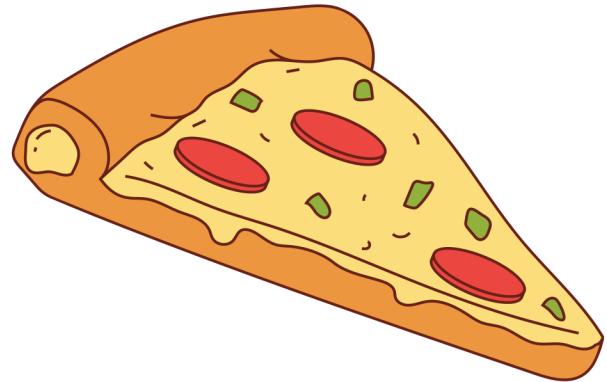


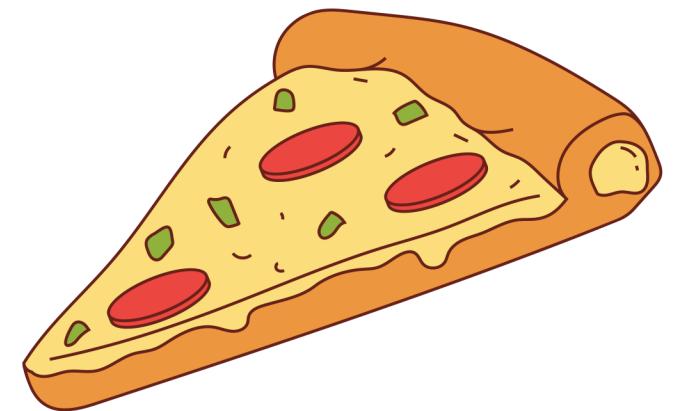
SQL PROJECT ON PIZZA-SALES



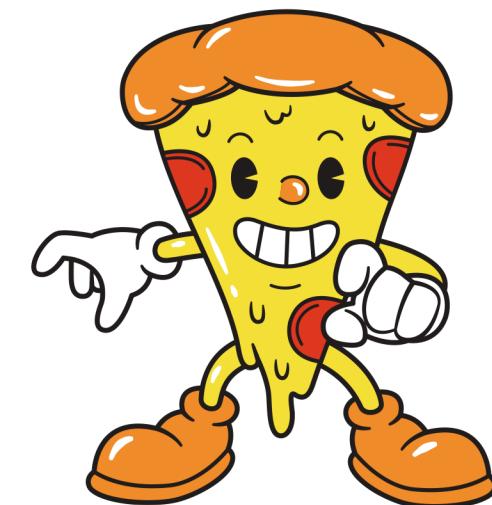
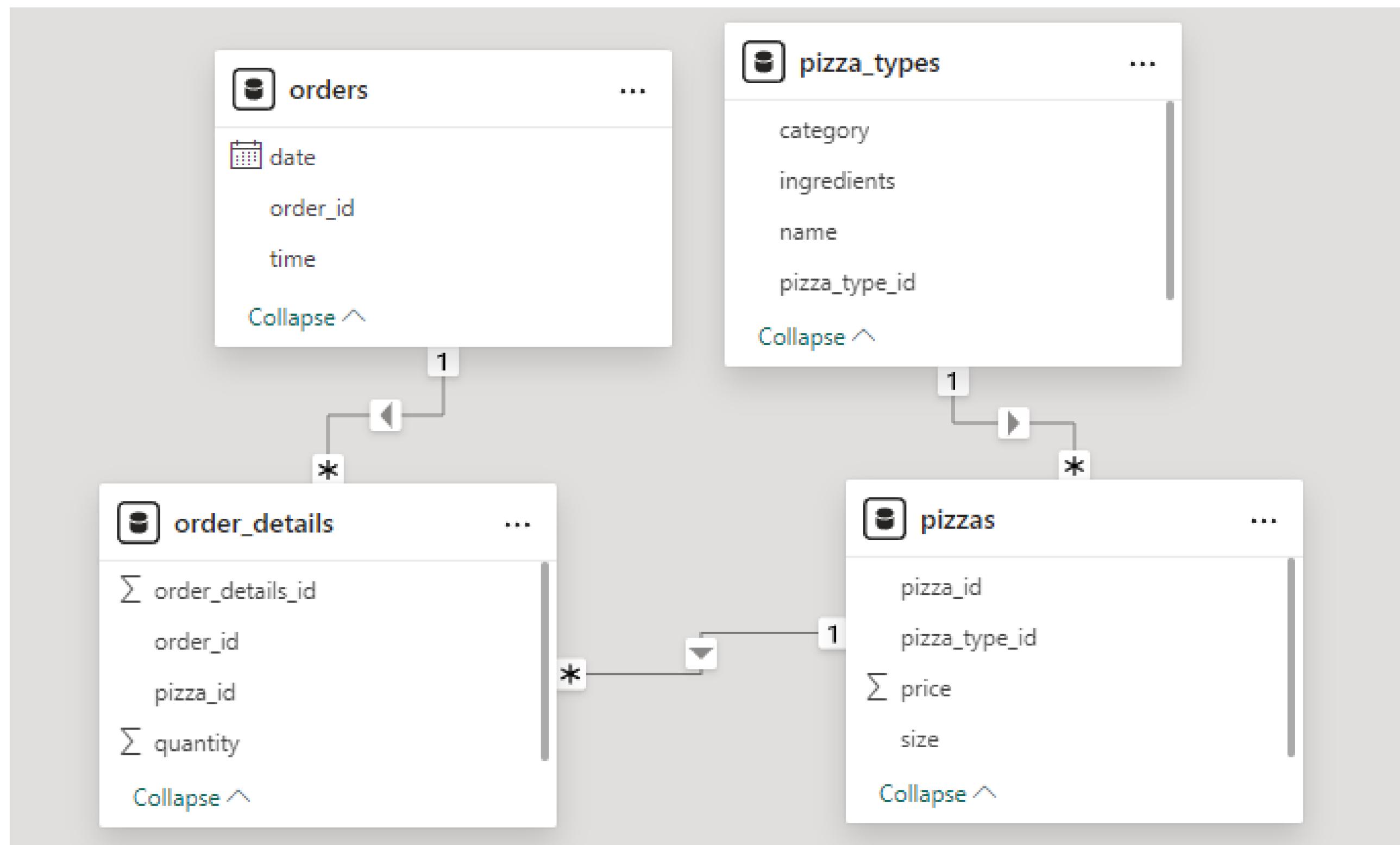
Hello!



My name is Vamshi Krishna in this Project
I have utilized SQL Queries to solve the
questions that were related to Pizza-Sales

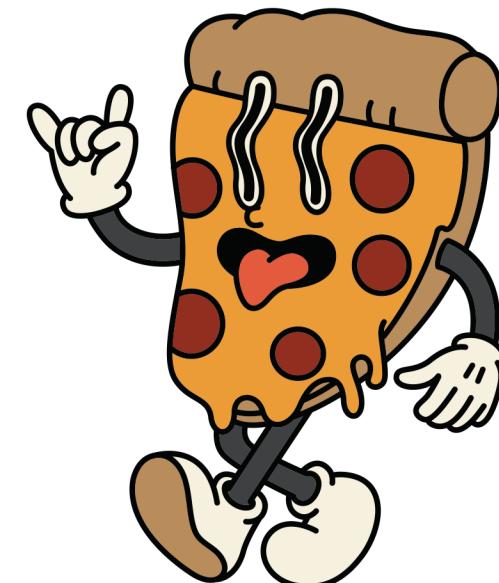


Schema



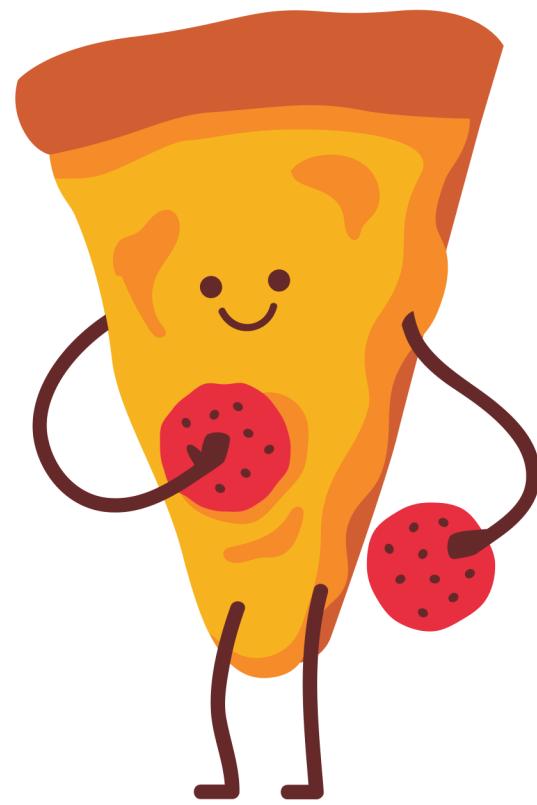
DATA OVERVIEW

- **order_details_id:** Unique identifier for each order detail record.
- **order_id:** Unique identifier for each order.
- **pizza_id:** Unique identifier for each pizza.
- **quantity:** The quantity of pizzas ordered for a particular order detail.
- **date:** The date when the order was placed.
- **time:** The time when the order was placed.
- **pizza_type_id:** Unique identifier for each type of pizza.
- **name:** The name of the pizza type.
- **category:** The category of the pizza (e.g., Chicken, Vegetarian).
- **ingredients:** The ingredients used in the pizza.
- **size:** The size of the pizza (e.g., Small, Medium, Large).
- **price:** The price of the pizza for a particular size.



Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS Total_Orders  
FROM  
    orders
```



	Total_Orders
▶	21350

Calculate the total revenue generated from pizza sales.

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



Result Grid	
	Total_Revenue
▶	817860.05

Identify the highest-priced pizza.

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



Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

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

```
SELECT  
    pt.name, SUM(od.quantity) AS quantity  
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 quantity DESC  
LIMIT 5
```



Result Grid | Filter Rows:

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

Identify the most common pizza size ordered.

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



Result Grid | Filter |

	size	order_count
▶	L	18526

Join the necessary tables to find the total quantity of each pizza category ordered.

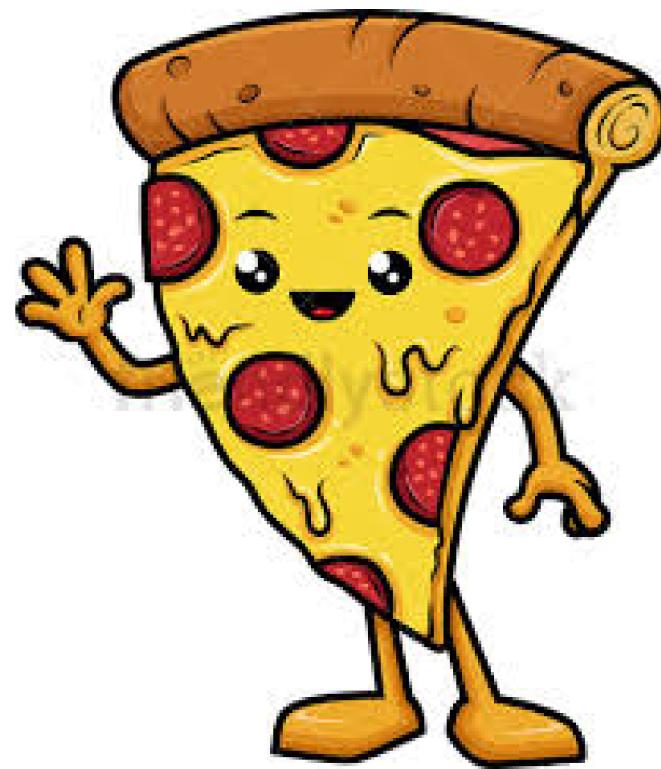
```
SELECT  
    pt.category, SUM(od.quantity) AS Total_Quantity  
FROM  
    pizza_types pt  
        JOIN  
    pizzas p ON p.pizza_type_id = pt.pizza_type_id  
        JOIN  
    order_details od ON od.pizza_id = p.pizza_id  
GROUP BY pt.category  
ORDER BY Total_Quantity DESC
```



	category	Total_Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(order_time) AS Hour, COUNT(order_id) AS Order_count  
FROM  
    orders  
GROUP BY HOUR(order_time)
```



	Hour	Order_count
▶	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
	10	8
	9	1

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

SELECT

category, COUNT(name) AS types

FROM

pizza_types

GROUP BY category



Result Grid

	category	types
▶	Chicken	6
▶	Classic	8
▶	Supreme	9
▶	Veggie	9

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

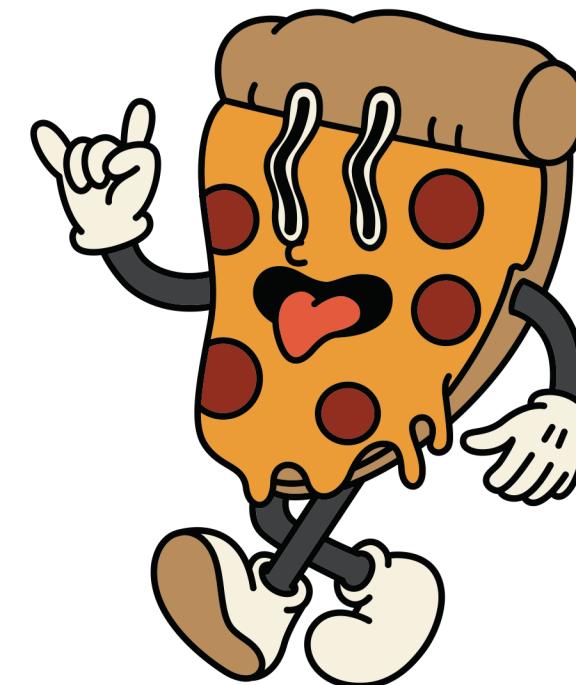
```
SELECT  
    ROUND(AVG(Total_quantity), 0) AS Avg_Orders  
FROM  
    (SELECT  
        o.order_date, SUM(od.quantity) AS Total_quantity  
    FROM  
        orders o  
    JOIN order_details od ON od.order_id = o.order_id  
    GROUP BY o.order_date) AS order_quantity
```



Result Grid	
	Avg_Orders
▶	138

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

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



	name	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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

SELECT

```
pt.category,  
ROUND(SUM(od.quantity * p.price) / (SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS Total_sales  
FROM  
    order_details od  
    JOIN  
    pizzas p ON p.pizza_id = od.pizza_id) * 100,  
2) AS Revenue  
FROM  
pizza_types pt  
JOIN  
pizzas p ON p.pizza_type_id = pt.pizza_type_id  
JOIN  
order_details od ON od.pizza_id = p.pizza_id  
GROUP BY pt.category
```



Result Grid |   Filter

	category	Revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

Analyze the cumulative revenue generated over time.

```
select order_date,  
       sum(Revenue) over(order by order_date) as cum_revenue  
  from  
  (select o.order_date,sum(od.quantity * p.price) as Revenue  
   from orders o  
   join order_details od on od.order_id = o.order_id  
   join pizzas p on p.pizza_id = od.pizza_id  
  group by o.order_date) as sales
```



	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	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
	2015-01-09	21526.4
	2015-01-10	23990.350000000002
	2015-01-11	25862.65

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

```
with cte1 as(  
    select pt.category,pt.name,sum(od.quantity * p.price) as revenue  
    from pizza_types pt  
    join pizzas p on p.pizza_type_id = pt.pizza_type_id  
    join order_details od on od.pizza_id = p.pizza_id  
    group by pt.category,pt.name  
    order by revenue desc)  
,cte2 as (  
    select *,  
    rank() over(partition by category order by revenue desc) as rn  
    from cte1)  
select name,revenue  
from cte2  
where rn < 4;
```



	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	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
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

THANK YOU

