Pizza Sales Analysis

SQL Project

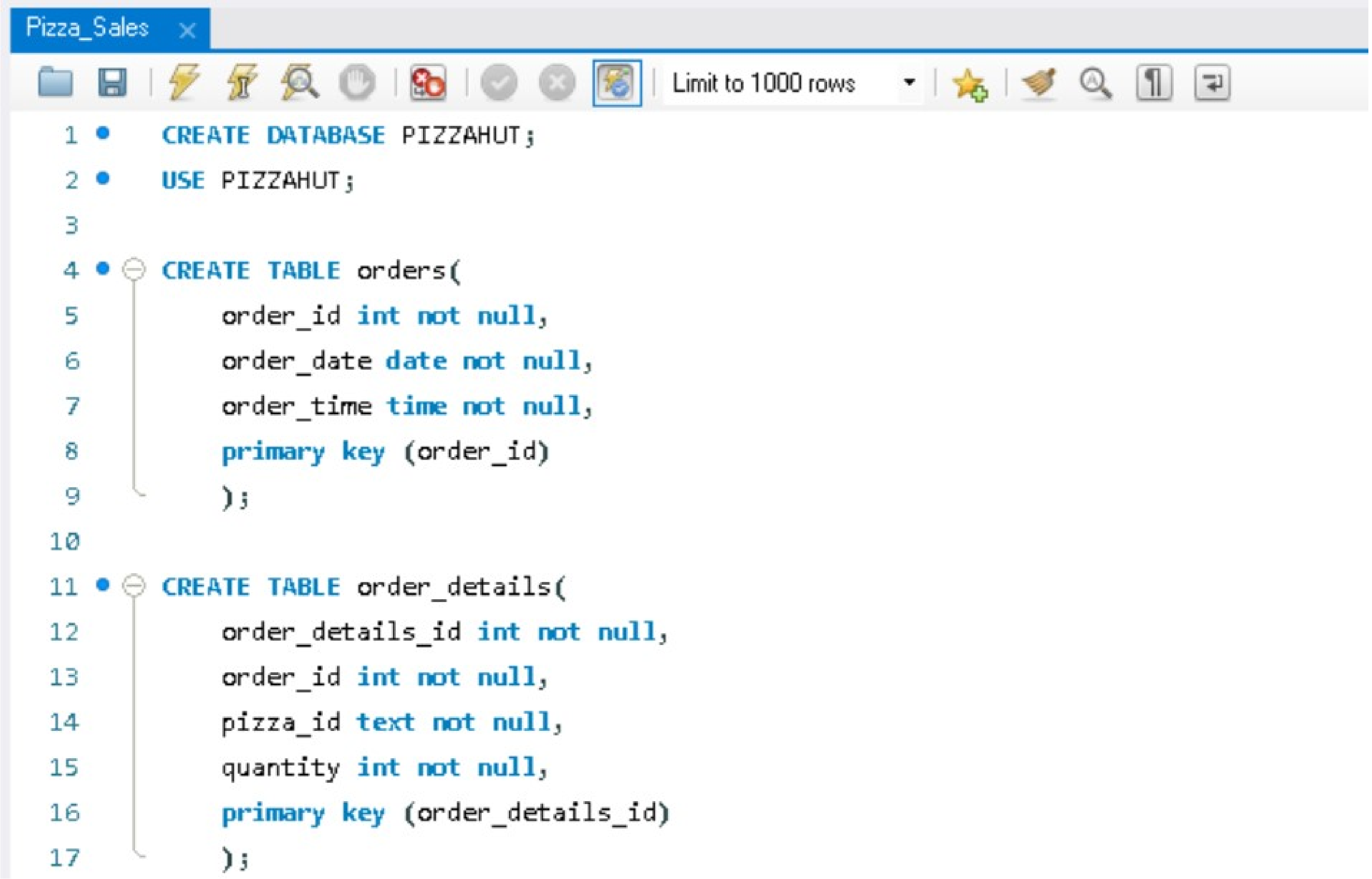
This project focuses on analyzing pizza sales data using SQL via XAMPP (MySQL). The goal is to extract meaningful business insights from order data.

- Tools Used

SQL
XAMPP

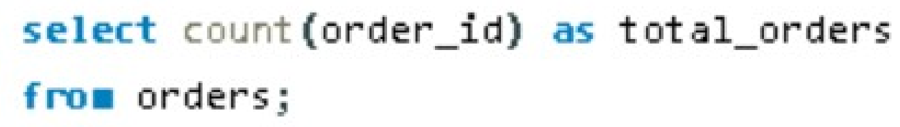
- Conclusion

Data cleaning and joining
Sales and time-based analysis
Revenue and performance tracking

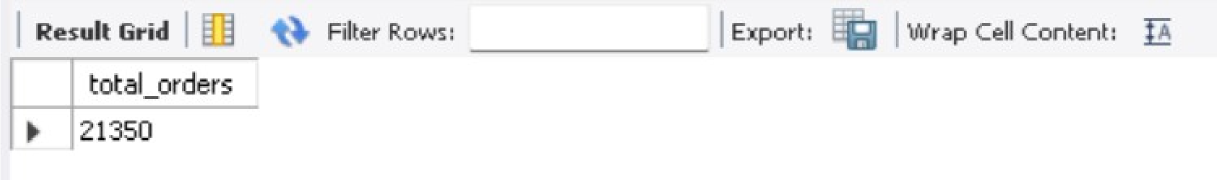


```
CREATE DATABASE PIZZAHUT;
USE PIZZAHUT;

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)
    );
```

(01) Retrieve the total number of orders placed.

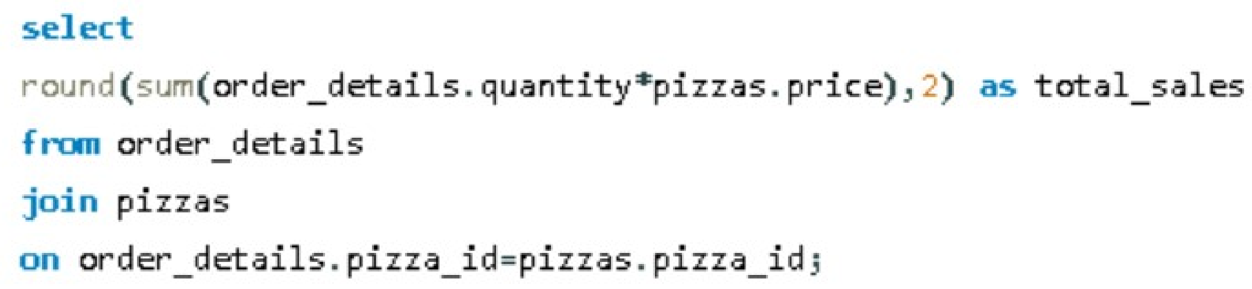


```
select count(order_id) as total_orders
from orders;
```

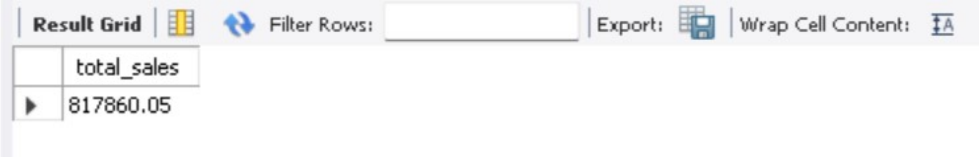


```
total_orders
21350
```

(02) Calculate the total revenue generated from pizza sales.

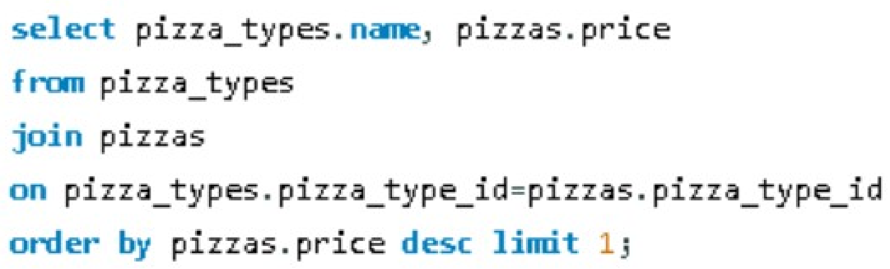


```
select
round(sum(order_details.quantity*pizzas.price),2) as total_sales
from order_details
join pizzas
on order_details.pizza_id=pizzas.pizza_id;
```

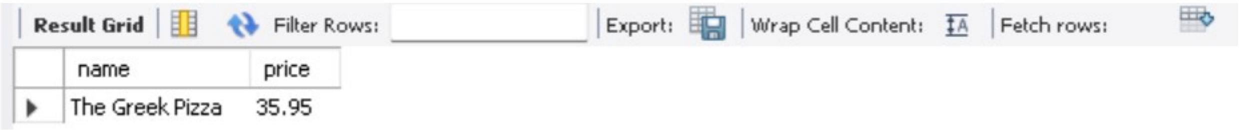


```
total_sales
817860.05
```

(03) Identify the highest-priced pizza.

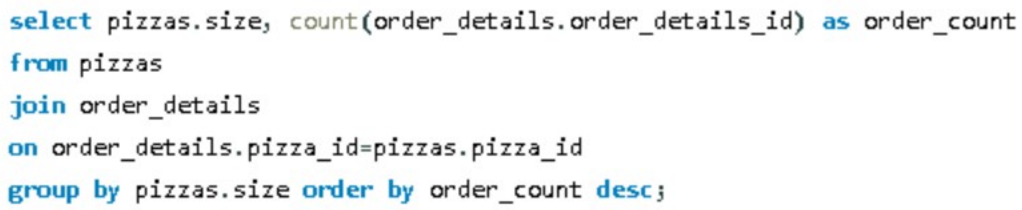


```sql
select pizza_types.name, pizzas.price
from pizza_types
join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
order by pizzas.price desc limit 1;
```

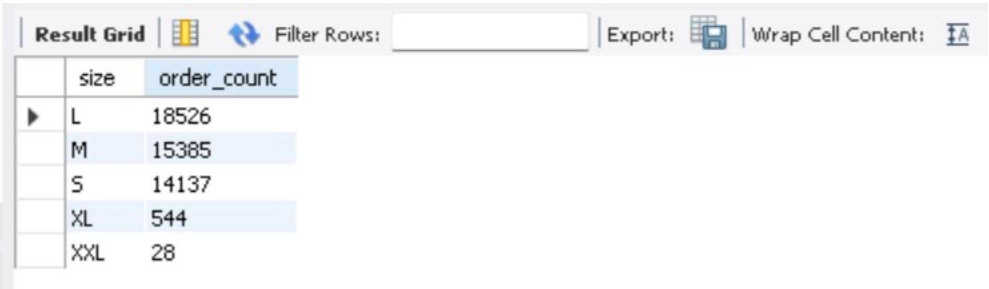


name	price
The Greek Pizza	35.95

(04) Identify the most common pizza size ordered.

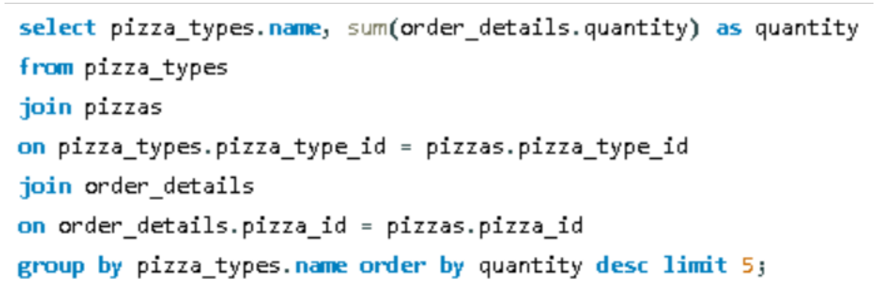


```sql
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizzas.size order by order_count desc;
```

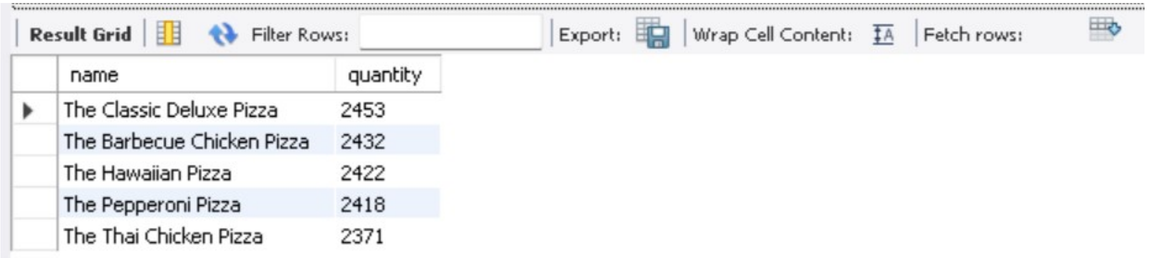


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

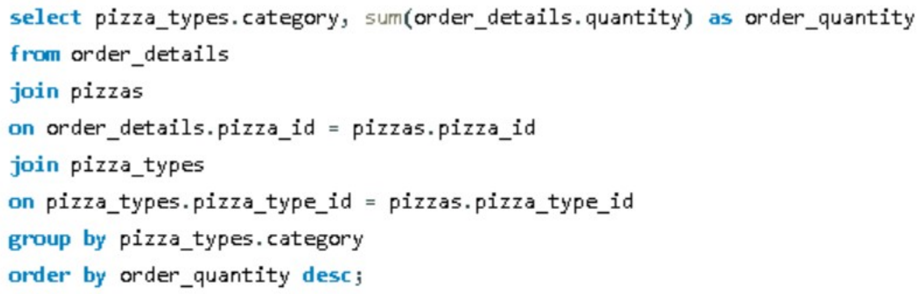


```sql
select pizza_types.name, sum(order_details.quantity) as quantity
from pizza_types
join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by quantity desc limit 5;
```

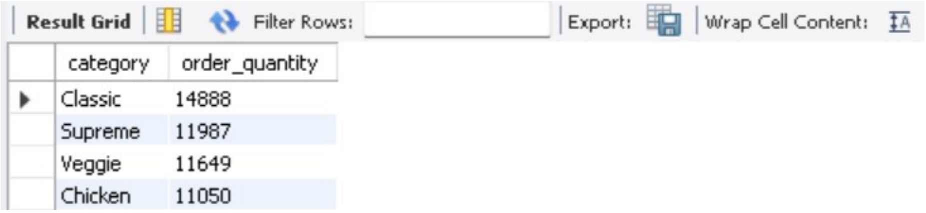


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

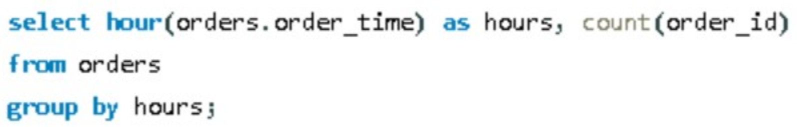


```sql
select pizza_types.category, sum(order_details.quantity) as order_quantity
from order_details
join pizzas
on order_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category
order by order_quantity desc;
```

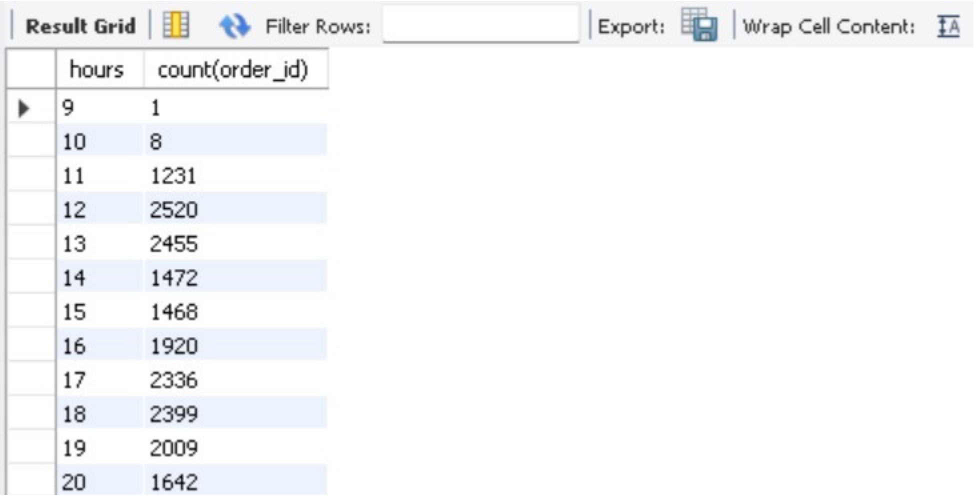


category	order_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

(07) Determine the distribution of orders by hour of the day.

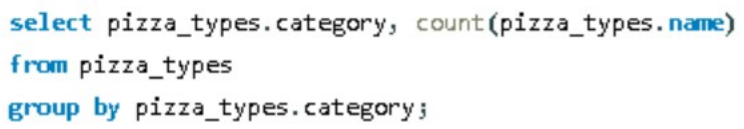


```sql
select hour(orders.order_time) as hours, count(order_id)
from orders
group by hours;
```

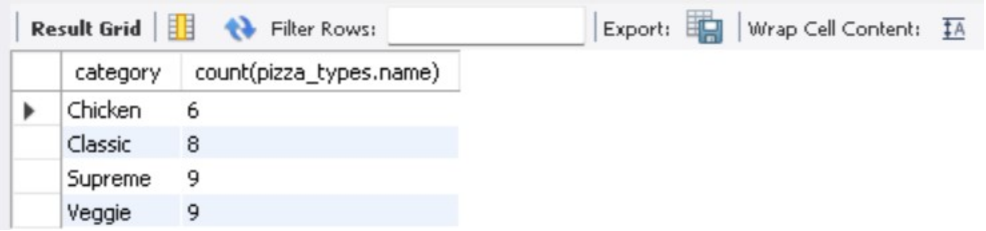


hours	count(order_id)
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

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

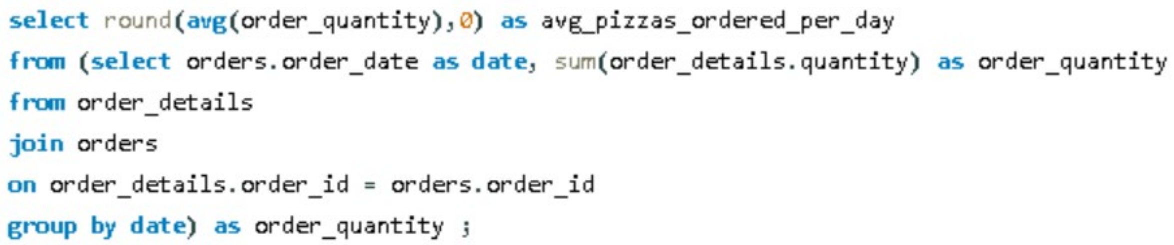


```sql
select pizza_types.category, count(pizza_types.name)
from pizza_types
group by pizza_types.category;
```

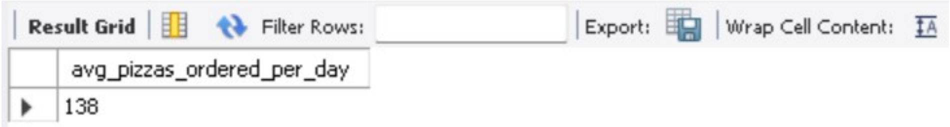


category	count(pizza_types.name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

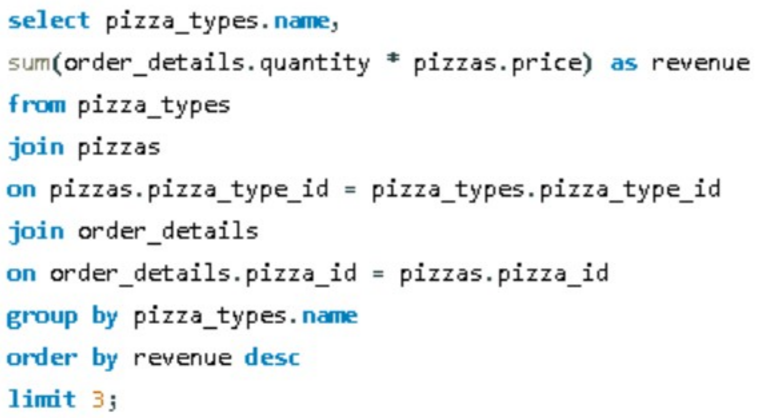


```
select round(avg(order_quantity),0) as avg_pizzas_ordered_per_day
from (select orders.order_date as date, sum(order_details.quantity) as order_quantity
from order_details
join orders
on order_details.order_id = orders.order_id
group by date) as order_quantity ;
```

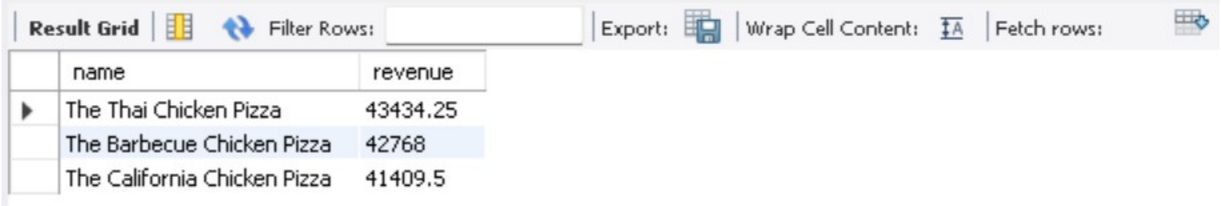


avg_pizzas_ordered_per_day
138

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

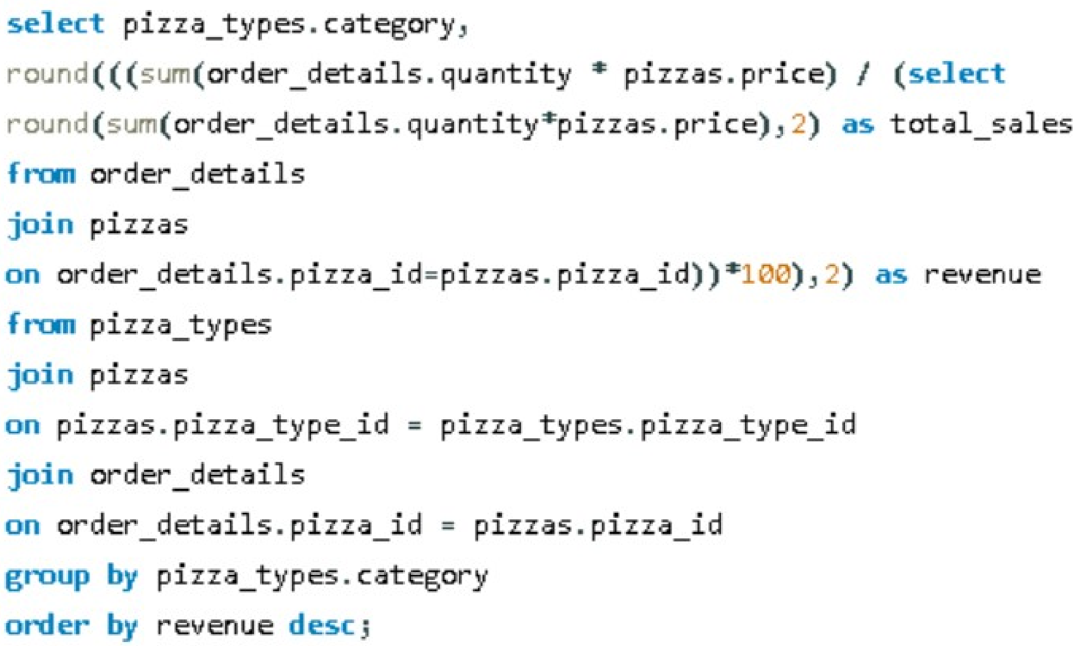


```
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types
join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by revenue desc
limit 3;
```

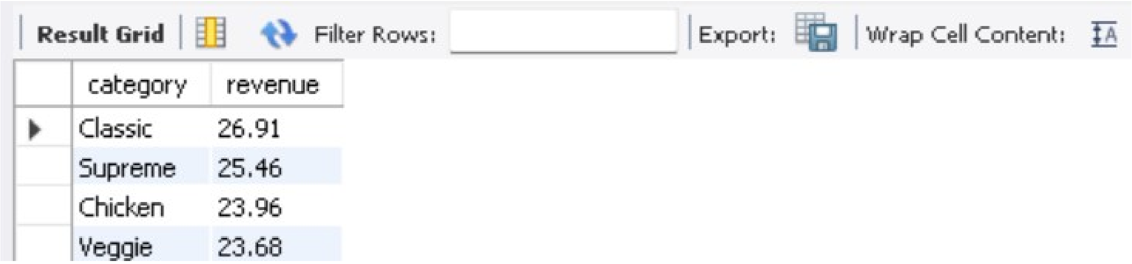


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

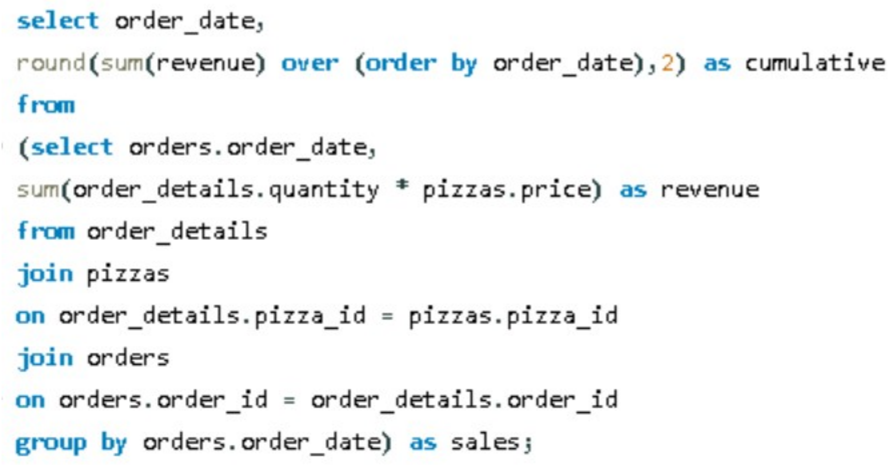


```
select pizza_types.category,
round(((sum(order_details.quantity * pizzas.price) / (select
round(sum(order_details.quantity*pizzas.price),2) as total_sales
from order_details
join pizzas
on order_details.pizza_id=pizzas.pizza_id))*100),2) as revenue
from pizza_types
join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by revenue desc;
```

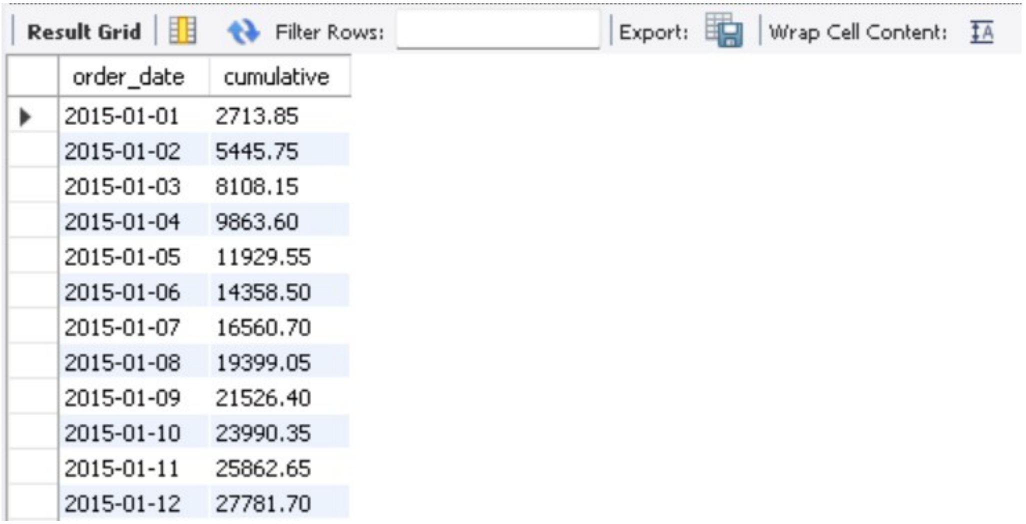


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

(12) Analyze the cumulative revenue generated over time.

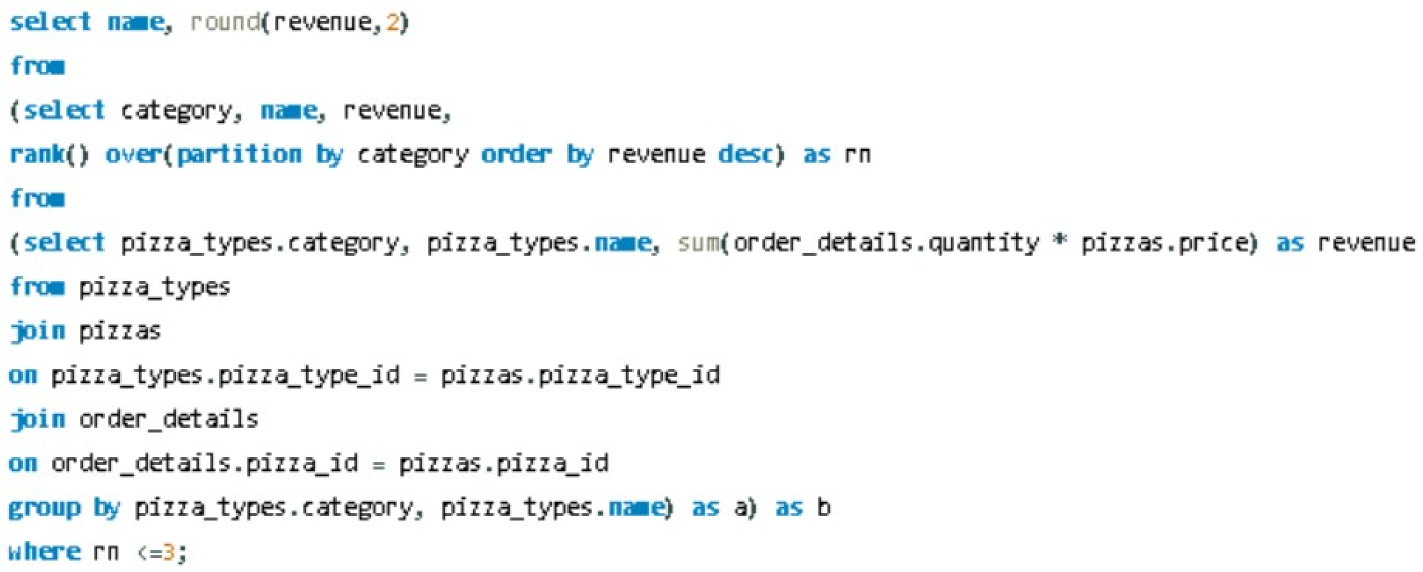


```
select order_date,
round(sum(revenue) over (order by order_date),2) as cumulative
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details
join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```

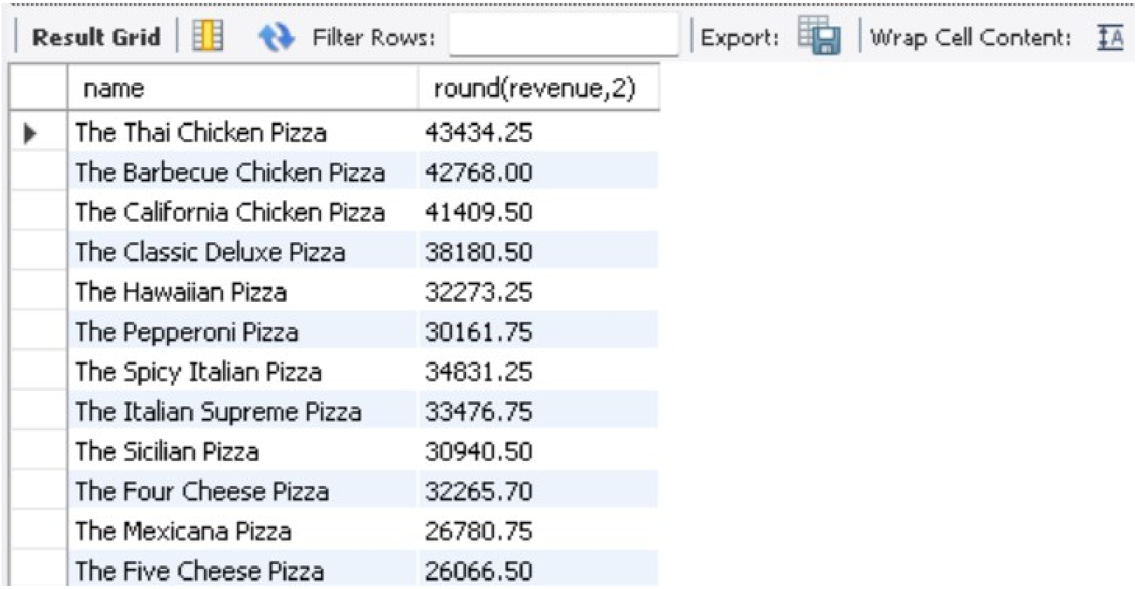


order_date	cumulative
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.60
2015-01-05	11929.55
2015-01-06	14358.50
2015-01-07	16560.70
2015-01-08	19399.05
2015-01-09	21526.40
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.70

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



```
select name, round(revenue,2)
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;
```



name	round(revenue,2)
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50
The Classic Deluxe Pizza	38180.50
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	30940.50
The Four Cheese Pizza	32265.70
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.50