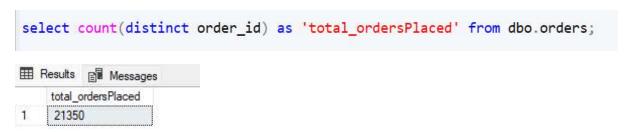
## Mario's Pizza

**Summary:** Mario's Pizza aims to expand its offerings by adding new pizza varieties and categories. They seek insights from their sales data, provided in a database with four tables: orders, order\_details, pizzas, and pizza\_type. The analysis will address key business questions to guide their scaling strategy.

1. Retrieve the total number of orders placed.



2. Calculate the total revenue generated from pizza sales.

```
| select round(sum(det.quantity * piz.price),2) as 'total_revenue_from_pizzaSales' | from dbo.order_details as det | join dbo.pizzas as piz | on det.pizza_id = piz.pizza_id; | total_revenue_from_pizzaSales | 1 | 817860.05
```

3. Identify the highest-priced pizza.

```
select top 1
typ.name,
piz.price
from dbo.pizza_types as typ
join dbo.pizzas as piz
on typ.pizza_type_id = piz.pizza_type_id
order by piz.price desc;
```

```
name price
1 The Greek Pizza 35.9500007629395
```

4. Identify the most common pizza size ordered.

```
select top 1
piz.size, count(det.order_id) as 'most_common size ordered'
from dbo.pizzas as piz
join dbo.order_details as det
on piz.pizza_id = det.pizza_id
group by piz.size
order by count(size) desc;
```

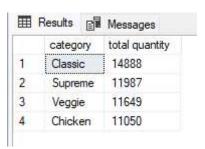
5. List the top 5 most ordered pizza types along with their quantities.

```
select top 5
typ.name,
sum(det.quantity) as 'Total_quantity'
from dbo.pizza_types as typ
join dbo.pizzas as piz on typ.pizza_type_id = piz.pizza_type_id
join dbo.order_details as det on piz.pizza_id = det.pizza_id
group by
typ.name
order by sum(det.quantity) desc;
```



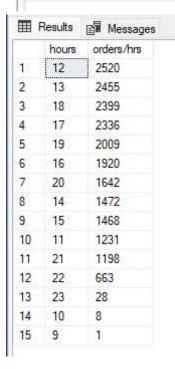
6. Find the total quantity of each pizza category ordered.

```
typ.category,
sum(det.quantity) as 'total quantity'
from dbo.pizza_types as typ
join dbo.pizzas as piz on typ.pizza_type_id = piz.pizza_type_id
join dbo.order_details as det on det.pizza_id = piz.pizza_id
group by
typ.category
order by 'total quantity' desc;
```



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

```
datepart(hh, time) as 'hours',
count(order_id) as 'orders/hrs'
from dbo.orders
group by datepart(hh, time)
order by count(order_id) desc;
```



8. Find the category-wise distribution of pizzas.

```
category,
count(name) as 'pizzas/category'
from dbo.pizza_types
group by category;
```

	category	pizzas/category
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

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

```
select avg(quantity) as "Avg pizzas ordered/day"
from
  (select
    ord.date,
    sum(det.quantity) as "quantity"
    from dbo.orders as ord
    join dbo.order_details as det on ord.order_id = det.order_id
    group by
    ord.date)
    as "order_quantity";
```



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

```
typ.name,
sum((det.quantity * piz.price)) as "Revenue"
from dbo.pizza_types as typ
join dbo.pizzas as piz on typ.pizza_type_id = piz.pizza_type_id
join dbo.order_details as det on piz.pizza_id = det.pizza_id
group by
typ.name
order by Revenue desc;
```

	name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

11. Calculate the percentage contribution of each pizza type to total revenue.

```
typ.category,
Round( sum((det.quantity * piz.price)) / (select sum(det.quantity * piz.price)
from dbo.pizza_types as typ join dbo.pizzas as piz on typ.pizza_type_id = piz.pizza_type_id
join dbo.order_details as det on piz.pizza_id = det.pizza_id ) * 100, 2) as "Revenue"

from dbo.pizza_types as typ
join dbo.pizzas as piz on typ.pizza_type_id = piz.pizza_type_id
join dbo.order_details as det on piz.pizza_id = det.pizza_id
group by
typ.category
order by Revenue desc;
```

	category	Revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

## 12. Analyze the cumulative revenue generated over time.

```
select date, sum(Revenue)
    over(order by date) as "Cumulative Revenue"
    from
    (select
    ord.date,
    sum((det.quantity * piz.price)) as "Revenue"
    from dbo.order_details as det
    join dbo.pizzas as piz on det.pizza_id = piz.pizza_id
    join dbo.orders as ord on ord.order_id = det.order_id
    group by
    ord.date)
    as "sales";
```

	date	Cumulative Revenue
1	2015-01-01	2713.85000228882
	2015-01-02	5445,7500038147
3	2015-01-03	8108.15000724792
4	2015-01-04	9863.60000801086
5	2015-01-05	11929.5500087738
6	2015-01-06	14358.5000114441
7	2015-01-07	16560.700012207
8	2015-01-08	19399.0500183105
9	2015-01-09	21526.4000225067
10	2015-01-10	23990.350025177
11	2015-01-11	25862.6500263214
12	2015-01-12	27781.7000274658
12	2015.01.12	20021 2000270472

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

```
select category, name, revenue
from
  (select category, name, revenue, rank() over(partition by category order by revenue desc) as
from
  (select
  typ.category,
  typ.name,
  sum((det.quantity * piz.price)) as "revenue"
  from dbo.pizza_types as typ
  join dbo.pizzas as piz on typ.pizza_type_id = piz.pizza_type_id
  join dbo.order_details as det on piz.pizza_id = det.pizza_id
  group by
  typ.category,
  typ.name ) as a) as b
  where rn <= 3;</pre>
```

	category	name	revenue
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.5
10	Veggie	The Four Cheese Pizza	32265
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5