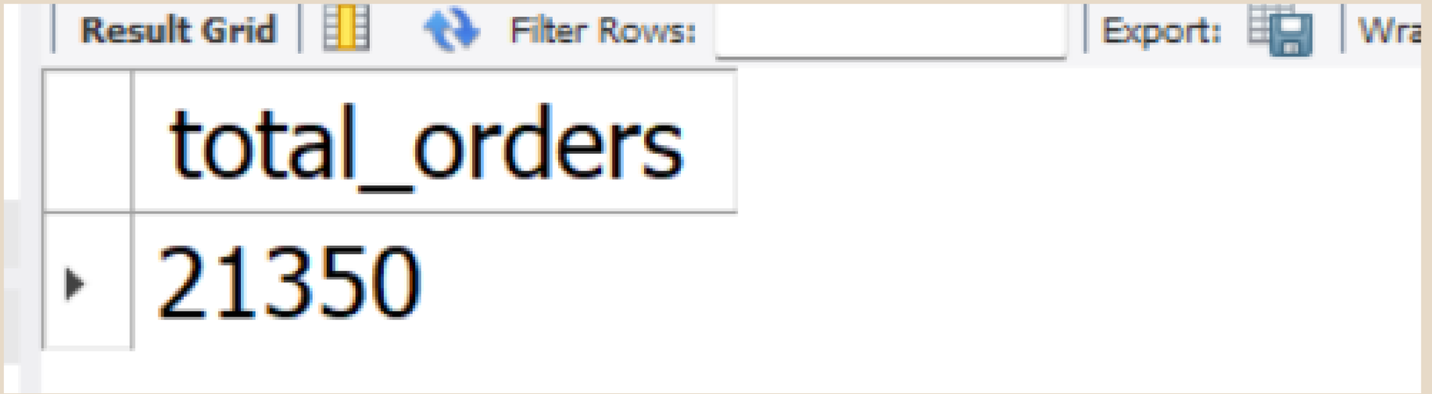
Pizza Sales Analysis

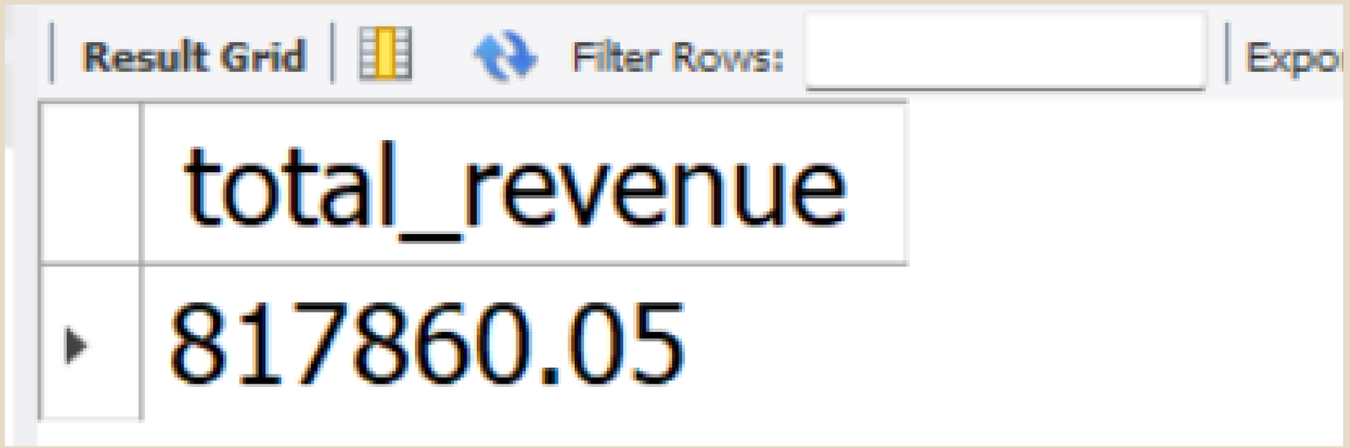
1.Retrieve the total number of orders placed.

```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_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.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 2 DESC
LIMIT 1;
```

name	price
The Greek Pizza	35.95

4.Identify the most common pizza size ordered

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS total_orders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

size	total_orders
L	18526

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

```sql
SELECT
    pizza_types.name,
    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 1 , 2
ORDER BY 3 DESC
```

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

name	category	Quantity
The Classic Delux...	Classic	2453
The Barbecue Chi...	Chicken	2432
The Hawaiian Pizza	Classic	2422
The Pepperoni Pi...	Classic	2418
The Thai Chicken...	Chicken	2371

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 1
ORDER BY 2 DESC;
```

Result Grid | Filter Rows:

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

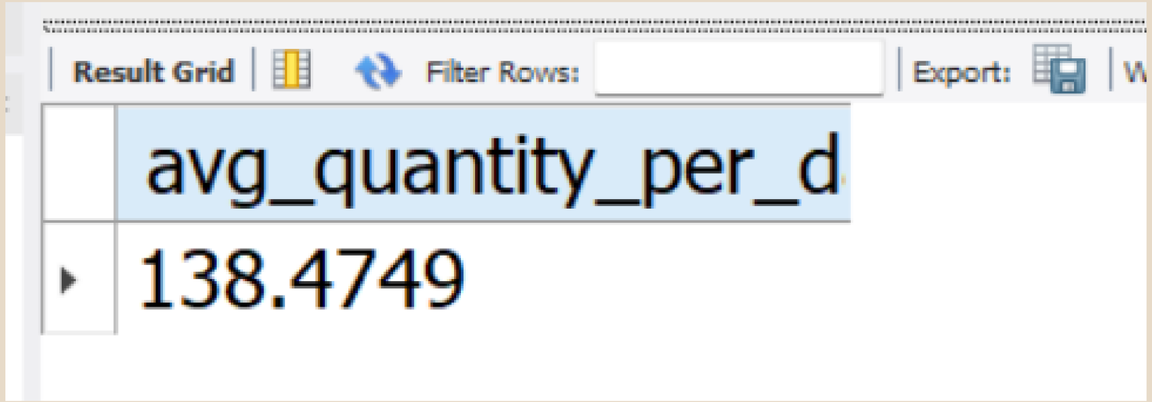
```sql
SELECT
    HOUR(order_time), COUNT(order_id) AS total_orders
FROM
    orders
GROUP BY 1;
```

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

hour(order_time)	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

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

```sql
SELECT
    AVG(total_quantity) AS avg_quantity_per_day
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS total_quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY 1) AS quantity
```



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

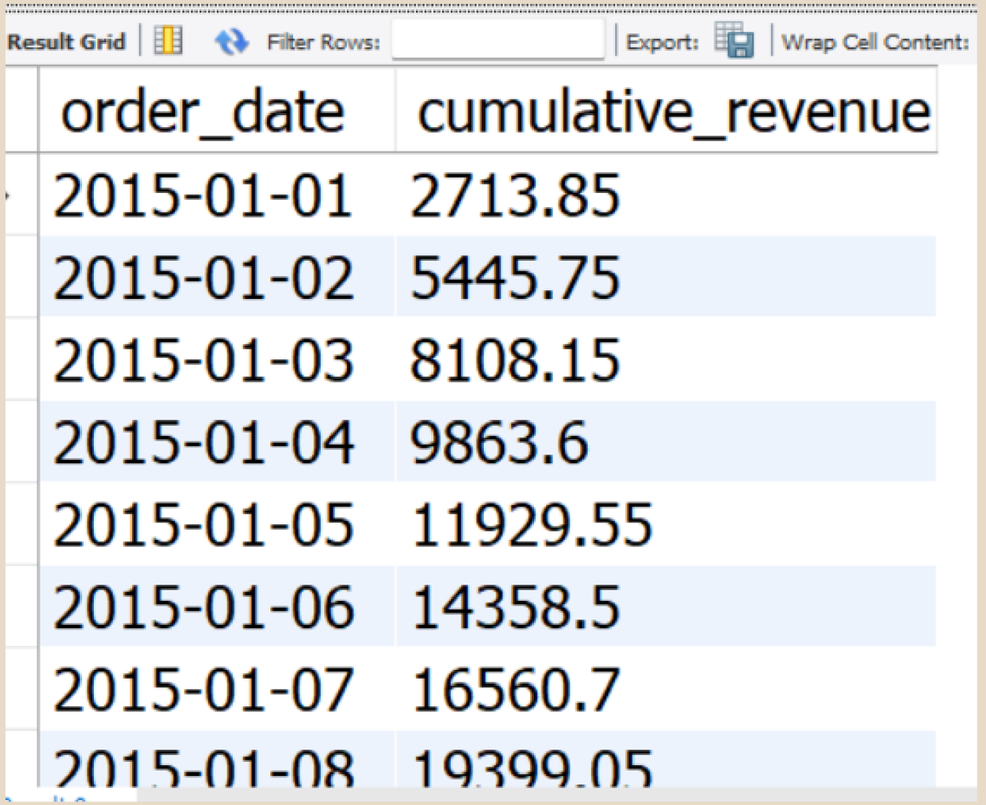
```sql
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3;
```

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

name	total_revenue
The Thai Chicken...	43434.25
The Barbecue Chi...	42768
The California Chi...	41409.5

10. Analyze the cumulative revenue generated over time.

```sql
Select order_date,
round(sum(total_revenue) over(order by order_date),2)
 as cumulative_revenue from
(SELECT
    orders.order_date,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    orders ON orders.order_id = order_details.order_id
GROUP BY 1) as sales;
```



order_date	cumulative_revenue
2015-01-01	2713.85
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

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

```sql
Select name, category, total_revenue, rn
from
(select name,category, total_revenue,
rank()over(partition by category order by total_revenue desc) as rn
from
(SELECT
    pizza_types.name, pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_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 1,2) as a) as b
where rn <= 3;
```

name	category	total_revenue	rn
The Thai Chicken...	Chicken	43434.25	1
The Barbecue Chi...	Chicken	42768	2
The California Chi...	Chicken	41409.5	3
The Classic Delux...	Classic	38180.5	1
The Hawaiian Pizza	Classic	32273.25	2
The Pepperoni Pi...	Classic	30161.75	3
The Spicy Italian ...	Supreme	34831.25	1
The Italian Supre...	Supreme	33476.75	2
The Sicilian Pizza	Supreme	30940.5	3
The Four Cheese ...	Veggie	32265.7	1
The Mexicana Pizza	Veggie	26780.75	2
The Five Cheese ...	Veggie	26066.5	3