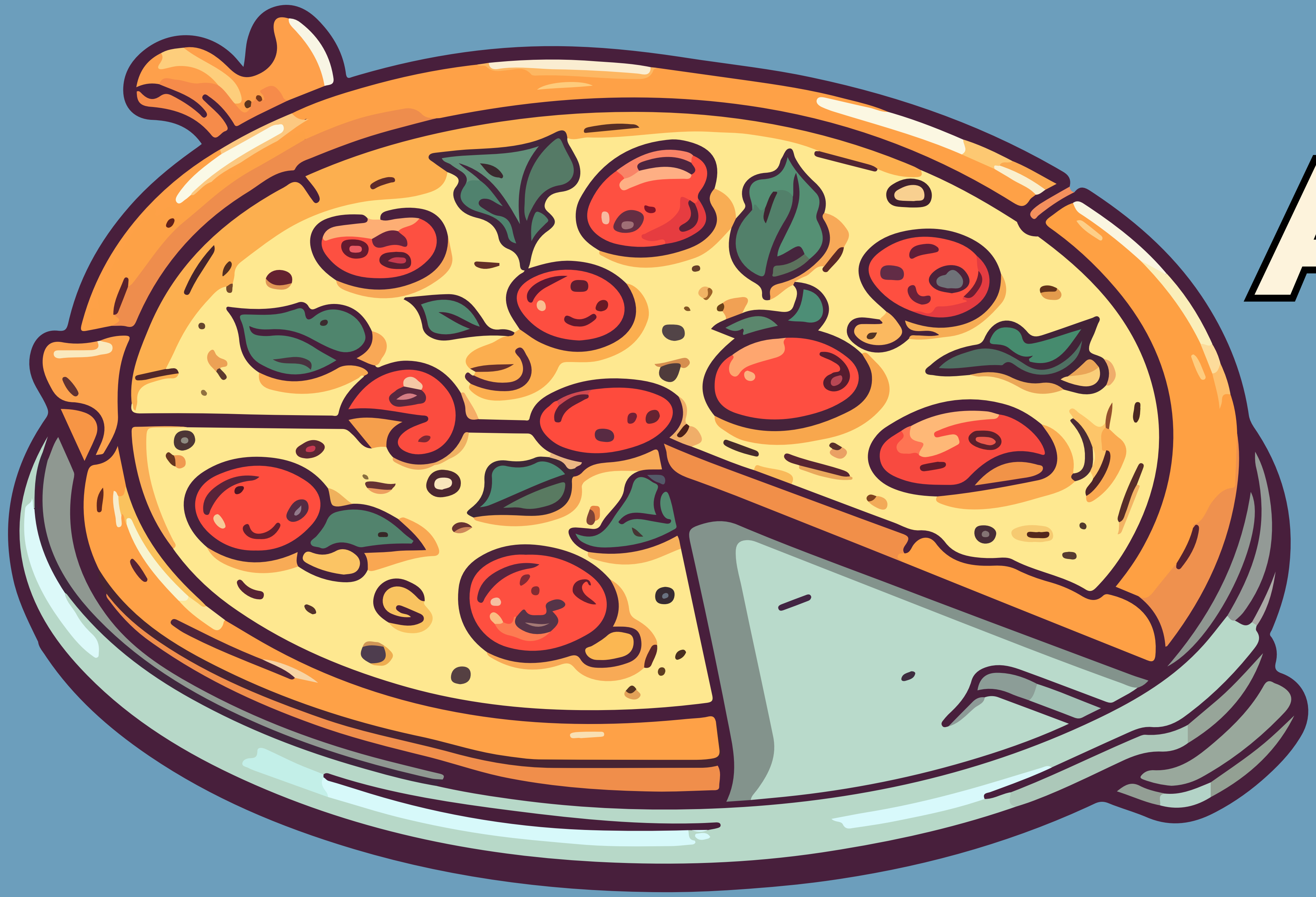


PIZZA SALES ANALYSIS

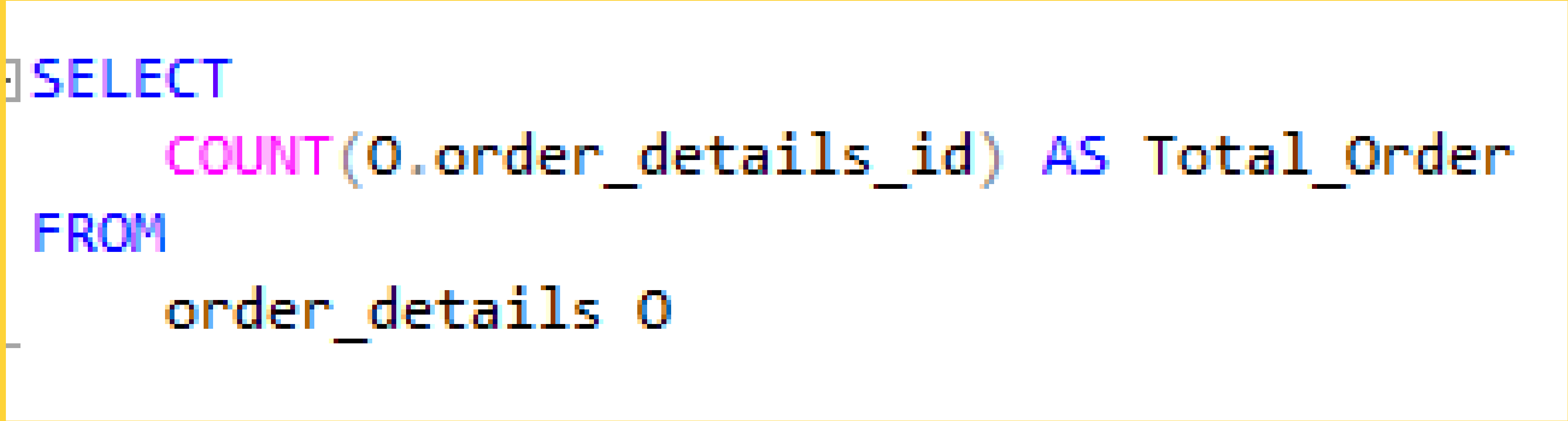
OBJECTIVE FOR PIZZA SALES ANALYSIS USING SQL:

The primary objective of this project is to leverage SQL to analyze pizza sales data and uncover key insights that can drive business improvements. This analysis aims to:

1. Extract and Analyze Sales Data using SQL queries to identify the top-selling pizzas and overall sales performance.
2. Identify Sales Trends by querying time-based data (daily, weekly, monthly) to understand customer ordering patterns.
3. Evaluate Pizza Size Preferences by using SQL to compare the sales of small, medium, and large pizzas.
4. Determine Peak Sales Periods by analyzing timestamps in the data to find peak ordering hours and days.

1

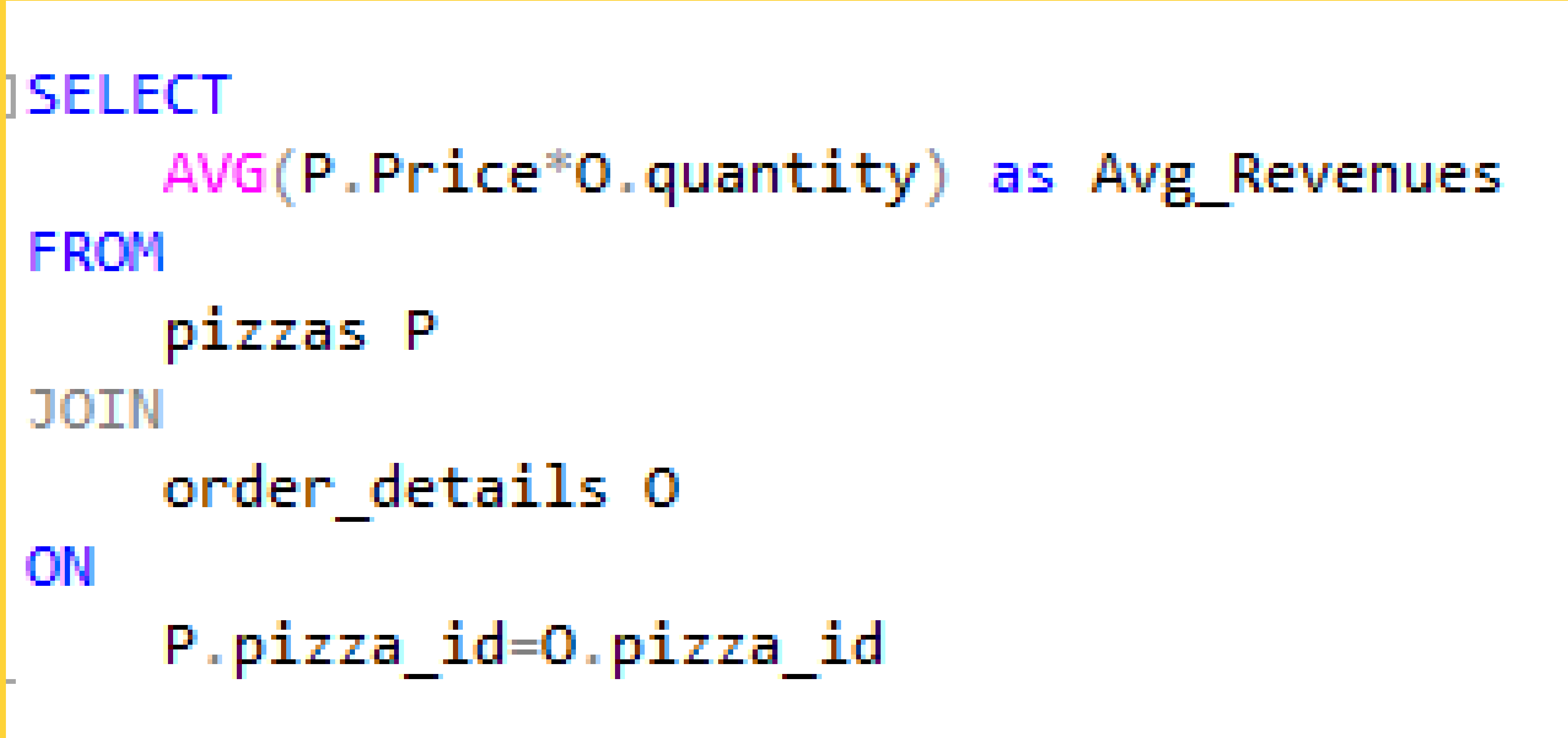
Retrieve the total number of orders placed.



```
SELECT
    COUNT(O.order_details_id) AS Total_Order
FROM
    order_details O
```

2

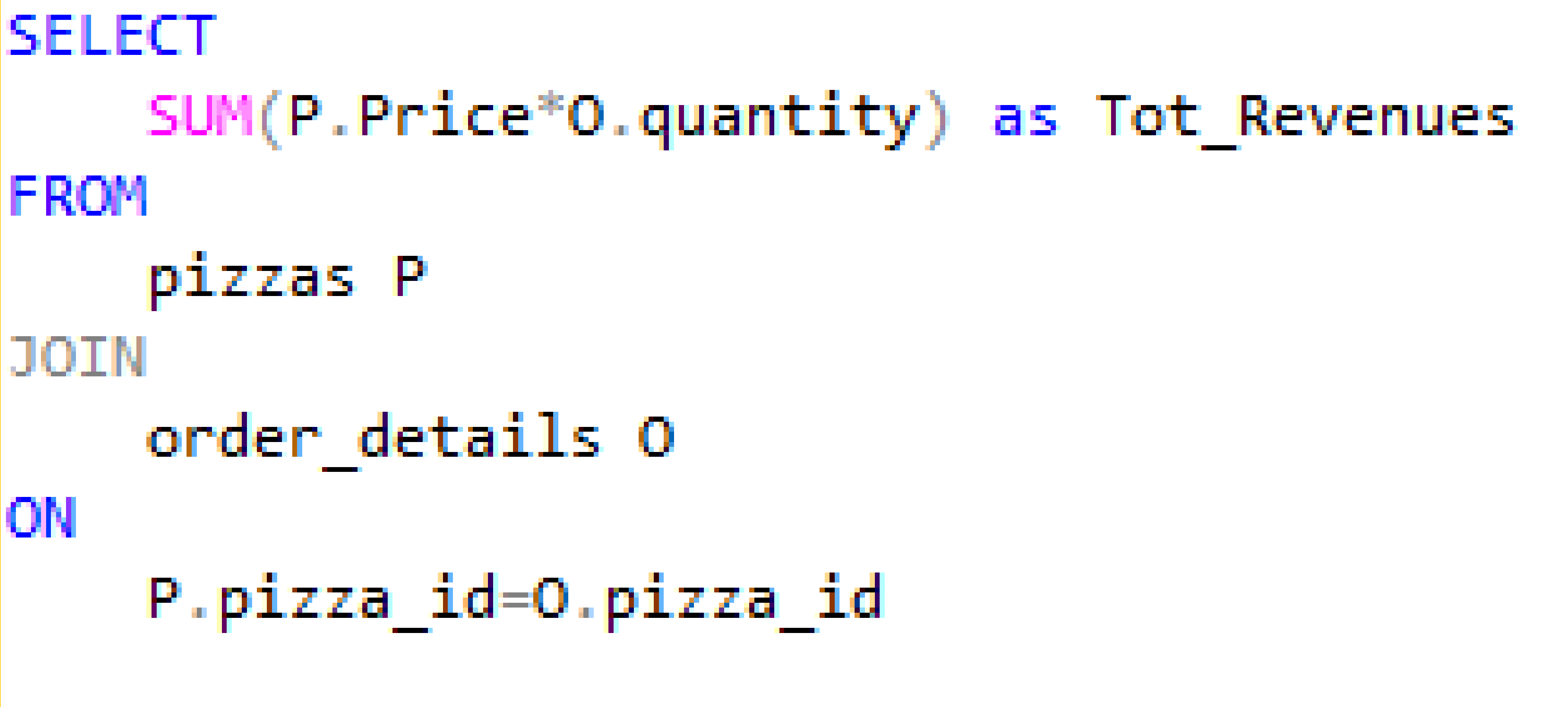
Calculate the average revenue from pizza sales.



```
SELECT
    AVG(P.Price*O.quantity) as Avg_Revenues
FROM
    pizzas P
JOIN
    order_details O
ON
    P.pizza_id=O.pizza_id
```

3

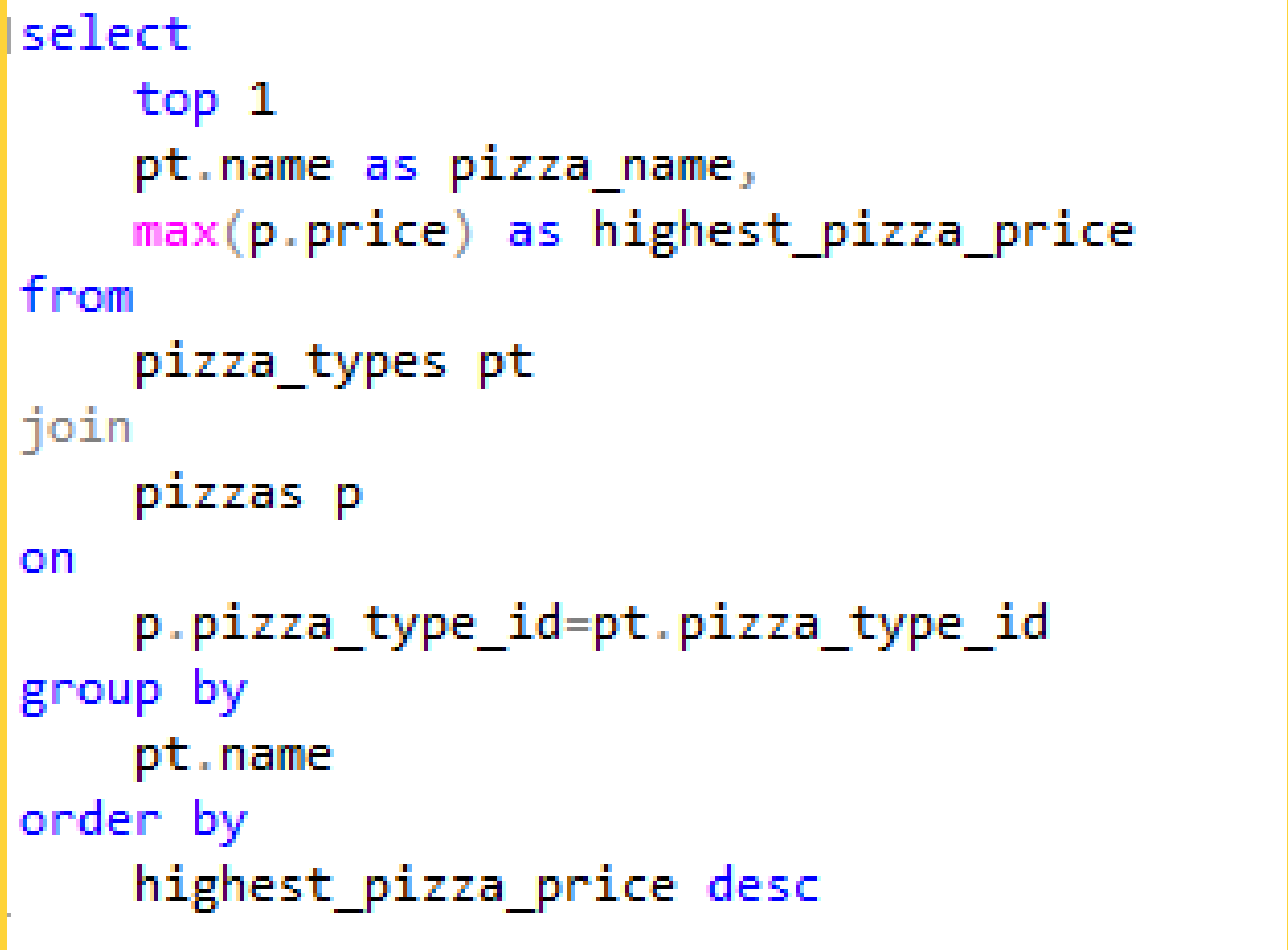
Calculate the total revenue generated from pizza sales.



```
SELECT
    SUM(P.Price*O.quantity) as Tot_Revenues
FROM
    pizzas P
JOIN
    order_details O
ON
    P.pizza_id=O.pizza_id
```

4

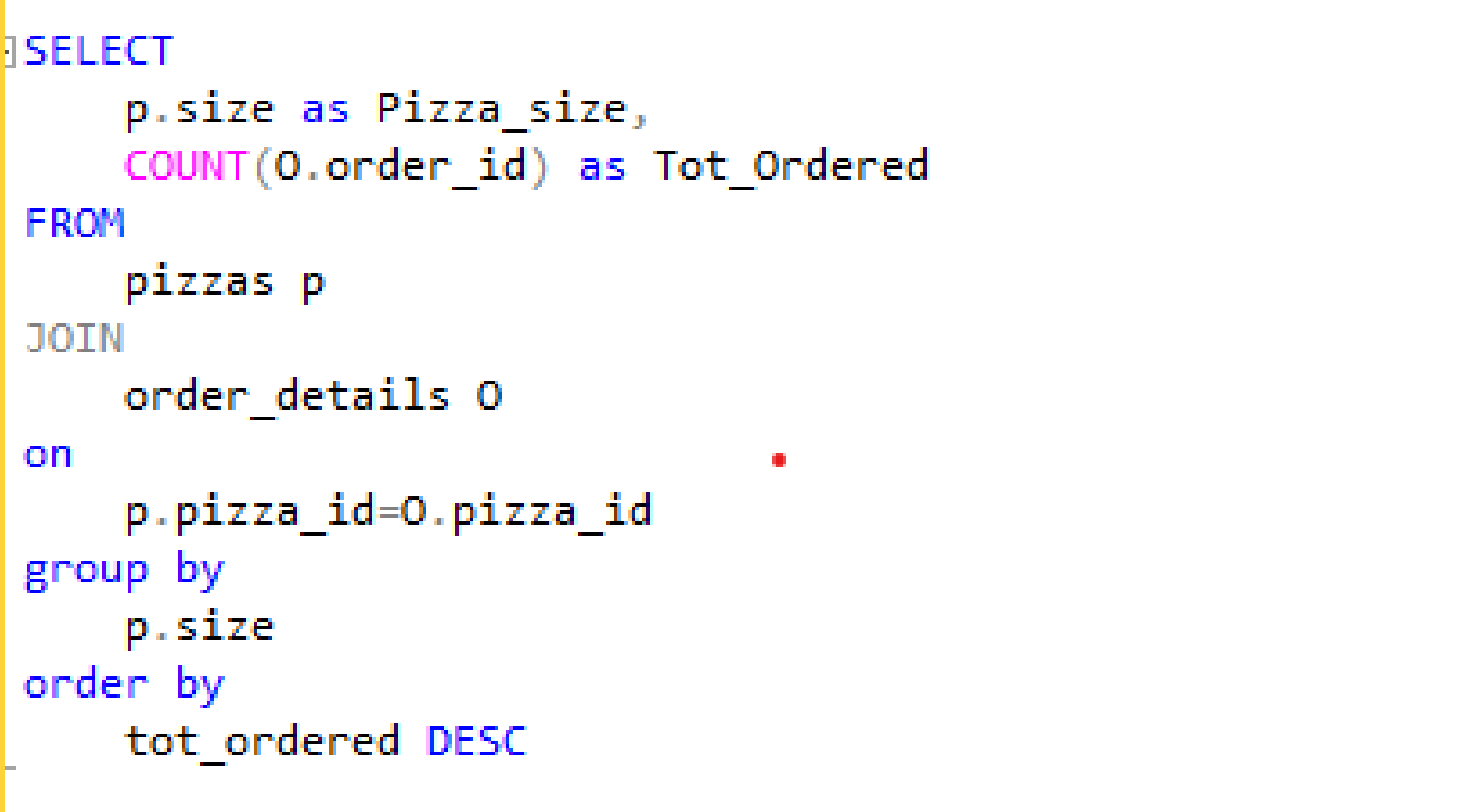
Identify the highest-priced pizza.



```
select
    top 1
    pt.name as pizza_name,
    max(p.price) as highest_pizza_price
from
    pizza_types pt
join
    pizzas p
on
    p.pizza_type_id=pt.pizza_type_id
group by
    pt.name
order by
    highest_pizza_price desc
```

5

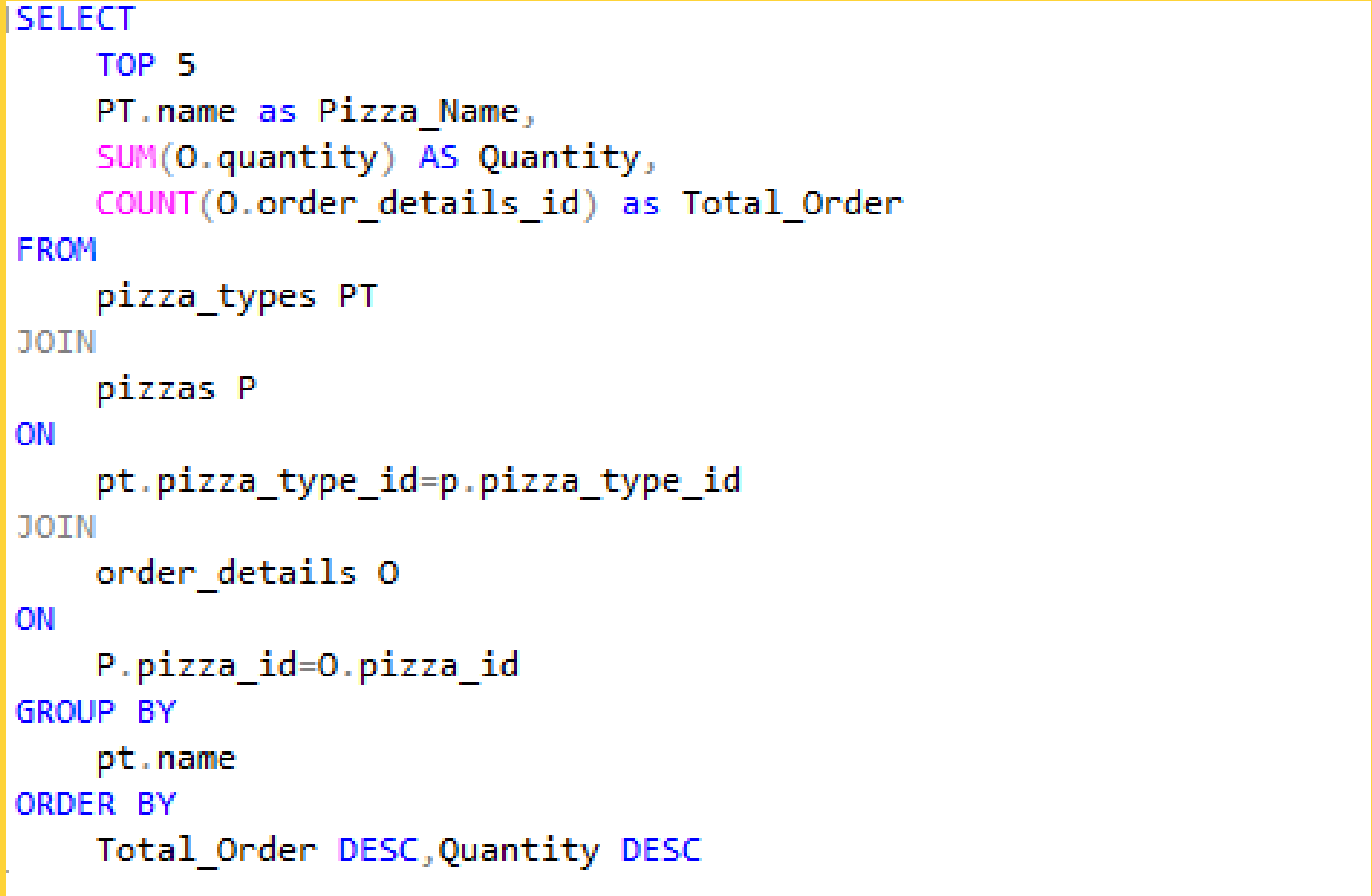
Identify the most common pizza size ordered.



```
SELECT
    p.size as Pizza_size,
    COUNT(O.order_id) as Tot_Ordered
FROM
    pizzas p
JOIN
    order_details O
on
    p.pizza_id=O.pizza_id
group by
    p.size
order by
    tot_ordered DESC
```

6

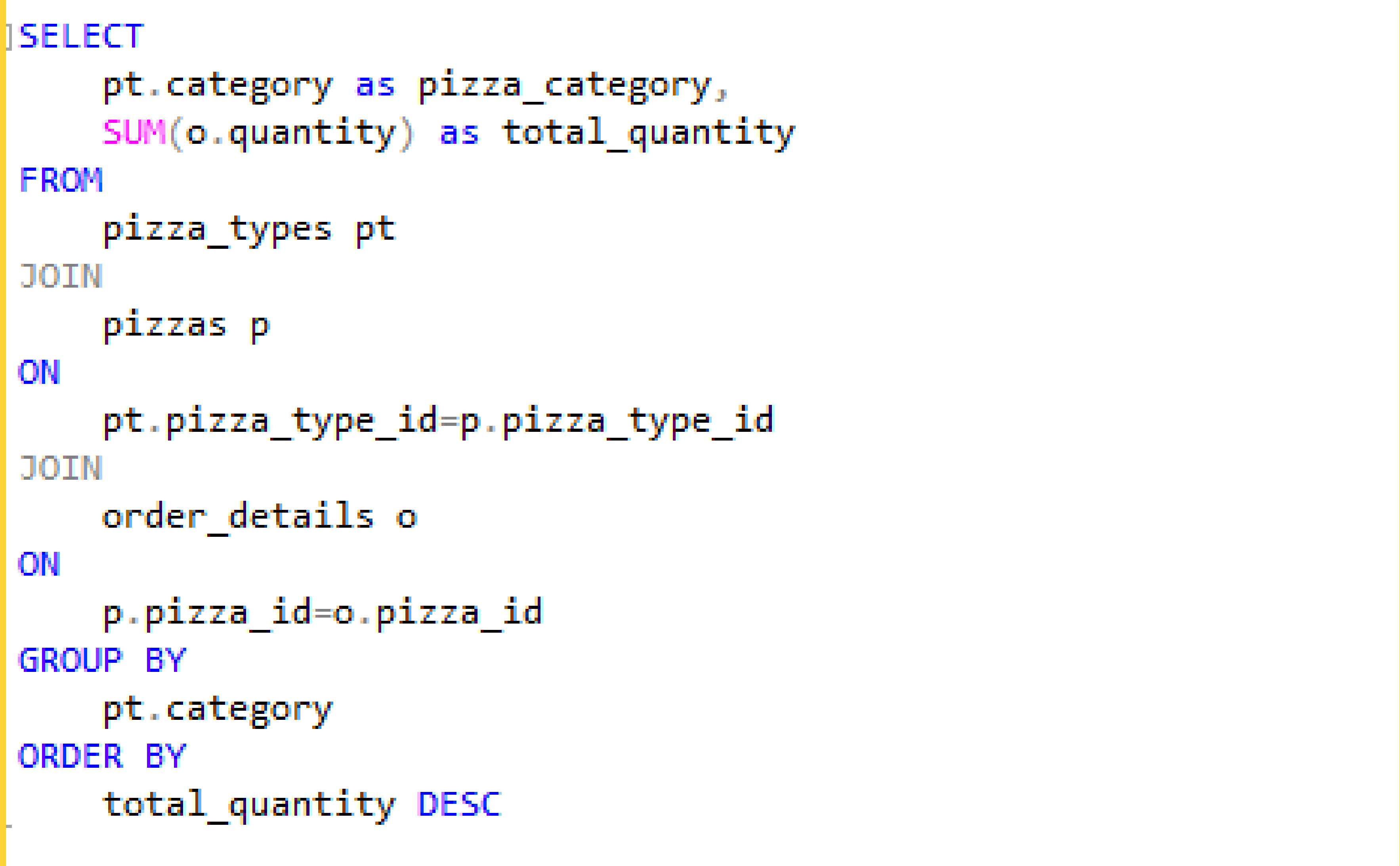
List the top 5 most ordered pizza types along with their quantities.



```
SELECT
    TOP 5
    PT.name as Pizza_Name,
    SUM(O.quantity) AS Quantity,
    COUNT(O.order_details_id) as Total_Order
FROM
    pizza_types PT
JOIN
    pizzas P
ON
    pt.pizza_type_id=p.pizza_type_id
JOIN
    order_details O
ON
    P.pizza_id=O.pizza_id
GROUP BY
    pt.name
ORDER BY
    Total_Order DESC,Quantity DESC
```

7

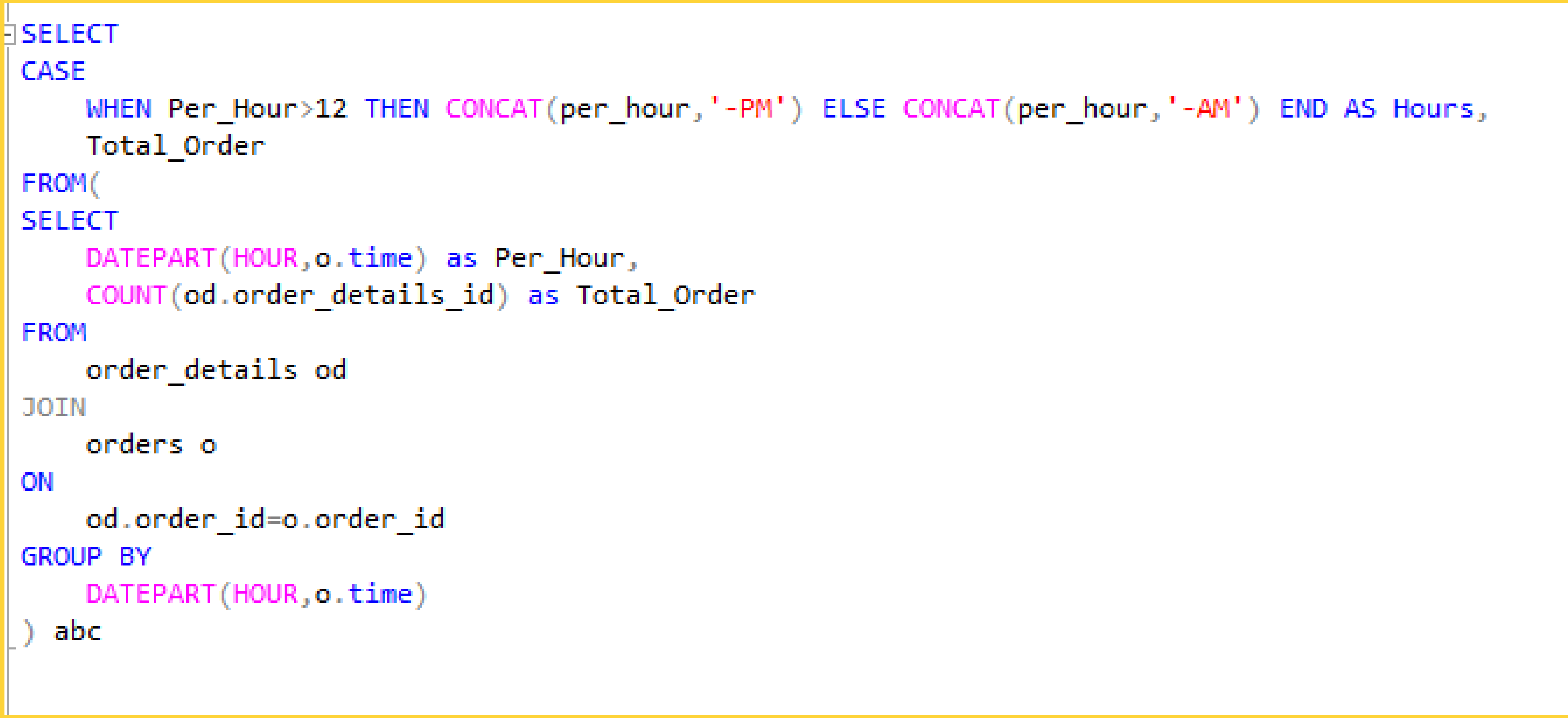
Join the necessary tables to find the total quantity of each pizza category ordered.



```
SELECT
    pt.category as pizza_category,
    SUM(o.quantity) as total_quantity
FROM
    pizza_types pt
JOIN
    pizzas p
ON
    pt.pizza_type_id=p.pizza_type_id
JOIN
    order_details o
ON
    p.pizza_id=o.pizza_id
GROUP BY
    pt.category
ORDER BY
    total_quantity DESC
```

8

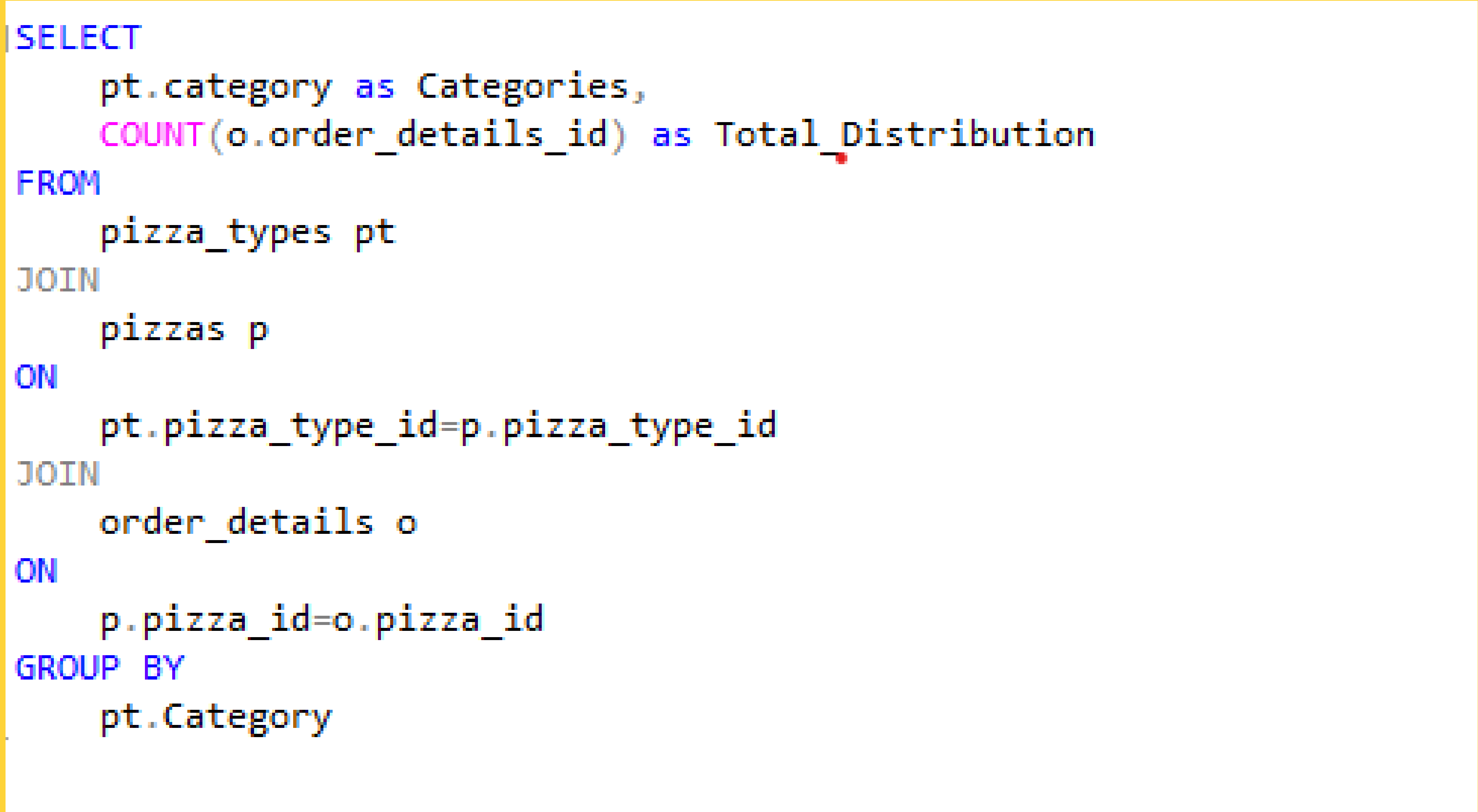
Determine the distribution of orders by hour of the day.



```
SELECT
CASE
    WHEN Per_Hour>12 THEN CONCAT(per_hour,'-PM') ELSE CONCAT(per_hour,'-AM') END AS Hours,
    Total_Order
FROM(
SELECT
    DATEPART(HOUR,o.time) as Per_Hour,
    COUNT(od.order_details_id) as Total_Order
FROM
    order_details od
JOIN
    orders o
ON
    od.order_id=o.order_id
GROUP BY
    DATEPART(HOUR,o.time)
) abc
```

9

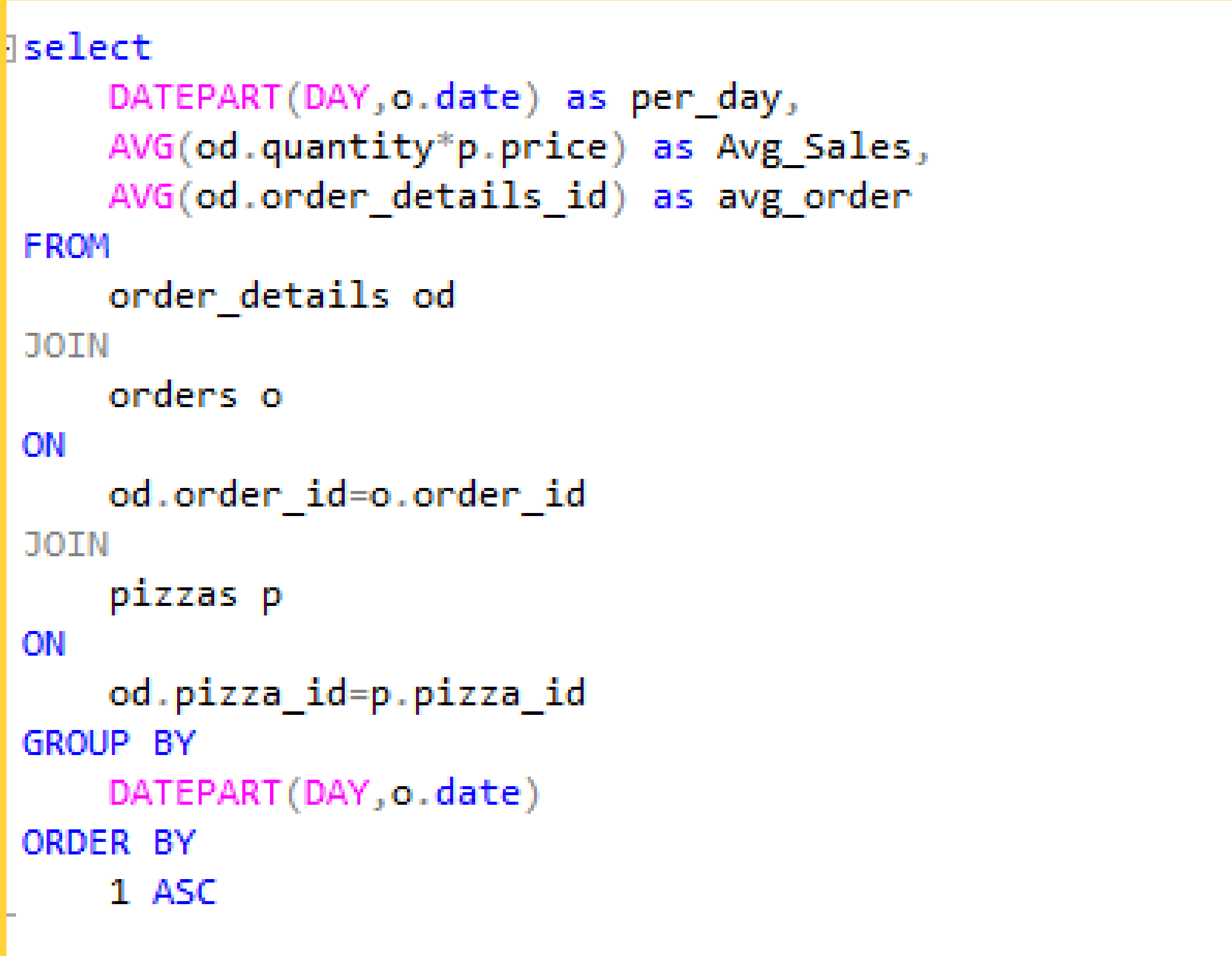
Find the category-wise distribution of pizzas.



```
SELECT
    pt.category as Categories,
    COUNT(o.order_details_id) as Total_Distribution
FROM
    pizza_types pt
JOIN
    pizzas p
ON
    pt.pizza_type_id=p.pizza_type_id
JOIN
    order_details o
ON
    p.pizza_id=o.pizza_id
GROUP BY
    pt.Category
```

10

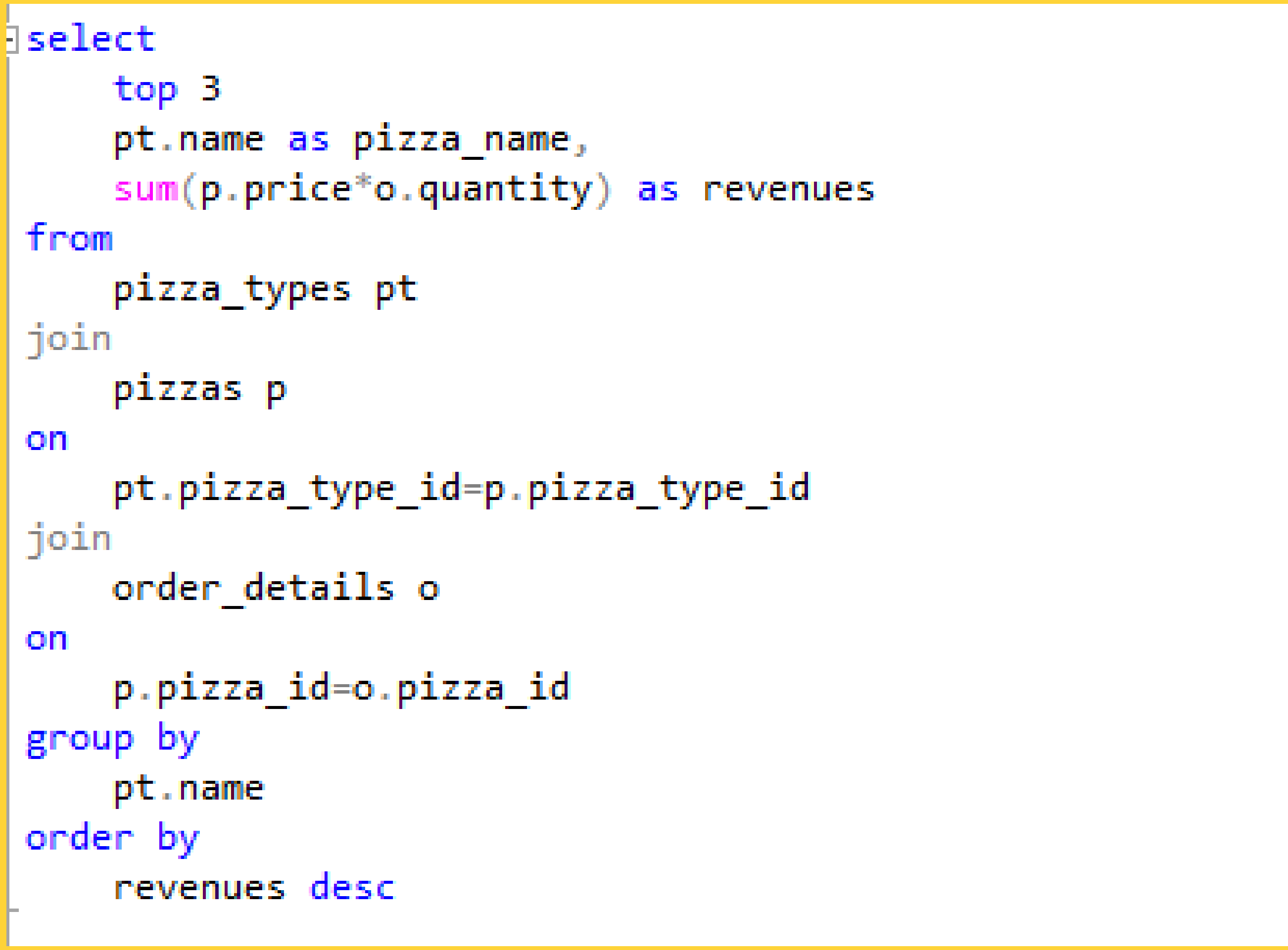
Group the orders by date and calculate the average number of pizzas ordered per day



```
select
    DATEPART(DAY,o.date) as per_day,
    AVG(od.quantity*p.price) as Avg_Sales,
    AVG(od.order_details_id) as avg_order
FROM
    order_details od
JOIN
    orders o
ON
    od.order_id=o.order_id
JOIN
    pizzas p
ON
    od.pizza_id=p.pizza_id
GROUP BY
    DATEPART(DAY,o.date)
ORDER BY
    1 ASC
```

11

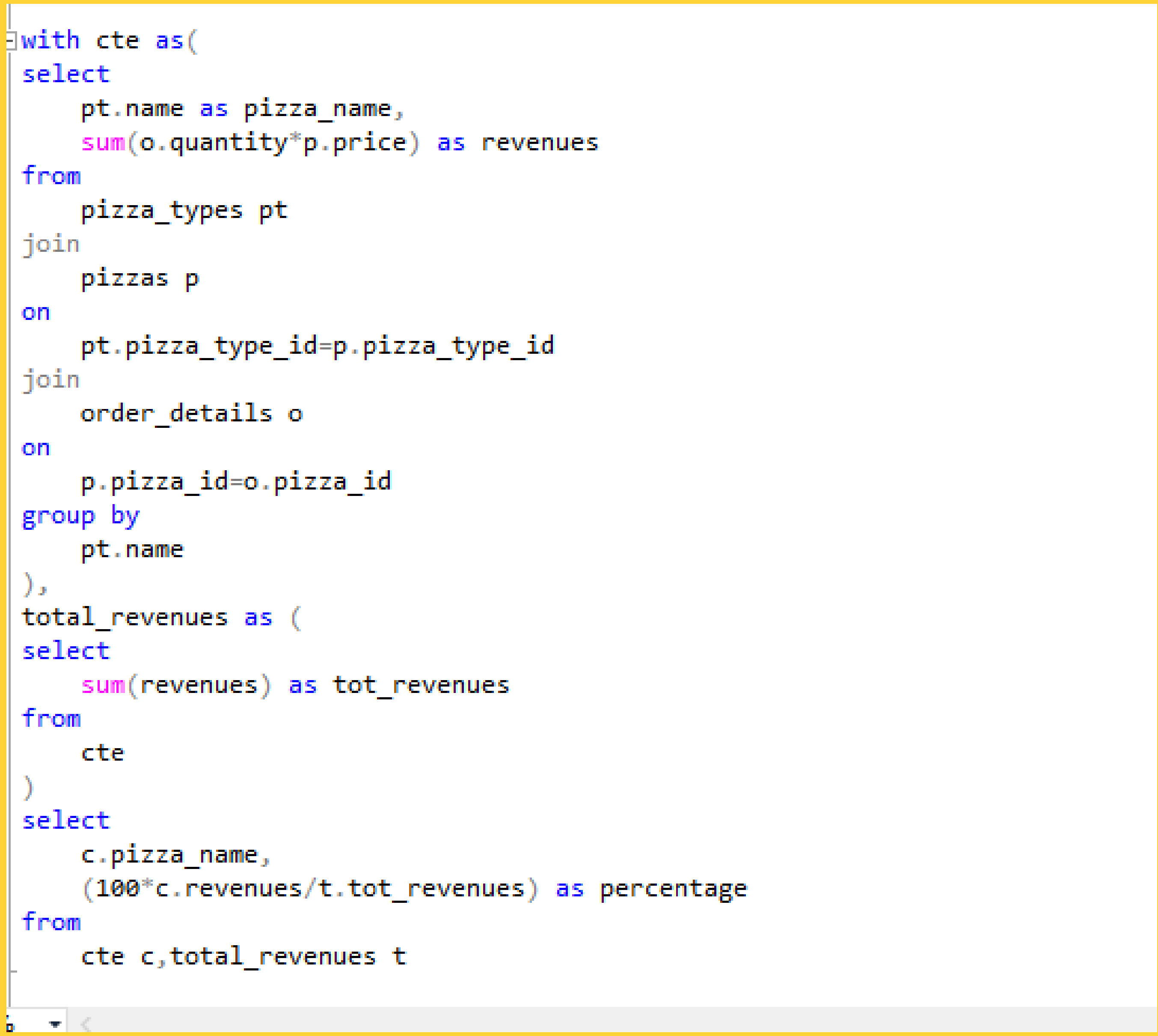
Determine the top 3 most ordered pizza types based on revenue.



```sql
select
    top 3
    pt.name as pizza_name,
    sum(p.price*o.quantity) as revenues
from
    pizza_types pt
join
    pizzas p
on
    pt.pizza_type_id=p.pizza_type_id
join
    order_details o
on
    p.pizza_id=o.pizza_id
group by
    pt.name
order by
    revenues desc
```

12

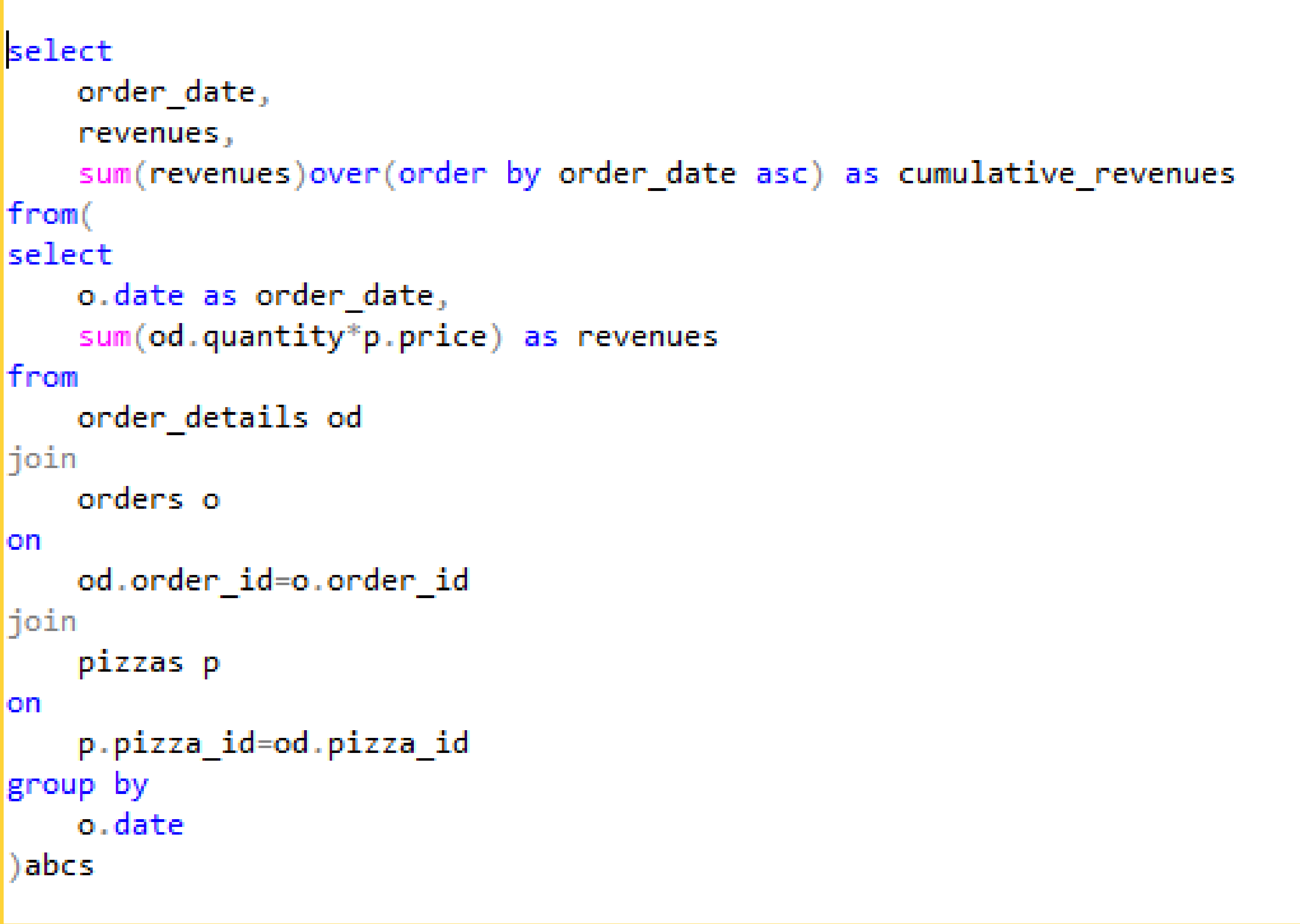
Calculate the percentage contribution of each pizza type to total revenue.



```sql
with cte as(
select
    pt.name as pizza_name,
    sum(o.quantity*p.price) as revenues
from
    pizza_types pt
join
    pizzas p
on
    pt.pizza_type_id=p.pizza_type_id
join
    order_details o
on
    p.pizza_id=o.pizza_id
group by
    pt.name
),
total_revenues as (
select
    sum(revenues) as tot_revenues
from
    cte
)
select
    c.pizza_name,
    (100*c.revenues/t.tot_revenues) as percentage
from
    cte c,total_revenues t
```

13

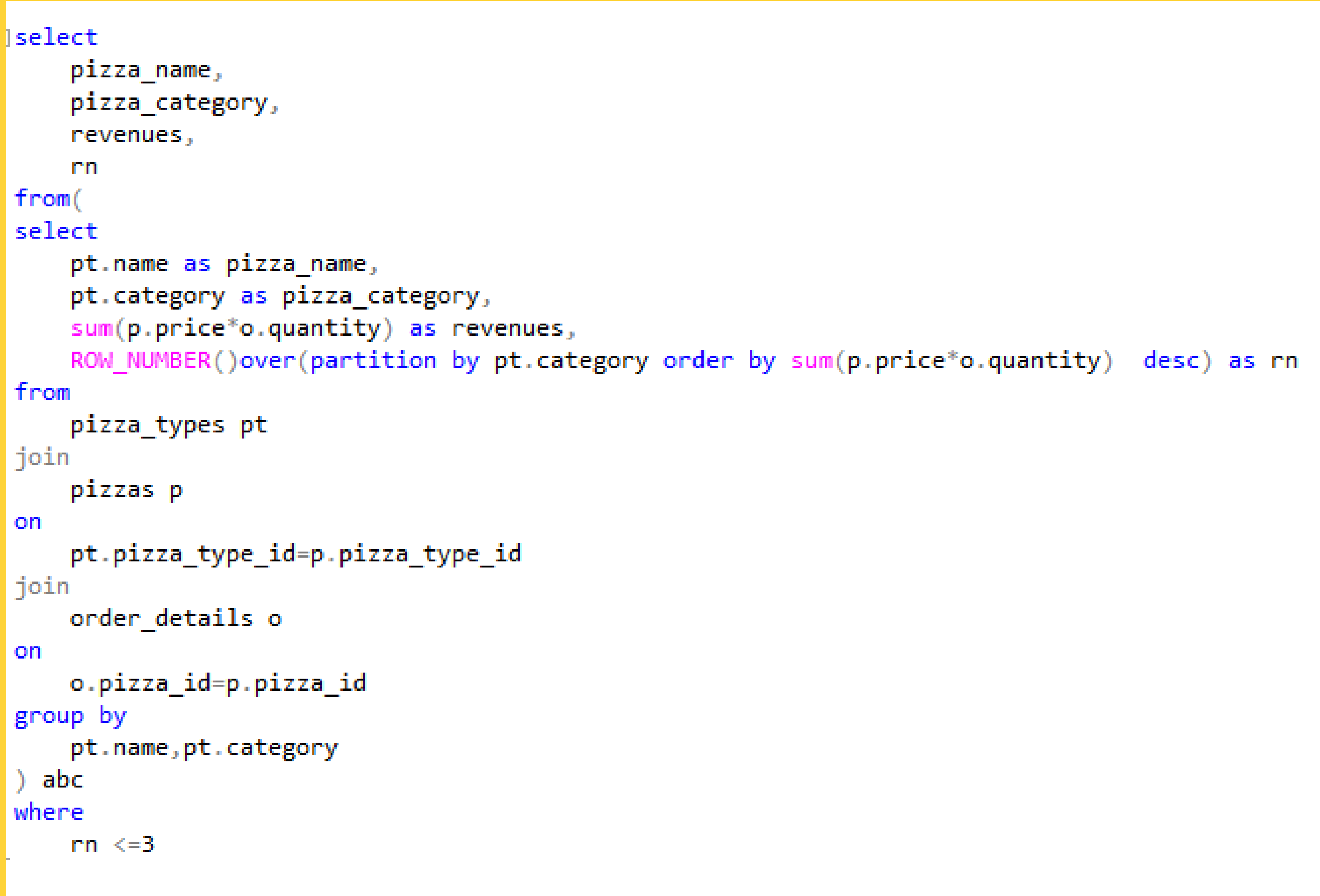
Analyze the cumulative revenue generated over time.



```
select
    order_date,
    revenues,
    sum(revenues)over(order by order_date asc) as cumulative_revenues
from(
select
    o.date as order_date,
    sum(od.quantity*p.price) as revenues
from
    order_details od
join
    orders o
on
    od.order_id=o.order_id
join
    pizzas p
on
    p.pizza_id=od.pizza_id
group by
    o.date
)abcs
```

14

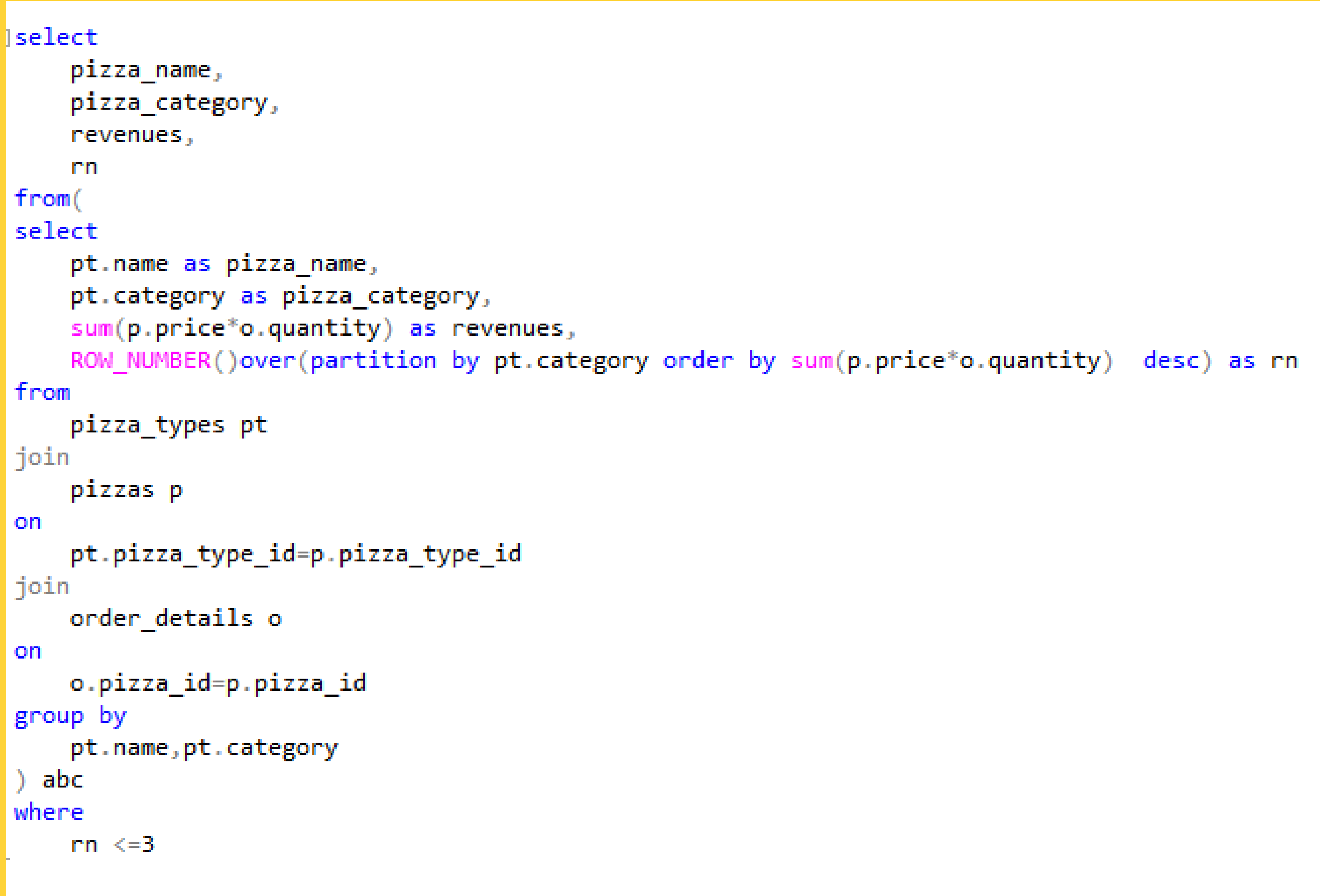
Determine the top 3 most ordered pizza types based on revenue for each pizza category



```
select
    pizza_name,
    pizza_category,
    revenues,
    rn
from(
select
    pt.name as pizza_name,
    pt.category as pizza_category,
    sum(p.price*o.quantity) as revenues,
    ROW_NUMBER()over(partition by pt.category order by sum(p.price*o.quantity)  desc) as rn
from
    pizza_types pt
join
    pizzas p
on
    pt.pizza_type_id=p.pizza_type_id
join
    order_details o
on
    o.pizza_id=p.pizza_id
group by
    pt.name,pt.category
) abc
where
    rn <=3
```

15

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



```
select
    pizza_name,
    pizza_category,
    revenues,
    rn
from(
select
    pt.name as pizza_name,
    pt.category as pizza_category,
    sum(p.price*o.quantity) as revenues,
    ROW_NUMBER()over(partition by pt.category order by sum(p.price*o.quantity)  desc) as rn
from
    pizza_types pt
join
    pizzas p
on
    pt.pizza_type_id=p.pizza_type_id
join
    order_details o
on
    o.pizza_id=p.pizza_id
group by
    pt.name,pt.category
) abc
where
    rn <=3
```

THANK YOU