



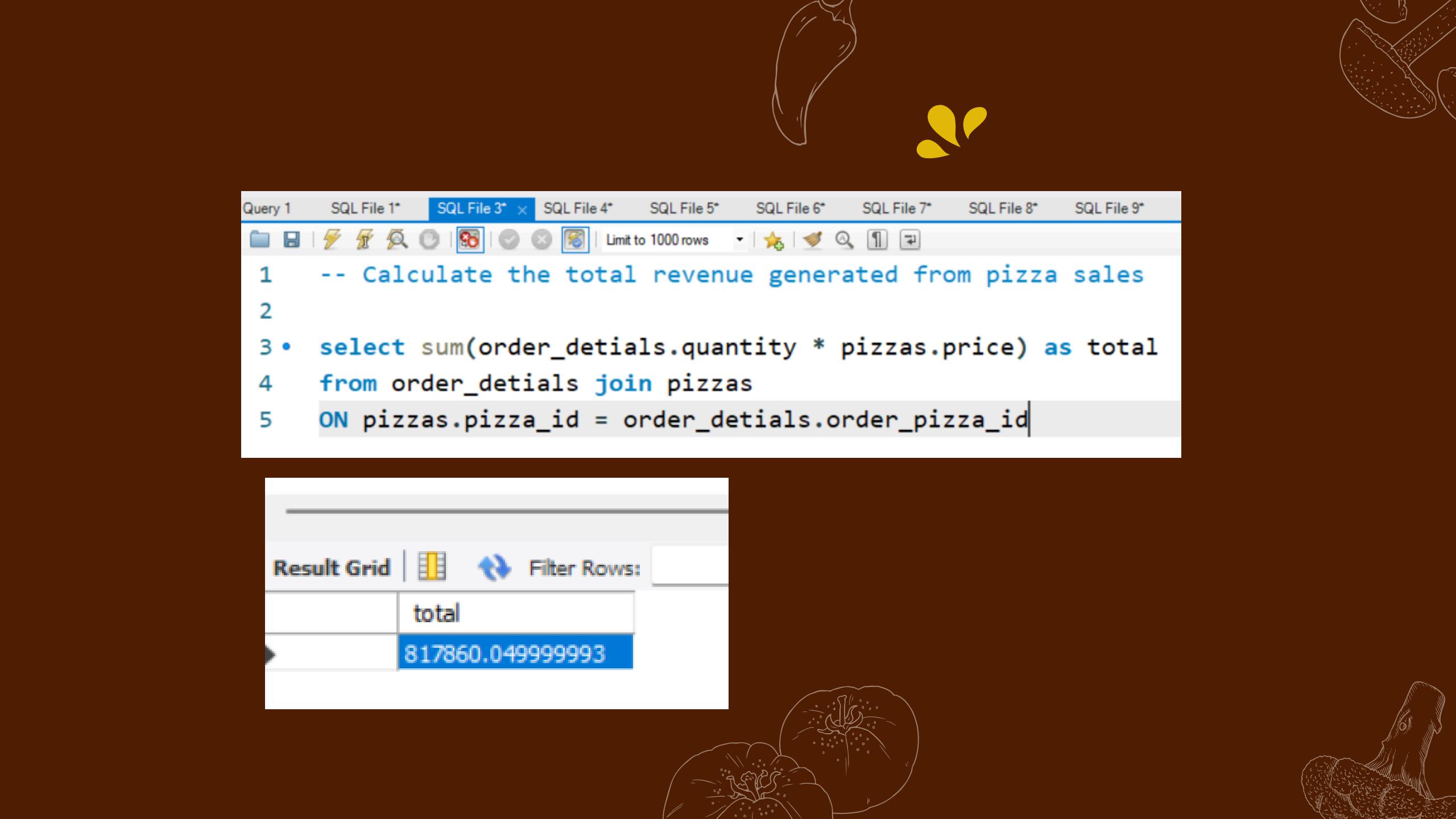
PIZZA SALE ANALYSIS USING SQL



```
1 -- Retrieve the total number of orders placed.  
2  
3 • select count(order_id)as total_orders from orders;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

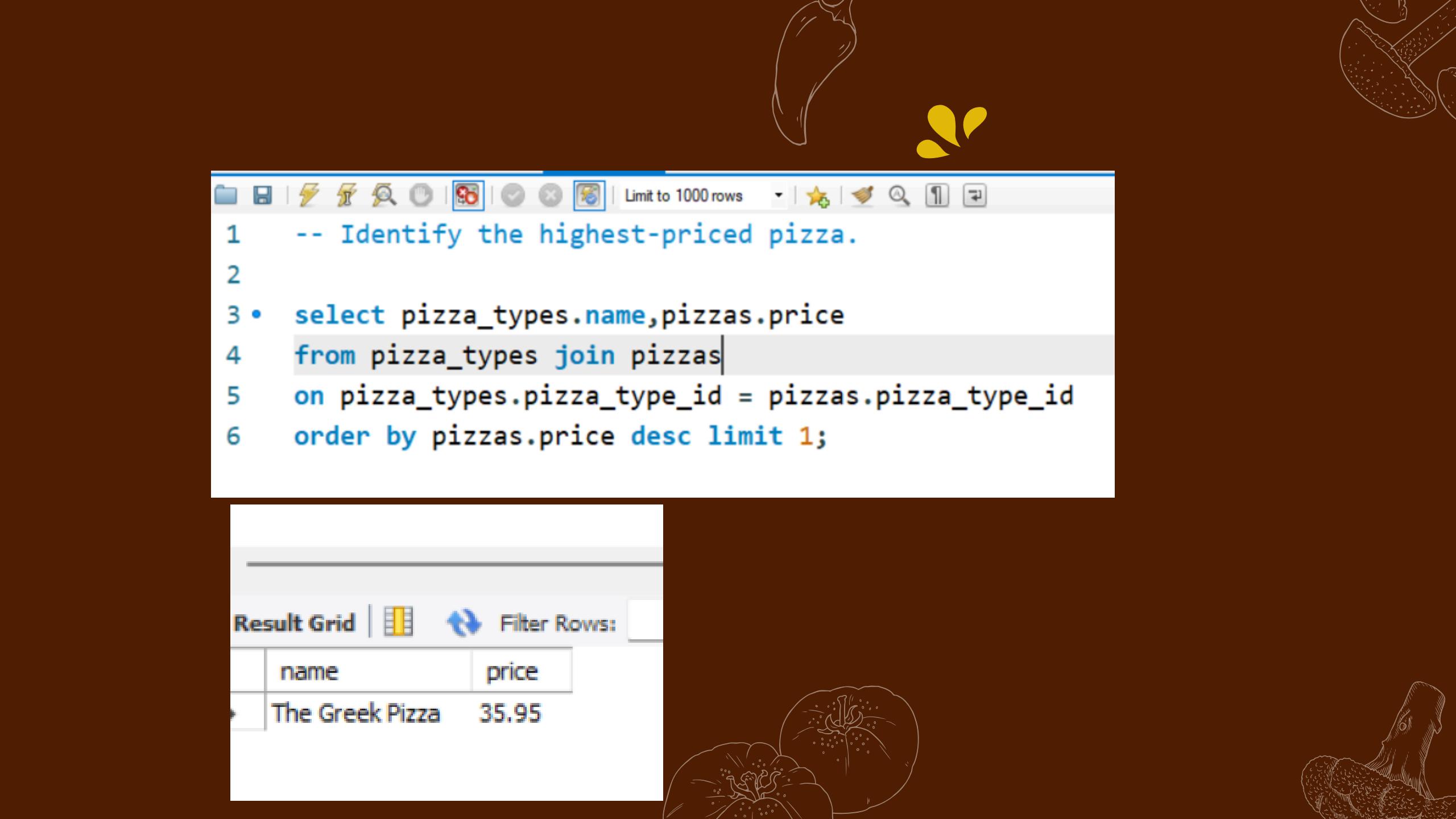
	total_orders
▶	21350



```
Query 1   SQL File 1*   SQL File 3* x   SQL File 4*   SQL File 5*   SQL File 6*   SQL File 7*   SQL File 8*   SQL File 9*
 1  -- Calculate the total revenue generated from pizza sales
 2
 3 • select sum(order_detials.quantity * pizzas.price) as total
 4  from order_detials join pizzas
 5  ON pizzas.pizza_id = order_detials.order_pizza_id
```

Result Grid | Filter Rows:

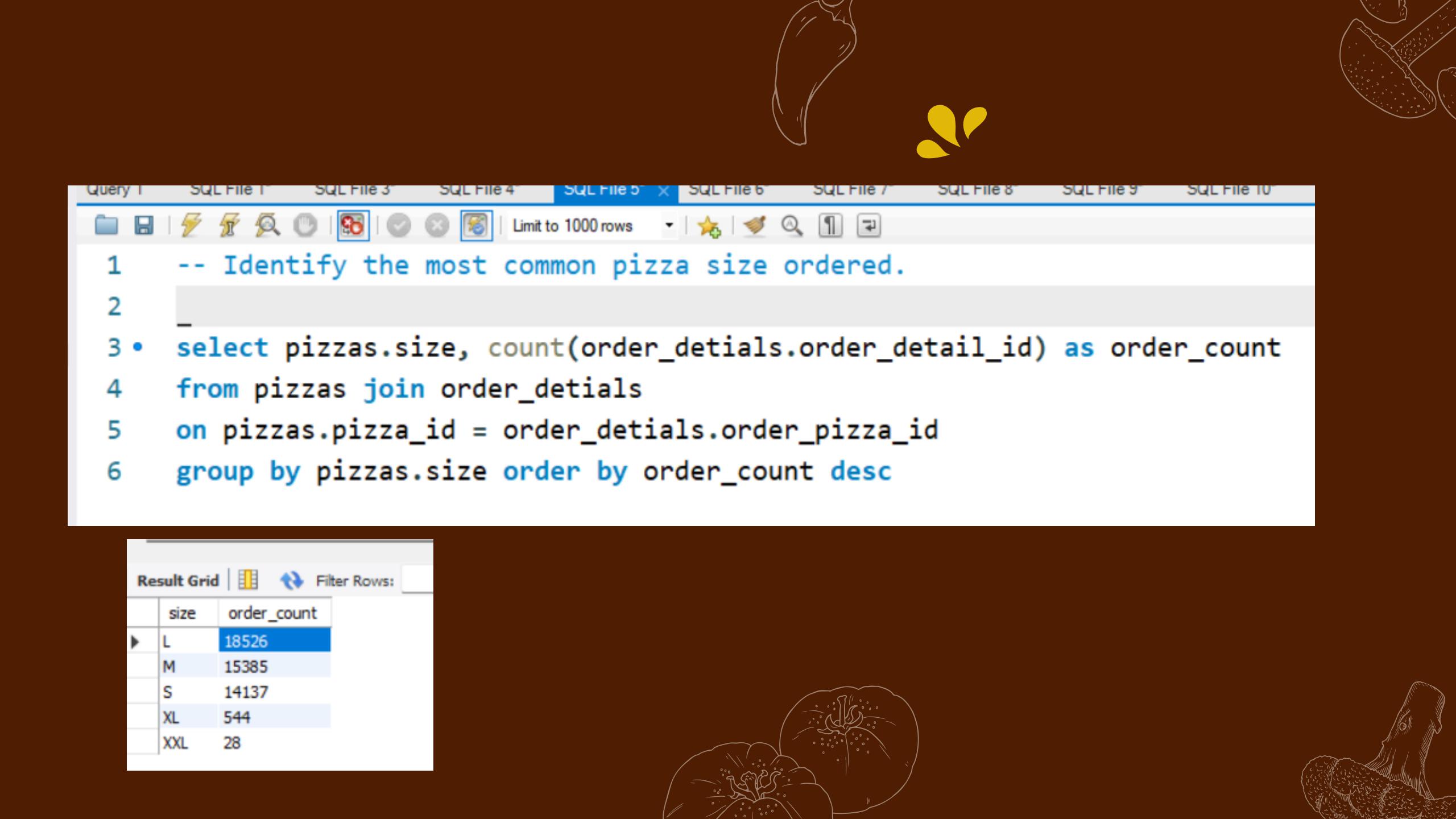
	total
	817860.049999993



```
1 -- Identify the highest-priced pizza.  
2  
3 • select pizza_types.name,pizzas.price  
4   from pizza_types join pizzas  
5     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6   order by pizzas.price desc limit 1;
```

Result Grid |  Filter Rows: 

	name	price
•	The Greek Pizza	35.95



```
Query 1 SQL File 1 SQL File 3 SQL File 4 SQL File 5 SQL File 6 SQL File 7 SQL File 8 SQL File 9 SQL File 10
 1 -- Identify the most common pizza size ordered.
 2 -
 3 • select pizzas.size, count(order_detials.order_detail_id) as order_count
 4   from pizzas join order_detials
 5     on pizzas.pizza_id = order_detials.order_pizza_id
 6   group by pizzas.size order by order_count desc
```

Result Grid | Filter Rows:

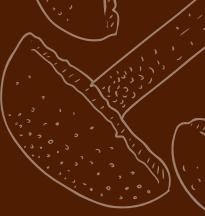
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28



```
1 -- List the top 5 most ordered pizza types along with their quantities.  
2  
3 • select pizza_types.name,sum(order_detials.quantity) as quantity  
4 from pizza_types  
5 join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6 join order_detials on order_detials.order_pizza_id = pizzas.pizza_id  
7 group by pizza_types.name order by quantity desc limit 5;
```



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



```
Query 1 SQL File 1 SQL File 3 SQL File 4 SQL File 5 SQL File 6 SQL File 7 X SQL File 8 SQL File 9 SQL File 10
 1 -- Join the necessary tables to find the total quantity of each pizza category ordered.
 2
 3 • select pizza_types.category,
 4     sum(order_detials.quantity) as quantity
 5   from pizza_types
 6   join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
 7   join order_detials on order_detials.order_pizza_id = pizzas.pizza_id
 8   group by pizza_types.category order by quantity desc ;
```

Result Grid | Filter Row

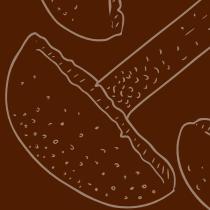
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



```
1 -- Determine the distribution of orders by hour of the day.  
2  
3 • select hour(order_time) as hour ,count(order_id) as order_count  
4   from orders  
5   group by hour(order_time);
```

Result Grid | Filter

	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



```
1 -- Join relevant tables to find the category-wise distribution of pizzas.  
2  
3 • select category, count(name) from pizza_types  
4 group by category
```

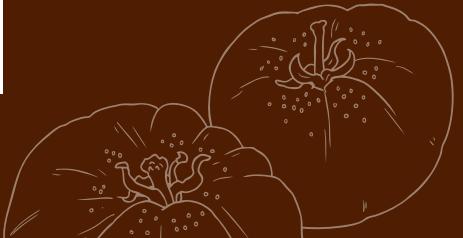
Result Grid | Filter Rows:

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

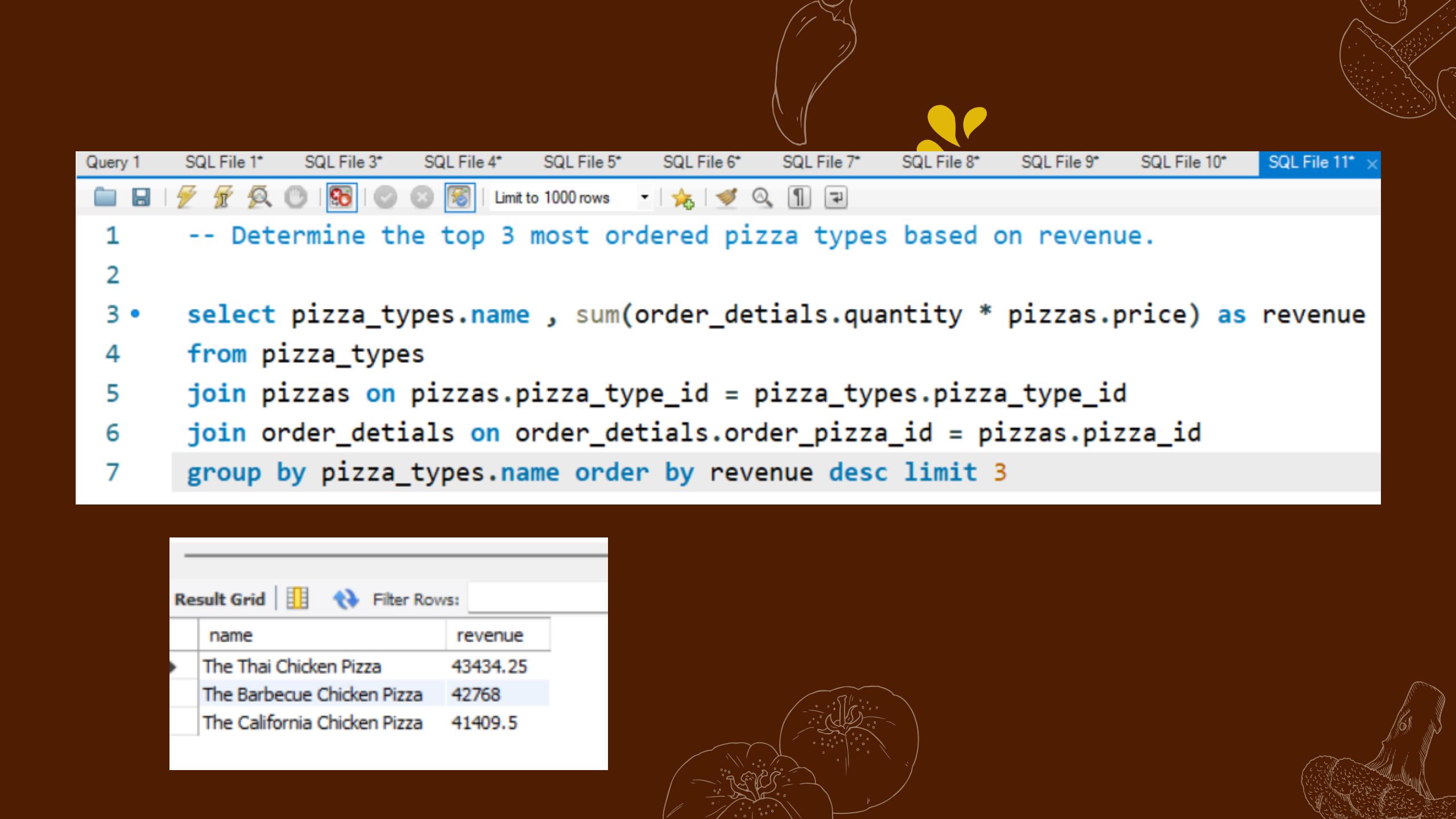




```
1 -- Group the orders by date and calculate the average
2 -- number of pizzas ordered per day
3
4 • select avg(quantity) from
5   (select orders.order_date, sum(order_detials.quantity) as quantity
6   from orders join order_detials
7   on orders.order_id = order_detials.order_id
8   group by orders.order_date ) as order_quantity;
```



	avg(quantity)
▶	138.4749



```
Query 1 SQL File 1* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* x
File Edit View Insert Tools Window Help
Limit to 1000 rows
1 -- Determine the top 3 most ordered pizza types based on revenue.
2
3 • select pizza_types.name , sum(order_detials.quantity * pizzas.price) as revenue
4   from pizza_types
5   join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id
6   join order_detials on order_detials.order_pizza_id = pizzas.pizza_id
7   group by pizza_types.name order by revenue desc limit 3
```

Result Grid | Filter Rows:

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

Query 1 SQL File 1* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL Fil

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

2

3 • select pizza_types.category , sum(order_detials.quantity * pizzas.price) /

4 (select sum(order_detials.quantity * pizzas.price) as total

5 from order_detials

6 join pizzas ON pizzas.pizza_id = order_detials.order_pizza_id)*100 as revenue

7 from pizza_types

8 join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id

9 join order_detials on order_detials.order_pizza_id = pizzas.pizza_id

10 group by pizza_types.category order by revenue desc

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.905960255669903
	Supreme	25.45631126009884
	Chicken	23.955137556847493
	Veggie	23.682590927384783

```
1 -- Analyze the cumulative revenue generated over time.
2 • select order_date, sum(revenue) over(order by order_date) as cum_revenue
3   from
4   (select orders.order_date , sum(order_detials.quantity * pizzas.price) as revenue
5     from order_detials
6       join pizzas on pizzas.pizza_id = order_detials.order_pizza_id
7       join orders on orders.order_id = order_detials.order_id
8     group by orders.order_date) as sales;
```

	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

```
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
2  
3 • select name,revenue from  
4   (select category, name, revenue,rank() over(partition by category order by revenue desc) as rn  
5    from  
6    (select pizza_types.category, pizza_types.name, sum(order_detials.quantity * pizzas.price) as revenue  
7     from pizza_types  
8      join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id  
9      join order_detials on order_detials.order_pizza_id = pizzas.pizza_id  
10     group by pizza_types.category, pizza_types.name) as a) as b  
11   where rn<=3;
```

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