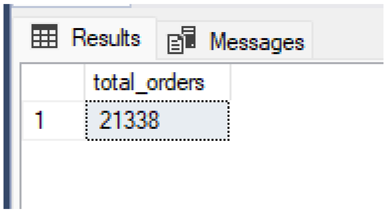
# Pizza Sales

## 1. Retrieve the total number of orders placed.

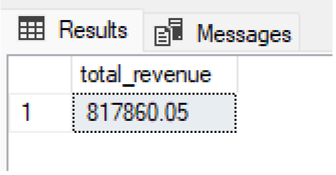
```
select COUNT(distinct order_id) as total_orders from orders;
```



|   | total_orders |
|---|---|
| 1 | 21338 |

## 2. Calculate the total revenue by pizza sales.

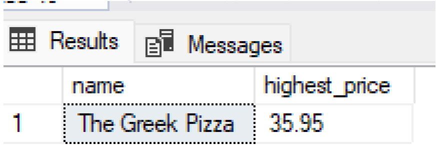
```
 select
   round(Sum(od.quantity * ps.price),2) as total_revenue
from order_details od
left join pizzas ps
on od.pizza_id=ps.pizza_id;
```



|   | total_revenue |
|---|---|
| 1 | 817860.05 |

## 3.Identify the highest price pizza.

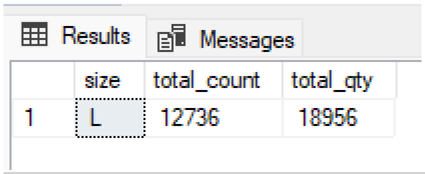
```
select top 1
   name, MAX(ps.price) as highest_price
from pizza_types pt
left join pizzas ps
on pt.pizza_type_id=ps.pizza_type_id
group by name
order by max(ps.price) desc;
```



|   | name | highest_price |
|---|---|---|
| 1 | The Greek Pizza | 35.95 |

## 4. Identify the most common pizza  size ordered.

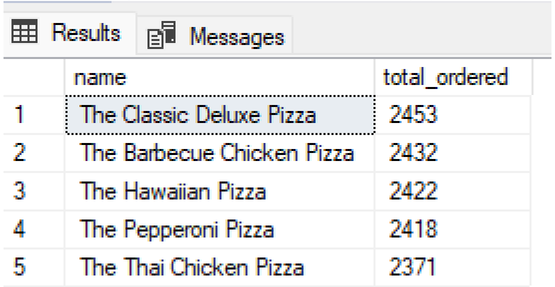
```
 select top 1 size, COUNT(distinct order_id) as total_count,
sum(od.quantity) as total_qty from order_details od
left join pizzas ps
on od.pizza_id=ps.pizza_id
group by size
order by COUNT(order_id) desc;
```



| | size | total_count | total_qty |
|---|---|---|---|
| 1 | L | 12736 | 18956 |

### 5.List the top 5 pizza types along with their qty.

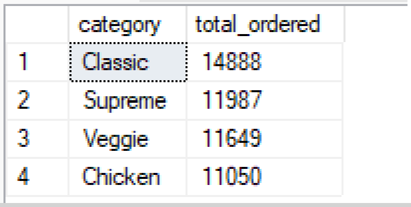
```
select top 5 pt.name, sum(od.quantity) as total_ordered from
order_details od
  join pizzas ps
on od.pizza_id=ps.pizza_id
  join pizza_types pt
on ps.pizza_type_id=pt.pizza_type_id
group by pt.name
order by sum(od.quantity) desc;
```



| | name | total_ordered |
|---|---|---|
| 1 | The Classic Deluxe Pizza | 2453 |
| 2 | The Barbecue Chicken Pizza | 2432 |
| 3 | The Hawaiian Pizza | 2422 |
| 4 | The Pepperoni Pizza | 2418 |
| 5 | The Thai Chicken Pizza | 2371 |

### 6.Find the total quantity of each pizza category ordered

```
select  pt.category, sum(od.quantity) as total_ordered from
order_details od
  join pizzas ps
on od.pizza_id=ps.pizza_id
  join pizza_types pt
on ps.pizza_type_id=pt.pizza_type_id
group by pt.category
order by sum(od.quantity) desc;
```

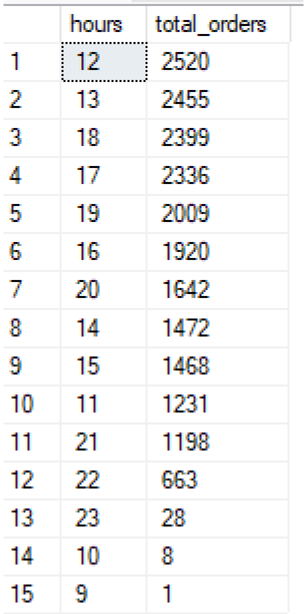


| | category | total_ordered |
|---|---|---|
| 1 | Classic | 14888 |
| 2 | Supreme | 11987 |
| 3 | Veggie | 11649 |
| 4 | Chicken | 11050 |

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

```
select DATEPART(hour,TIME) as hours, COUNT(distinct order_id) as
total_orders
from orders
group by DATEPART(hour,TIME)
```

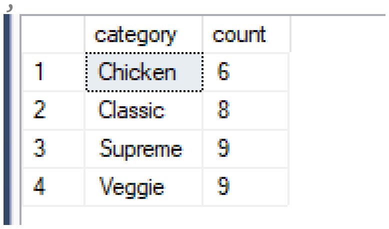
```
order by COUNT(distinct order_id) desc;
```



|  | hours | total_orders |
|---|---|---|
| 1 | 12 | 2520 |
| 2 | 13 | 2455 |
| 3 | 18 | 2399 |
| 4 | 17 | 2336 |
| 5 | 19 | 2009 |
| 6 | 16 | 1920 |
| 7 | 20 | 1642 |
| 8 | 14 | 1472 |
| 9 | 15 | 1468 |
| 10 | 11 | 1231 |
| 11 | 21 | 1198 |
| 12 | 22 | 663 |
| 13 | 23 | 28 |
| 14 | 10 | 8 |
| 15 | 9 | 1 |

8.Find the category wise distribution of pizza

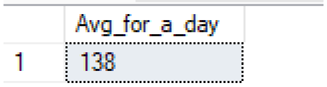
```
  select category , COUNT(distinct pizza_type_id) as count
from pizza_types
group by category
;
```



|  | category | count |
|---|---|---|
| 1 | Chicken | 6 |
| 2 | Classic | 8 |
| 3 | Supreme | 9 |
| 4 | Veggie | 9 |

9.Group the orders by date and calculate the avg numb of pizza ordered per day.

```
 with cte as (select o.date , sum(od.quantity)as orders from orders o
join order_details od
on o.order_id=od.order_id
group by o.date)
 select AVG(orders) as Avg_for_a_day from cte
```

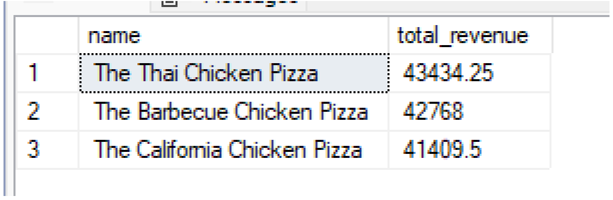


|  | Avg_for_a_day |
|---|---|
| 1 | 138 |

10.Determine the top 3 most ordered pizza by revenue

```
 select top 3 pt.name,sum(od.quantity*ps.price) as total_revenue from
order_details od
join pizzas ps
on od.pizza_id= ps.pizza_id
join pizza_types pt
on ps.pizza_type_id=pt.pizza_type_id
group by pt.name
```

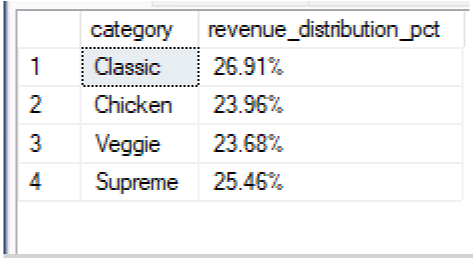
```
order by sum(od.quantity*ps.price) desc;
```



|  | name | total_revenue |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43434.25 |
| 2 | The Barbecue Chicken Pizza | 42768 |
| 3 | The California Chicken Pizza | 41409.5 |

11.Calculate the % contribution of each pizza to total revenue

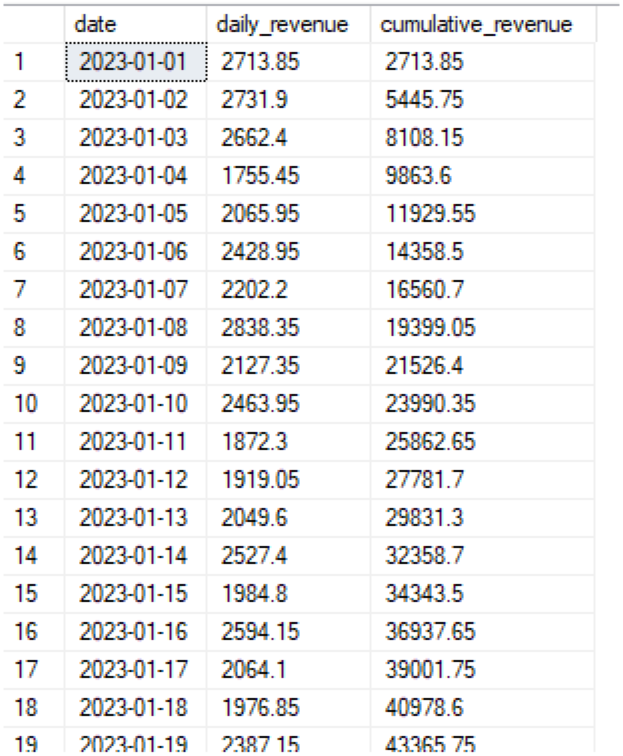
```
 SELECT
    pt.category,
    CONCAT(CAST(
        (SUM(od.quantity * ps.price) /
        (SELECT SUM(od2.quantity * ps2.price)
         FROM order_details od2
         JOIN pizzas ps2 ON od2.pizza_id = ps2.pizza_id)) * 100
        AS DECIMAL(10,2)), '%') AS revenue_distribution_pct
FROM
    order_details od
JOIN
    pizzas ps ON od.pizza_id = ps.pizza_id
JOIN
    pizza_types pt ON ps.pizza_type_id = pt.pizza_type_id
GROUP BY
    pt.category;
```



|  | category | revenue_distribution_pct |
|---|---|---|
| 1 | Classic | 26.91% |
| 2 | Chicken | 23.96% |
| 3 | Veggie | 23.68% |
| 4 | Supreme | 25.46% |

12.Analyze the cumulative revenue over time

```
 SELECT
    o.date,
    SUM(od.quantity * ps.price) AS daily_revenue,
    SUM(SUM(od.quantity * ps.price)) OVER (ORDER BY o.date) AS
cumulative_revenue
FROM
    order_details od
join orders o on od.order_id=o.order_id
JOIN
    pizzas ps ON od.pizza_id = ps.pizza_id
GROUP BY
    o.date
ORDER BY
    o.date;
```

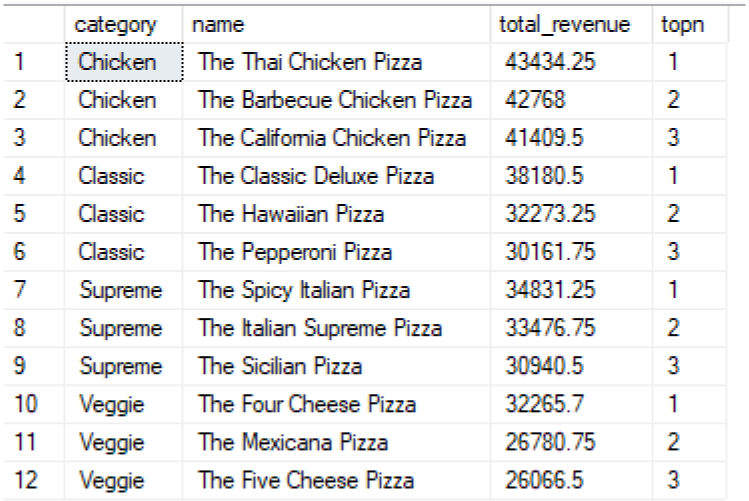


| | date | daily_revenue | cumulative_revenue |
|---|---|---|---|
| 1 | 2023-01-01 | 2713.85 | 2713.85 |
| 2 | 2023-01-02 | 2731.9 | 5445.75 |
| 3 | 2023-01-03 | 2662.4 | 8108.15 |
| 4 | 2023-01-04 | 1755.45 | 9863.6 |
| 5 | 2023-01-05 | 2065.95 | 11929.55 |
| 6 | 2023-01-06 | 2428.95 | 14358.5 |
| 7 | 2023-01-07 | 2202.2 | 16560.7 |
| 8 | 2023-01-08 | 2838.35 | 19399.05 |
| 9 | 2023-01-09 | 2127.35 | 21526.4 |
| 10 | 2023-01-10 | 2463.95 | 23990.35 |
| 11 | 2023-01-11 | 1872.3 | 25862.65 |
| 12 | 2023-01-12 | 1919.05 | 27781.7 |
| 13 | 2023-01-13 | 2049.6 | 29831.3 |
| 14 | 2023-01-14 | 2527.4 | 32358.7 |
| 15 | 2023-01-15 | 1984.8 | 34343.5 |
| 16 | 2023-01-16 | 2594.15 | 36937.65 |
| 17 | 2023-01-17 | 2064.1 | 39001.75 |
| 18 | 2023-01-18 | 1976.85 | 40978.6 |
| 19 | 2023-01-19 | 2387.15 | 43365.75 |

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

```
with cte as
( select  name ,category, sum(od.quantity*ps.price) as total_revenue,
ROW_NUMBER() over(partition by category order by
sum(od.quantity*ps.price) desc) as rn
from order_details od
join pizzas ps
on od.pizza_id= ps.pizza_id
join pizza_types pt
on ps.pizza_type_id=pt.pizza_type_id
group by pt.name, category)

select category, name, round(total_revenue,2) as total_revenue, rn as
topn
from cte
where rn in (1,2,3)
```



| | category | name | total_revenue | topn |
|---|---|---|---|---|
| 1 | Chicken | The Thai Chicken Pizza | 43434.25 | 1 |
| 2 | Chicken | The Barbecue Chicken Pizza | 42768 | 2 |
| 3 | Chicken | The California Chicken Pizza | 41409.5 | 3 |
| 4 | Classic | The Classic Deluxe Pizza | 38180.5 | 1 |
| 5 | Classic | The Hawaiian Pizza | 32273.25 | 2 |
| 6 | Classic | The Pepperoni Pizza | 30161.75 | 3 |
| 7 | Supreme | The Spicy Italian Pizza | 34831.25 | 1 |
| 8 | Supreme | The Italian Supreme Pizza | 33476.75 | 2 |
| 9 | Supreme | The Sicilian Pizza | 30940.5 | 3 |
| 10 | Veggie | The Four Cheese Pizza | 32265.7 | 1 |
| 11 | Veggie | The Mexicana Pizza | 26780.75 | 2 |
| 12 | Veggie | The Five Cheese Pizza | 26066.5 | 3 |