SQL PROJECT

PIZZA SALES SQL PROJECT

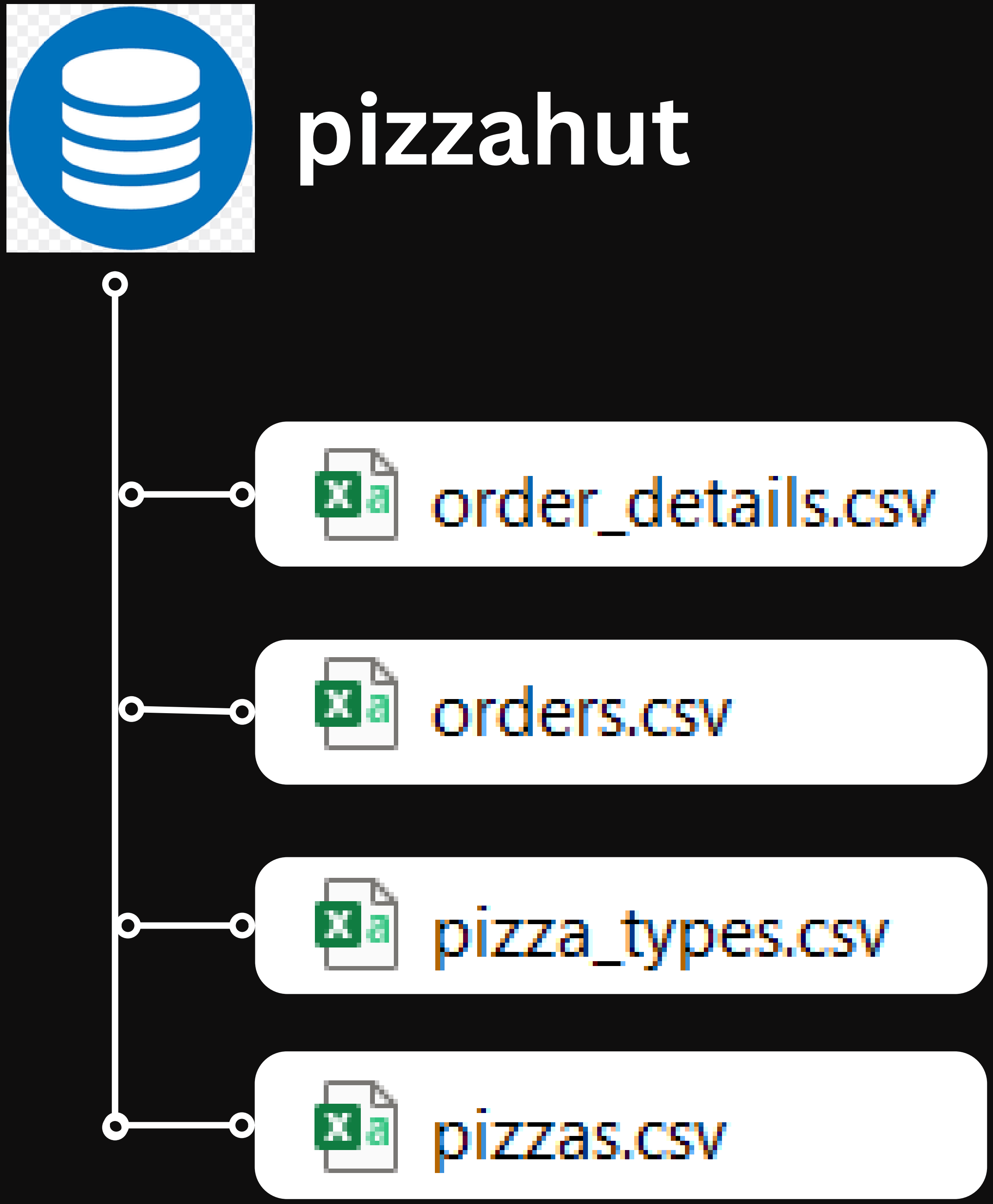
ANALYZING AND ANSWERING REAL-WORLD SQL QUESTIONS

JAYANTA NATH

IIT Guwahati

nathjayanta772@gmail.com

PIZZA SALES SQL PROJECT



THE COLUMNS :

order_details_id / order_id / pizza_id / quantity

order_id / date / time

pizza_type_id / name / category / ingredients

pizza_id / pizza_type_id / size / price

PIZZA SALES SQL PROJECT

>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

BASIC:

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:

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

ADVANCED:

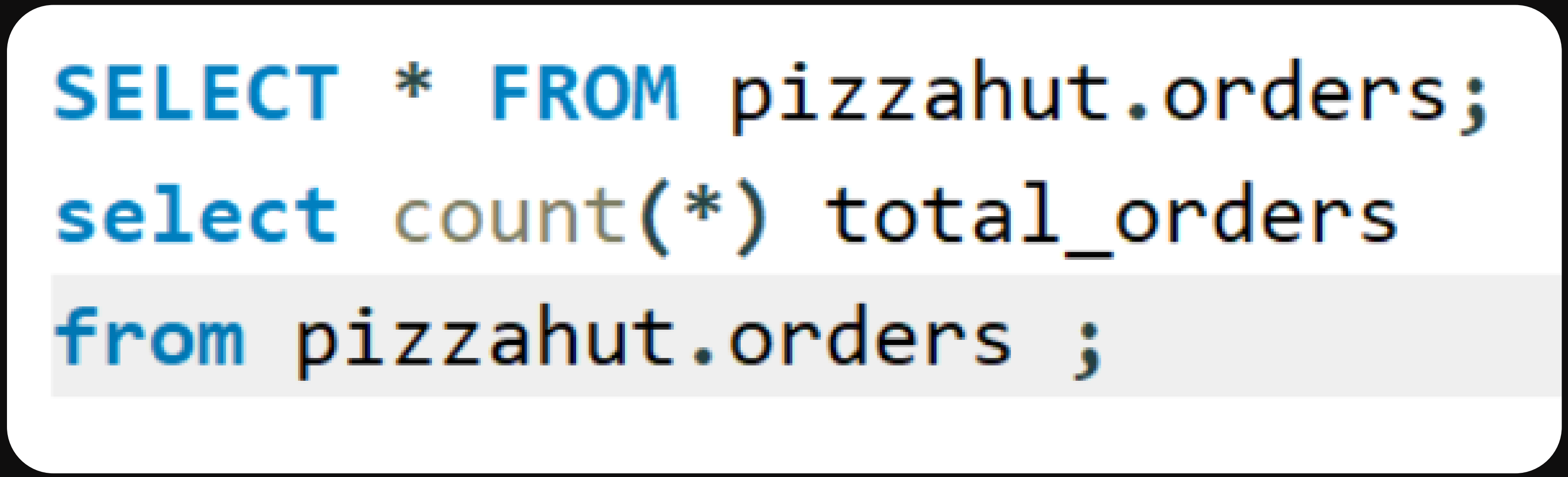
1.Calculate the percentage contribution of each pizza type to total revenue.
2.Analyze the cumulative revenue generated over time.
3.Determine the top 3 most ordered pizza types based on revenue for each pizza category.

PIZZA SALES SQL PROJECT

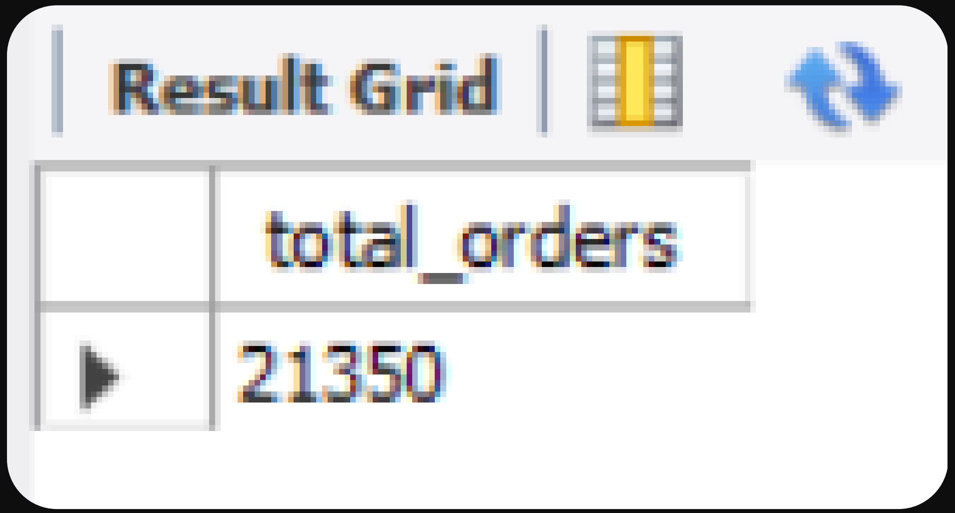
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

BASIC:

1.Retrieve the total number of orders placed.



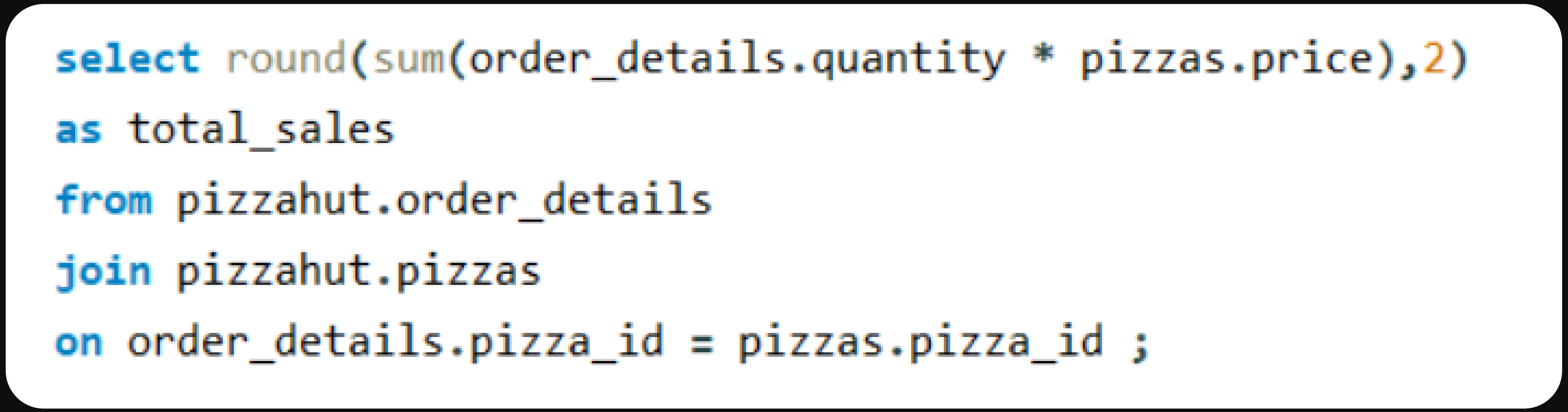
```sql
SELECT * FROM pizzahut.orders;
select count(*) total_orders
from pizzahut.orders ;
```



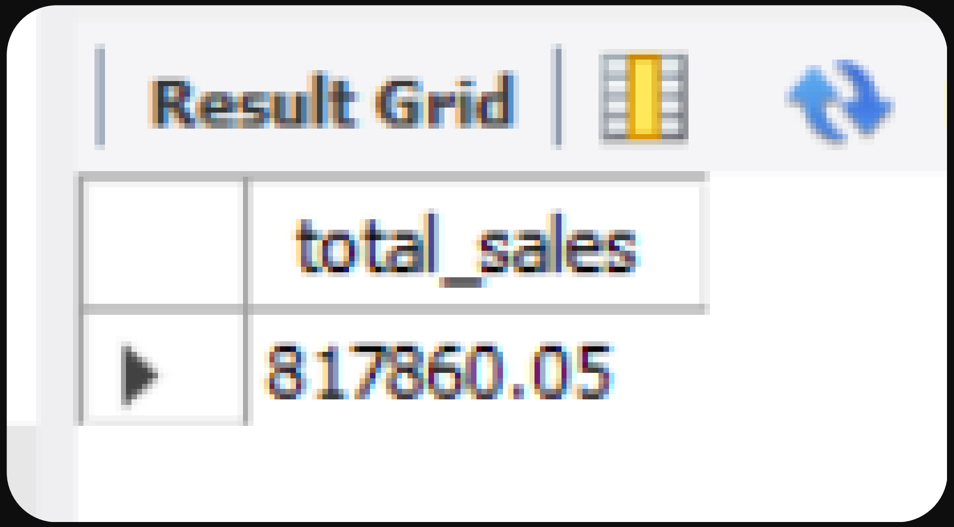
```
total_orders
21350
```

BASIC:

2.Calculate the total revenue generated from pizza sales.



```sql
select round(sum(order_details.quantity * pizzas.price),2)
as total_sales
from pizzahut.order_details
join pizzahut.pizzas
on order_details.pizza_id = pizzas.pizza_id ;
```



```
total_sales
817860.05
```

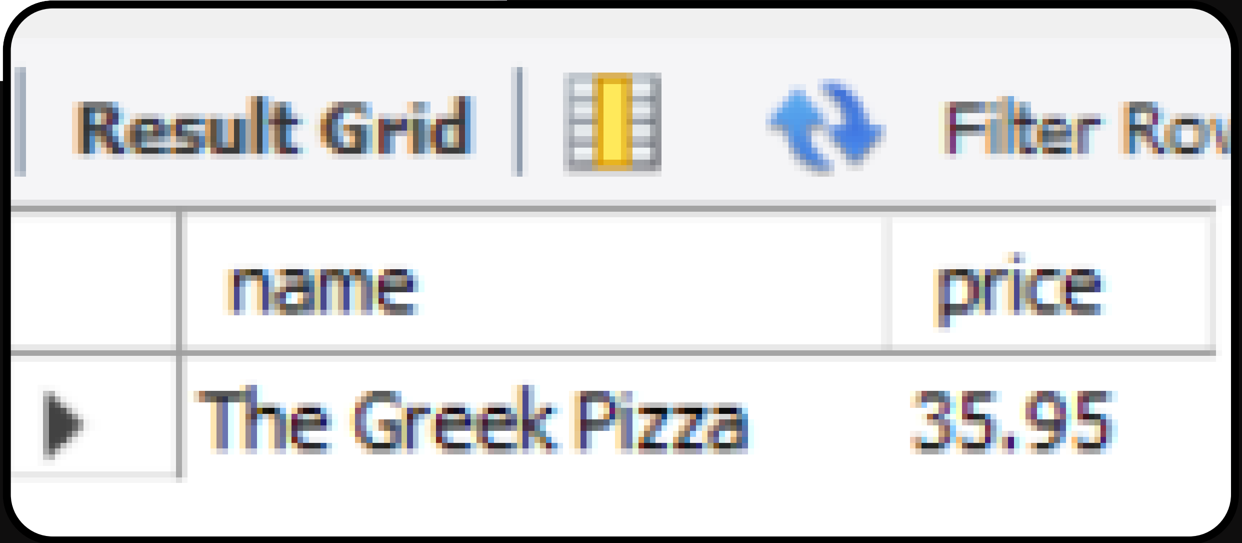
PIZZA SALES SQL PROJECT

>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

BASIC:

3.Identify the highest-priced pizza.

```
SELECT * FROM pizzahut.pizzas;
select pizza_types.name, pizzas.price
from pizzahut.pizzas
join pizzahut.pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc limit 1;
```



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

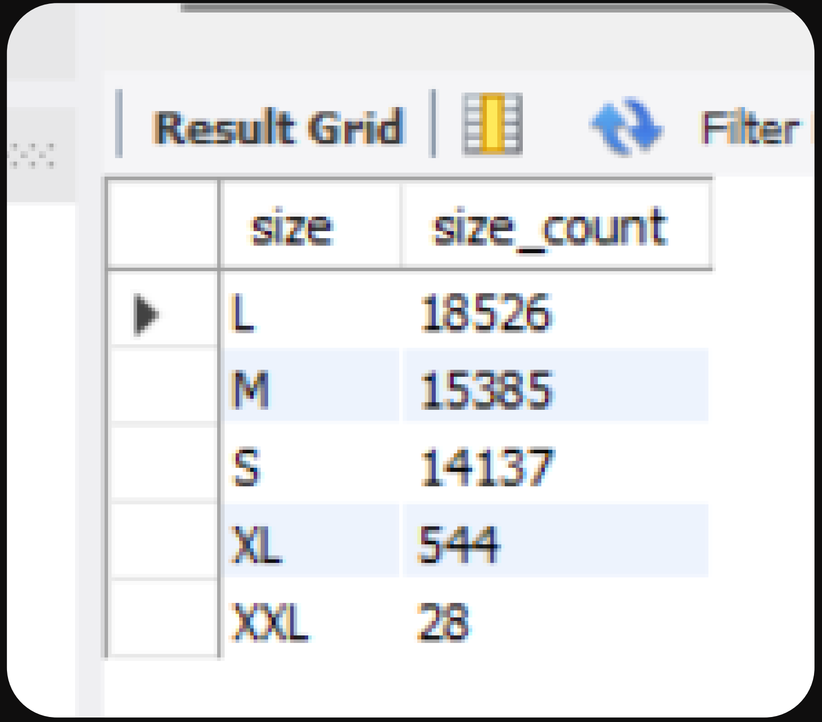
PIZZA SALES SQL PROJECT

>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

BASIC:

4.Identify the most common pizza size ordered.

```
select pizzas.size, count(order_details.order_details_id)
as size_count
from pizzahut.pizzas
join pizzahut.order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by size_count desc;
```


Result Grid

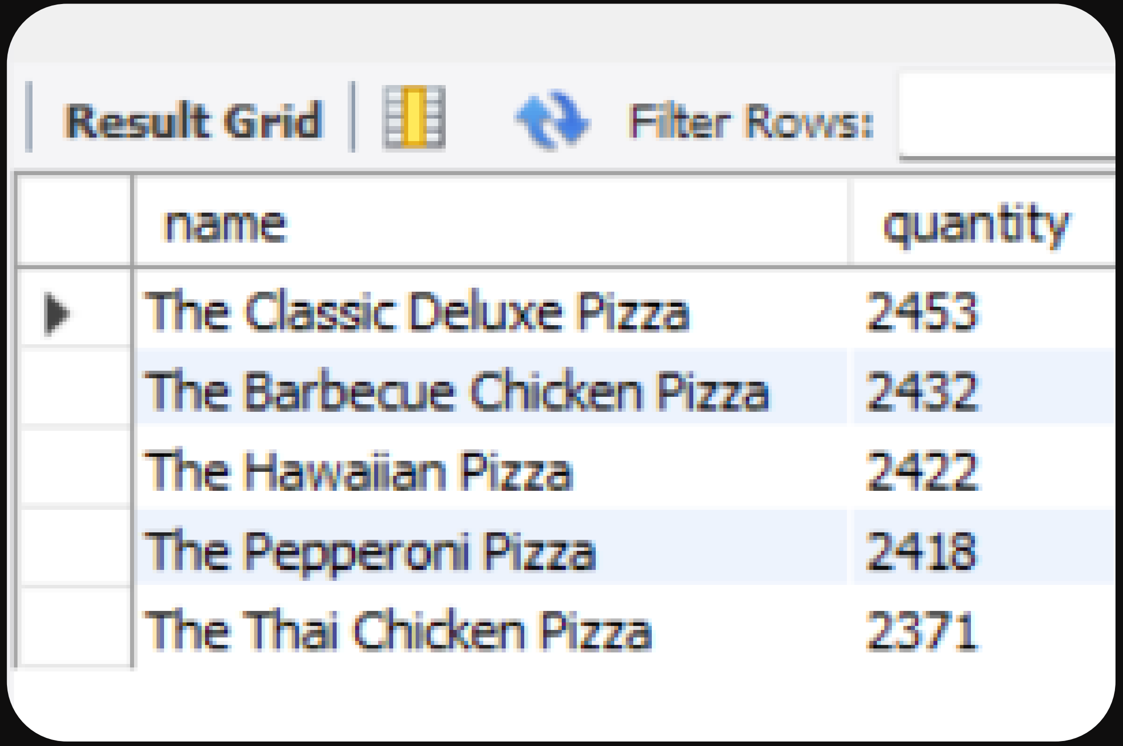
size	size_count
L	18526
M	15385
S	14137
XL	544
XXL	28

>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

BASIC:

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

```
select pizza_types.name, sum(order_details.quantity)
as quantity
from pizzahut.pizza_types
join pizzahut.pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizzahut.order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.name
order by quantity desc
limit 5;
```



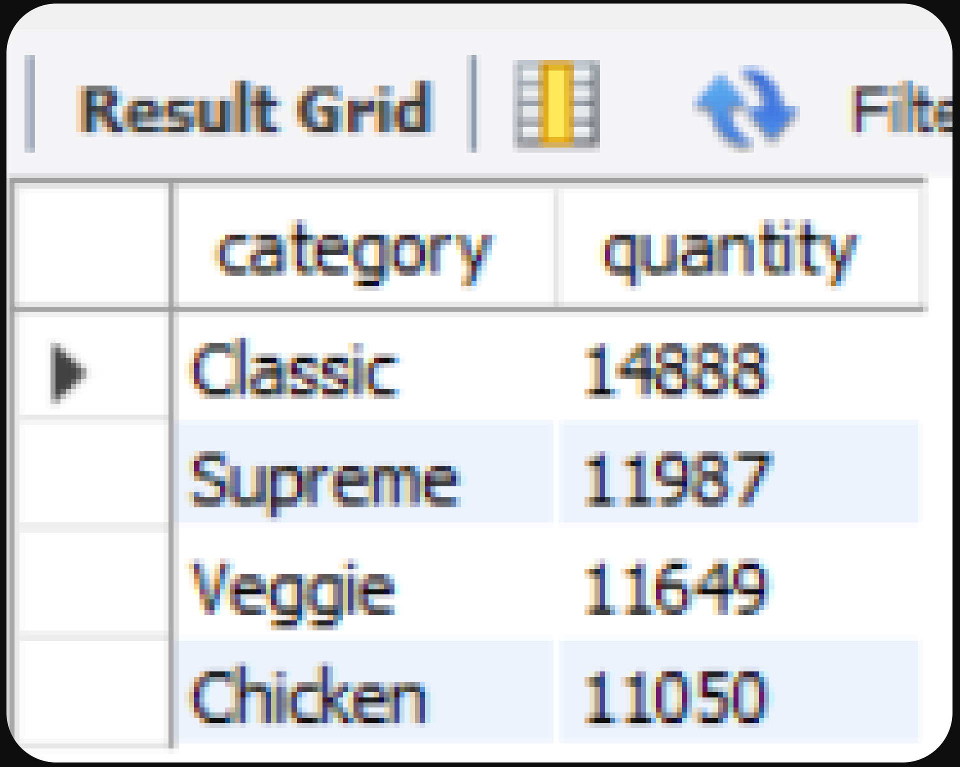
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

INTERMEDIATE:

1.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 pizzahut.pizza_types join pizzahut.pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizzahut.order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by quantity desc ;
```



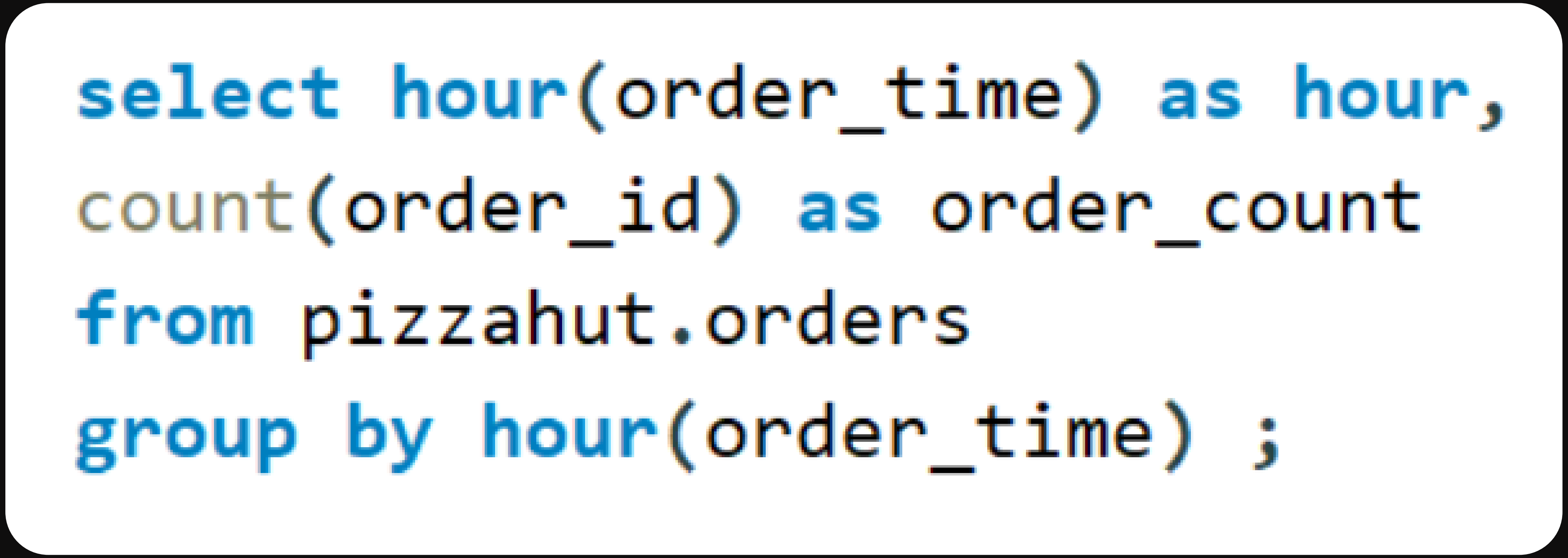
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

PIZZA SALES SQL PROJECT

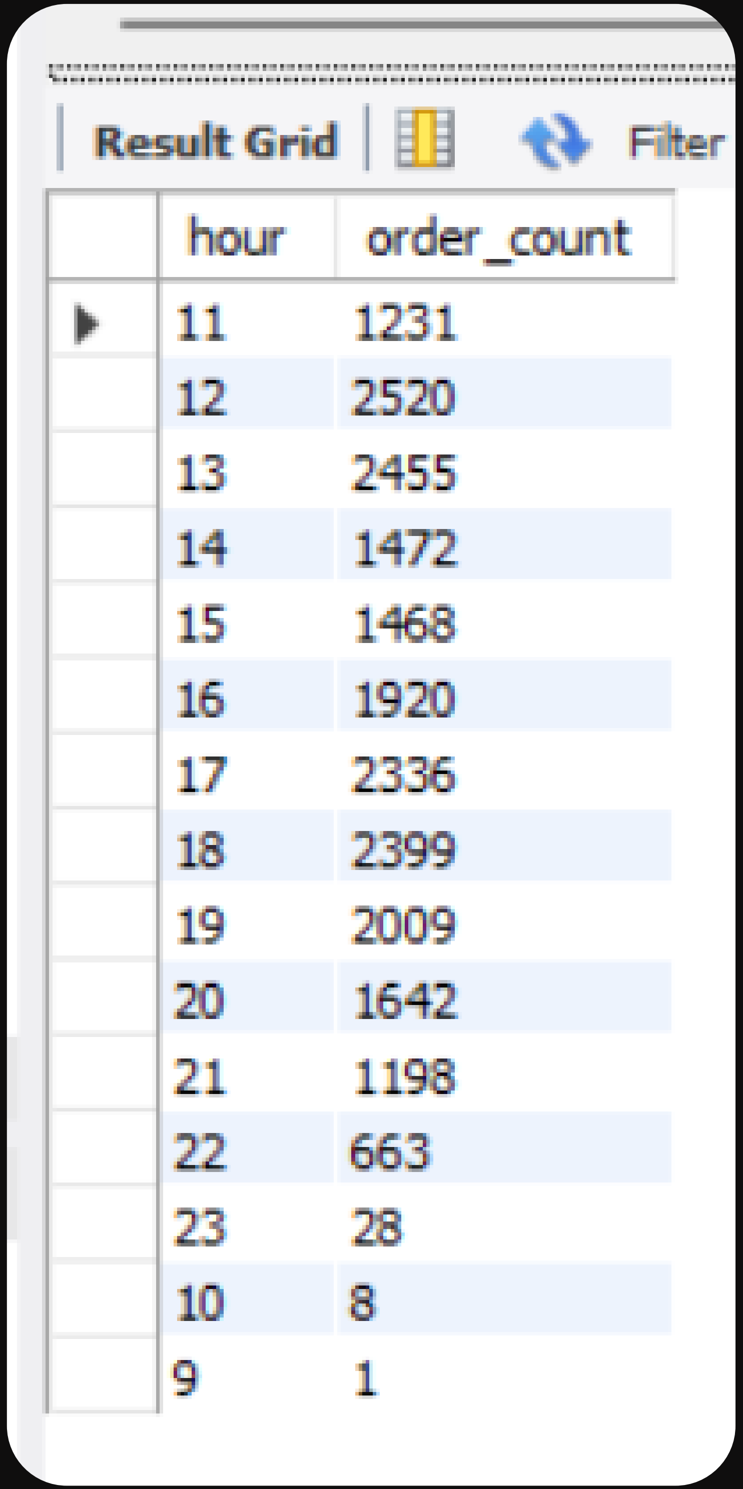
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

INTERMEDIATE:

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



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

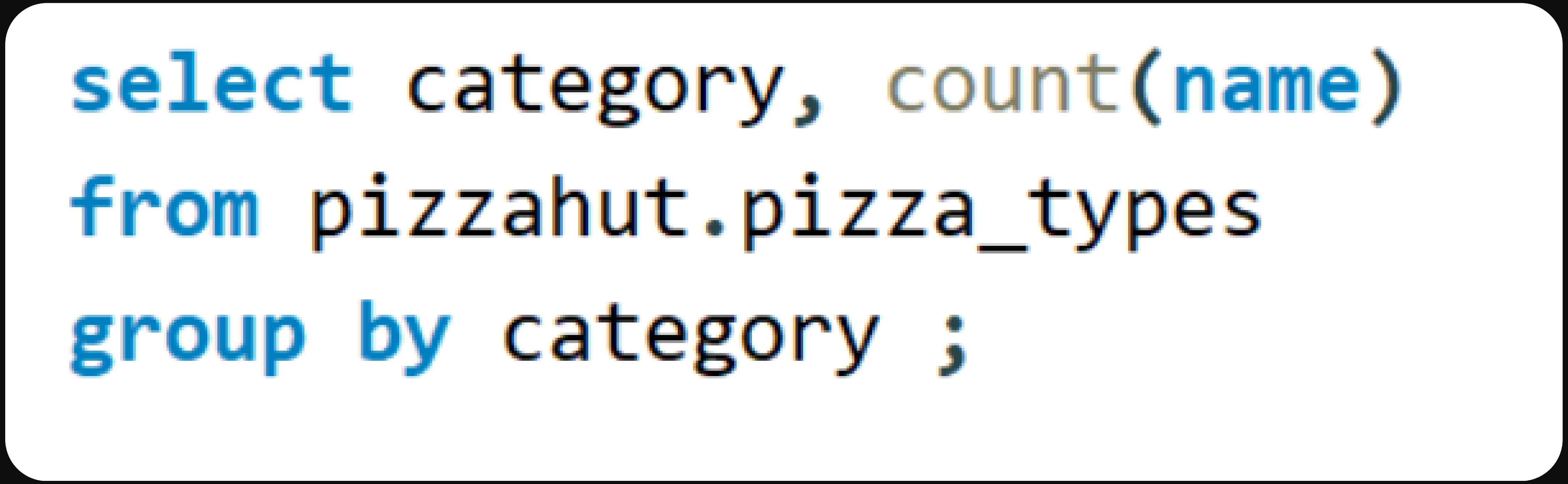


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

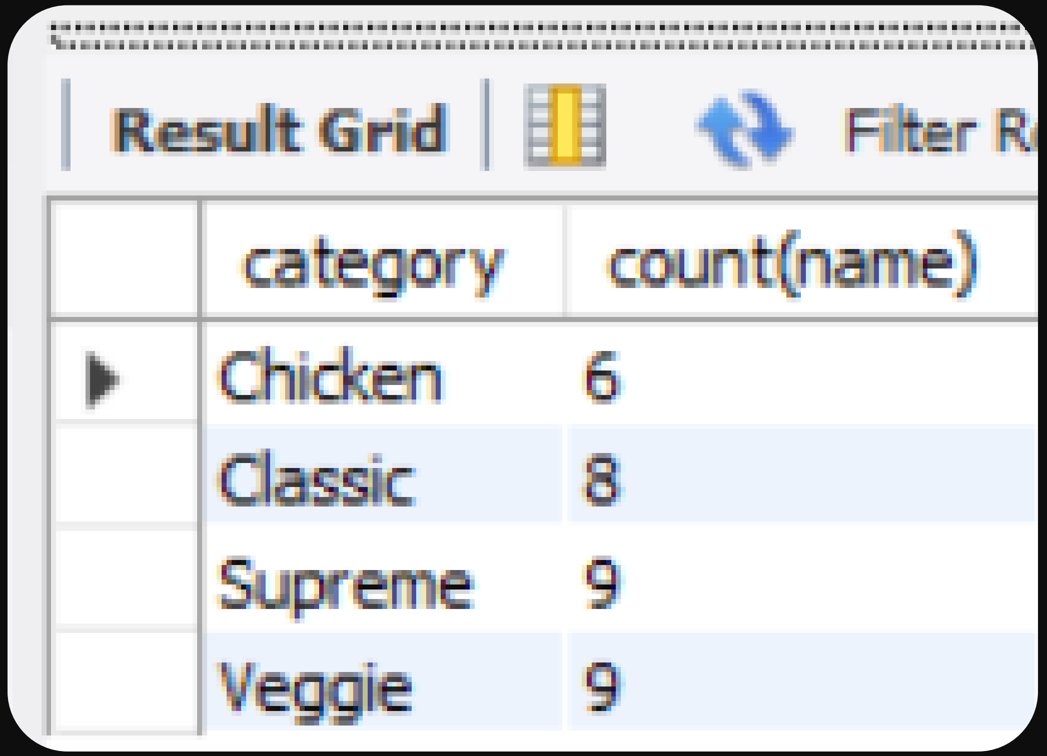
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

INTERMEDIATE:

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



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

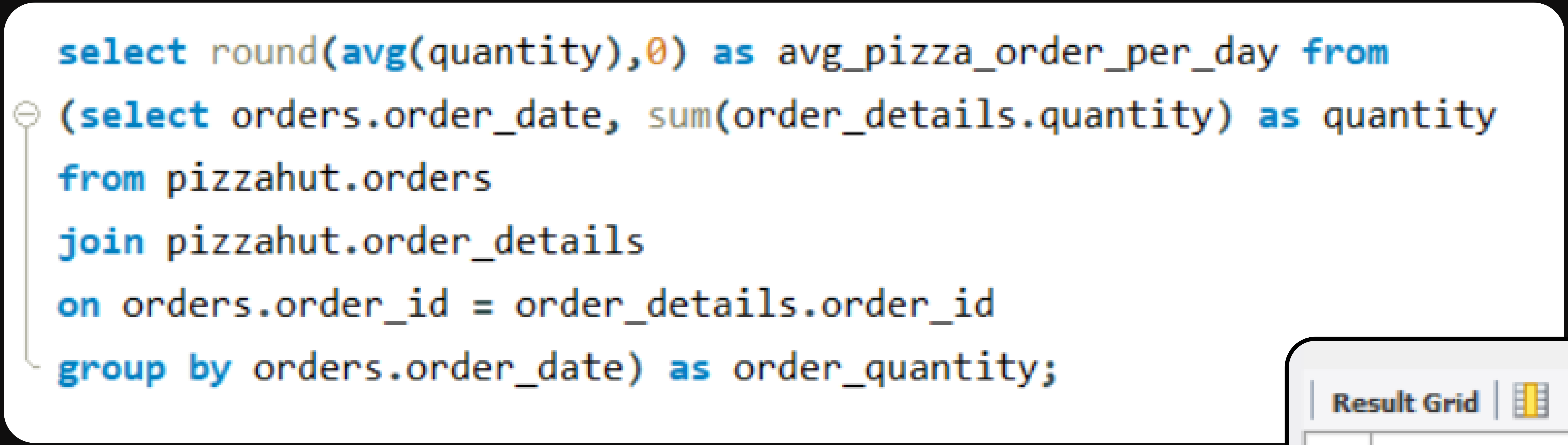


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

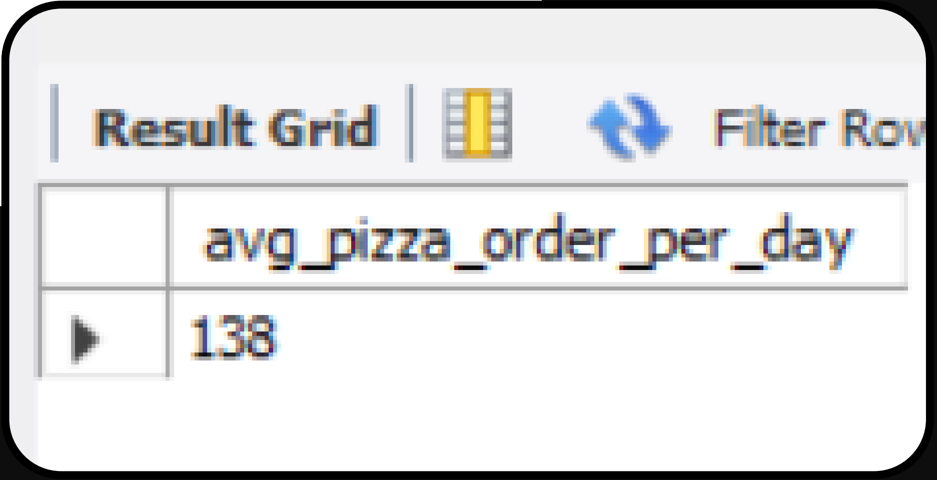
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

INTERMEDIATE:

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



```
select round(avg(quantity),0) as avg_pizza_order_per_day from
(select orders.order_date, sum(order_details.quantity) as quantity
from pizzahut.orders
join pizzahut.order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```



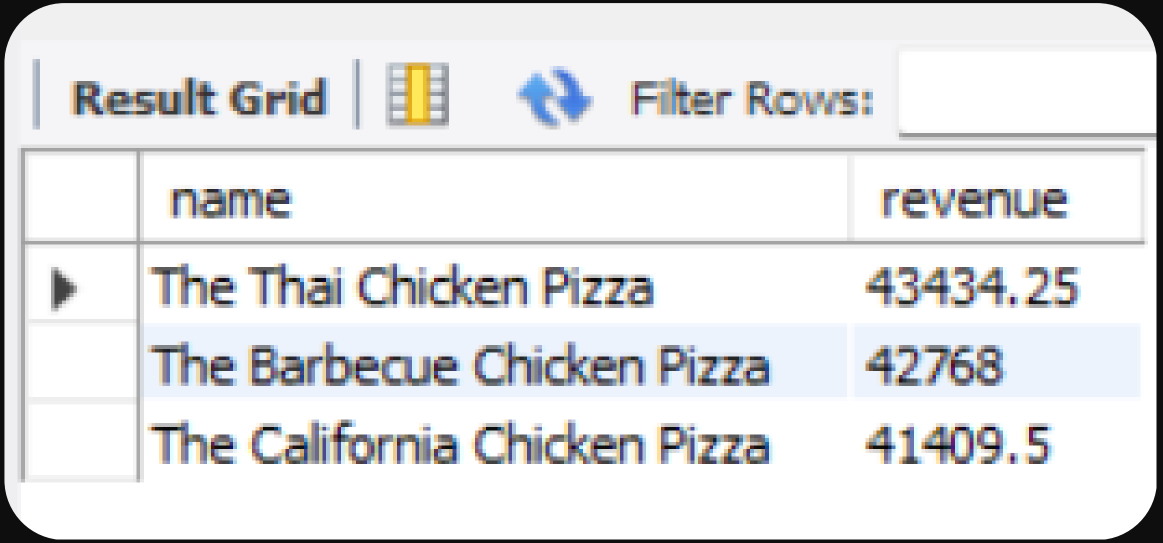
```
avg_pizza_order_per_day
138
```

>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

INTERMEDIATE:

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

```
select pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue
from pizzahut.pizza_types join pizzahut.pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizzahut.order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by revenue desc
limit 3 ;
```



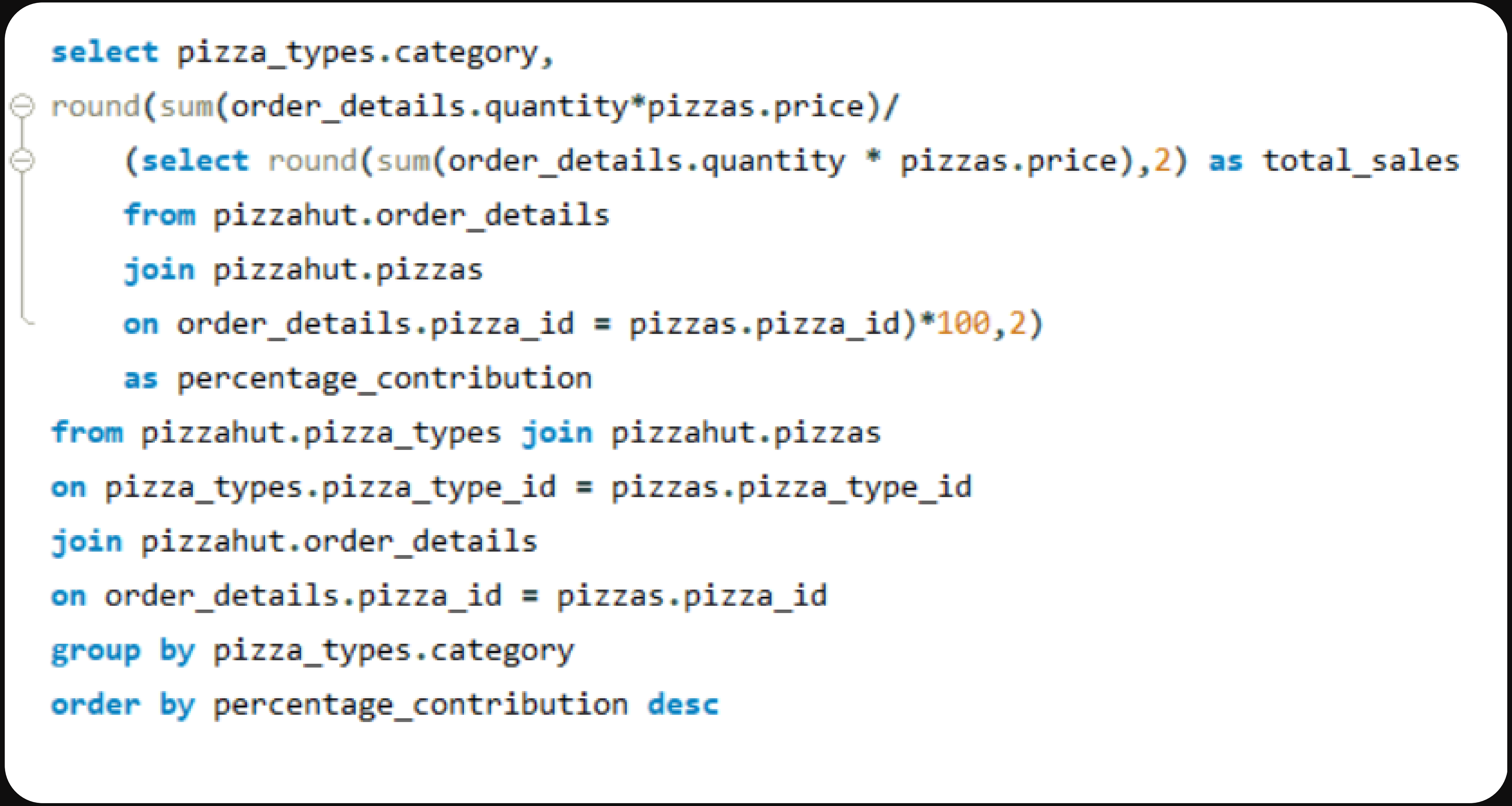
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

PIZZA SALES SQL PROJECT

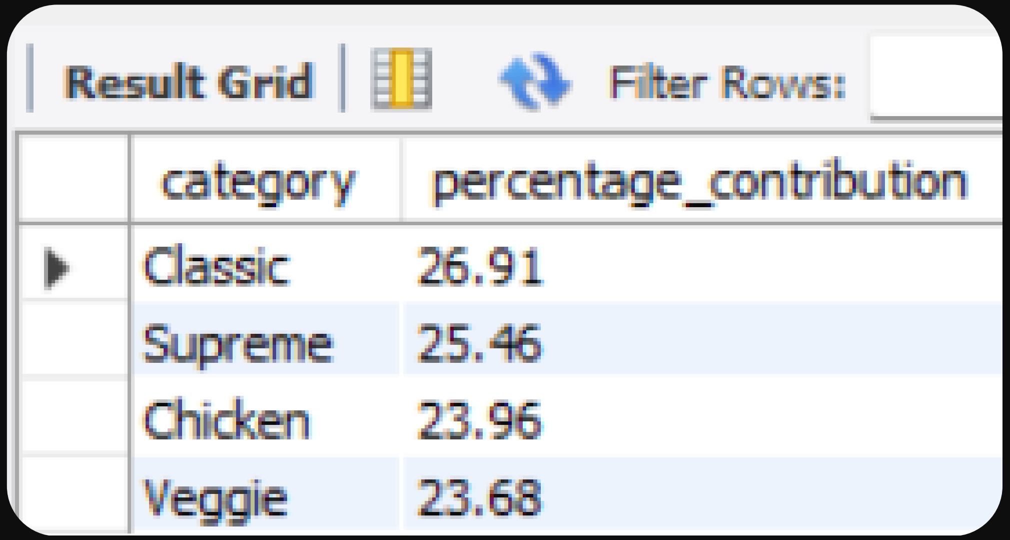
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

ADVANCED:

1.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 pizzahut.order_details
    join pizzahut.pizzas
    on order_details.pizza_id = pizzas.pizza_id)*100,2)
    as percentage_contribution
from pizzahut.pizza_types join pizzahut.pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizzahut.order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by percentage_contribution desc
```



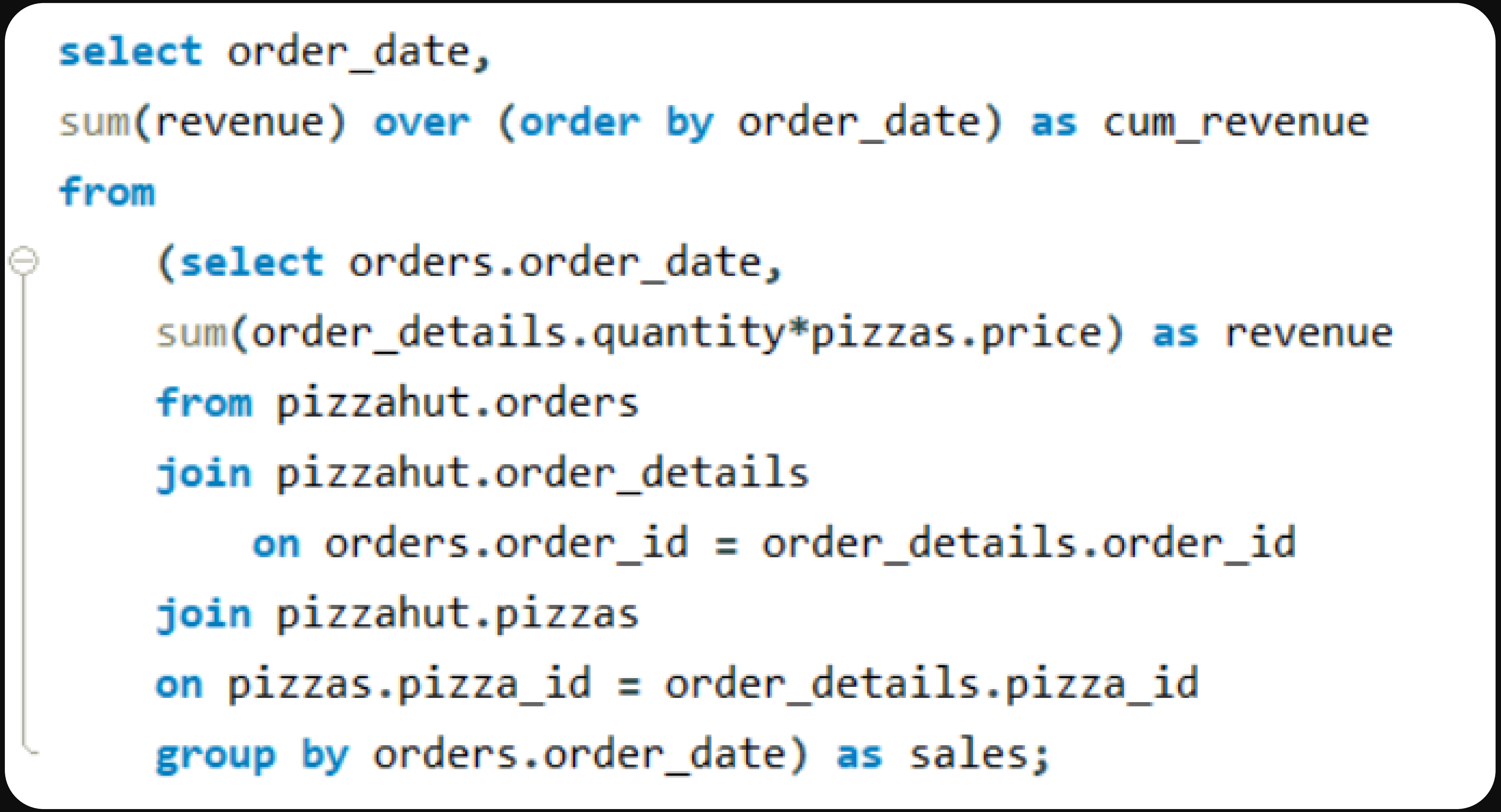
category	percentage_contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

PIZZA SALES SQL PROJECT

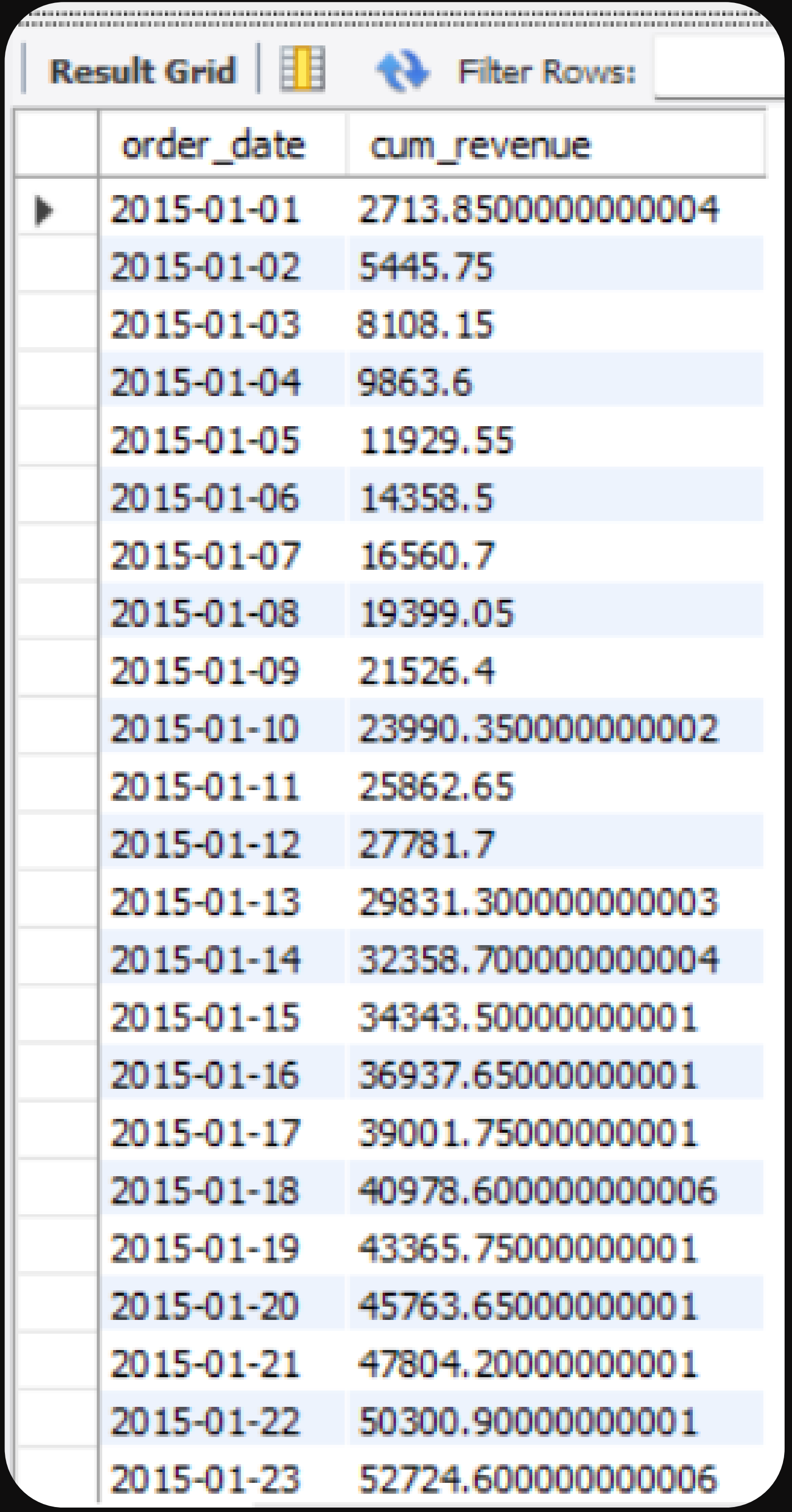
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

ADVANCED:

2.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 pizzahut.orders
    join pizzahut.order_details
        on orders.order_id = order_details.order_id
    join pizzahut.pizzas
    on pizzas.pizza_id = order_details.pizza_id
    group by orders.order_date) as sales;
```



Result Grid | Filter Rows:

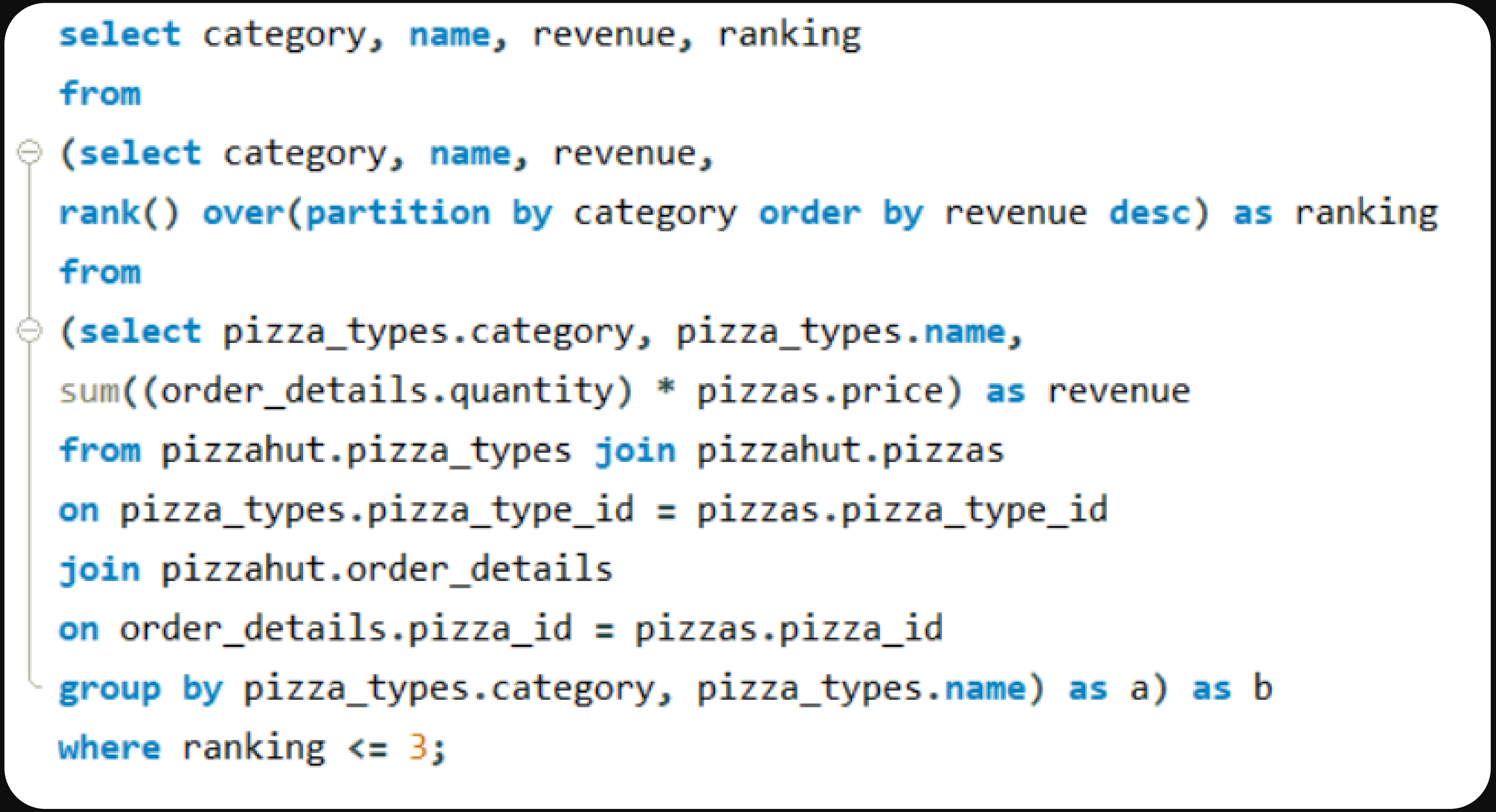
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006

PIZZA SALES SQL PROJECT

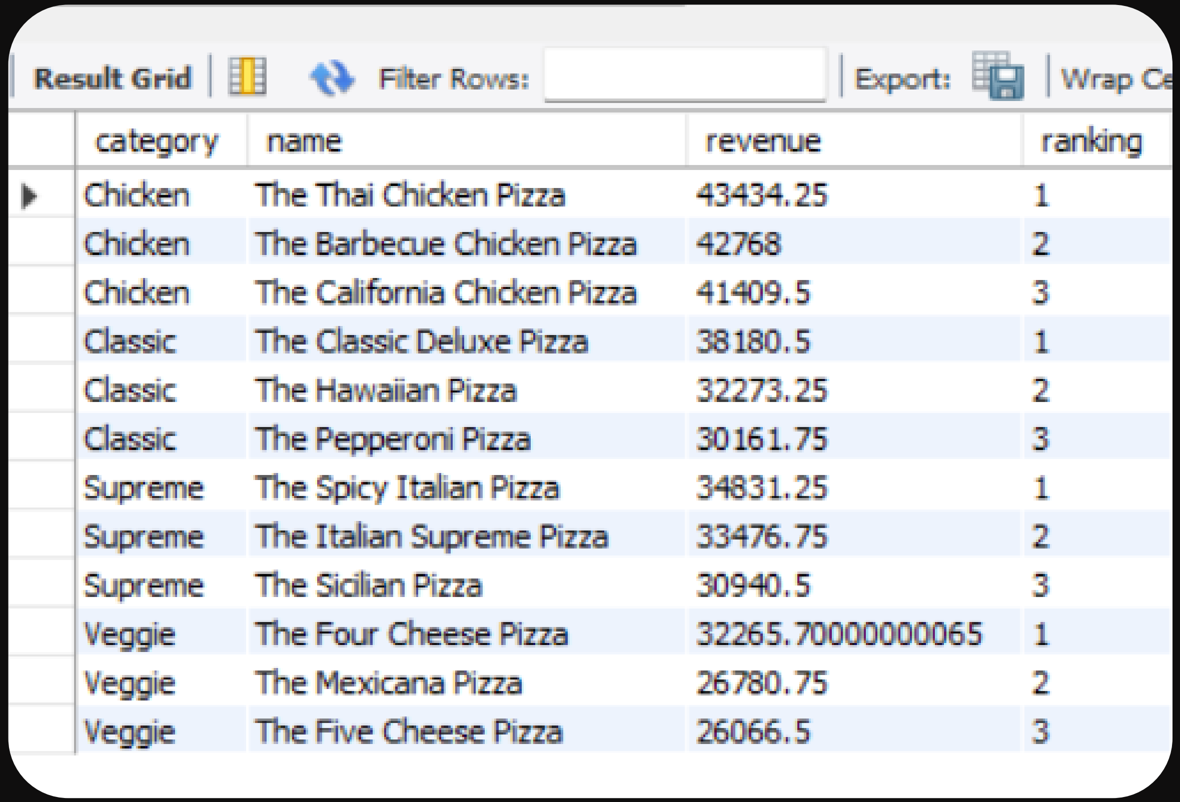
>>>THE QUESTIONS : FROM BASIC TO ADVANCED LEVEL

ADVANCED:

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



```
select category, name, revenue, ranking
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as ranking
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
from pizzahut.pizza_types join pizzahut.pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizzahut.order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where ranking <= 3;
```



category	name	revenue	ranking
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

SQL PROJECT

THANK YOU!

Feel free to ask questions or connect with me for collaboration!

Jayanta Nath, IIT Guwahati

Github Source

nathjayanta772@gmail.com

Guwahati-781039 , Assam