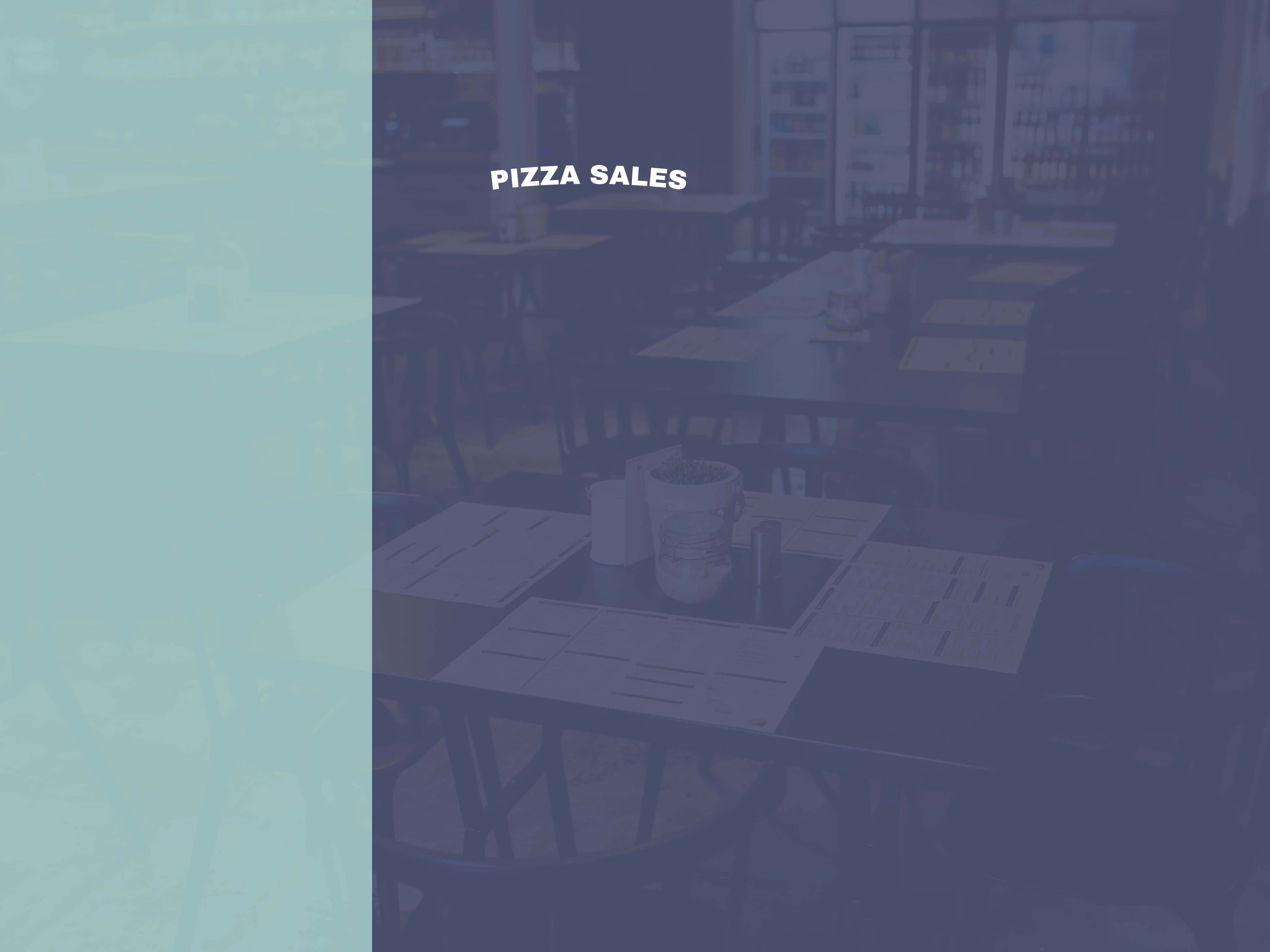
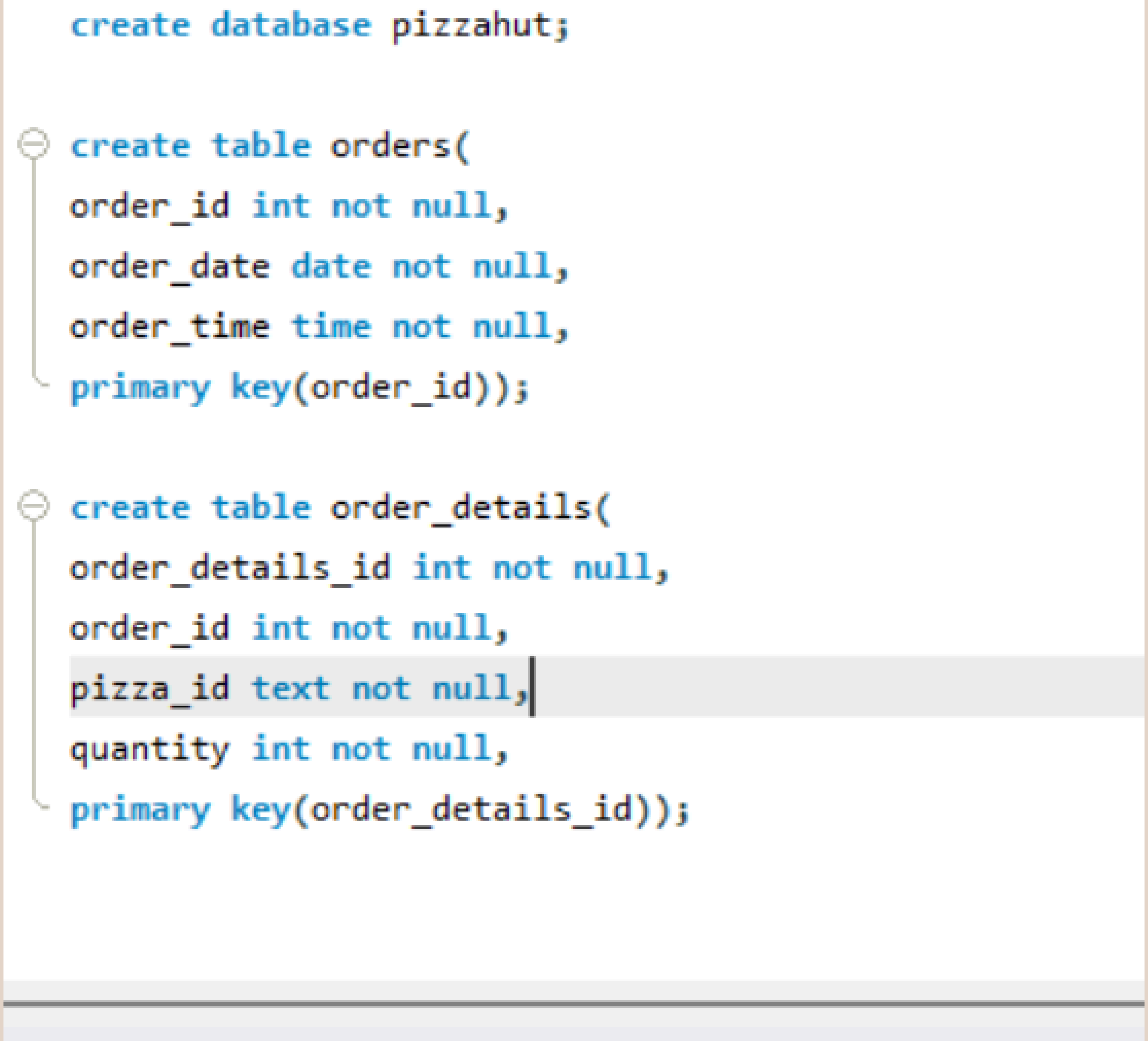
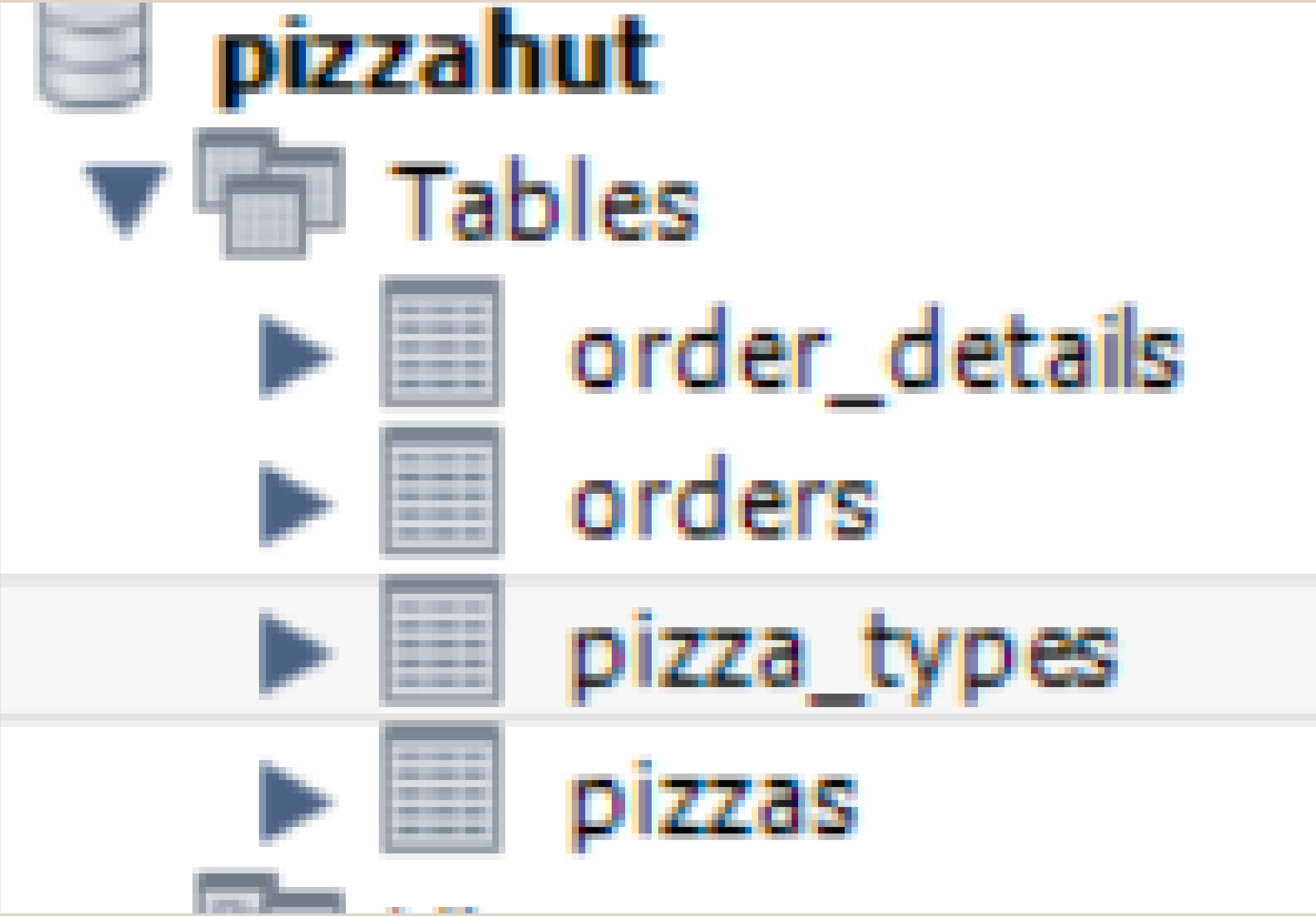
PIZZA SALES

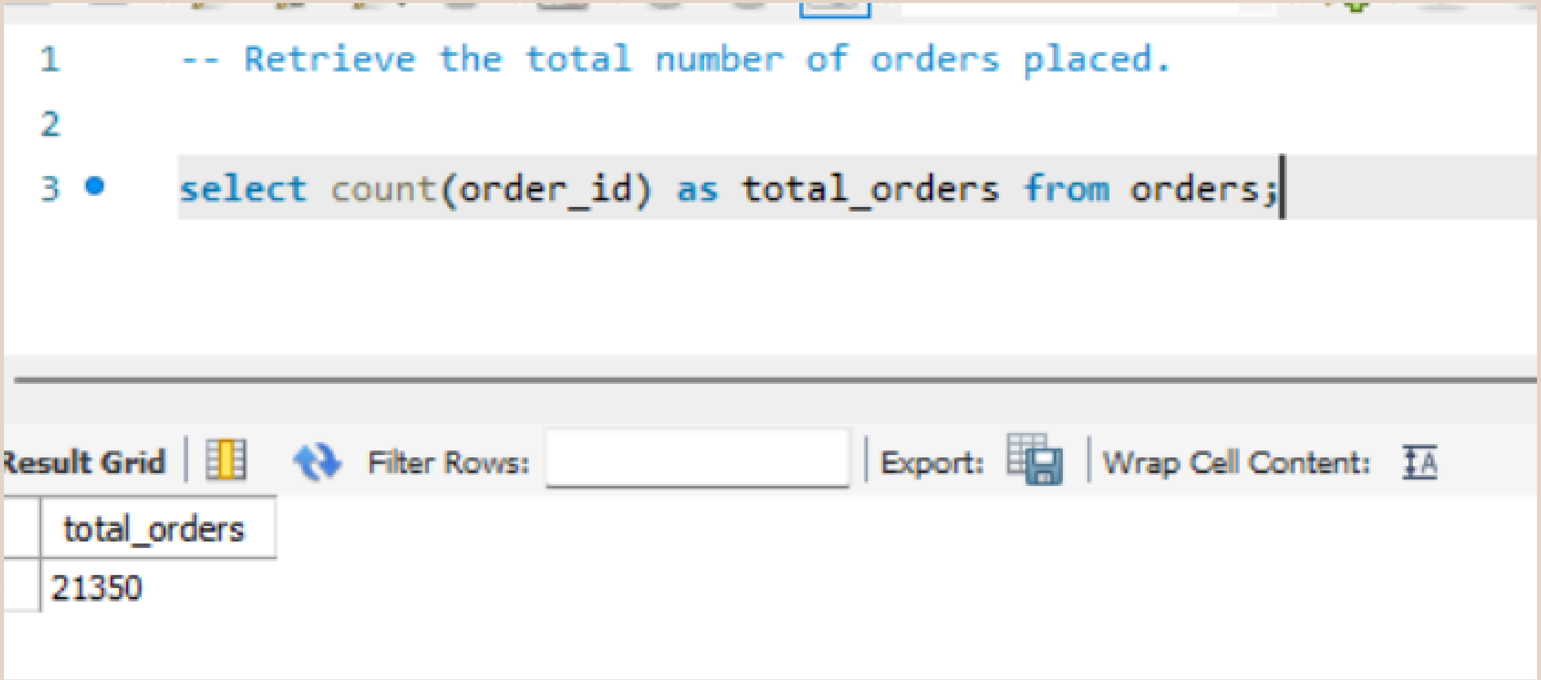




```sql
create database pizzahut;

create table orders(
order_id int not null,
order_date date not null,
order_time time not null,
primary key(order_id));

create table order_details(
order_details_id int not null,
order_id int not null,
pizza_id text not null,
quantity int not null,
primary key(order_details_id));
```



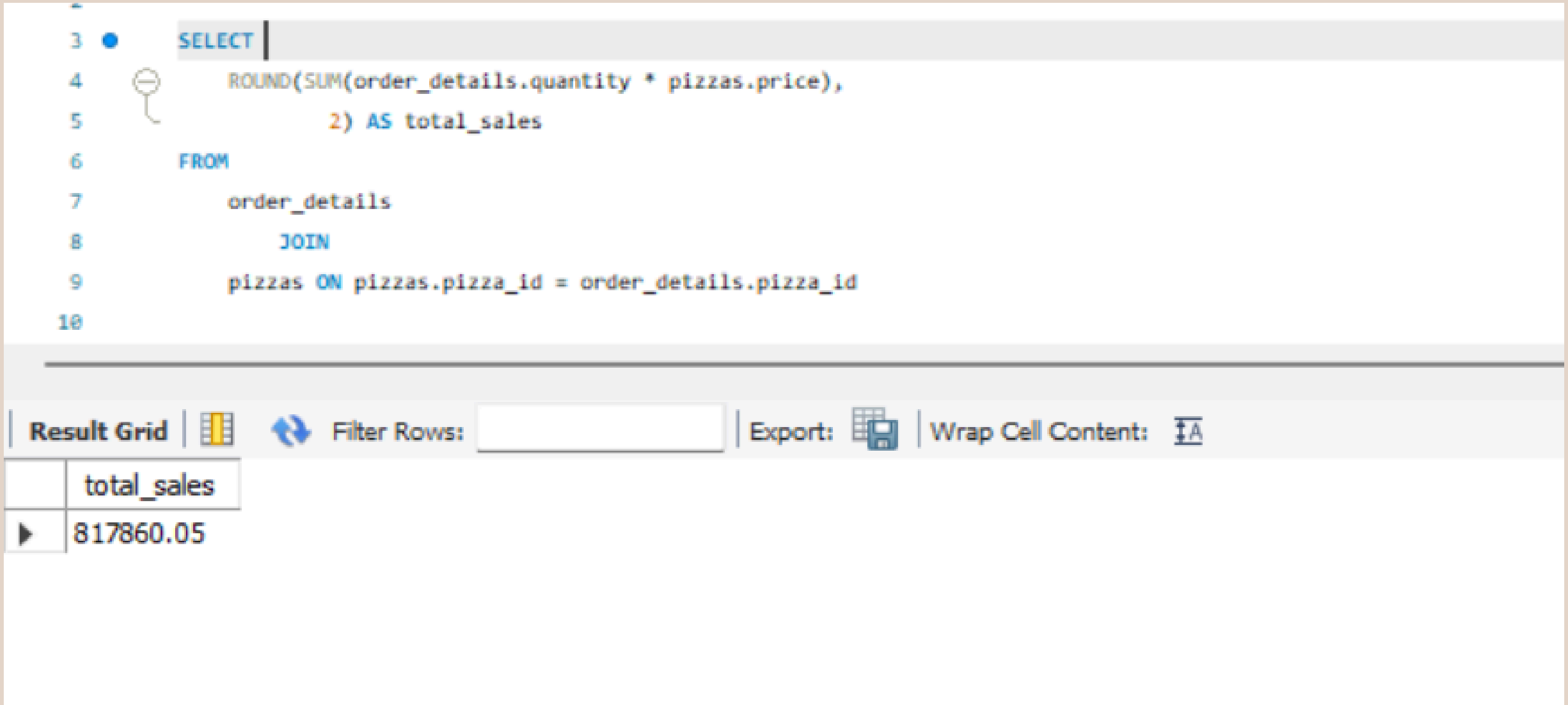
pizzahut
Tables
order_details
orders
pizza_types
pizzas



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

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

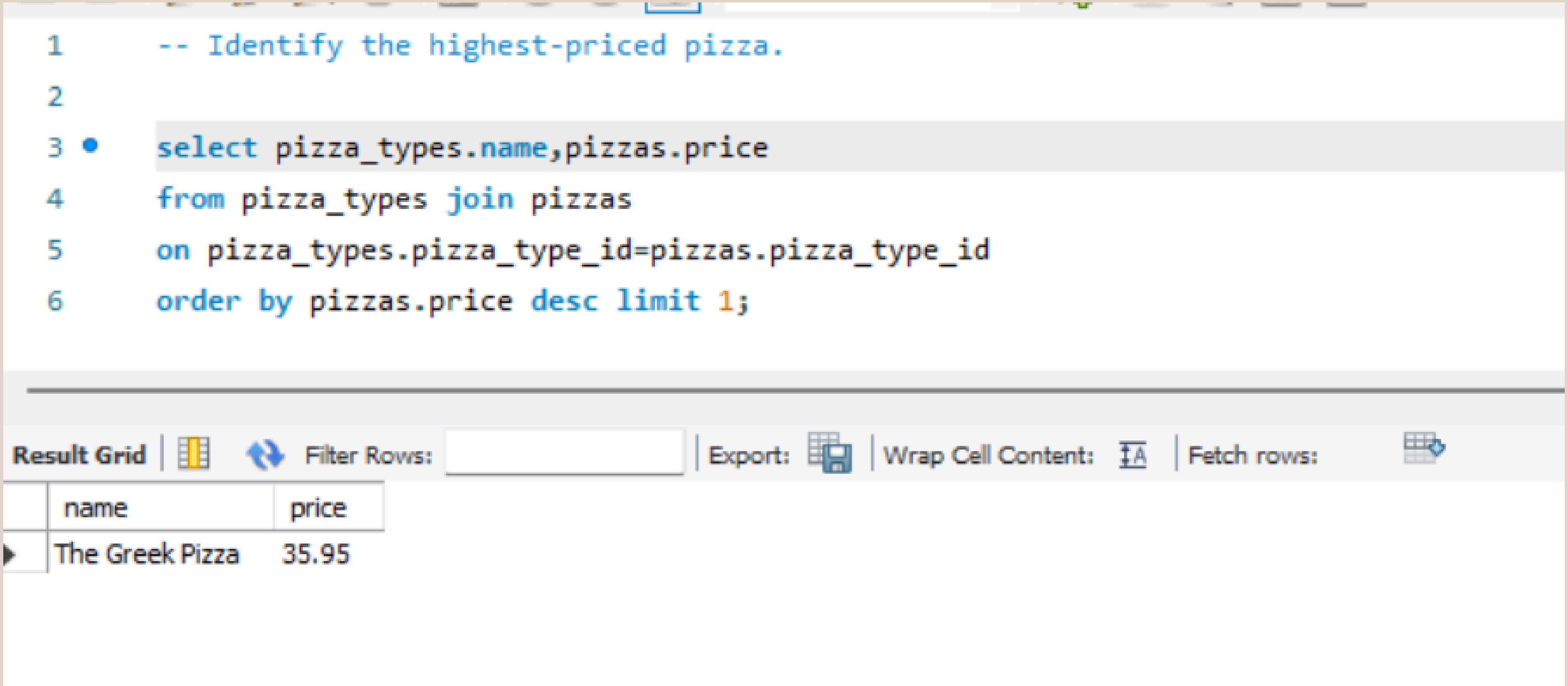
```
total_orders
21350
```

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



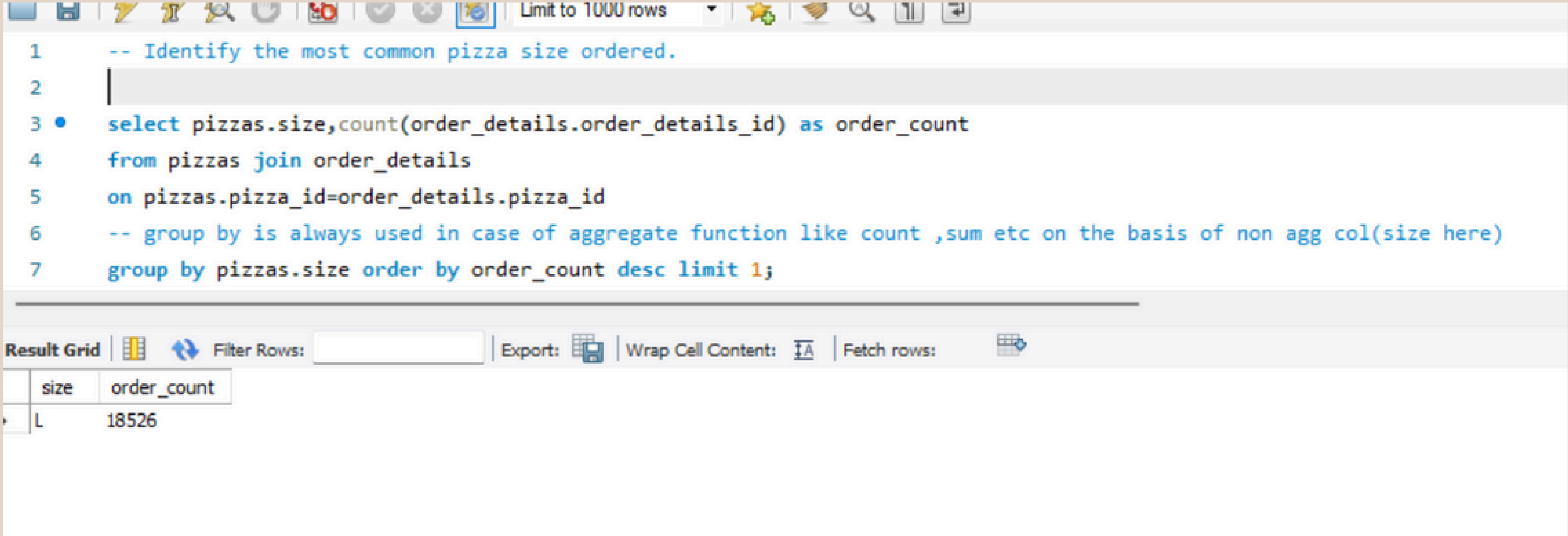
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

```
total_sales
817860.05
```



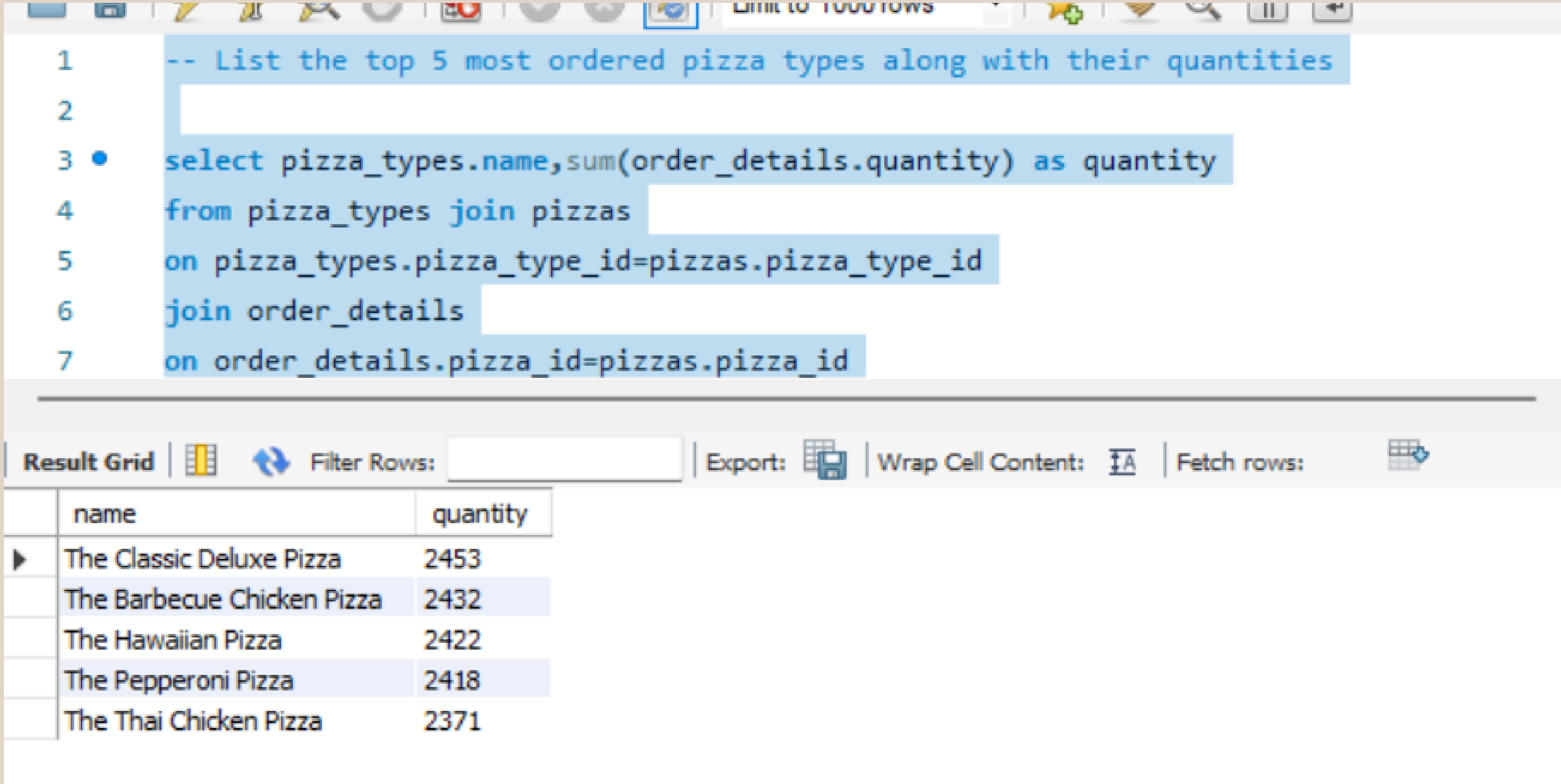
```sql
-- Identify the highest-priced pizza.

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



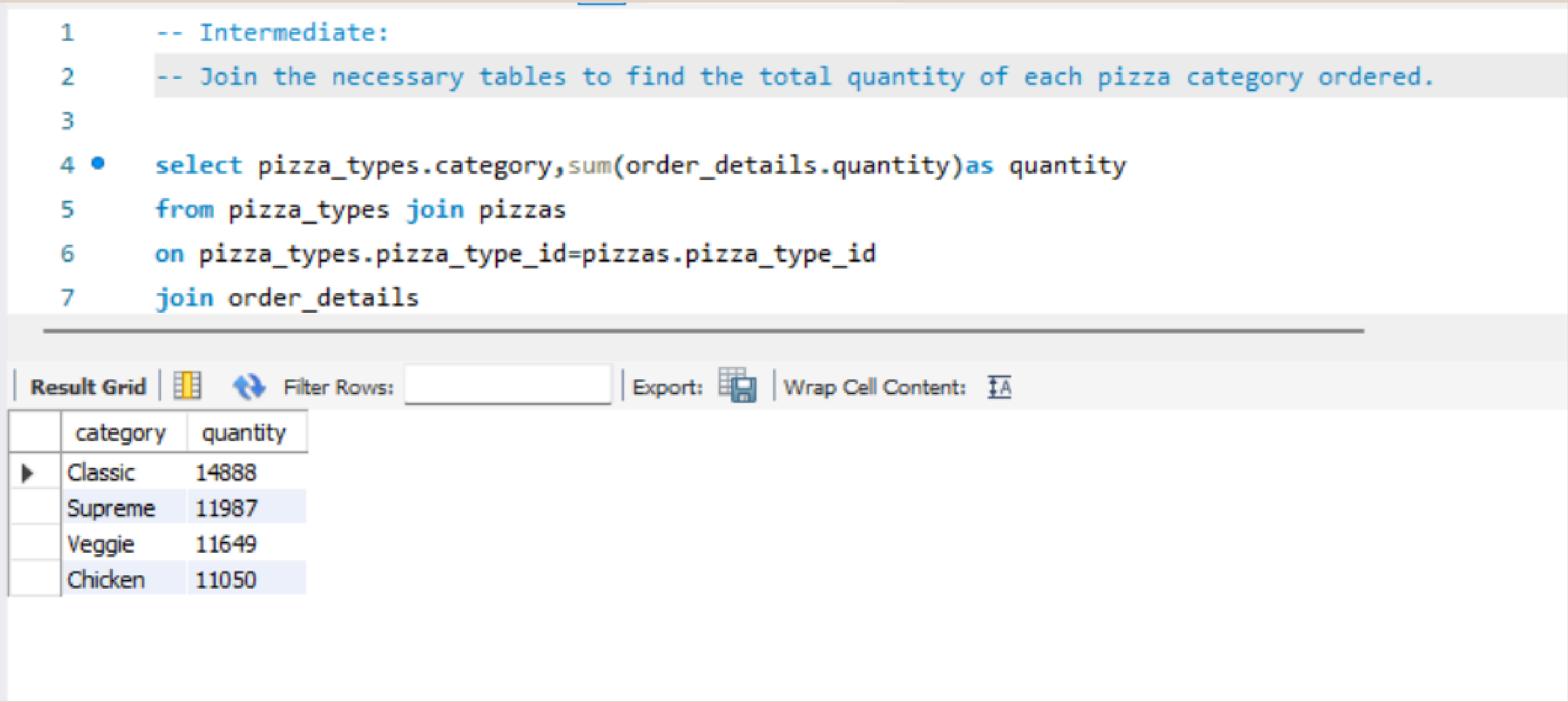
```sql
-- Identify the most common pizza size ordered.

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 is always used in case of aggregate function like count ,sum etc on the basis of non agg col(size here)
group by pizzas.size order by order_count desc limit 1;
```

```
size   order_count
L      18526
```



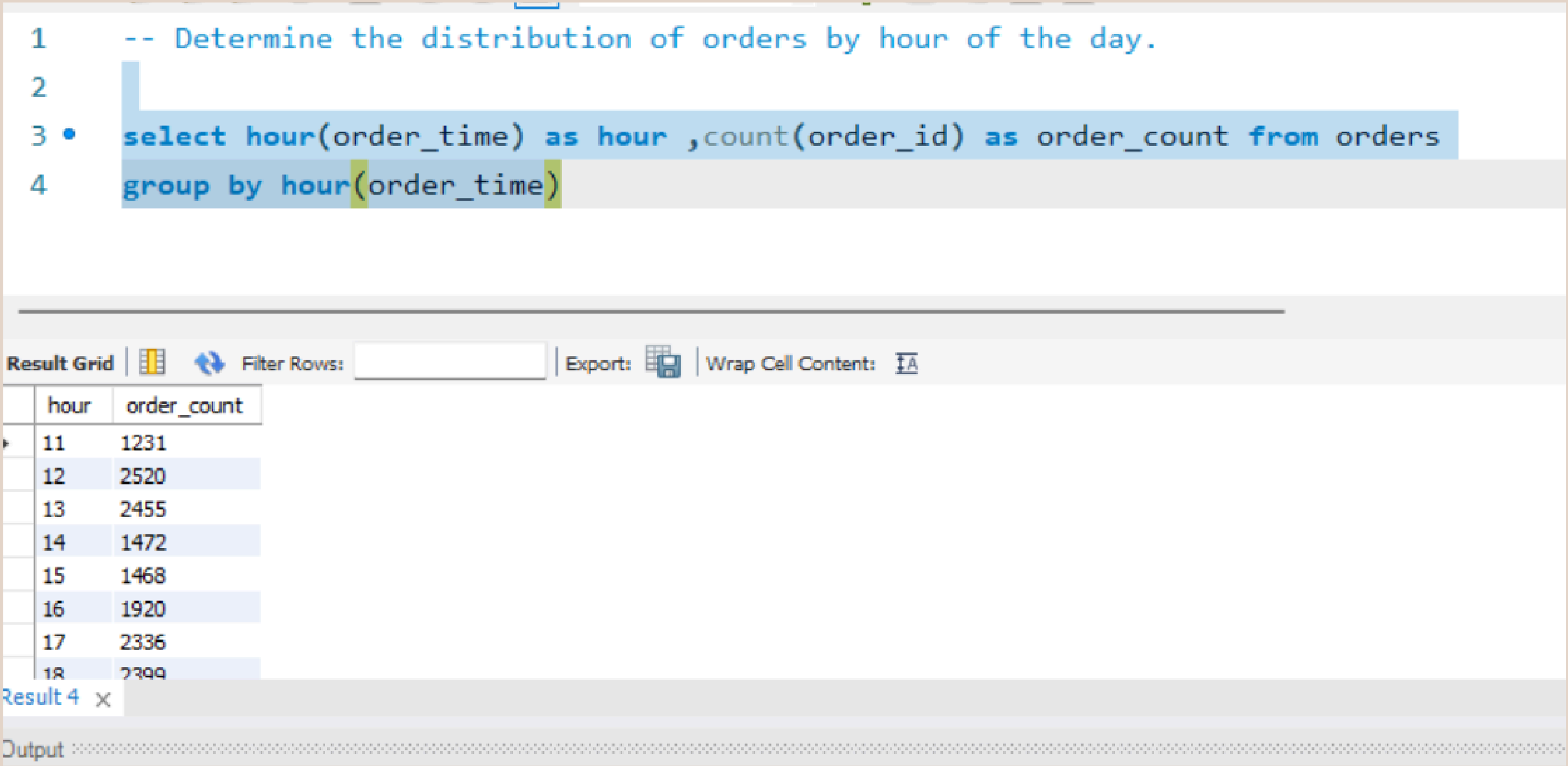
```sql
-- List the top 5 most ordered pizza types along with their quantities

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
```

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



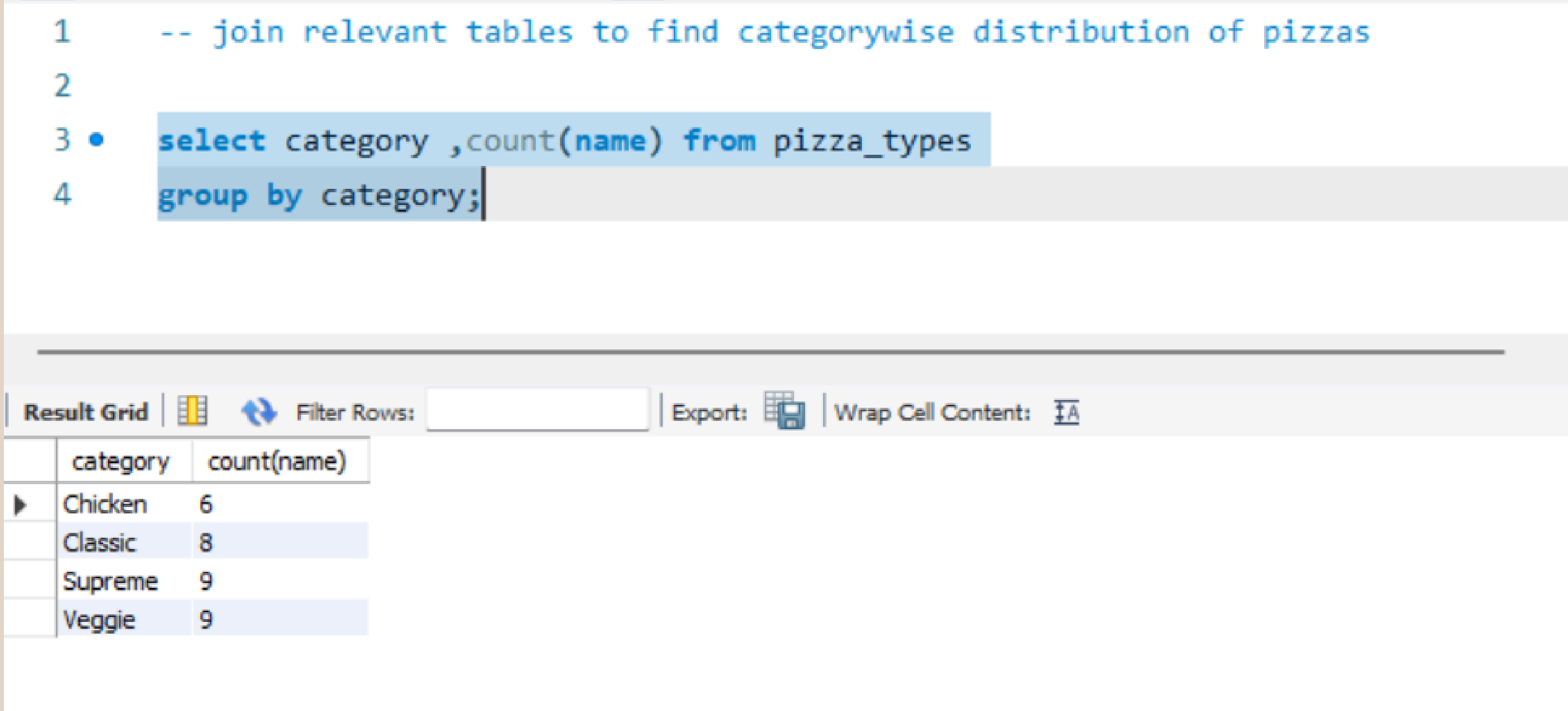
```sql
-- Intermediate:
-- Join the necessary tables to find the total quantity of each pizza category ordered.

select pizza_types.category,sum(order_details.quantity)as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



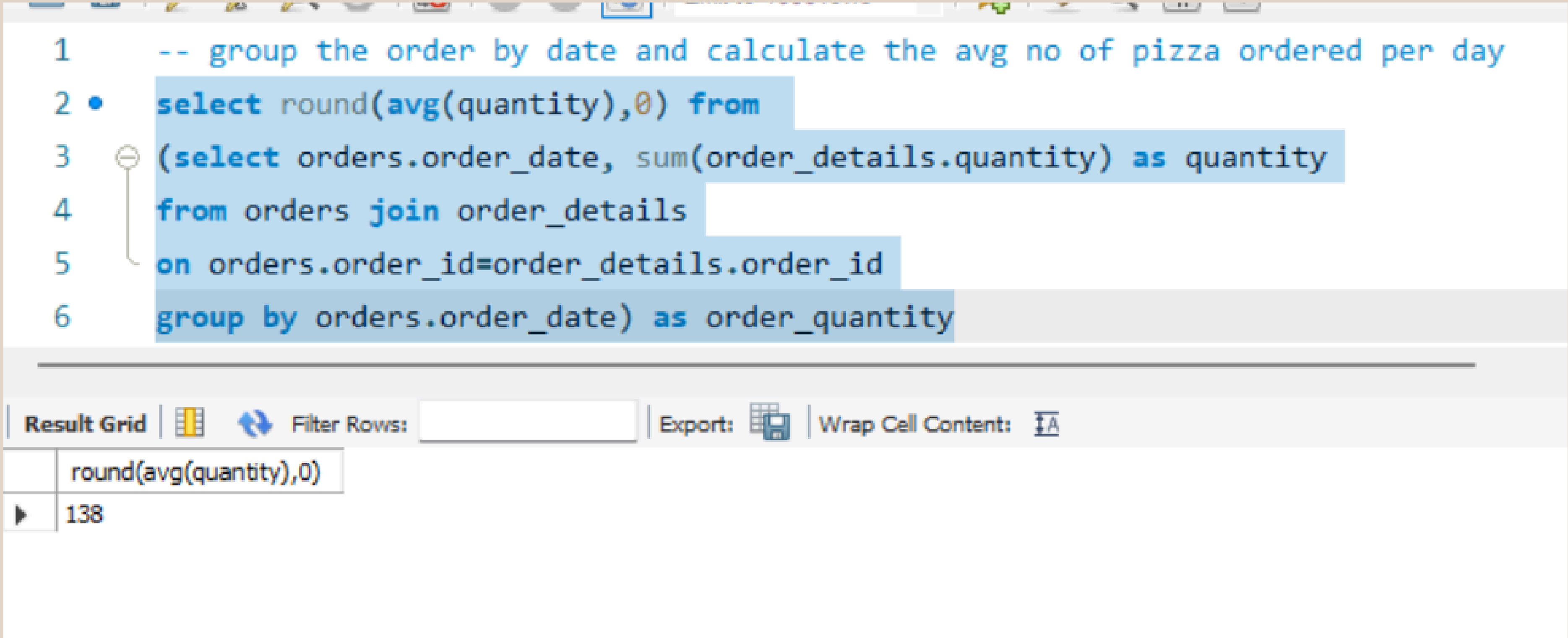
```sql
-- Determine the distribution of orders by hour of the day.

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
15	1468
16	1920
17	2336
18	2399



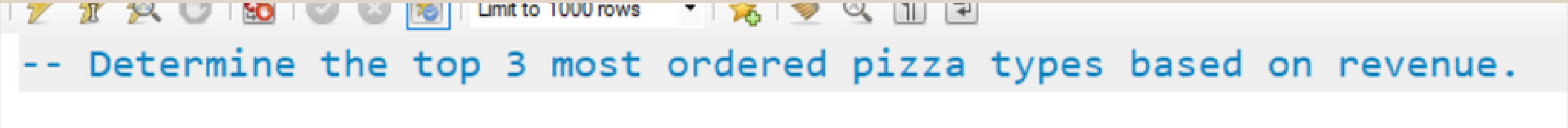
```sql
-- join relevant tables to find categorywise distribution of pizzas

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

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

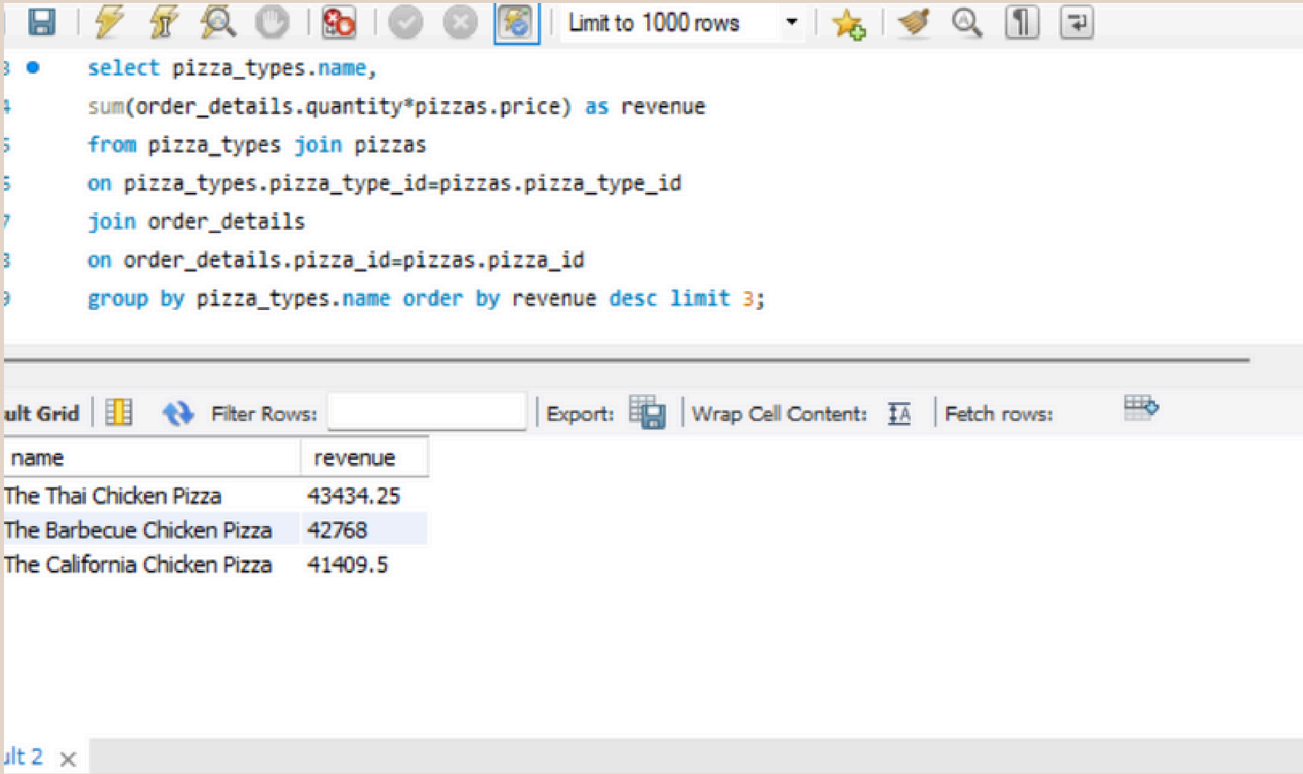


```
-- group the order by date and calculate the avg no of pizza ordered per day
select round(avg(quantity),0) from
(select orders.order_date, sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id=order_details.order_id
group by orders.order_date) as order_quantity
```

```
round(avg(quantity),0)
138
```

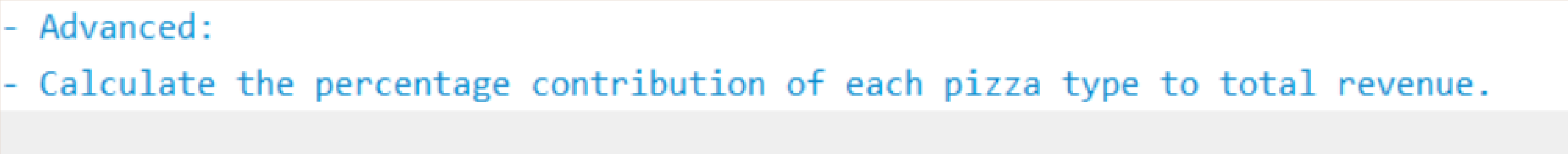


```
-- Determine the top 3 most ordered pizza types based on revenue.
```

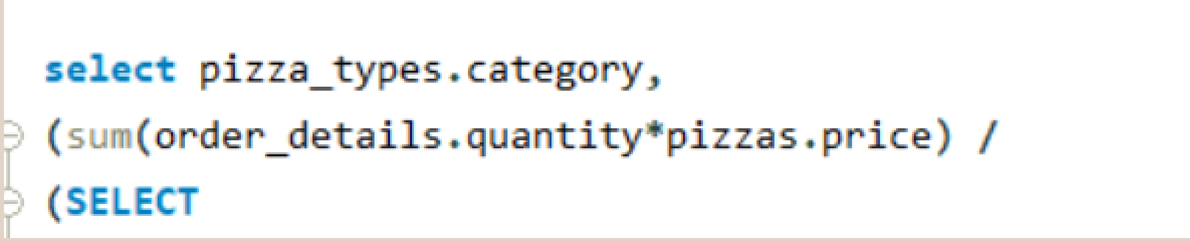


```
select pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

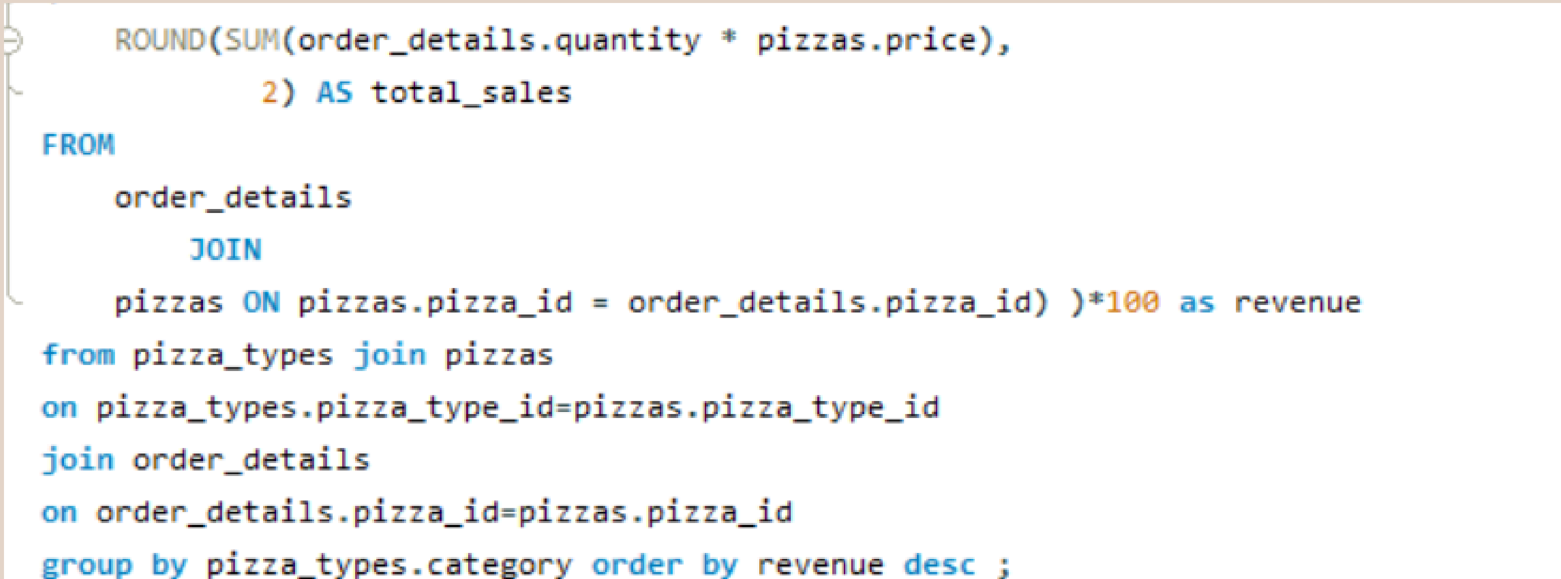
```
name                          revenue
The Thai Chicken Pizza        43434.25
The Barbecue Chicken Pizza    42768
The California Chicken Pizza  41409.5
```



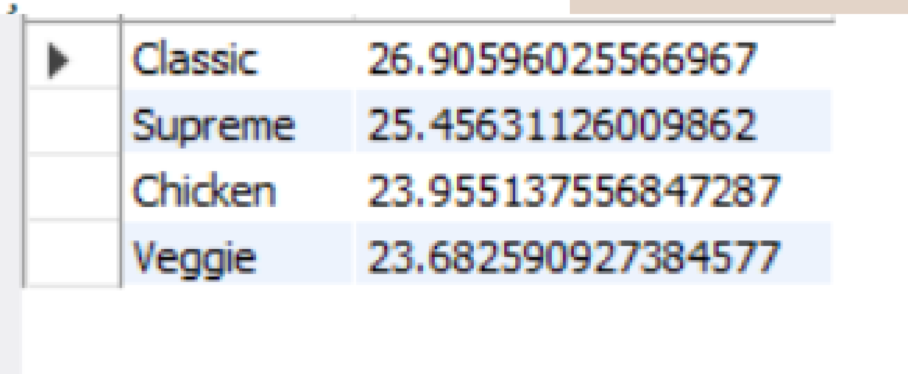
```
- Advanced:
- Calculate the percentage contribution of each pizza type to total revenue.
```



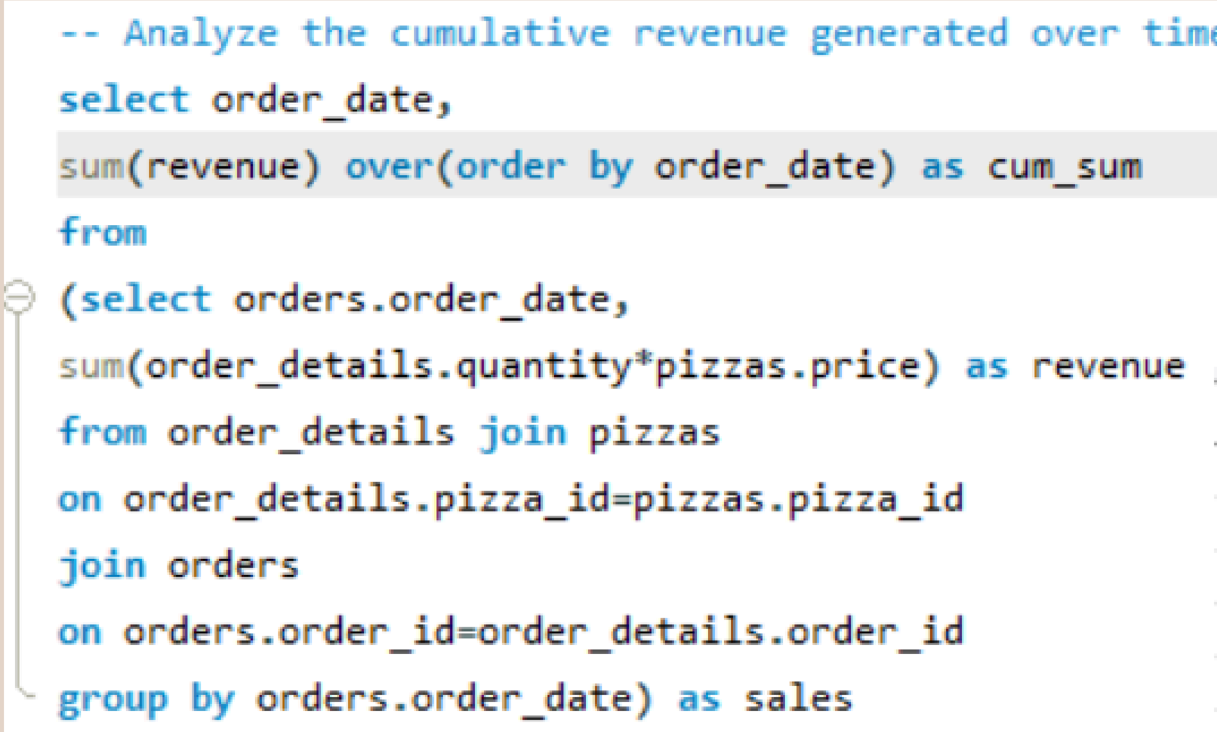
```
select pizza_types.category,
(sum(order_details.quantity*pizzas.price) /
(SELECT
```



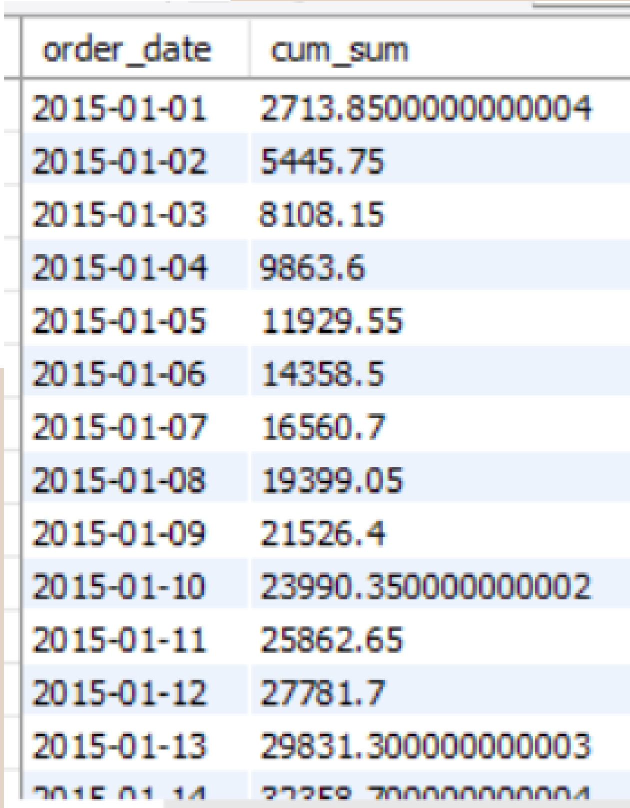
```
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id) )*100 as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category order by revenue desc ;
```



```
▶ Classic    26.9059602566967
  Supreme    25.4563112600862
  Chicken    23.9551375568472
  Veggie     23.68259092738457
```

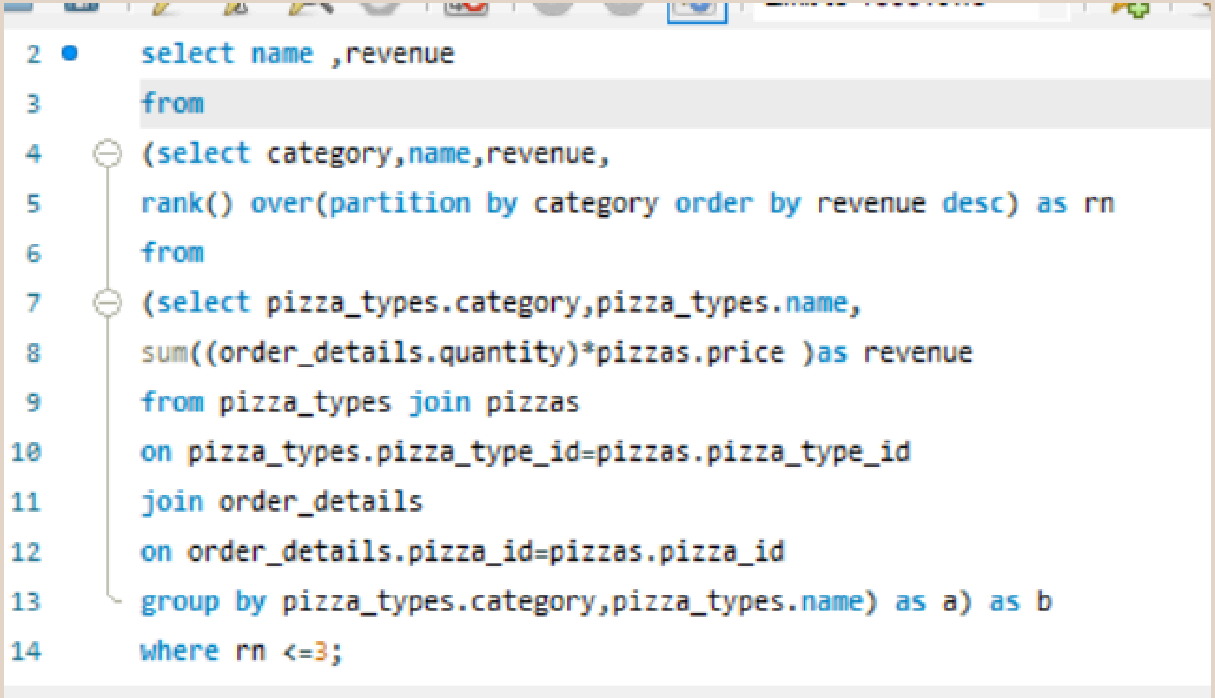


```
-- Analyze the cumulative revenue generated over time.
select order_date,
sum(revenue) over(order by order_date) as cum_sum
from
(select orders.order_date,
sum(order_details.quantity*pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=order_details.order_id
group by orders.order_date) as sales
```



order_date	cum_sum
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

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 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,pizza_types.name) as a) as b
where rn <=3;
```