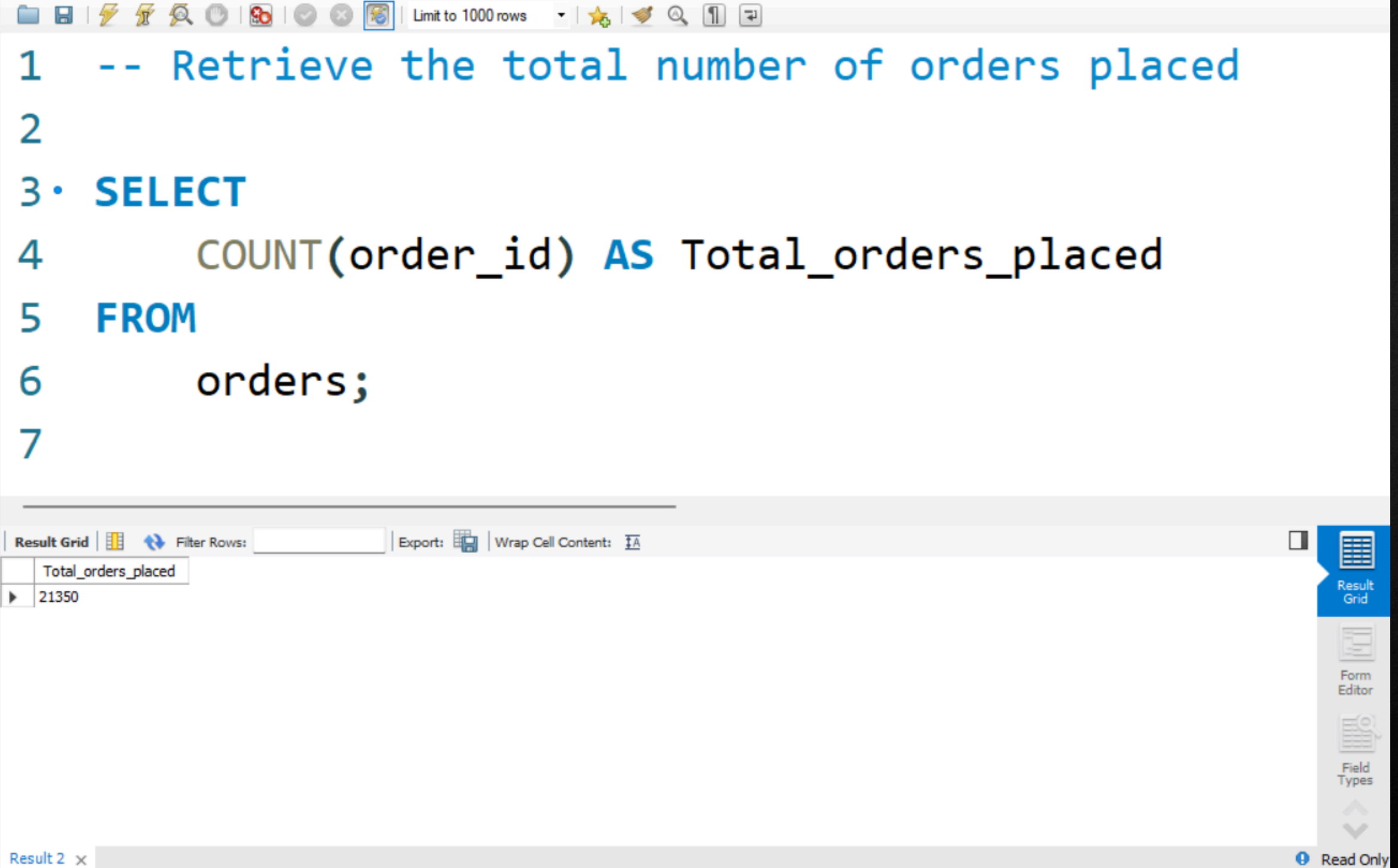
PIZZA SALES ANALYSIS

This analysis explores key insights into pizza sales trends, customer preferences, and revenue distribution, answering critical business questions.


Limit to 1000 rows
1 -- Retrieve the total number of orders placed
2
3 SELECT
4 COUNT(order_id) AS Total_orders_placed
5 FROM
6 orders;
7
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Total_orders_placed
21350
Result Grid
Form Editor
Field Types
Result 2
Read Only

```sql
-- Calculate the total revenue generated from pizza sales

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

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

Total_Revenue_Generated
817860.05

Result Grid | Form Editor | Field Types

Result 4 | Read Only

```sql
-- Identify the highest-priced pizza

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

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

name	price
The Greek Pizza	35.95

```sql
-- List the top 5 most ordered pizza types along with their quantities

SELECT
    pt.name, SUM(od.quantity) AS total_order_pizza
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_order_pizza DESC
LIMIT 5
```

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

name	total_order_pizza
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Result Grid | Form Editor | Field Types

Result 2 | Read Only

```sql
-- Identify the most common pizza size ordered.

SELECT
    p.size, COUNT(od.order_details_id) AS Most_common_pizza_size
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY size
ORDER BY Most_common_pizza_size DESC
LIMIT 1
```

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

size	Most_common_pizza_size
L	18526

Result Grid | Form Editor | Field Types

Result 4 | Read Only

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered

SELECT
    pt.category,
    SUM(od.quantity) AS total_pizaa_category_ordered
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY category
ORDER BY total_pizaa_category_ordered DESC
```

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

category	total_pizaa_category_ordered
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Result Grid | Form Editor | Field Types

Result 1 | Read Only

```sql
-- Determine the distribution of orders
-- by hour of the day

SELECT
    HOUR(time) AS Hour, COUNT(order_id) AS Total_orde
FROM
    orders
GROUP BY Hour
```

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

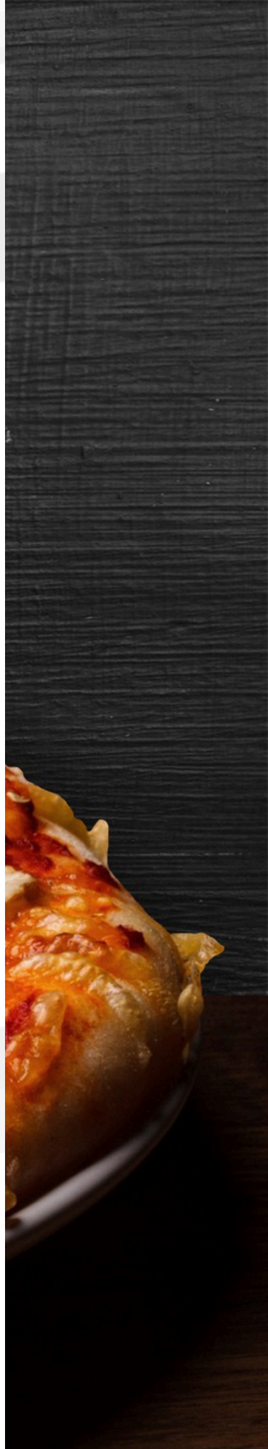
Hour	Total_order_per_hour
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198

Result 1 | Read Only

```sql
-- Join relevant tables to find the category-wise
-- distribution of pizzas

SELECT
    category, COUNT(name) AS total_pizza_category
FROM
    pizza_types
GROUP BY category
ORDER BY total pizza category DESC
```

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

category	total_pizza_category
Supreme	9
Veggie	9
Classic	8
Chicken	6



SQL File 2 | SQL File 3 | SQL File 4 | SQL File 5 | SQL File 6 | SQL File 7 | SQL File 8 | SQL File 9 | SQL File 10 | SQL File 11 | SQL File 12

Limit to 1000 rows

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

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

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

round(avg(total_order_per_day),0)
138

Result Grid

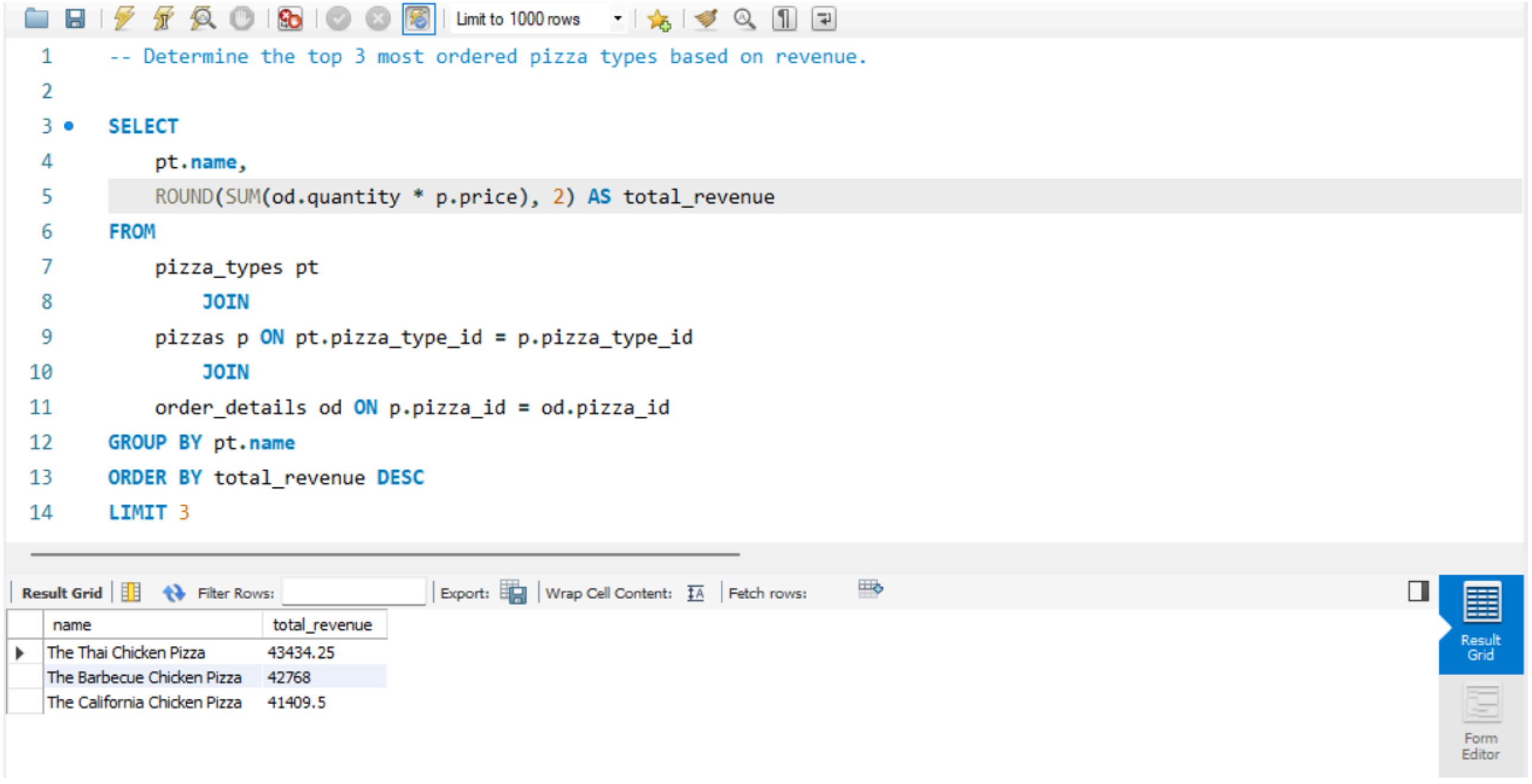
```sql
-- Determine the top 3 most ordered pizza types based on revenue.

SELECT
    pt.name,
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3
```

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

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

Result Grid

Form Editor


Limit to 1000 rows
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
pt.name,
ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM
pizza_types pt
JOIN
pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN
order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
name
total_revenue
The Thai Chicken Pizza
43434.25
The Barbecue Chicken Pizza
42768
The California Chicken Pizza
41409.5
Result Grid
Form Editor

```sql
-- Calculate the percentage contribution of each pizza type to total revenue

SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
            ROUND(SUM(od_sub.quantity * p_sub.price), 2)
        FROM
            order_details od_sub
                JOIN
            pizzas p_sub ON od_sub.pizza_id = p_sub.pizza_id) * 100, 2) AS total_revenue_pct
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY total_revenue_pct DESC
LIMIT 1000;
```

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

category	total_revenue_pct
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

```sql
-- Analyze the cumulative revenue generated over time

select date,round(sum(revenue) over(order by date),2)
as cumulative_revenue
from
(select o.date, sum(od.quantity*p.price) as revenue from order_details od
join
pizzas p on od.pizza_id=p.pizza_id
join orders o on
o.order_id=od.order_id
group by o.date) as total_revenue
```

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

date	cumulative_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6

Result Grid

Form


```sql
-- Determine the top 3 most ordered 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) a
from
(select pt.category,pt.name,sum(od.quantity*p.price) as revenue from pizza_types pt
join pizzas p on pt.pizza_type_id=p.pizza_type_id
join order_details od on
od.pizza_id=p.pizza_id
group by pt.category,pt.name) as rev_by_cat) as rnk
where rn<=3
```


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

name	revenue
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065