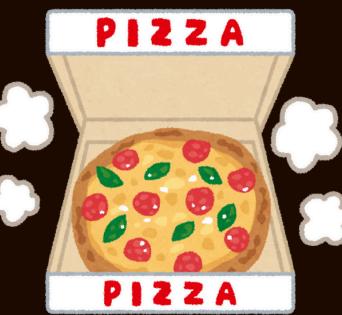


PLATO'S PIZZA

RESTAURANT

SQL DATA ANALYSIS PROJECT





Hello, my name is Akash Pandhare. In this project, I utilized SQL queries to analyze and solve various questions related to pizza sales for Plato's Pizza. My SQL expertise includes working with complex joins, aggregations, window functions, and subqueries to extract meaningful insights from large datasets. Through this project, I demonstrated my ability to manipulate and analyze data effectively, providing actionable insights that can help enhance business strategies and operational efficiency.

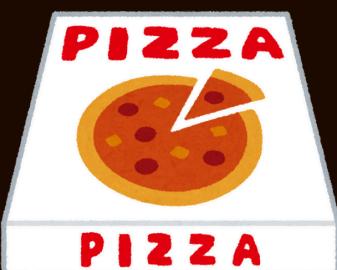
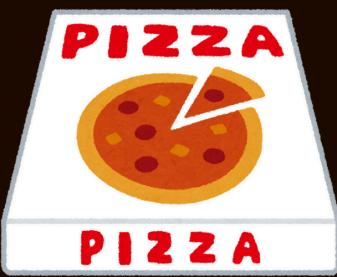
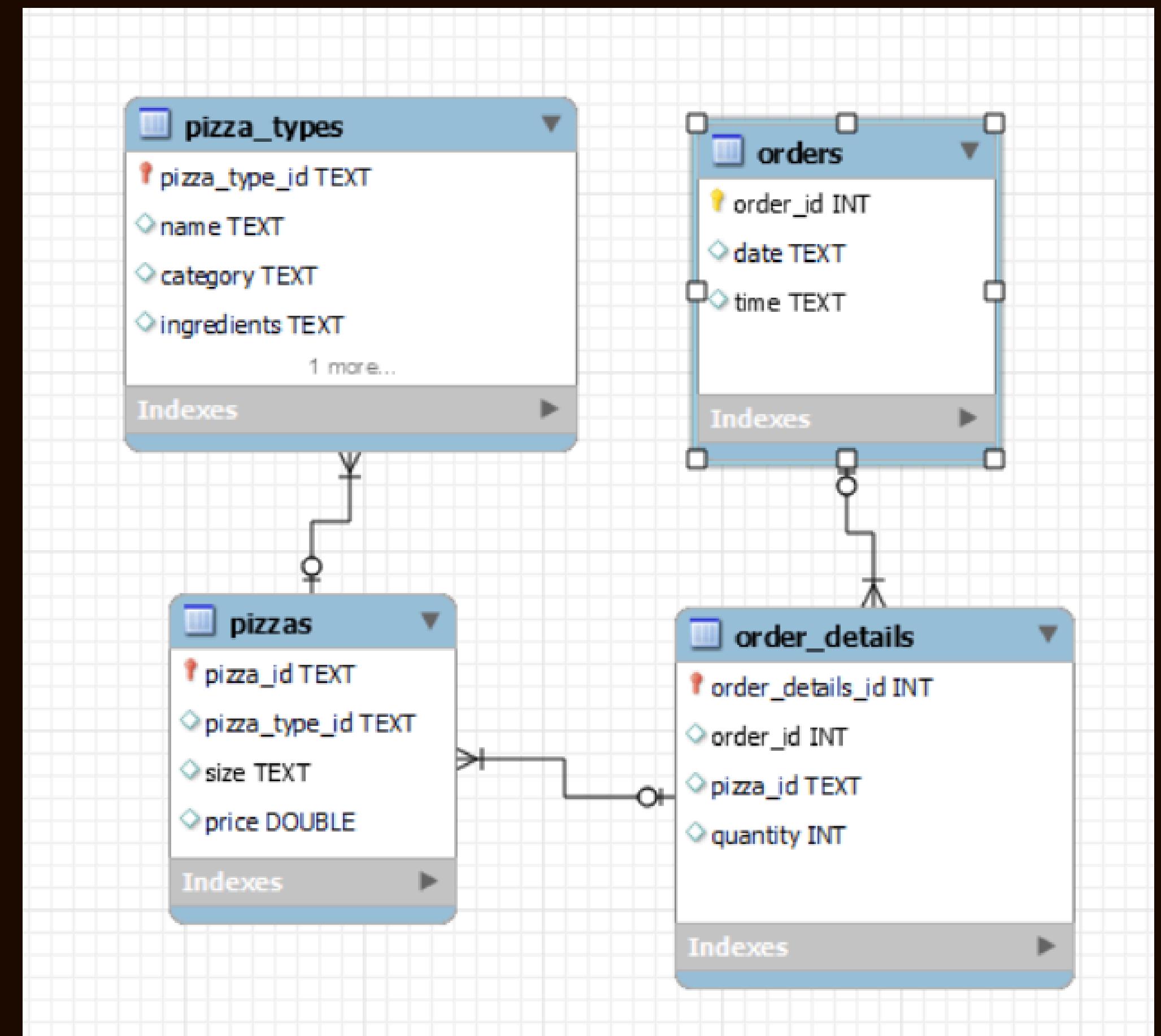
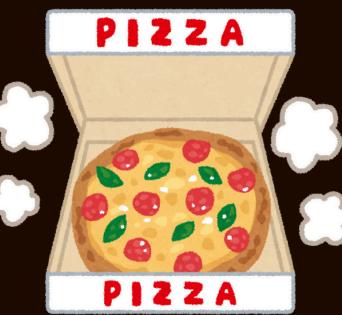




Table schemas

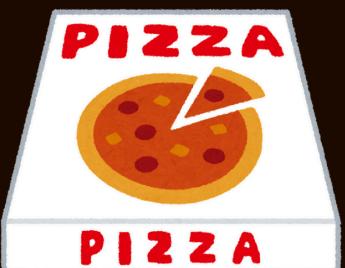


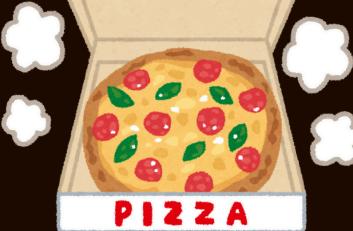
Retrieve the total number of orders placed.



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

	Total_orders
▶	21350

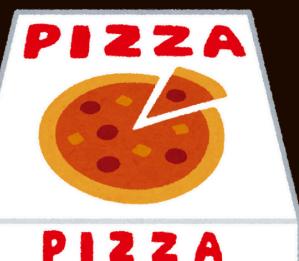




Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(p.price * o.quantity), 1) AS Revenue  
FROM  
    pizzas p  
    JOIN  
    order_details o ON p.pizza_id = o.pizza_id;
```

	Revenue
▶	817860

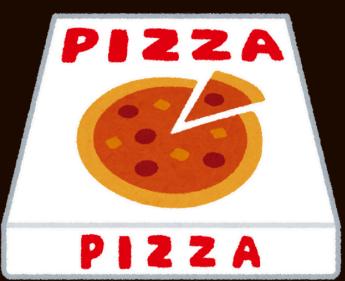


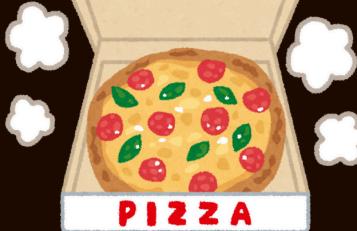


Identify the highest-priced pizza.

```
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;
```

	name	price
▶	The Greek Pizza	35.95



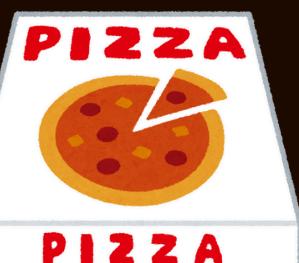


Identify the most common pizza size ordered.

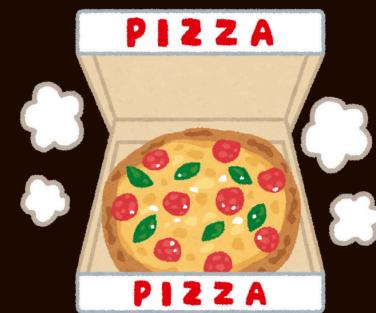
SELECT

```
p.size, COUNT(o.order_id) AS total_order  
FROM  
    pizzas p  
        JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
        JOIN  
    orders o ON o.order_id = od.order_id  
GROUP BY p.size  
ORDER BY 2 DESC;
```

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



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



SELECT

```
pt.name, SUM(od.quantity) AS total_order
```

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
```

JOIN

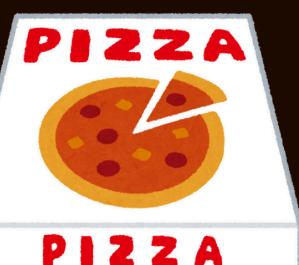
```
orders o ON o.order_id = od.order_id
```

GROUP BY 1

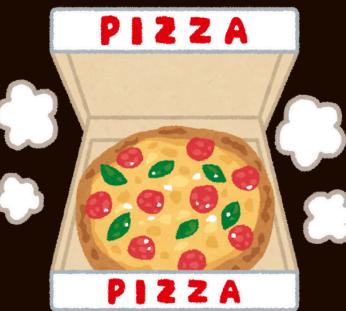
ORDER BY 2 DESC

LIMIT 5;

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

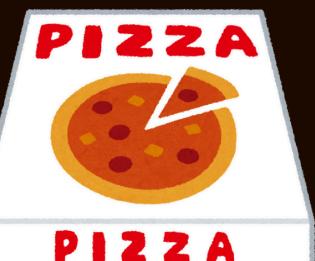


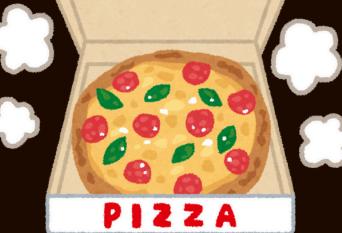
Join the necessary tables to find the total quantity of each pizza category ordered.



```
SELECT
    pt.category, SUM(od.quantity) AS total_order
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
        JOIN
    orders o ON o.order_id = od.order_id
GROUP BY 1
ORDER BY 2 DESC;
```

	category	total_order
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

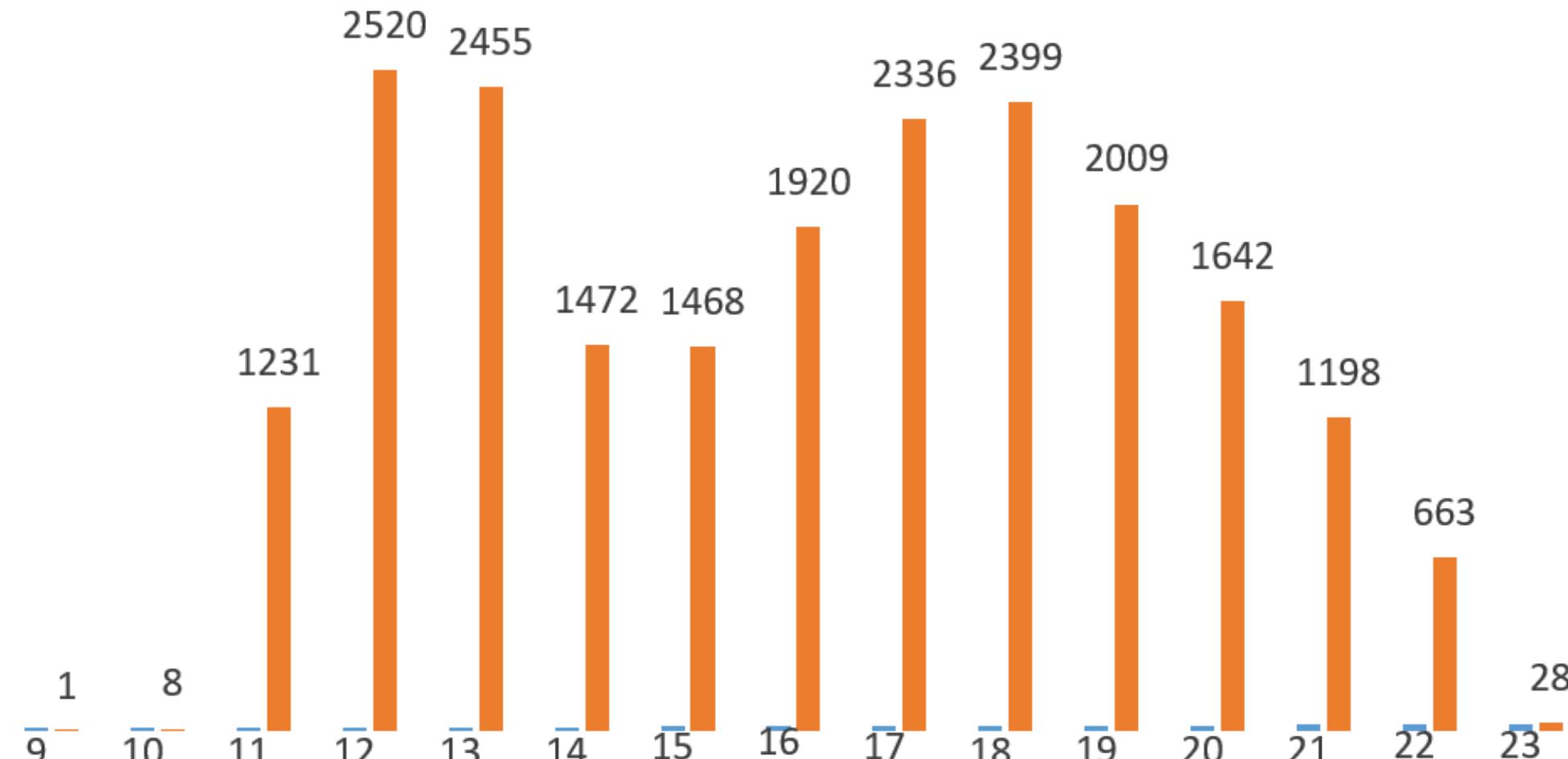




Determine the distribution of orders by hour of the day.

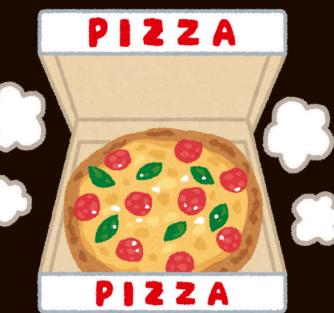
```
SELECT DISTINCT
    HOUR(time) AS hour, COUNT(order_id) AS total_order
FROM
    orders
GROUP BY 1;
```

Hourly order



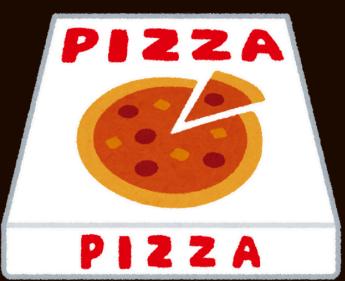
	hour	total_order
11	1231	
12	2520	
13	2455	
14	1472	
15	1468	
16	1920	
17	2336	
18	2399	
19	2009	
20	1642	
21	1198	
22	663	
23	28	
10	8	
9	1	

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

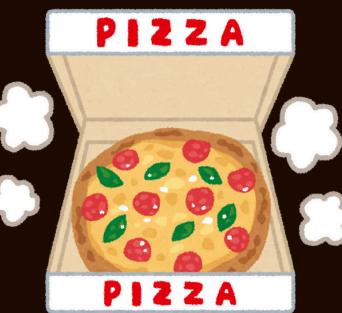


```
select category , count(name) as pizzas from pizza_types  
group by 1  
order by 2 desc;
```

	category	pizzas
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6



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



SELECT

```
ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
```

FROM

```
(SELECT DISTINCT
```

```
o.date, SUM(od.quantity) AS quantity
```

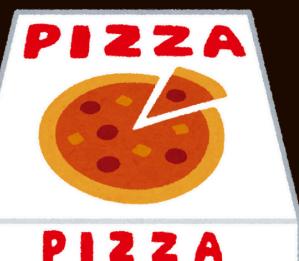
FROM

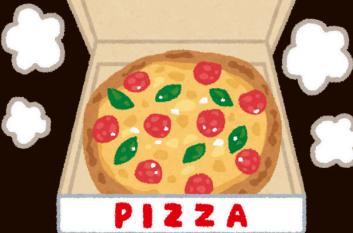
```
orders o
```

```
JOIN order_details od ON od.order_id = o.order_id
```

```
GROUP BY 1) AS order_quantity;
```

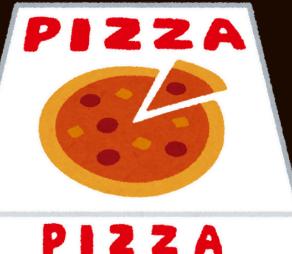
	avg_pizza_ordered_per_day
▶	138





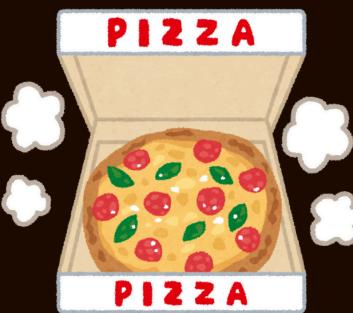
Determine the top 3 most ordered pizza types based on revenue.

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

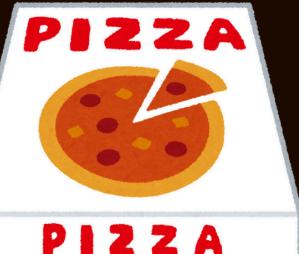


	name	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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



```
SELECT
    pt.category,
    ROUND((SUM(od.quantity * p.price) / (SELECT
        ROUND(SUM(p.price * o.quantity), 1) AS Revenue
    FROM
        pizzas p
        JOIN
        order_details o ON p.pizza_id = o.pizza_id)) * 100,
    2) AS percentage_revenue
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 1
ORDER BY 2 DESC;
```



category	percentage_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

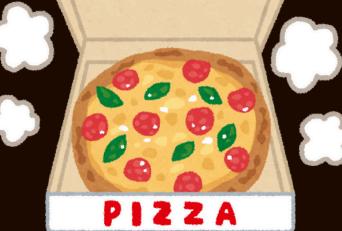
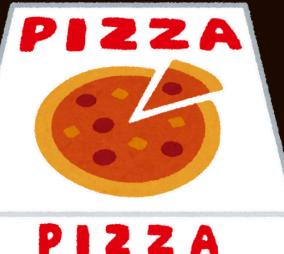
Analyze the cumulative revenue generated over time.

```

with cte1 as
  (SELECT
    MONTH(o.date) AS month,
    ROUND(SUM(od.quantity * p.price), 2) AS revenue
  FROM
    pizzas p
    JOIN
    order_details od ON p.pizza_id = od.pizza_id
    JOIN
    orders o ON o.order_id = od.order_id
  GROUP BY 1
  ORDER BY 1 )
  select month ,
  Round(sum(revenue) over (order by month ) , 2 ) as cumulative_revenue
  from cte1 ;

```

month	cumulative_revenue
1	69793.3
2	134952.9
3	205350
4	274086.8
5	345489.55
6	413719.75
7	486277.65
8	554555.9
9	618735.95
10	682763.55
11	753158.9
12	817860.05





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

```

WITH cte1 AS (
    SELECT pt.category, pt.name,
           round(SUM(od.quantity * p.price),2) AS revenue
    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.category, pt.name
),
cte2 AS (
    SELECT category, name, revenue,
           ROW_NUMBER() OVER (PARTITION BY category ORDER BY revenue DESC) AS r_number
    FROM
        cte1
)
SELECT category, name, revenue
FROM cte2
WHERE r_number <= 3;

```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

