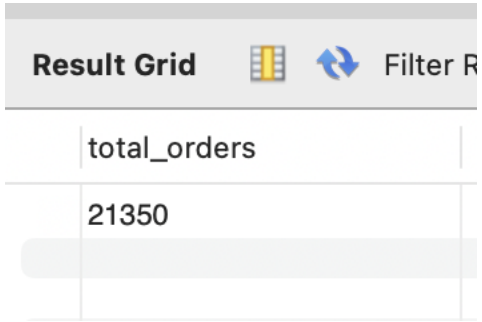
Pizza Sales Analysis

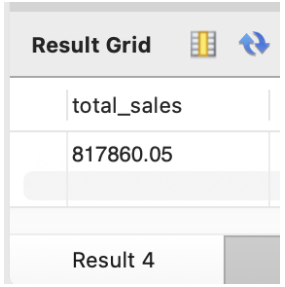
1. Retrieve the total no of orders placed

```sql
select count(order_id) as total_orders from orders;
```



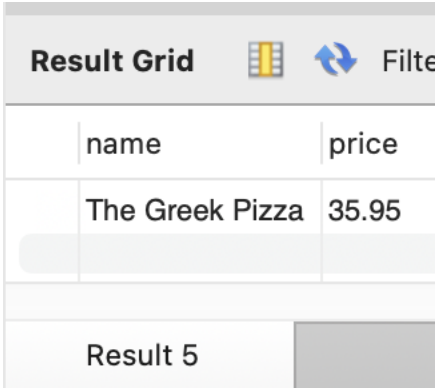
2. Calculate the “total revenue” generated from pizza sales

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



3. Identify the highest priced pizza

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



4. Identify the most common pizza size ordered

```sql
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;
```

Result Grid   Filter R

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

```sql
SELECT
    pizza_types.name, sum(order_details.quantity) as quantity
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.name order by quantity limit 5;
```

Result Grid   Filter Rows: 

name	quantity
The Brie Carre Pizza	490
The Mediterranean Pizza	934
The Calabrese Pizza	937
The Spinach Supreme Pizza	950
The Soppressata Pizza	961

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

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
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 quantity DESC;
```

Result Grid

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

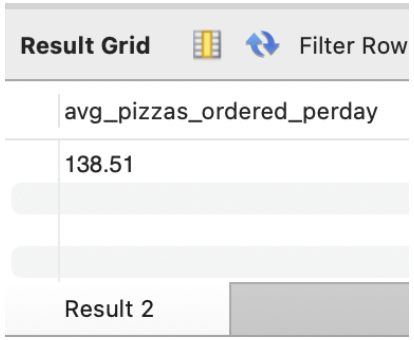
```sql
SELECT
    HOUR(time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(time);
```

Result Grid

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198

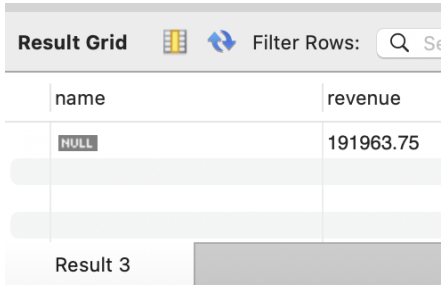
8. Group the orders by date and calculate the average number of pizzas ordered per day

```sql
SELECT
    round(AVG(quantity),2) as avg_pizzas_ordered_perday
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```



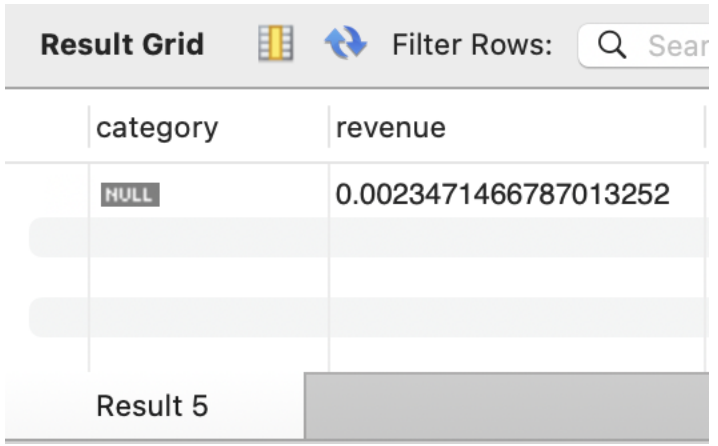
9. Determine the top 3 most ordered pizza types bsaed on revenue

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



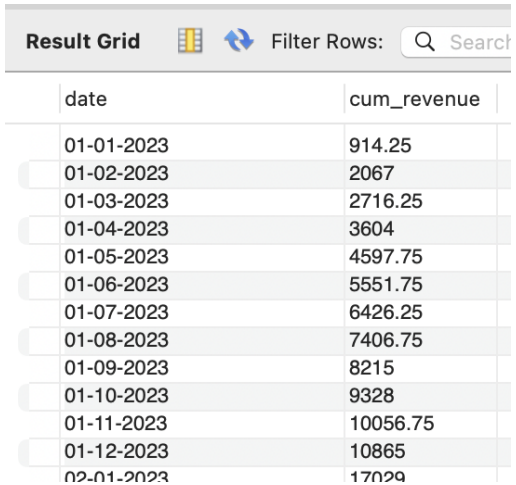
10. Calculate the percentage contribution of each pizza type to total revenue

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            order_details
                JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)/100 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;
```



11. Analyze the cumulative revenue generated over time

```sql
SELECT date, sum(revenue) over(ORDER BY date) as cum_revenue
from
(SELECT
    orders.date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    orders ON orders.date = order_details.order_id
GROUP BY orders.date) as sales;
```



Result Grid Filter Rows: Search

date	cum_revenue
01-01-2023	914.25
01-02-2023	2067
01-03-2023	2716.25
01-04-2023	3604
01-05-2023	4597.75
01-06-2023	5551.75
01-07-2023	6426.25
01-08-2023	7406.75
01-09-2023	8215
01-10-2023	9328
01-11-2023	10056.75
01-12-2023	10865
02-01-2023	17029

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

```sql
SELECT name, revenue
FROM
(SELECT category, name, revenue, rank() over(partition by category ORDER BY revenue desc) as rn
FROM
(SELECT
    pizza_types.category,
    pizza_types.name,
    SUM((order_details.quantity) * pizzas.price) 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 , pizza_types.name) as a) as b where rn<=3 ;
```