LARANA PIZZA

WELCOME TO

LARANA PIZZA BY GURU

LARANA PIZZA

VISSION & MISSION

VISSION

The vision of this project is to leverage data-driven insights from pizza sales to enhance decision-making, optimize inventory management, and boost overall profitability. By analyzing customer preferences and sales trends, we aim to drive more targeted marketing and operational efficiency.

MISSION

In this data analysis project, we utilized MySQL to analyze pizza sales data by querying key metrics such as total sales by day, average order value, and popular pizza types. We created SQL queries to answer specific questions like "Which pizza flavor generates the highest revenue?" and "What is the average number of pizzas sold per transaction?" This approach helped uncover trends, optimize inventory, and provide actionable insights for improving sales strategies.

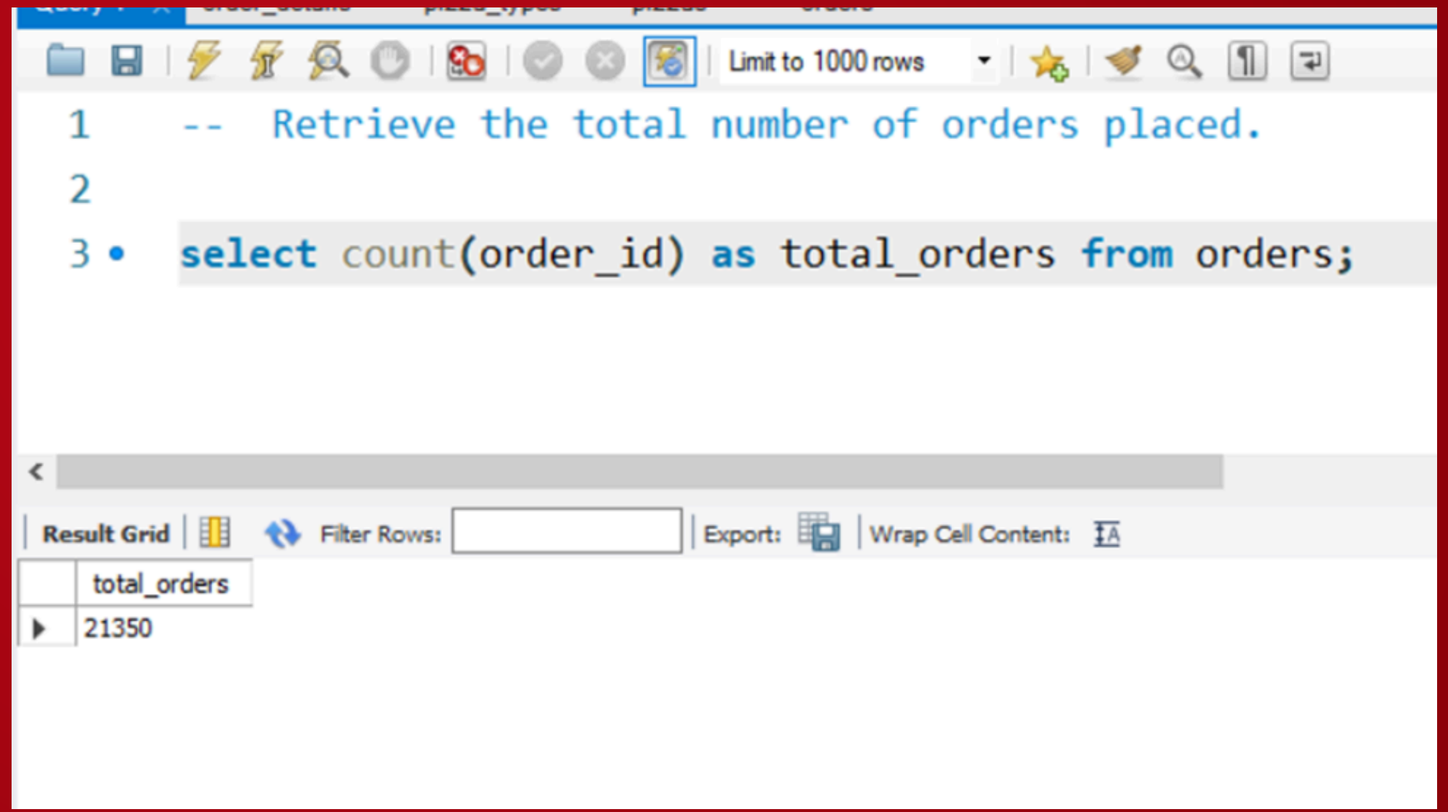
LARANA PIZZA

HELLO!!

MY NAME IS GURUDAS U. JADHAV
HERE ONWARDS PROJECT
BEGINS!

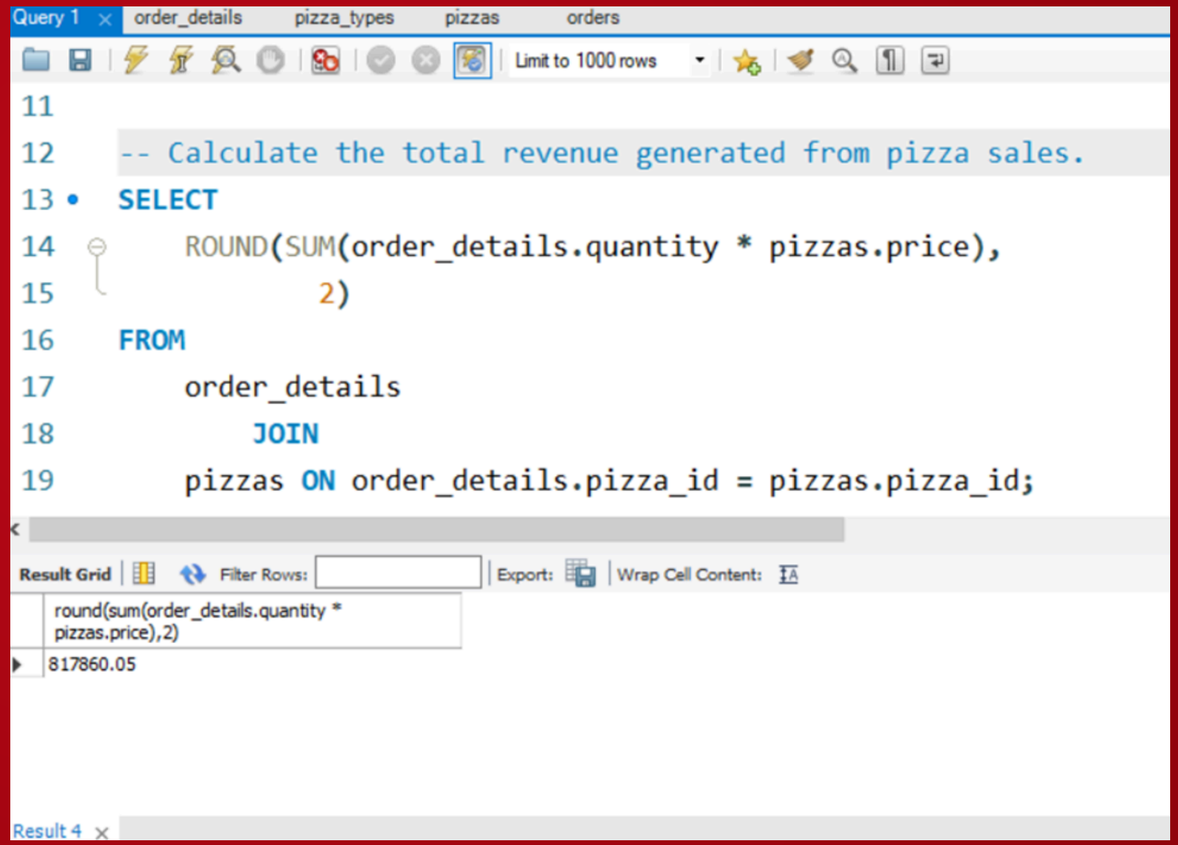
LARANA PIZZA

-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



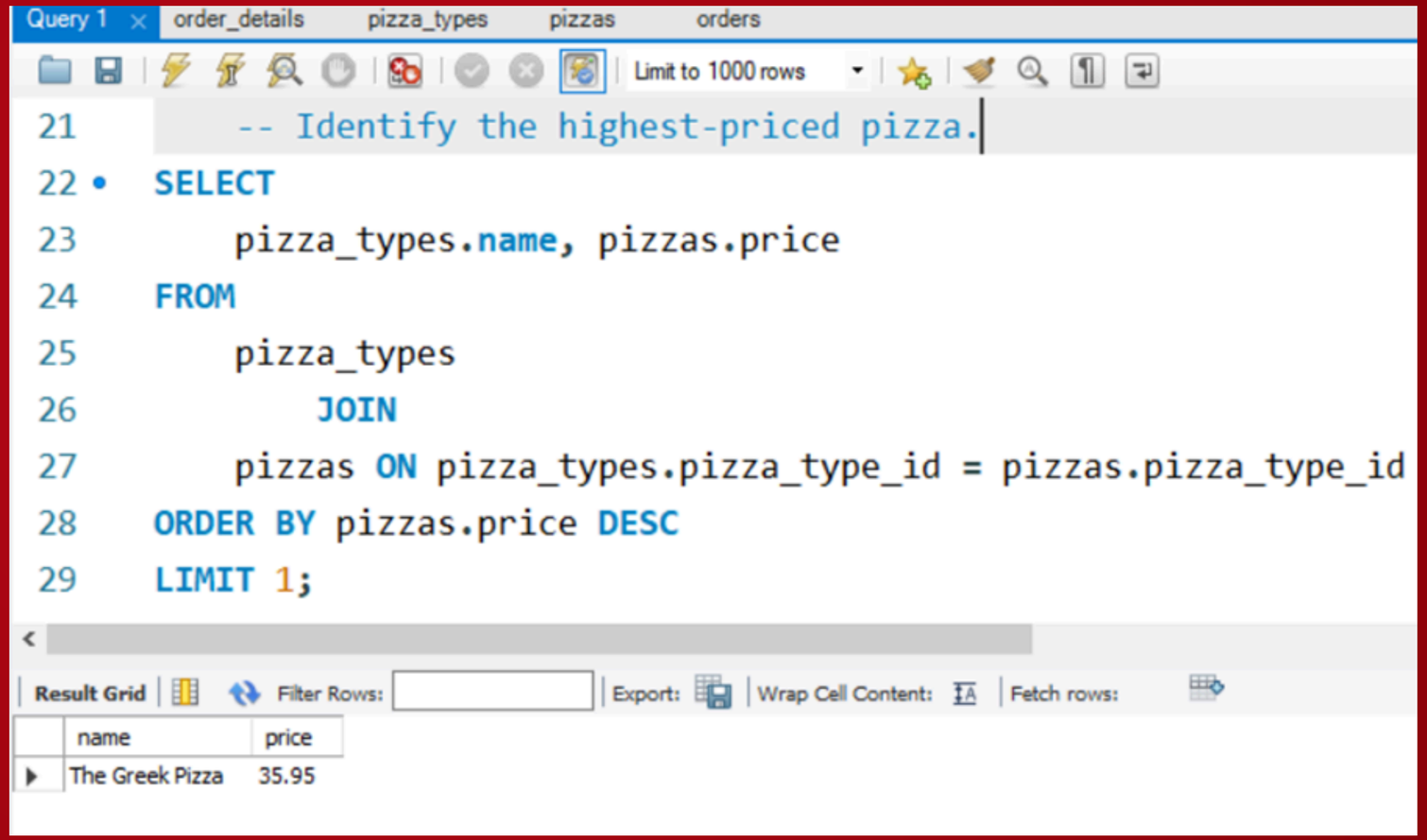
LARANA PIZZA

-- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



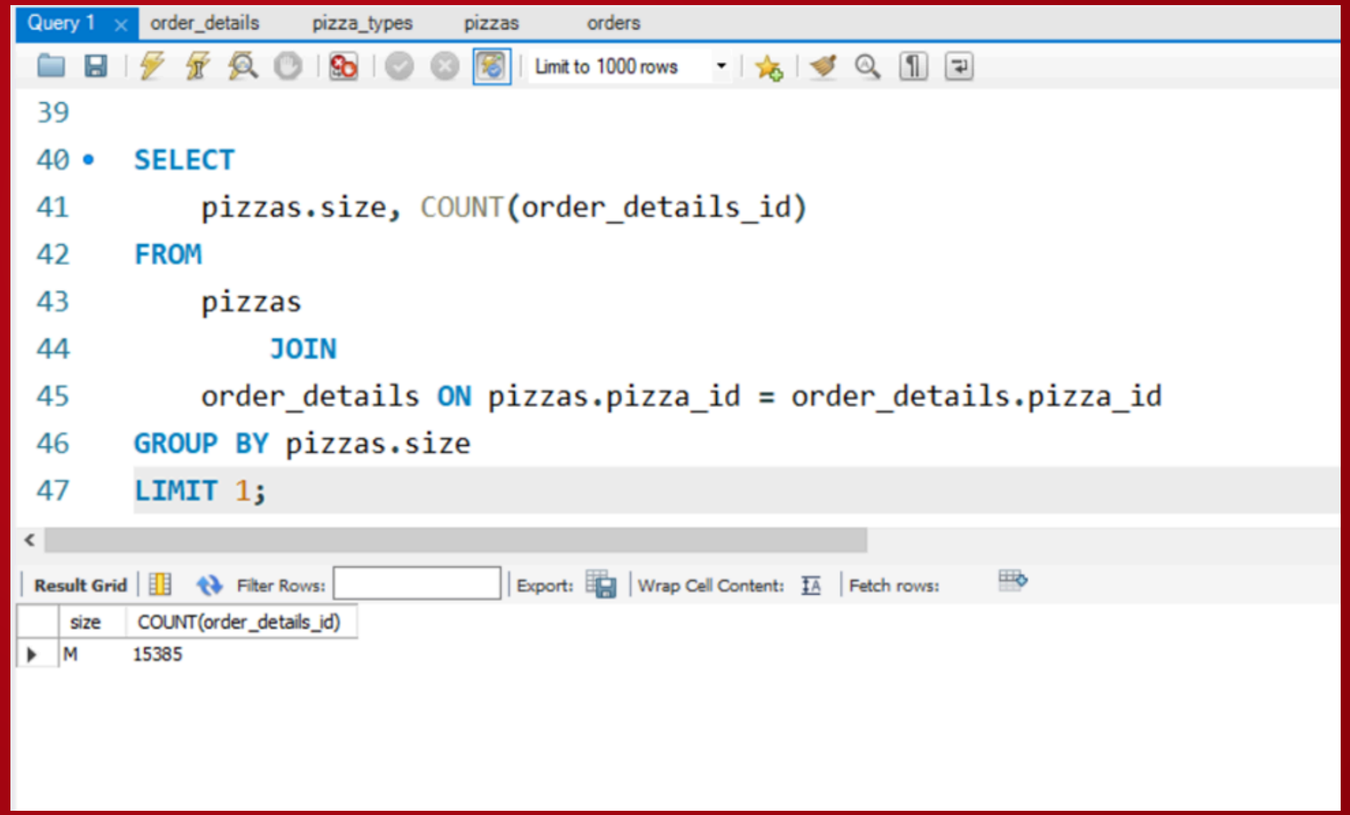
LARANA PIZZA

-- IDENTIFY THE HIGHEST-PRICED PIZZA.



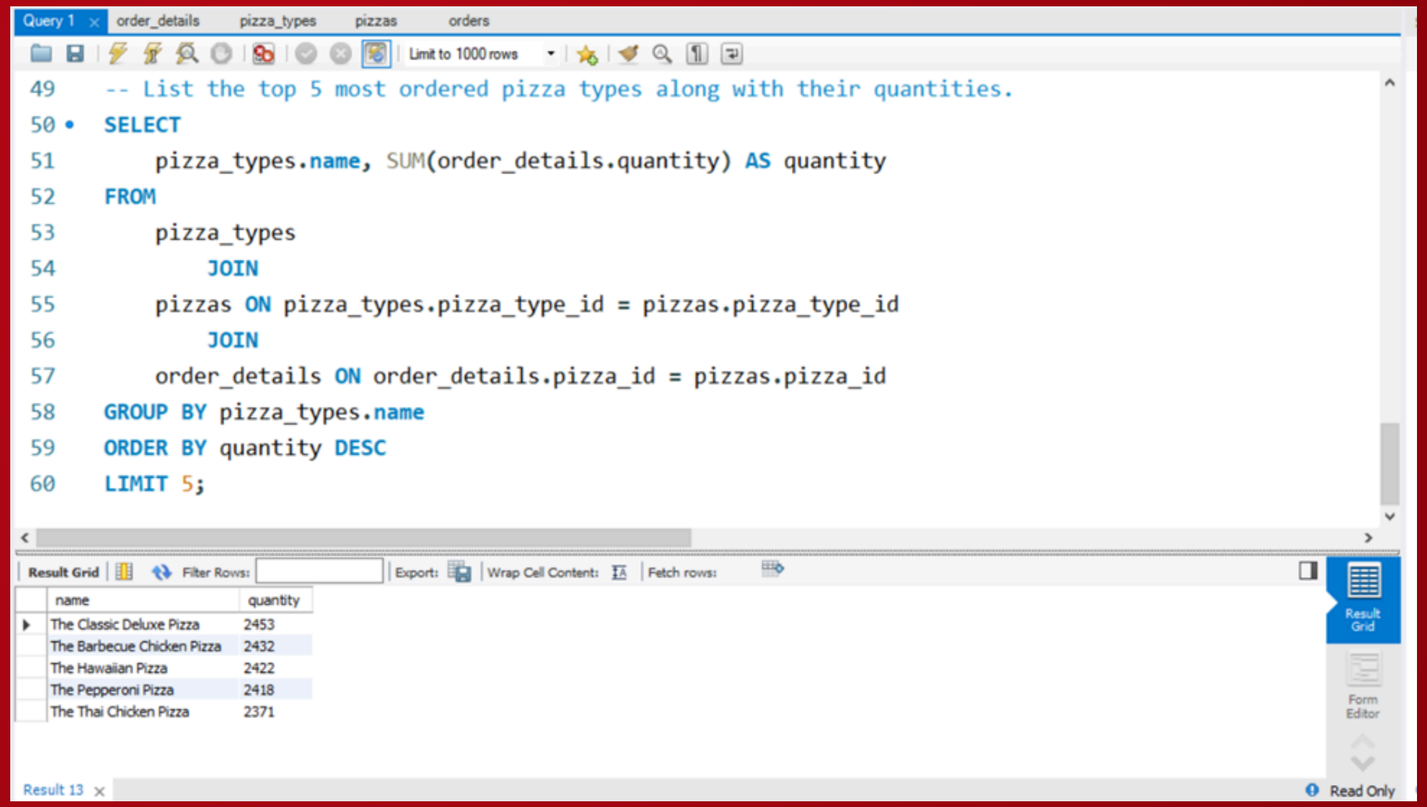
LARANA PIZZA

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



LARANA PIZZA

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



LARANA PIZZA

-- USE NECESSARY JOIN TO FIND TOTALQUANTITY OF EACH PIZZA CATEGORY ORDERED

```
62
63      -- use necessary join to find totalquantity of each pizza category ordered
64
65  •   select pizza_types.category,
66      count(order_details.quantity) as quantity
67      from pizza_types join pizzas
68      on pizza_types.pizza_type_id=pizzas.pizza_type_id
69      join order_details on order_details.pizza_id=pizzas.pizza_id
70      group by pizza_types.category order by quantity desc;
71
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Classic	14579
Supreme	11777
Veggie	11449
Chicken	10815

LARANA PIZZA

-- DETERMINE THE DISTRIBUTION OF ORDERS NY HOURS PER DAY

```
76
77      -- determine the distribution of orders ny hours per day
78
79  •   SELECT
80          HOUR(order_time), COUNT(order_id)
81      FROM
82          orders
83      GROUP BY HOUR(order_time);
84
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

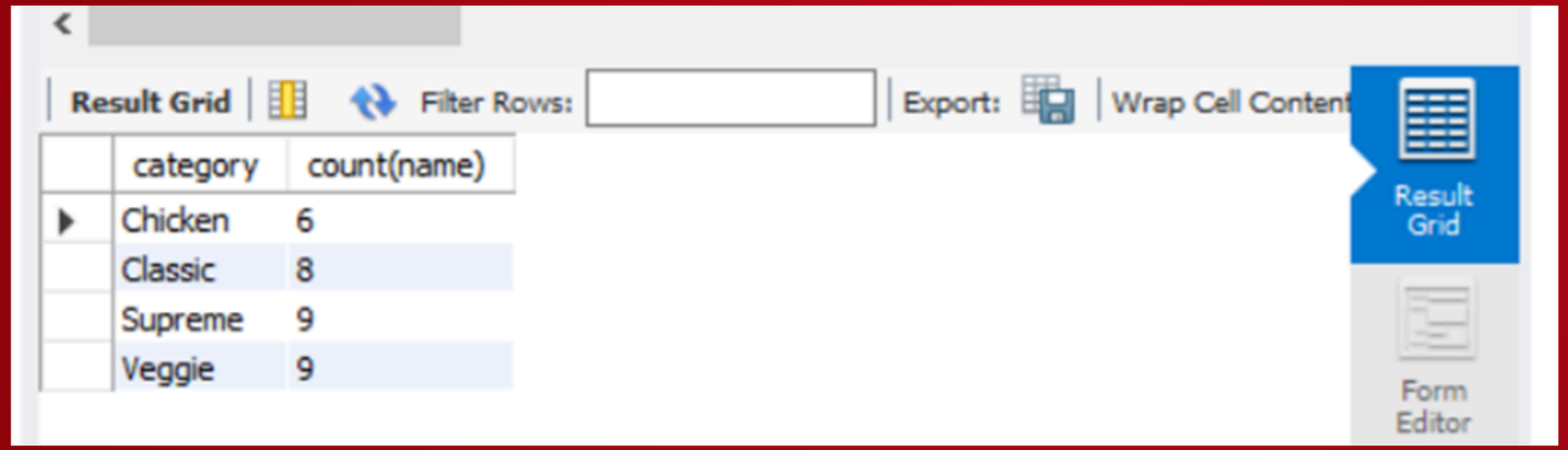
HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8

Result Grid
Form Editor
Field Types
Query Stats

LARANA PIZZA

-- CATEGORY WISE DISTRIBUTION OF PIZZAS

```
-- category wise distribution of pizzas
select category ,count(name)from pizza_types
group by category;
```



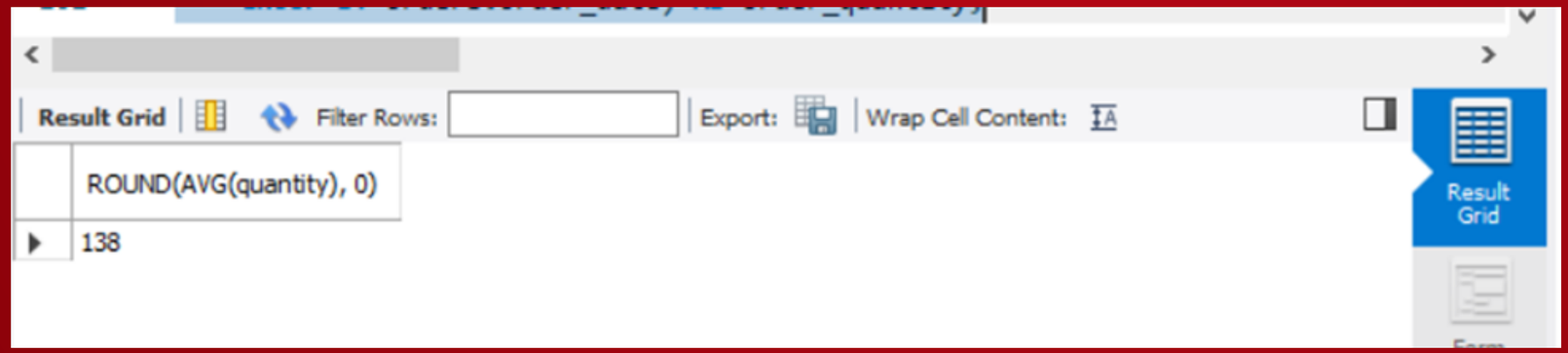
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

LARANA PIZZA

-- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE -- NUMBER OF PIZZAS ORDER PER DAY

```sql
-- group the orders by date and calculate the average
-- number of pizzas order per day

SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



LARANA PIZZA

-- DETERMINE THE TOP 3 MOST ORDER PIZZA TYPES BASED ON REVENUE

```sql
-- determine the top 3 most order pizza types based on revenue

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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

LARANA PIZZA

-- CALCULATE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TATAL REVENUE

```sql
-- calculate percentage contribution of each pizza type to tatal revenue

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

LARANA PIZZA

-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
138         -- analyze the cumulative revenue generated over time
139       SELECT order_date,
140           SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
141   FROM (
142       SELECT orders.order_date,
143           SUM(order_details.quantity * pizzas.price) AS revenue
144       FROM order_details
145       JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
146       JOIN orders ON orders.order_id = order_details.order_id  -- Adjusted this line
147       GROUP BY orders.order_date
148   ) AS sales;
149
150
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	cum_revenue
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
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004

Result Grid
Form Editor
Field Types
Query Stats

LARANA PIZZA

-- DETERMINE TOP 3 MOST ORDER PIZZA TYPES -- BASED ON REVENUE FOR EACH PIZZA CATEGORY

```sql
-- determine top 3 most order pizza types
-- based on revenue for each pizza category
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 AS 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	revenue
The Southwest Chicken Pizza	34705.75
The Chicken Alfredo Pizza	16900.25
The Chicken Pesto Pizza	16701.75
The Greek Pizza	28454.100000000013
The Italian Capocollo Pizza	25094
The Napolitana Pizza	24087
The Big Meat Pizza	22968
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Pepper Salami Pizza	25529
The Prosciutto and Arugula Pizza	24193.25
The Soppressata Pizza	16425.75
The Calabrese Pizza	15934.25
The Spinach Supreme Pizza	15277.75
The Brie Carre Pizza	11588.4999999999

LARANA PIZZA

THANK YOU!