

SQL PROJECT

on

pizza



Kousar Ali .I

Find Us!



BEST

pizza

Hello,

My name is Kousar Ali.I ,And in this project i have utilized sql query to solve questions which are related to pizzas sales



pizza

COMBO



QUESTIONS

BASIC-ADVANCE

Basic:

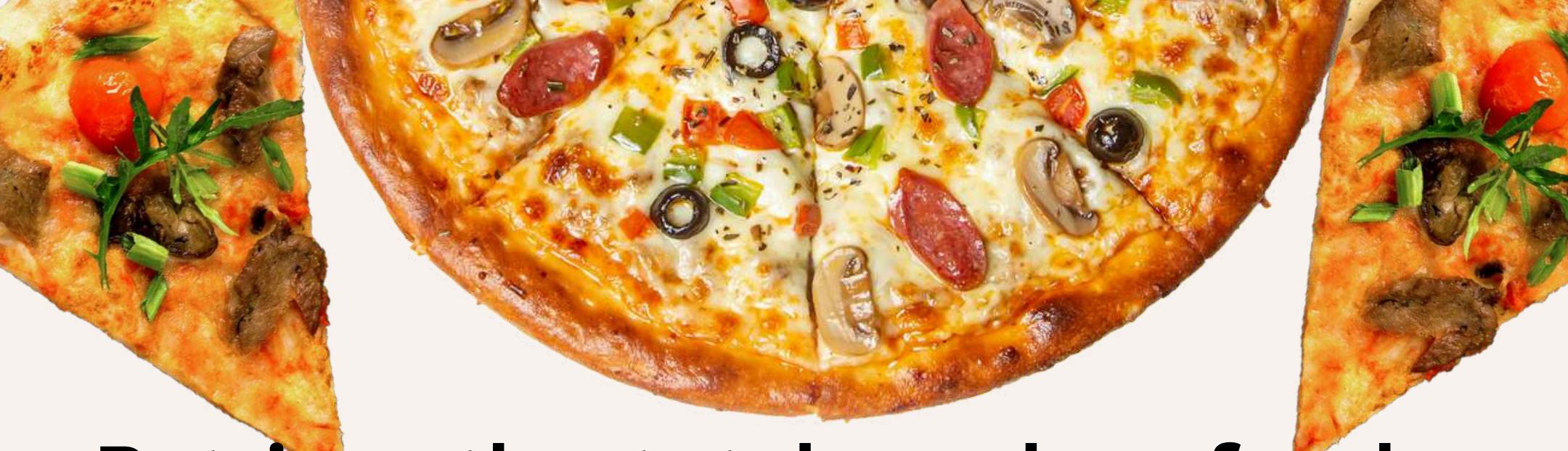
1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

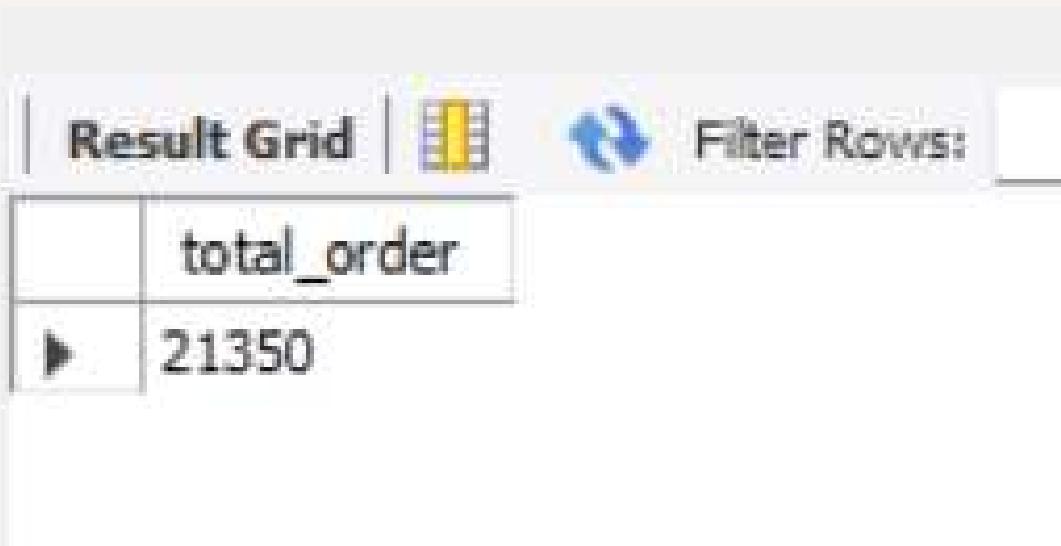
1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



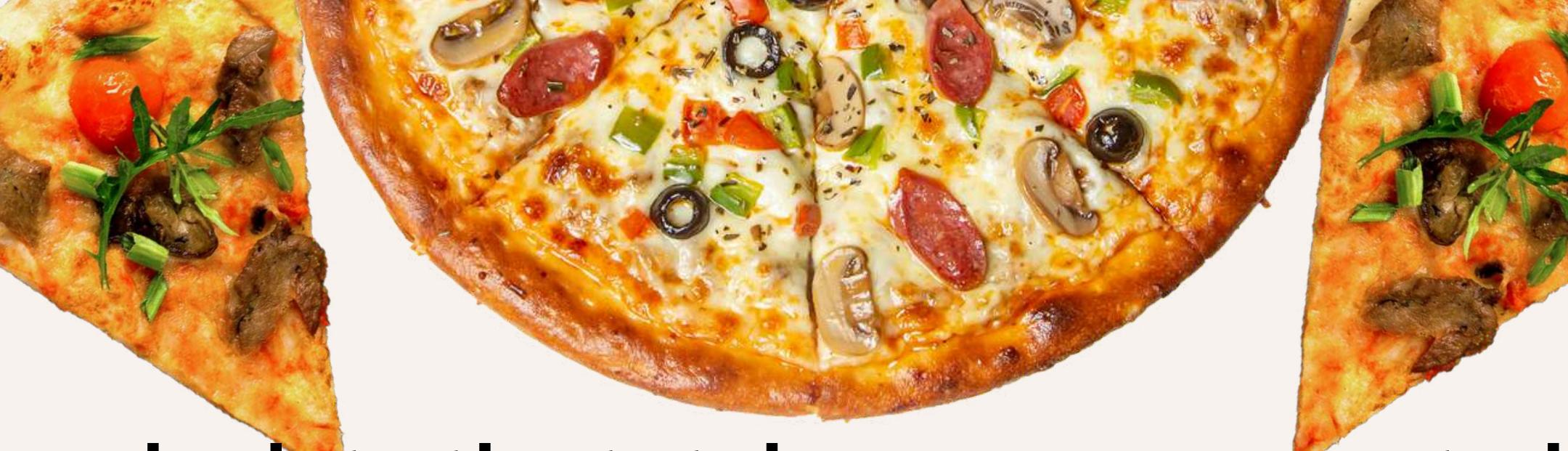
Retrieve the total number of orders placed

```
# retrieve the total number of orders placed
```

```
SELECT  
    COUNT(order_id) AS total_order  
FROM  
    orders;
```



total_order
21350



calculate the total revenue generated from pizza sales

```
SELECT  
    ROUND(SUM(order_detail.quantity * pizzas.price),  
        2) AS total_sales  
FROM  
    order_detail  
    JOIN  
    pizzas ON pizzas.pizza_id = order_detail.pizza_id;
```



Result Grid	
	total_sales
▶	817860.05



Identify the highest price-pizza

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

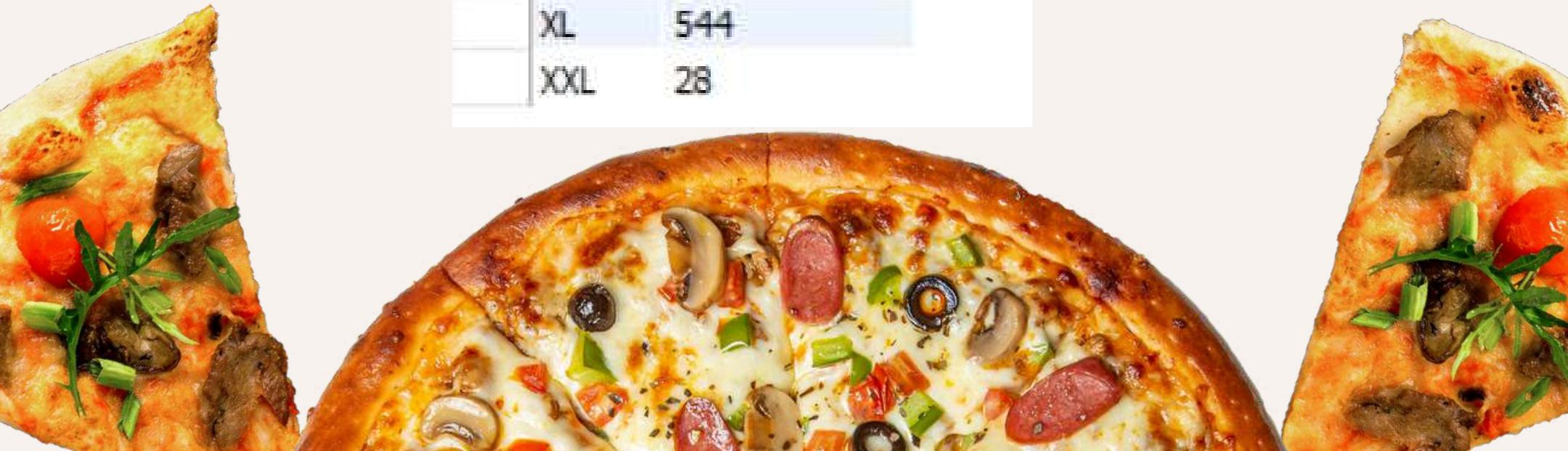
Result Grid | Filter Rows:

	Pizza_name	price
→	The Greek Pizza	35.95



Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(order_detail.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_detail ON pizzas.pizza_id = order_detail.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

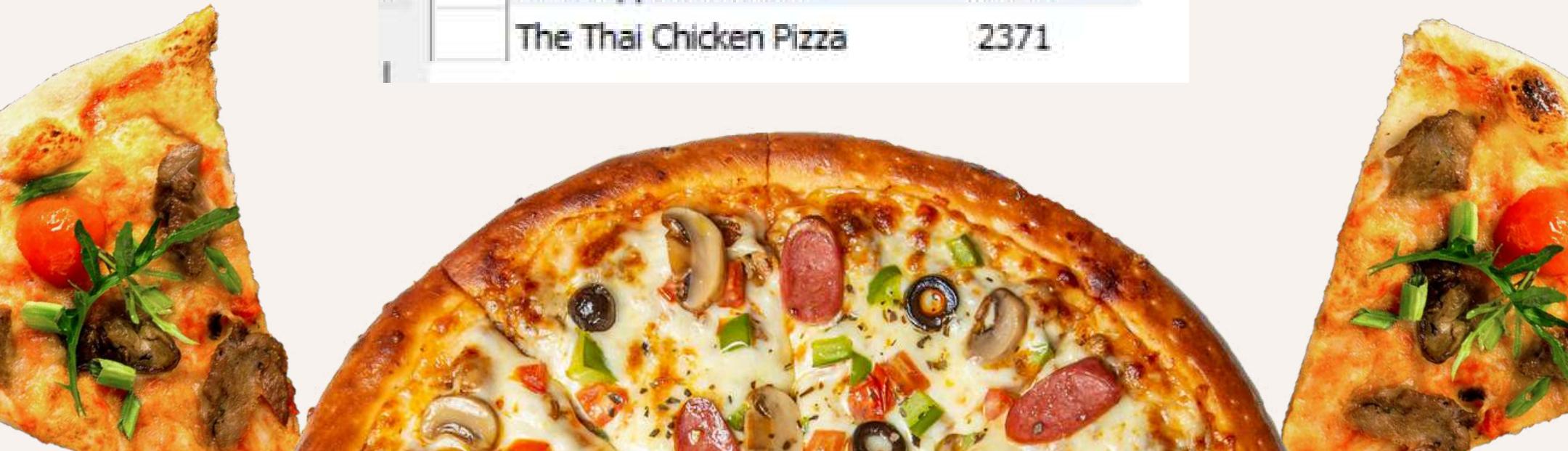


	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



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

```
select pizza_types.name,  
sum(order_detail.quantity)as quantity from pizza_types join  
pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_detail on  
order_detail.pizza_id = pizzas.pizza_id  
group by name order by quantity desc limit 5;
```



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



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

SELECT

ordered.

AS quantity

FROM

 pizza_types

JOIN

 pizzas **ON** pizza_types.pizza_type_id = pizzas.pizza_type_id

JOIN

 order_detail **ON** order_detail.pizza_id = pizzas.pizza_id

GROUP BY pizza_types.category

ORDER BY quantity **DESC**;

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

SELECT

category, COUNT(name) **AS** distribution

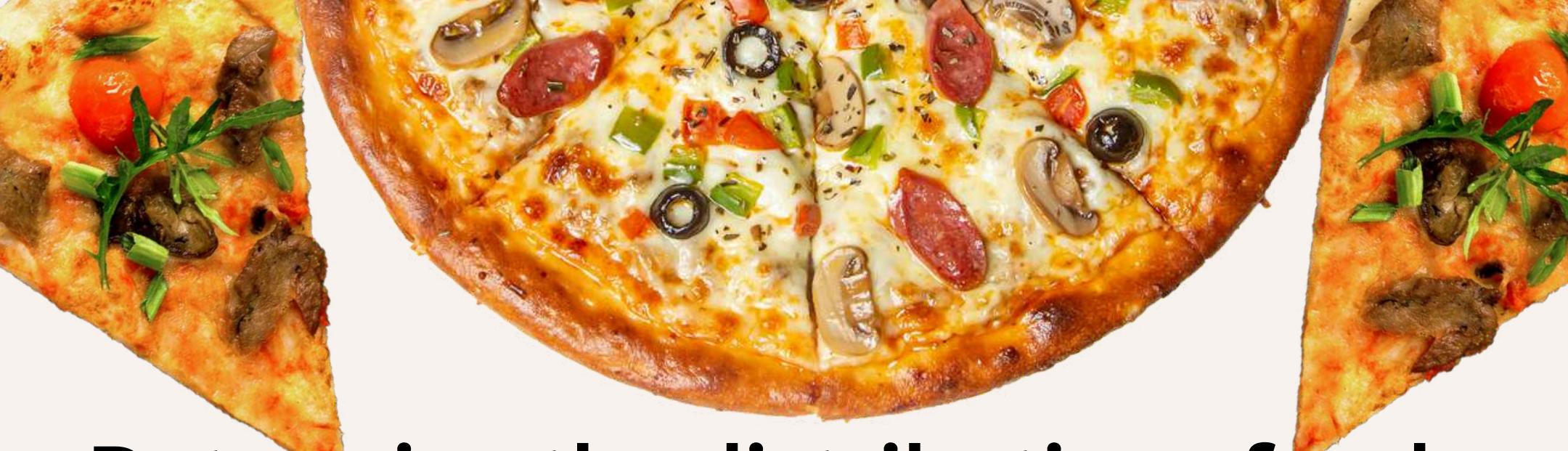
FROM

pizza_types

GROUP BY category;

Result Grid | Filter Rows:

	category	distribution
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



Determine the distribution of orders by hour of the day.

SELECT

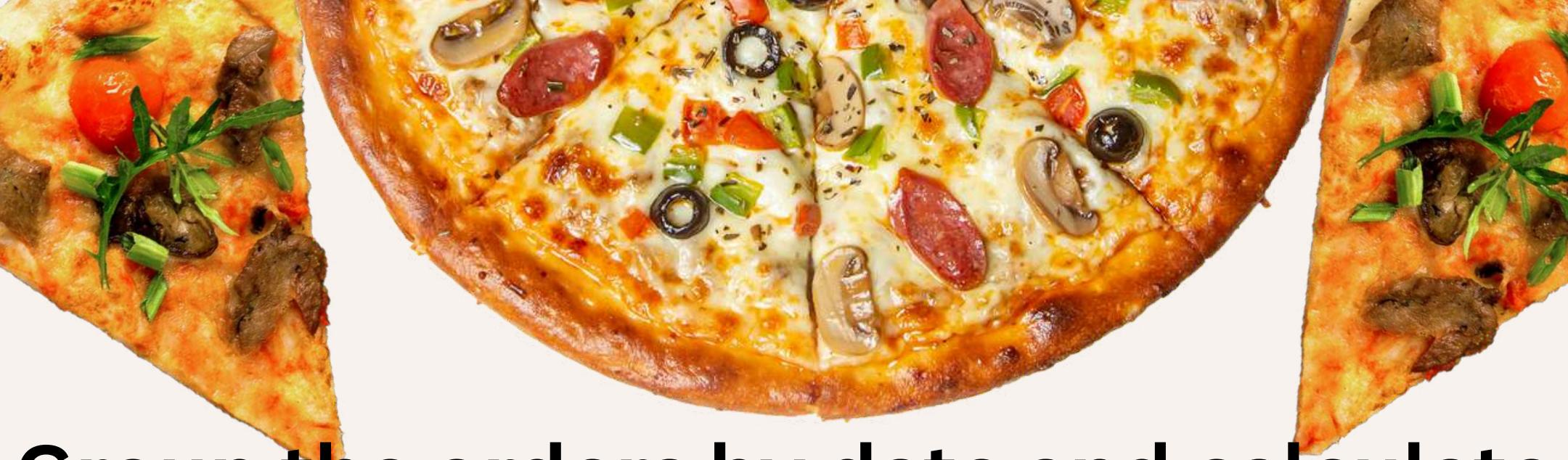
HOUR(order_time) AS hour, COUNT(Order_id)

FROM

orders

GROUP BY hour;

	hour	COUNT(Order_id)
▶	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

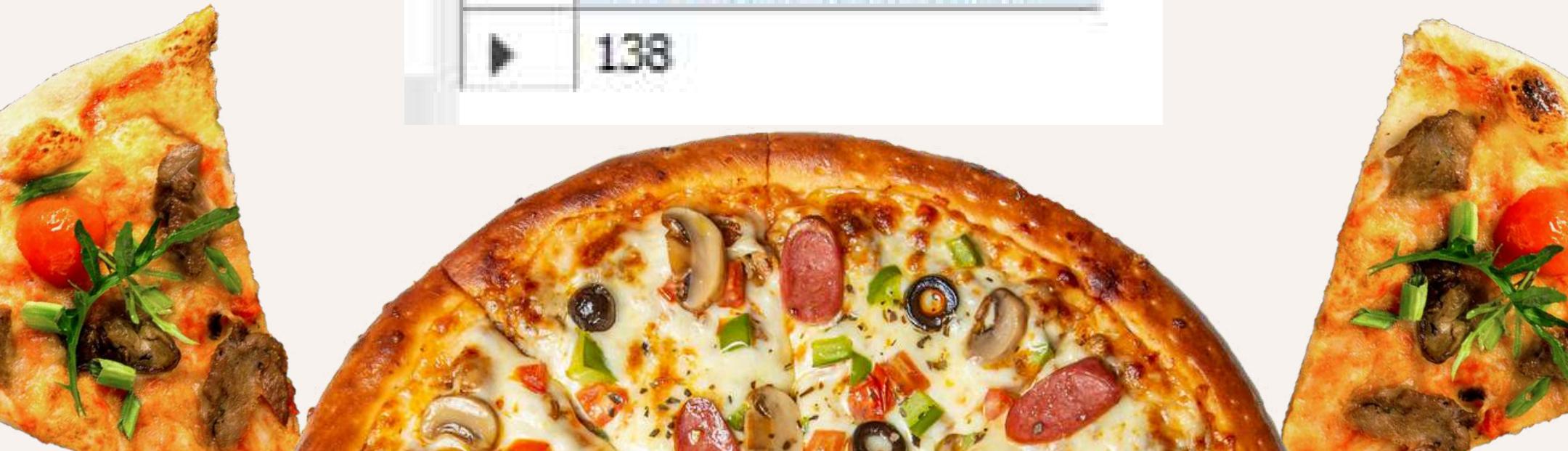


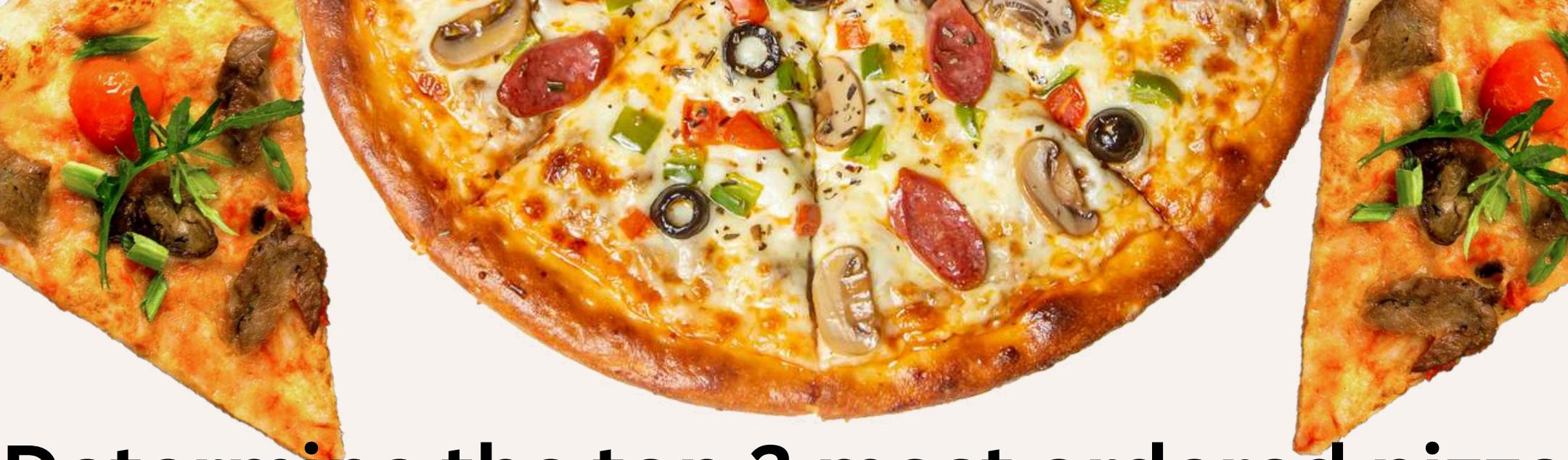
Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(order_quantity), 0) AS Avg_per_day_order
FROM
    (SELECT
        orders.order_date,
        SUM(order_detail.quantity) AS order_quantity
    FROM
        orders
    JOIN order_detail ON orders.order_id = order_detail.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter

	Avg_per_day_order
▶	138





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

```
SELECT
    pizza_types.name,
    SUM(order_detail.quantity * pizzas.price) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.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 pizza_types.category,  
round(sum(order_detail.quantity*pizzas.price) / (select  
round(sum(order_detail.quantity*pizzas.price),2)as total_sales  
from order_detail join  
pizzas on pizzas.pizza_id =order_detail.pizza_id)*100,2) as revenue  
from  
pizza_types join pizzas  
on pizza_types.pizza_type_id=pizzas.pizza_type_id  
join order_detail  
on order_detail.pizza_id =pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```

Result Grid | Filter R

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

```
select order_date,  
       sum(revenue) over(order by order_date) as cum_revenue  
  from  
    (select orders.order_date,  
           sum(order_detail.quantity * pizzas.price) as revenue  
      from order_detail join pizzas  
        on order_detail.pizza_id = pizzas.pizza_id  
     join orders  
       on orders.order_id=order_detail.order_id  
      group by orders.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	14359.5



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

```
select name,revenue from
(select category,name,revenue,
dense_rank() over(partition by category order by revenue desc) as rnk
from
(select pizza_types.category,pizza_types.name,
sum(order_detail.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_detail
on order_detail.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name)as a)as b
where rnk<=3;
```

	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

THANK



YOU!!

Kousar Ali I(Sql project)

