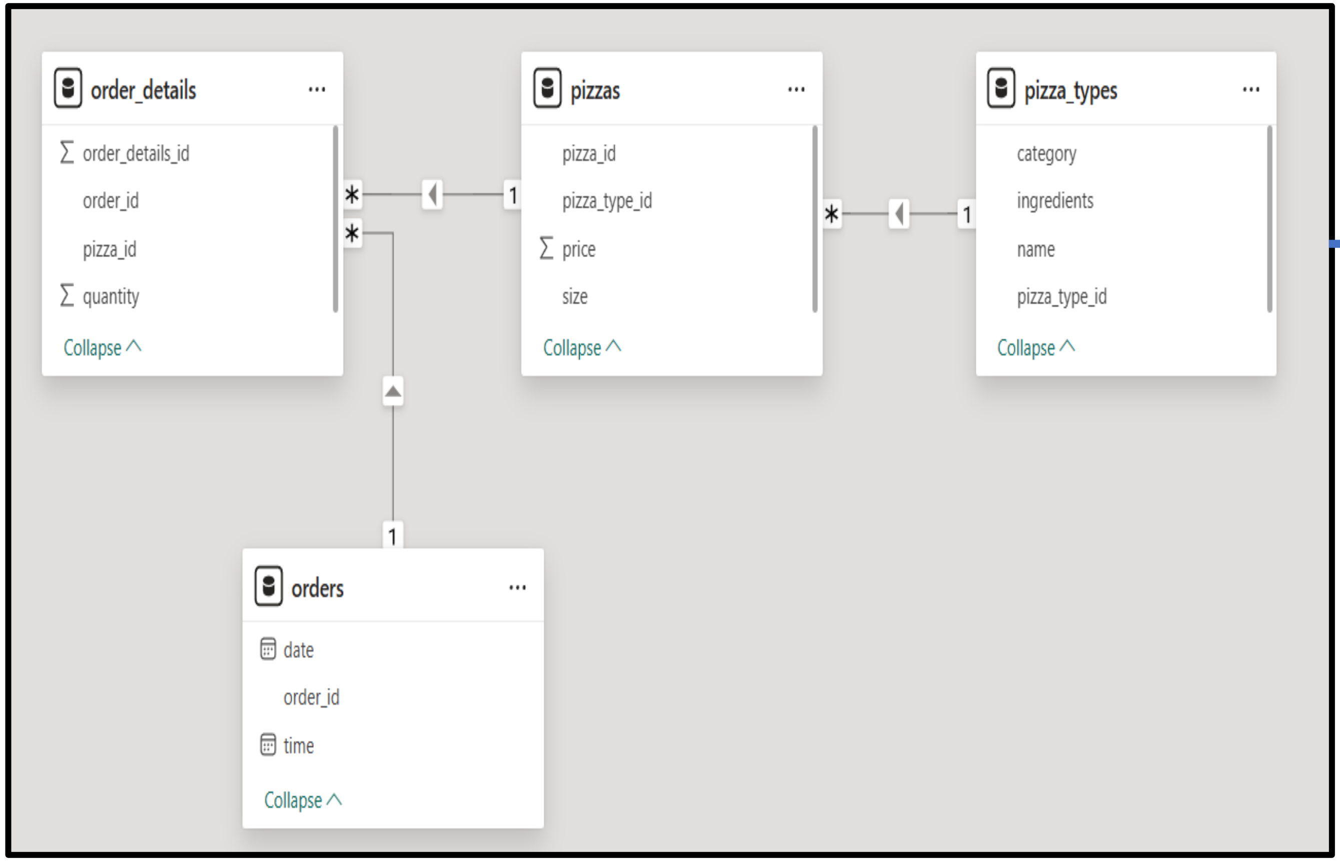
Data Analysis On Pizza Sales

Pizza Hut Dataset

SQL Analysis

- # Data Model Presentation



This schema captures pizza sales using a normalized structure.

1. **order_details:** Fact table with order_id, pizza_id and quantity.
2. **orders:** Adds date and time for each order.
3. **pizzas:** Contains price, size and links to pizza type.
4. **pizza_types:** Defines name, category and ingredients

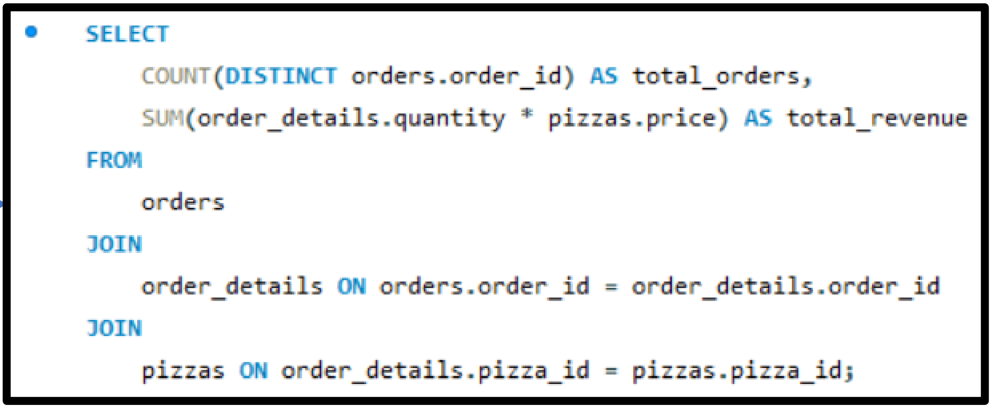
Note: This model follows a Snowflake Schema due to multiple related dimension tables. It ensures data consistency and is well-suited for analysis in BI tools

SQL Question Addressed:

1. Retrieve the total number of orders placed. Calculate the total revenue generated from pizza sales.
2. Identify the highest-priced pizza. Identify the most common pizza size ordered.
3. List the top 5 most ordered pizza types along with their quantities.
4. Join the necessary tables to find the total quantity of each pizza category ordered.
5. Determine the distribution of orders by hour of the day. Join relevant tables to find the category-wise distribution of pizzas.
6. Group orders by date to find the average pizzas ordered per day and list the top 5 dates with the highest pizza orders.
7. Determine the top 3 most ordered pizza types based on revenue.
8. Calculate the percentage contribution of each pizza type to total revenue.
9. Analyze the cumulative revenue generated over time.
10. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

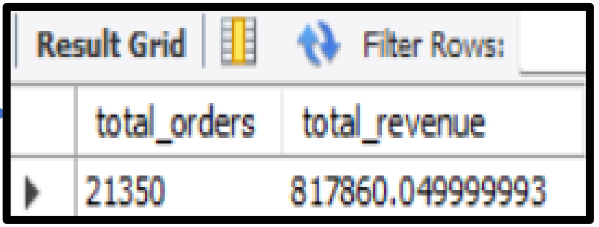
1.Retrieve the total number of orders placed. Calculate the total revenue generated from pizza sales.

Input Query



```
SELECT
    COUNT(DISTINCT orders.order_id) AS total_orders,
    SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM
    orders
JOIN
    order_details ON orders.order_id = order_details.order_id
JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result

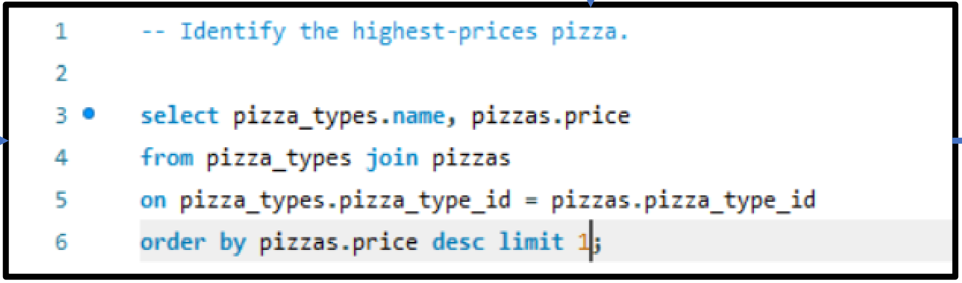


total_orders	total_revenue
21350	817860.049999993

2. Identify the highest-priced pizza. Identify the most common pizza size ordered.

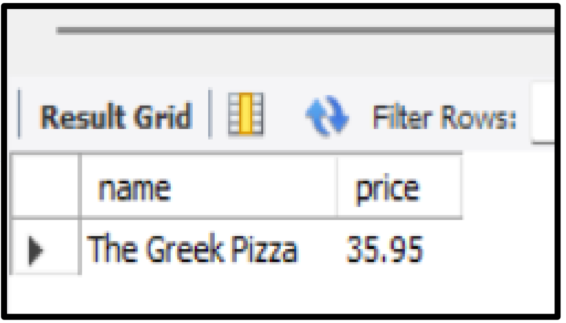
Highest Price Pizza

Input Query



```sql
-- Identify the highest-prices 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;
```

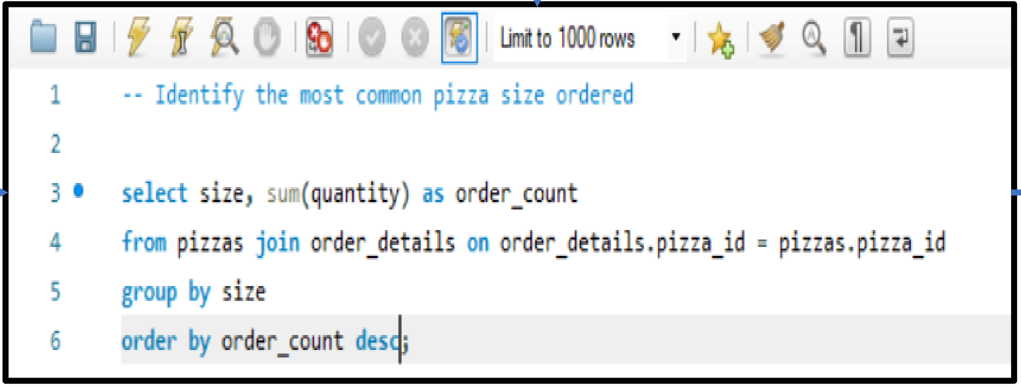
Result



name	price
The Greek Pizza	35.95

Most Common Pizza Size Ordered

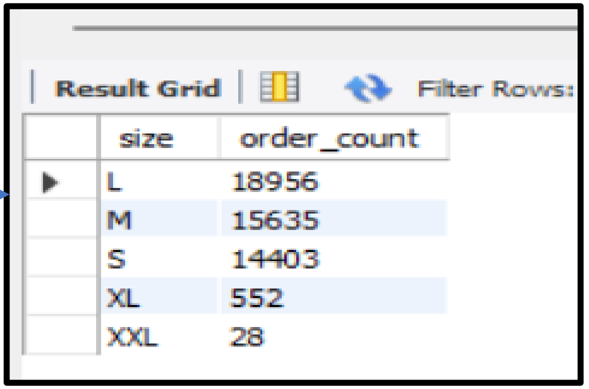
Input Query



Limit to 1000 rows

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

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

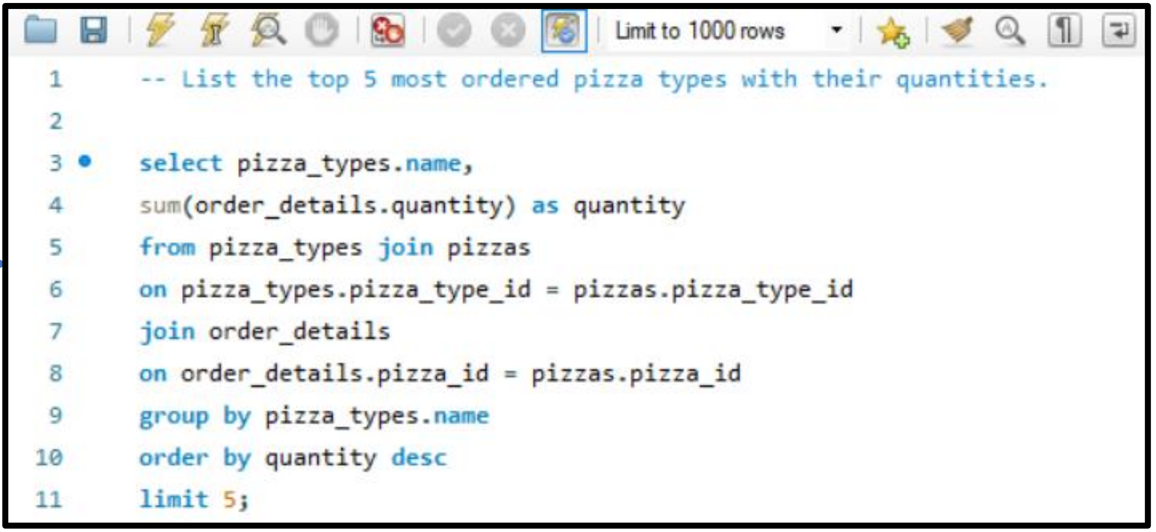
Result



size	order_count
L	18956
M	15635
S	14403
XL	552
XXL	28

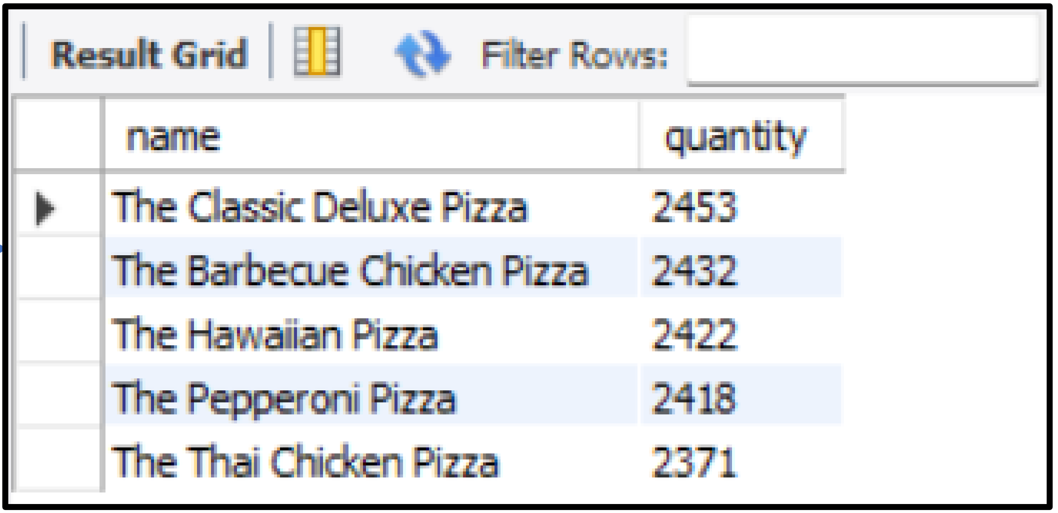
3. List the top 5 most ordered pizza types along with their quantities.

Input Query



```sql
-- List the top 5 most ordered pizza types 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
group by pizza_types.name
order by quantity desc
limit 5;
```

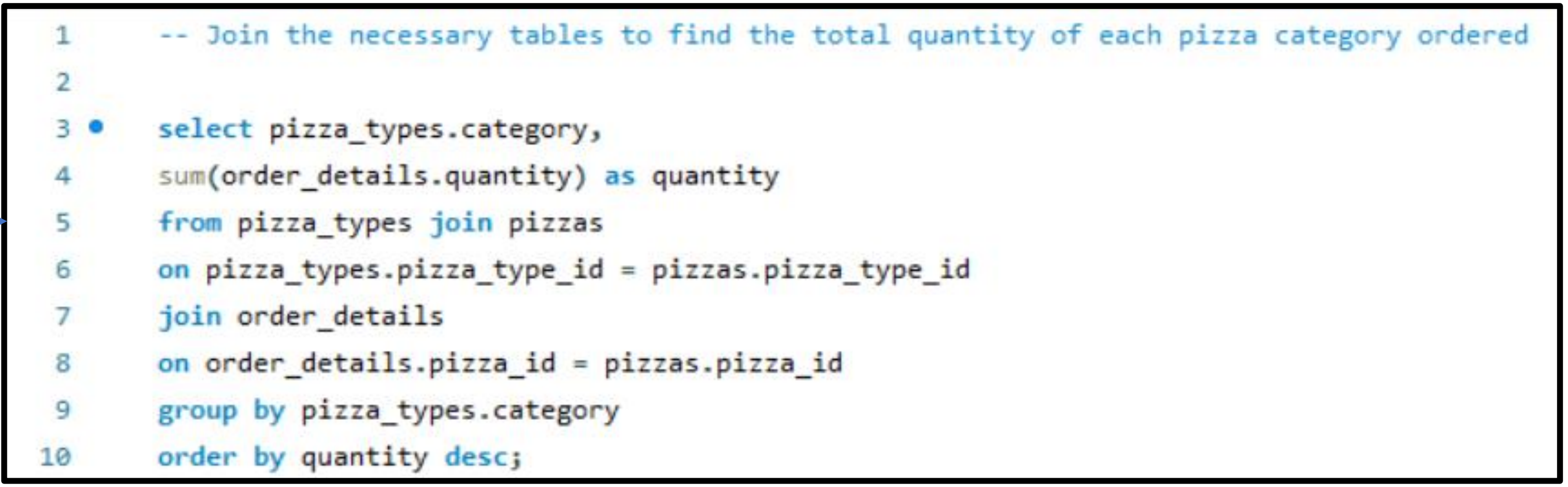
Result



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

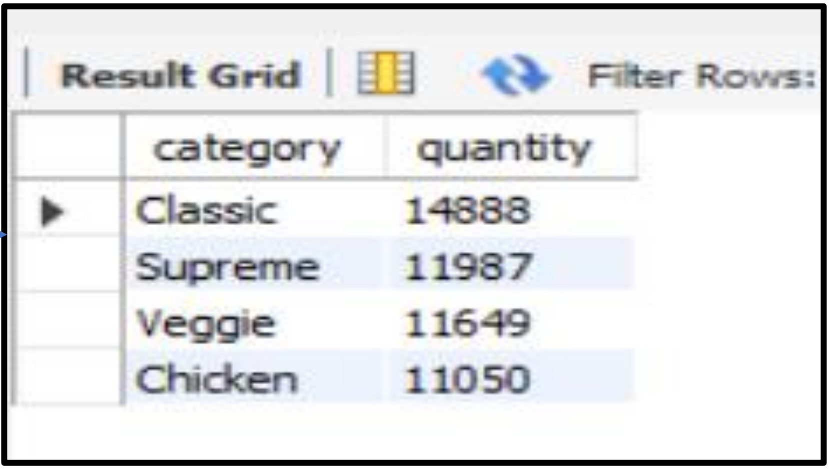
4. Join the necessary tables to find the total quantity of each pizza category ordered.

Input Query



```
-- 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
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by quantity desc;
```

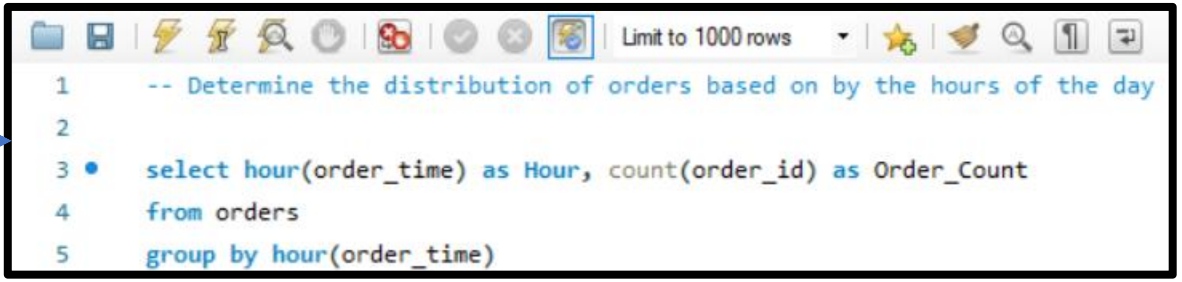
Result



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

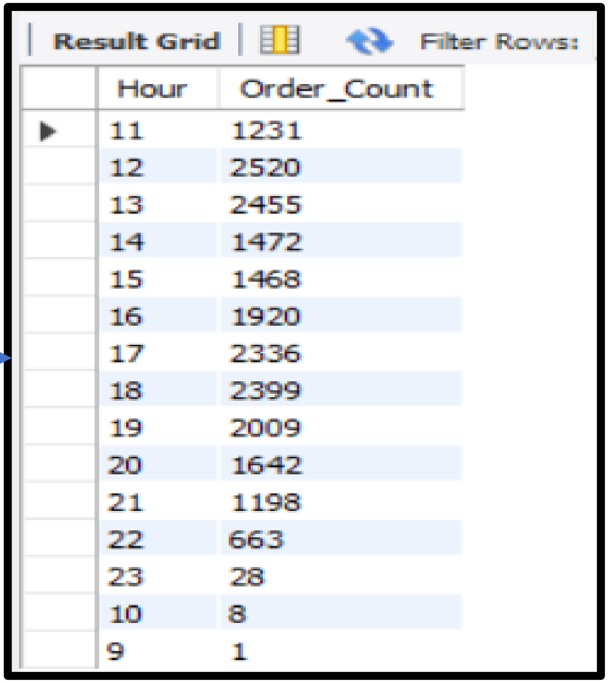
5. Determine the distribution of orders by hour of the day. Join relevant tables to find the category-wise distribution of pizzas.

Input Query



```sql
-- Determine the distribution of orders based on by the hours of the day

select hour(order_time) as Hour, count(order_id) as Order_Count
from orders
group by hour(order_time)
```

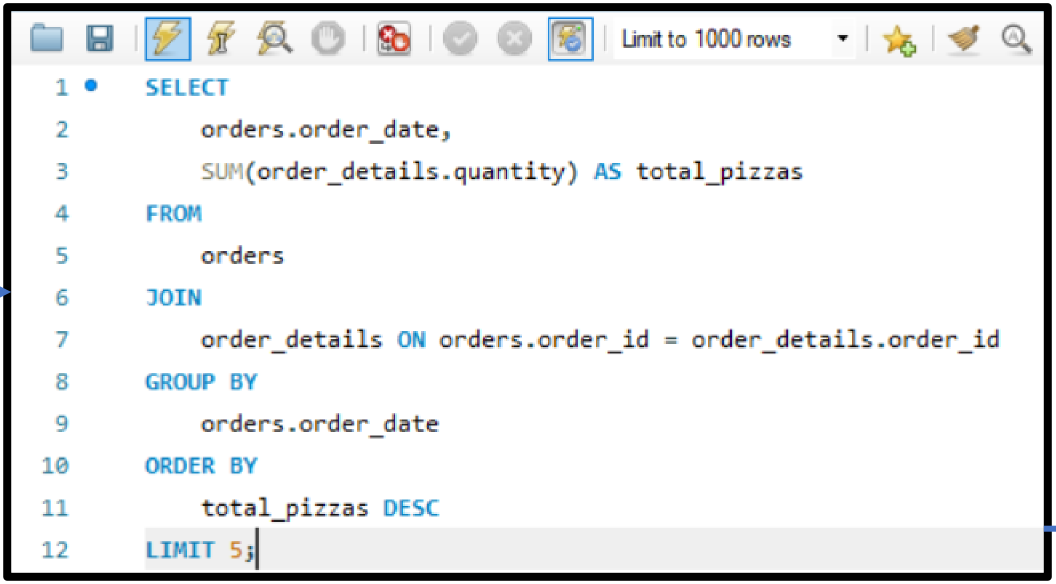
Result



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

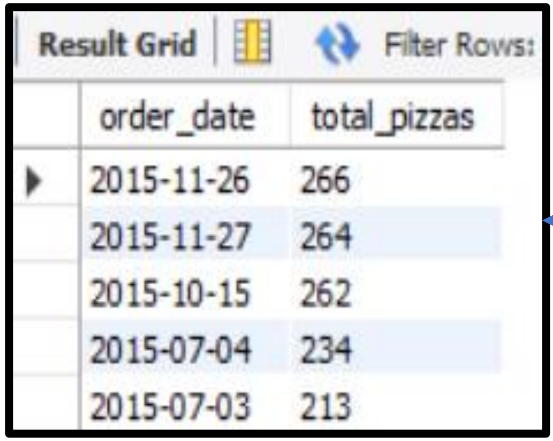
6. "Group orders by date to find the average pizzas ordered per day and list the top 5 dates with the highest pizza orders."

Input Query



```sql
SELECT
    orders.order_date,
    SUM(order_details.quantity) AS total_pizzas
FROM
    orders
JOIN
    order_details ON orders.order_id = order_details.order_id
GROUP BY
    orders.order_date
ORDER BY
    total_pizzas DESC
LIMIT 5;
```

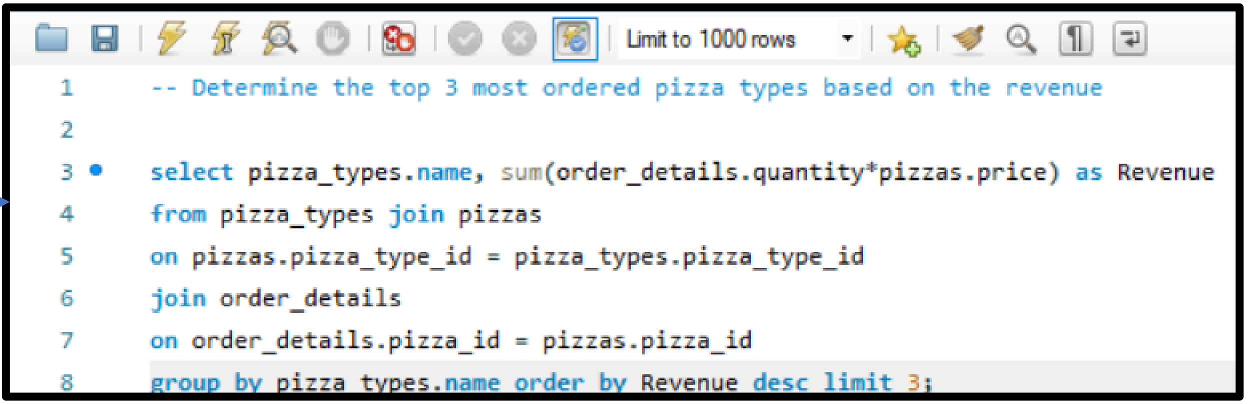
Result



order_date	total_pizzas
2015-11-26	266
2015-11-27	264
2015-10-15	262
2015-07-04	234
2015-07-03	213

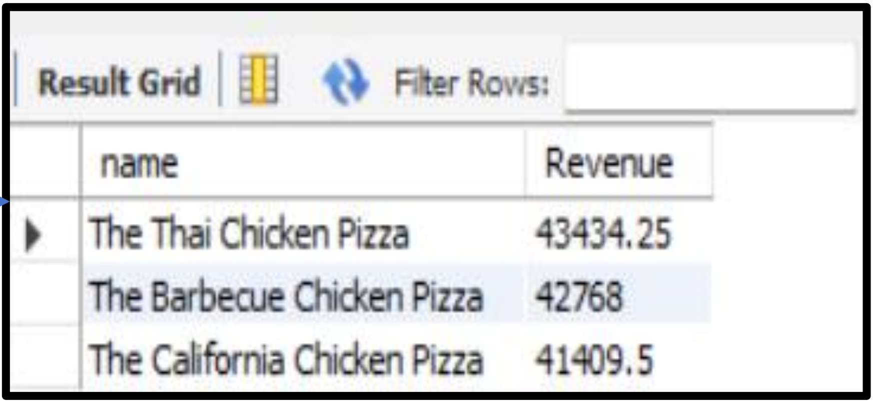
7. Determine the top 3 most ordered pizza types based on revenue.

Input Query



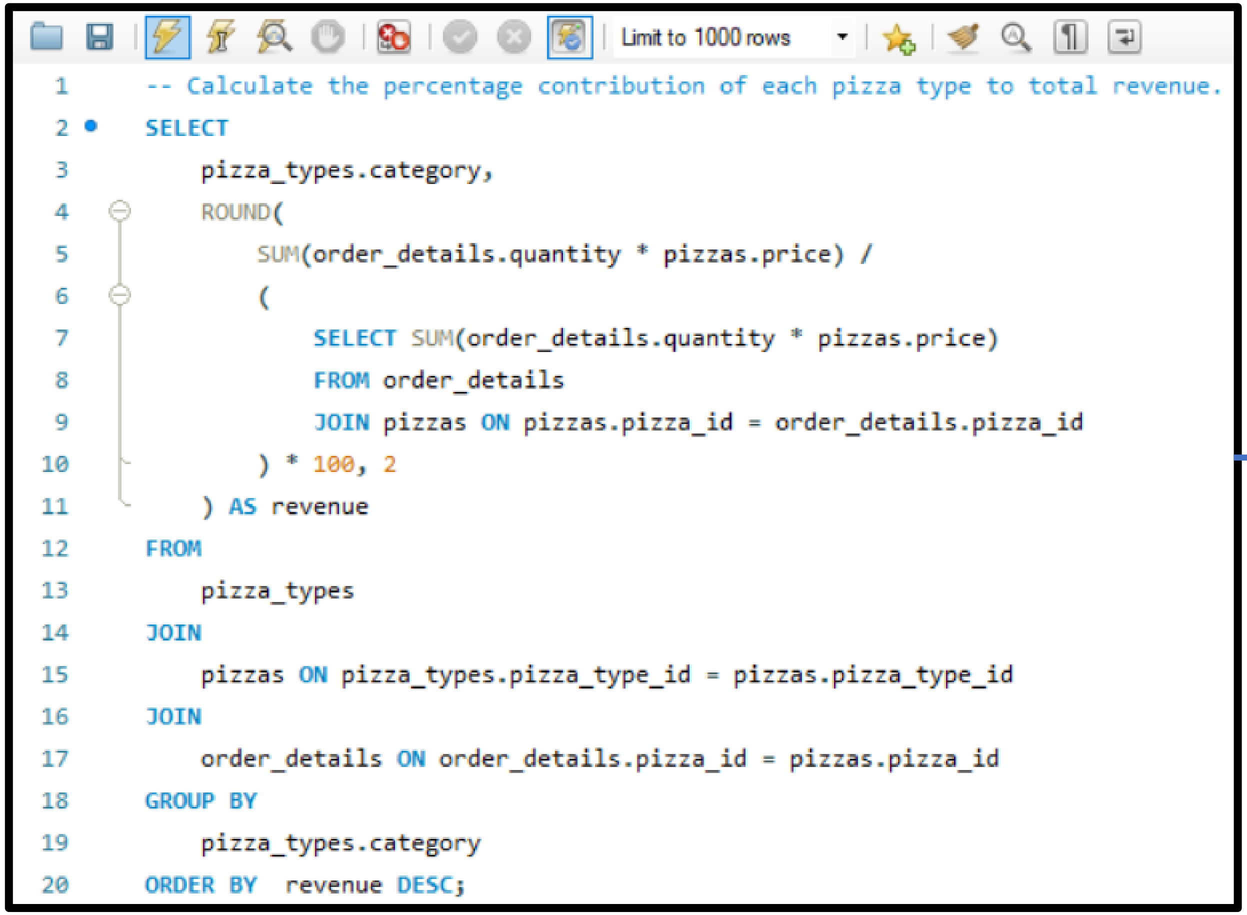
```
-- Determine the top 3 most ordered pizza types based on the revenue

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;
```

Result



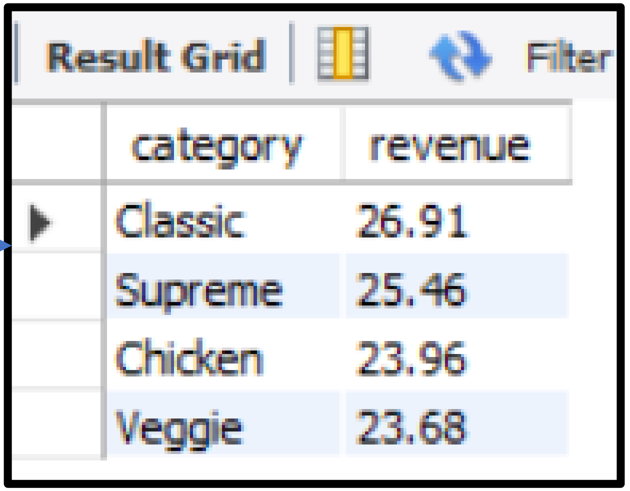
name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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



```
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    ROUND(
        SUM(order_details.quantity * pizzas.price) /
        (
            SELECT SUM(order_details.quantity * pizzas.price)
            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;
```

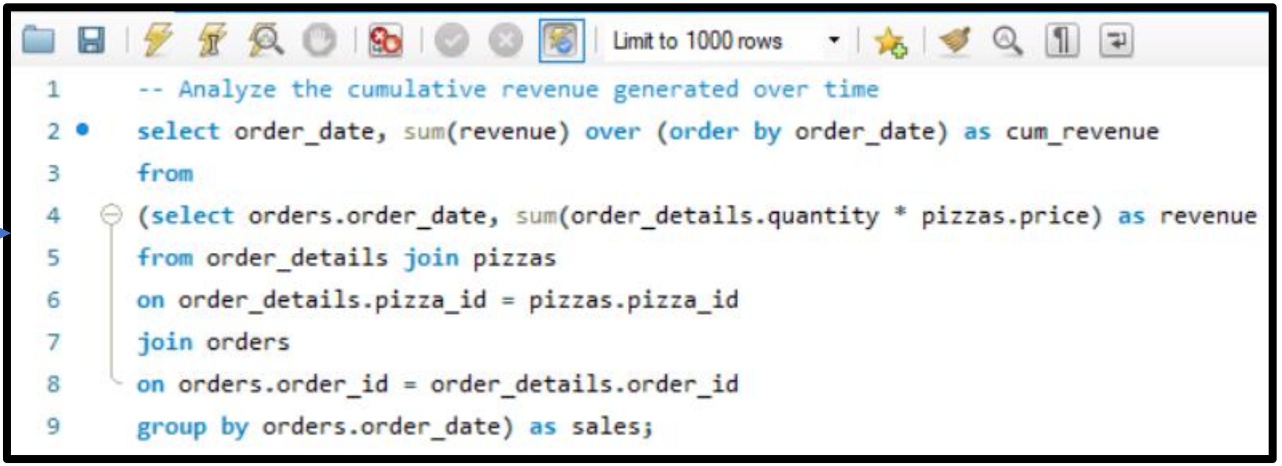
Result



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

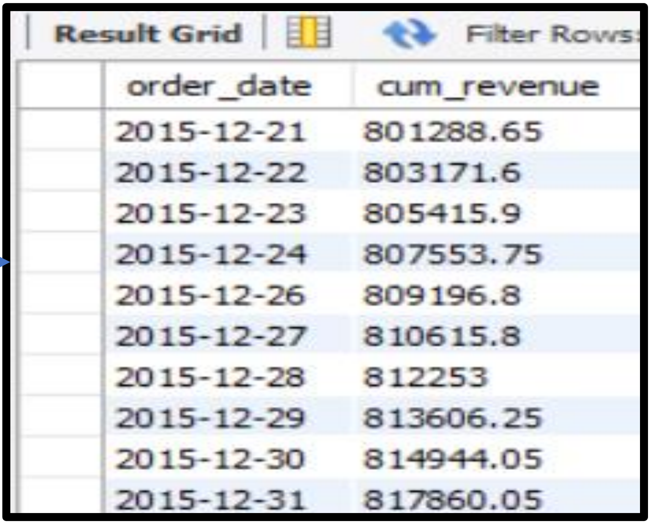
9. Analyze the cumulative revenue generated over time

Input Query



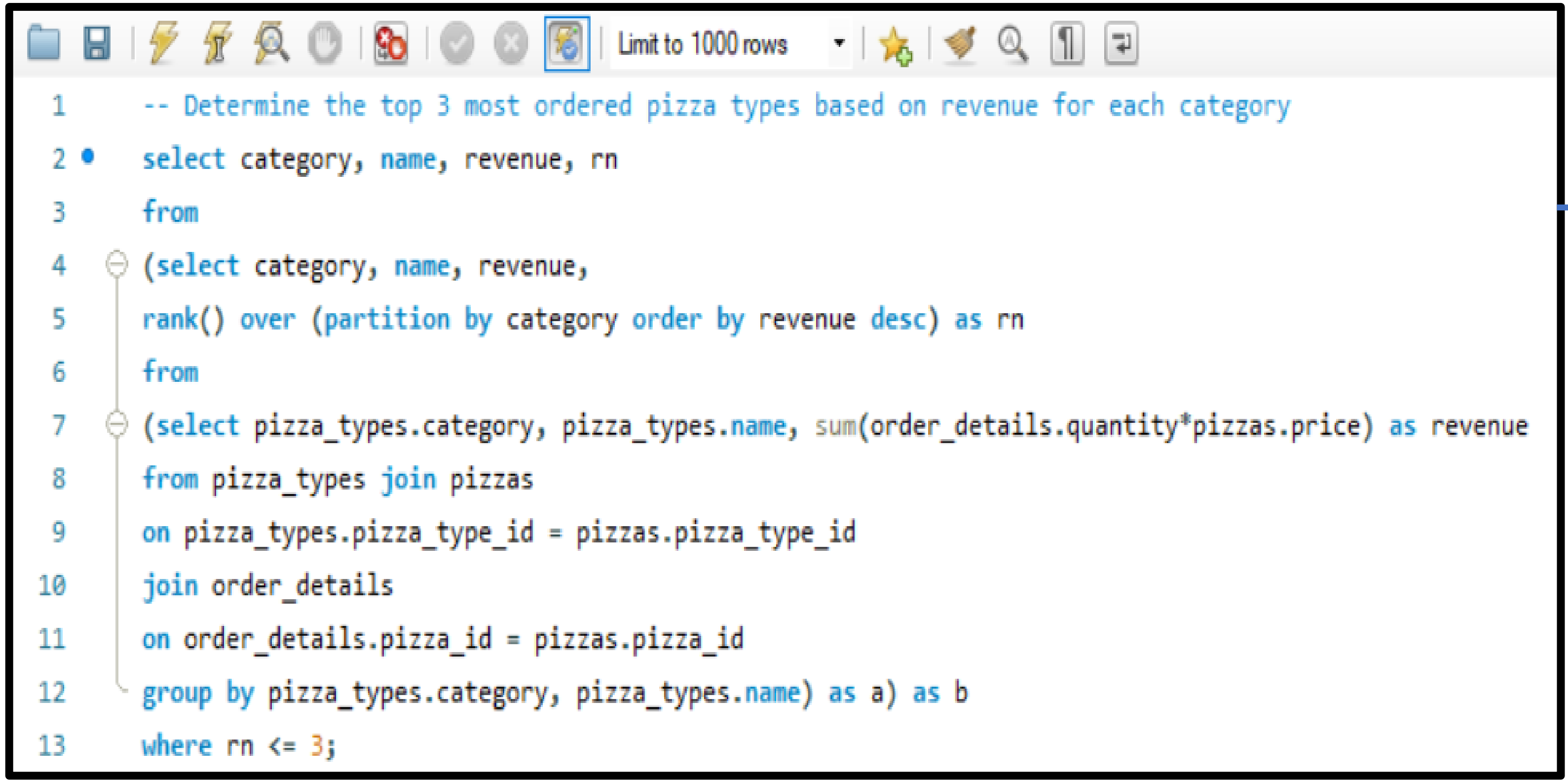
```
-- Analyze the cumulative revenue generated over time
select order_date, sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date, sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders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

Result



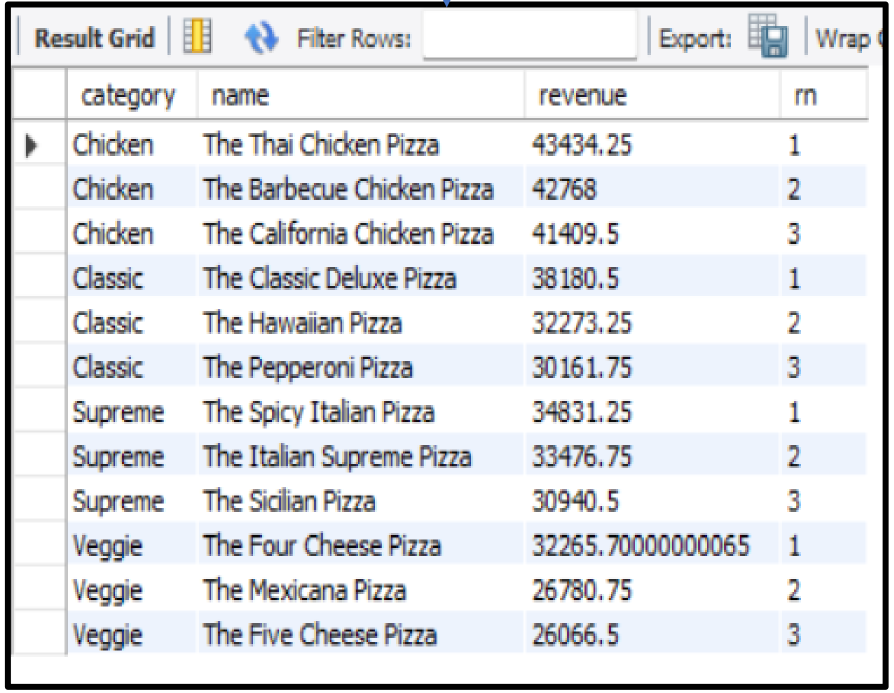
order_date	cum_revenue
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

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



```sql
-- Determine the top 3 most ordered pizza types based on revenue for each category
select category, name, revenue, rn
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;
```

Result



category	name	revenue	rn
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