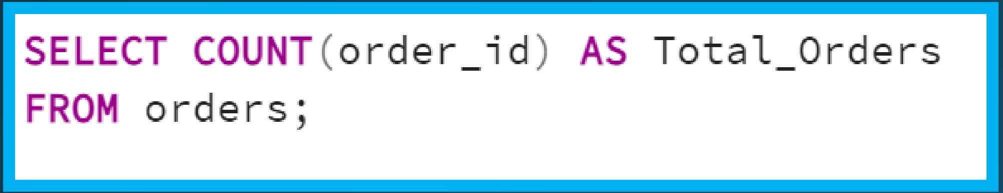
Report on Pizza Sales Analysis

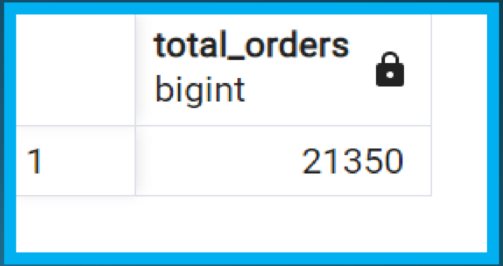
PREPARED BY
SHAKIL AHAMMED

In this project I have utilized SQL queries to solve questions that were related to pizza sales.

1. Retrieve the total number of orders placed.

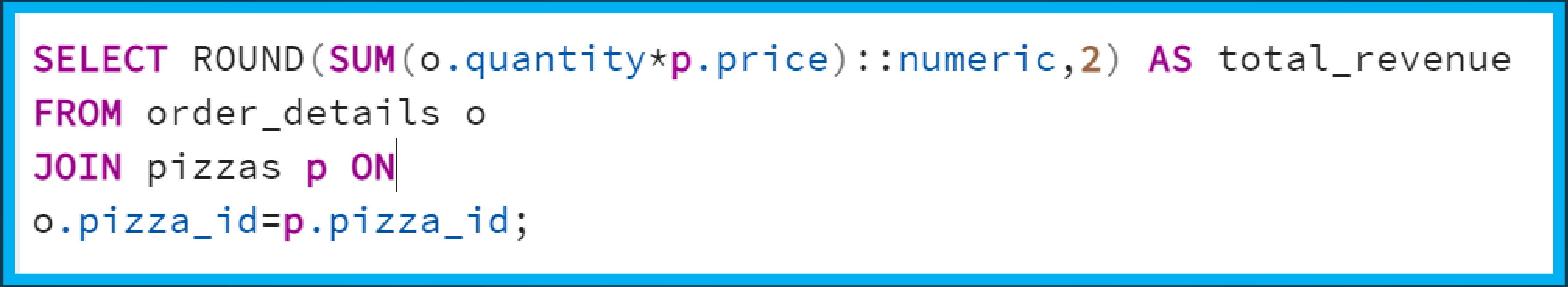


```sql
SELECT COUNT(order_id) AS Total_Orders
FROM orders;
```

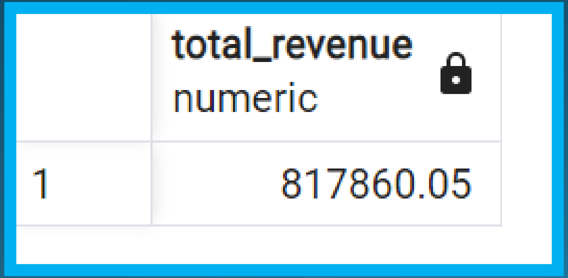


	total_orders bigint
1	21350

2. Calculate the total revenue generated from pizza sales.

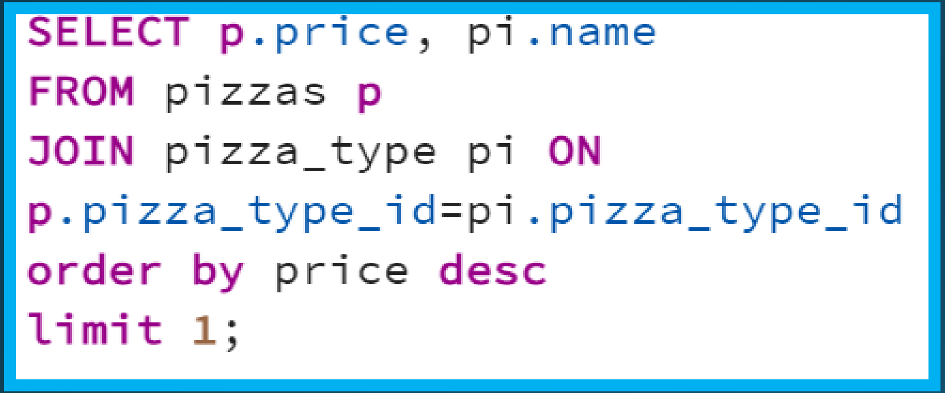


```
SELECT ROUND(SUM(o.quantity*p.price)::numeric,2) AS total_revenue
FROM order_details o
JOIN pizzas p ON
o.pizza_id=p.pizza_id;
```

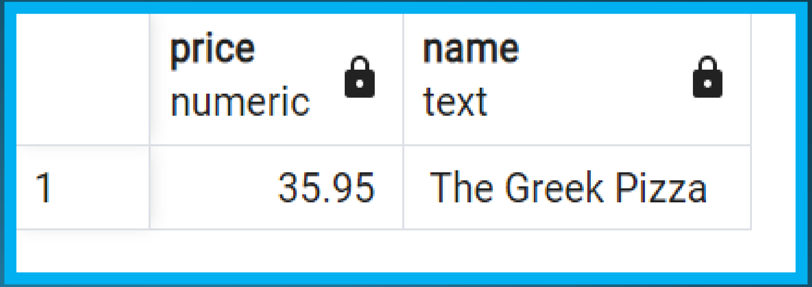


	total_revenue numeric
1	817860.05

3. Identify the highest-priced pizza.

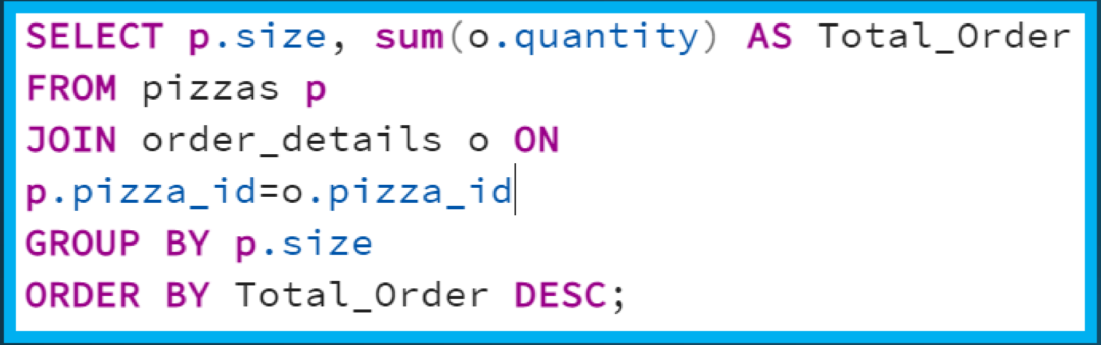


```sql
SELECT p.price, pi.name
FROM pizzas p
JOIN pizza_type pi ON
p.pizza_type_id=pi.pizza_type_id
order by price desc
limit 1;
```

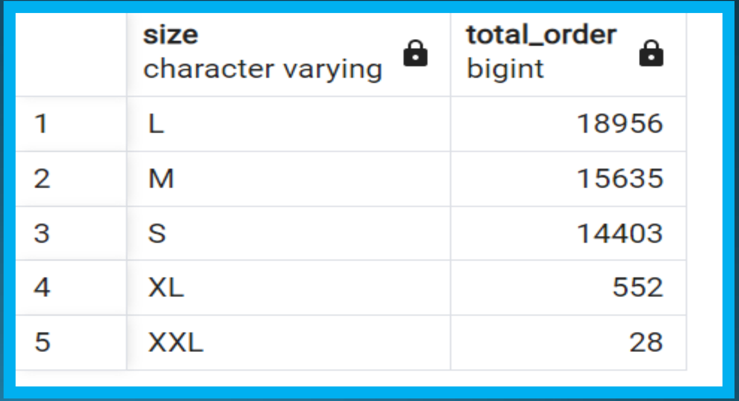


	price numeric 🔒	name text 🔒
1	35.95	The Greek Pizza

4. Identify the most common pizza size ordered.

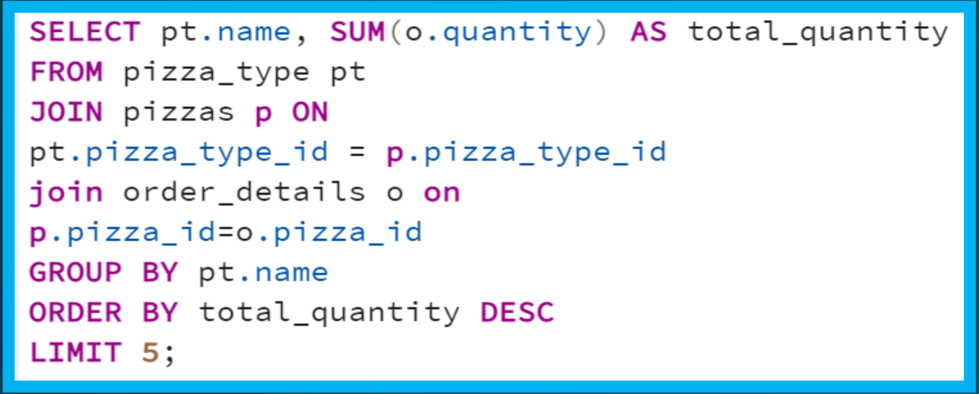


```
SELECT p.size, sum(o.quantity) AS Total_Order
FROM pizzas p
JOIN order_details o ON
p.pizza_id=o.pizza_id
GROUP BY p.size
ORDER BY Total_Order DESC;
```

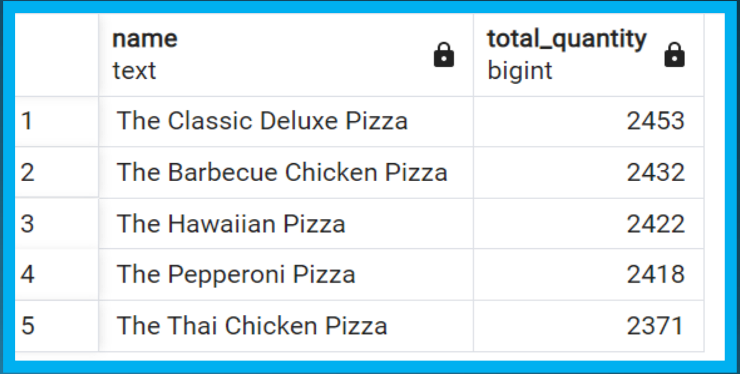


	size character varying	total_order bigint
1	L	18956
2	M	15635
3	S	14403
4	XL	552
5	XXL	28

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

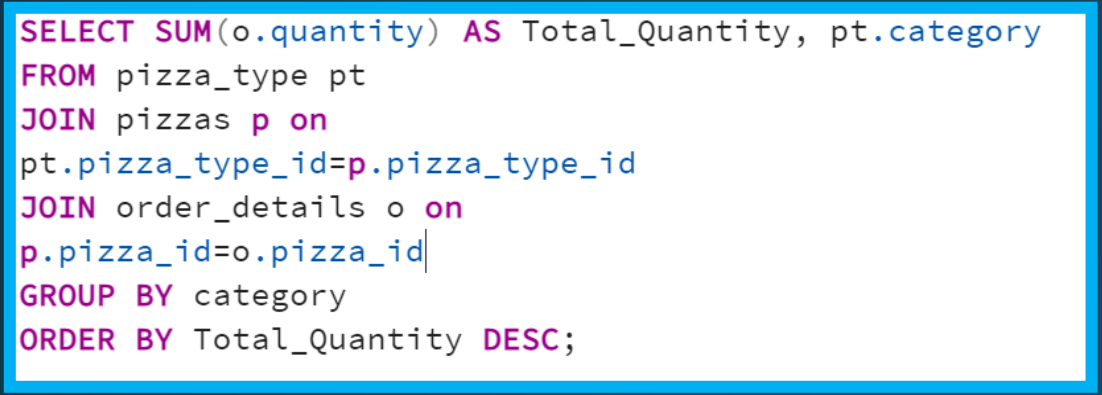


```sql
SELECT pt.name, SUM(o.quantity) AS total_quantity
FROM pizza_type pt
JOIN pizzas p ON
pt.pizza_type_id = p.pizza_type_id
join order_details o on
p.pizza_id=o.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

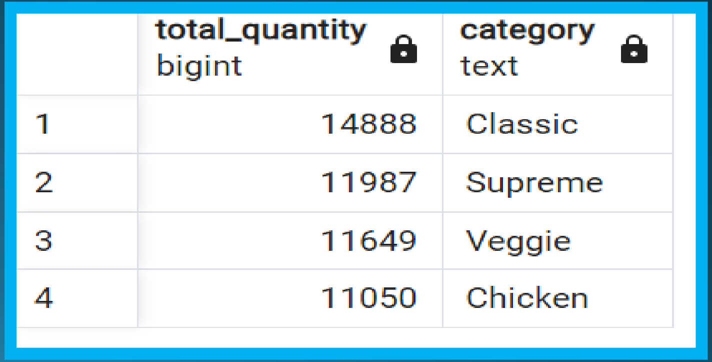


	name text	total_quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

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

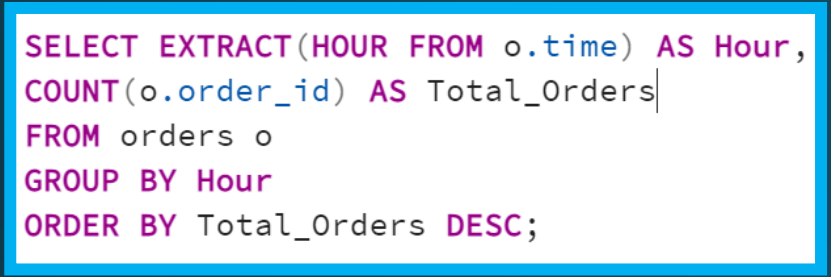


```sql
SELECT SUM(o.quantity) AS Total_Quantity, pt.category
FROM pizza_type pt
JOIN pizzas p on
pt.pizza_type_id=p.pizza_type_id
JOIN order_details o on
p.pizza_id=o.pizza_id
GROUP BY category
ORDER BY Total_Quantity DESC;
```

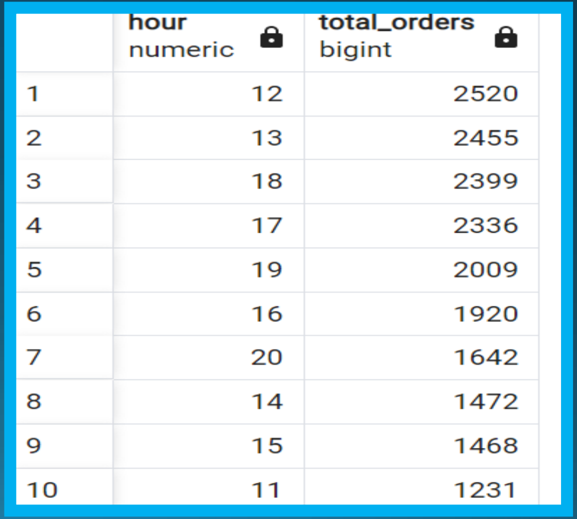


	total_quantity 🔒 bigint	category 🔒 text
1	14888	Classic
2	11987	Supreme
3	11649	Veggie
4	11050	Chicken

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

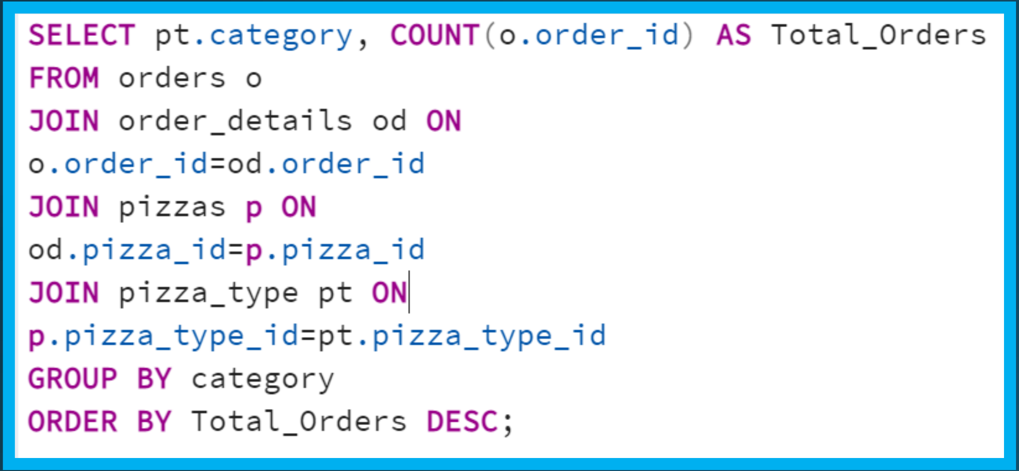


```sql
SELECT EXTRACT(HOUR FROM o.time) AS Hour,
COUNT(o.order_id) AS Total_Orders
FROM orders o
GROUP BY Hour
ORDER BY Total_Orders DESC;
```

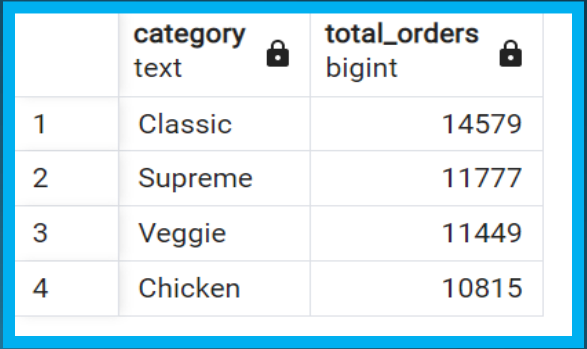


	hour numeric	total_orders bigint
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009
6	16	1920
7	20	1642
8	14	1472
9	15	1468
10	11	1231

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

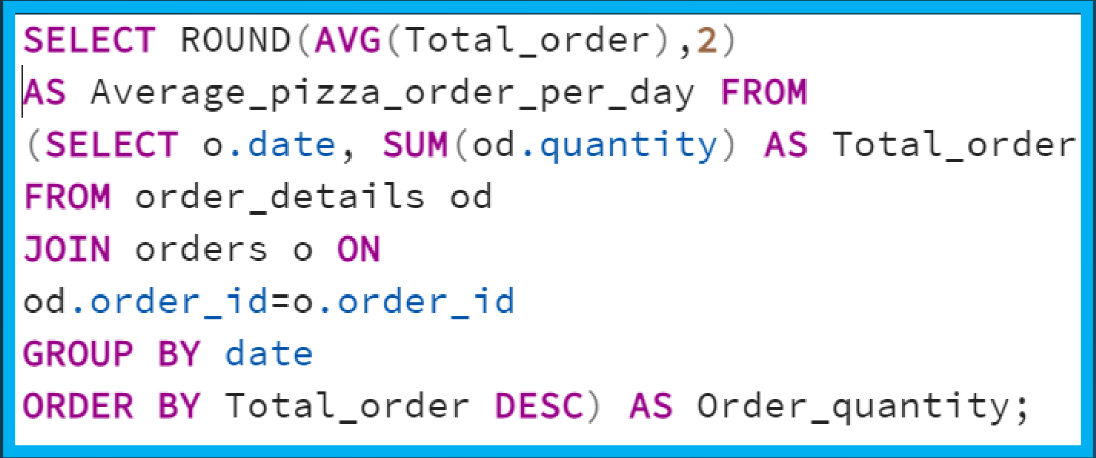


```
SELECT pt.category, COUNT(o.order_id) AS Total_Orders
FROM orders o
JOIN order_details od ON
o.order_id=od.order_id
JOIN pizzas p ON
od.pizza_id=p.pizza_id
JOIN pizza_type pt ON
p.pizza_type_id=pt.pizza_type_id
GROUP BY category
ORDER BY Total_Orders DESC;
```

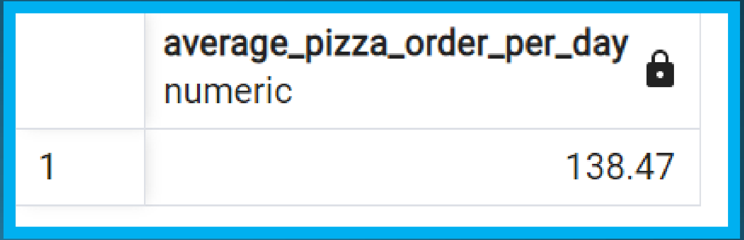


	category text	total_orders bigint
1	Classic	14579
2	Supreme	11777
3	Veggie	11449
4	Chicken	10815

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

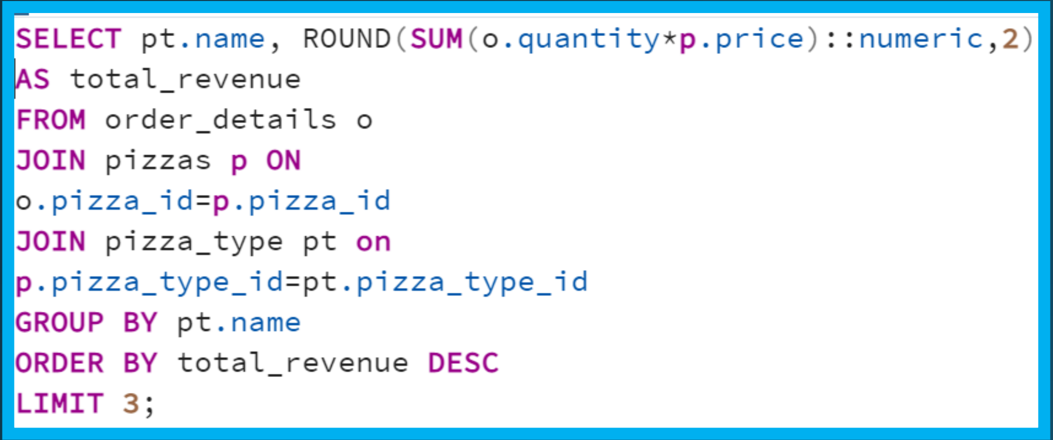


```
SELECT ROUND(AVG(Total_order),2)
AS Average_pizza_order_per_day FROM
(SELECT o.date, SUM(od.quantity) AS Total_order
FROM order_details od
JOIN orders o ON
od.order_id=o.order_id
GROUP BY date
ORDER BY Total_order DESC) AS Order_quantity;
```

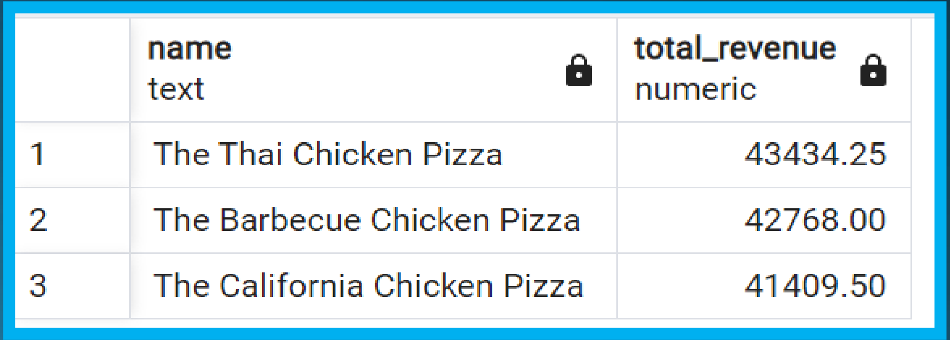


	average_pizza_order_per_day numeric
1	138.47

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

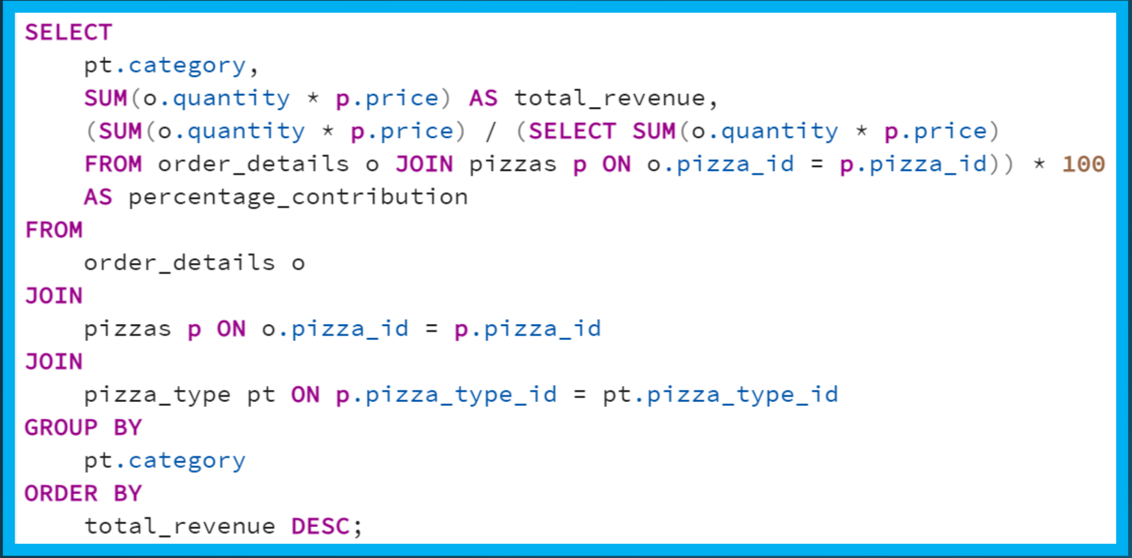


```
SELECT pt.name, ROUND(SUM(o.quantity*p.price)::numeric,2)
AS total_revenue
FROM order_details o
JOIN pizzas p ON
o.pizza_id=p.pizza_id
JOIN pizza_type pt on
p.pizza_type_id=pt.pizza_type_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3;
```

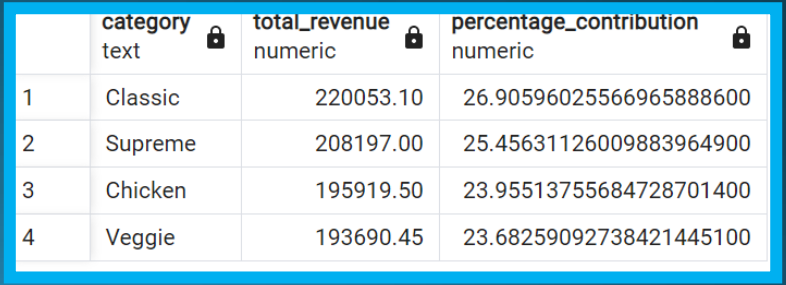


	name text	total_revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

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



```
SELECT
    pt.category,
    SUM(o.quantity * p.price) AS total_revenue,
    (SUM(o.quantity * p.price) / (SELECT SUM(o.quantity * p.price)
    FROM order_details o JOIN pizzas p ON o.pizza_id = p.pizza_id)) * 100
    AS percentage_contribution
FROM
    order_details o
JOIN
    pizzas p ON o.pizza_id = p.pizza_id
JOIN
    pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY
    pt.category
ORDER BY
    total_revenue DESC;
```



	category text	total_revenue numeric	percentage_contribution numeric
1	Classic	220053.10	26.9059602556696588600
2	Supreme	208197.00	25.4563112600988396490
3	Chicken	195919.50	23.9551375568472870140
4	Veggie	193690.45	23.6825909273842144510