LARANA PIZZA

MY SQL

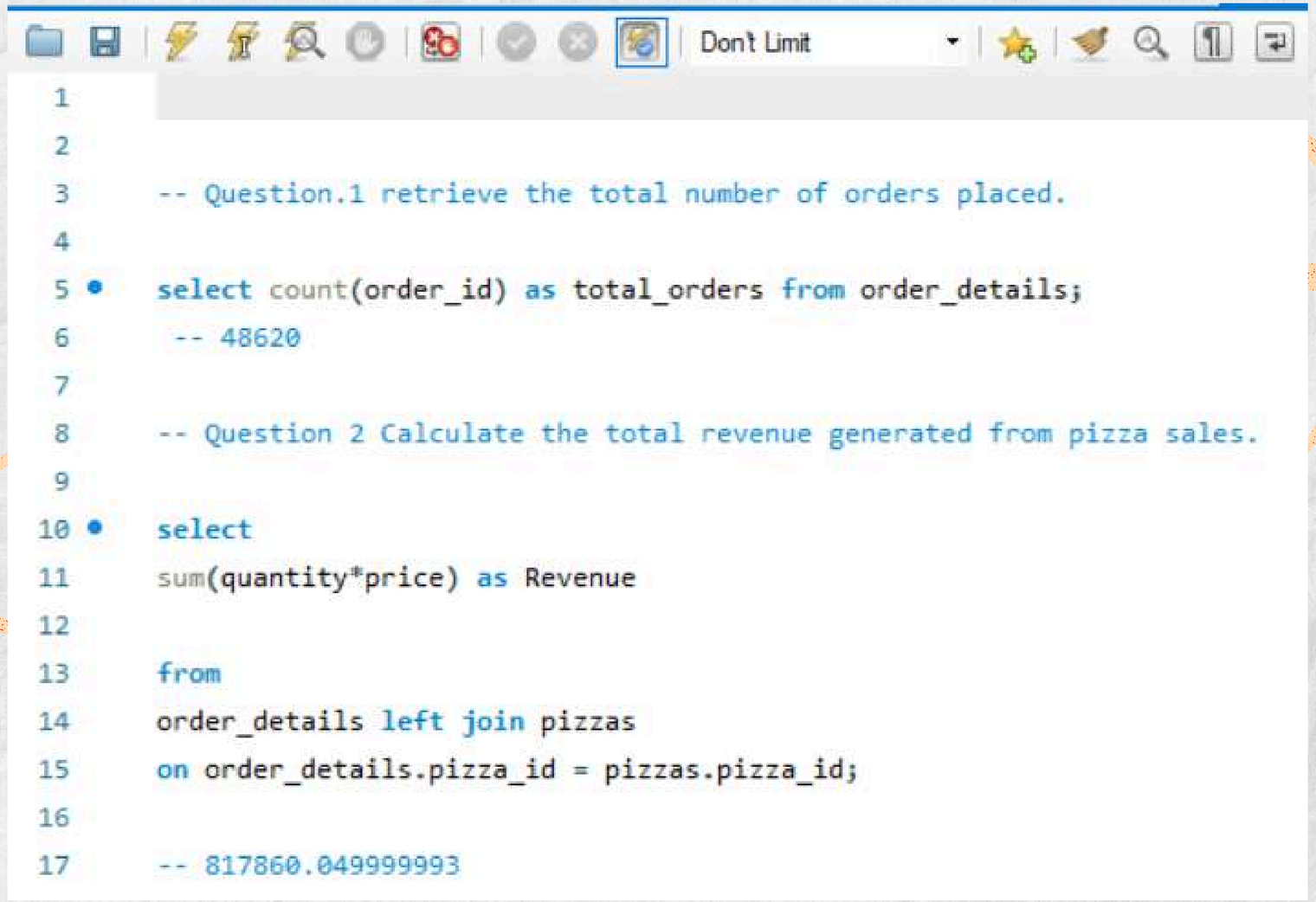
PIZZA PROJECT

NEXT >

AIM

- Analyze Sales Data: Gain insights into pizza orders, including total sales, revenue, popular items, and distribution patterns.
- Optimize Business Strategies: Provide actionable data on customer preferences, revenue streams, and sales trends to support decision-making.
- Enhance Operational Efficiency: Highlight trends like peak order hours or popular categories for resource allocation and inventory planning.
- Monitor Revenue Growth: Track cumulative and percentage-based contributions to evaluate the success of product offerings.

QUERIES



< BACK

NEXT >

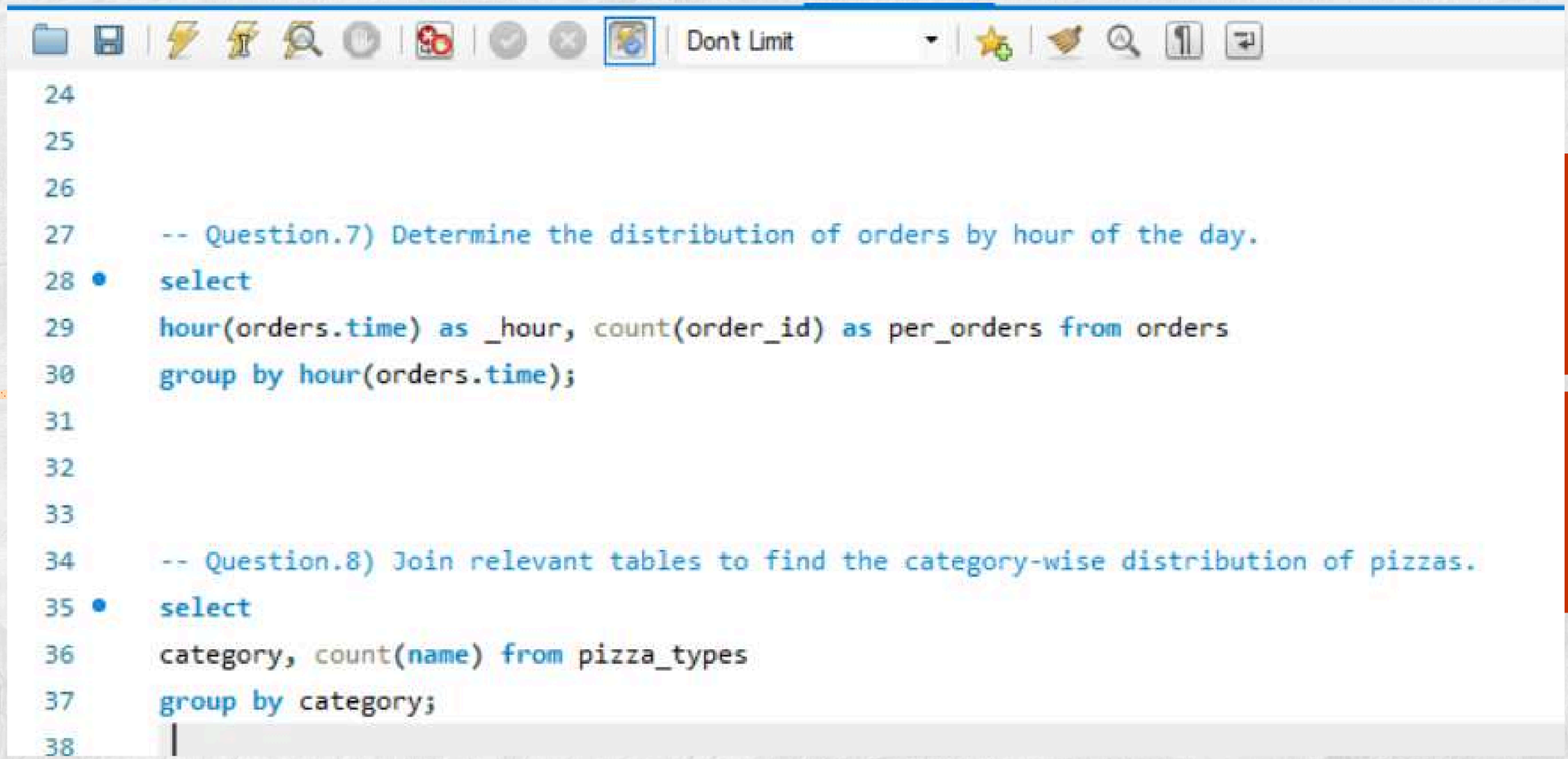
```sql
-- Question 3 Identify the highest-priced pizza.

select price, pizza_id from pizzas
order by price desc;

-- 35.95 usd is highest price pizza(the_greek_xxl)


 -- Question 4 Identify the most common pizza size ordered.
 use pizza_shop;
 select
 pizzas.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 pizzas.size order by order_count desc
 limit 5;
-- L    18526
-- M    15385
-- S    14137
-- XL   544
-- XXL  28
```

Don't Limit

```
40
41
42      -- Question 5 List the top 5 most ordered pizza types along with their quantities.
43
44  •   select pizza_types.name,
45      sum(order_details.quantity) as total
46      from pizza_types join  pizzas
47      on pizza_types.pizza_type_id = pizzas.pizza_type_id
48      join order_details
49      on order_details.pizza_id = pizzas.pizza_id
50      group by pizza_types.name order by total desc limit 5;
51      -- ANSWER 5
52      -- 21350    1   bbq_ckn
53      -- 21349    1   mexicana
54      -- 21348    1   napolitana
55      -- 21348    1   four_cheese
56      -- 21348    1   ckn_alfredo
57
58
59
60
61
```

Don't Limit

```sql
-- Question 6 -- Join the necessary tables to find the total quantity of each pizza category ordered.?

select * from pizza_types;
select * from order_details;

select
sum(quantity) as total_quantity,
category
from
pizzas join pizza_types
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 total_quantity desc;
```

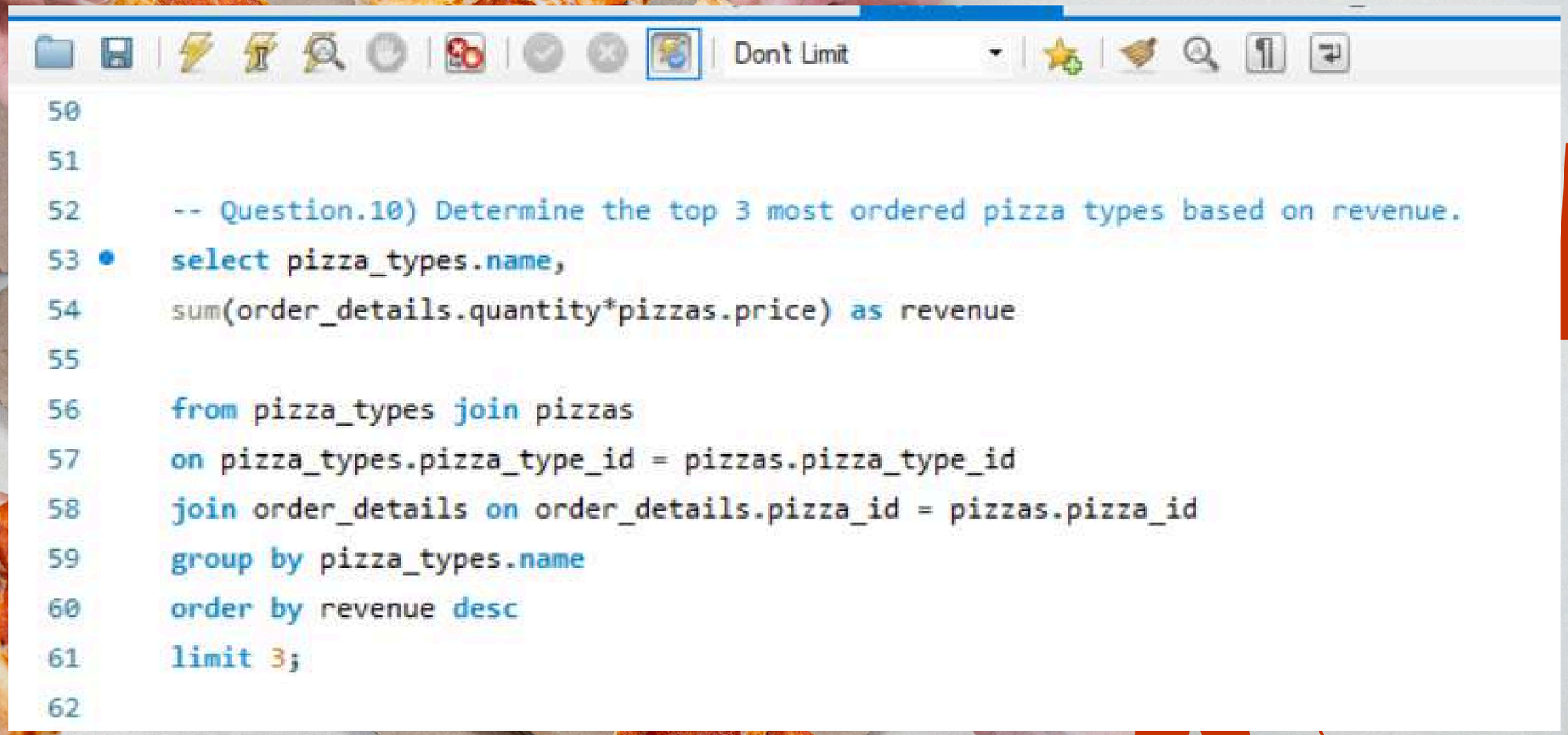

Don't Limit
24
25
26
27 -- Question.7) Determine the distribution of orders by hour of the day.
28 select
29 hour(orders.time) as _hour, count(order_id) as per_orders from orders
30 group by hour(orders.time);
31
32
33
34 -- Question.8) Join relevant tables to find the category-wise distribution of pizzas.
35 select
36 category, count(name) from pizza_types
37 group by category;
38

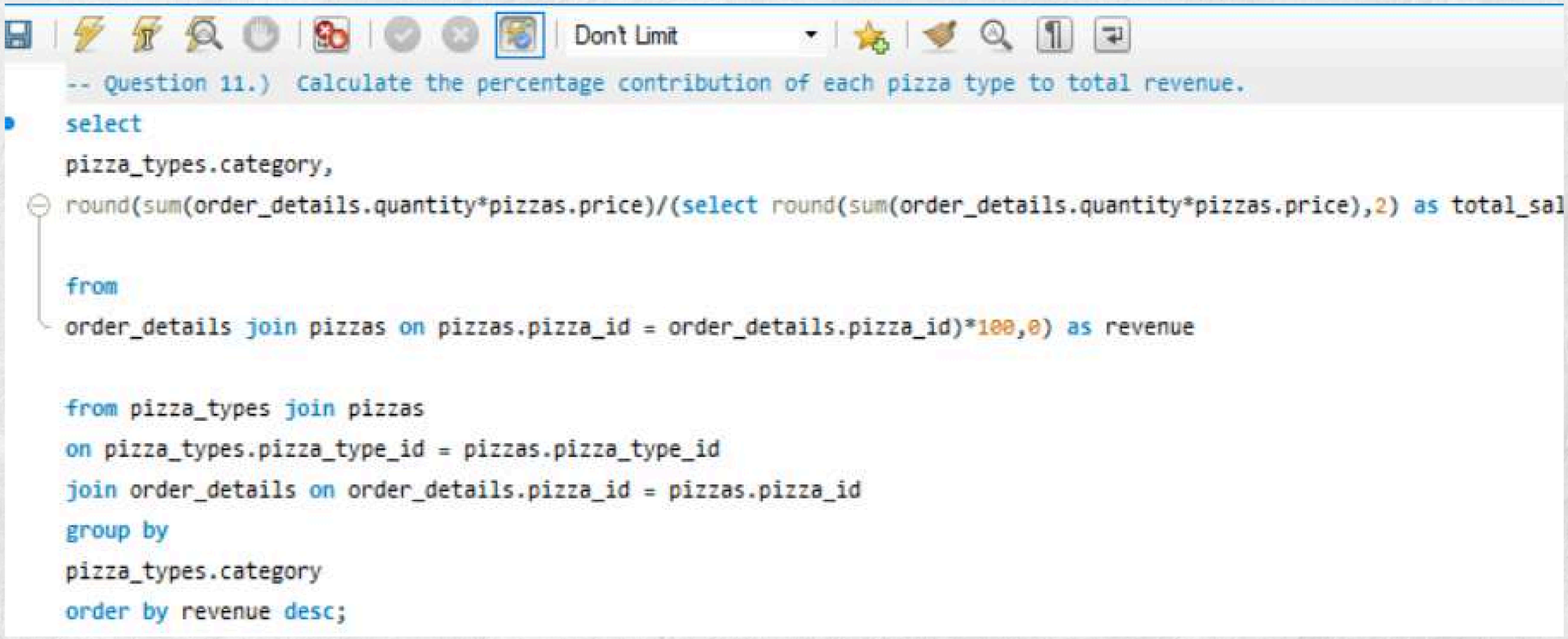
NG
NG"

FREE DELIVERY

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

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

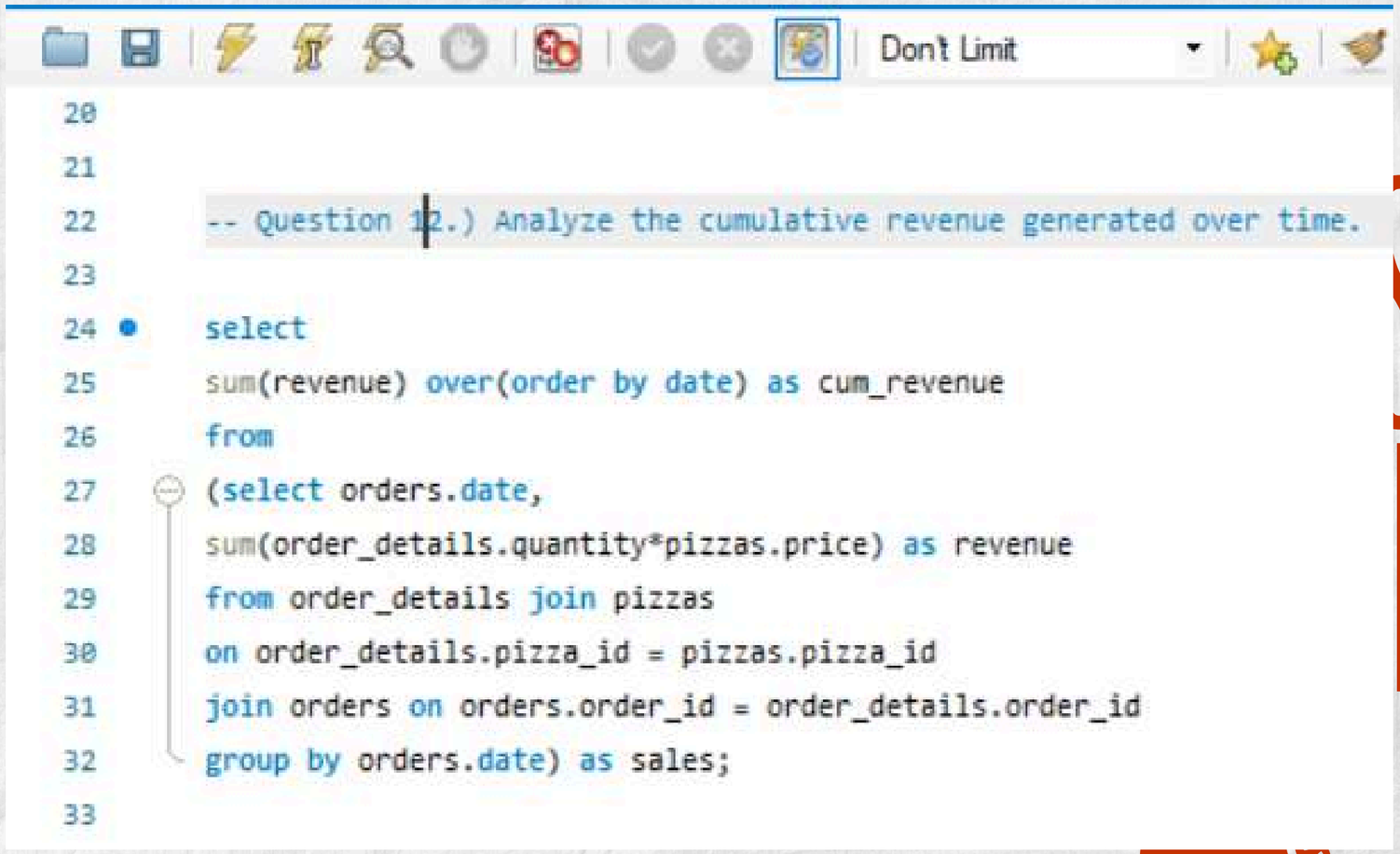
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.


Don't Limit
50
51
52 -- Question.10) Determine the top 3 most ordered pizza types based on revenue.
53 select pizza_types.name,
54 sum(order_details.quantity*pizzas.price) as revenue
55
56 from pizza_types join pizzas
57 on pizza_types.pizza_type_id = pizzas.pizza_type_id
58 join order_details on order_details.pizza_id = pizzas.pizza_id
59 group by pizza_types.name
60 order by revenue desc
61 limit 3;
62



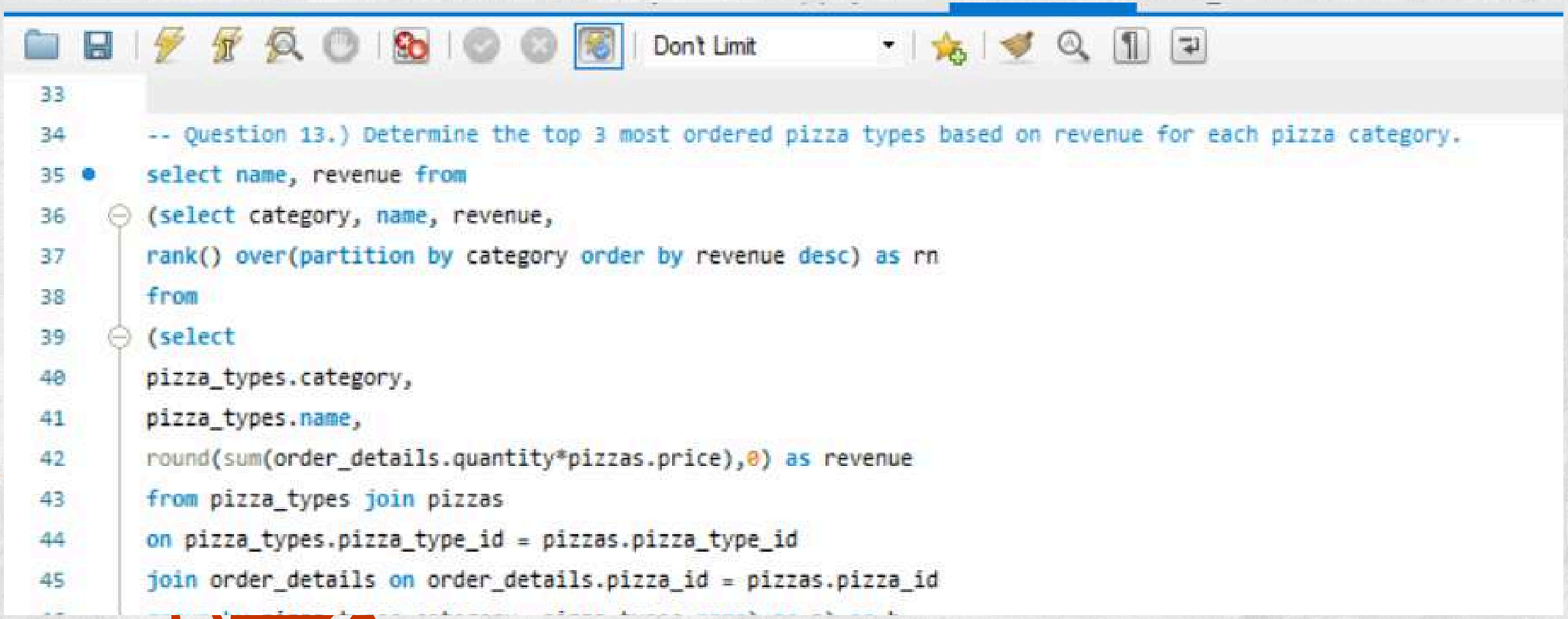
Don't Limit

```
-- Question 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_sal

from
order_details join pizzas on pizzas.pizza_id = order_details.pizza_id)*100,0) 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;
```


Don't Limit
20
21
22 -- Question 12.) Analyze the cumulative revenue generated over time.
23
24 select
25 sum(revenue) over(order by date) as cum_revenue
26 from
27 (select orders.date,
28 sum(order_details.quantity*pizzas.price) as revenue
29 from order_details join pizzas
30 on order_details.pizza_id = pizzas.pizza_id
31 join orders on orders.order_id = order_details.order_id
32 group by orders.date) as sales;
33

S PIZZA
IME!

FREE DELIVERY



THANKYOU

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.