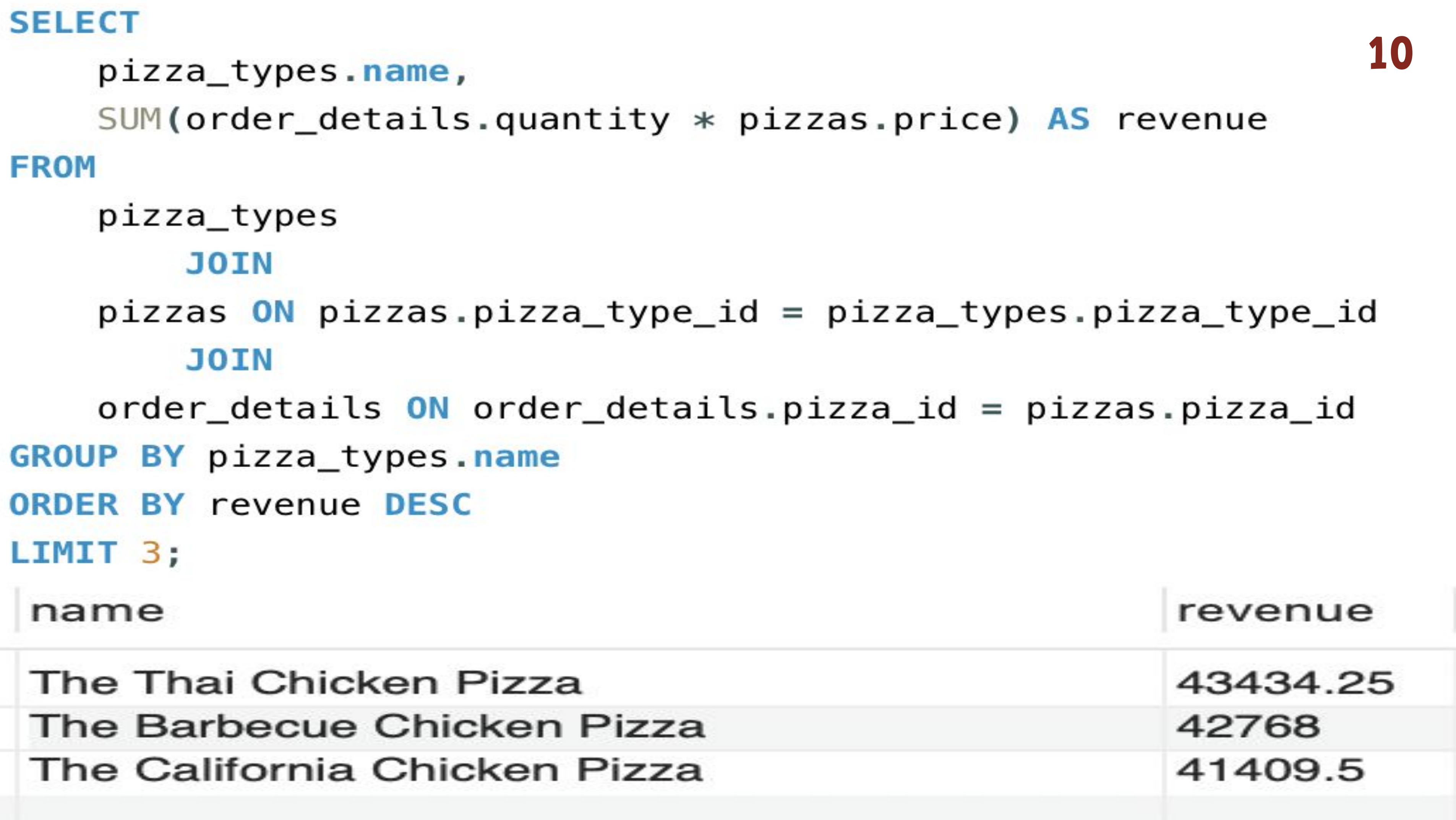
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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



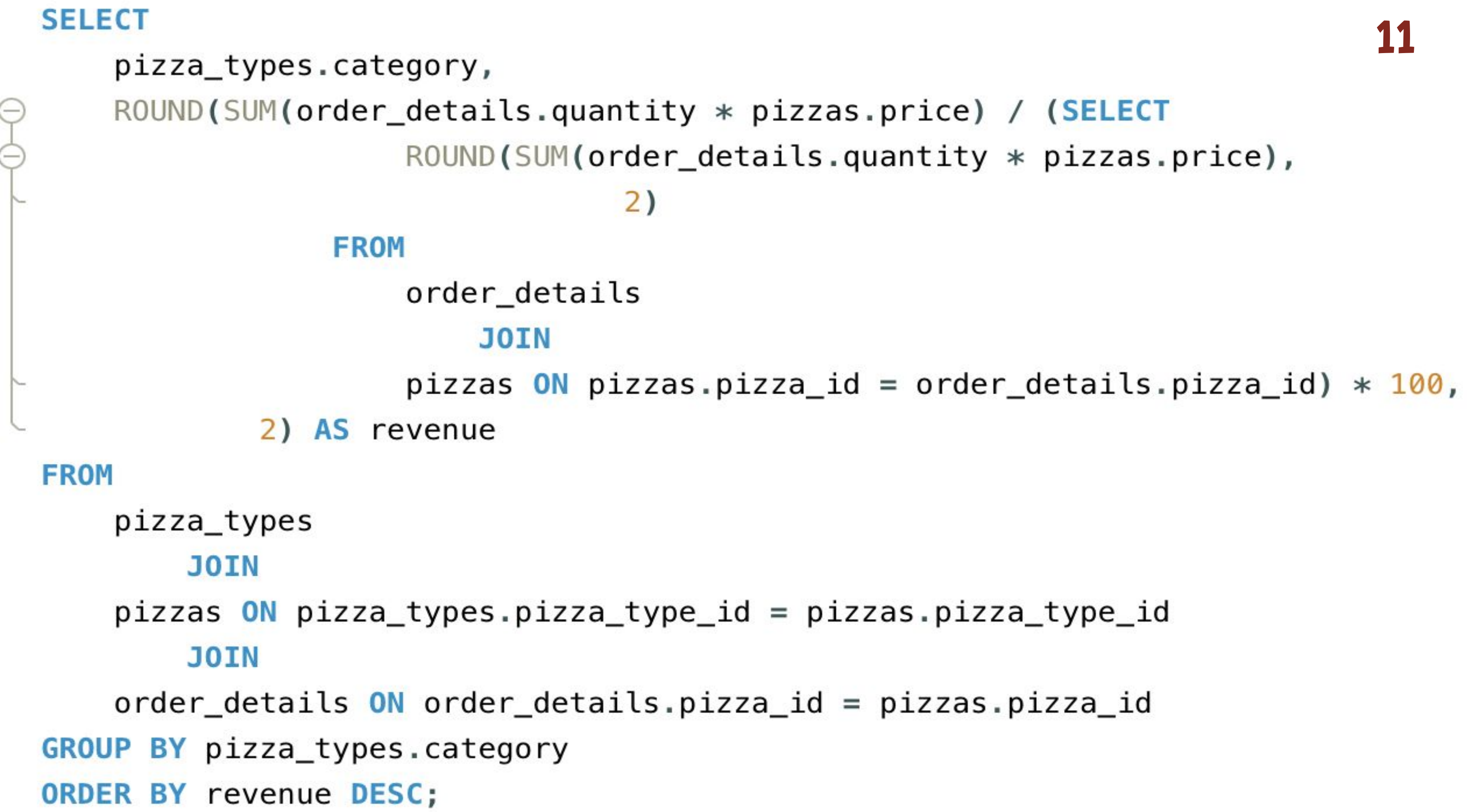
```
SELECT
    ROUND(AVG(quantity), 0) as avg_sales_of_pizza_perday
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS quantity_perday;
```

```
avg_sales_of_pizza_perday
138
```



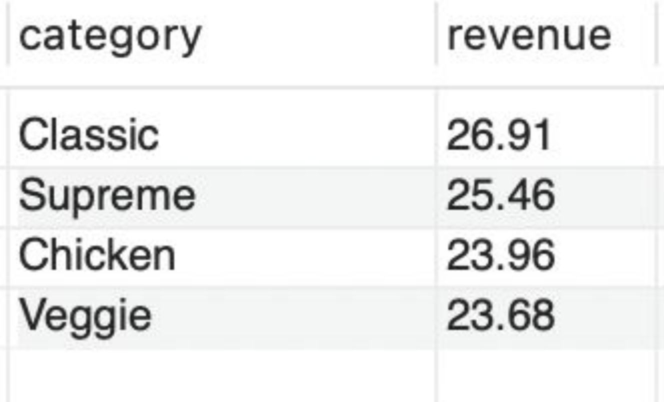
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

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



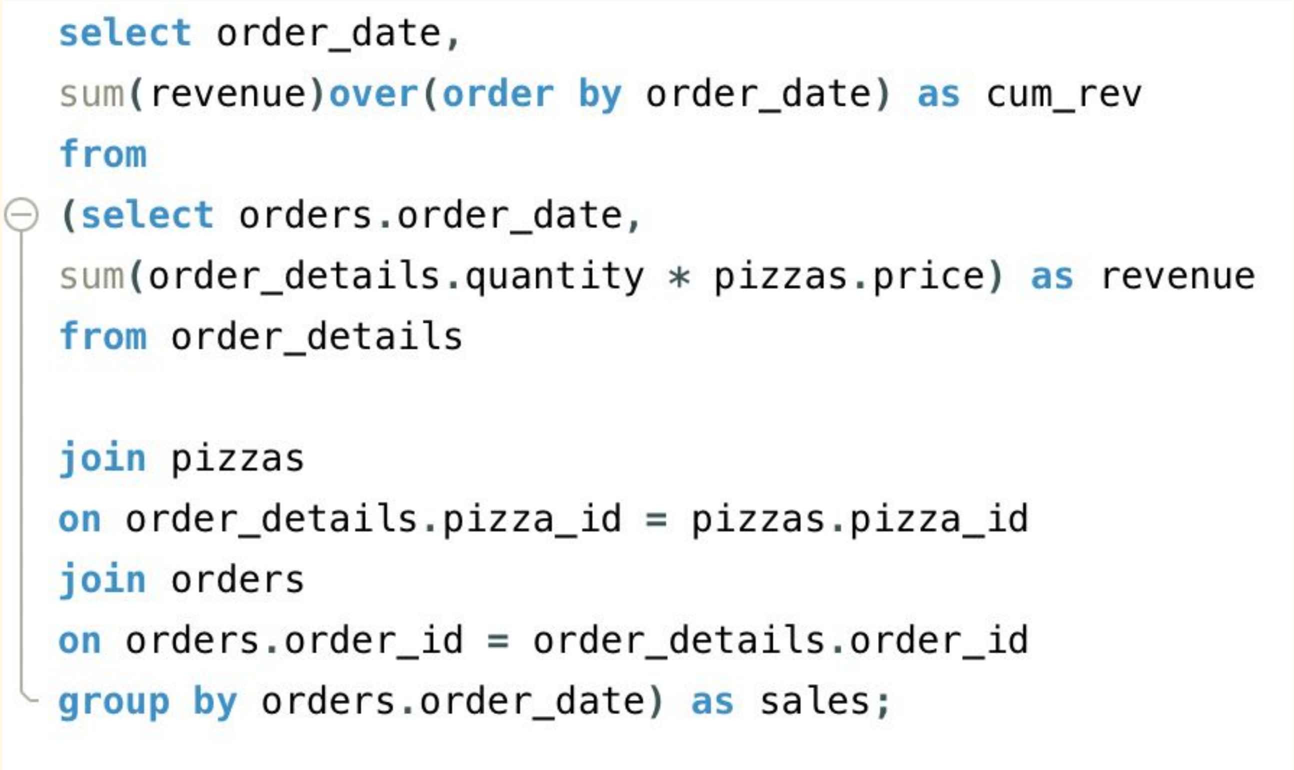
```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2)
                FROM
                    order_details
                        JOIN
                    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;
```

11

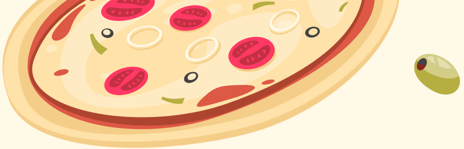


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

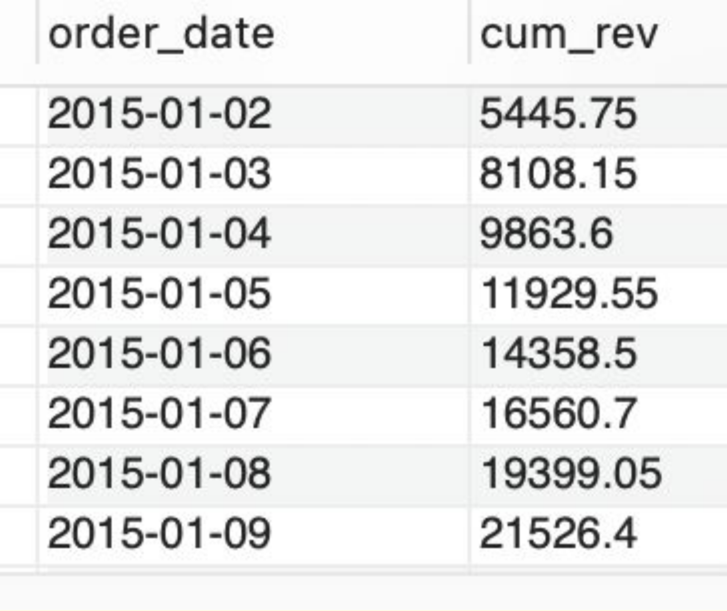
12. Analyse the cumulative revenue generated over time.



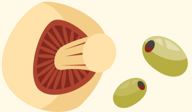
```
select order_date,
sum(revenue)over(order by order_date) as cum_rev
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details

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;
```

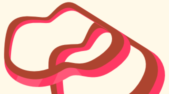


12

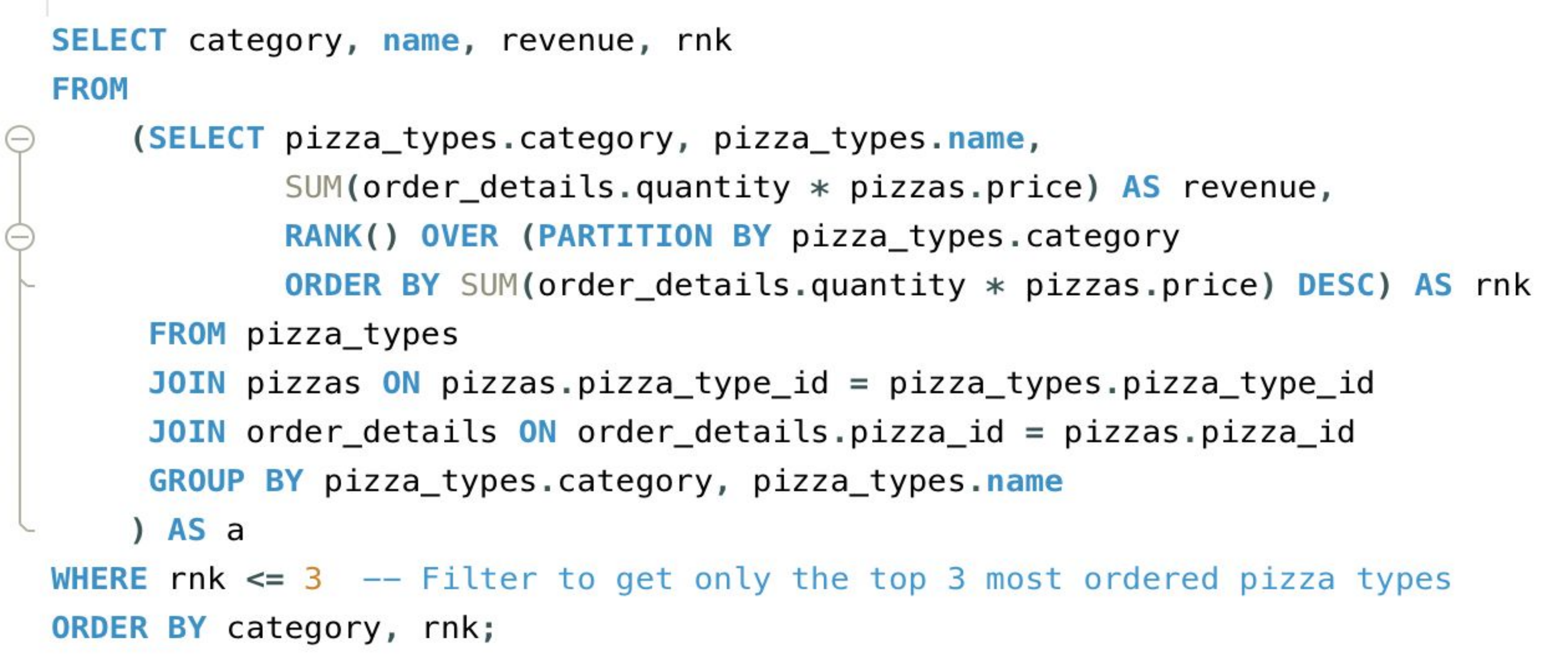


order_date	cum_rev
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4

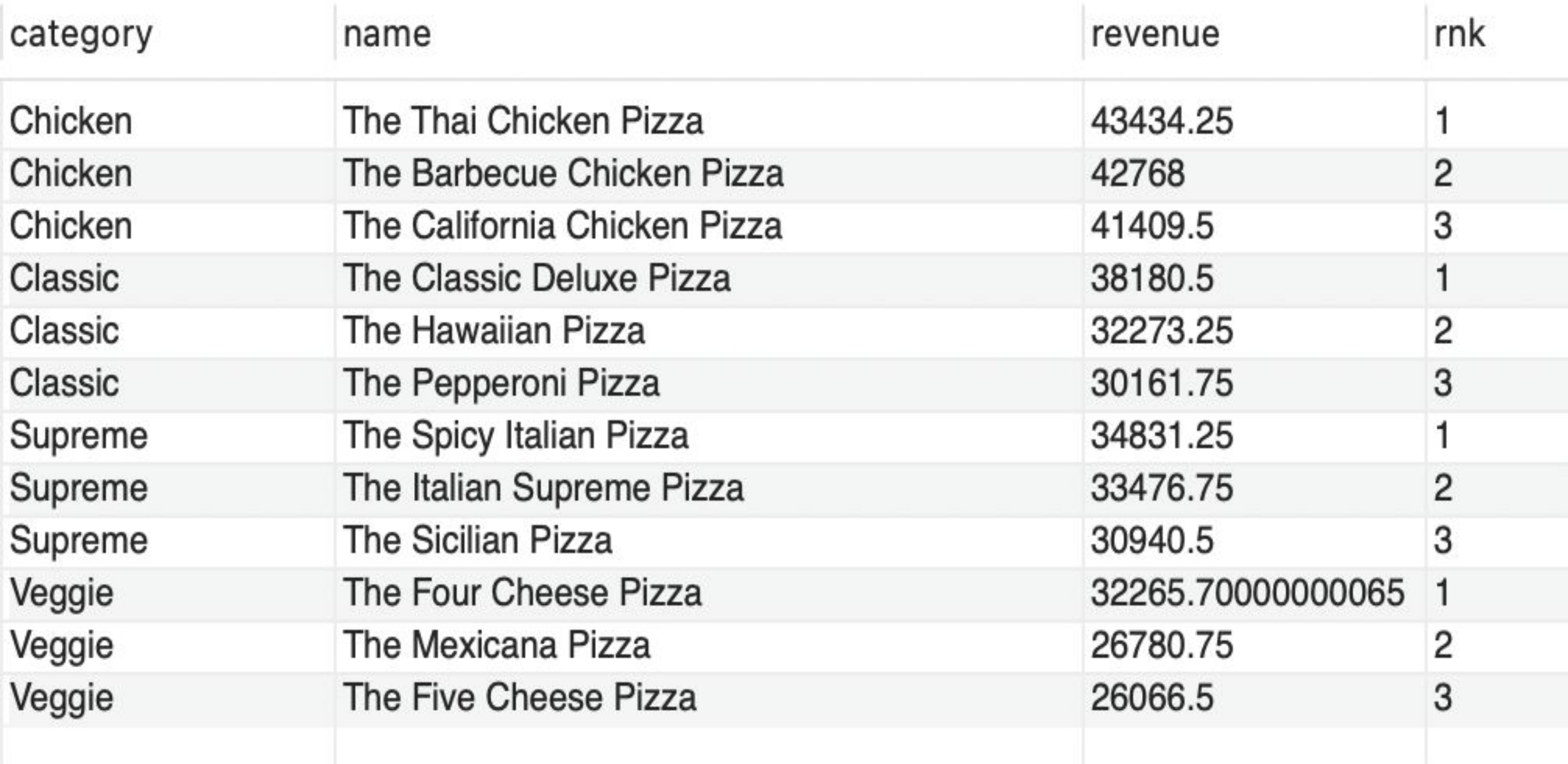




13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```sql
SELECT category, name, revenue, rnk
FROM
    (SELECT pizza_types.category, pizza_types.name,
            SUM(order_details.quantity * pizzas.price) AS revenue,
            RANK() OVER (PARTITION BY pizza_types.category
            ORDER BY SUM(order_details.quantity * pizzas.price) DESC) AS rnk
     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.category, pizza_types.name
    ) AS a
WHERE rnk <= 3  -- Filter to get only the top 3 most ordered pizza types
ORDER BY category, rnk;
```



category	name	revenue	rnk
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	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

Thank You !!!!