PizzaHUT Data Analysis

Company Overview

Pizza Hut is a global restaurant chain known for its diverse range of pizzas, pastas, and side dishes. Founded in 1958 by Dan and Frank Carney in Wichita, Kansas, Pizza Hut quickly became one of the most recognizable pizza brands worldwide. Its menu features signature items like the Pan Pizza, Thin 'N Crispy, and Stuffed Crust, offering a variety of toppings and crust styles. Pizza Hut operates in over 100 countries, providing both dine-in and delivery options. The company is a subsidiary of Yum! Brands, which also owns other popular fast-food chains like KFC and Taco Bell.

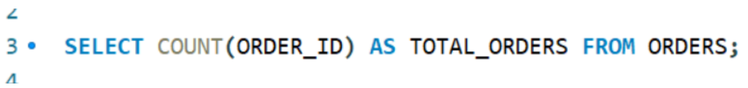
Project Overview

This project involves a comprehensive analysis of PizzaHut data using SQL. The goal is to extract valuable insights and answer various business questions based on the dataset. The following README provides a detailed account of the project's objectives, business problems, solutions, findings, and conclusions.

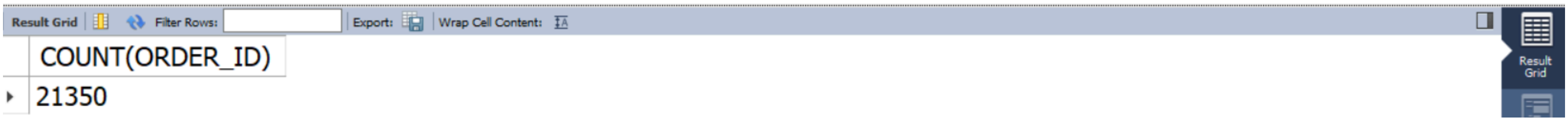
Business Problems and solutions

Basic:

1. Retrieve the total number of orders placed.

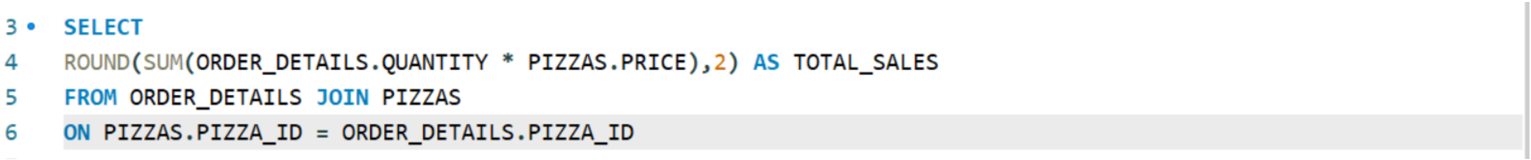


```sql
SELECT COUNT(ORDER_ID) AS TOTAL_ORDERS FROM ORDERS;
```

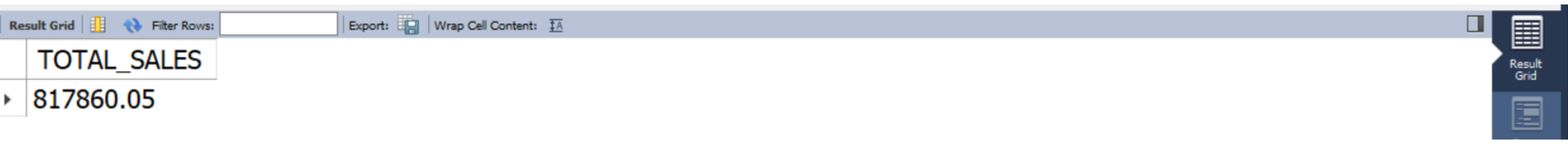


COUNT(ORDER_ID)
21350

2. Calculate the total revenue generated from pizza sales.

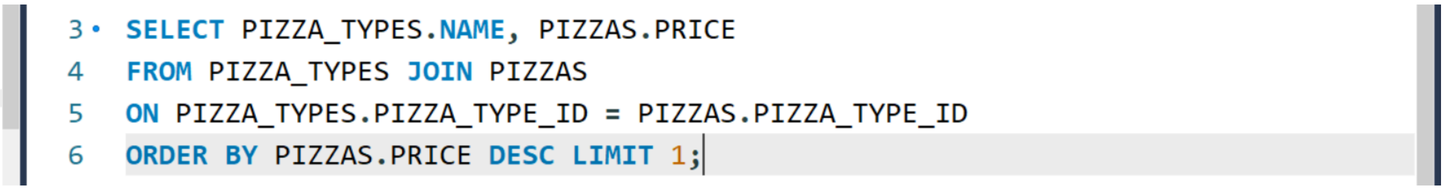


```
SELECT
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),2) AS TOTAL_SALES
FROM ORDER_DETAILS JOIN PIZZAS
ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
```

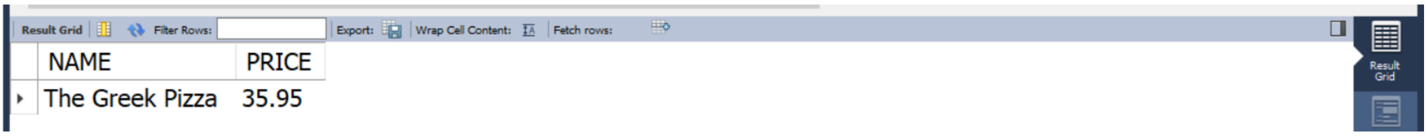


TOTAL_SALES
817860.05

3. Identify the highest-priced pizza.

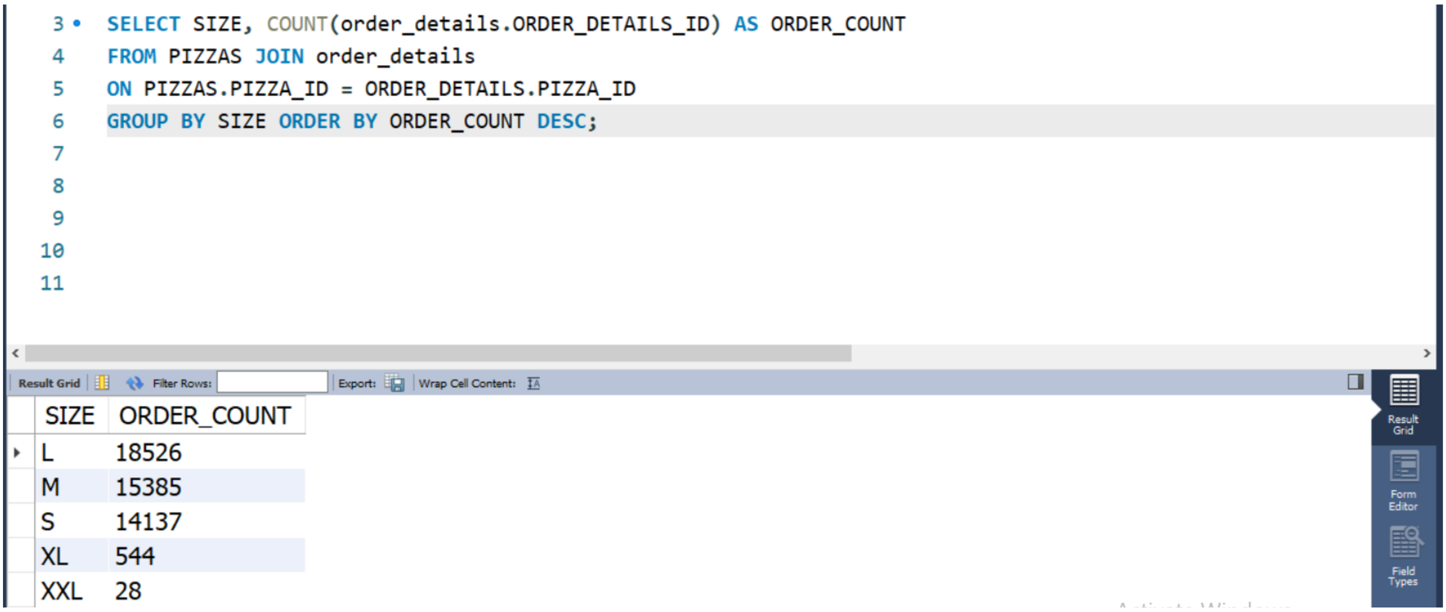


```
SELECT PIZZA_TYPES.NAME, PIZZAS.PRICE
FROM PIZZA_TYPES JOIN PIZZAS
ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
ORDER BY PIZZAS.PRICE DESC LIMIT 1;
```



NAME	PRICE
The Greek Pizza	35.95

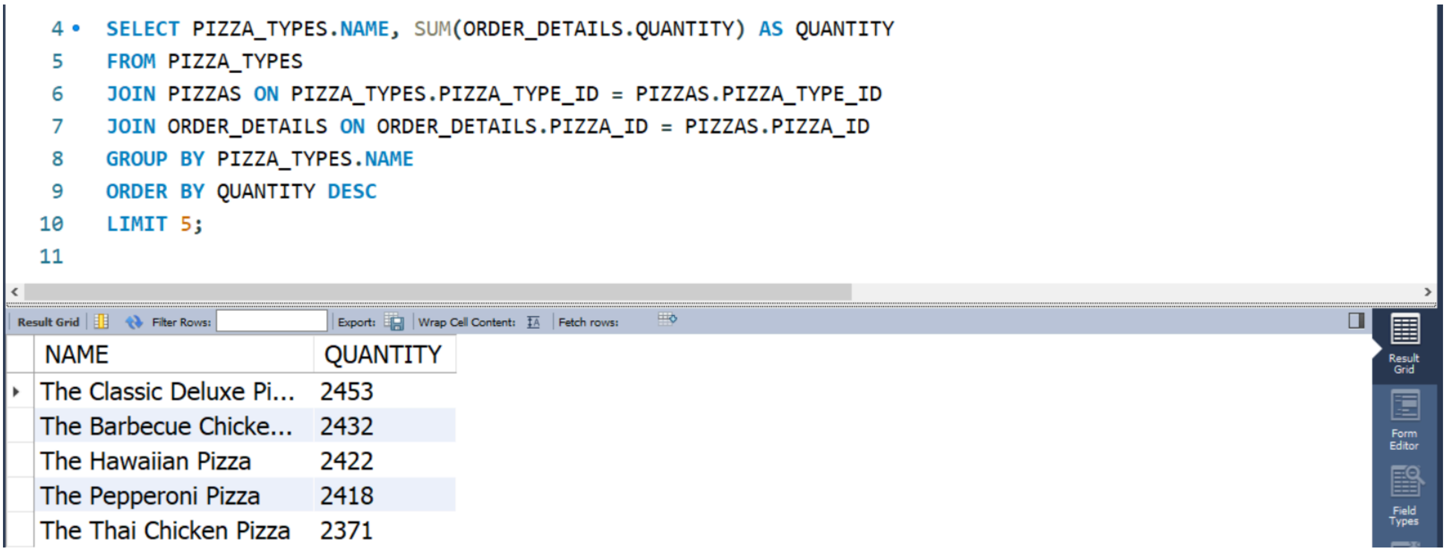
4. Identify the most common pizza size ordered.



```
SELECT 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 SIZE ORDER BY ORDER_COUNT DESC;
```

SIZE	ORDER_COUNT
L	18526
M	15385
S	14137
XL	544
XXL	28

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

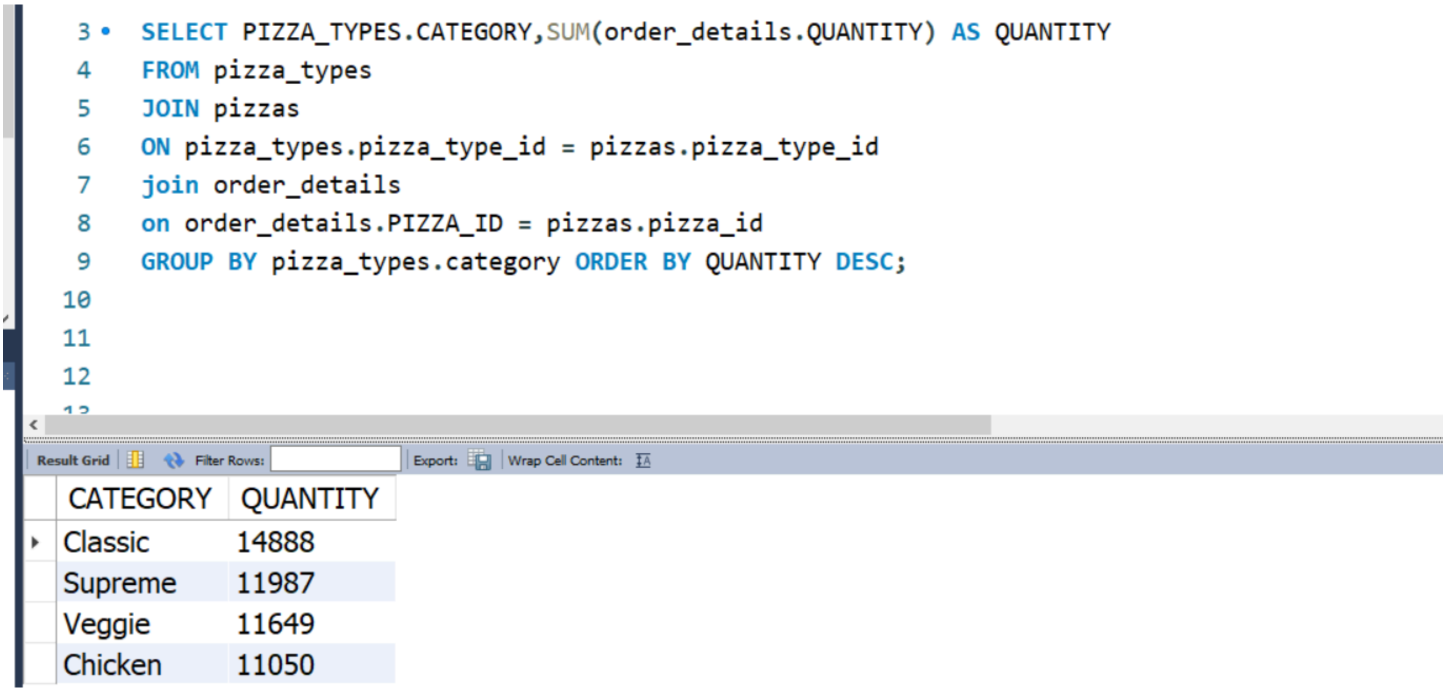


```sql
SELECT PIZZA_TYPES.NAME, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
FROM PIZZA_TYPES
JOIN PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
JOIN ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
GROUP BY PIZZA_TYPES.NAME
ORDER BY QUANTITY DESC
LIMIT 5;
```

NAME	QUANTITY
The Classic Deluxe Pi...	2453
The Barbecue Chicke...	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Intermediate:

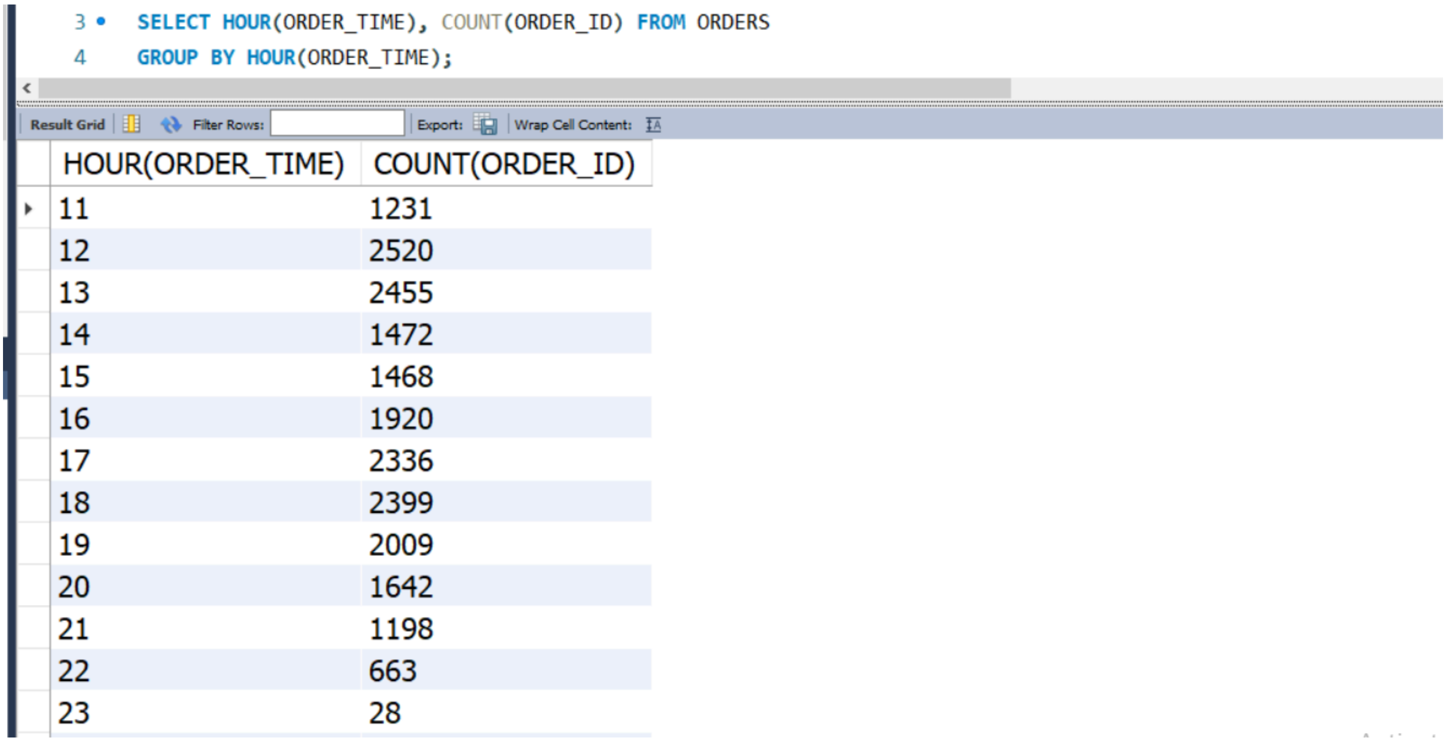
6. Join the necessary tables to find the total quantity of each pizza category ordered.



```sql
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;
```

CATEGORY	QUANTITY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

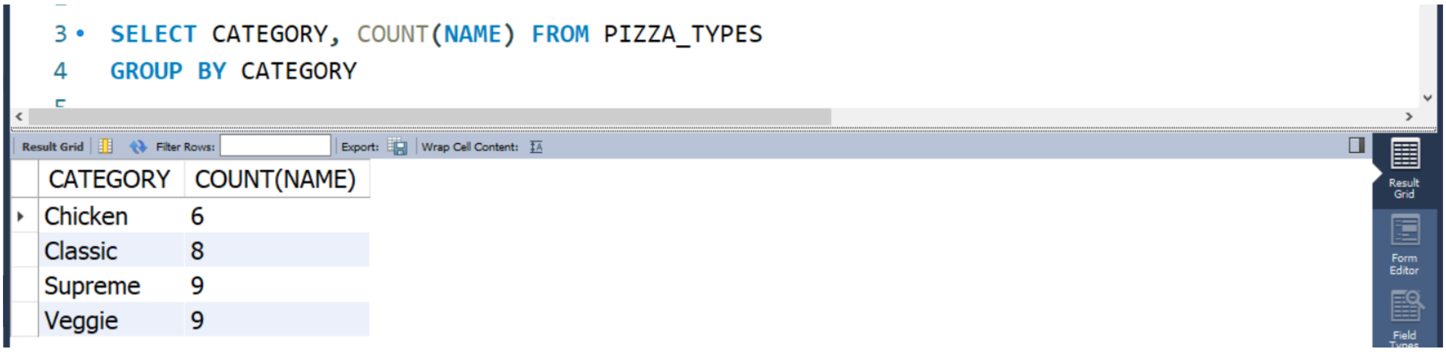
7. Determine the distribution of orders by hour of the day.



```sql
SELECT HOUR(ORDER_TIME), COUNT(ORDER_ID) FROM ORDERS
GROUP BY HOUR(ORDER_TIME);
```

HOUR(ORDER_TIME)	COUNT(ORDER_ID)
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

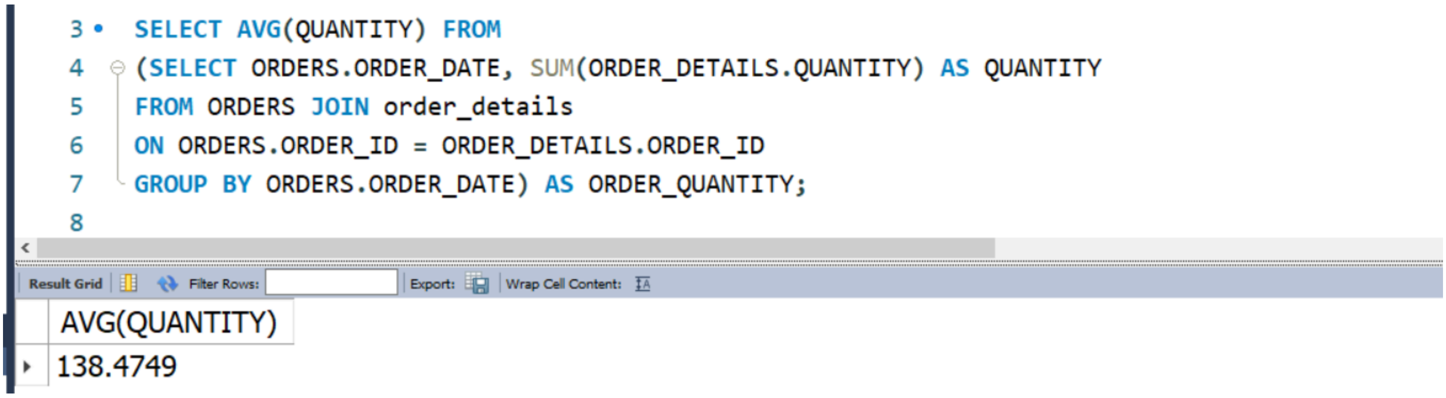
8. Join relevant tables to find the category-wise distribution of pizzas.



```sql
SELECT CATEGORY, COUNT(NAME) FROM PIZZA_TYPES
GROUP BY CATEGORY
```

CATEGORY	COUNT(NAME)
Chicken	6
Classic	8
Supreme	9
Veggie	9

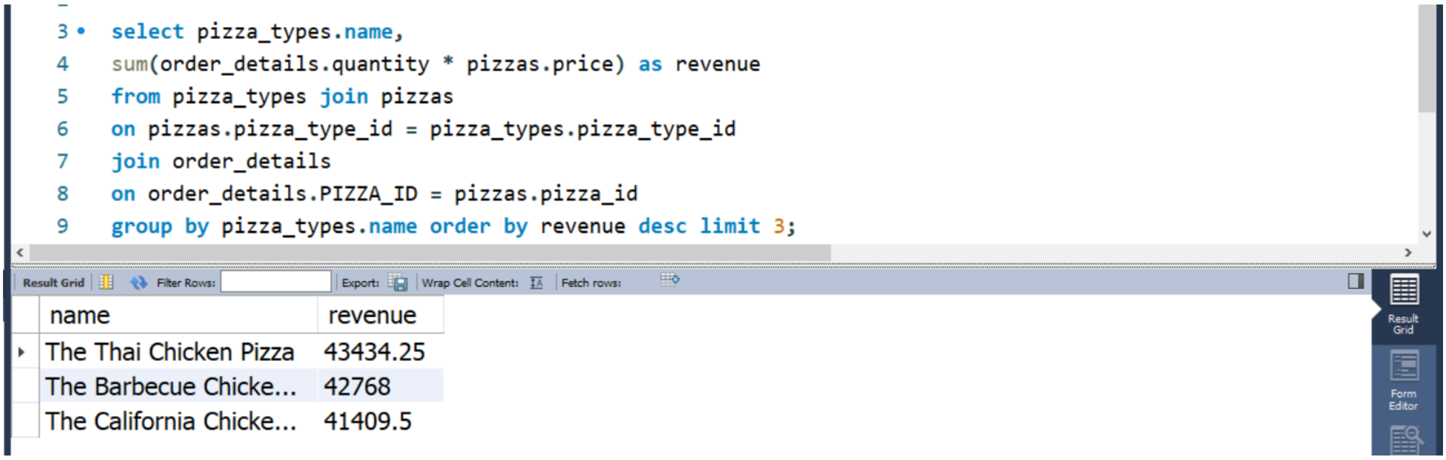
9. Group the orders by date and calculate the average number of pizzas ordered per day.



```sql
SELECT AVG(QUANTITY) 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;
```

AVG(QUANTITY)
138.4749

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

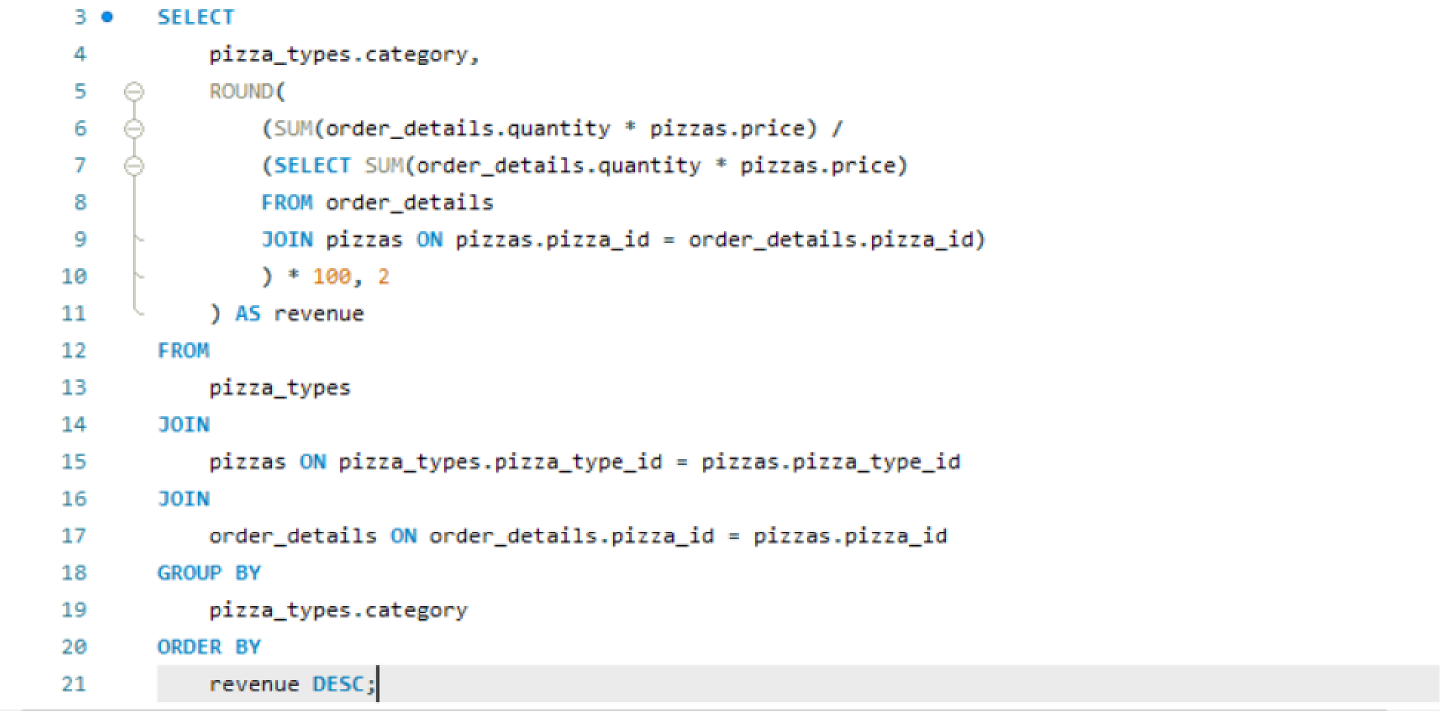


```sql
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.PIZZA_ID = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

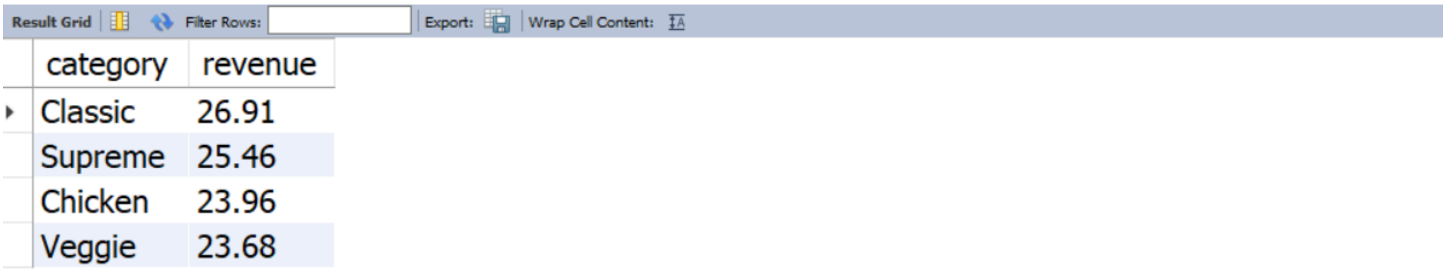
```
name                    revenue
The Thai Chicken Pizza  43434.25
The Barbecue Chicke...  42768
The California Chicke...  41409.5
```

Advanced:

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

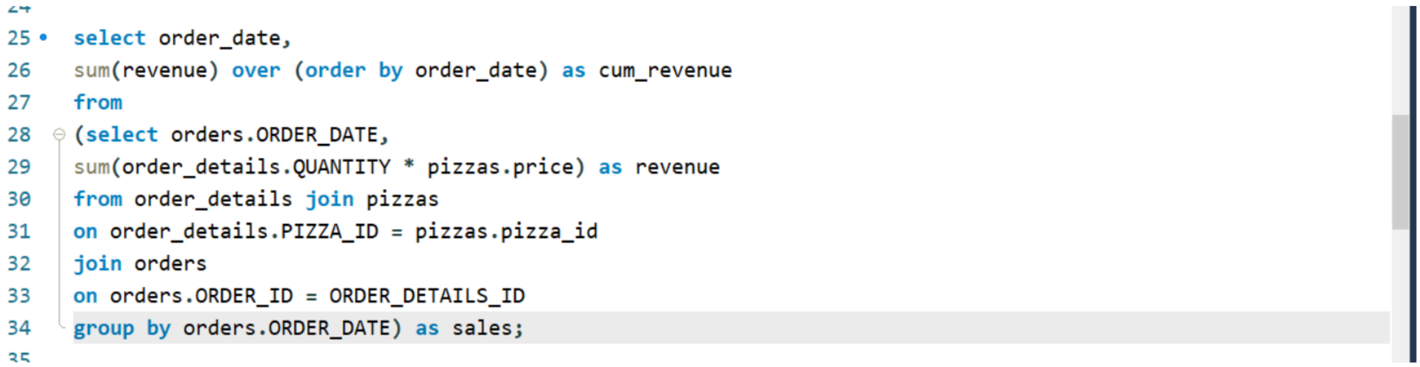


```sql
SELECT
    pizza_types.category,
    ROUND(
        (SUM(order_details.quantity * pizzas.price) /
        (SELECT SUM(order_details.quantity * pizzas.price)
        FROM order_details
        JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)
        ) * 100, 2
    ) AS revenue
FROM
    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;
```

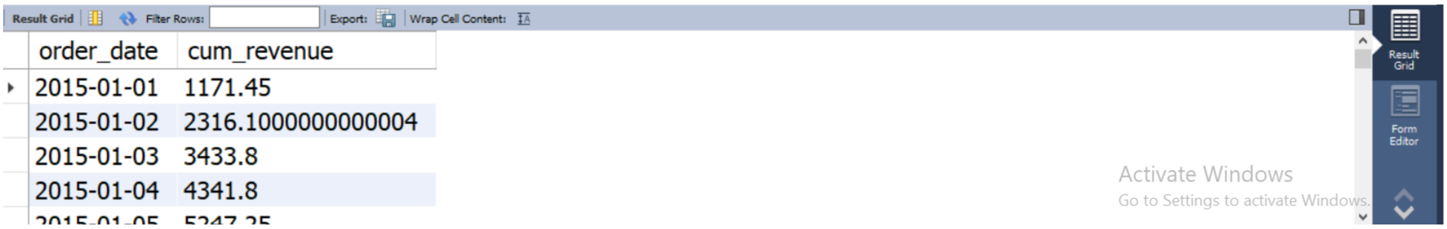


```
category  revenue
Classic   26.91
Supreme   25.46
Chicken   23.96
Veggie    23.68
```

12. Analyze the cumulative revenue generated over time.

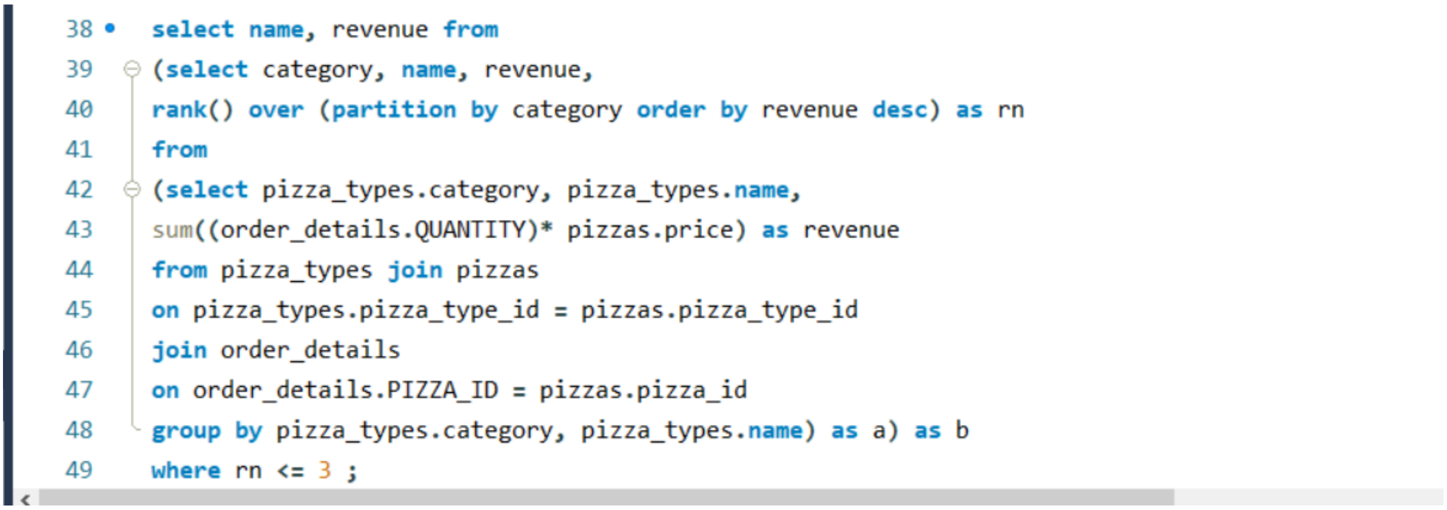


```sql
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.ORDER_DATE,
sum(order_details.QUANTITY * pizzas.price) as revenue
from order_details join pizzas
on order_details.PIZZA_ID = pizzas.pizza_id
join orders
on orders.ORDER_ID = ORDER_DETAILS_ID
group by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

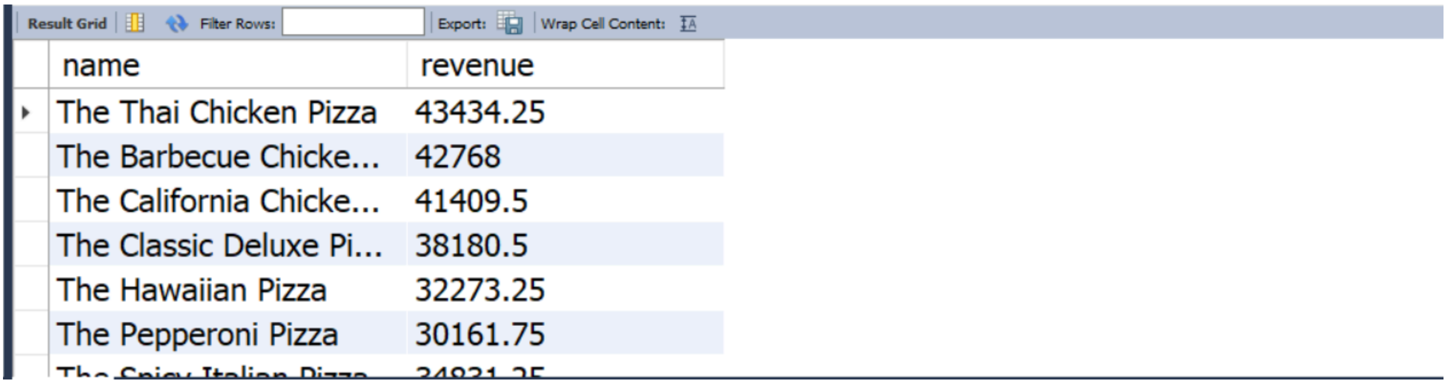


order_date	cum_revenue
2015-01-01	1171.45
2015-01-02	2316.1000000000004
2015-01-03	3433.8
2015-01-04	4341.8
2015-01-05	5247.25

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



```sql
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 ;
```



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicke...	42768
The California Chicke...	41409.5
The Classic Deluxe Pi...	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25