PIZZA SALES SQL PROJECT

Divakar gulati

This is a pizza sales SQL Project based on the pizza shop database.

Following are the questions which will be answered from this database :-

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

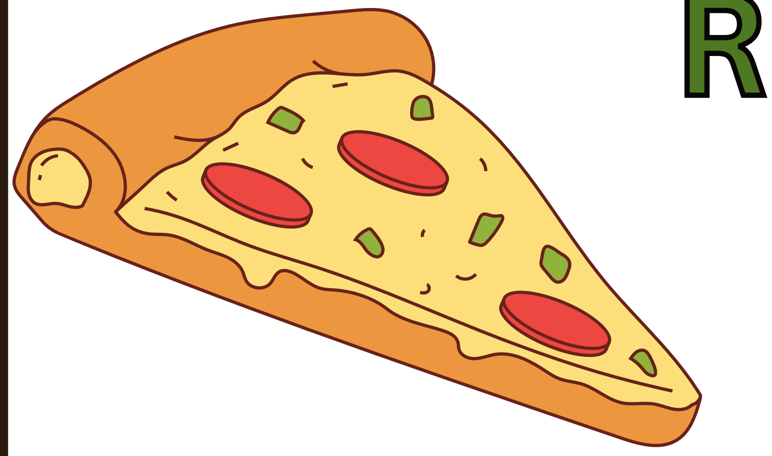
Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

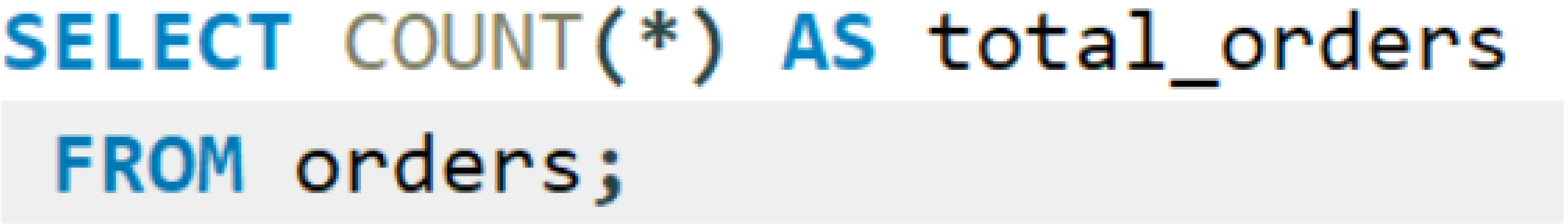
Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Divakar gulati

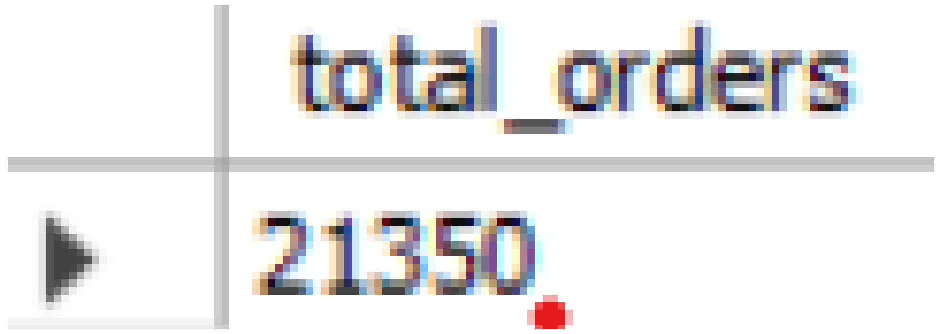


Retrieve the total number of orders placed.

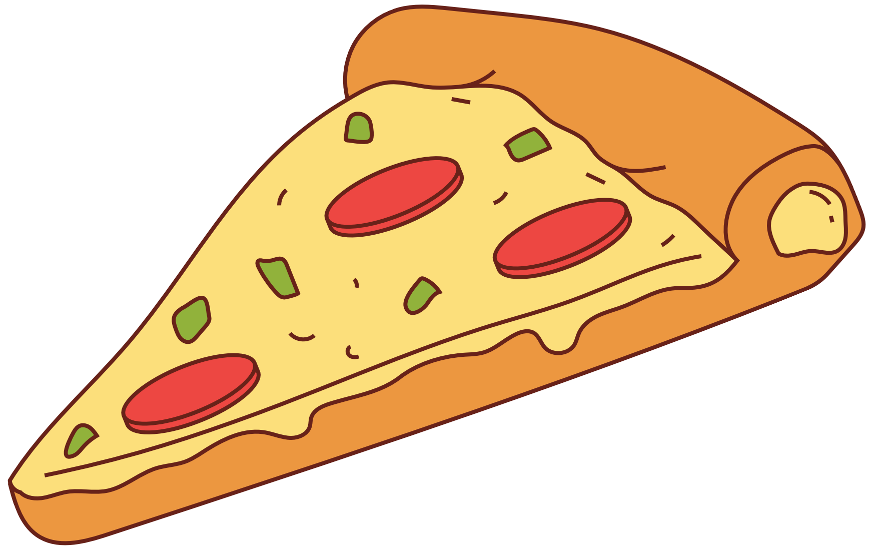


```sql
SELECT COUNT(*) AS total_orders
 FROM orders;
```

Output

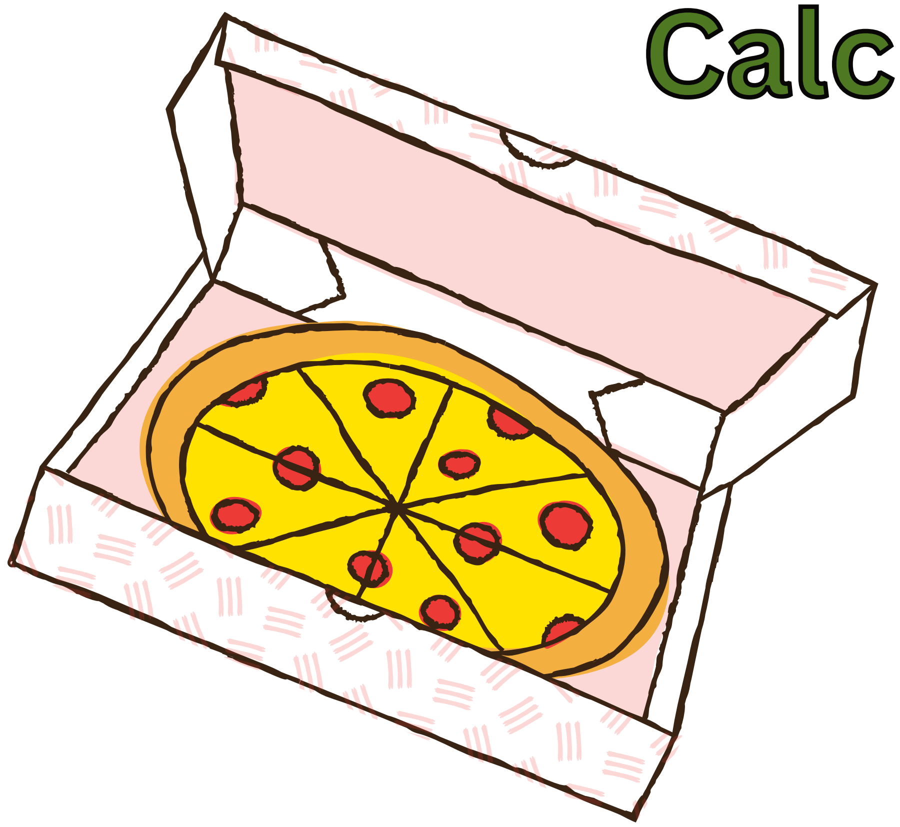


```
total_orders
21350
```

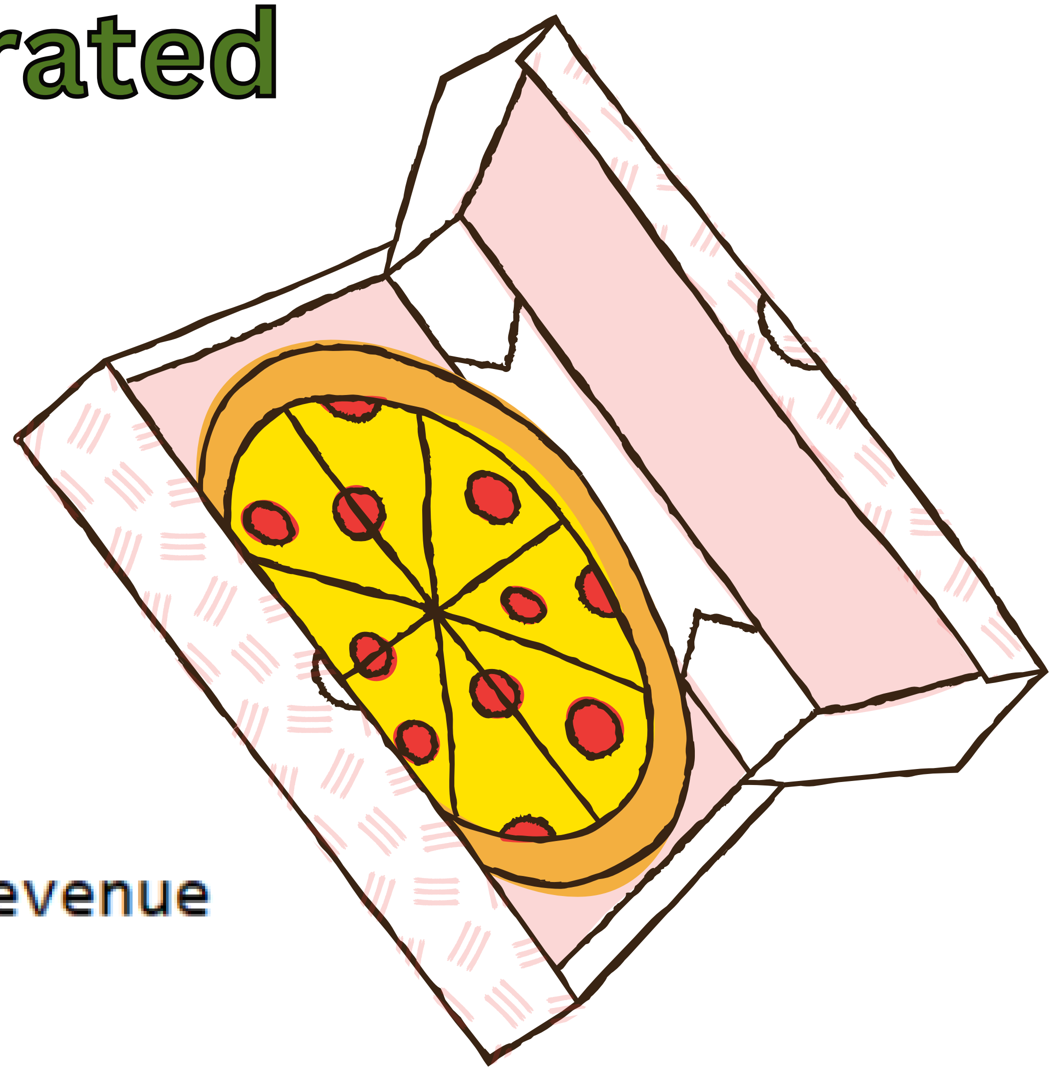


This pizza is amazing!

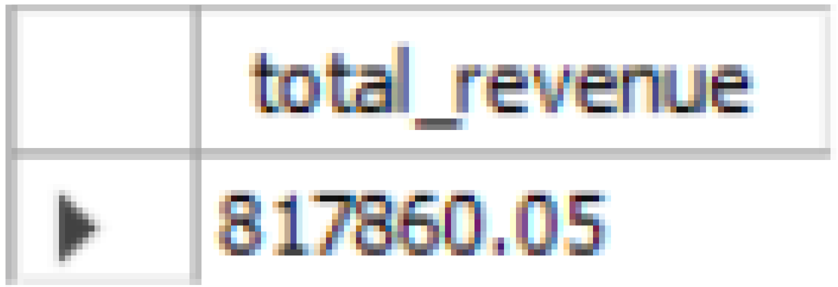
Divakar gulati



Calculate the total revenue generated from pizza sales.



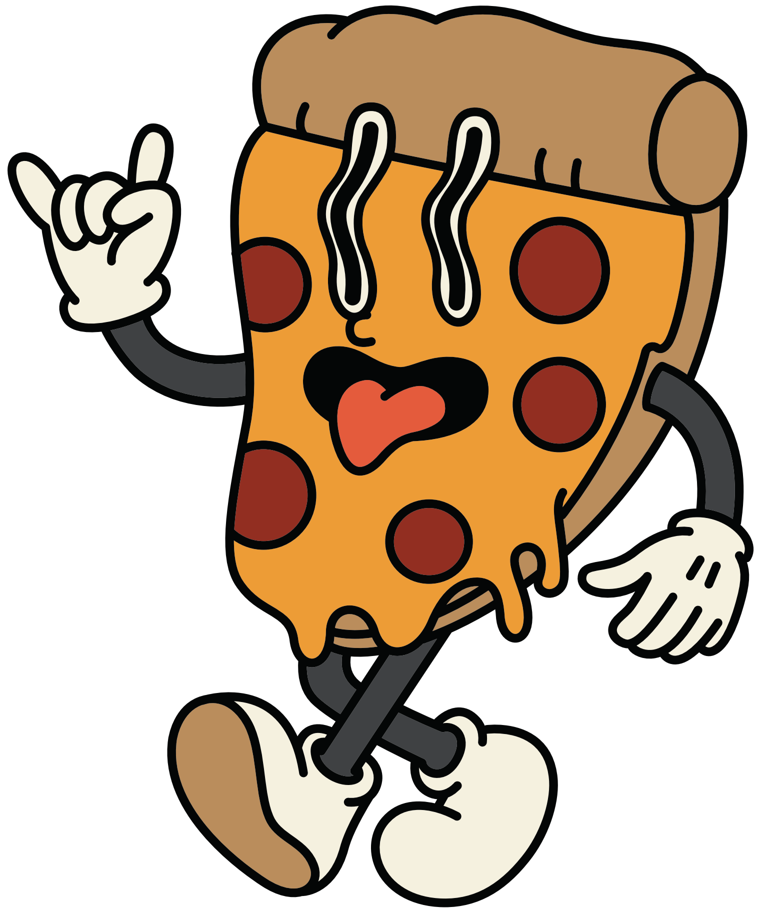
```
SELECT ROUND(SUM(order_details.quantity*pizzas.price),2) AS total_revenue
FROM order_details
INNER JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id;
```



```
total_revenue
817860.05
```

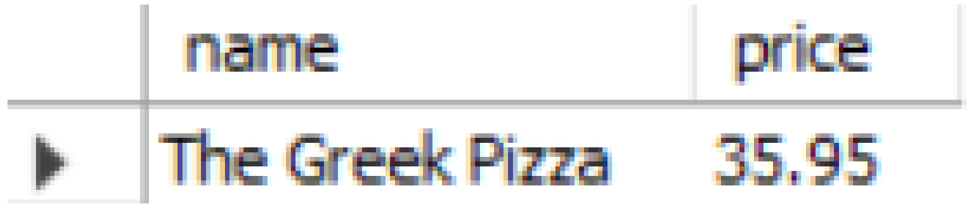
Thank you for the pizza.

Identify the highest-priced pizza.



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

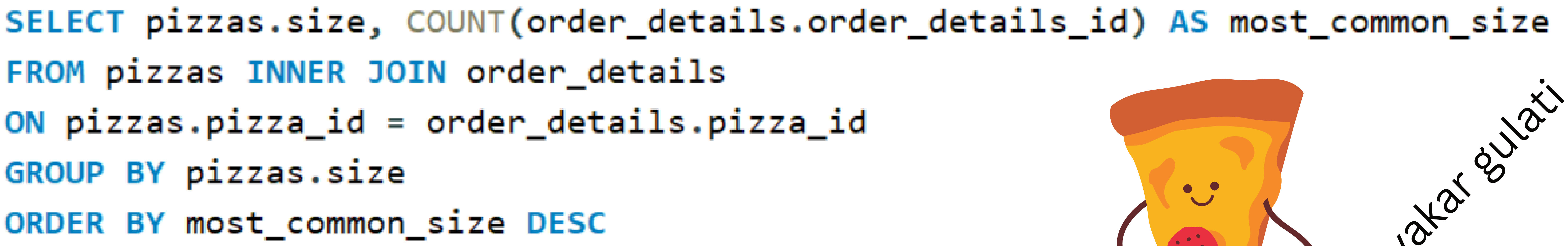
Output



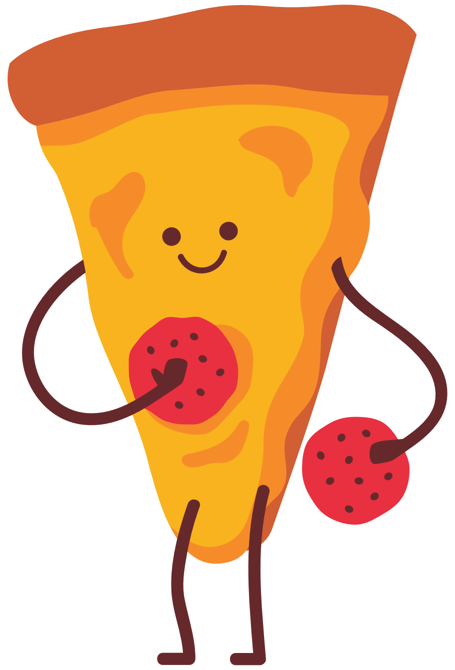
name	price
The Greek Pizza	35.95

Divakar gulati

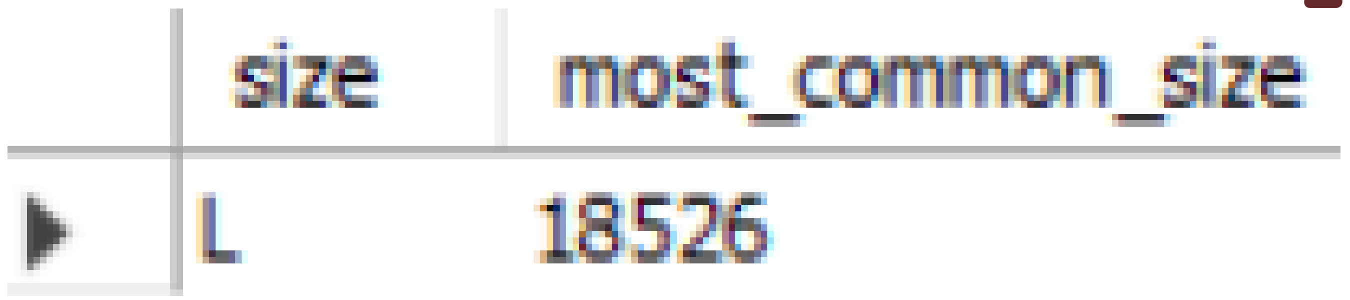
Identify the most common pizza size ordered.



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

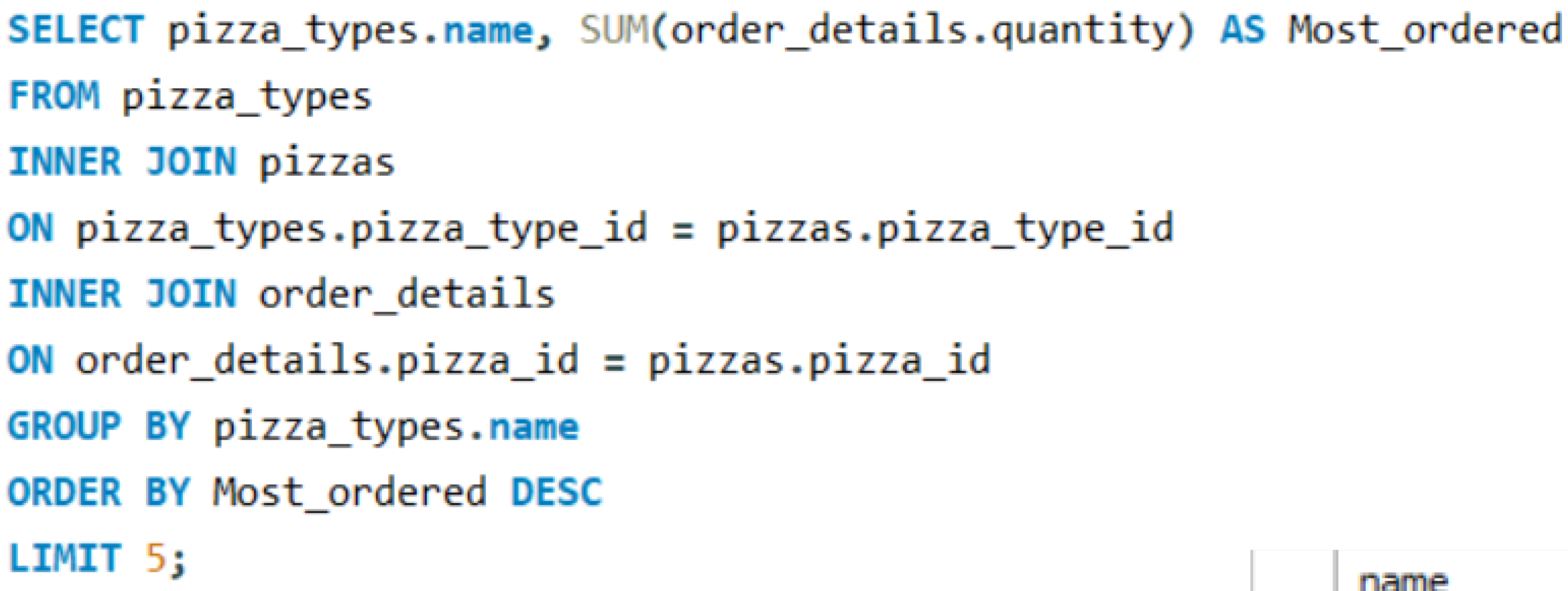


Divakar gulati

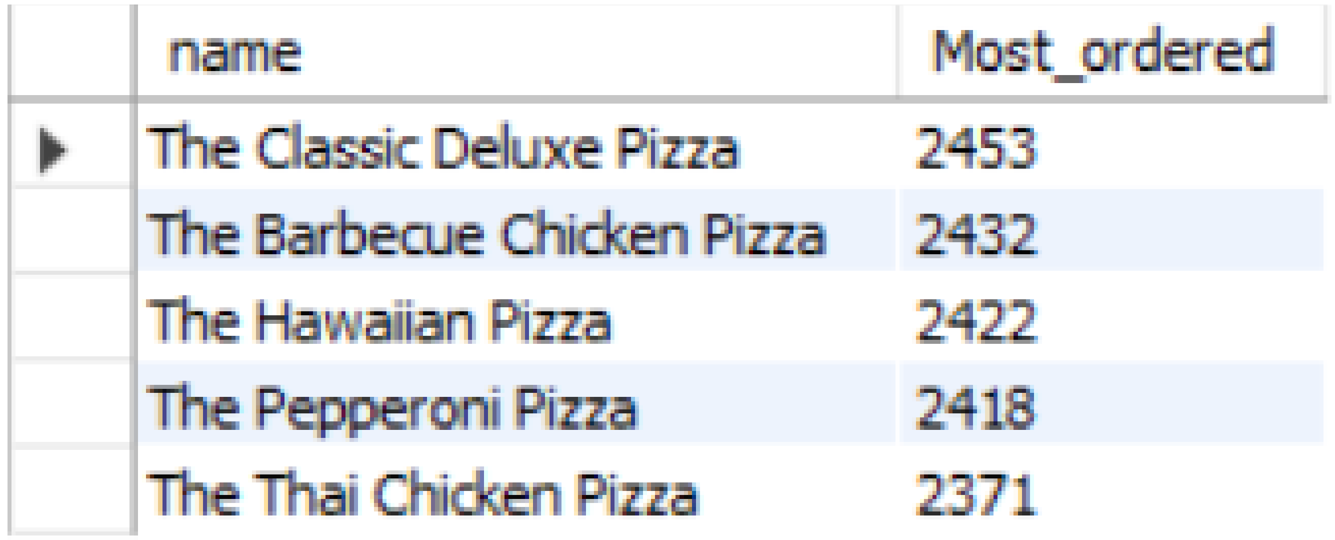


	size	most_common_size
▶	L	18526

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

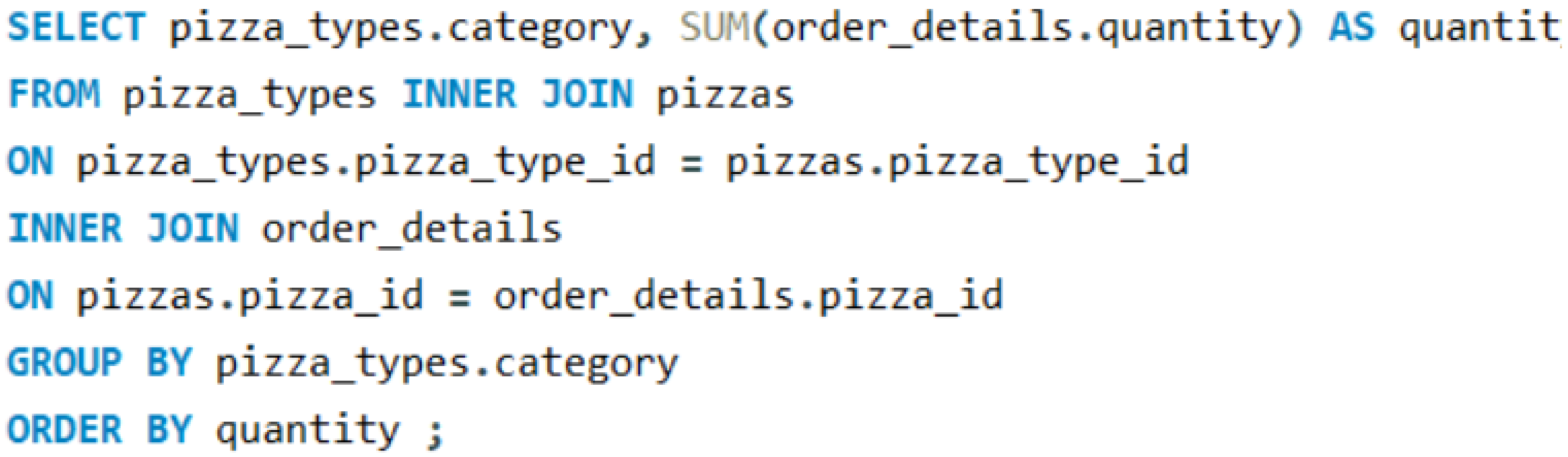


```sql
SELECT pizza_types.name, SUM(order_details.quantity) AS Most_ordered
FROM pizza_types
INNER JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
INNER JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Most_ordered DESC
LIMIT 5;
```

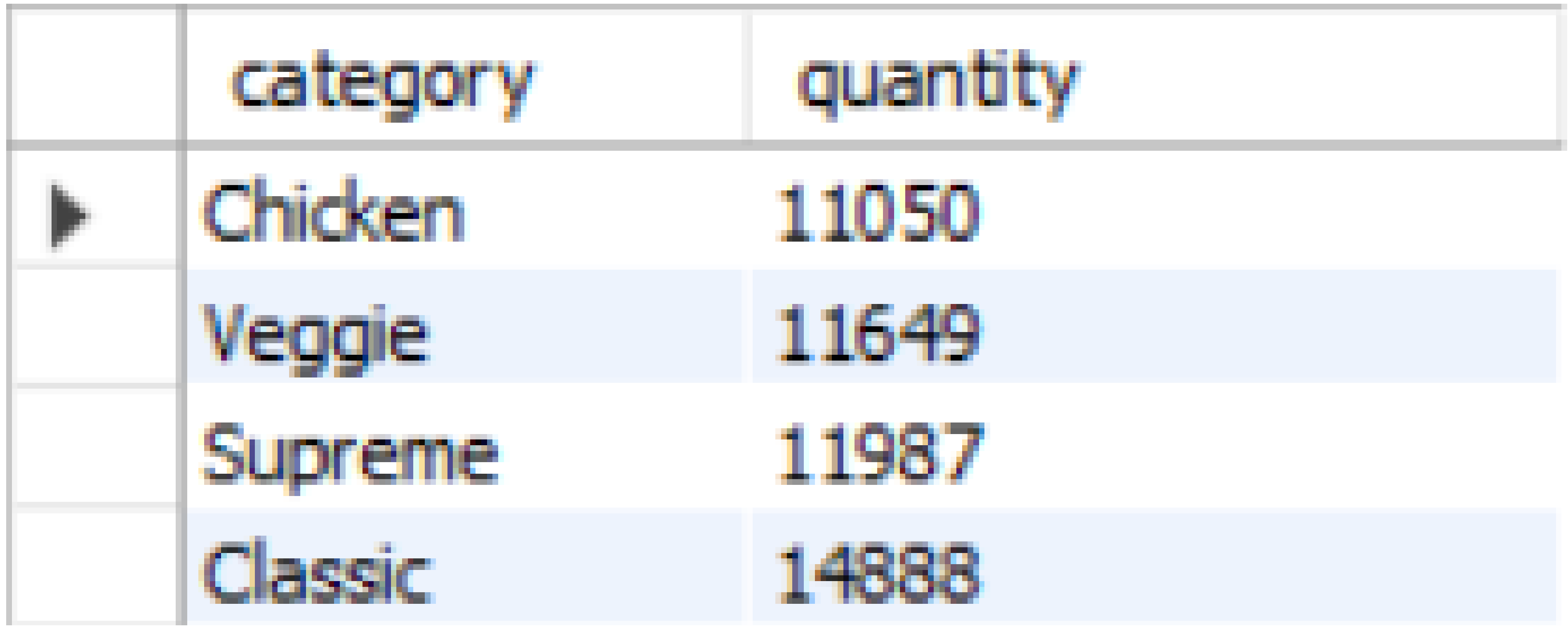


name	Most_ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

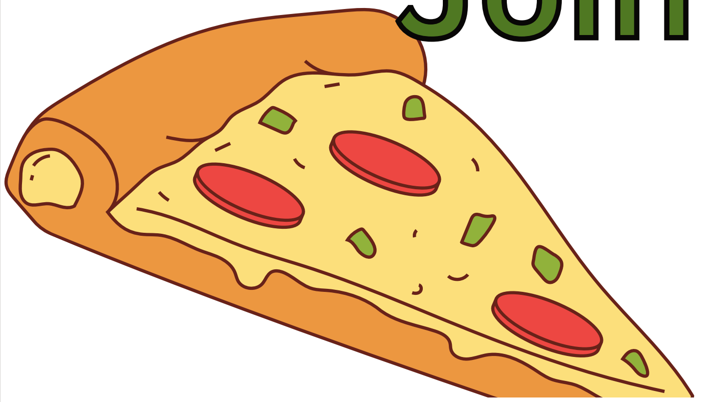


```
SELECT pizza_types.category, SUM(order_details.quantity) AS quantit
FROM pizza_types INNER JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
INNER JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity ;
```

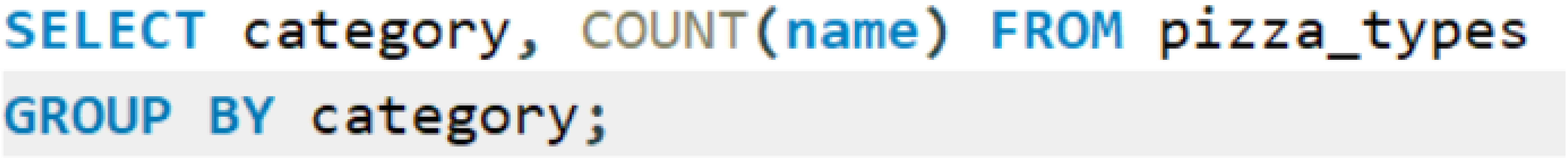


	category	quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

Divakar gulati

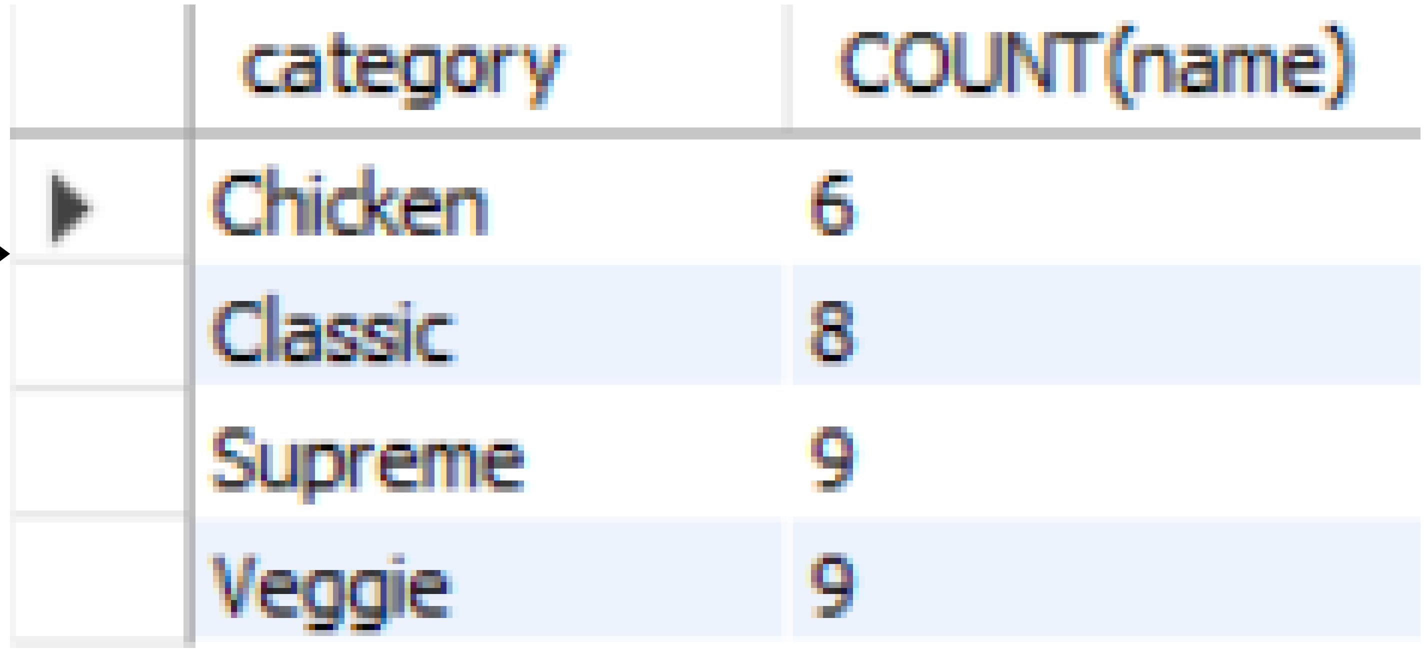


Join relevant tables to find the category-wise distribution of pizzas.



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

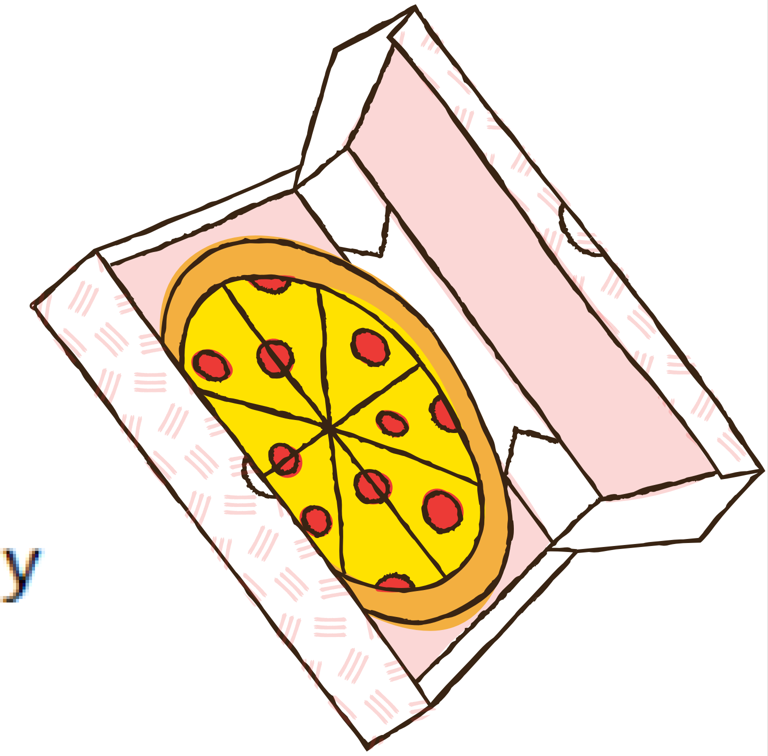
Output

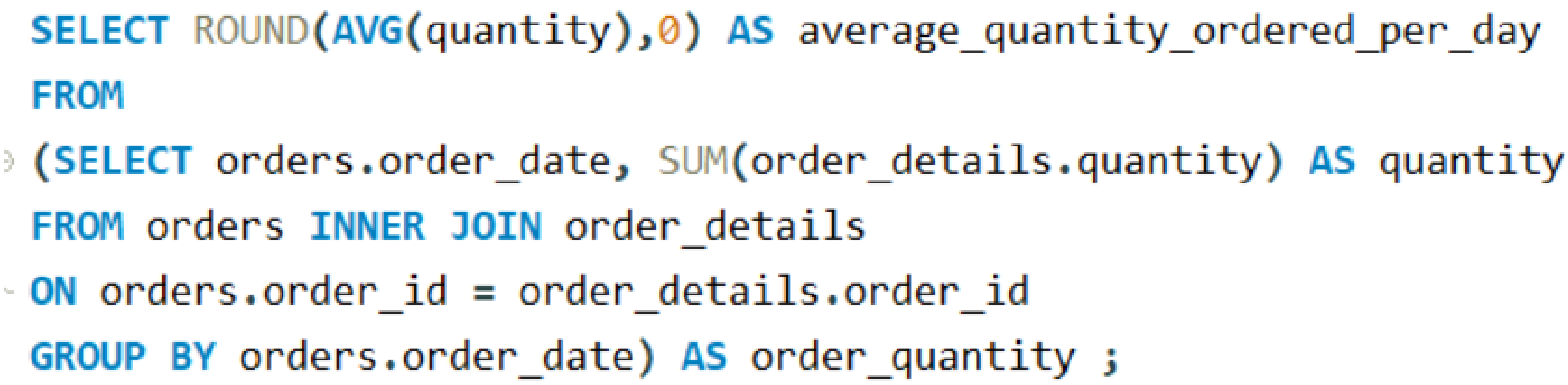


	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

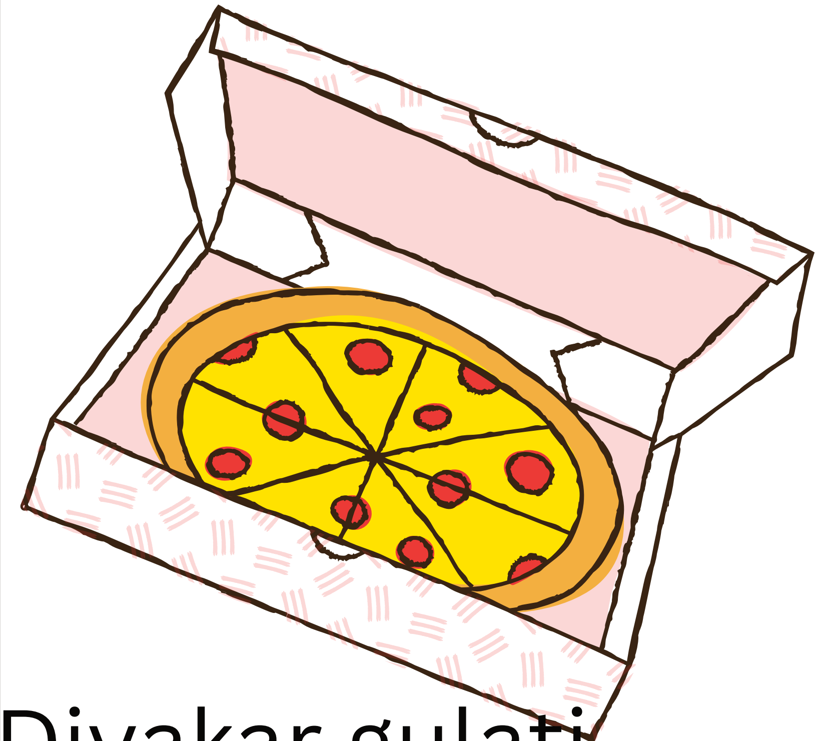
Divakar gulati

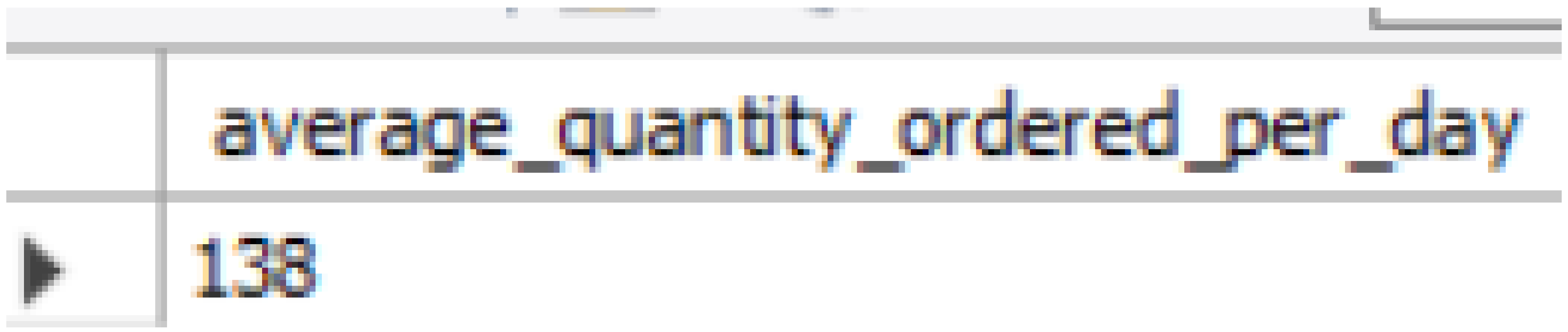
Group the orders by date and calculate the average number of pizzas ordered per day.





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

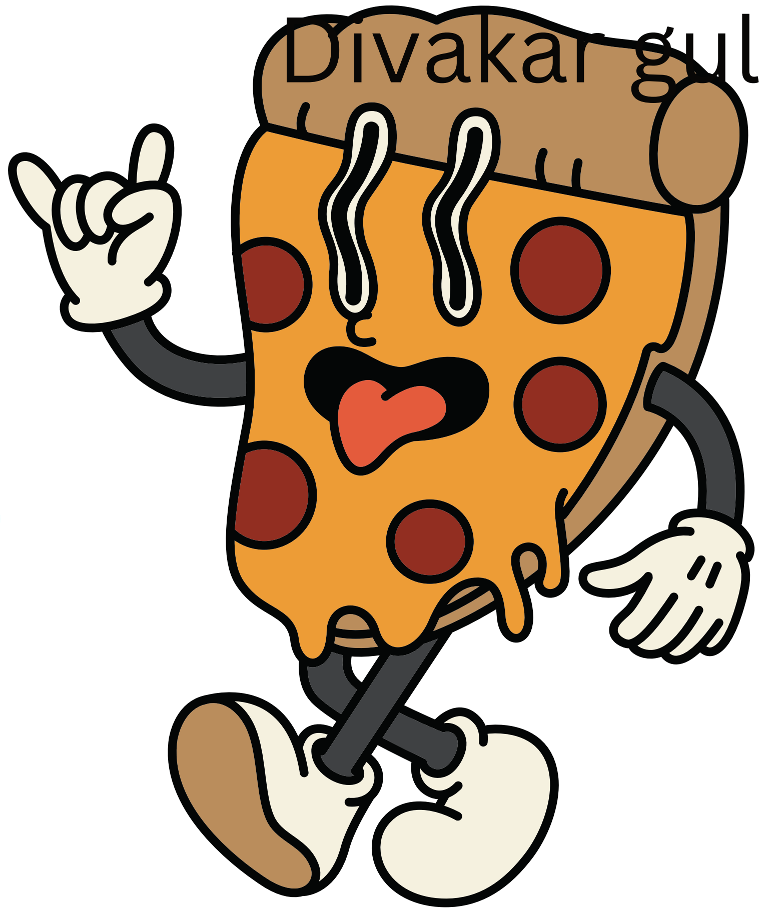




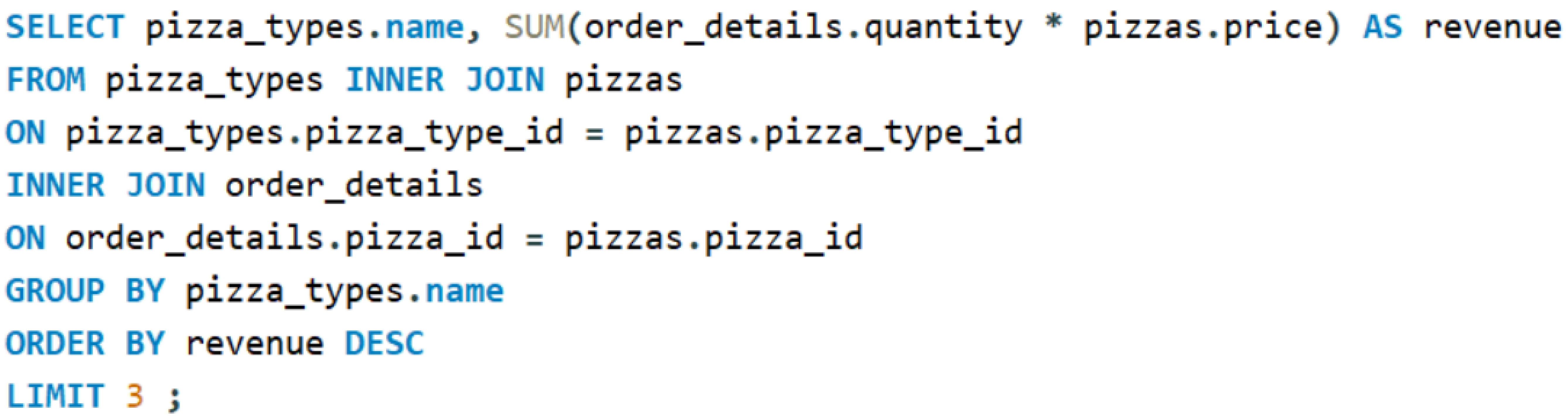
```
average_quantity_ordered_per_day
138
```

Divakar gulati

Divakar gulati

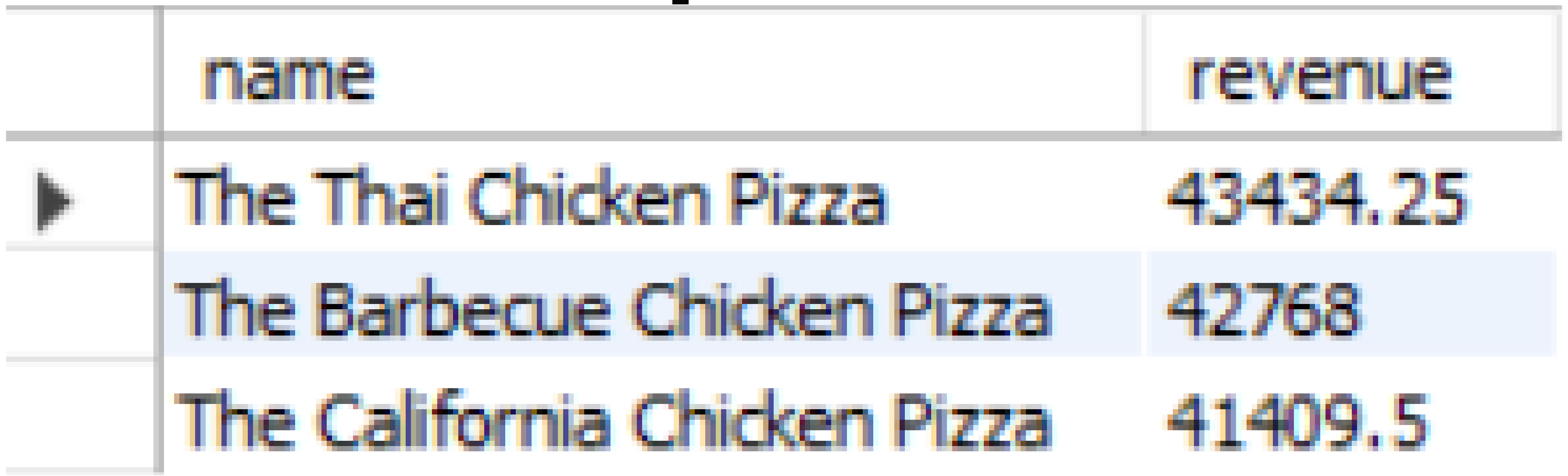


Determine the top 3 most ordered pizza types based on revenue.



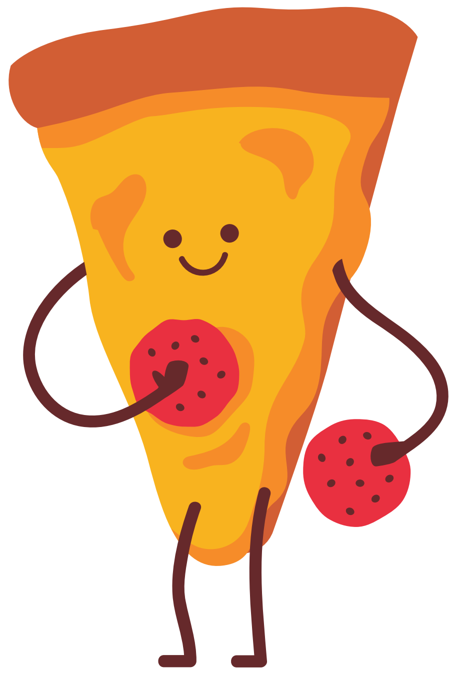
```
SELECT pizza_types.name, SUM(order_details.quantity * pizzas.price) AS revenue
FROM pizza_types INNER JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
INNER JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3 ;
```

Output

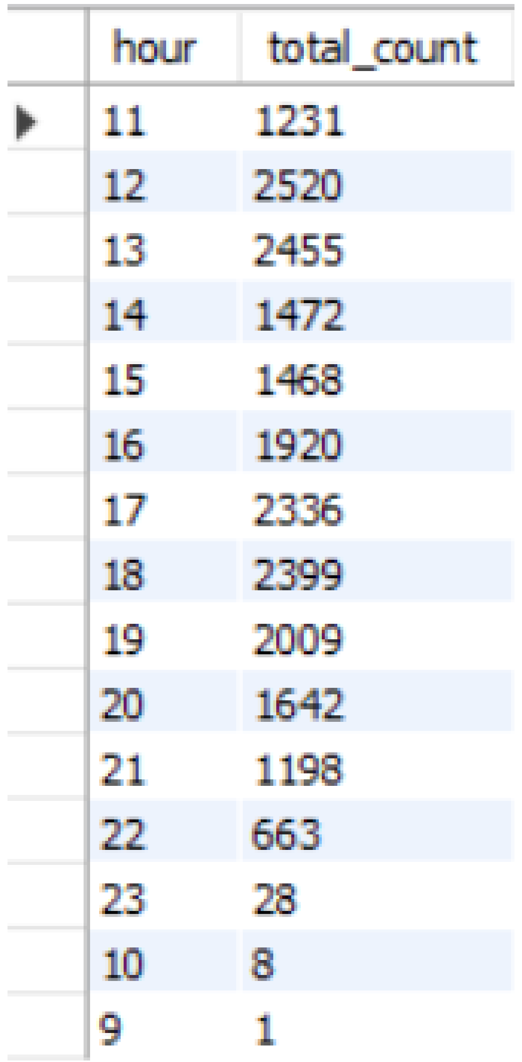


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Determine the distribution of orders by hour of the day.



```
SELECT hour(order_time) AS hour, count(order_id) AS total_count
FROM orders
GROUP BY hour(order_time);
```

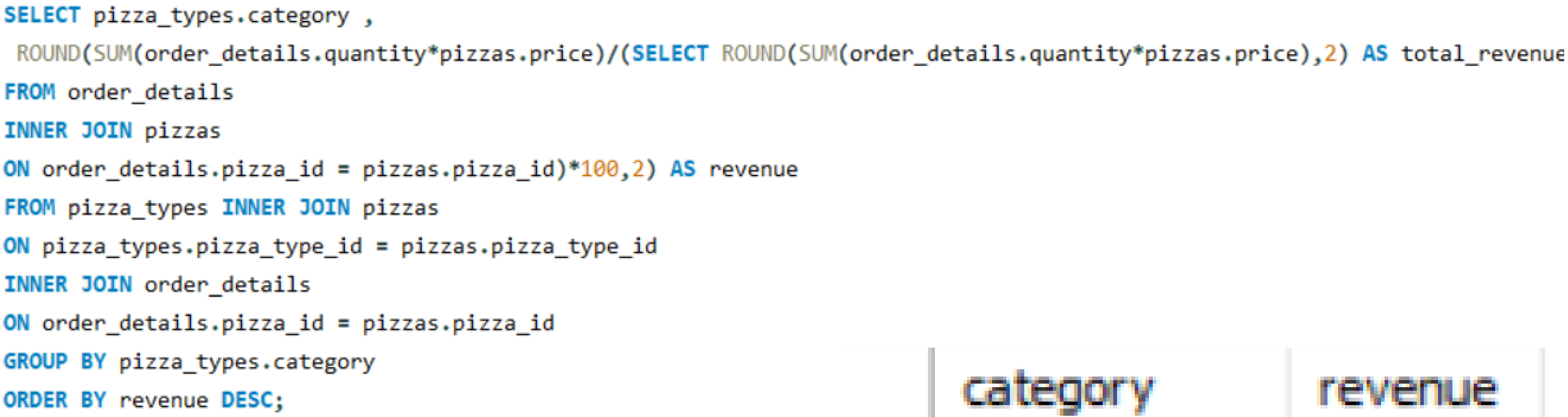


hour	total_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

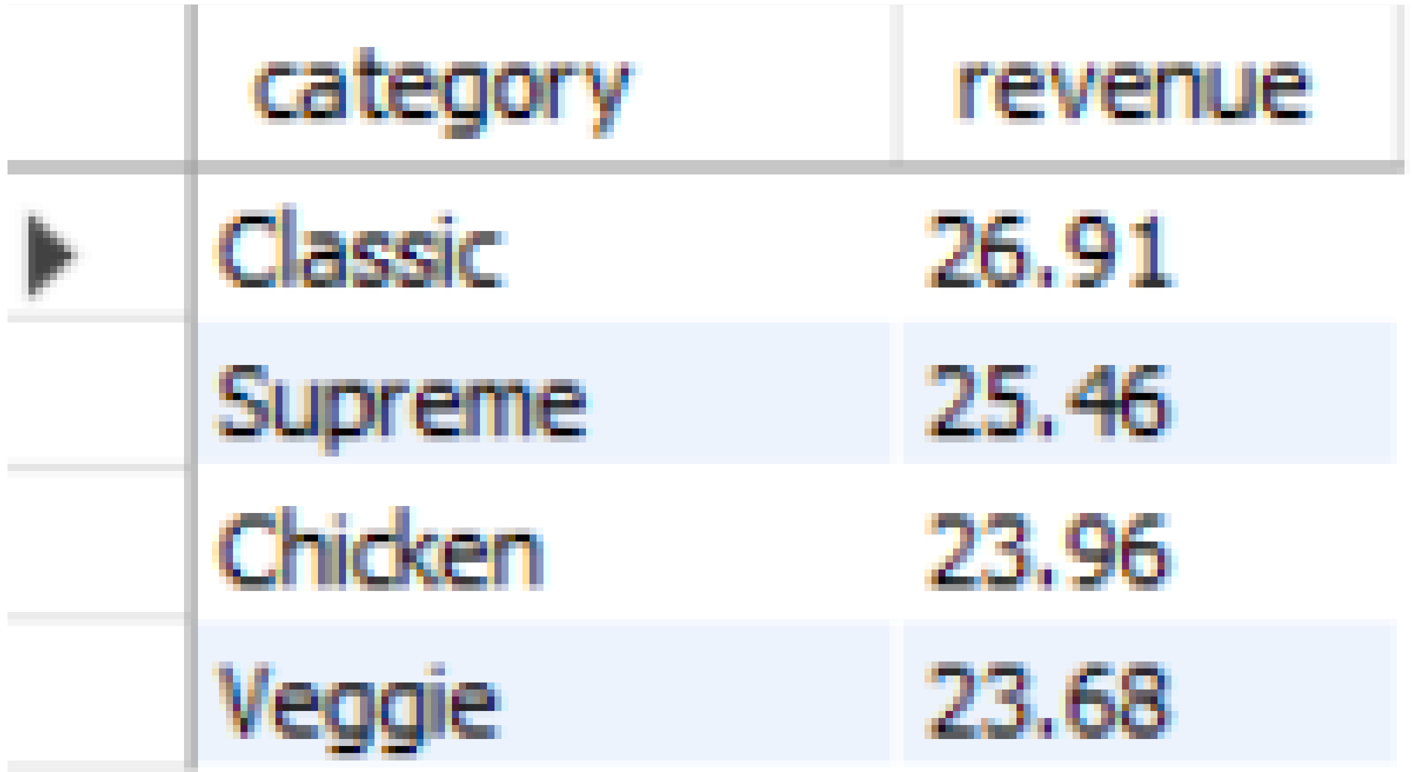
Divakar gulati

Divakar gulati

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



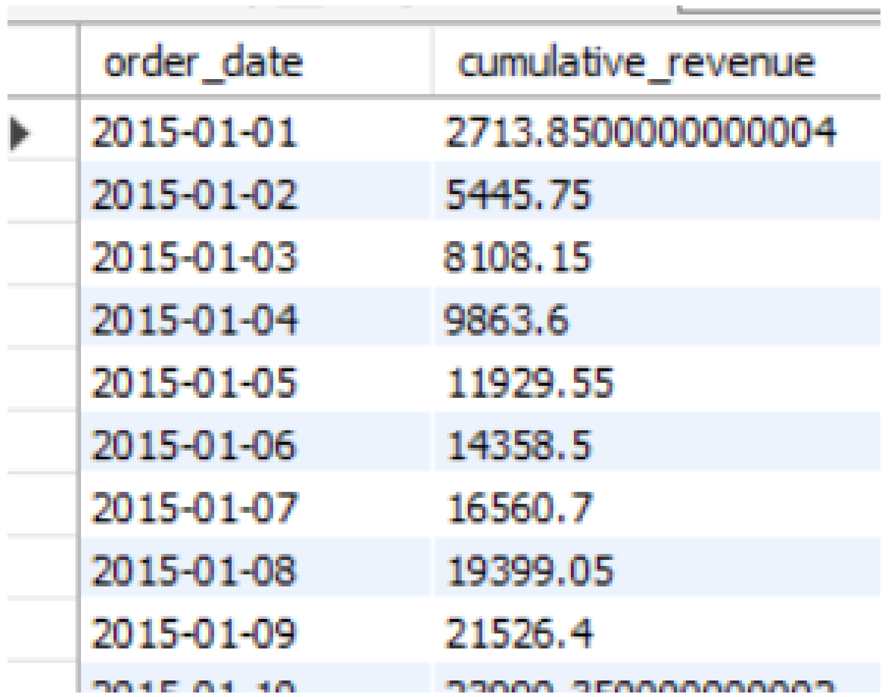
```
SELECT pizza_types.category ,
 ROUND(SUM(order_details.quantity*pizzas.price)/(SELECT ROUND(SUM(order_details.quantity*pizzas.price),2) AS total_revenue
FROM order_details
INNER JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id)*100,2) AS revenue
FROM pizza_types INNER JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
INNER 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

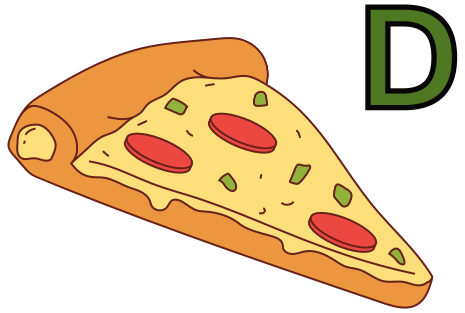
Analyze the cumulative revenue generated over time.

```
SELECT order_date,
sum(revenue) OVER(ORDER BY order_date) AS cumulative_revenue
FROM
(SELECT orders.order_date,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM order_details INNER JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
INNER JOIN orders
ON order_details.order_id = orders.order_id
GROUP BY orders.order_date) AS sales;
```

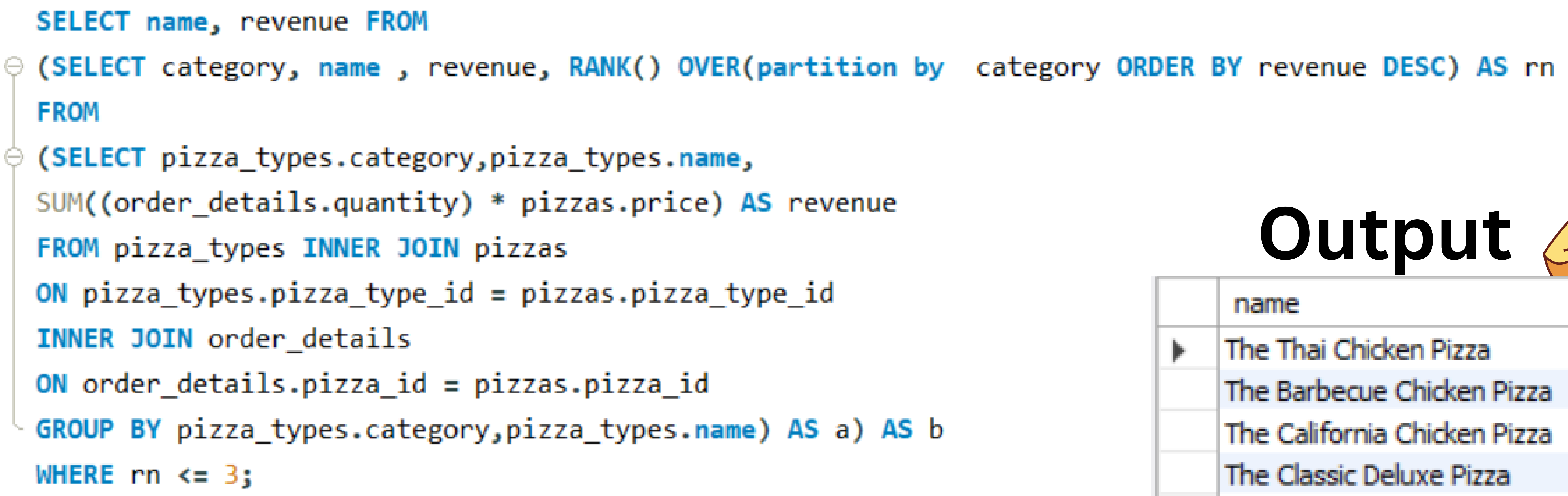


order_date	cumulative_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000003

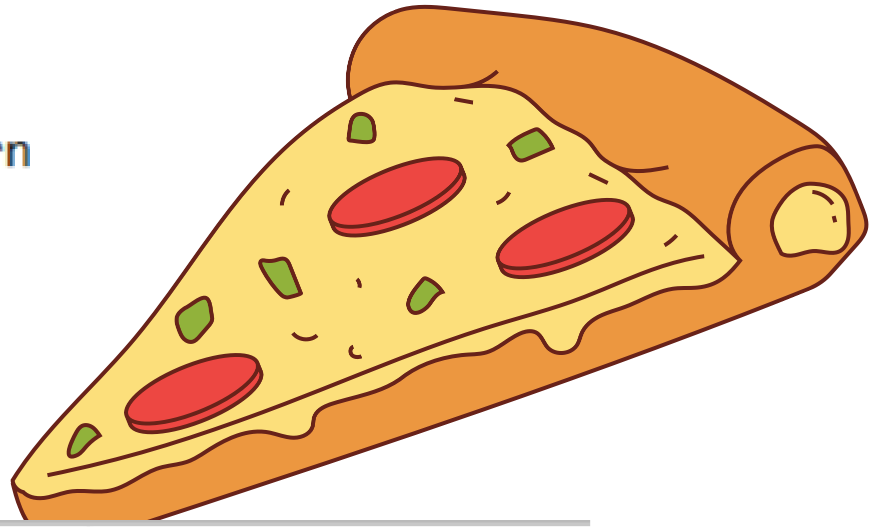
Divakar gulati



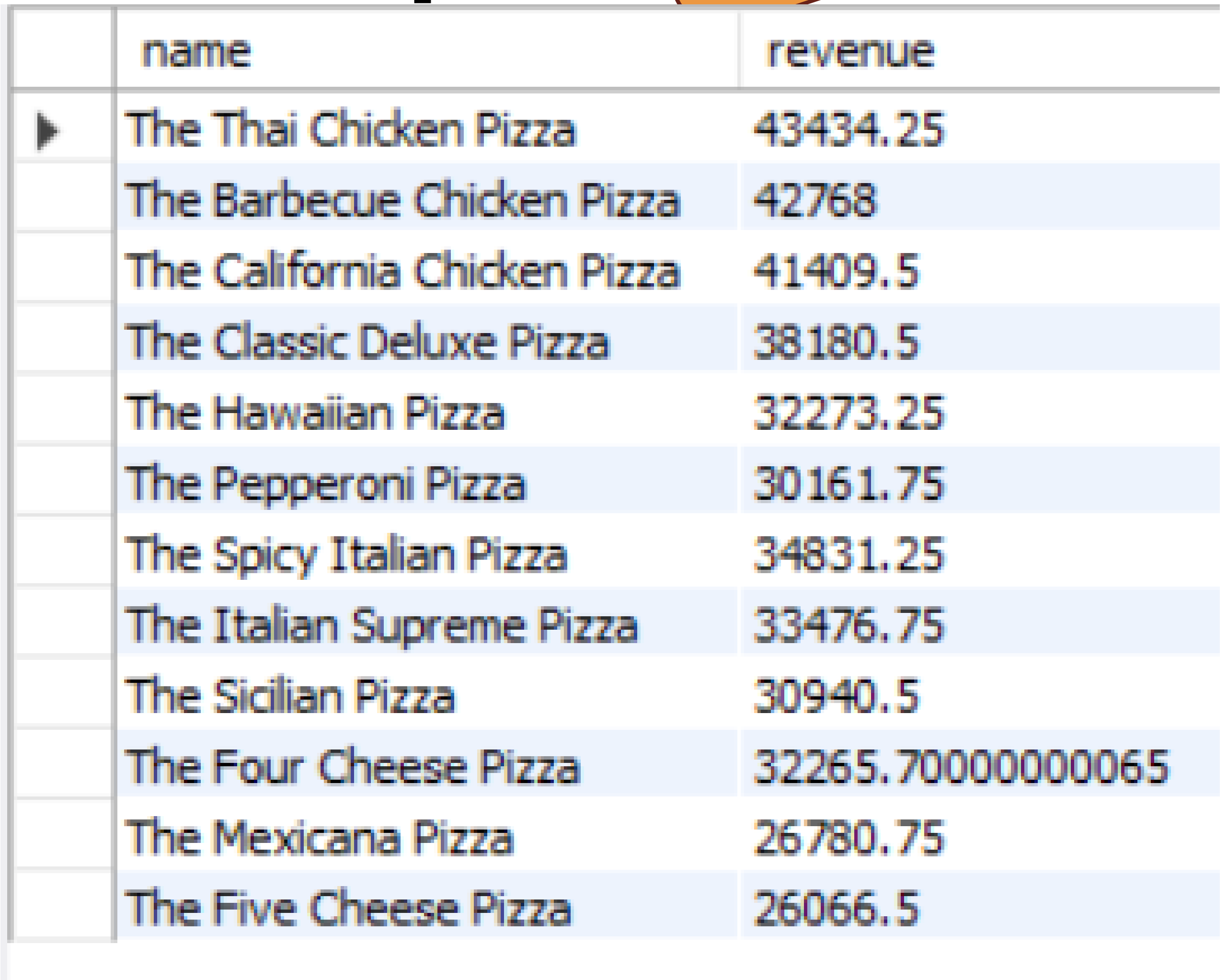
Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```
SELECT name, revenue FROM
(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 INNER JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
INNER JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category,pizza_types.name) AS a) AS b
WHERE rn <= 3;
```



Output



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Divakar gulati

Thanks For watching