SQL For Pizza Sales Analysis Full Project

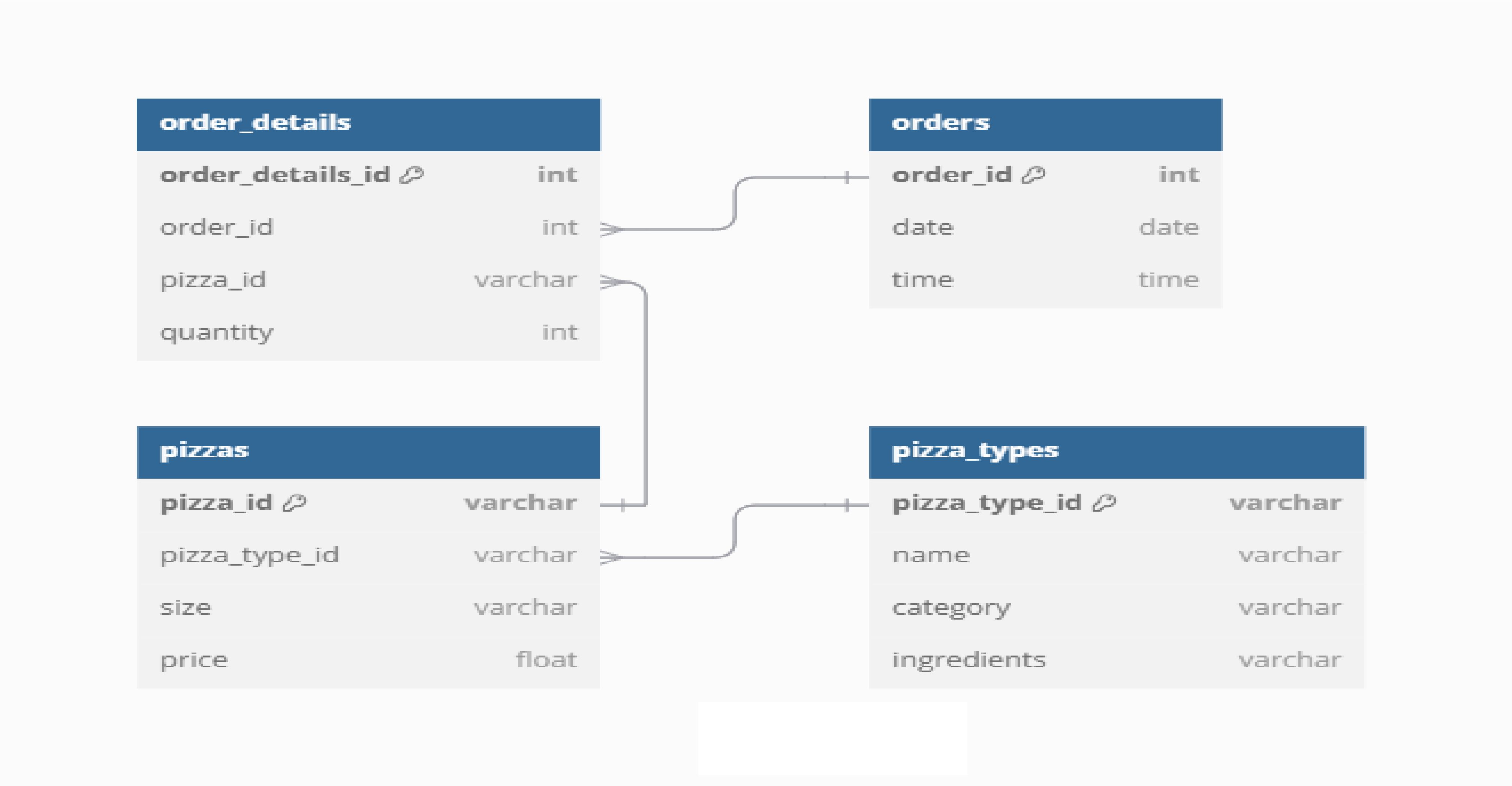
Presented by:

Shakil Ahammed (Joy)

ERP Implementation Specialist

/saj0cse /saj0cse /saj0cse shakil.cse40@gmail.com

Database Schema



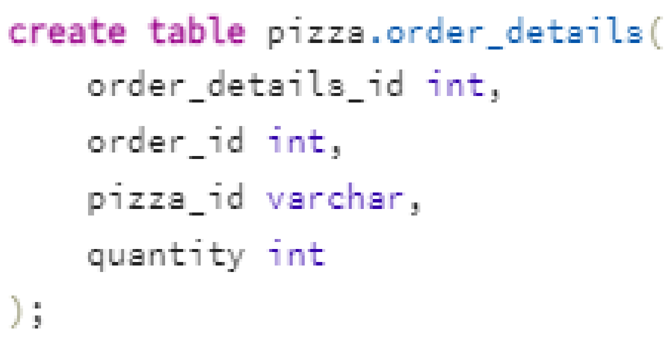
order_details	
order_details_id 🔑	**int**
order_id	int
pizza_id	varchar
quantity	int

orders	
order_id 🔑	**int**
date	date
time	time

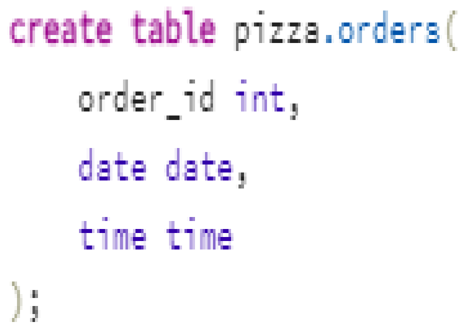
pizzas	
pizza_id 🔑	**varchar**
pizza_type_id	varchar
size	varchar
price	float

pizza_types	
pizza_type_id 🔑	**varchar**
name	varchar
category	varchar
ingredients	varchar

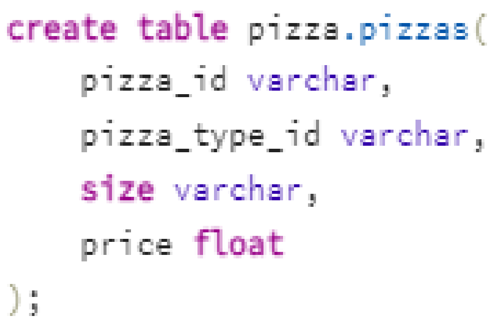
Create Table



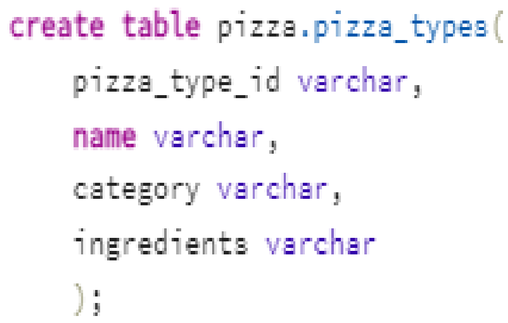
```
create table pizza.order_details(
    order_details_id int,
    order_id int,
    pizza_id varchar,
    quantity int
);
```



```
create table pizza.orders(
    order_id int,
    date date,
    time time
);
```

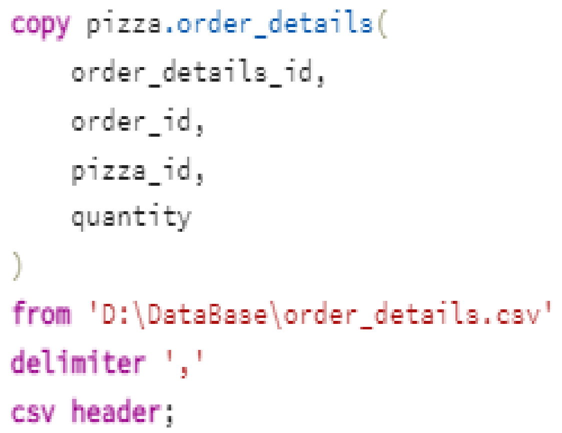


```
create table pizza.pizzas(
    pizza_id varchar,
    pizza_type_id varchar,
    size varchar,
    price float
);
```

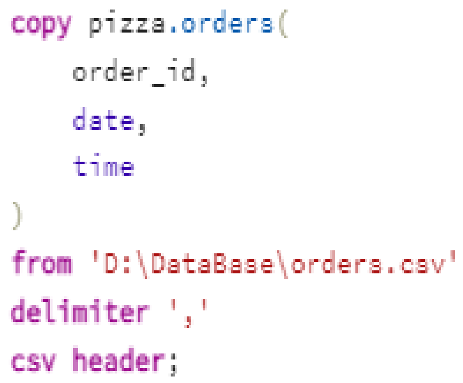


```
create table pizza.pizza_types(
    pizza_type_id varchar,
    name varchar,
    category varchar,
    ingredients varchar
);
```

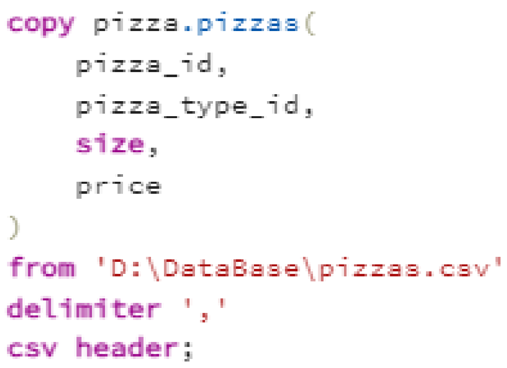
Data Insert



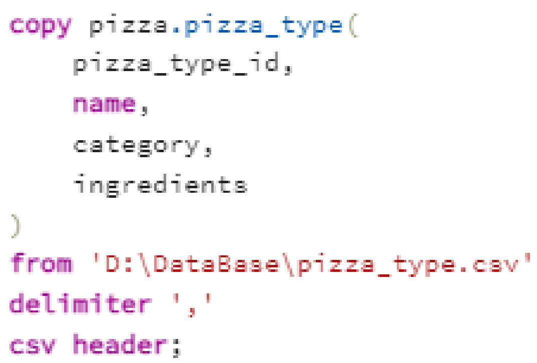
```
copy pizza.order_details(
    order_details_id,
    order_id,
    pizza_id,
    quantity
)
from 'D:\DataBase\order_details.csv'
delimiter ','
csv header;
```



```
copy pizza.orders(
    order_id,
    date,
    time
)
from 'D:\DataBase\orders.csv'
delimiter ','
csv header;
```

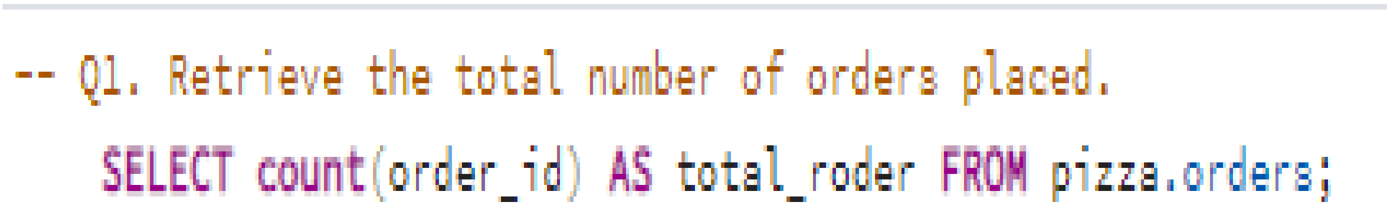


```
copy pizza.pizzas(
    pizza_id,
    pizza_type_id,
    size,
    price
)
from 'D:\DataBase\pizzas.csv'
delimiter ','
csv header;
```

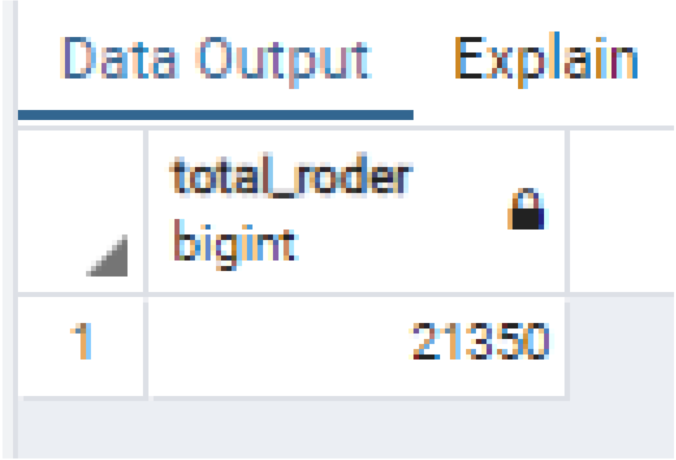


```
copy pizza.pizza_type(
    pizza_type_id,
    name,
    category,
    ingredients
)
from 'D:\DataBase\pizza_type.csv'
delimiter ','
csv header;
```

Question No: 01

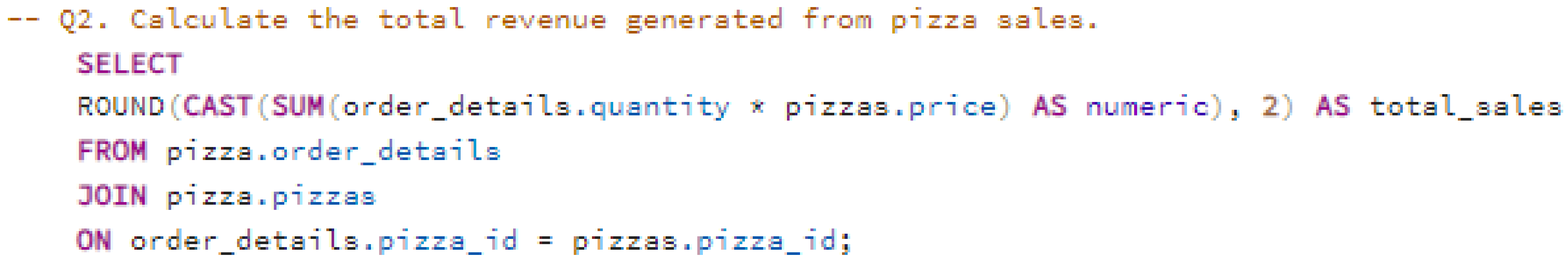


```sql
-- Q1. Retrieve the total number of orders placed.
    SELECT count(order_id) AS total_roder FROM pizza.orders;
```

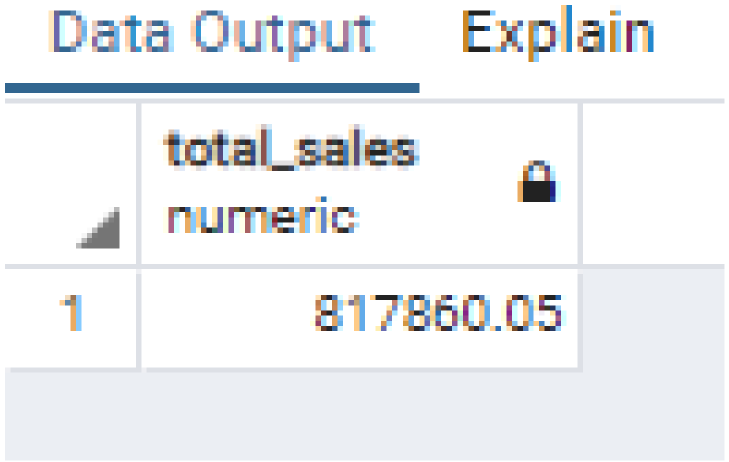


```
     total_roder
     bigint
1          21350
```

Question No: 02

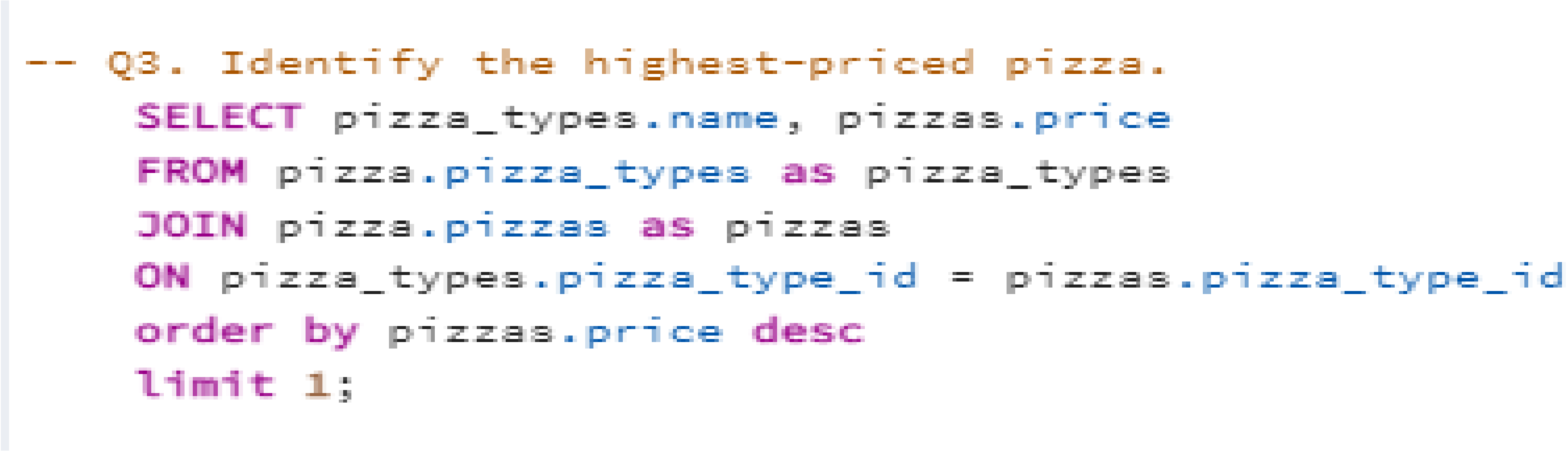


```
-- Q2. Calculate the total revenue generated from pizza sales.
    SELECT
    ROUND(CAST(SUM(order_details.quantity * pizzas.price) AS numeric), 2) AS total_sales
    FROM pizza.order_details
    JOIN pizza.pizzas
    ON order_details.pizza_id = pizzas.pizza_id;
```

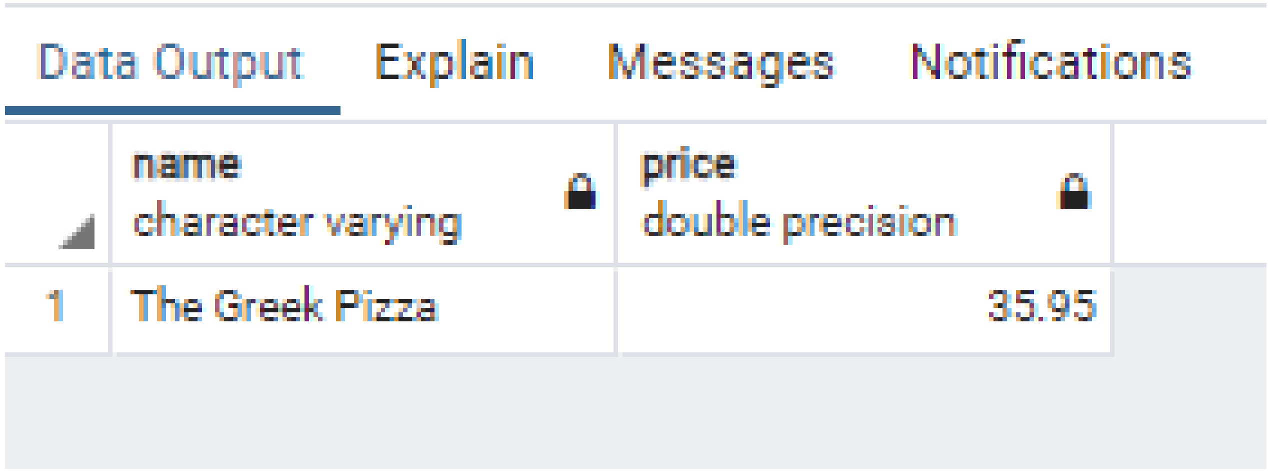


```
   total_sales
   numeric
1    817860.05
```

Question No: 03

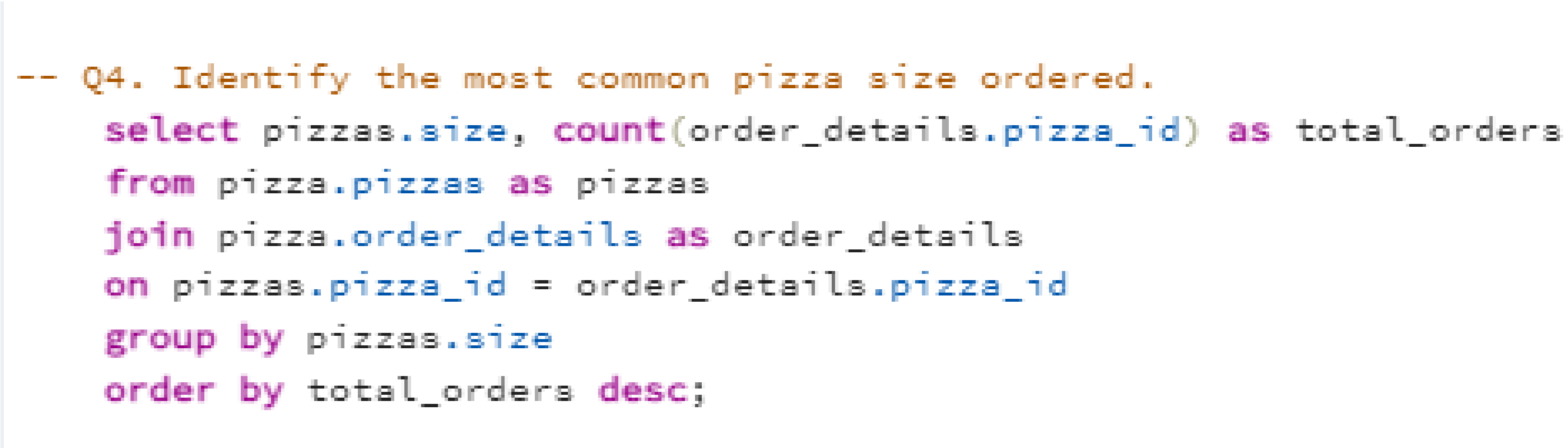


```
-- Q3. Identify the highest-priced pizza.
     SELECT pizza_types.name, pizzas.price
     FROM pizza.pizza_types as pizza_types
     JOIN pizza.pizzas as pizzas
     ON pizza_types.pizza_type_id = pizzas.pizza_type_id
     order by pizzas.price desc
     limit 1;
```

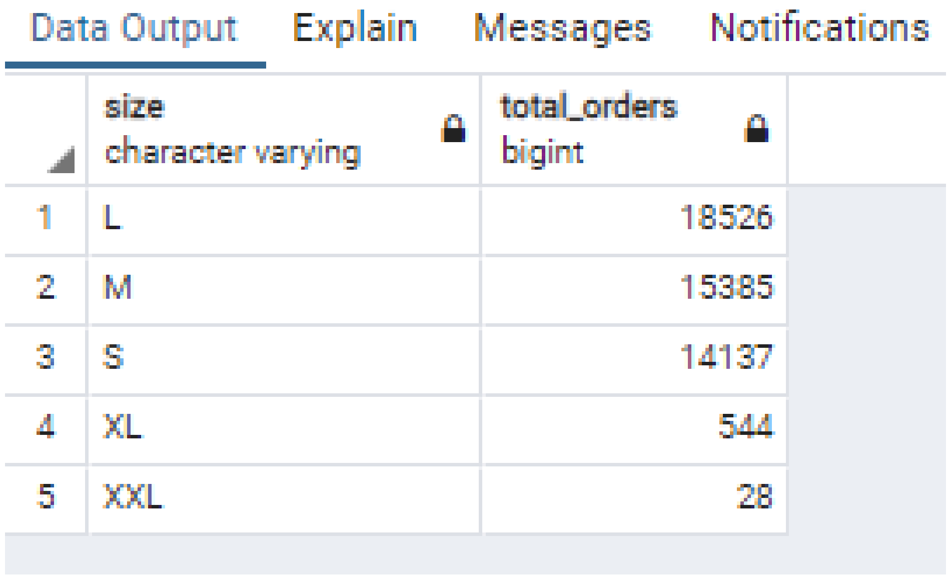


	name character varying	price double precision
1	The Greek Pizza	35.95

Question No: 04

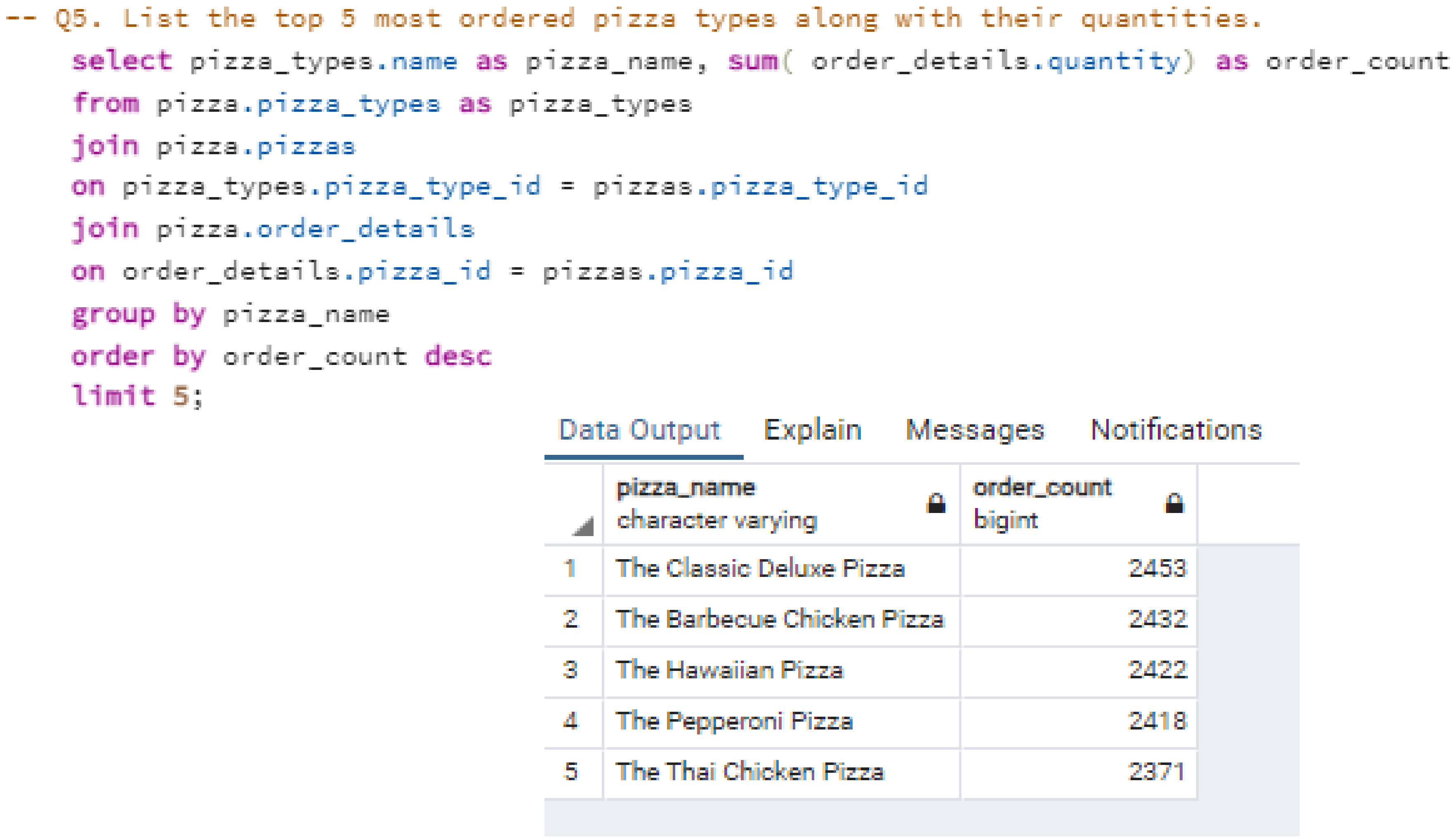


```
-- Q4. Identify the most common pizza size ordered.
    select pizzas.size, count(order_details.pizza_id) as total_orders
    from pizza.pizzas as pizzas
    join pizza.order_details as order_details
    on pizzas.pizza_id = order_details.pizza_id
    group by pizzas.size
    order by total_orders desc;
```



	size character varying	total_orders bigint
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

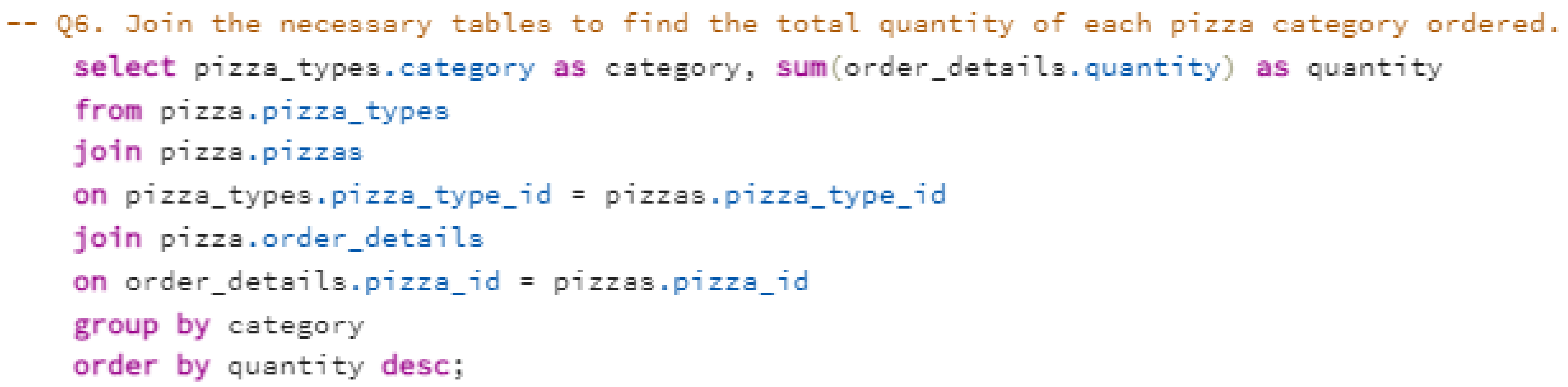
Question No: 05



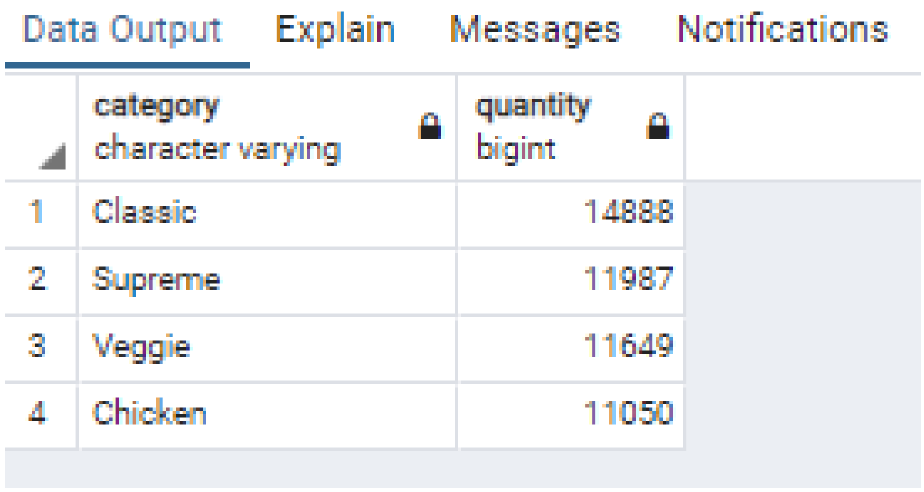
```
-- Q5. List the top 5 most ordered pizza types along with their quantities.
    select pizza_types.name as pizza_name, sum( order_details.quantity) as order_count
    from pizza.pizza_types as pizza_types
    join pizza.pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
    join pizza.order_details
    on order_details.pizza_id = pizzas.pizza_id
    group by pizza_name
    order by order_count desc
    limit 5;
```

	pizza_name character varying	order_count bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

Question No: 06

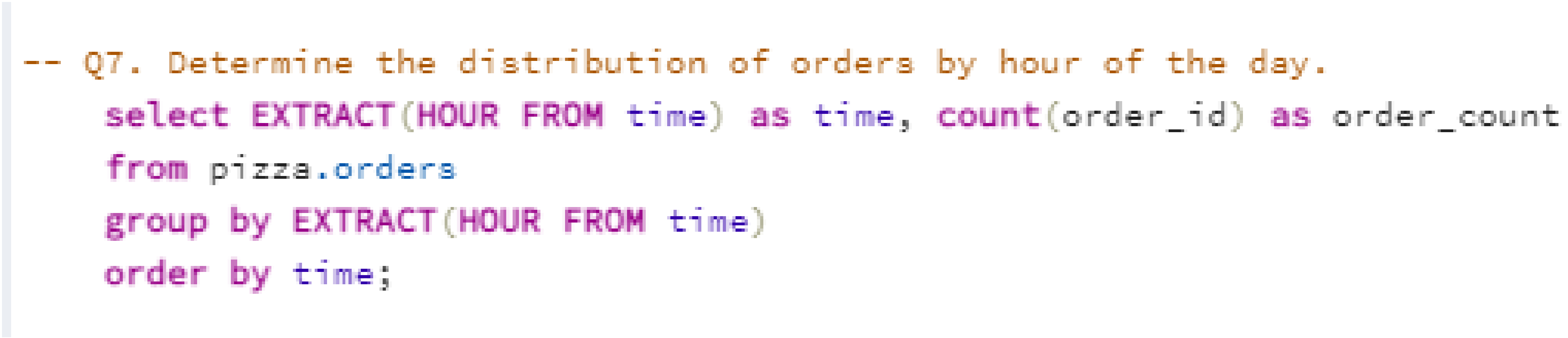


```sql
-- Q6. Join the necessary tables to find the total quantity of each pizza category ordered.
    select pizza_types.category as category, sum(order_details.quantity) as quantity
    from pizza.pizza_types
    join pizza.pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
    join pizza.order_details
    on order_details.pizza_id = pizzas.pizza_id
    group by category
    order by quantity desc;
```

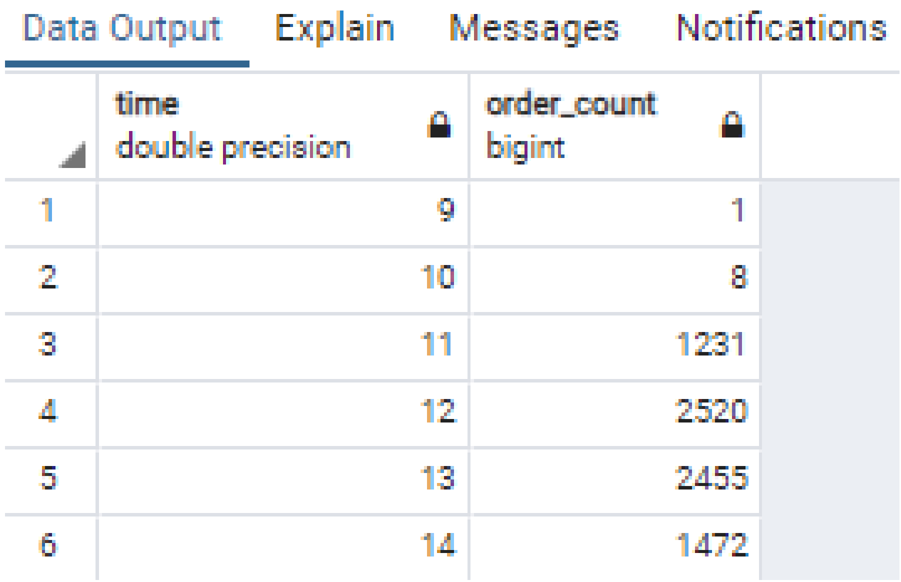


	category character varying	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

Question No: 07

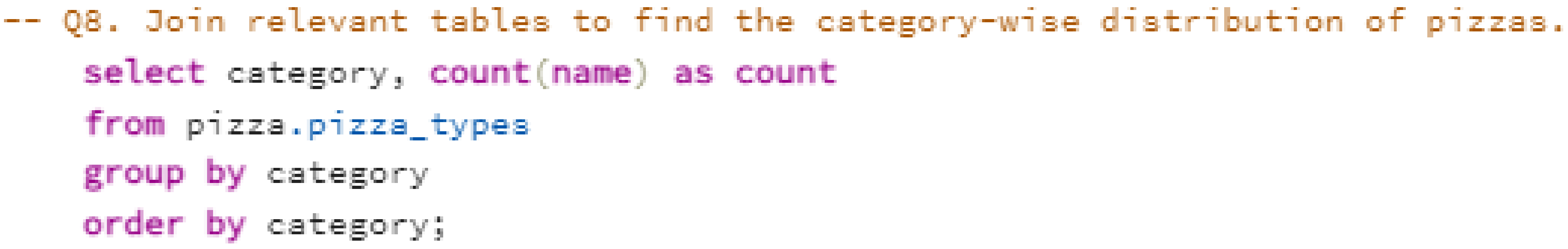


```
-- Q7. Determine the distribution of orders by hour of the day.
    select EXTRACT(HOUR FROM time) as time, count(order_id) as order_count
    from pizza.orders
    group by EXTRACT(HOUR FROM time)
    order by time;
```

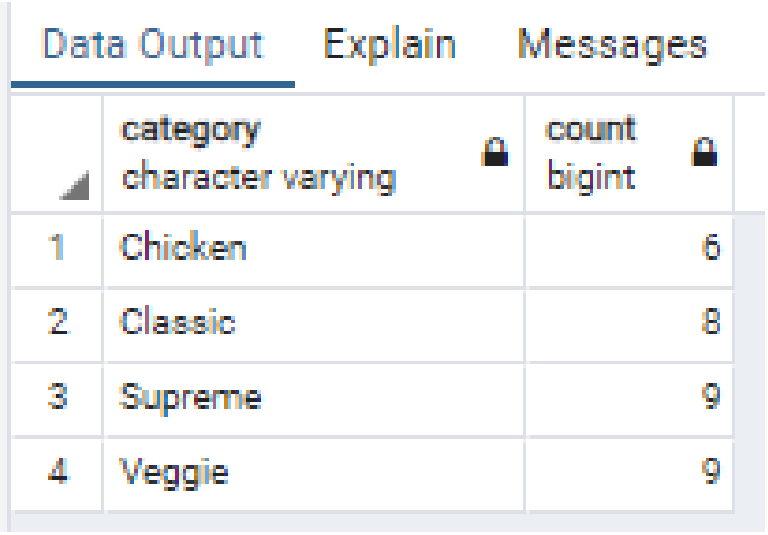


	time double precision	order_count bigint
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472

Question No: 08

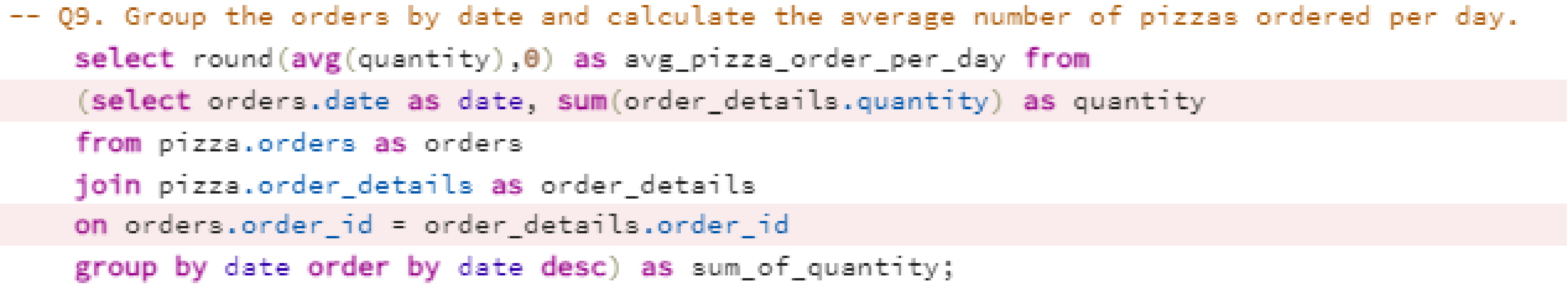


```
-- Q8. Join relevant tables to find the category-wise distribution of pizzas.
    select category, count(name) as count
    from pizza.pizza_types
    group by category
    order by category;
```

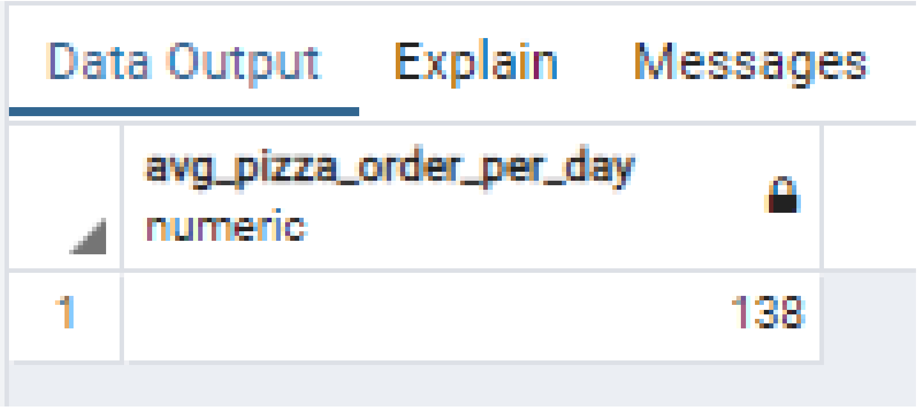


	category character varying	count bigint
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

Question No: 09

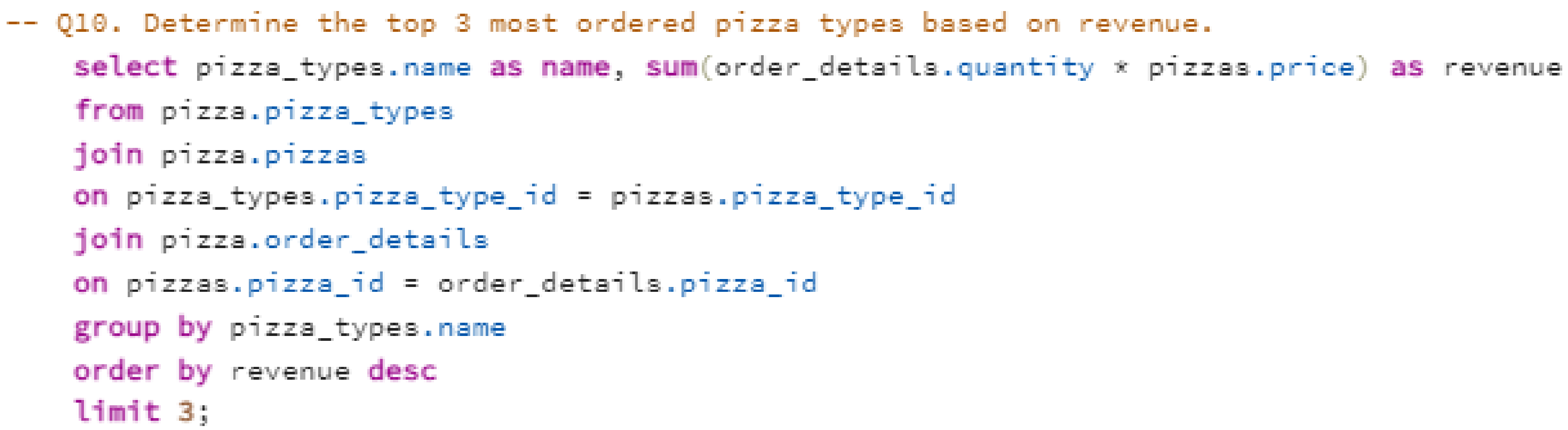


```
-- Q9. Group the orders by date and calculate the average number of pizzas ordered per day.
    select round(avg(quantity),0) as avg_pizza_order_per_day from
    (select orders.date as date, sum(order_details.quantity) as quantity
    from pizza.orders as orders
    join pizza.order_details as order_details
    on orders.order_id = order_details.order_id
    group by date order by date desc) as sum_of_quantity;
```

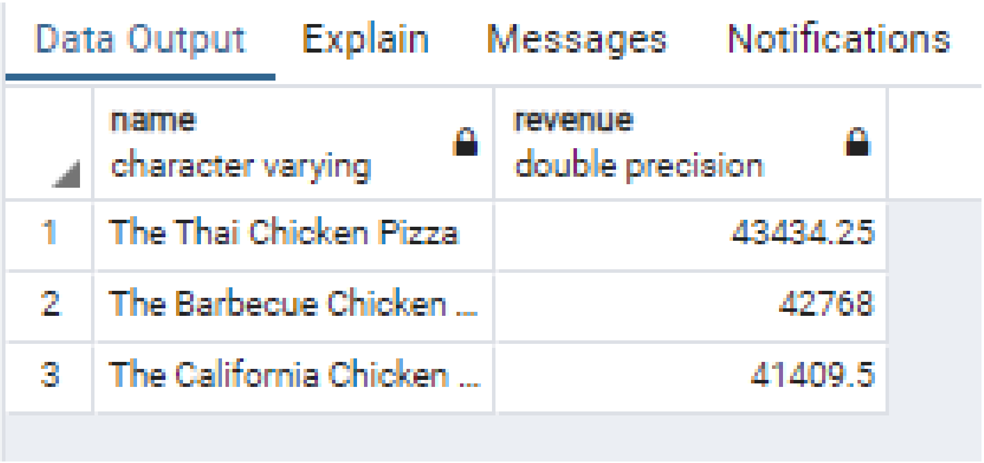


	avg_pizza_order_per_day numeric
1	138

Question No: 10

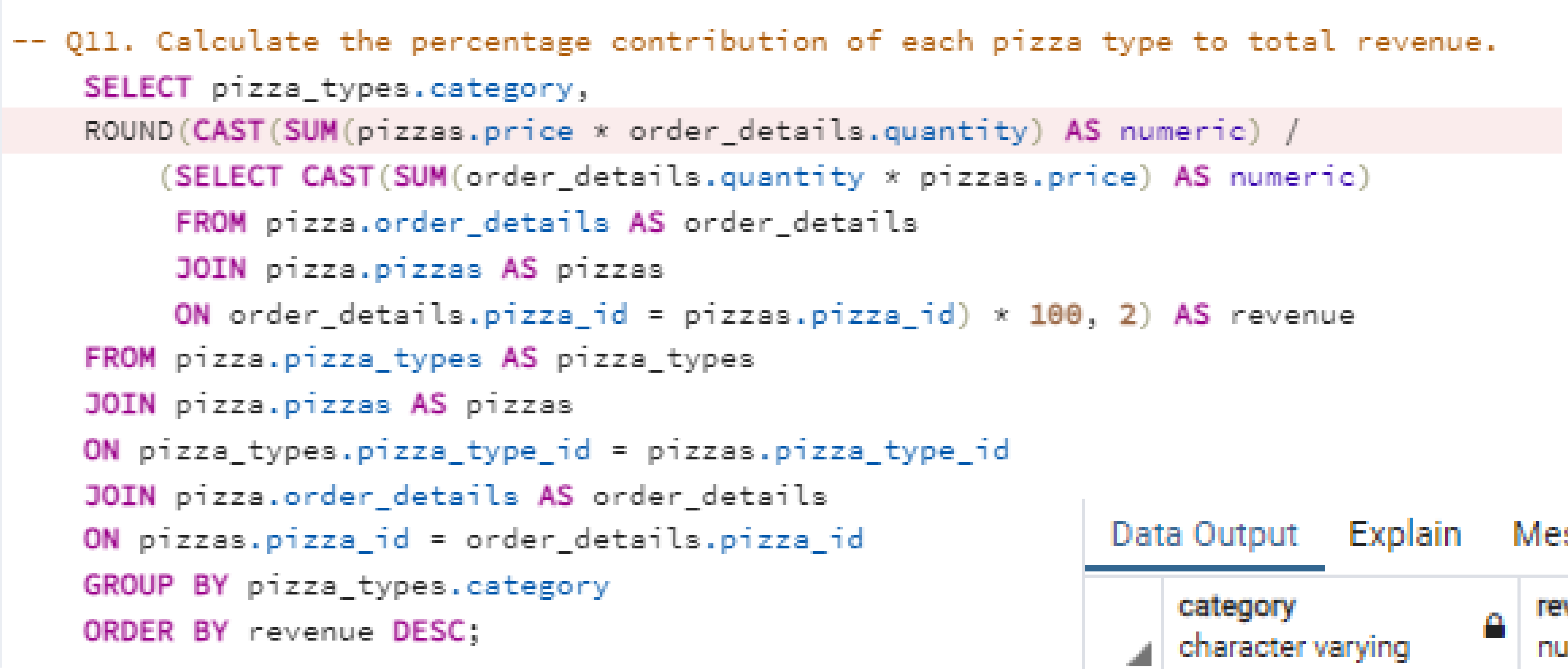


```
-- Q10. Determine the top 3 most ordered pizza types based on revenue.
    select pizza_types.name as name, sum(order_details.quantity * pizzas.price) as revenue
    from pizza.pizza_types
    join pizza.pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
    join pizza.order_details
    on pizzas.pizza_id = order_details.pizza_id
    group by pizza_types.name
    order by revenue desc
    limit 3;
```

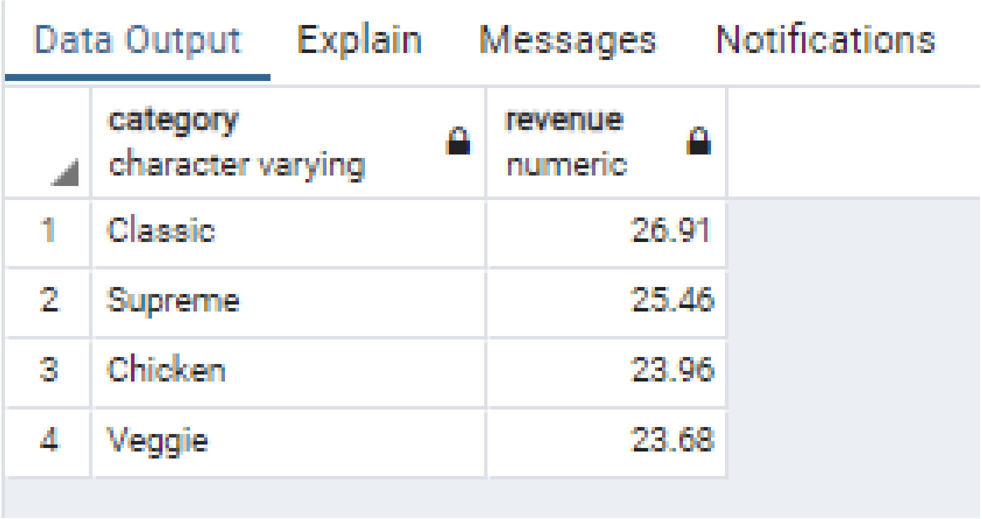


	name character varying	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken ...	42768
3	The California Chicken ...	41409.5

Question No: 11

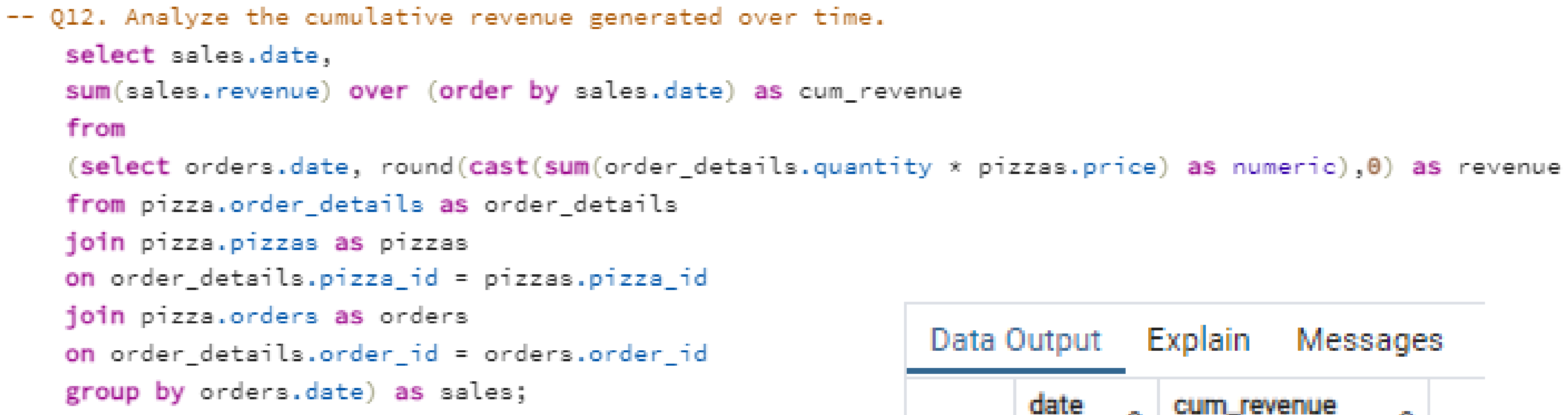


```
-- Q11. Calculate the percentage contribution of each pizza type to total revenue.
    SELECT pizza_types.category,
    ROUND(CAST(SUM(pizzas.price * order_details.quantity) AS numeric) /
        (SELECT CAST(SUM(order_details.quantity * pizzas.price) AS numeric)
         FROM pizza.order_details AS order_details
         JOIN pizza.pizzas AS pizzas
         ON order_details.pizza_id = pizzas.pizza_id) * 100, 2) AS revenue
    FROM pizza.pizza_types AS pizza_types
    JOIN pizza.pizzas AS pizzas
    ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN pizza.order_details AS order_details
    ON pizzas.pizza_id = order_details.pizza_id
    GROUP BY pizza_types.category
    ORDER BY revenue DESC;
```

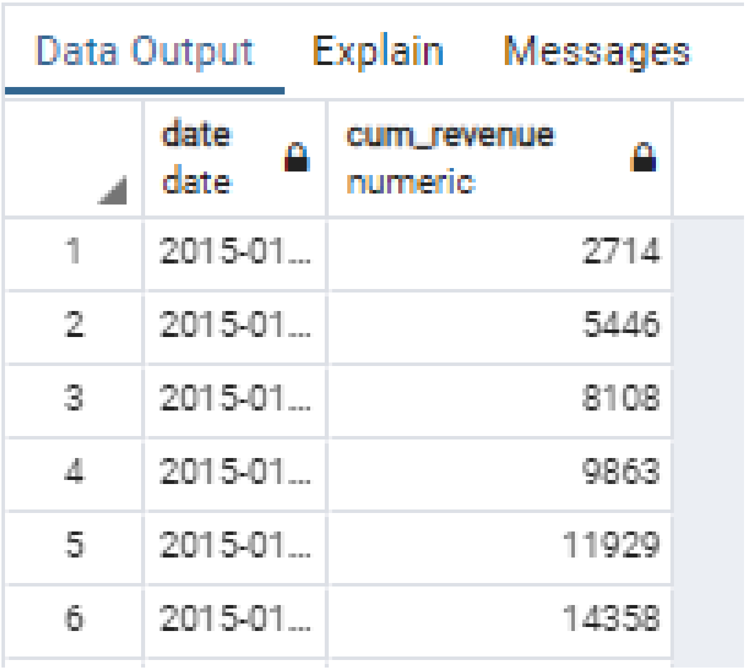


	category character varying	revenue numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

Question No: 12

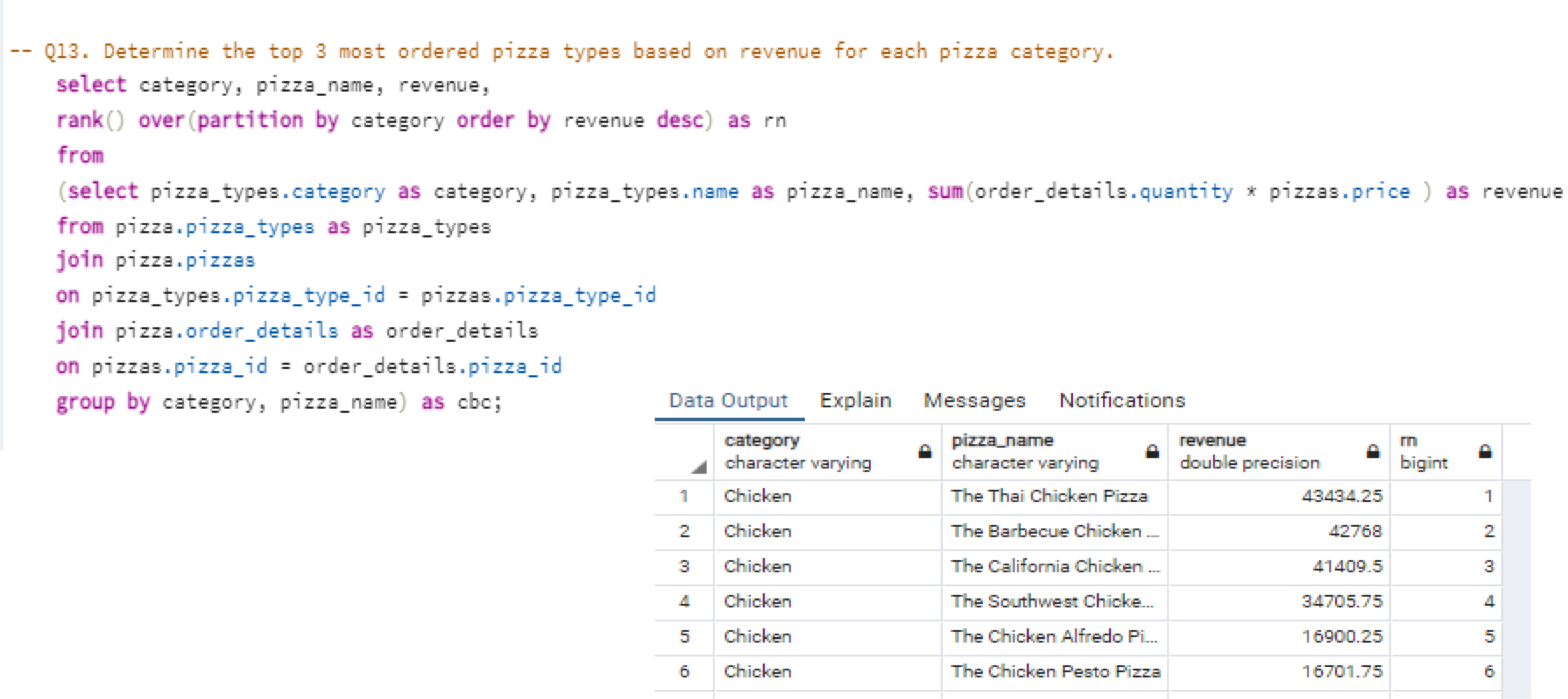


```
-- Q12. Analyze the cumulative revenue generated over time.
    select sales.date,
    sum(sales.revenue) over (order by sales.date) as cum_revenue
    from
    (select orders.date, round(cast(sum(order_details.quantity * pizzas.price) as numeric),0) as revenue
    from pizza.order_details as order_details
    join pizza.pizzas as pizzas
    on order_details.pizza_id = pizzas.pizza_id
    join pizza.orders as orders
    on order_details.order_id = orders.order_id
    group by orders.date) as sales;
```



	date date	cum_revenue numeric
1	2015-01...	2714
2	2015-01...	5446
3	2015-01...	8108
4	2015-01...	9863
5	2015-01...	11929
6	2015-01...	14358

Question No: 13



```sql
-- Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.
   select category, pizza_name, revenue,
   rank() over(partition by category order by revenue desc) as rn
   from
   (select pizza_types.category as category, pizza_types.name as pizza_name, sum(order_details.quantity * pizzas.price ) as revenue
   from pizza.pizza_types as pizza_types
   join pizza.pizzas
   on pizza_types.pizza_type_id = pizzas.pizza_type_id
   join pizza.order_details as order_details
   on pizzas.pizza_id = order_details.pizza_id
   group by category, pizza_name) as cbc;
```

	category character varying	pizza_name character varying	revenue double precision	rn bigint
1	Chicken	The Thai Chicken Pizza	43434.25	1
2	Chicken	The Barbecue Chicken ...	42768	2
3	Chicken	The California Chicken ...	41409.5	3
4	Chicken	The Southwest Chicke...	34705.75	4
5	Chicken	The Chicken Alfredo Pi...	16900.25	5
6	Chicken	The Chicken Pesto Pizza	16701.75	6

/saj0cse /saj0cse /saj0cse shakil.cse40@gmail.com