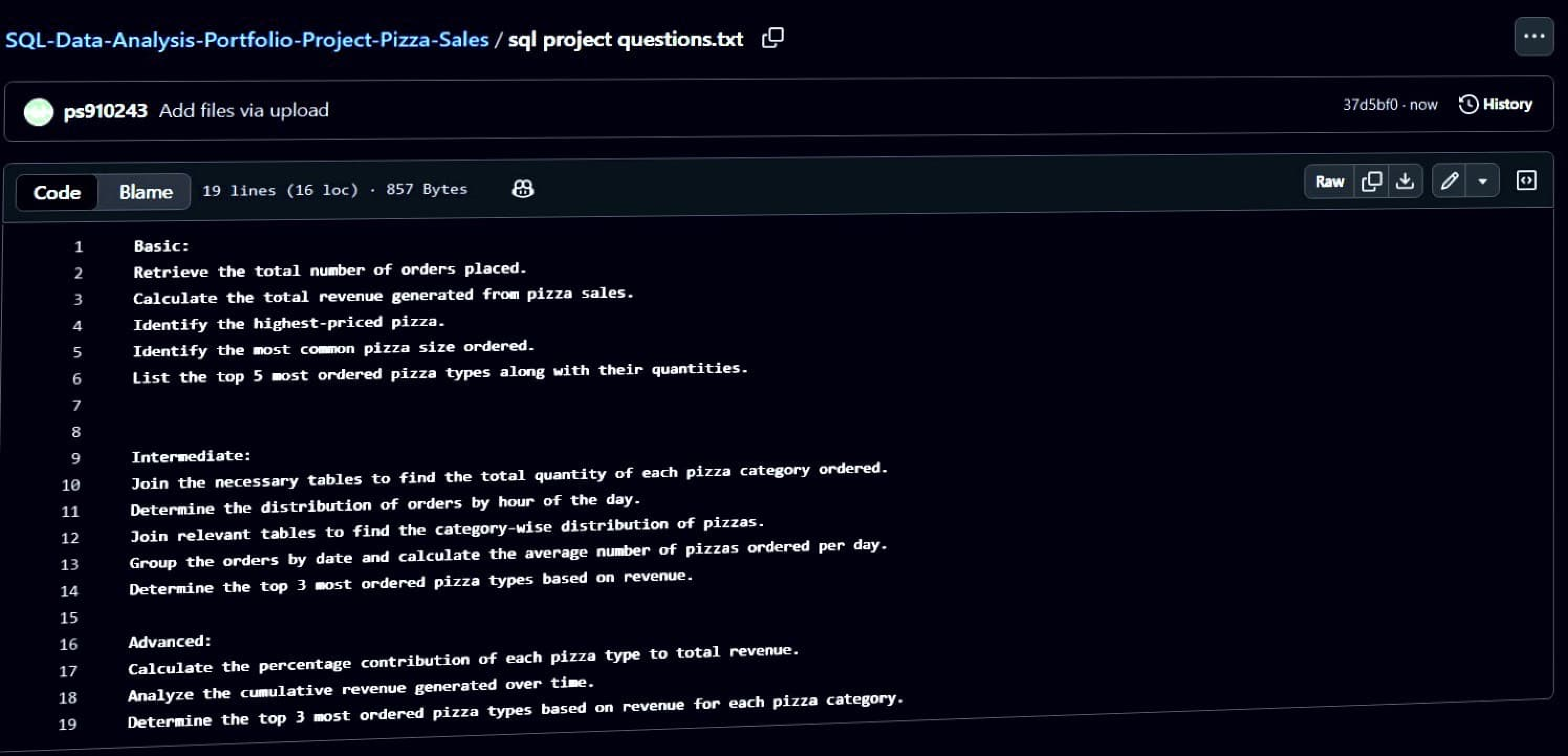
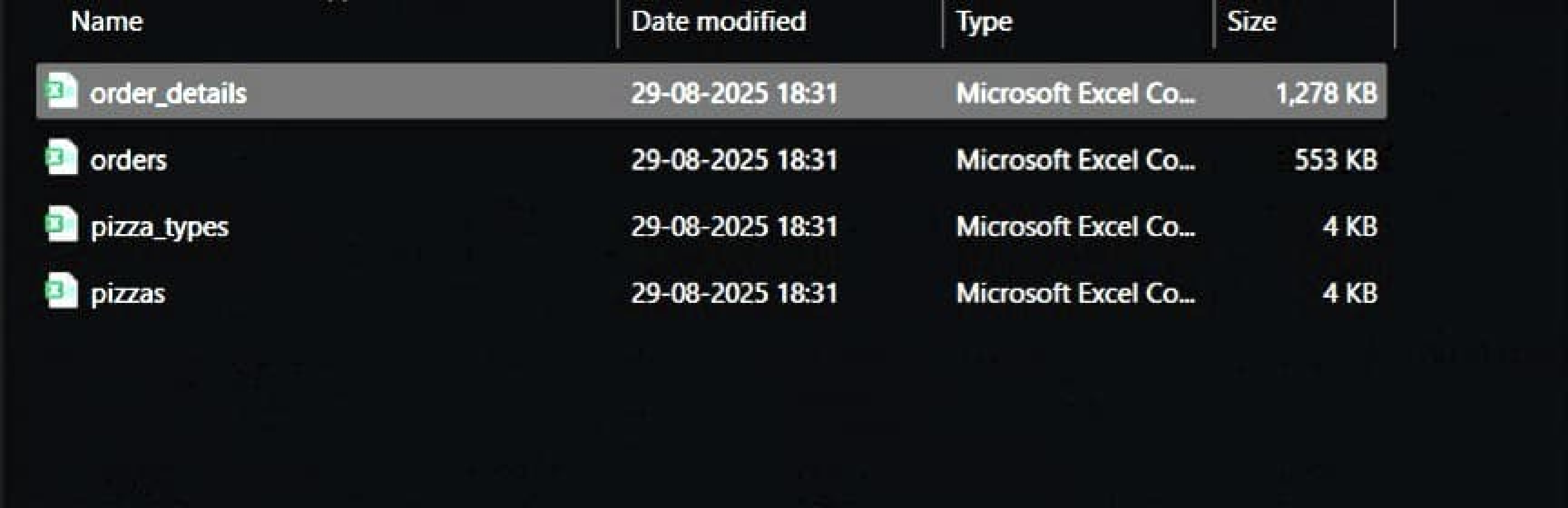

SQL-Data-Analysis-Portfolio-Project-Pizza-Sales / sql project questions.txt
ps910243 Add files via upload
37d5bf0 · now
History
Code
Blame
19 lines (16 loc) · 857 Bytes
Raw

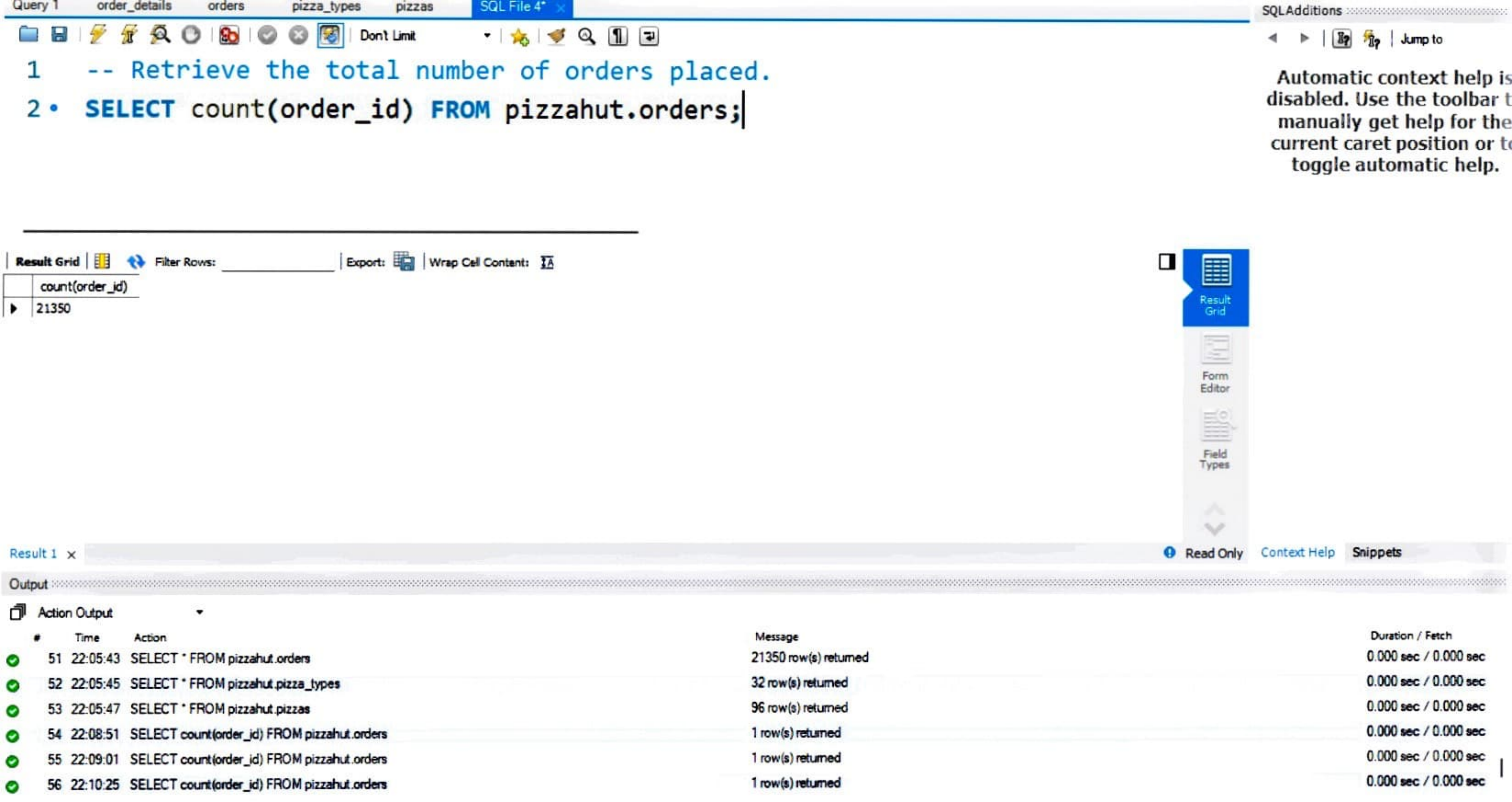
Basic:
Retrieve the total number of orders placed.
Calculate the total revenue generated from pizza sales.
Identify the highest-priced pizza.
Identify the most common pizza size ordered.
List the top 5 most ordered pizza types along with their quantities.

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.



Name	Date modified	Type	Size
order_details	29-08-2025 18:31	Microsoft Excel Co...	1,278 KB
orders	29-08-2025 18:31	Microsoft Excel Co...	553 KB
pizza_types	29-08-2025 18:31	Microsoft Excel Co...	4 KB
pizzas	29-08-2025 18:31	Microsoft Excel Co...	4 KB

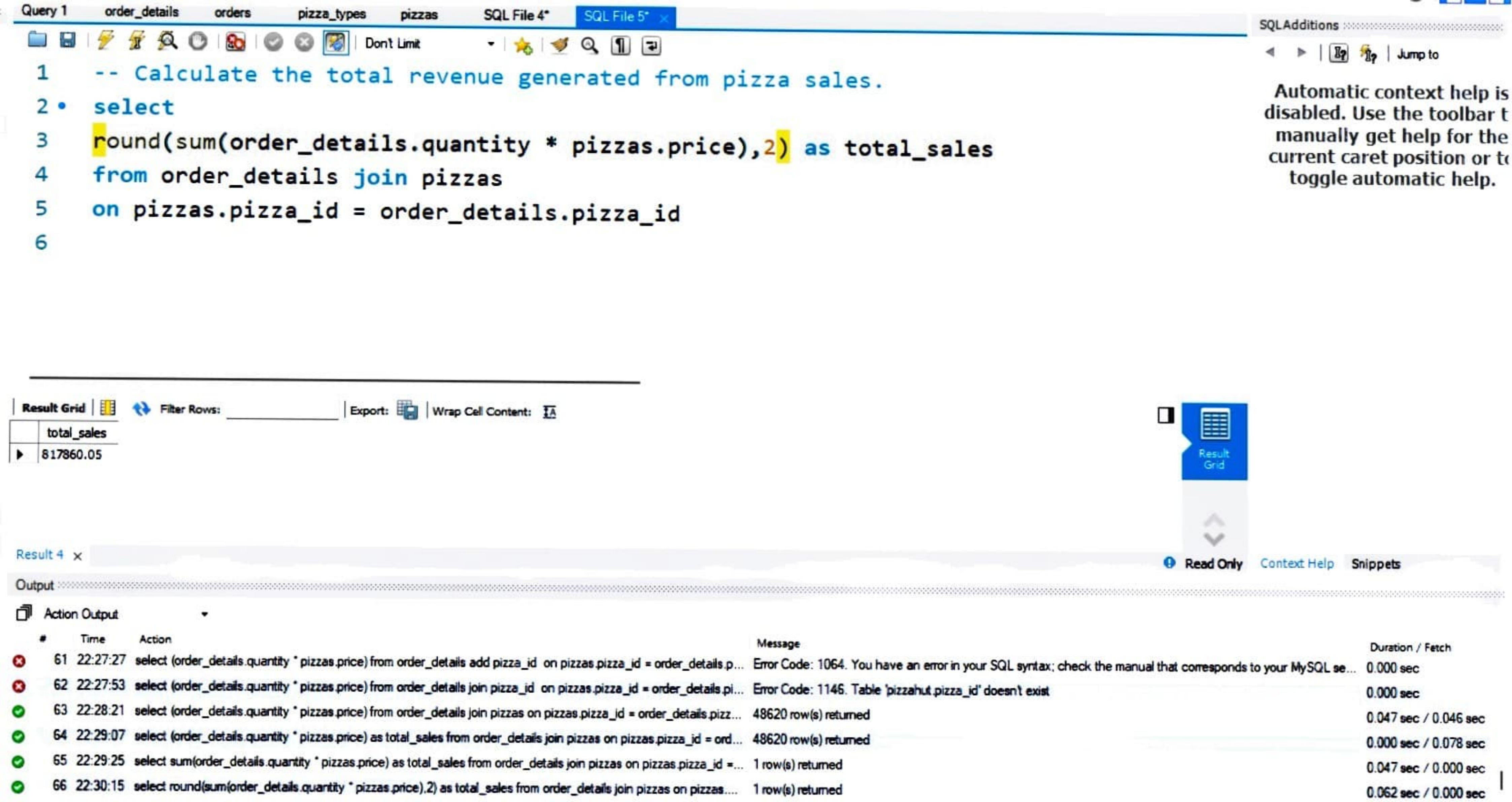


```sql
-- Retrieve the total number of orders placed.
SELECT count(order_id) FROM pizzahut.orders;
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

```
count(order_id)
21350
```

```
#   Time      Action                                      Message                 Duration / Fetch
51  22:05:43  SELECT * FROM pizzahut.orders               21350 row(s) returned   0.000 sec / 0.000 sec
52  22:05:45  SELECT * FROM pizzahut.pizza_types          32 row(s) returned      0.000 sec / 0.000 sec
53  22:05:47  SELECT * FROM pizzahut.pizzas               96 row(s) returned      0.000 sec / 0.000 sec
54  22:08:51  SELECT count(order_id) FROM pizzahut.orders  1 row(s) returned       0.000 sec / 0.000 sec
55  22:09:01  SELECT count(order_id) FROM pizzahut.orders  1 row(s) returned       0.000 sec / 0.000 sec
56  22:10:25  SELECT count(order_id) FROM pizzahut.orders  1 row(s) returned       0.000 sec / 0.000 sec
```

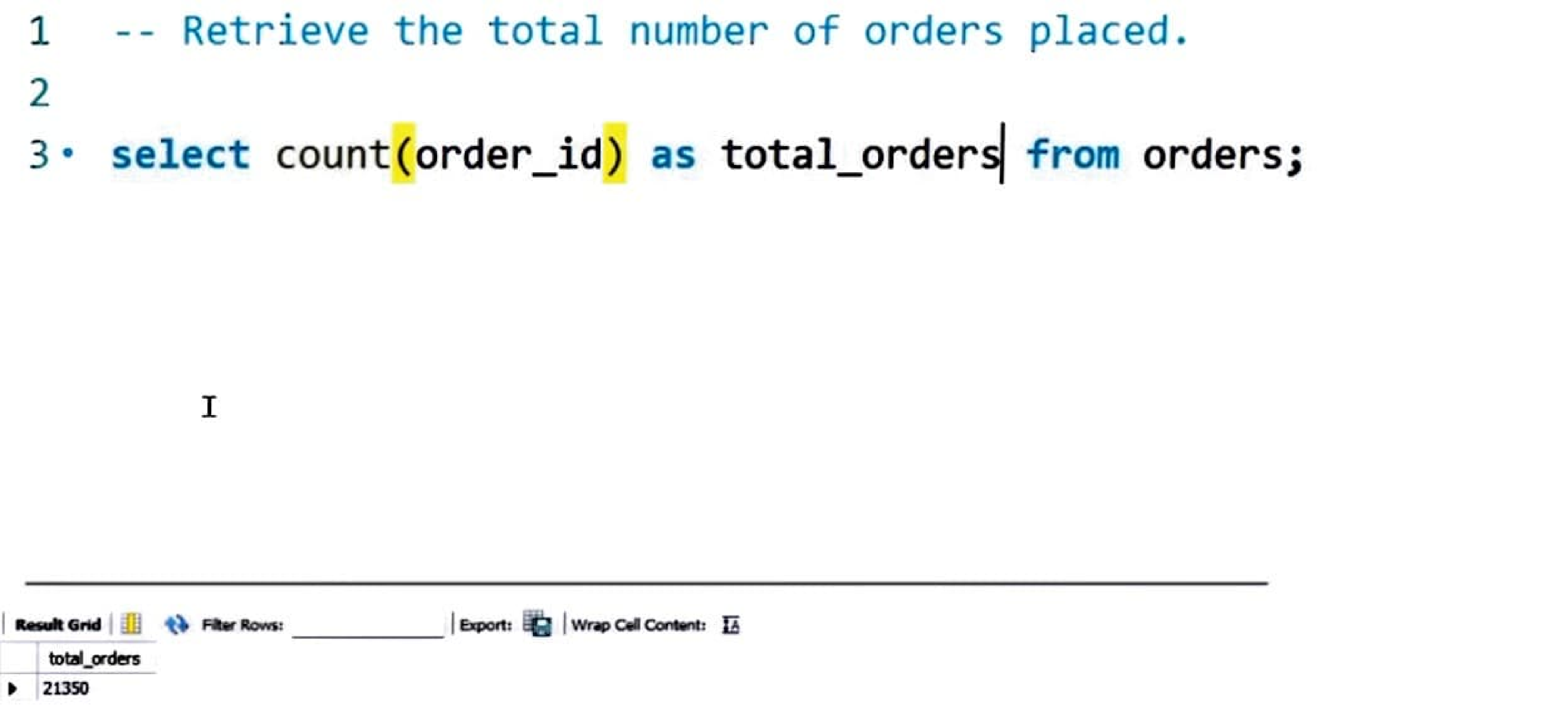


```sql
-- Calculate the total revenue generated from pizza sales.
select
round(sum(order_details.quantity * pizzas.price),2) as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

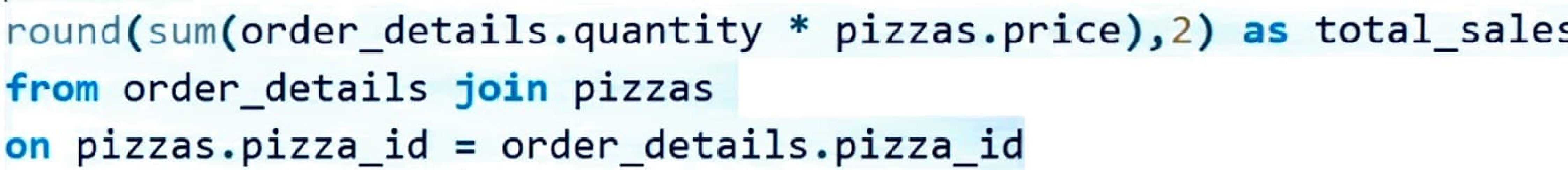
total_sales
817860.05

```
 #  Time      Action                                                                                                  Message                                                                                  Duration / Fetch
61  22:27:27  select (order_details.quantity * pizzas.price) from order_details add pizza_id  on pizzas.pizza_id = order_details.p...  Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...  0.000 sec
62  22:27:53  select (order_details.quantity * pizzas.price) from order_details join pizza_id  on pizzas.pizza_id = order_details.pi...  Error Code: 1146. Table 'pizzahut.pizza_id' doesn't exist                                  0.000 sec
63  22:28:21  select (order_details.quantity * pizzas.price) from order_details join pizzas on pizzas.pizza_id = order_details.pizz...  48620 row(s) returned                                                                     0.047 sec / 0.046 sec
64  22:29:07  select (order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id = ord...  48620 row(s) returned                                                                     0.000 sec / 0.078 sec
65  22:29:25  select sum(order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id =...  1 row(s) returned                                                                         0.047 sec / 0.000 sec
66  22:30:15  select round(sum(order_details.quantity * pizzas.price),2) as total_sales from order_details join pizzas on pizzas....  1 row(s) returned                                                                         0.062 sec / 0.000 sec
```

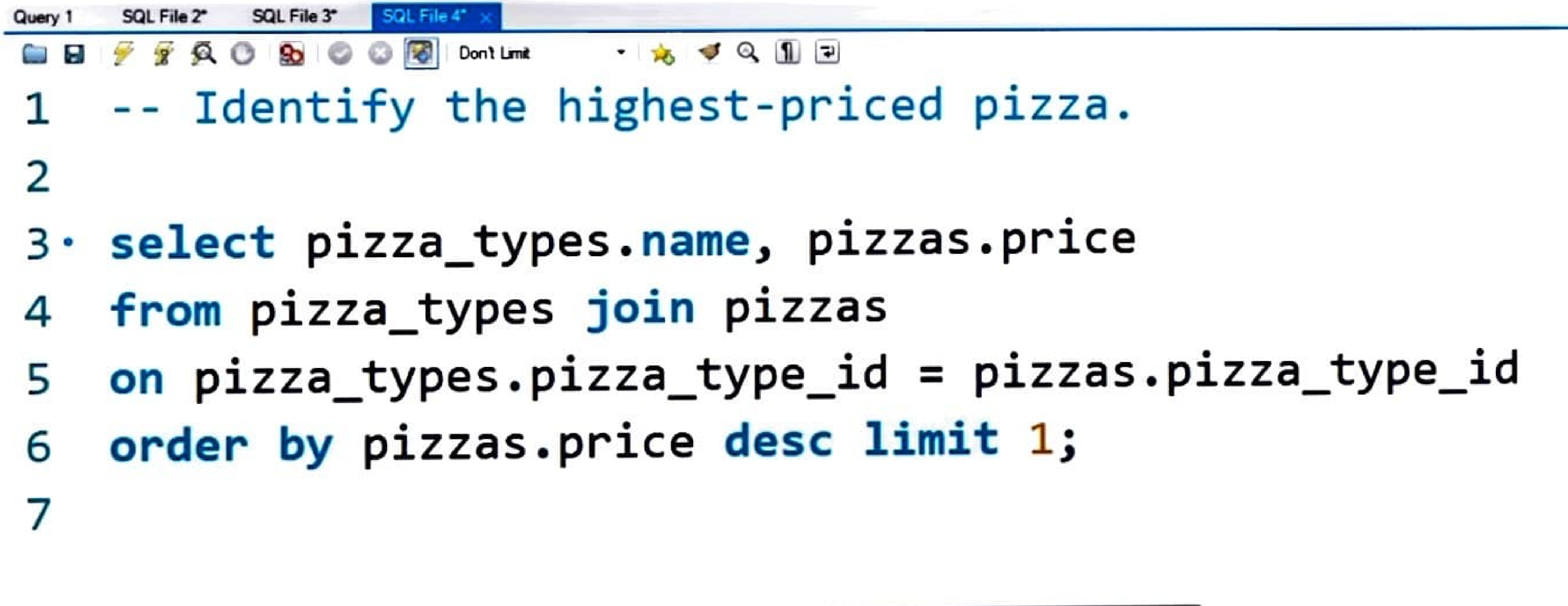


```sql
-- Retrieve the total number of orders placed.

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

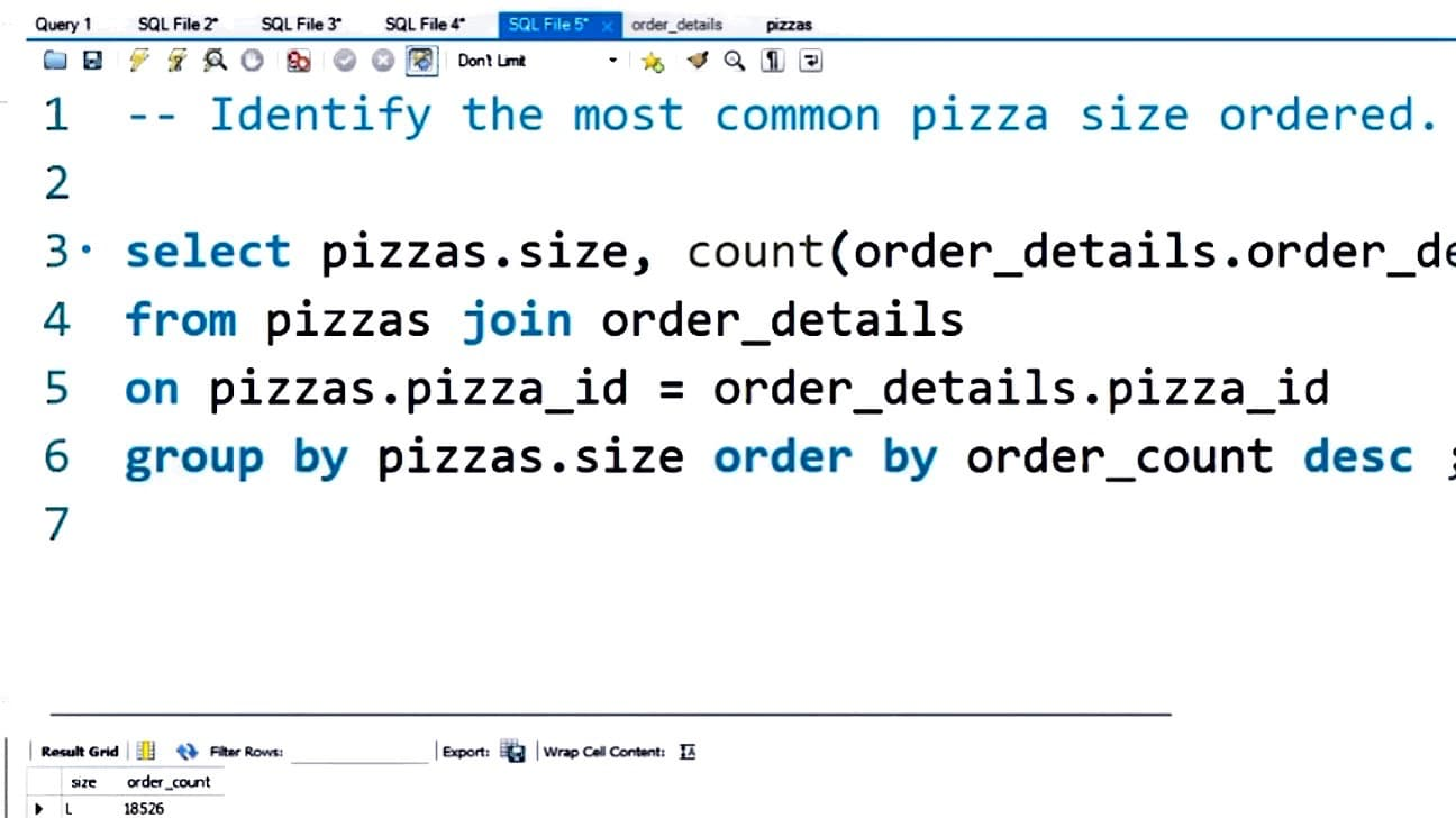
```
total_orders
21350
```



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

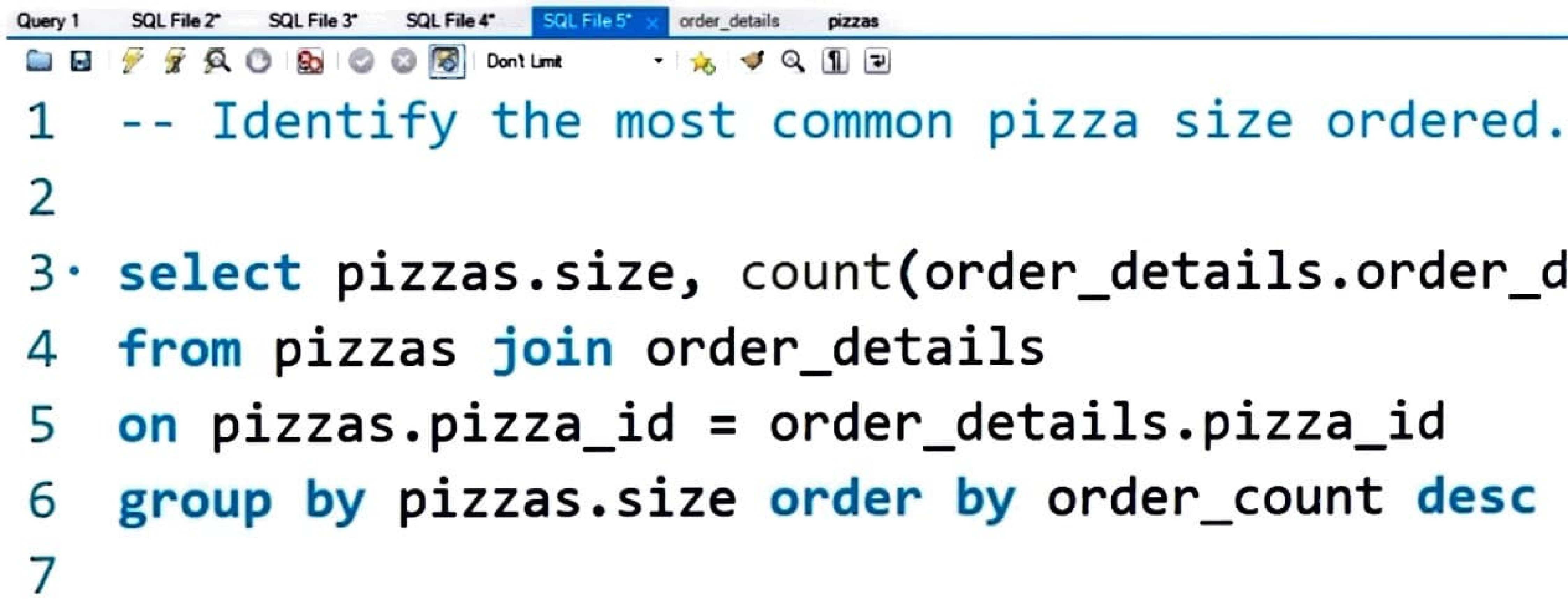


```sql
-- Identify the highest-priced pizza.

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;

```

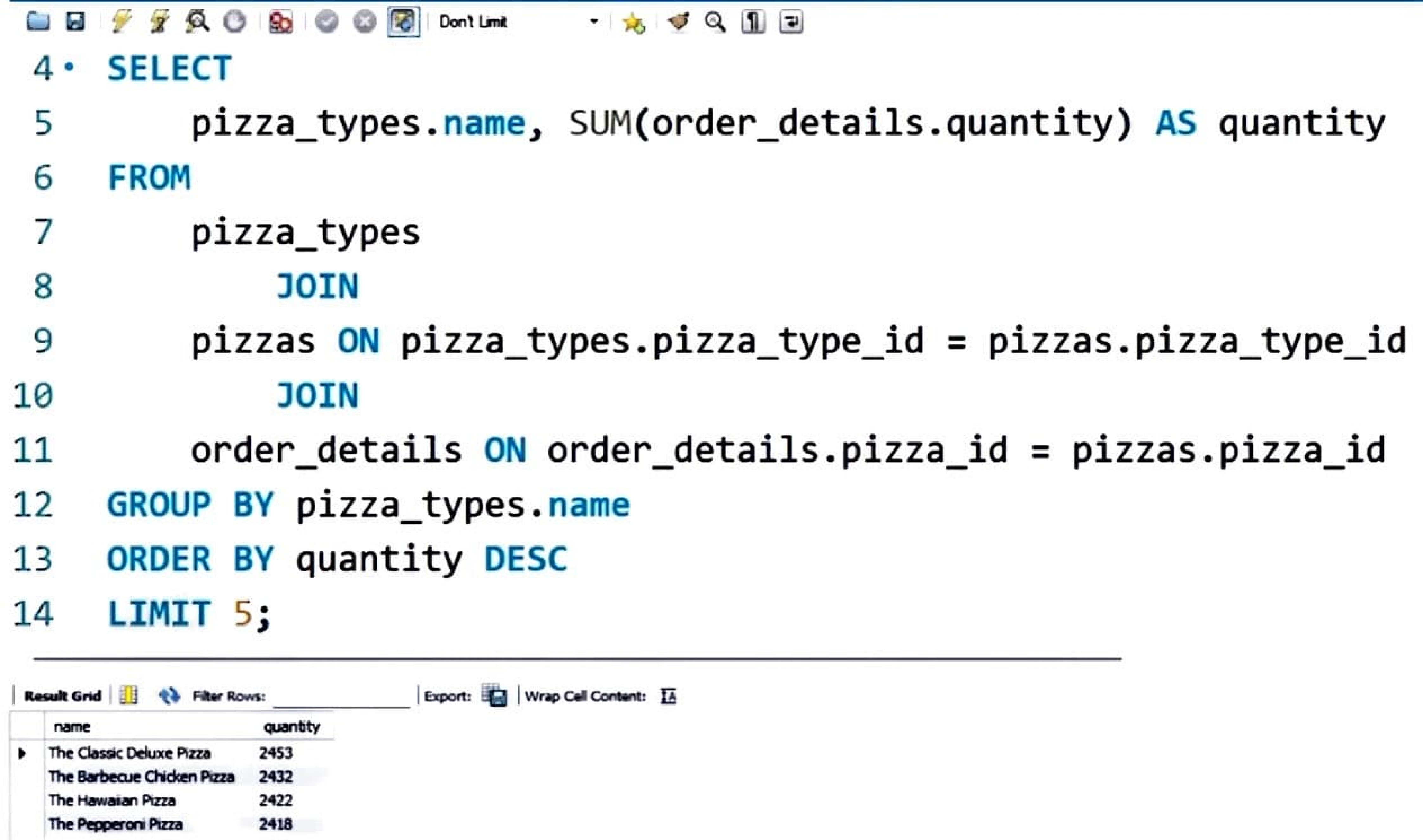


```sql
-- Identify the most common pizza size ordered.

select pizzas.size, count(order_details.order_de
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

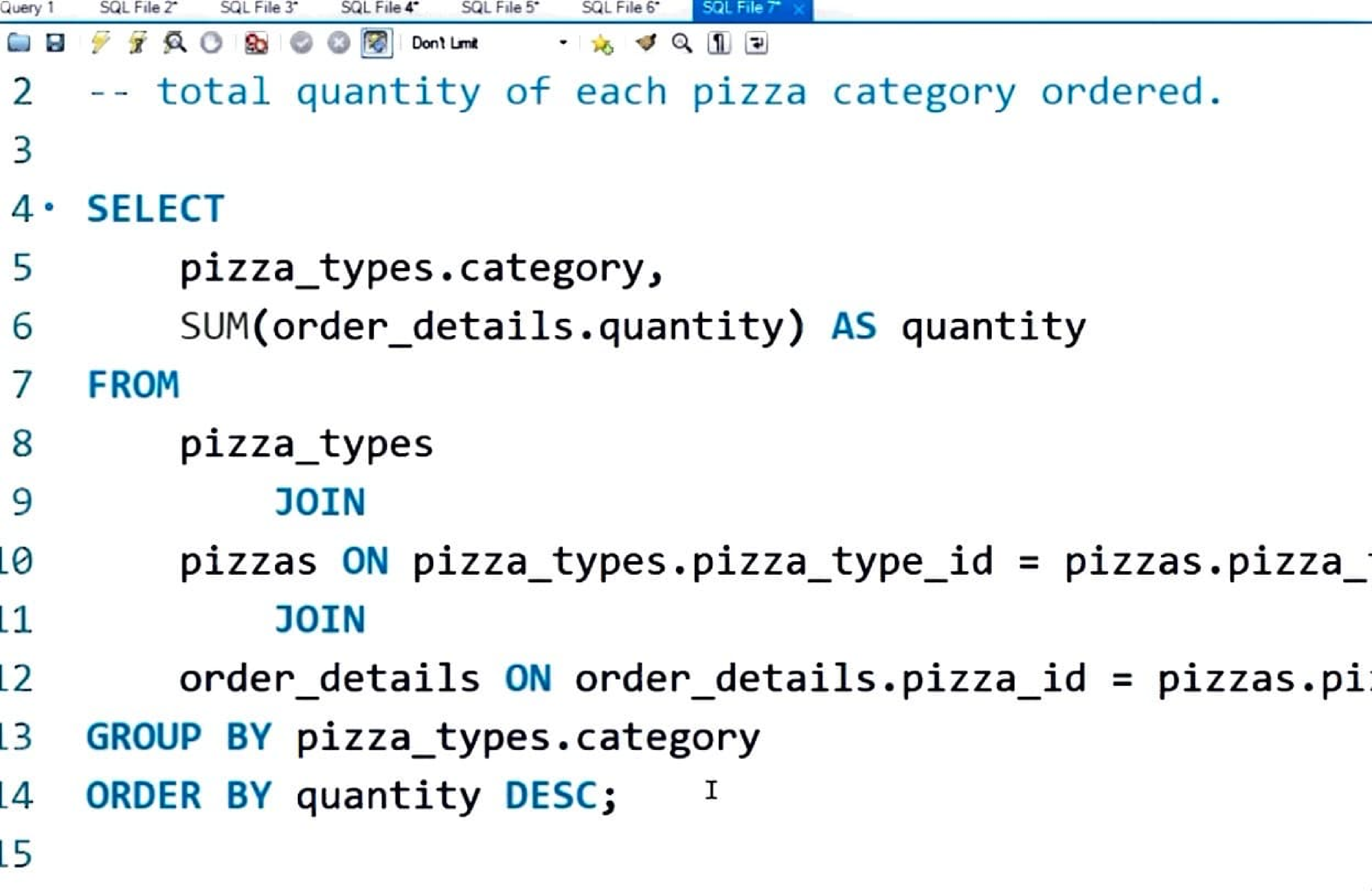


```
-- Identify the most common pizza size ordered.

select pizzas.size, count(order_details.order_d
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc

```

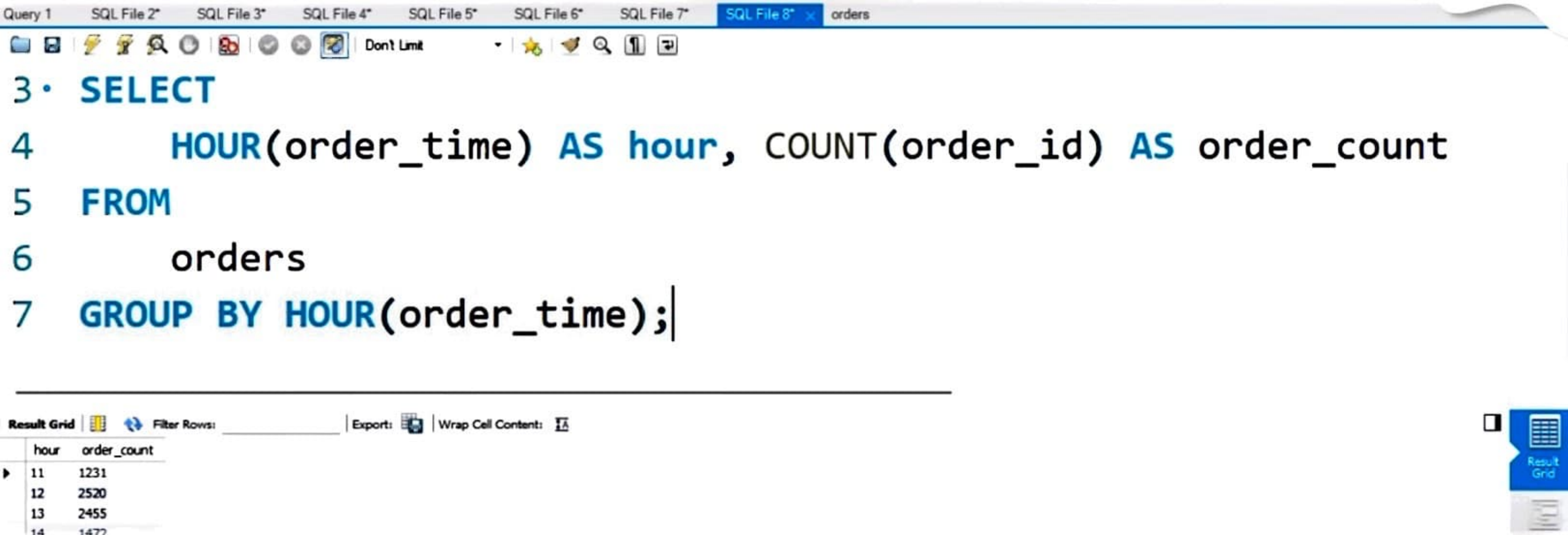


```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

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

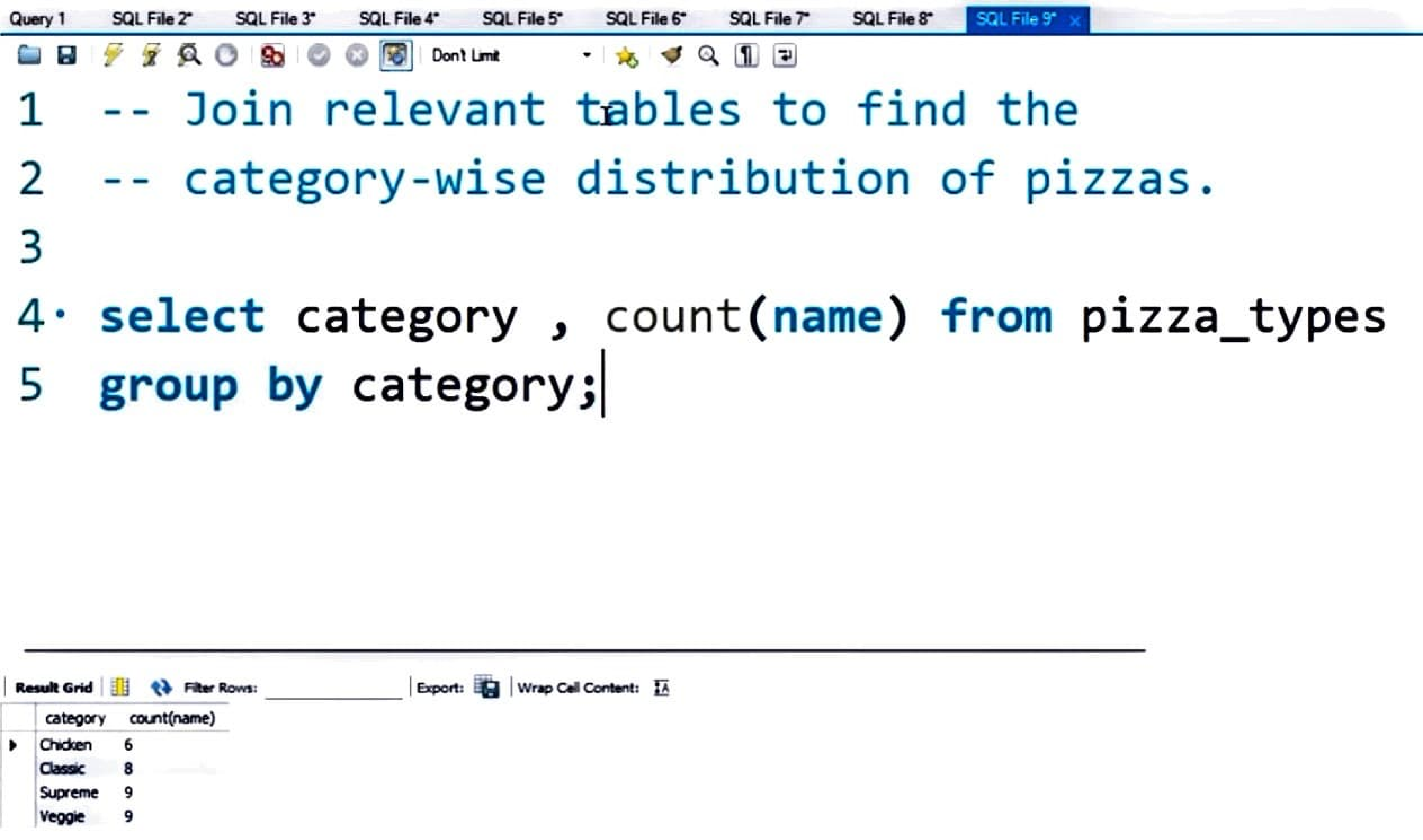


```
-- total quantity of each pizza category ordered.

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_
        JOIN
    order_details ON order_details.pizza_id = pizzas.pi
GROUP BY pizza_types.category
ORDER BY quantity DESC;

```



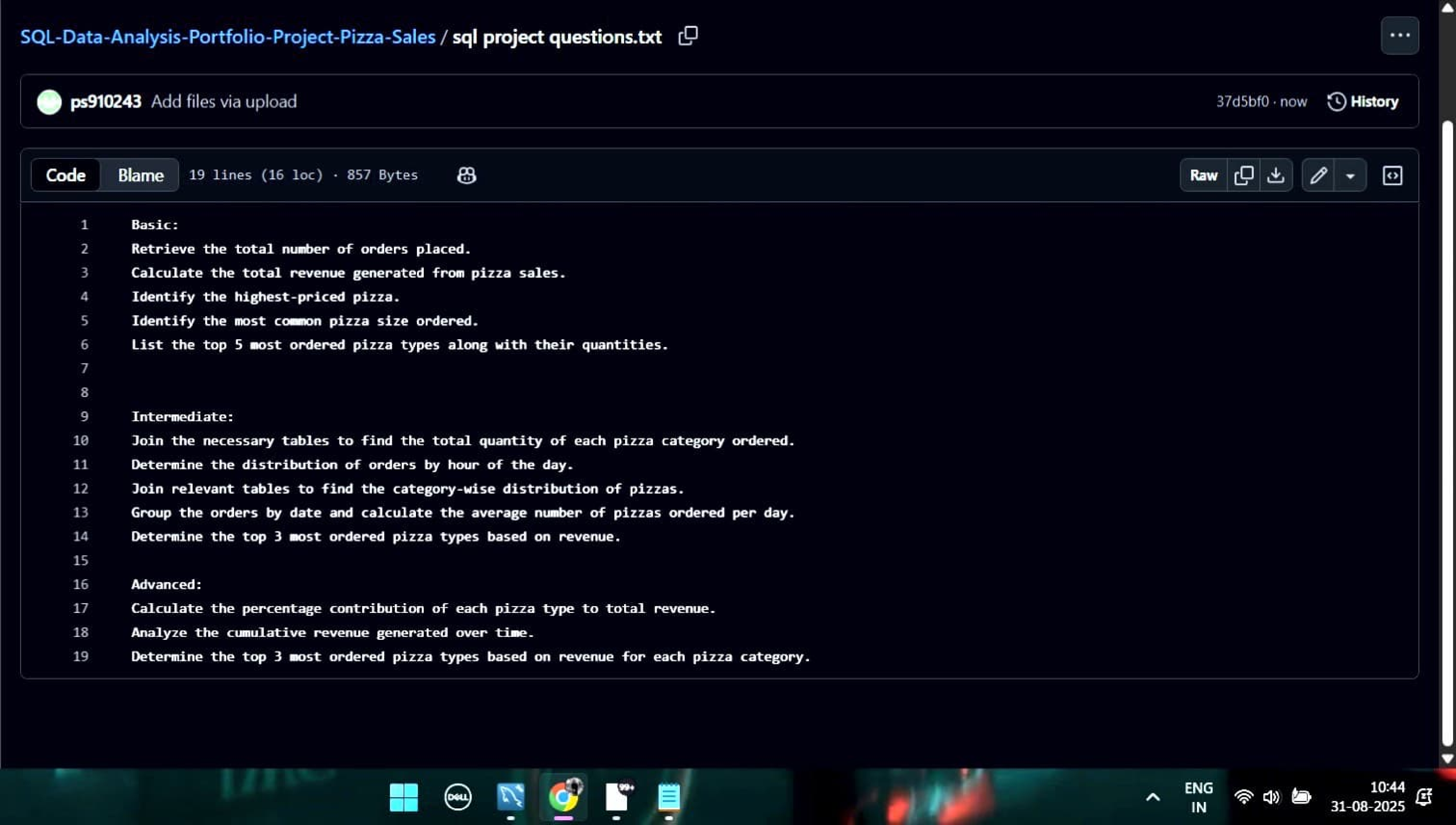
```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

```
hour  order_count
11    1231
12    2520
13    2455
14    1472
```



```sql
-- 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
```

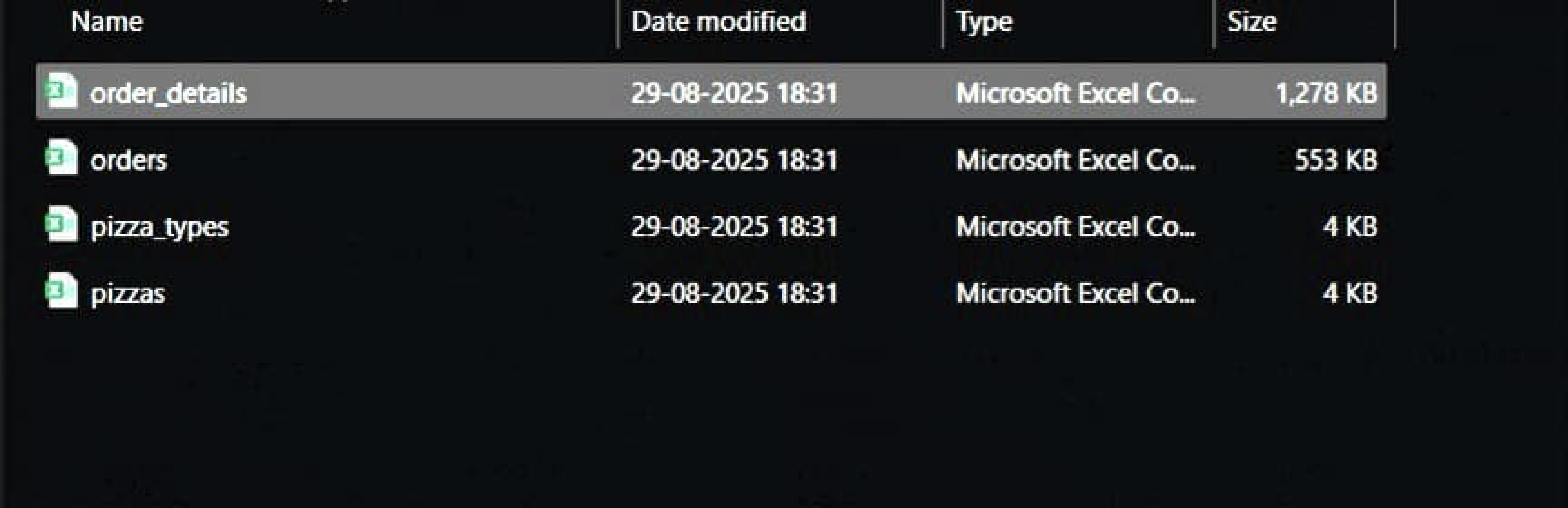


SQL-Data-Analysis-Portfolio-Project-Pizza-Sales / sql project questions.txt

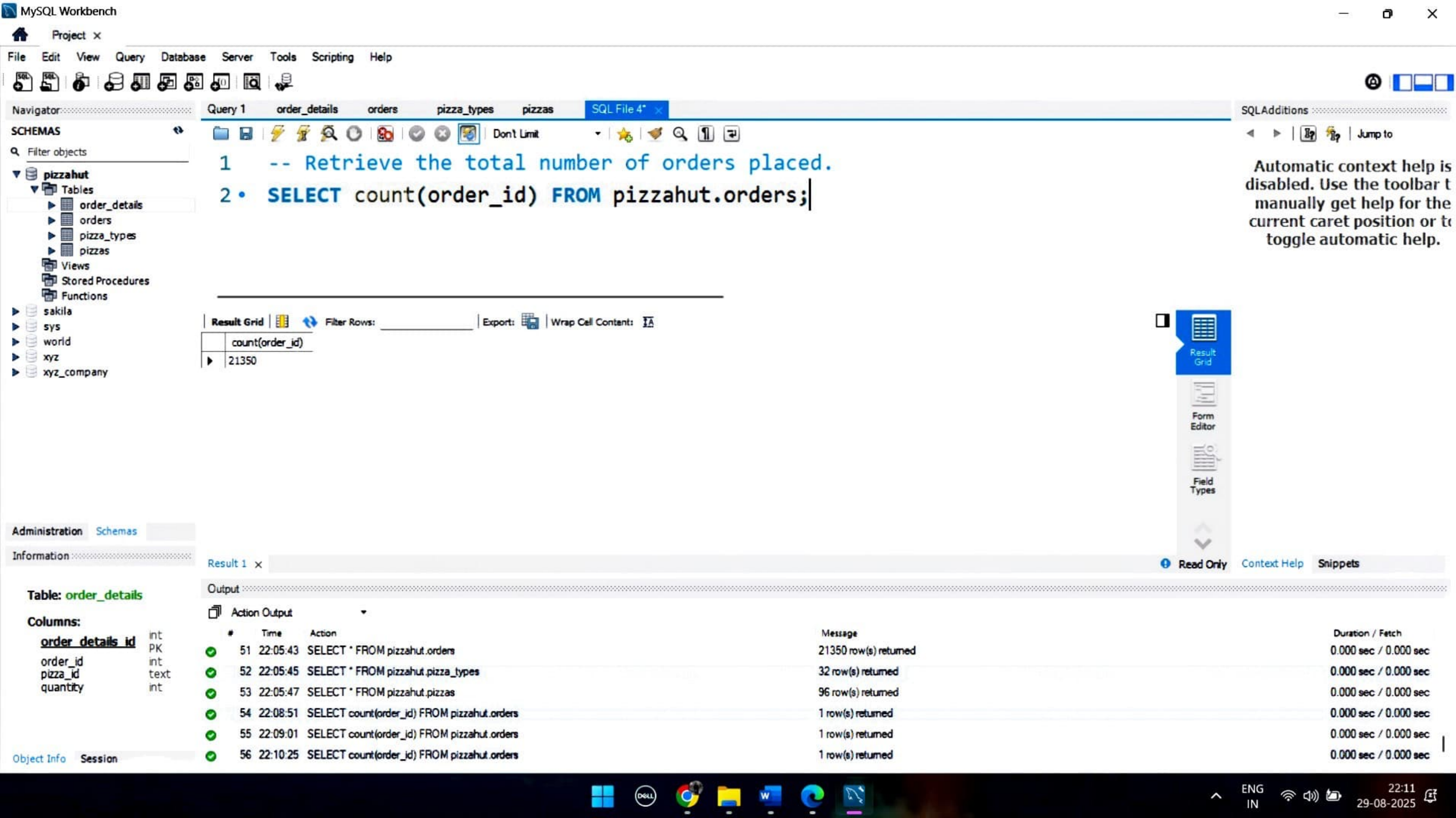
ps910243 Add files via upload 37d5bf0 · now

19 lines (16 loc) · 857 Bytes

```
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.
```



Name	Date modified	Type	Size
order_details	29-08-2025 18:31	Microsoft Excel Co...	1,278 KB
orders	29-08-2025 18:31	Microsoft Excel Co...	553 KB
pizza_types	29-08-2025 18:31	Microsoft Excel Co...	4 KB
pizzas	29-08-2025 18:31	Microsoft Excel Co...	4 KB



```sql
-- Retrieve the total number of orders placed.
SELECT count(order_id) FROM pizzahut.orders;
```

count(order_id)
21350

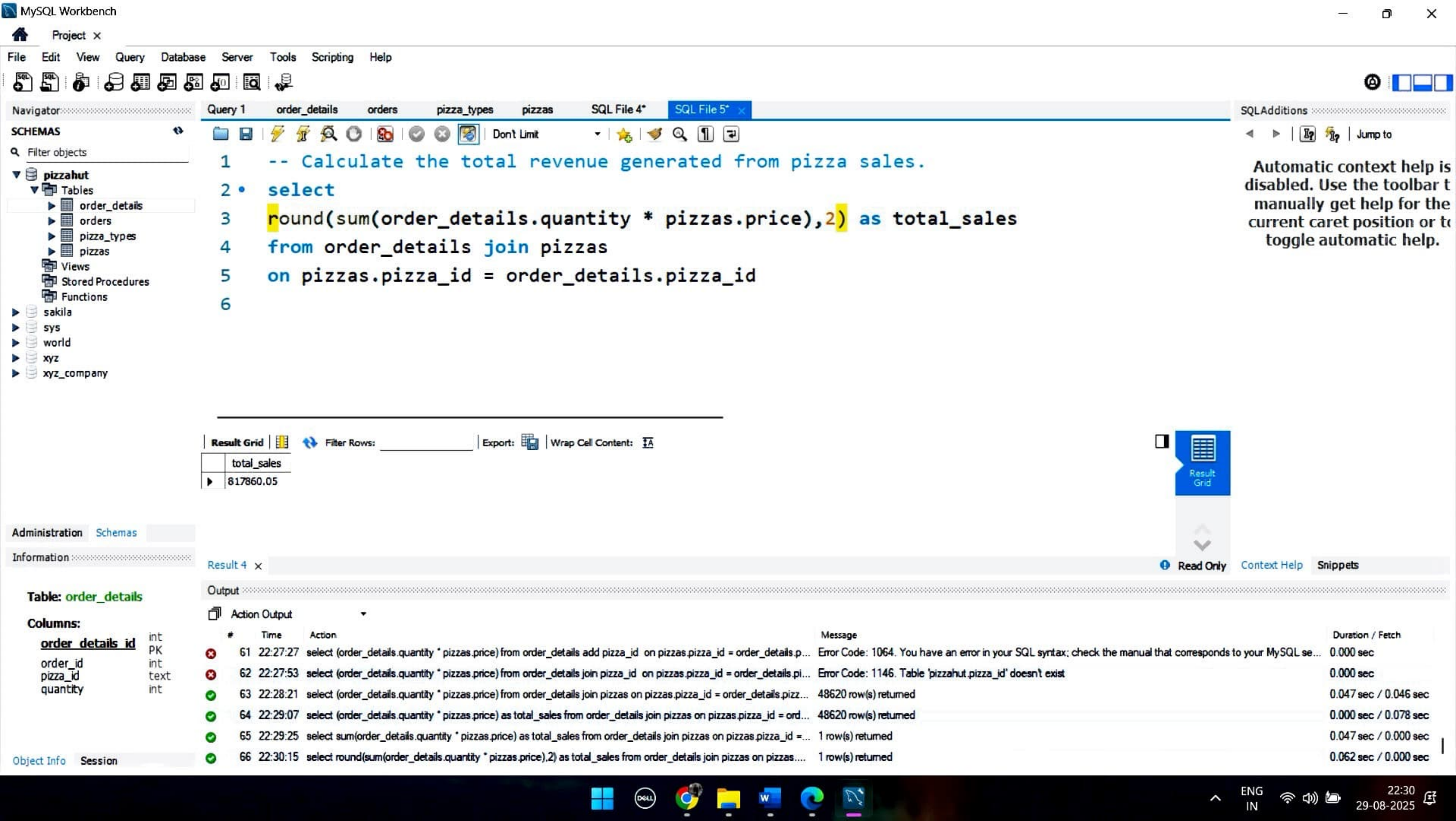
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Table: order_details

Columns:

order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

```
#   Time      Action                                        Message              Duration / Fetch
51  22:05:43  SELECT * FROM pizzahut.orders                 21350 row(s) returned  0.000 sec / 0.000 sec
52  22:05:45  SELECT * FROM pizzahut.pizza_types            32 row(s) returned     0.000 sec / 0.000 sec
53  22:05:47  SELECT * FROM pizzahut.pizzas                 96 row(s) returned     0.000 sec / 0.000 sec
54  22:08:51  SELECT count(order_id) FROM pizzahut.orders   1 row(s) returned      0.000 sec / 0.000 sec
55  22:09:01  SELECT count(order_id) FROM pizzahut.orders   1 row(s) returned      0.000 sec / 0.000 sec
56  22:10:25  SELECT count(order_id) FROM pizzahut.orders   1 row(s) returned      0.000 sec / 0.000 sec
```



```sql
-- 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
```

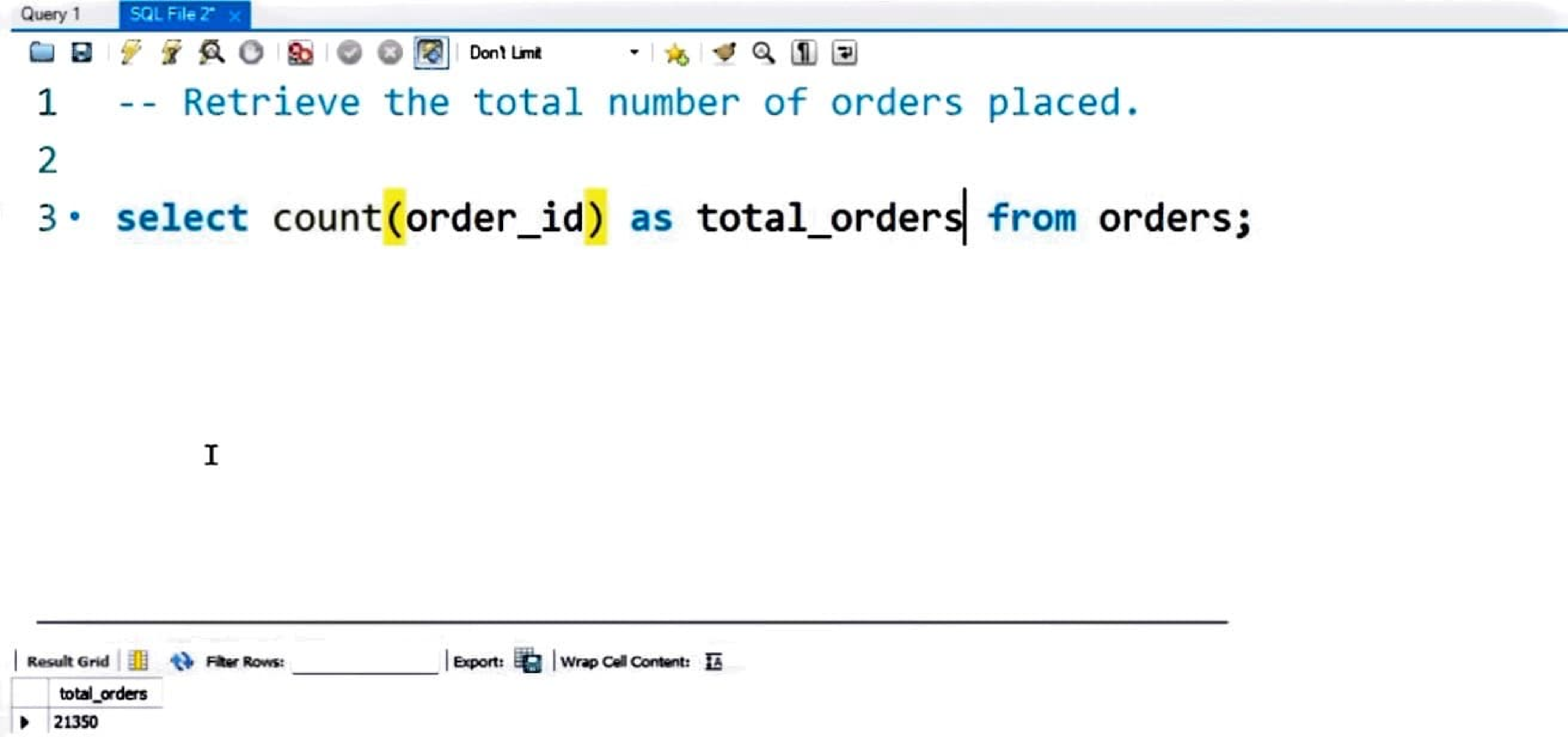
total_sales
817860.05

Table: order_details

Columns:

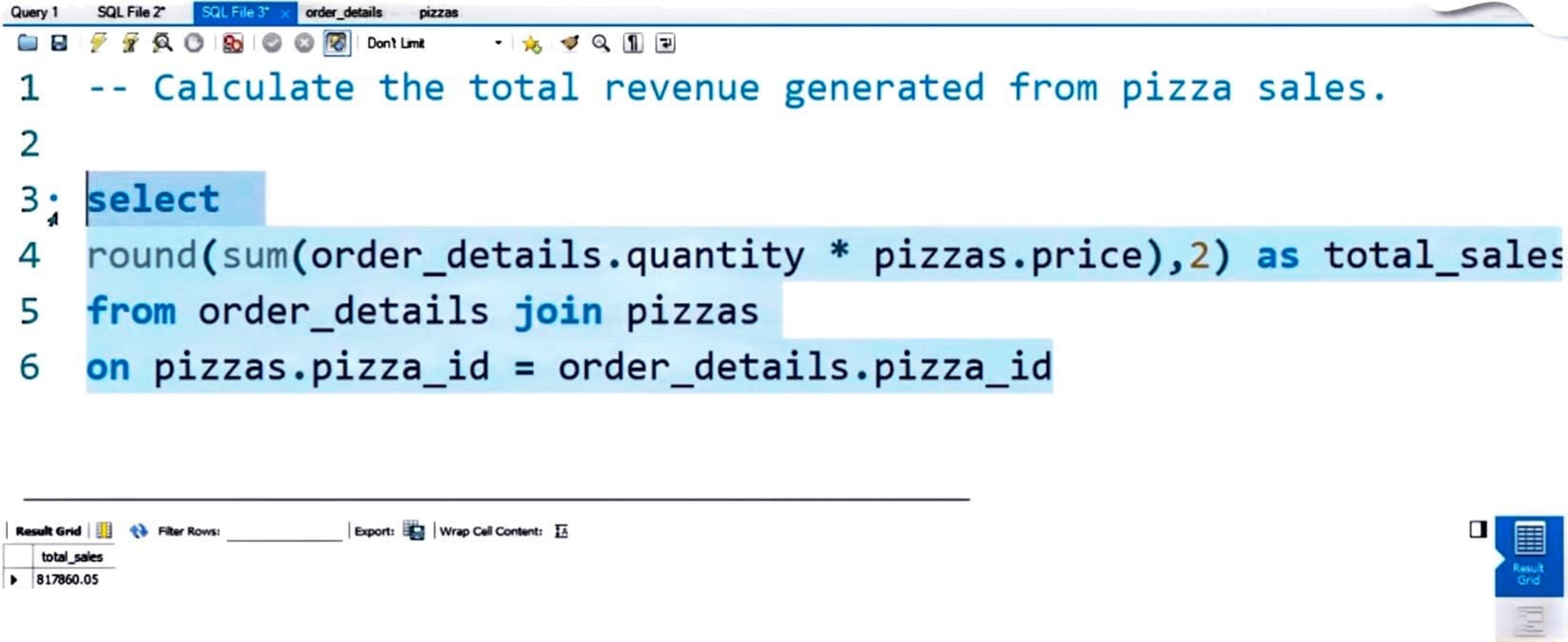
order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

#	Time	Action	Message	Duration / Fetch
61	22:27:27	select (order_details.quantity * pizzas.price) from order_details add pizza_id on pizzas.pizza_id = order_details.p...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
62	22:27:53	select (order_details.quantity * pizzas.price) from order_details join pizza_id on pizzas.pizza_id = order_details.pi...	Error Code: 1146. Table 'pizzahut.pizza_id' doesn't exist	0.000 sec
63	22:28:21	select (order_details.quantity * pizzas.price) from order_details join pizzas on pizzas.pizza_id = order_details.pizz...	48620 row(s) returned	0.047 sec / 0.046 sec
64	22:29:07	select (order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id = ord...	48620 row(s) returned	0.000 sec / 0.078 sec
65	22:29:25	select sum(order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id =...	1 row(s) returned	0.047 sec / 0.000 sec
66	22:30:15	select round(sum(order_details.quantity * pizzas.price),2) as total_sales from order_details join pizzas on pizzas....	1 row(s) returned	0.062 sec / 0.000 sec



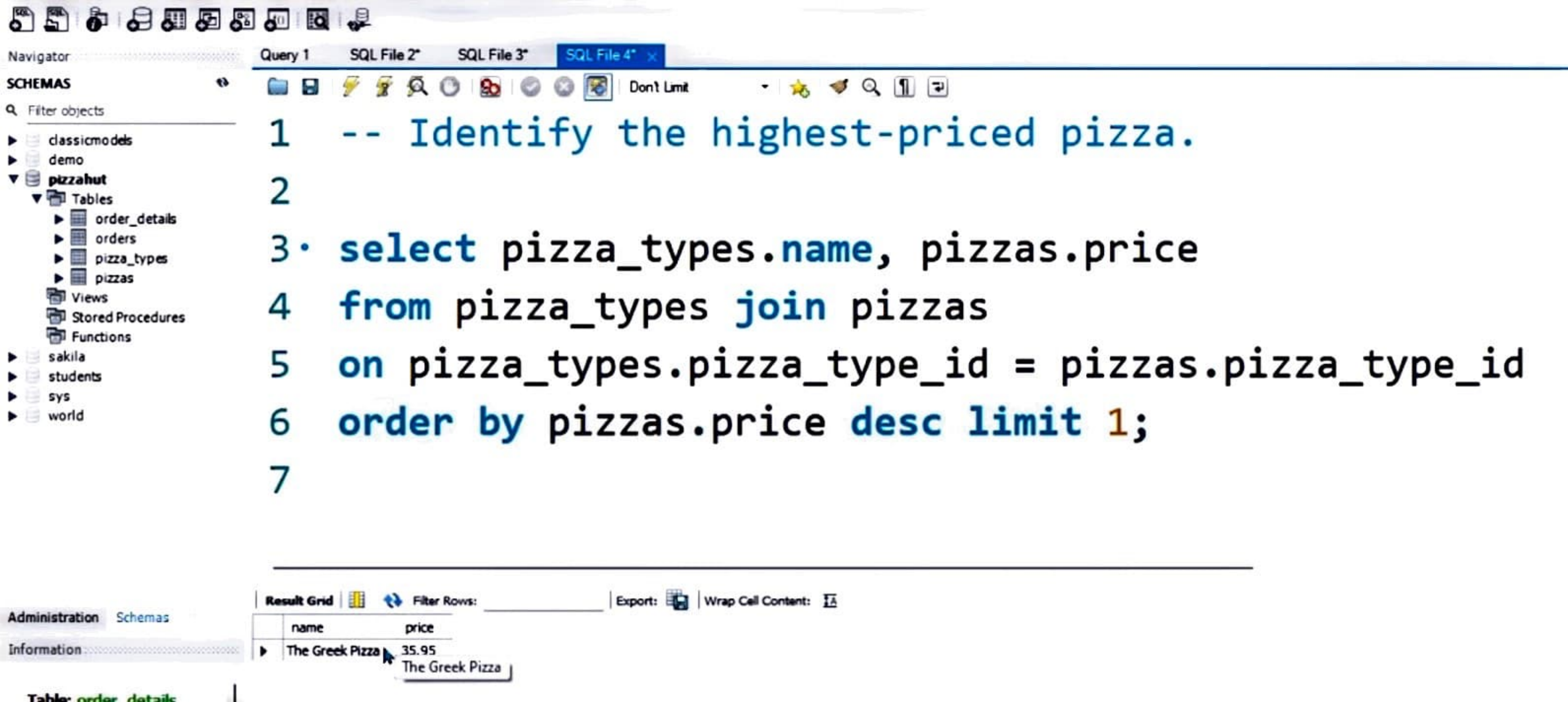
```sql
-- Retrieve the total number of orders placed.

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

```
total_orders
21350
```



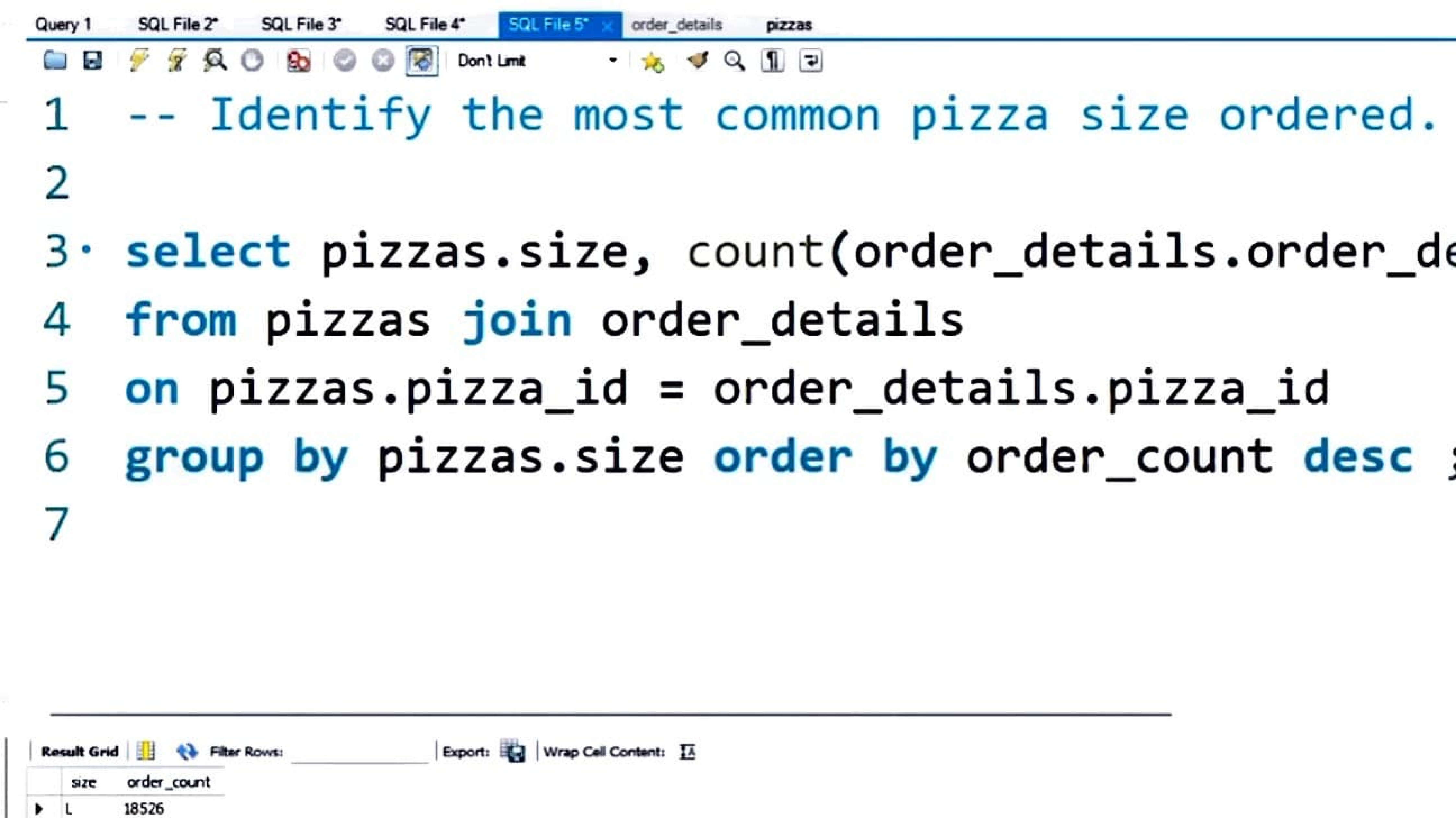
```sql
-- Calculate the total revenue generated from pizza sales.

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
```



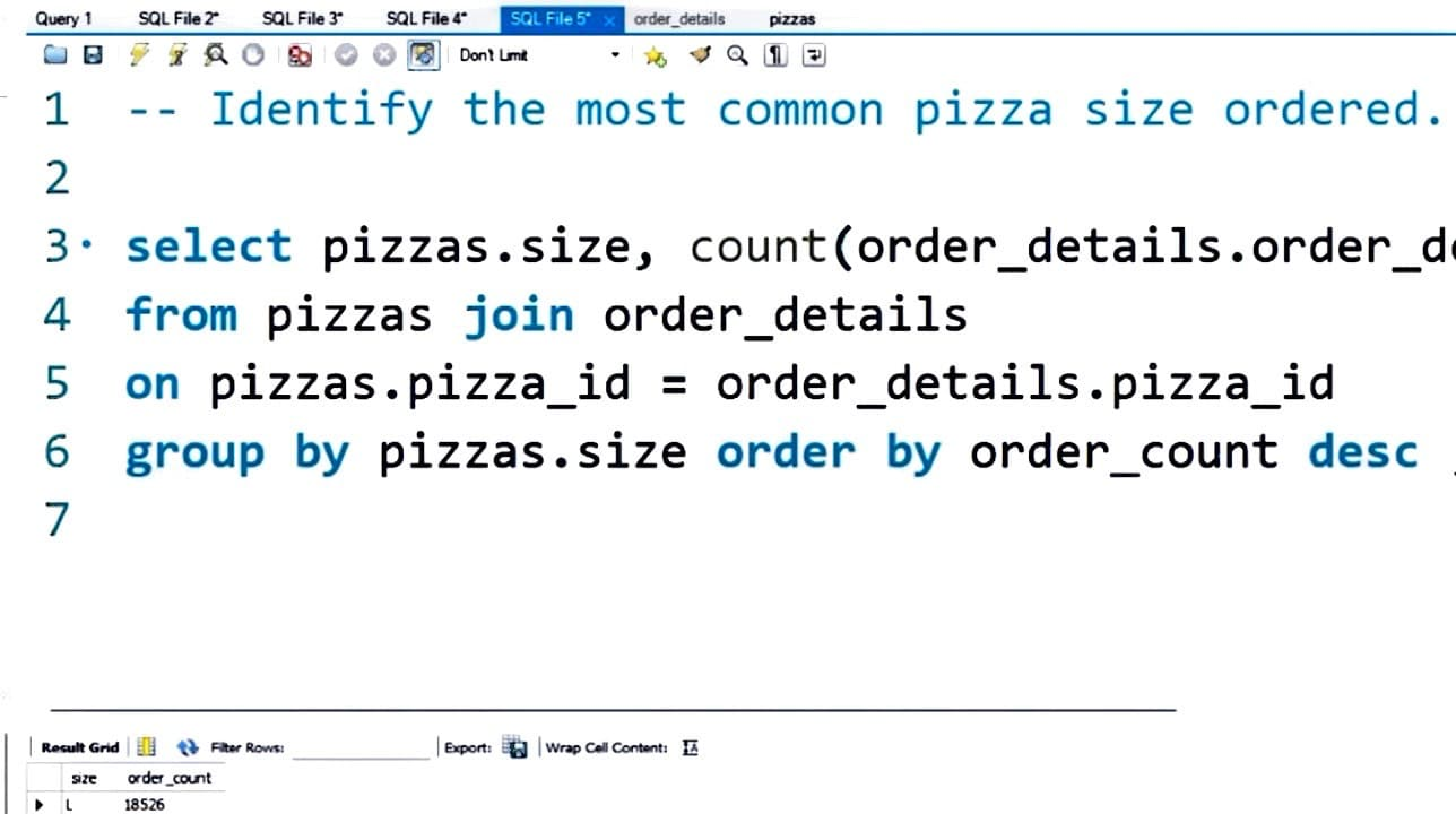
```sql
-- Identify the highest-priced pizza.

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



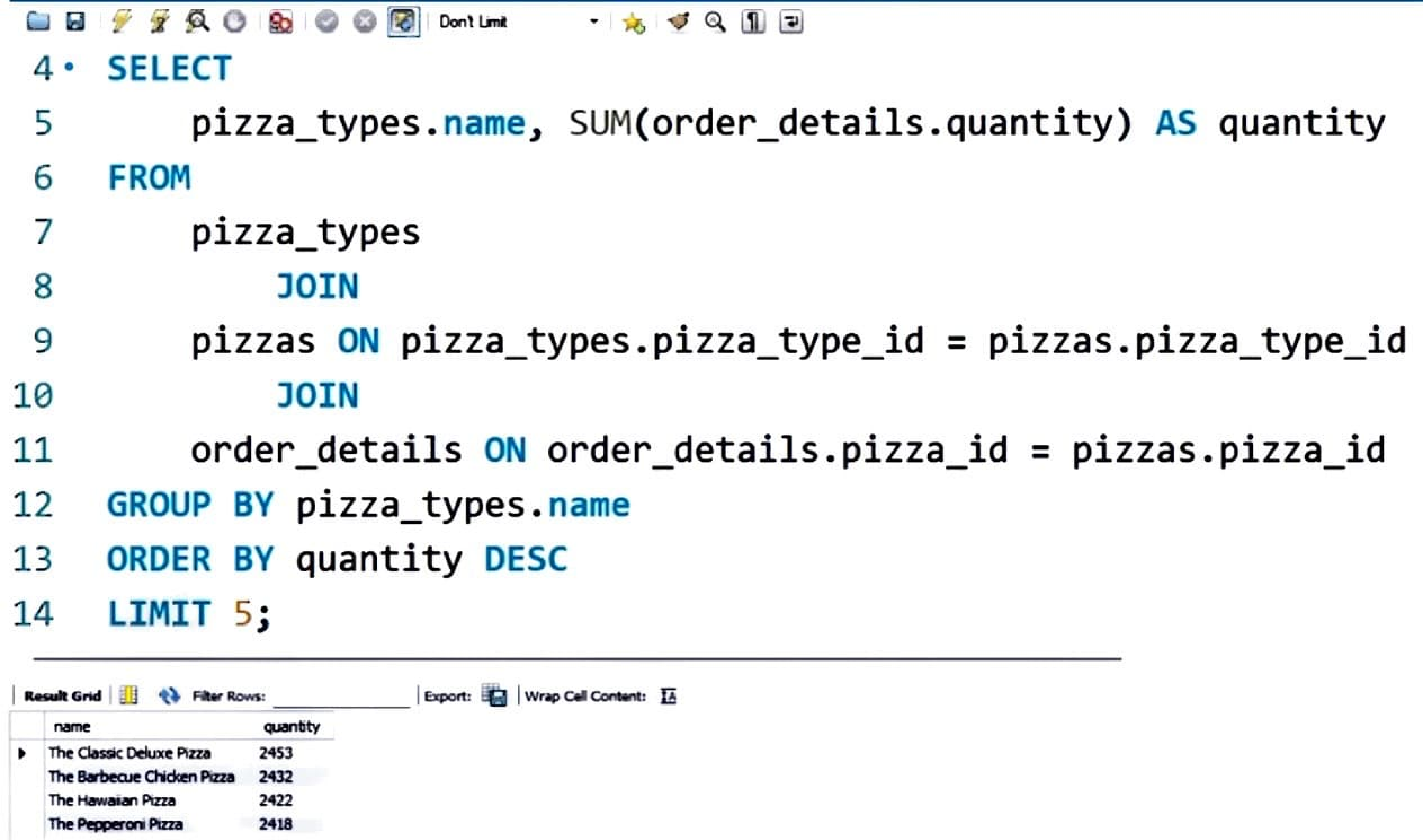
```
-- Identify the most common pizza size ordered.

select pizzas.size, count(order_details.order_de
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



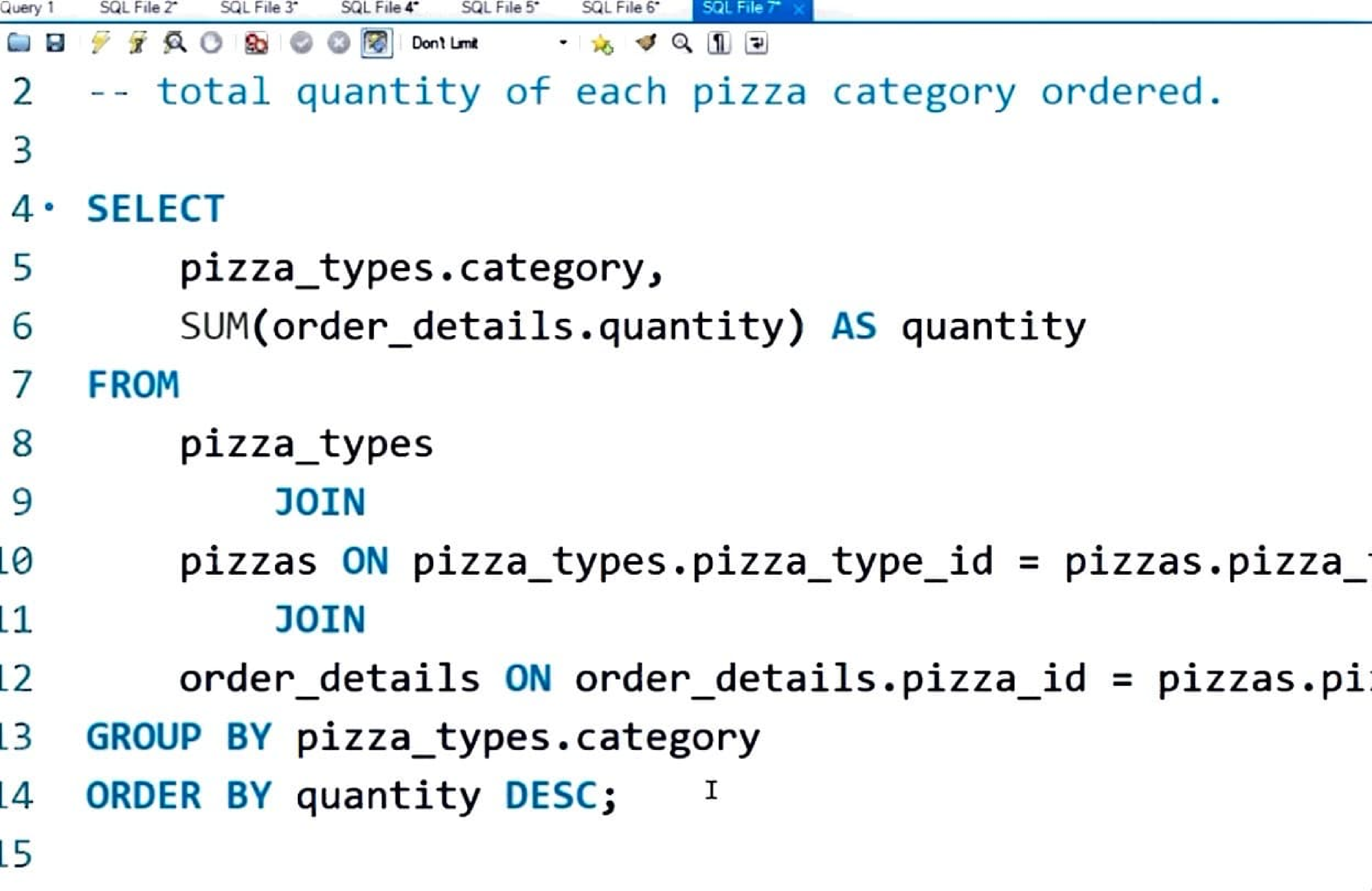
```
-- Identify the most common pizza size ordered.

select pizzas.size, count(order_details.order_de
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
```

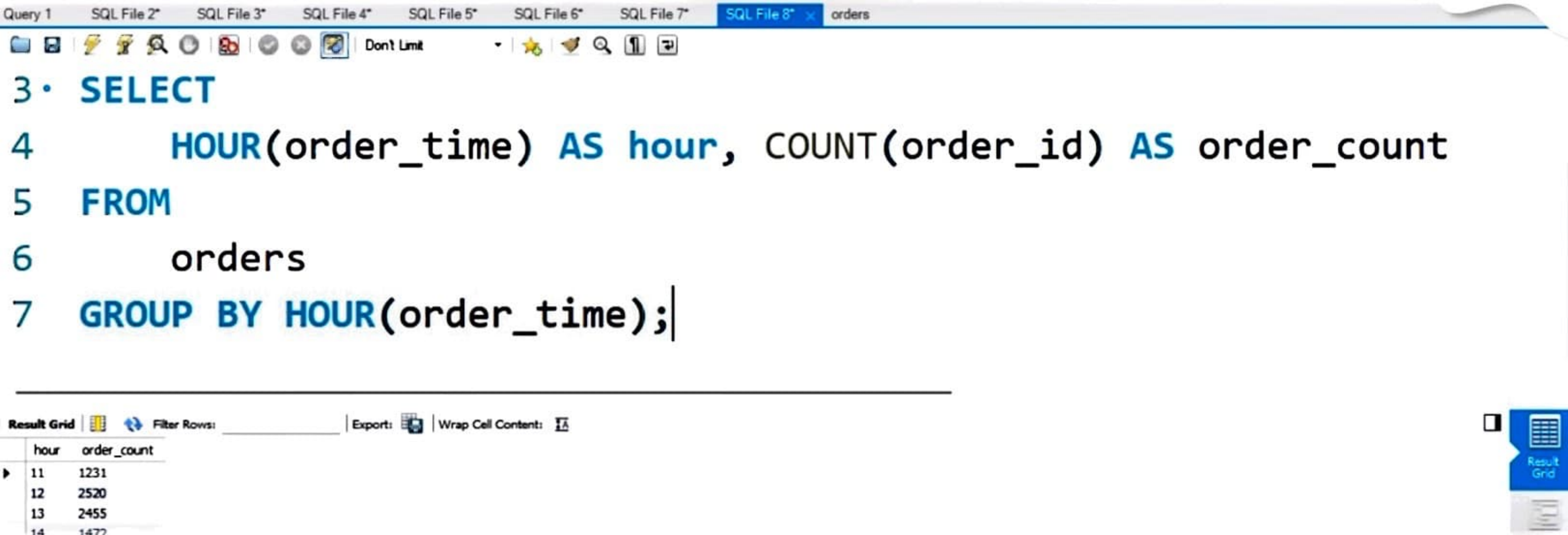


```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

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

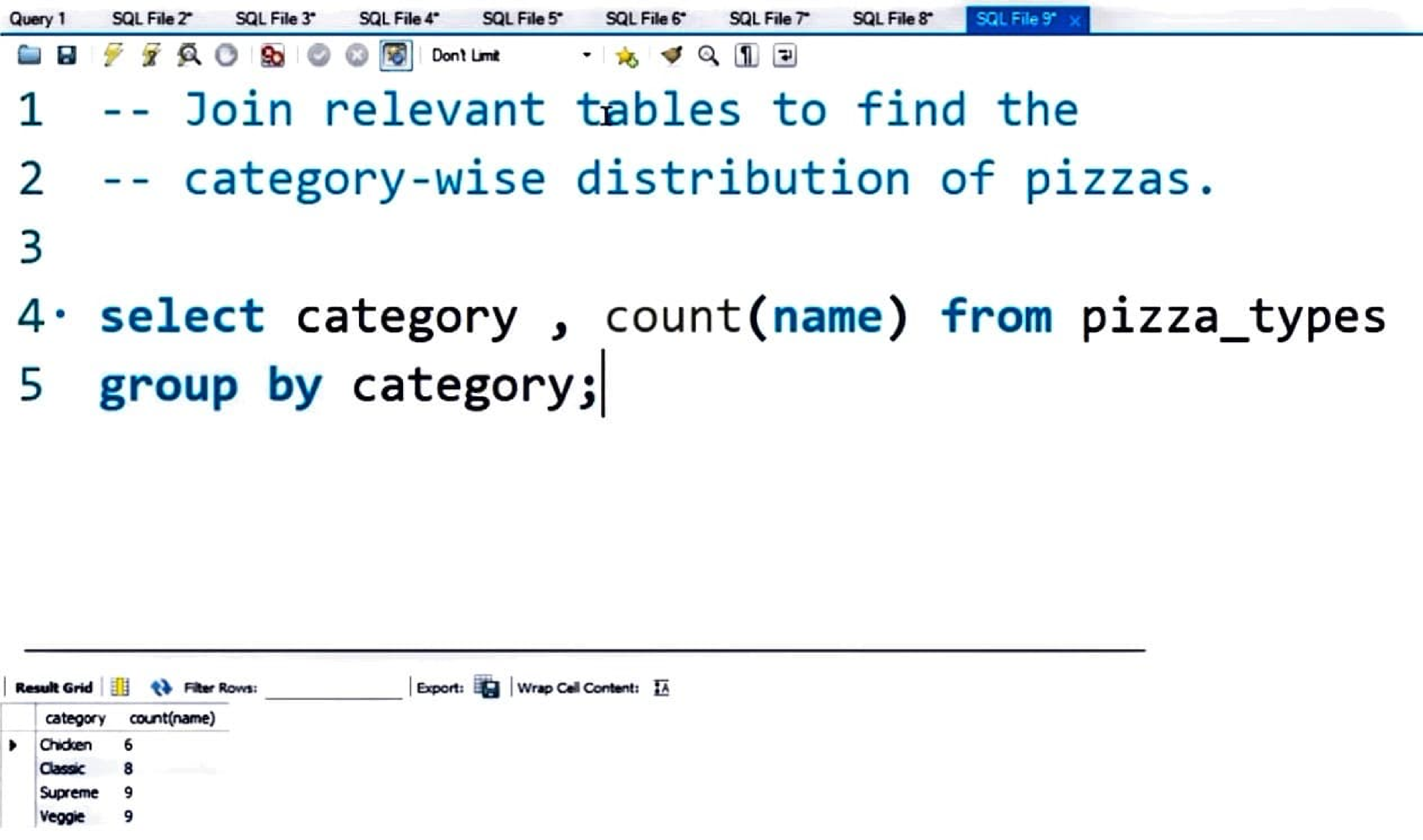


```sql
-- total quantity of each pizza category ordered.

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_
        JOIN
    order_details ON order_details.pizza_id = pizzas.pi
GROUP BY pizza_types.category
ORDER BY quantity DESC;

```



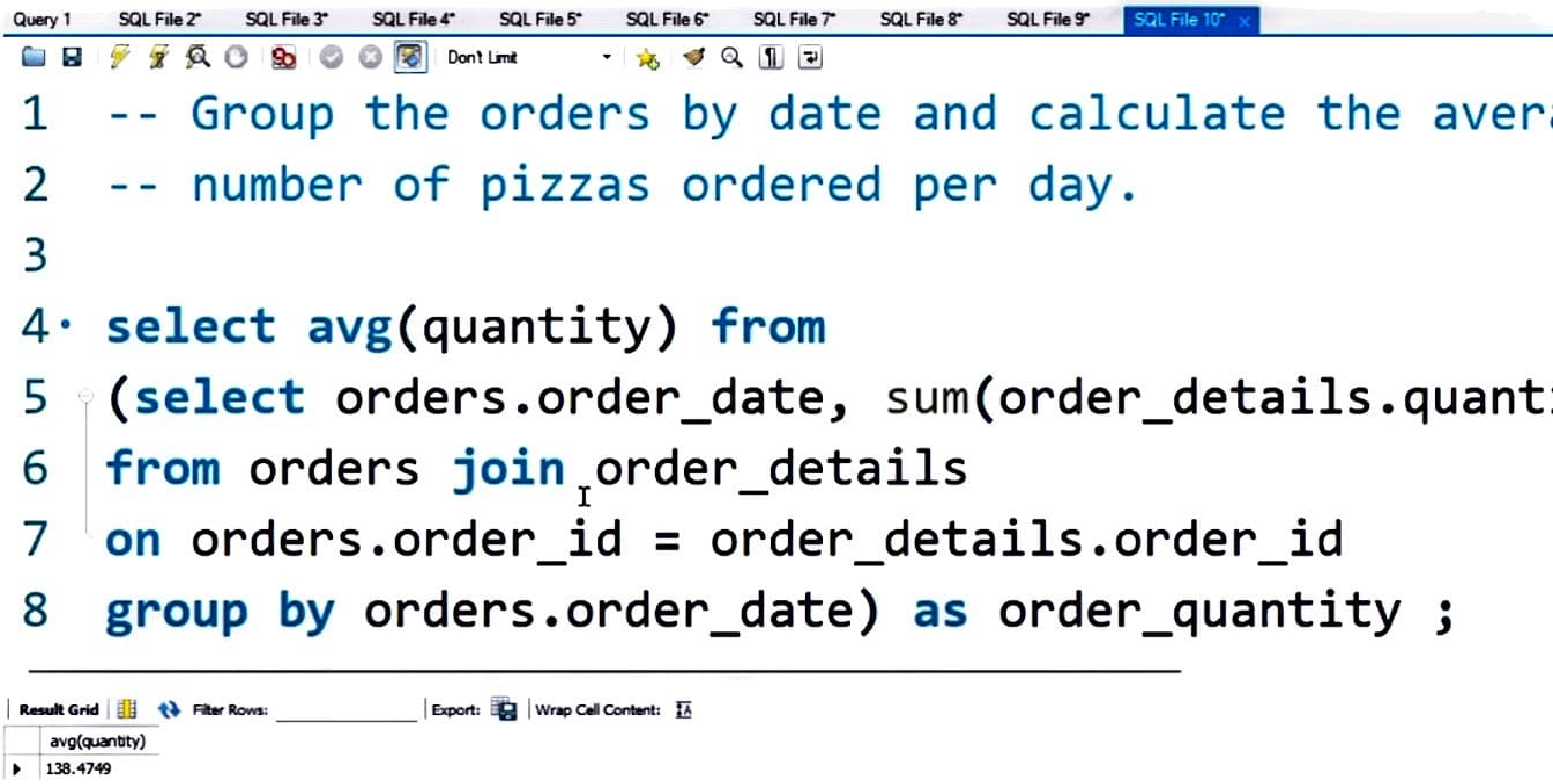
```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

```
hour  order_count
11    1231
12    2520
13    2455
14    1472
```



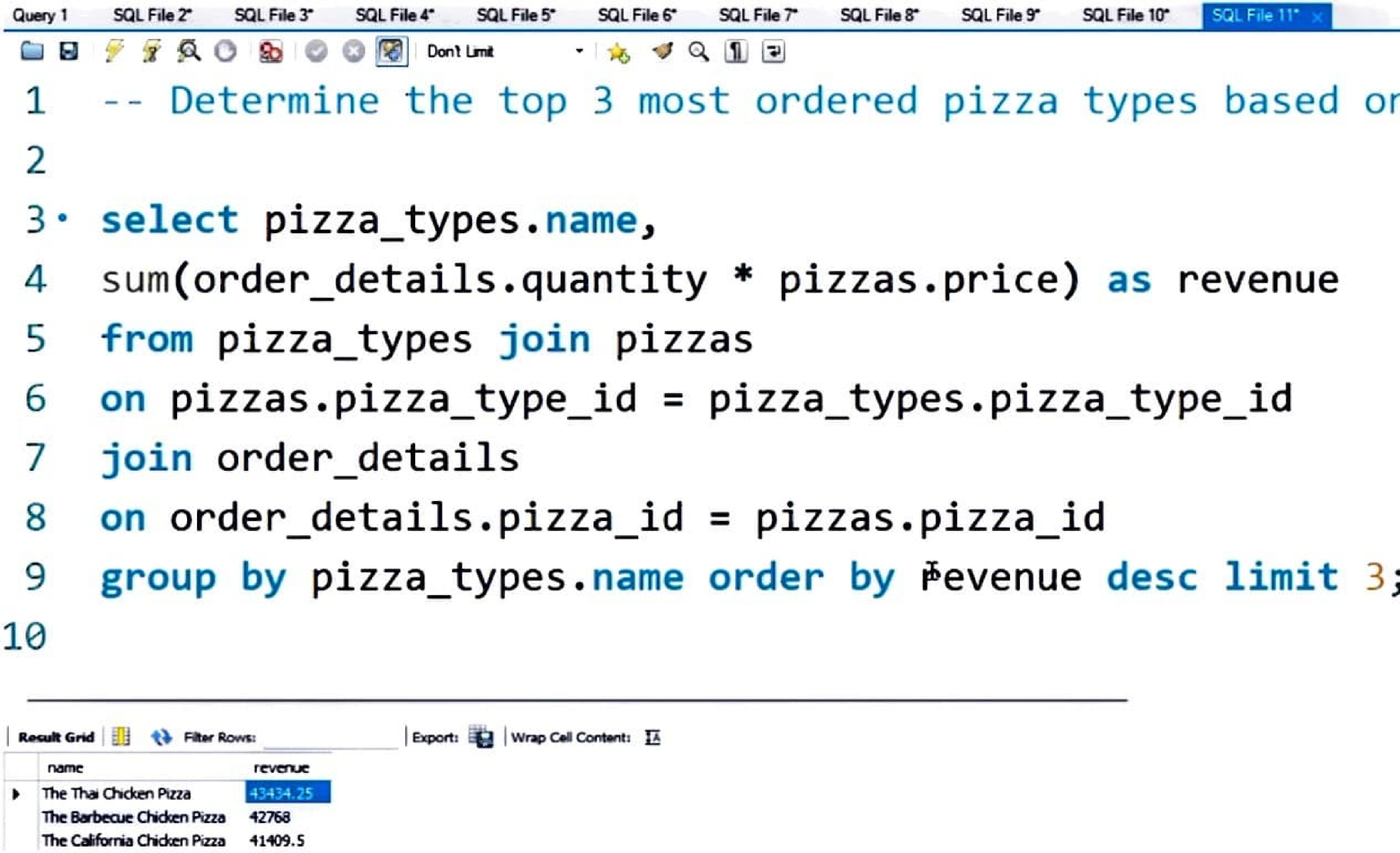
```sql
-- Join relevant tables to find the
-- category-wise distribution of pizzas.

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

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



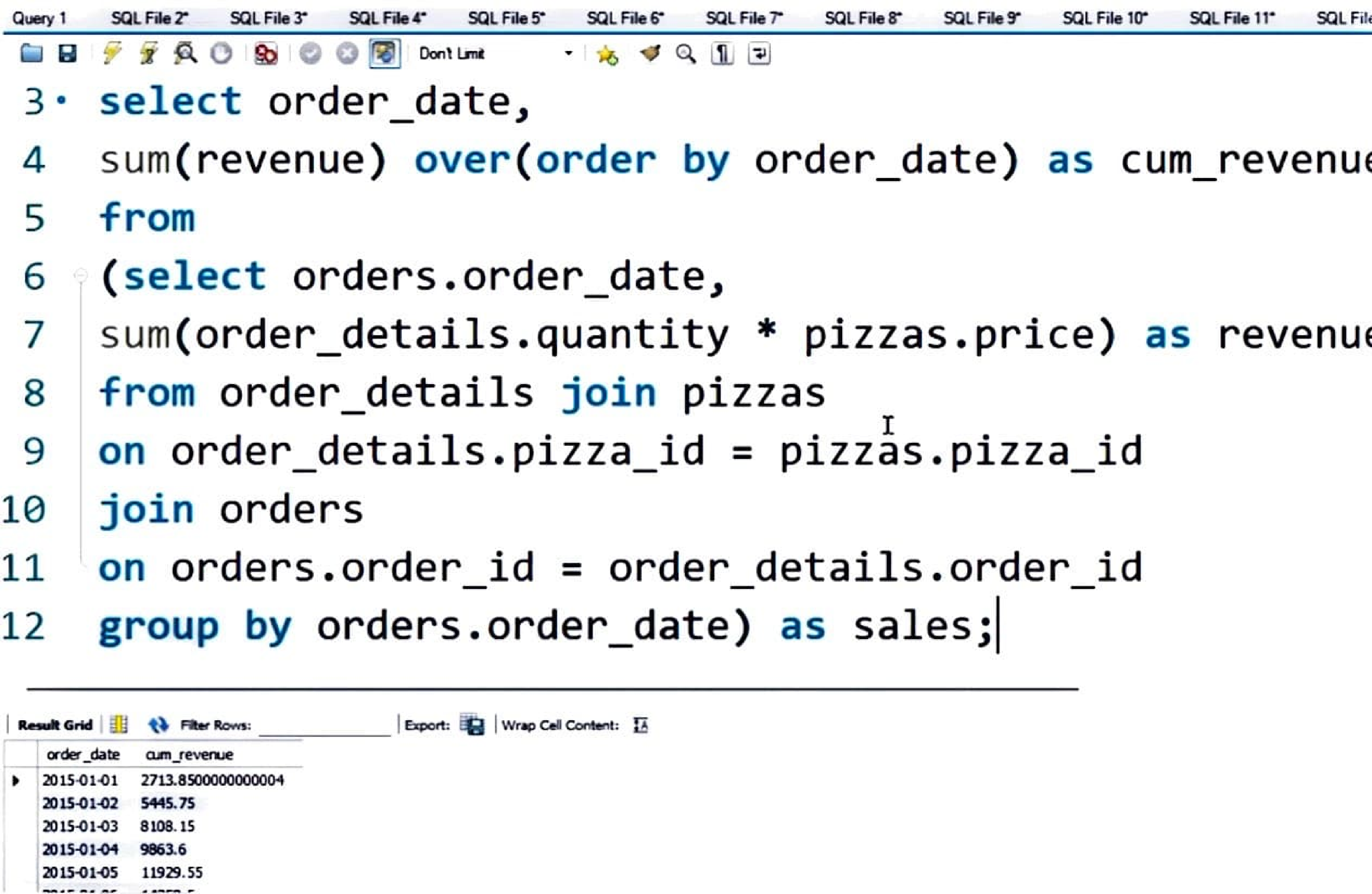
```
-- Group the orders by date and calculate the avera
-- number of pizzas ordered per day.

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

```
avg(quantity)
138.4749
```



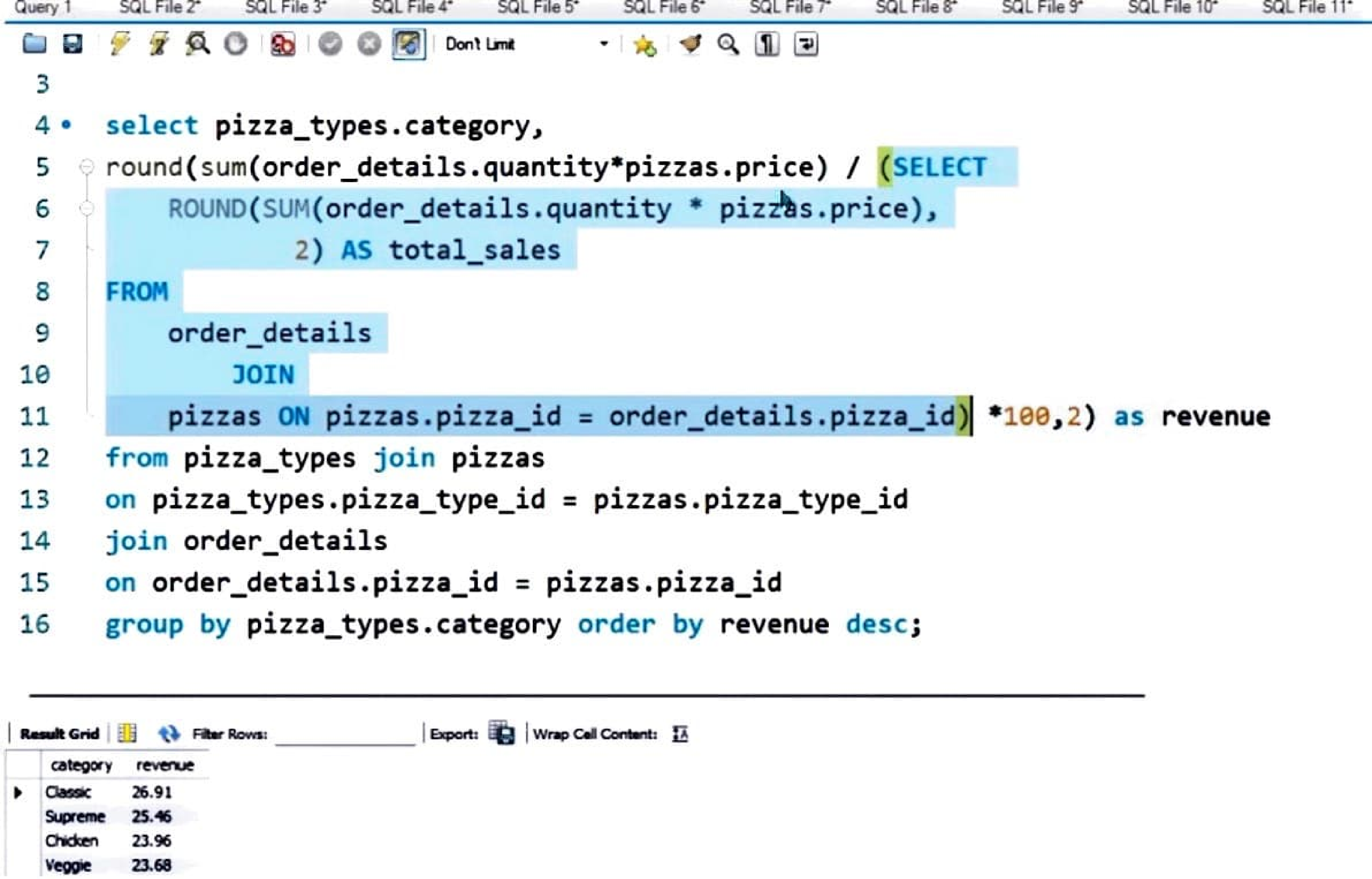
```sql
-- Determine the top 3 most ordered pizza types based o

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



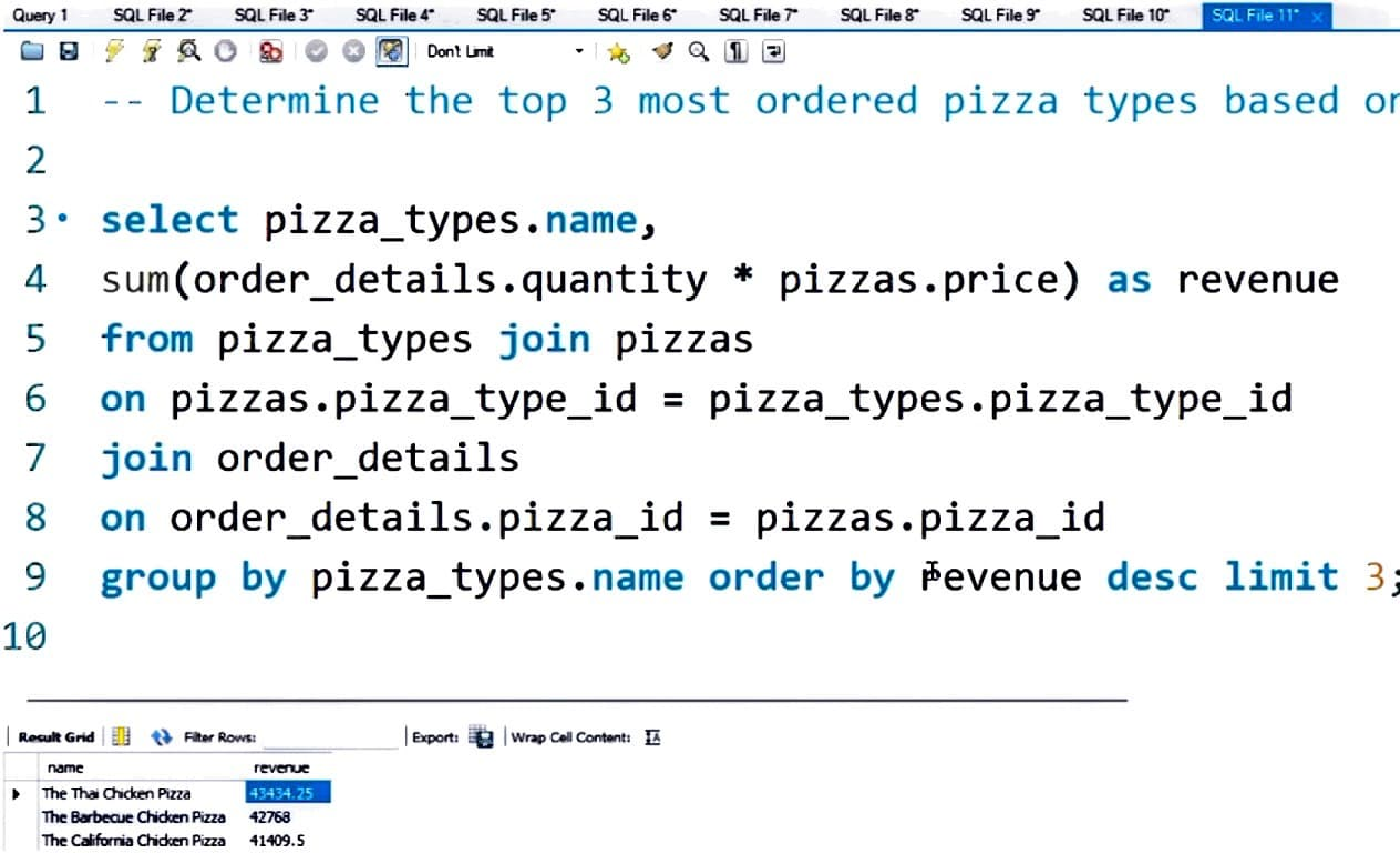
```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```

```
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
```



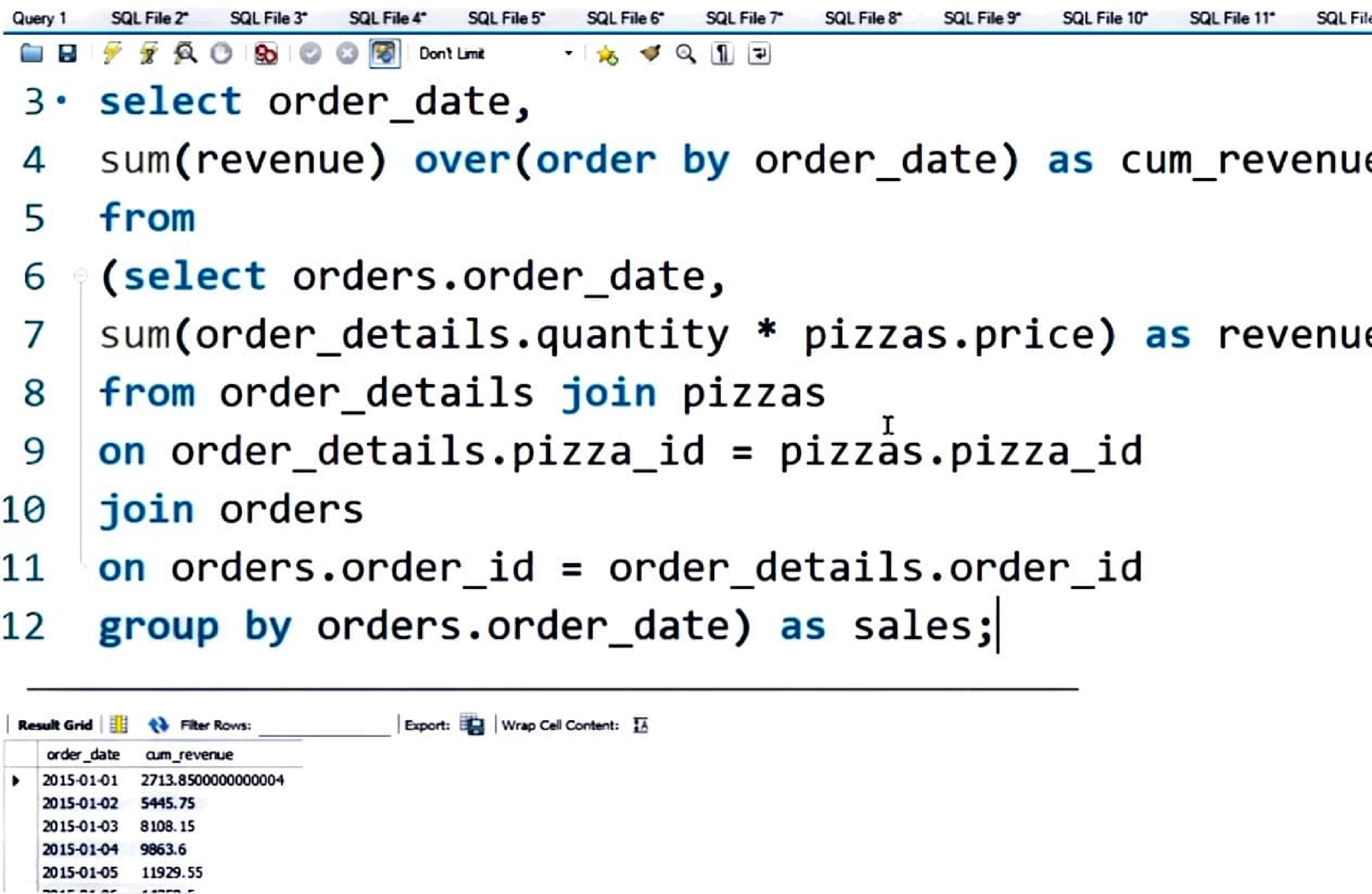
```sql

select pizza_types.category,
round(sum(order_details.quantity*pizzas.price) / (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) *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;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



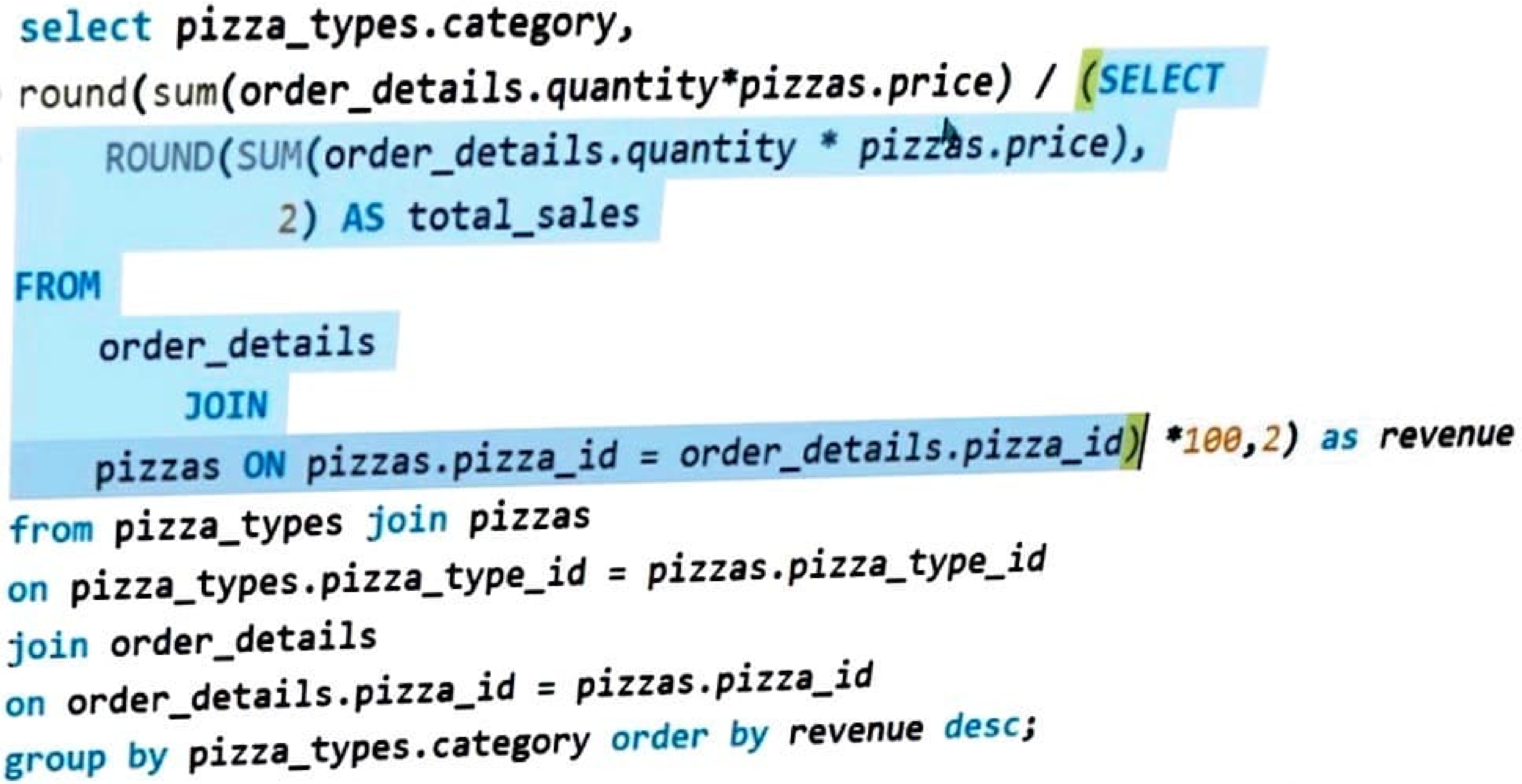
```sql
-- Determine the top 3 most ordered pizza types based or

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



```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 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;
```

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



```
select pizza_types.category,
round(sum(order_details.quantity*pizzas.price) / (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) *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;
```