ANURUP GOLUI

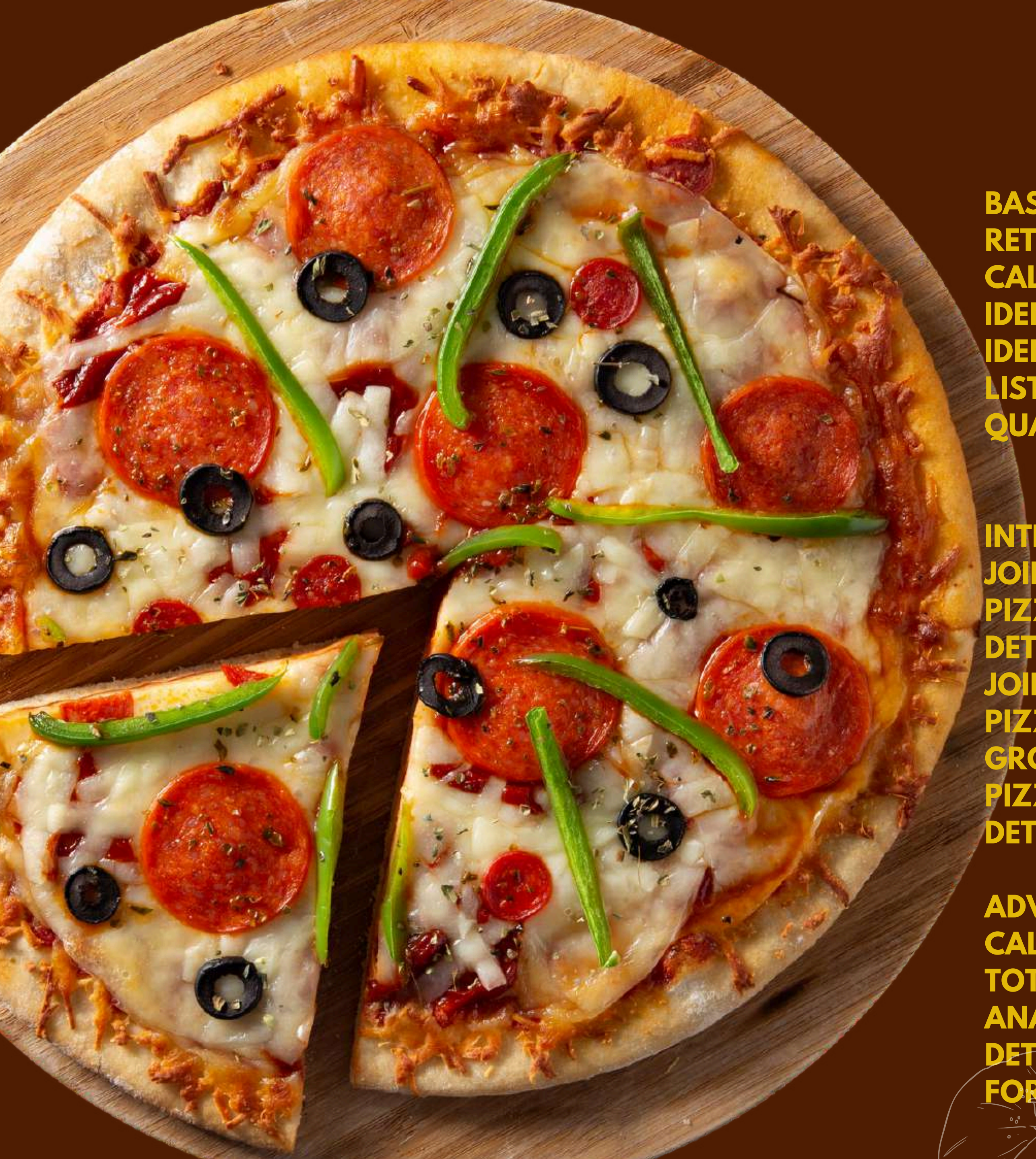
PRESENTS

SQL PROJECT

PIZZA

SALES ANALYSIS

WATCH NOW



QUESTIONS

BASIC:
RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
IDENTIFY THE HIGHEST-PRICED PIZZA.
IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

INTERMEDIATE:
JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

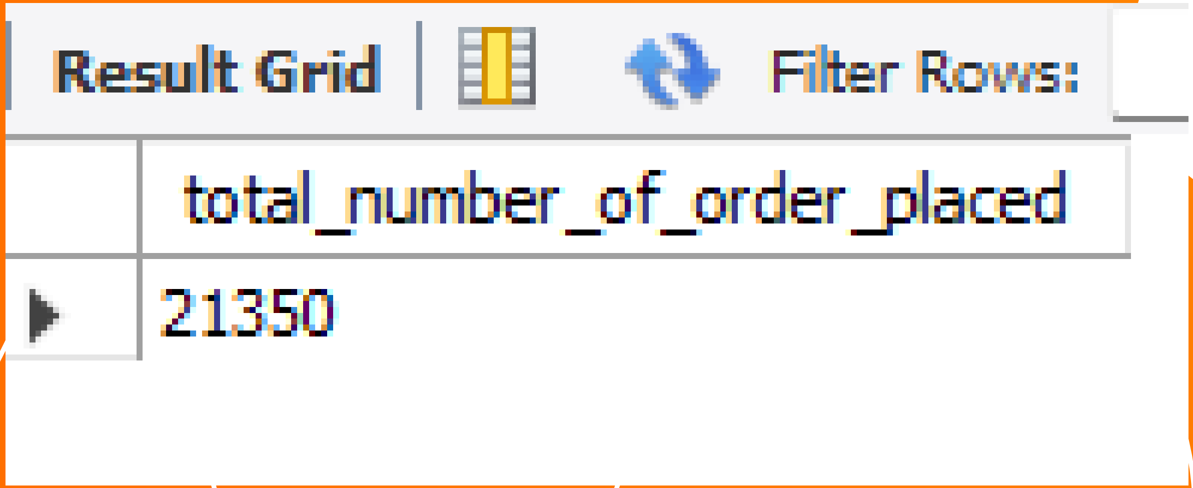
ADVANCED:
CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

QUESTION

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

ANSWER

```
SELECT COUNT(ORDER_ID) AS
TOTAL_NUMBER_OF_ORDER_PLACED
FROM ORDERS;
```

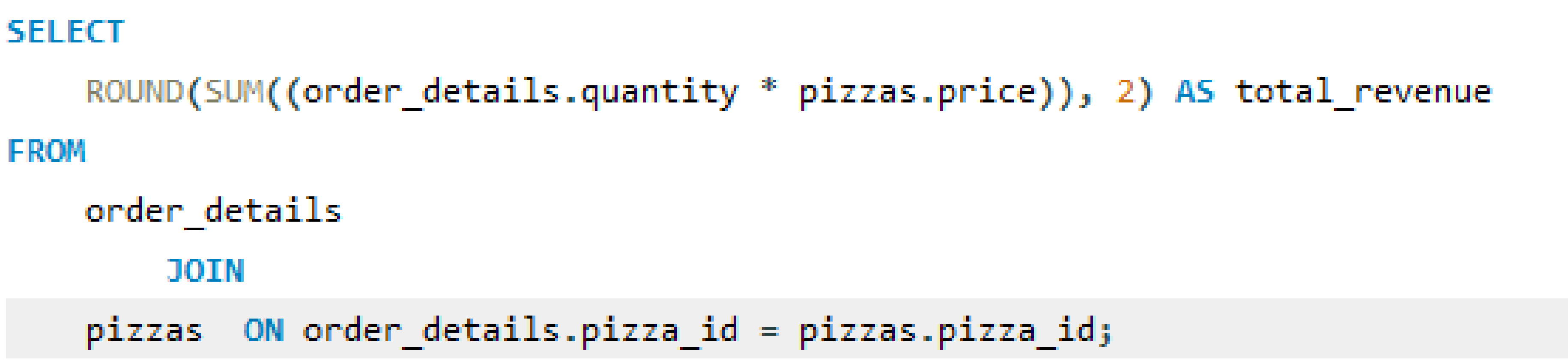


total_number_of_order_placed
21350

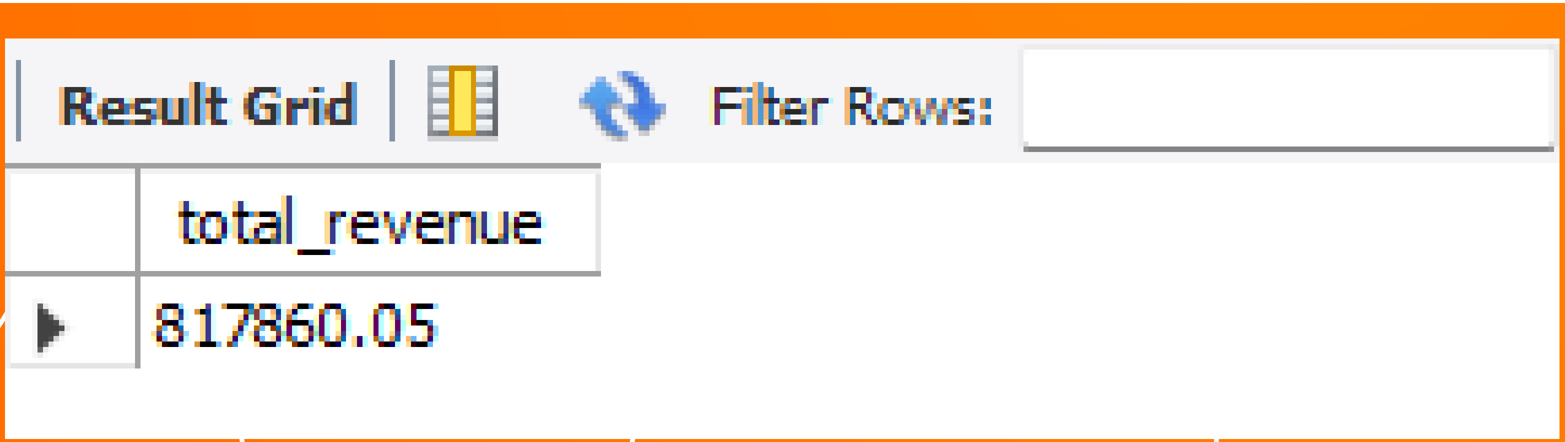
QUESTION

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

ANSWER



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

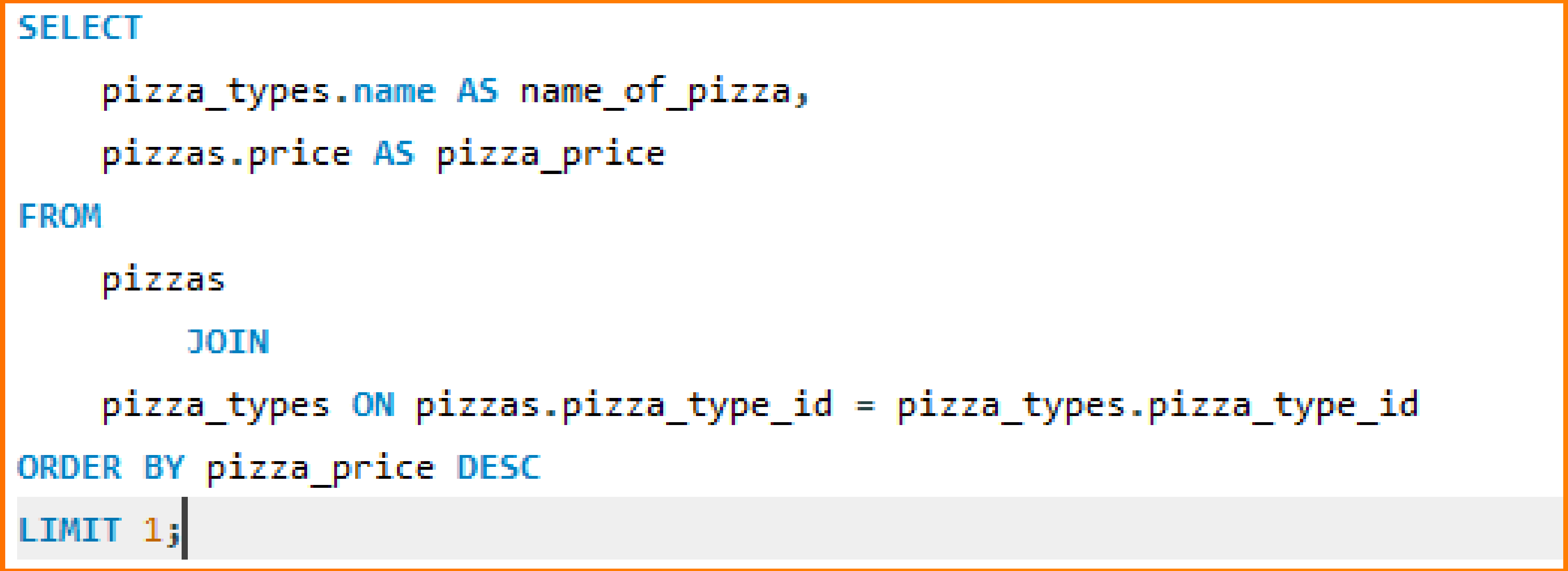


```
total_revenue
817860.05
```

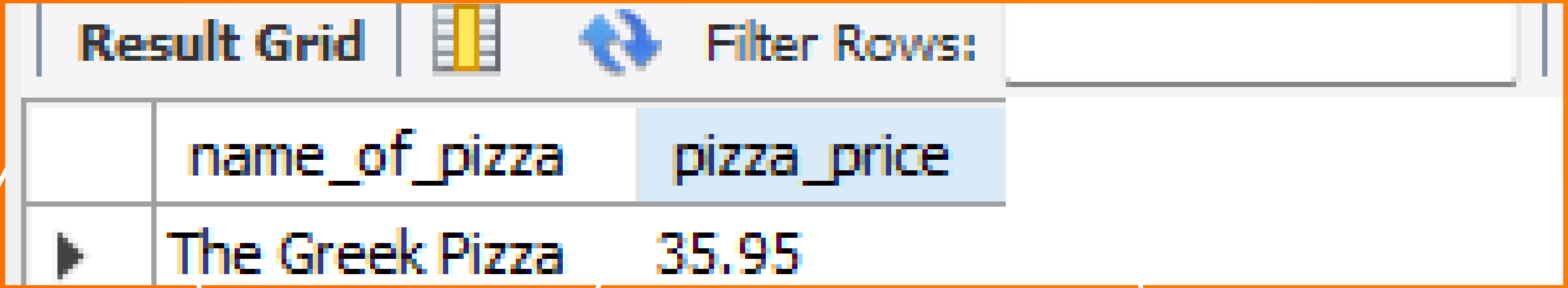
QUESTION

IDENTIFY THE HIGHEST-PRICED PIZZA.

ANSWER



```
SELECT
    pizza_types.name AS name_of_pizza,
    pizzas.price AS pizza_price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizza_price DESC
LIMIT 1;
```

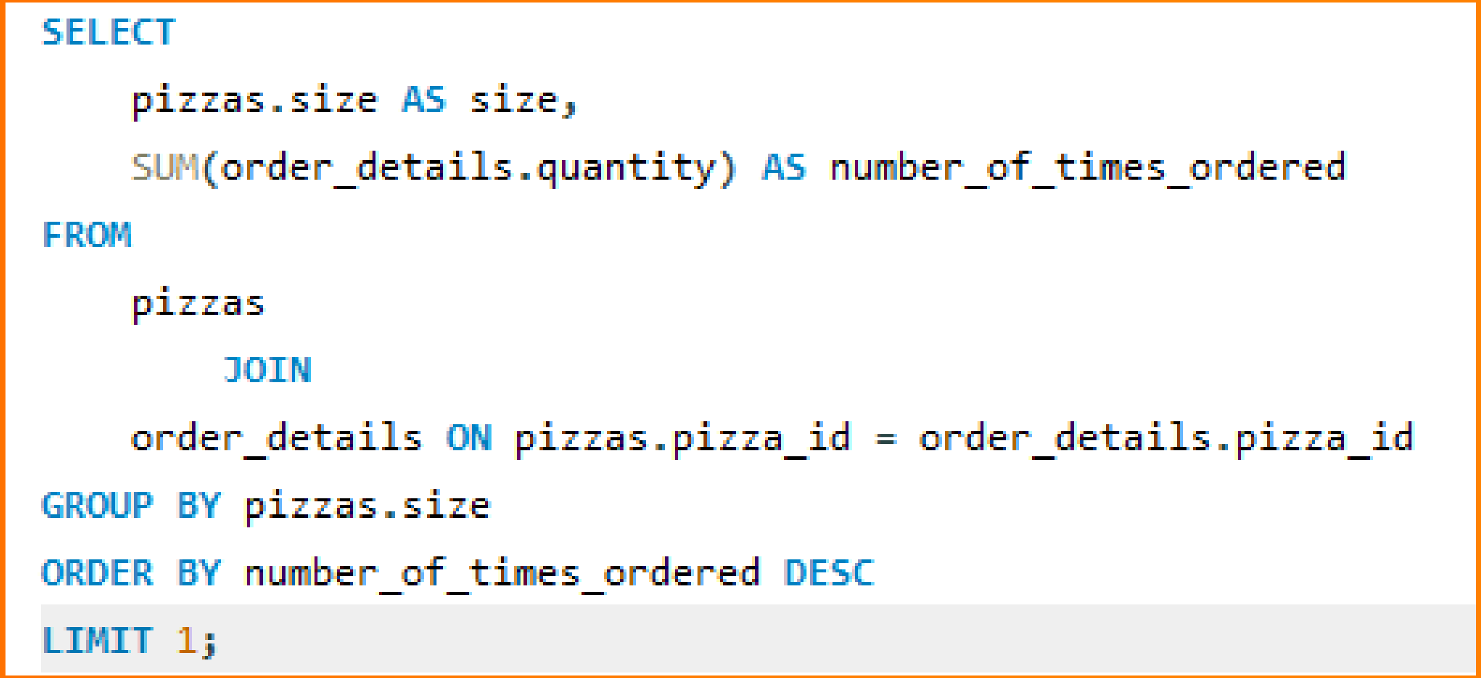


name_of_pizza	pizza_price
The Greek Pizza	35.95

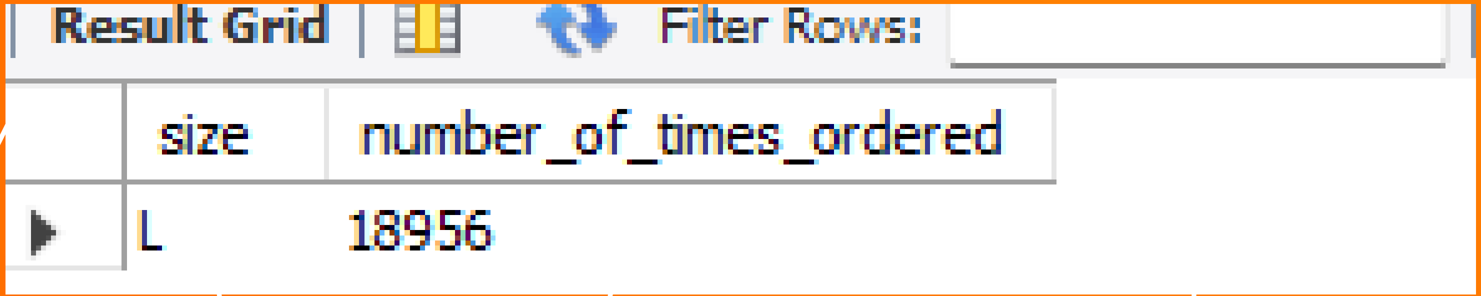
QUESTION

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

ANSWER



```
SELECT
    pizzas.size AS size,
    SUM(order_details.quantity) AS number_of_times_ordered
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY number_of_times_ordered DESC
LIMIT 1;
```

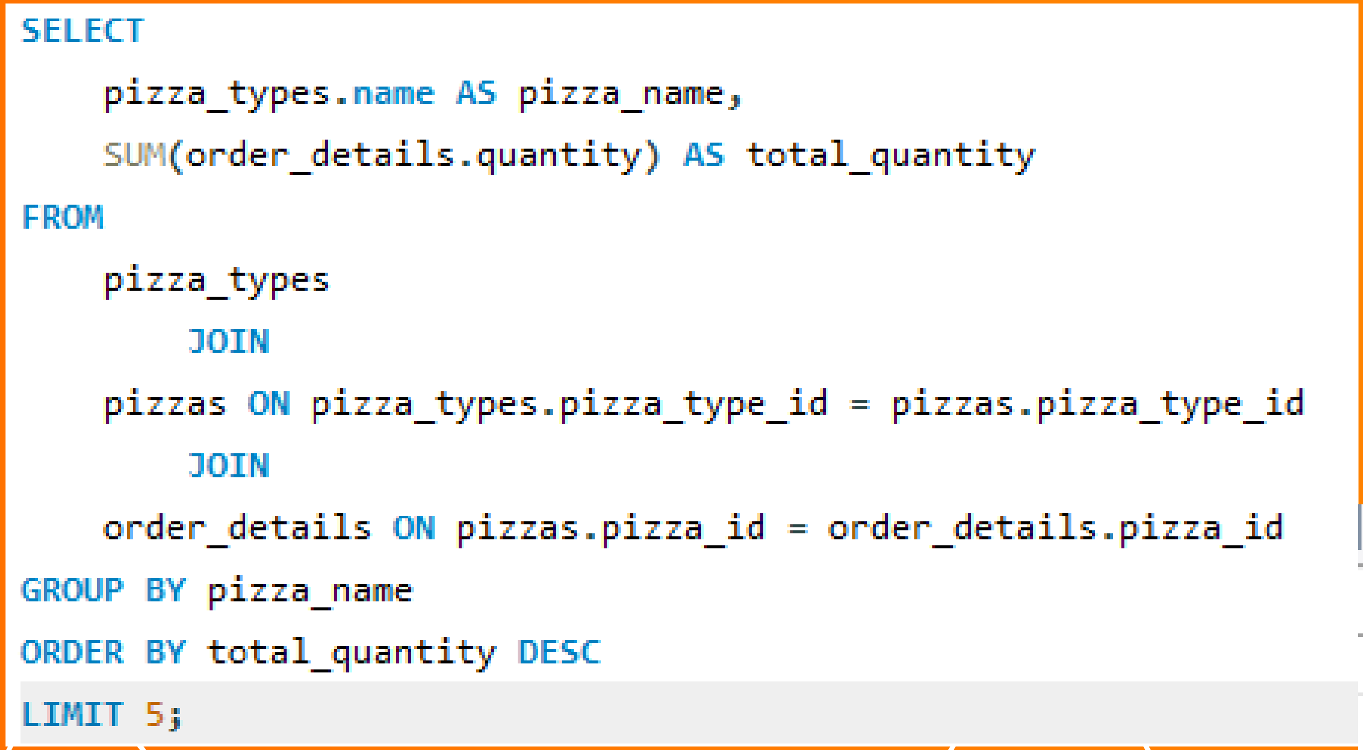


size	number_of_times_ordered
L	18956

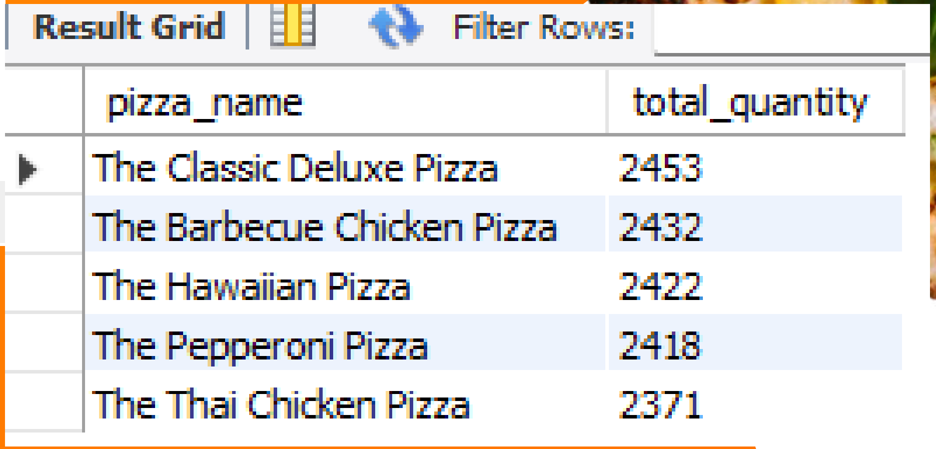
QUESTION

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

ANSWER



```
SELECT
    pizza_types.name AS pizza_name,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_name
ORDER BY total_quantity DESC
LIMIT 5;
```

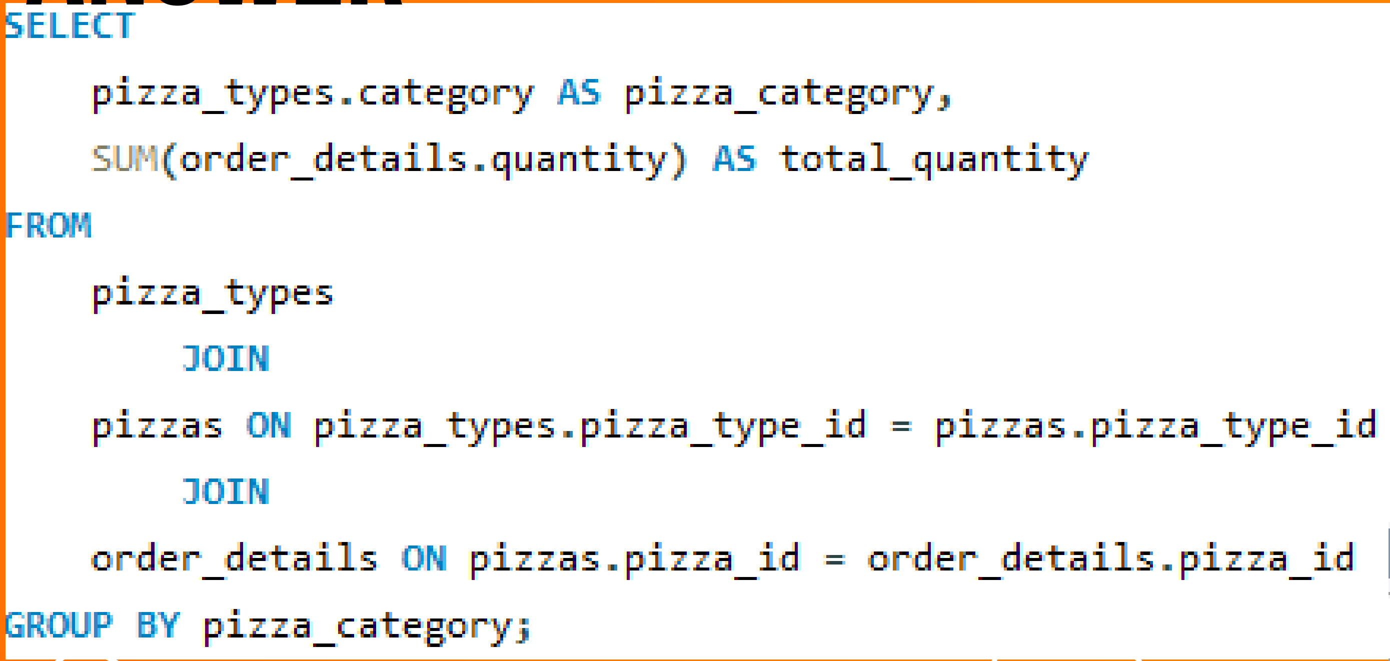


pizza_name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

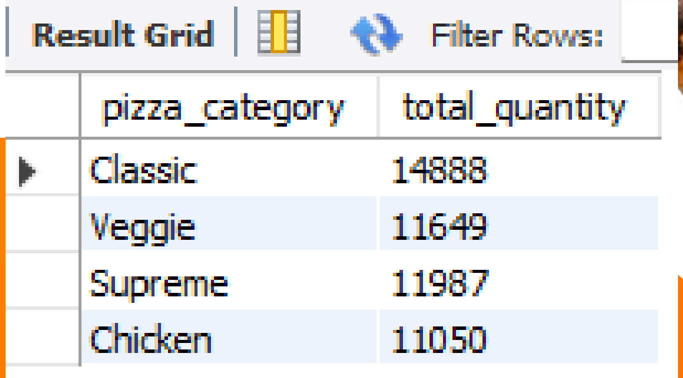
QUESTION

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

ANSWER



```
SELECT
    pizza_types.category AS pizza_category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_category;
```

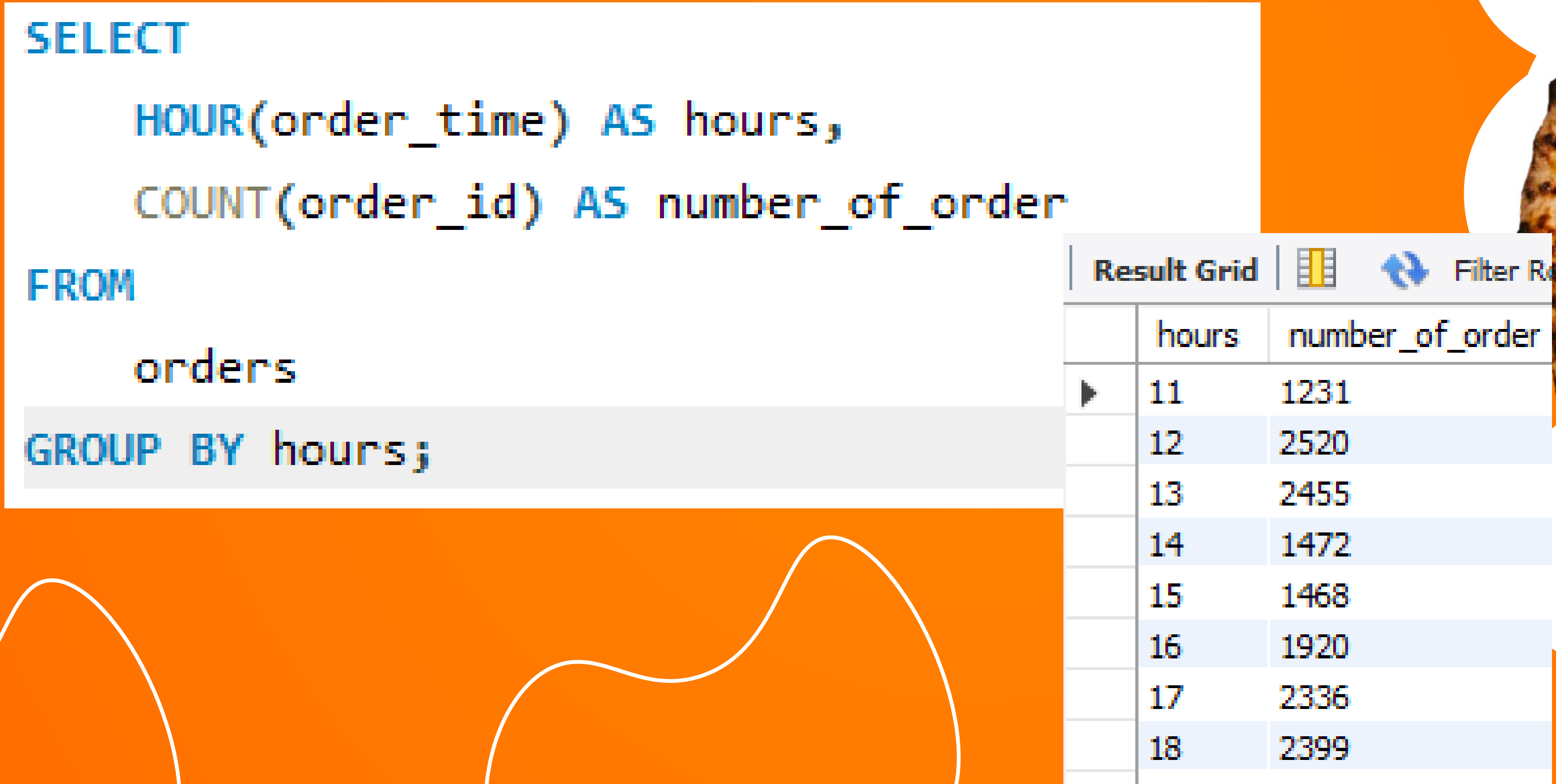


pizza_category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

QUESTION

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

ANSWER



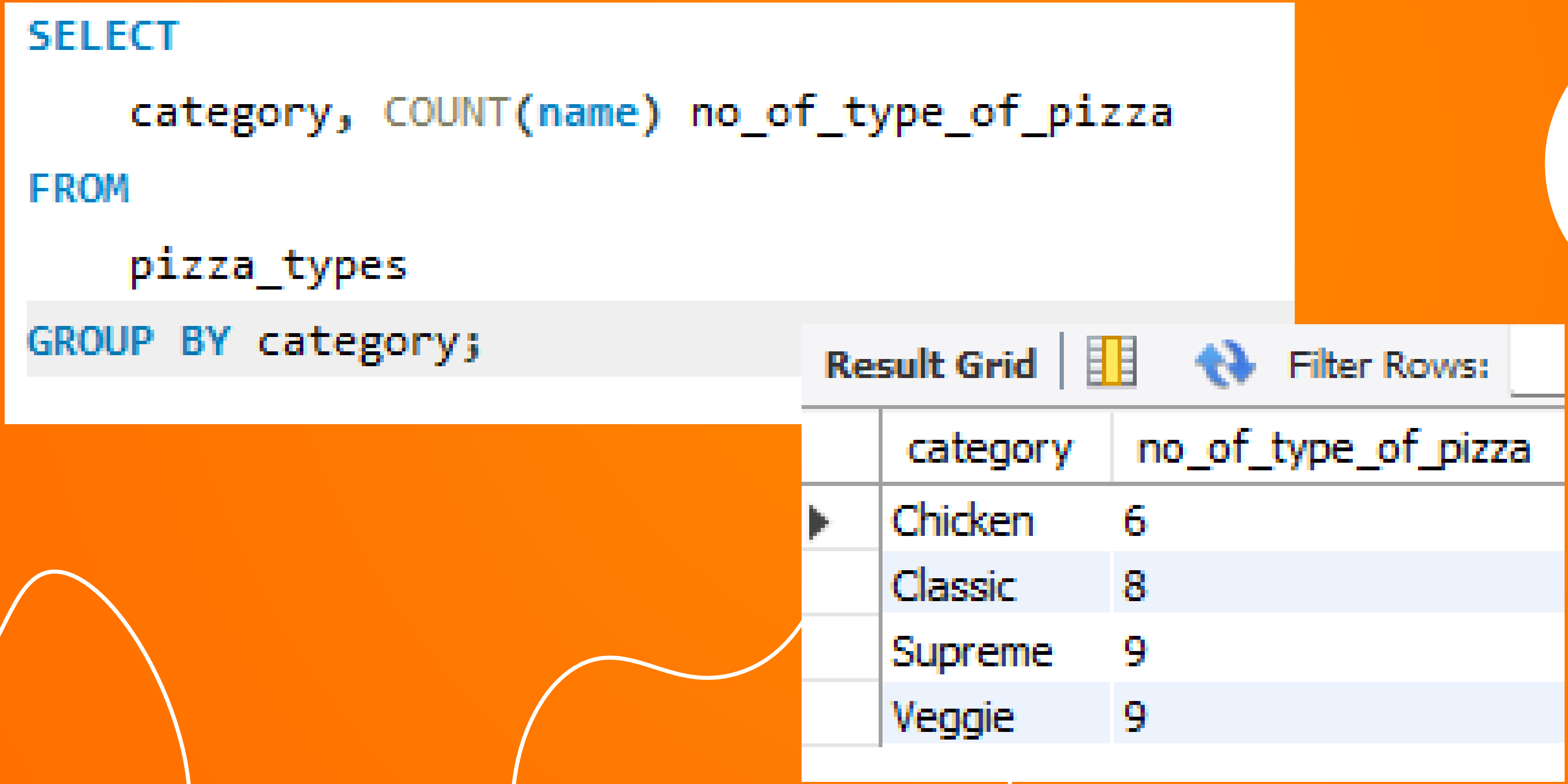
```sql
SELECT
    HOUR(order_time) AS hours,
    COUNT(order_id) AS number_of_order
FROM
    orders
GROUP BY hours;
```

hours	number_of_order
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

QUESTION

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

ANSWER



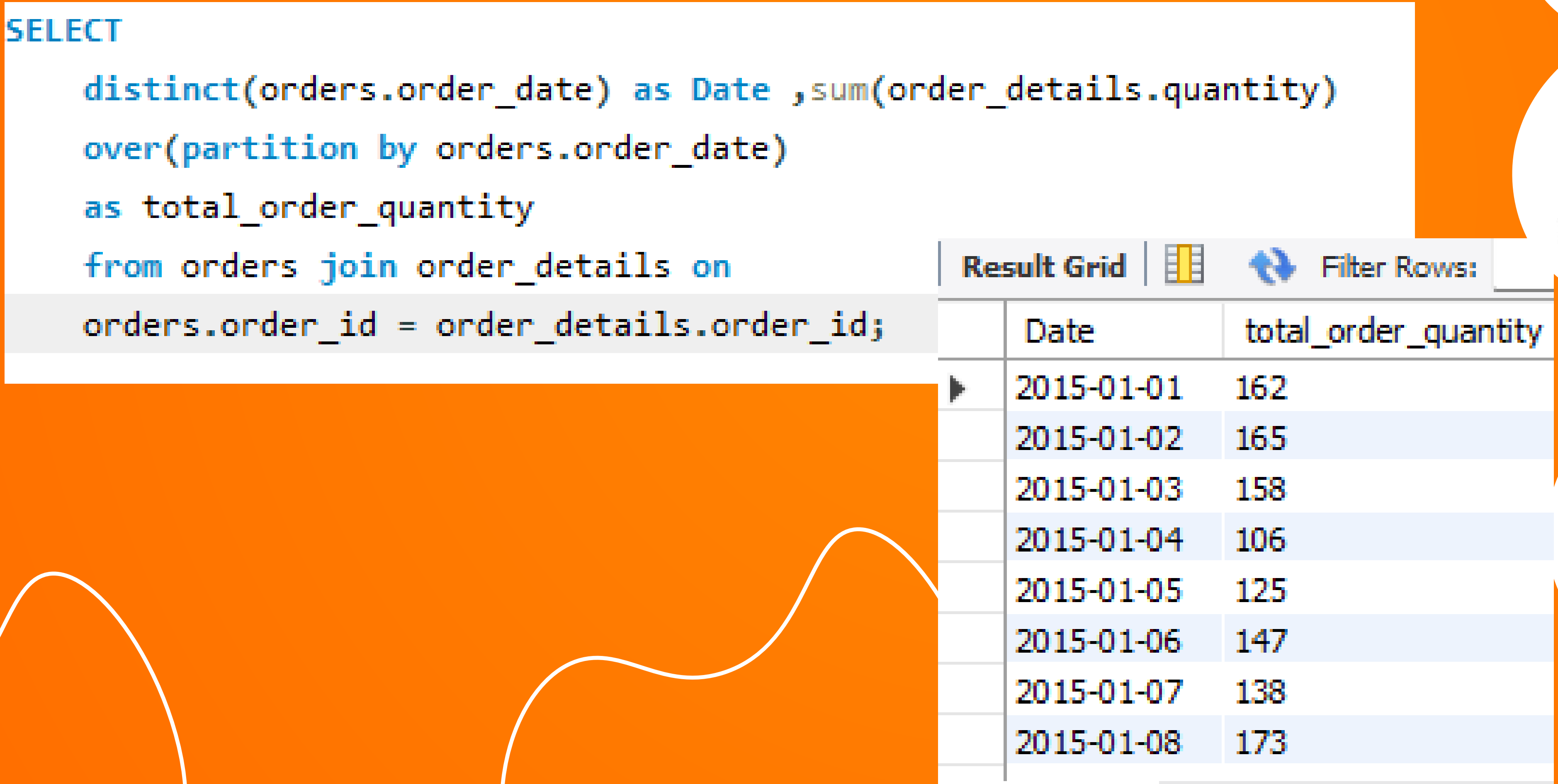
```
SELECT
    category, COUNT(name) no_of_type_of_pizza
FROM
    pizza_types
GROUP BY category;
```

category	no_of_type_of_pizza
Chicken	6
Classic	8
Supreme	9
Veggie	9

QUESTION

GROUP THE ORDERS BY DATE AND CALCULATE THE NUMBER OF PIZZAS ORDERED PER DAY

ANSWER



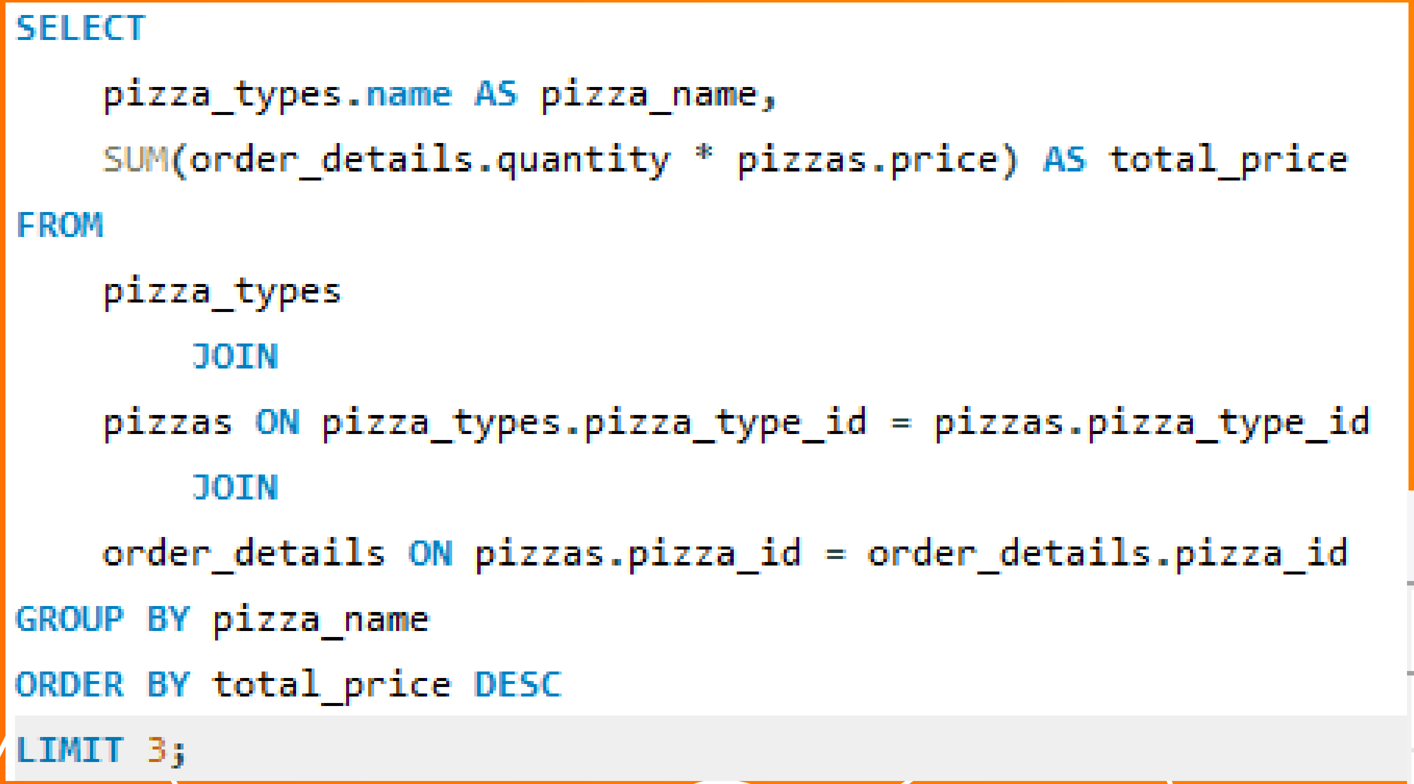
```
SELECT
    distinct(orders.order_date) as Date ,sum(order_details.quantity)
    over(partition by orders.order_date)
    as total_order_quantity
    from orders join order_details on
    orders.order_id = order_details.order_id;
```

Date	total_order_quantity
2015-01-01	162
2015-01-02	165
2015-01-03	158
2015-01-04	106
2015-01-05	125
2015-01-06	147
2015-01-07	138
2015-01-08	173

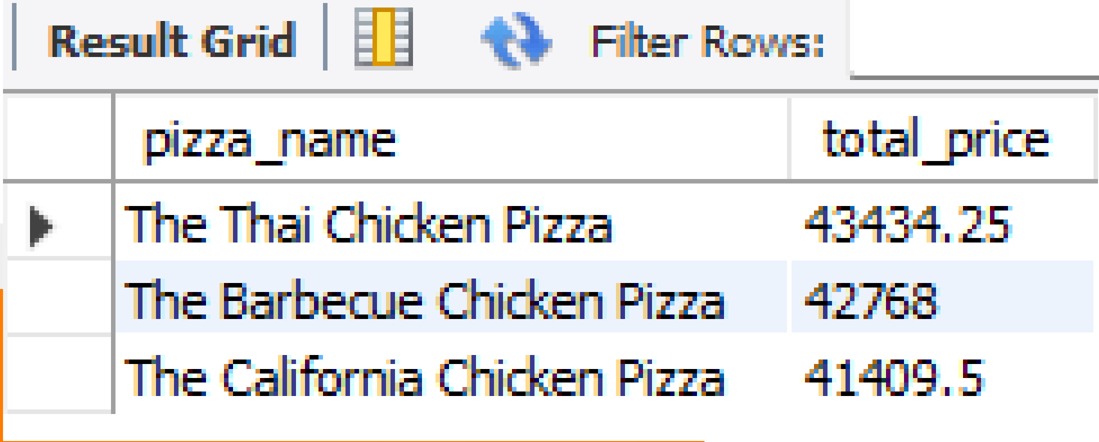
QUESTION

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

ANSWER



```
SELECT
    pizza_types.name AS pizza_name,
    SUM(order_details.quantity * pizzas.price) AS total_price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_name
ORDER BY total_price DESC
LIMIT 3;
```

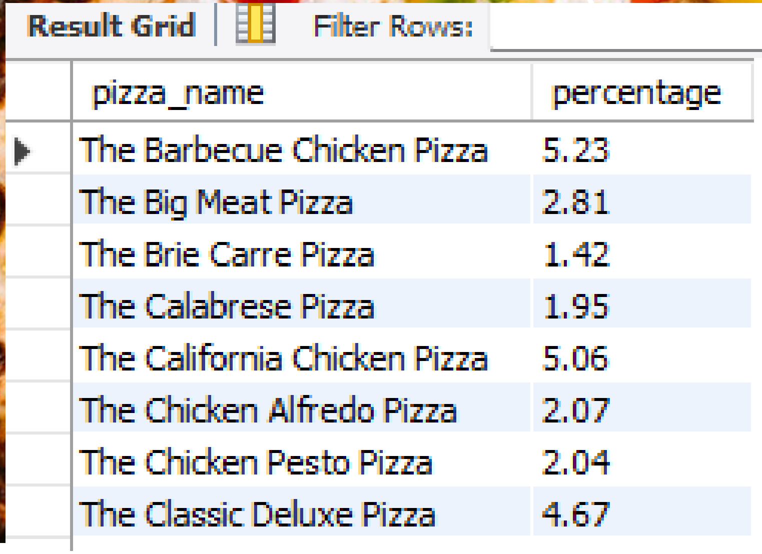


pizza_name	total_price
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

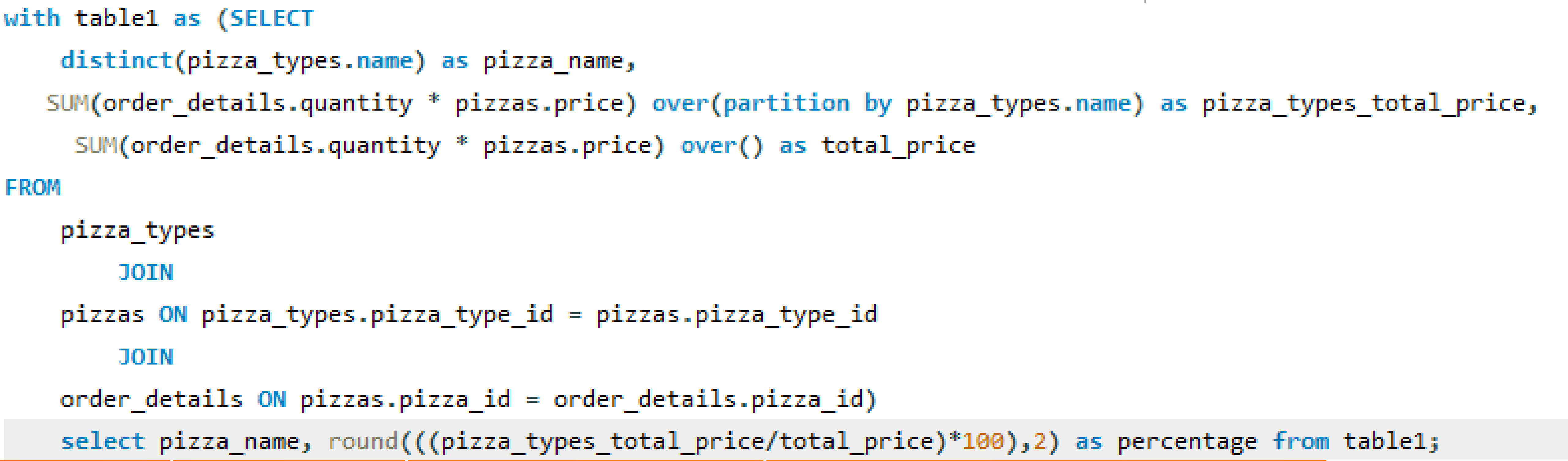
QUESTION

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

ANSWER



pizza_name	percentage
The Barbecue Chicken Pizza	5.23
The Big Meat Pizza	2.81
The Brie Carre Pizza	1.42
The Calabrese Pizza	1.95
The California Chicken Pizza	5.06
The Chicken Alfredo Pizza	2.07
The Chicken Pesto Pizza	2.04
The Classic Deluxe Pizza	4.67

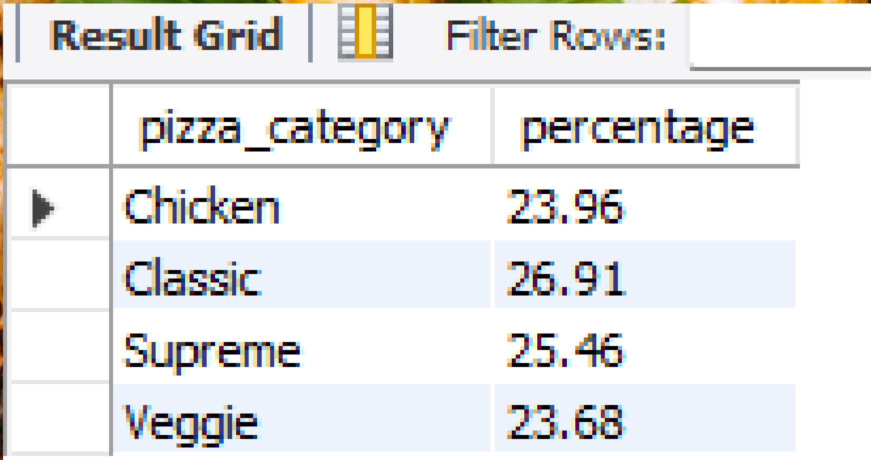


```sql
with table1 as (SELECT
    distinct(pizza_types.name) as pizza_name,
   SUM(order_details.quantity * pizzas.price) over(partition by pizza_types.name) as pizza_types_total_price,
     SUM(order_details.quantity * pizzas.price) over() as total_price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id)
    select pizza_name, round(((pizza_types_total_price/total_price)*100),2) as percentage from table1;
```

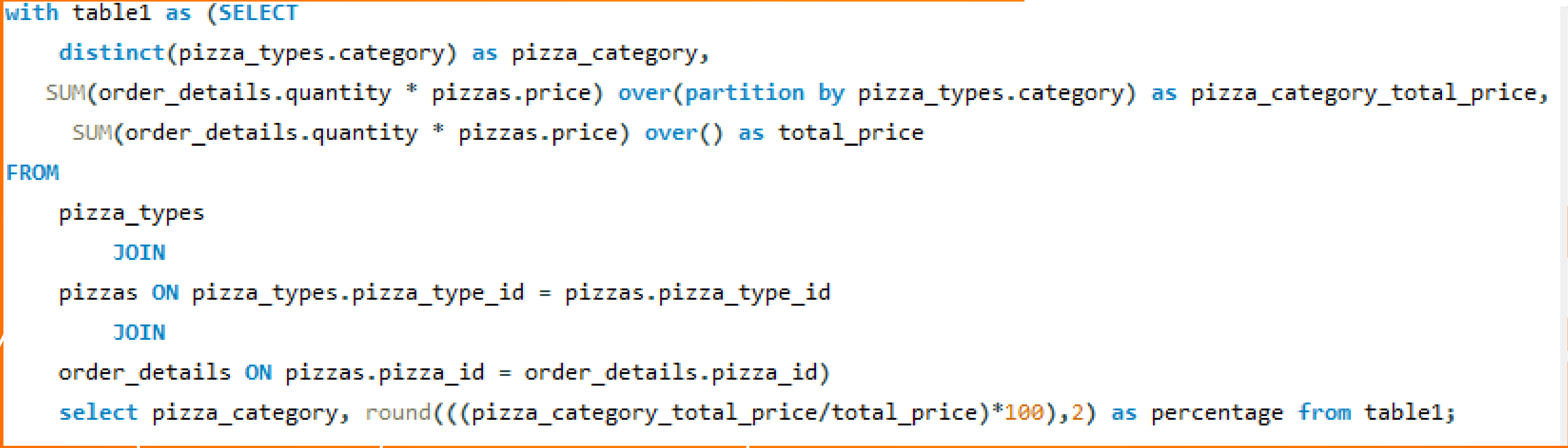
QUESTION

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA CATEGORY TO TOTAL REVENUE.

ANSWER



pizza_category	percentage
Chicken	23.96
Classic	26.91
Supreme	25.46
Veggie	23.68

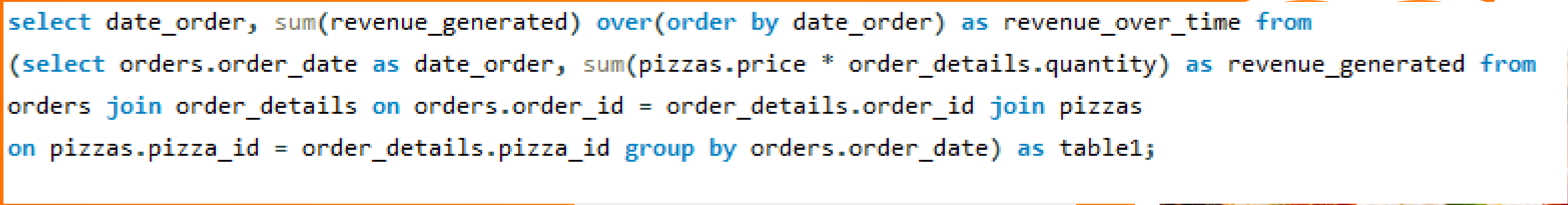


```
with table1 as (SELECT
    distinct(pizza_types.category) as pizza_category,
   SUM(order_details.quantity * pizzas.price) over(partition by pizza_types.category) as pizza_category_total_price,
     SUM(order_details.quantity * pizzas.price) over() as total_price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id)
    select pizza_category, round(((pizza_category_total_price/total_price)*100),2) as percentage from table1;
```

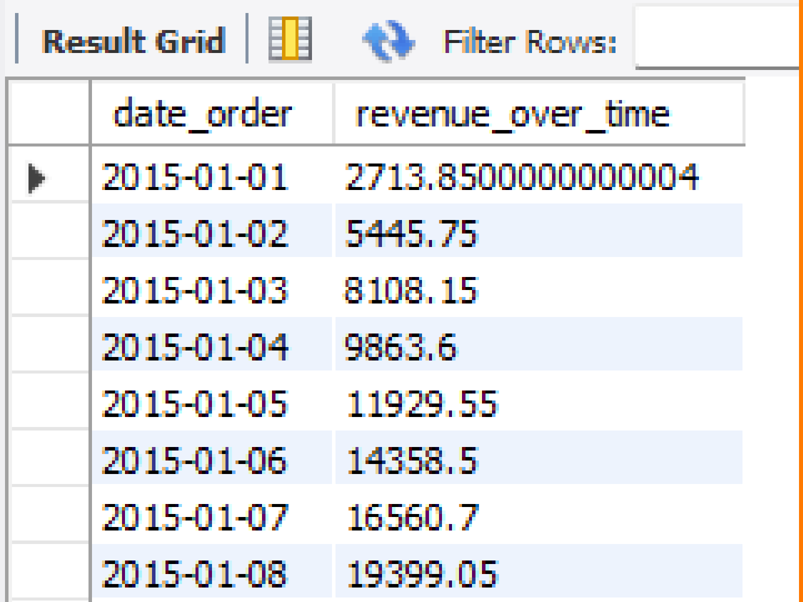
QUESTION

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

ANSWER



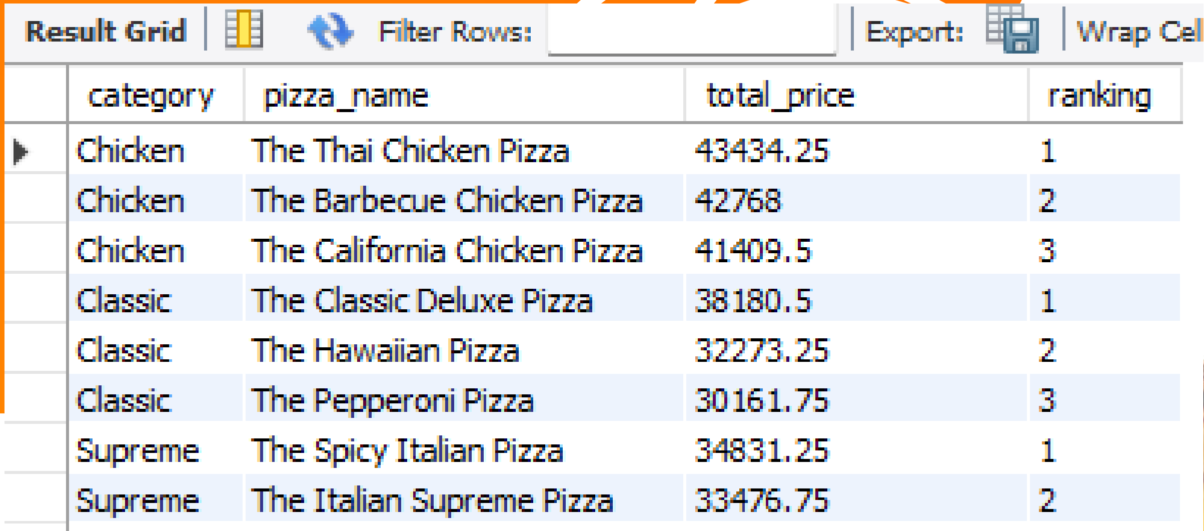
```
select date_order, sum(revenue_generated) over(order by date_order) as revenue_over_time from
(select orders.order_date as date_order, sum(pizzas.price * order_details.quantity) as revenue_generated from
orders join order_details on orders.order_id = order_details.order_id join pizzas
on pizzas.pizza_id = order_details.pizza_id group by orders.order_date) as table1;
```



date_order	revenue_over_time
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

QUESTION

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



category	pizza_name	total_price	ranking
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

ANSWER

```
select category, pizza_name,total_price, ranking from
(select category, pizza_name, total_price,
rank() over(partition by category order by total_price desc ) as ranking from
(SELECT
    pizza_types.category as category , pizza_types.name AS pizza_name,
    SUM(order_details.quantity * pizzas.price) AS total_price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
    group by category , pizza_name) as table1) as table2 where ranking <=3;
```

THANK YOU