**Pizza\_db Data Analysis using SQL**

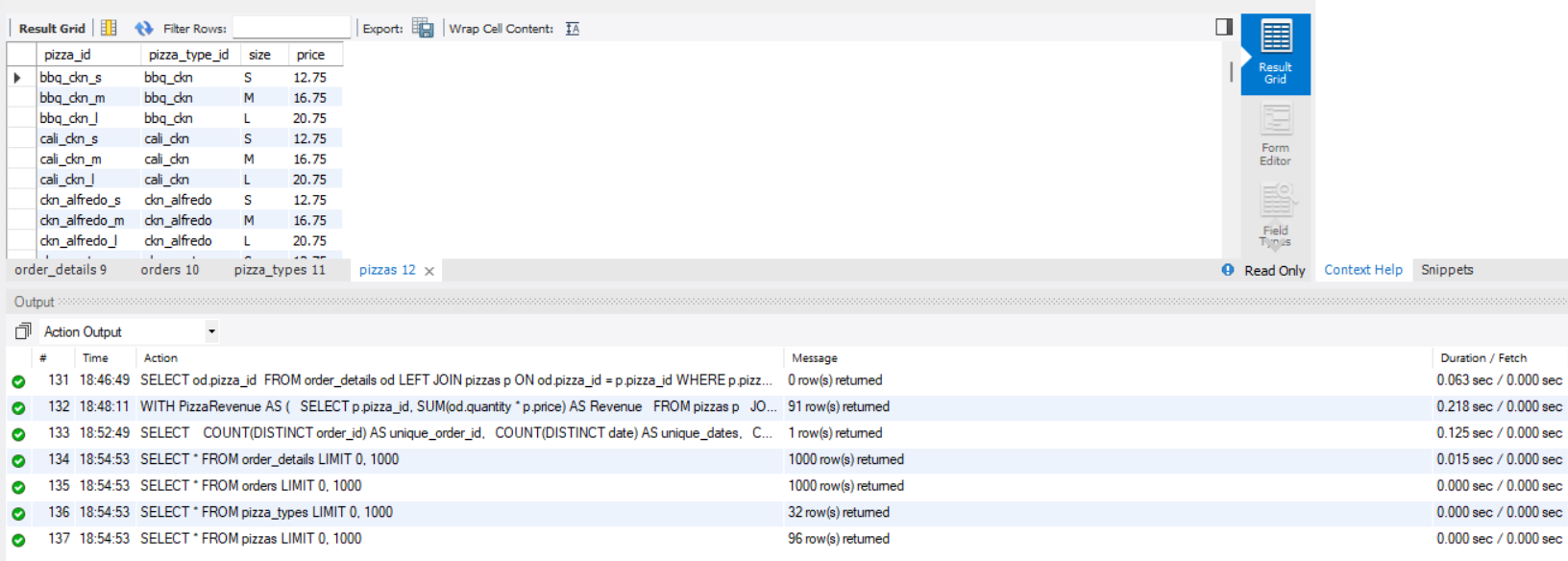
**# Getting an idea about all the features of data**

SELECT \* FROM order\_details;

SELECT \* FROM orders;

SELECT \* FROM pizza\_types;

SELECT \* FROM pizzas;



**#Value counts of all features**

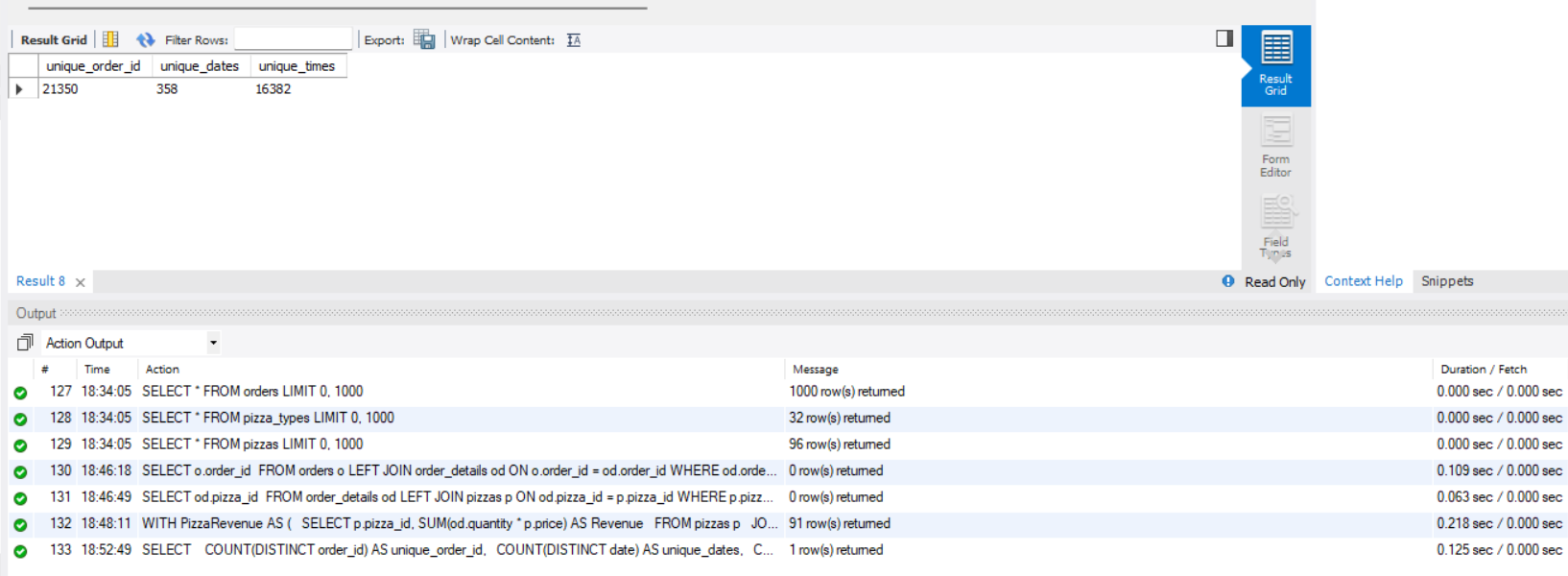
SELECT

COUNT(DISTINCT order\_id) AS unique\_order\_id,

COUNT(DISTINCT date) AS unique\_dates,

COUNT(DISTINCT time) AS unique\_times

FROM orders;



1. order\_id, has '21350' unique values
2. dates has '358' unique values
3. times has '16382' unique values

**#sales Revenue for each pizzas id**

WITH PizzaRevenue AS (

SELECT p.pizza\_id, SUM(od.quantity \* p.price) AS Revenue

FROM pizzas p

JOIN order\_details od ON od.pizza\_id = p.pizza\_id

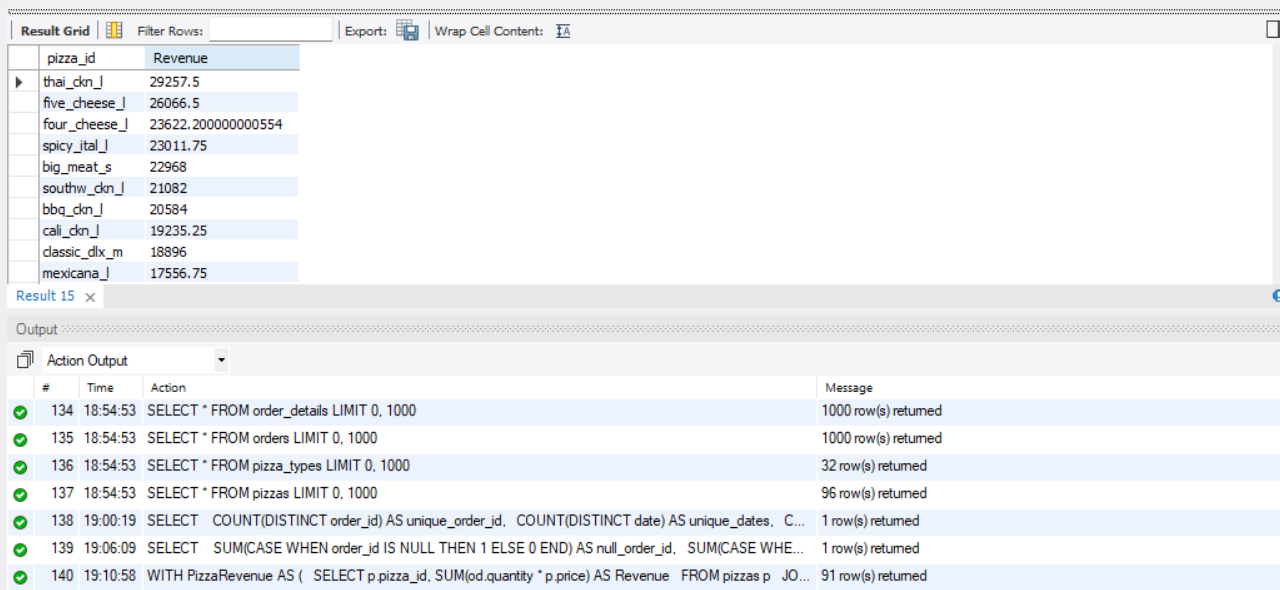
GROUP BY p.pizza\_id

)

SELECT \*

FROM PizzaRevenue

ORDER BY Revenue DESC;



**#Detect unusually high quantity orders (outlier analysis)**

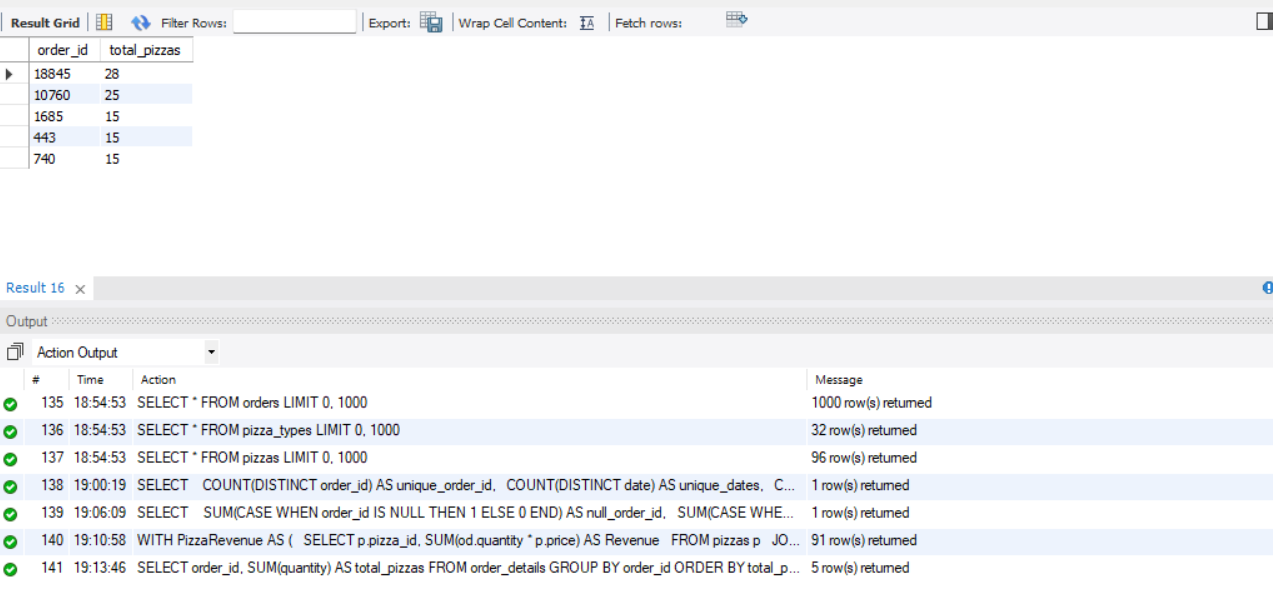
SELECT order\_id, SUM(quantity) AS total\_pizzas

FROM order\_details

GROUP BY order\_id

ORDER BY total\_pizzas DESC

LIMIT 5;



----> Outliers are present in order\_id and total\_pizzas features

#### **Basic:**

1. **Retrieve the total number of orders placed.  
     
   Objective:** Understand the total volume of orders.

SELECT COUNT(DISTINCT(order\_id)) AS count\_orders

FROM orders;

SELECT COUNT(DISTINCT(order\_id)) AS count\_orders

FROM order\_details;

#### 

#### **---> Total number of orders placed is 21350**

1. **Calculate the total revenue generated from pizza sales.  
     
   Objective:** Calculate the total revenue generated from all pizza orders.

SELECT SUM(od.quantity \* p.price) AS Total\_Revenue

FROM order\_details AS od

JOIN pizzas AS p

ON od.pizza\_id = p.pizza\_id;

SELECT ROUND(SUM(od.quantity \* p.price), 3) AS Total\_Revenue

FROM order\_details AS od

JOIN pizzas AS p

ON od.pizza\_id = p.pizza\_id;

SELECT CEIL(SUM(od.quantity \* p.price)) AS Total\_Revenue

FROM order\_details AS od

JOIN pizzas AS p

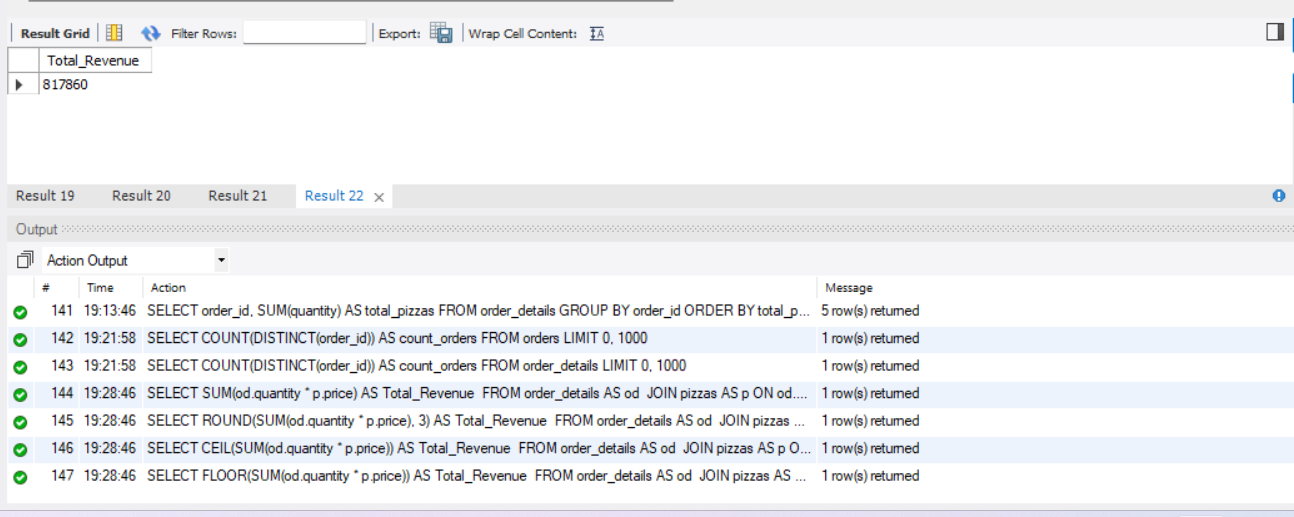
ON od.pizza\_id = p.pizza\_id;

SELECT FLOOR(SUM(od.quantity \* p.price)) AS Total\_Revenue

FROM order\_details AS od

JOIN pizzas AS p

ON od.pizza\_id = p.pizza\_id;



**--->Total revenue generated from pizza sales is 817860**

1. **Identify the highest-priced pizza.  
     
   Objective:** Find out which pizza is the most expensive

SELECT \* FROM pizza\_types;

SELECT \* FROM pizzas;

SELECT \* FROM order\_details;

SELECT \* FROM orders;

SELECT pt.name, p.price

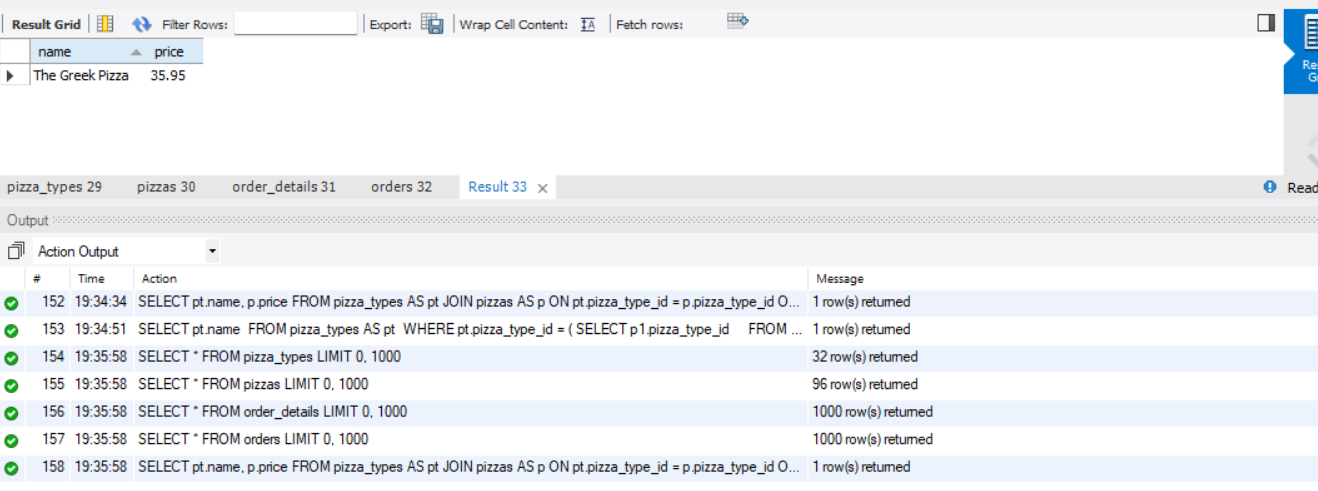
FROM pizza\_types AS pt

JOIN pizzas AS p

ON pt.pizza\_type\_id = p.pizza\_type\_id

ORDER BY p.price DESC

LIMIT 1;



--->**highest-priced pizza is 'The Greek Pizza' with price of '35.95'**

1. **Identify the most common pizza size ordered.  
     
   Objective:** Determine which pizza size (e.g., small, medium, large) is ordered the most.

SELECT size, COUNT(\*) AS Number\_of\_Orders

FROM (

SELECT

SUBSTRING\_INDEX(pizza\_id, '\_', -1) AS size

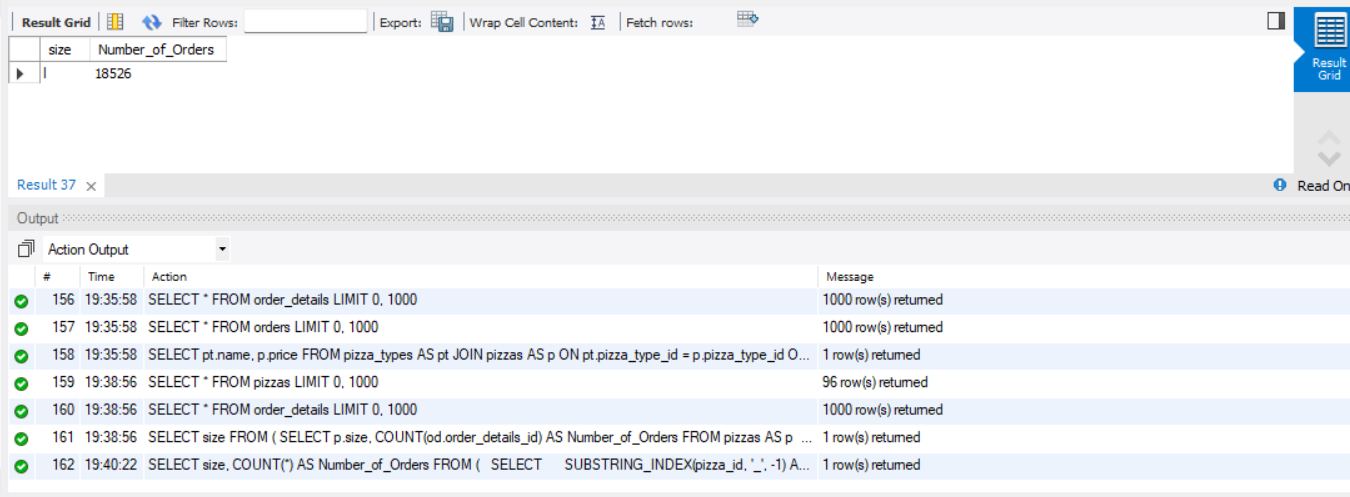
FROM order\_details

) AS t

GROUP BY size

ORDER BY Number\_of\_Orders DESC

LIMIT 1;



--->**Most common pizza size ordered is L**

1. **List the top 5 most ordered pizza types along with their quantities.  
     
   Objective:** Find out which pizza types are most frequently ordered.

SELECT p.pizza\_type\_id, SUM(od.quantity) AS Total\_Quantity

FROM pizzas AS p

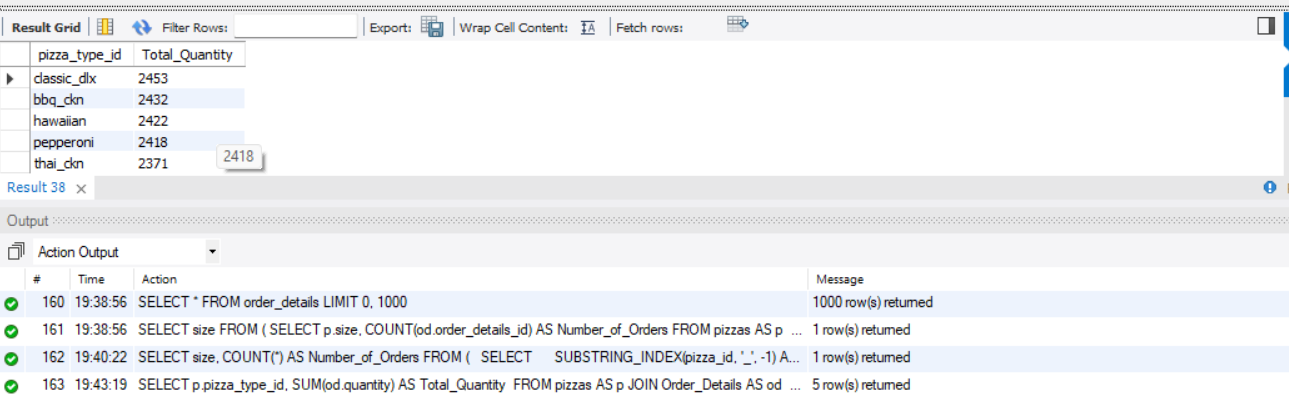
JOIN Order\_Details AS od

ON od.pizza\_id = p.pizza\_id

GROUP BY p.pizza\_type\_id

ORDER BY Total\_Quantity DESC

LIMIT 5;



**---> Top 5 most ordered pizzas are:**

**pizza\_type\_id Total\_Quantity**

**classic\_dlx 2453**

**bbq\_ckn 2432**

**hawaiian 2422**

**pepperoni 2418**

**thai\_ckn 2371**

**Intermediate:**

1. **Join the necessary tables to find the total quantity of each pizza category ordered.  
     
   Objective:** Explore the relationship between pizza categories and quantities ordered.

SELECT pt.category, SUM(od.quantity) AS Total\_Quantity

FROM pizza\_types AS pt

JOIN pizzas AS p

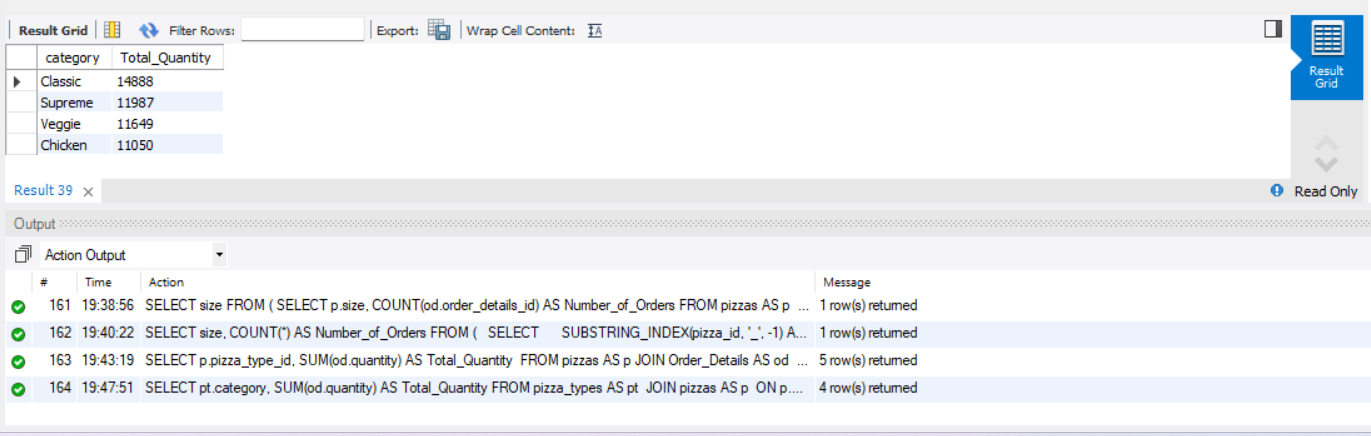
ON p.pizza\_type\_id = pt.pizza\_type\_id

JOIN Order\_Details AS od

ON od.pizza\_id = p.pizza\_id

GROUP BY pt.category

ORDER BY Total\_Quantity DESC;



# category Total\_Quantity

Classic 14888

Supreme 11987

Veggie 11649

Chicken 11050

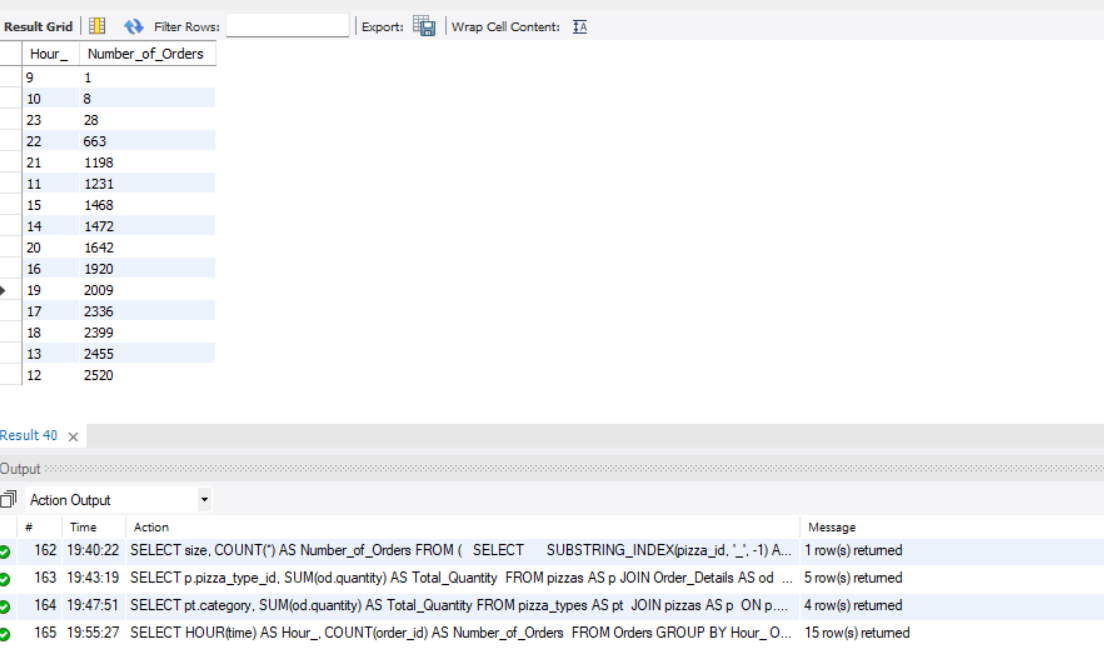
1. **Determine the distribution of orders by hour of the day.  
     
   Objective:** Analyze how orders are distributed across different times of day.

SELECT HOUR(time) AS Hour\_, COUNT(order\_id) AS Number\_of\_Orders

FROM Orders

GROUP BY Hour\_

ORDER BY Hour\_;



**# Hour Number\_of\_Orders**

**9 1**

**10 8**

**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**

1. **Join relevant tables to find the category-wise distribution of pizzas.  
     
   Objective:** Find out how pizzas from different categories are ordered.

SELECT \* FROM order\_details;

SELECT \* FROM pizza\_types;

SELECT \* FROM pizzas;

SELECT pt.category, COUNT(od.order\_id) AS Number\_of\_Orders

FROM pizza\_types AS pt

JOIN pizzas AS p

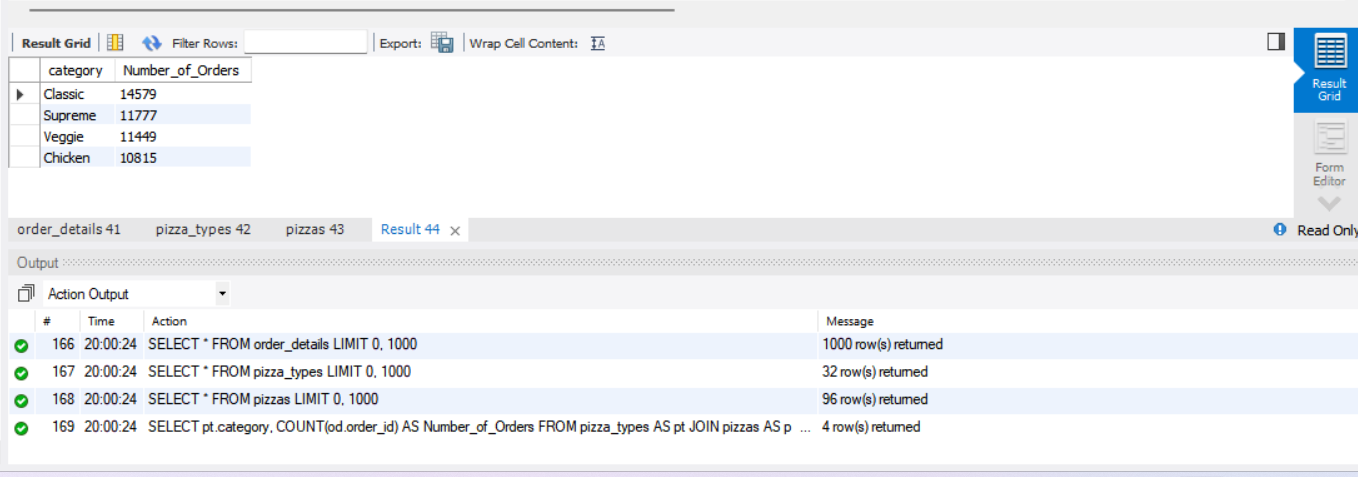
ON pt.pizza\_type\_id = p.pizza\_type\_id

JOIN order\_details AS od

ON od.pizza\_id = p.pizza\_id

GROUP BY pt.category

ORDER BY Number\_of\_Orders DESC;



**---->Category-wise distribution of pizzas**

**# category Number\_of\_Orders**

**Classic 14579**

**Supreme 11777**

**Veggie 11449**

**Chicken 10815**

1. **Group the orders by date and calculate the average number of pizzas ordered per day.  
     
   Objective:** Analyze daily order trends and average quantities.

SELECT FLOOR(AVG(Quantity)) AS Avg\_Pizzas\_Ordered\_Per\_Day

FROM (

SELECT o.date, SUM(od.quantity) AS Quantity

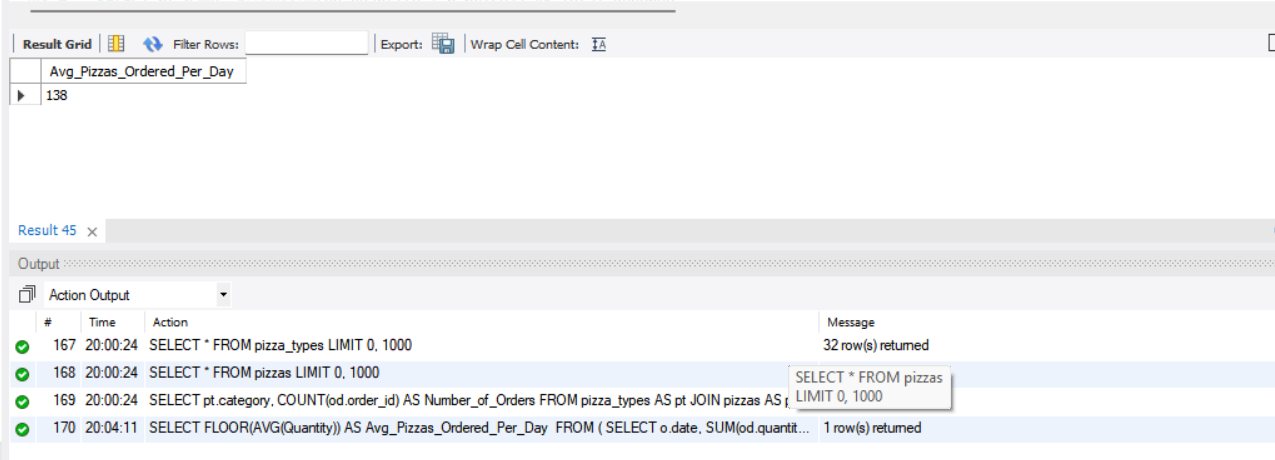
FROM Orders AS o

JOIN Order\_Details AS od

ON o.order\_id = od.order\_id

GROUP BY o.date

) AS a;



**---> Average number of Pizzas Ordered per day are 138**

1. **Determine the top 3 most ordered pizza types based on revenue.  
     
   Objective:** Identify the pizza types that generated the most revenue.

SELECT pt.name, FLOOR(SUM(od.quantity \* p.price)) AS Total\_Revenue

FROM pizza\_types AS pt

JOIN pizzas AS p

ON p.pizza\_type\_id = pt.pizza\_type\_id

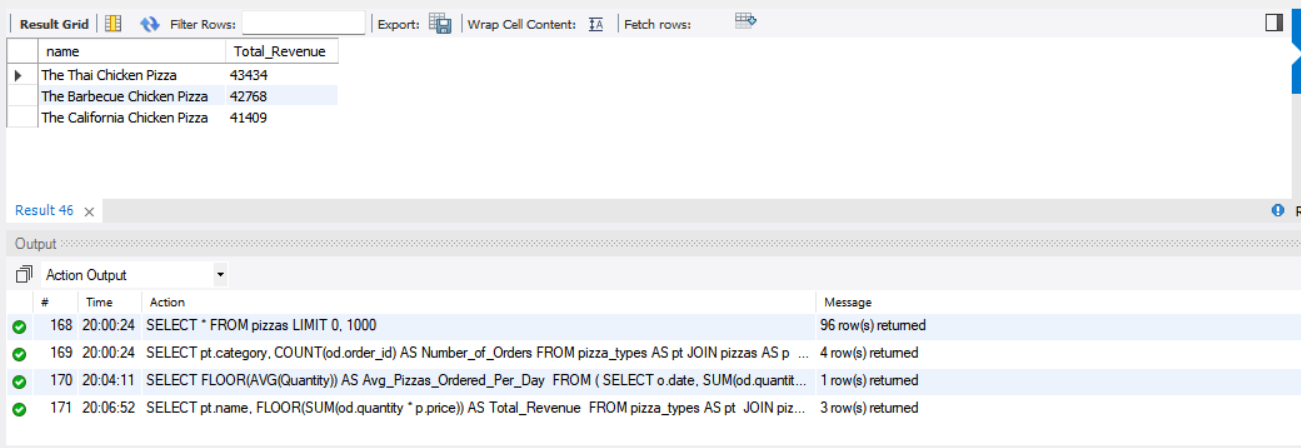
JOIN order\_details AS od

ON od.pizza\_id = p.pizza\_id

GROUP BY pt.name

ORDER BY Total\_Revenue DESC

LIMIT 3;



**--->Top 3 most ordered pizza types based on revenue**

**name Total\_Revenue**

**The Thai Chicken Pizza 43434**

**The Barbecue Chicken Pizza 42768**

**The California Chicken Pizza 41409**

**Advanced:**

1. **Calculate the percentage contribution of each pizza type to total revenue.  
     
   Objective:** Understand each pizza's contribution to overall sales.

SELECT pt.category,

ROUND((SUM(od1.quantity \* p1.price) / (

SELECT SUM(od2.quantity \* p2.price)

FROM order\_details AS od2

JOIN pizzas AS p2

ON od2.pizza\_id = p2.pizza\_id)) \* 100, 2) AS Proportion

FROM order\_details AS od1

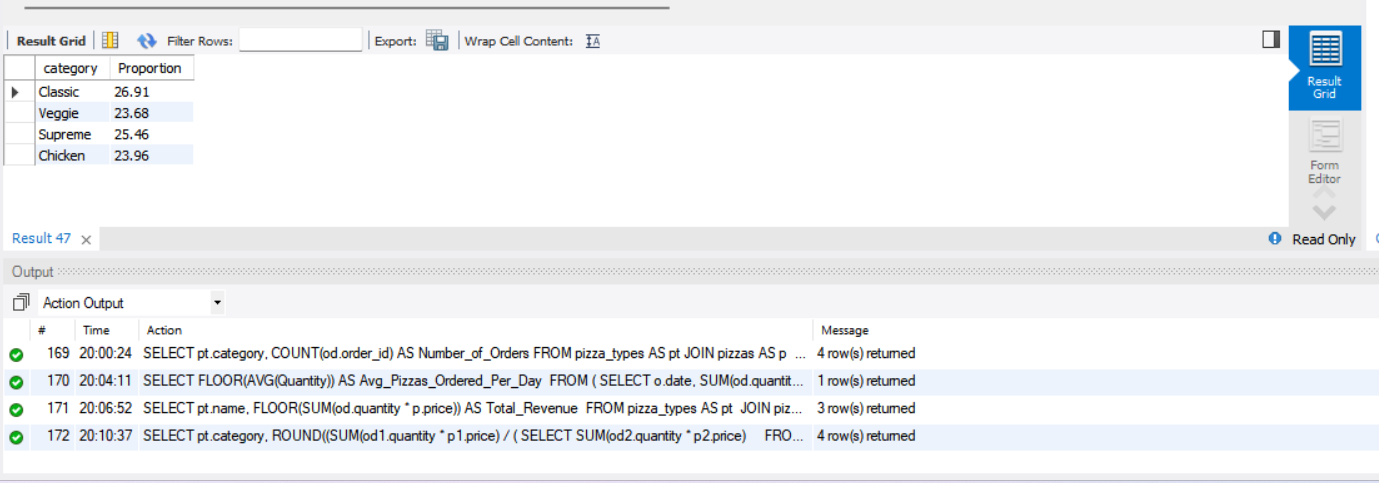
JOIN pizzas AS p1

ON od1.pizza\_id = p1.pizza\_id

JOIN pizza\_types AS pt

ON p1.pizza\_type\_id = pt.pizza\_type\_id

GROUP BY pt.category;



**---> Percentage contribution of each pizza type to total revenue**

**category Proportion**

**Classic 26.91**

**Veggie 23.68**

**Supreme 25.46**

**Chicken 23.96**

1. **Analyze the cumulative revenue generated over time.  
     
   Objective:** Track how revenue accumulates over time.

SELECT date, FLOOR(Revenue) AS Revenue,

FLOOR(SUM(Revenue) OVER (ORDER BY date)) AS Cum\_Revenue

FROM (

SELECT o.date,

SUM(od.quantity \* p.price) AS Revenue

FROM Order\_Details AS od

JOIN Pizzas AS p

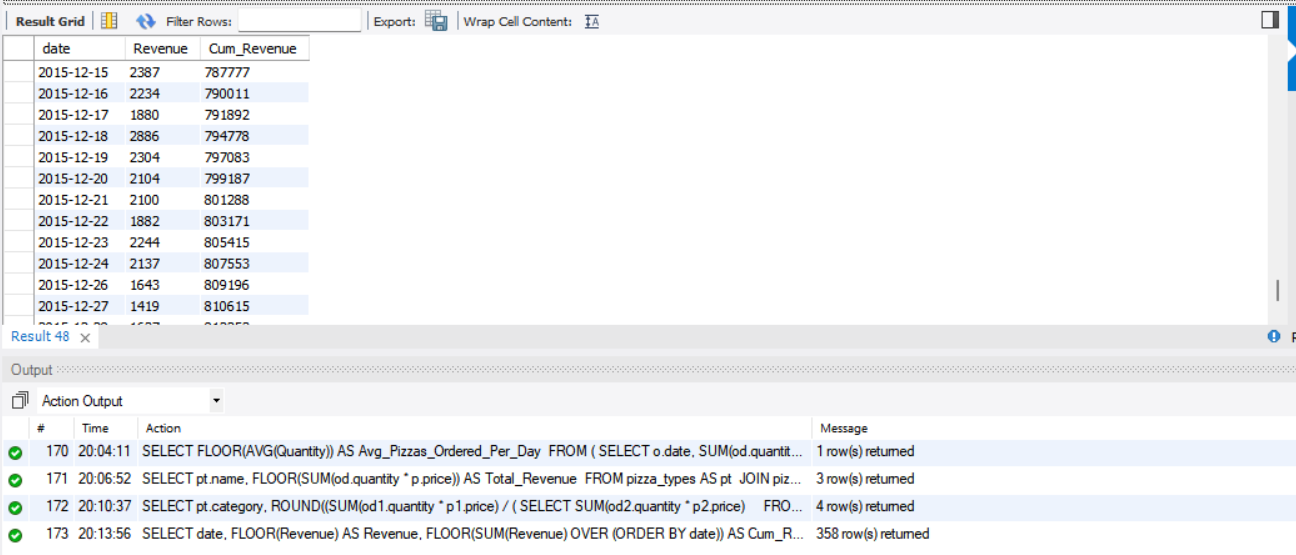
ON p.pizza\_id = od.pizza\_id

JOIN Orders AS o

ON o.order\_id = od.order\_id

GROUP BY o.date

) AS s;



1. **Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
     
   Objective:** Find the highest-grossing pizzas within each category.

SELECT category, name, Revenue

FROM (

SELECT category, name, revenue,

RANK() OVER (PARTITION BY category ORDER BY Revenue DESC) AS rank\_

FROM (

SELECT pt.category, pt.name,

SUM(od.quantity \* p.price) AS Revenue

FROM pizza\_types AS pt

JOIN pizzas AS p

ON pt.pizza\_type\_id = p.pizza\_type\_id

JOIN order\_details AS od

ON od.pizza\_id = p.pizza\_id

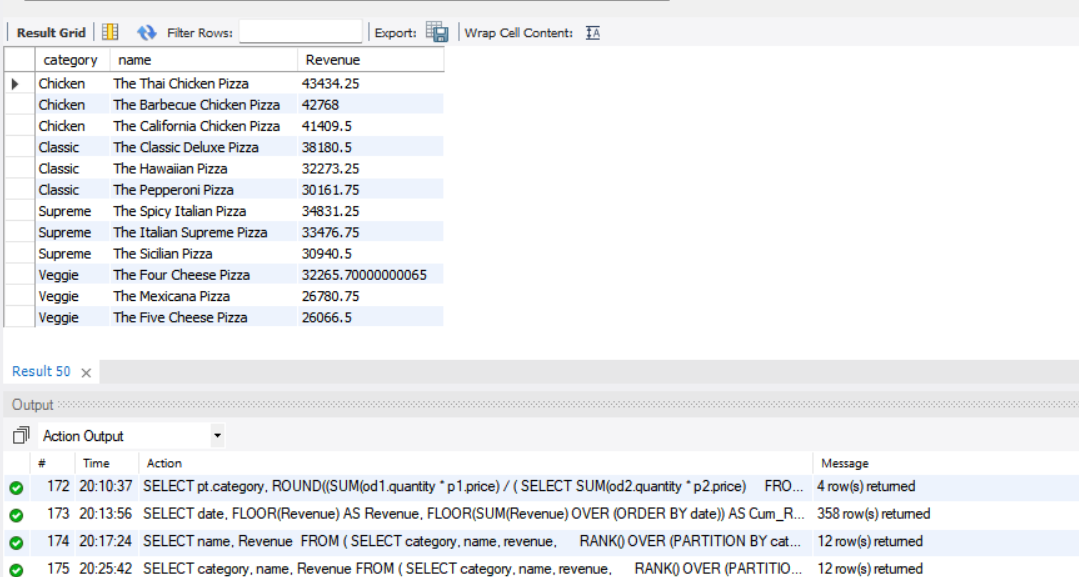
GROUP BY pt.category, pt.name

) AS a

) AS b

WHERE rank\_ <= 3

ORDER BY category, Revenue DESC;



| **Category** | **Name** | **Revenue ($)** |
| --- | --- | --- |
| **Chicken** | The Thai Chicken Pizza | 43,434.25 |
| **Chicken** | The Barbecue Chicken Pizza | 42,768.00 |
| **Chicken** | The California Chicken Pizza | 41,409.50 |
| **Classic** | The Classic Deluxe Pizza | 38,180.50 |
| **Classic** | The Hawaiian Pizza | 32,273.25 |
| **Classic** | The Pepperoni Pizza | 30,161.75 |
| **Supreme** | The Spicy Italian Pizza | 34,831.25 |
| **Supreme** | The Italian Supreme Pizza | 33,476.75 |
| **Supreme** | The Sicilian Pizza | 30,940.50 |
| **Veggie** | The Four Cheese Pizza | 32,265.70 |
| **Veggie** | The Mexicana Pizza | 26,780.75 |
| **Veggie** | The Five Cheese Pizza | 26066.5 |