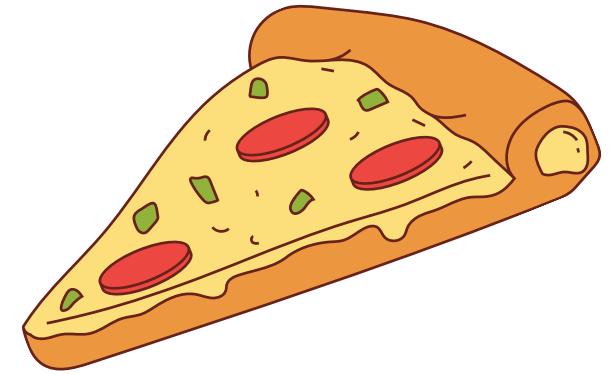
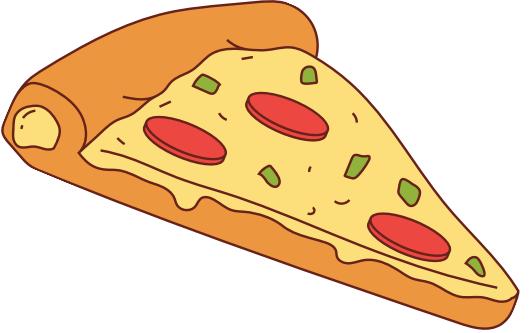


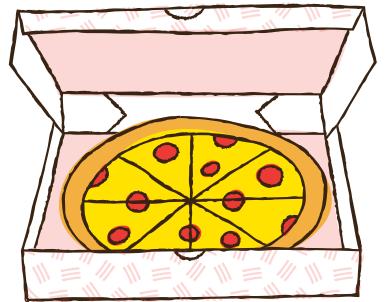
# **PIZZA SALES USING SQL**



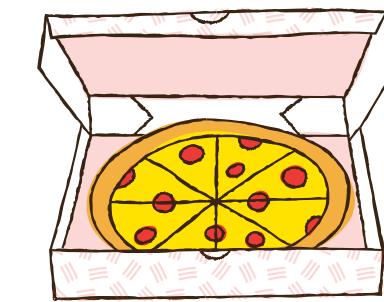
# introduction



In this analysis, we explore pizza sales data from January to December 2015 for a specific store. Pizza, a beloved food choice worldwide, offers insights into consumer preferences and market dynamics. By examining sales volume, revenue, and factors like seasonal variations and promotions, we aim to understand what drives pizza purchases. This study not only benefits the store in question but also sheds light on broader trends within the food industry, helping businesses adapt to consumer needs effectively.



# questions



## Basic:

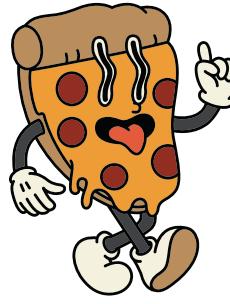
- 1 Retrieve the total number of orders placed.
- 2 Calculate the total revenue generated from pizza sales.
- 3 Identify the highest-priced pizza.
- 4 Identify the most common pizza size ordered.
- 5 List the top 5 most ordered pizza types along with their quantities.

## Intermediate:

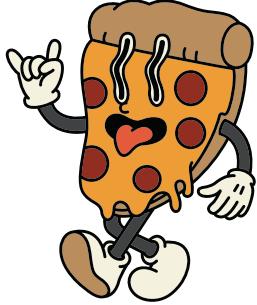
- 6 Join the necessary tables to find the total quantity of each pizza category ordered.
- 7 Determine the distribution of orders by hour of the day.
- 8 Join relevant tables to find the category-wise distribution of pizzas.
- 9 Group the orders by date and calculate the average number of pizzas ordered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

## Advanced:

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



# Retrieve the total number of orders placed.



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

Result Grid	
	total_orders
▶	21350



# calculate total revenue generated from pizza sales.



SELECT

```
    ROUND(SUM(order_details.quantity * pizzas.price),  
          2) AS Total_revenue
```

FROM

```
order_details
```

JOIN

```
pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

	Total_revenue
▶	817860.05



# identify the heighest price pizza.



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

Pizza_name	Heighest_price
The Greek Pizza	35.95



# identify the most common pizza size ordered.



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

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

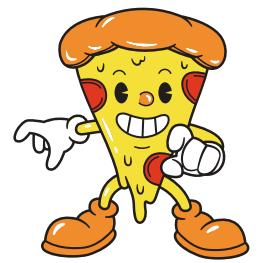
# List the top 5 most ordered pizza types along with their quantities.



```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5
```

	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

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



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

	category	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

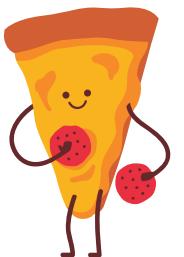
# Join relevant tables to find the category-wise distribution of pizzas



```
select category , count(name) from pizza_types  
group by category
```

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

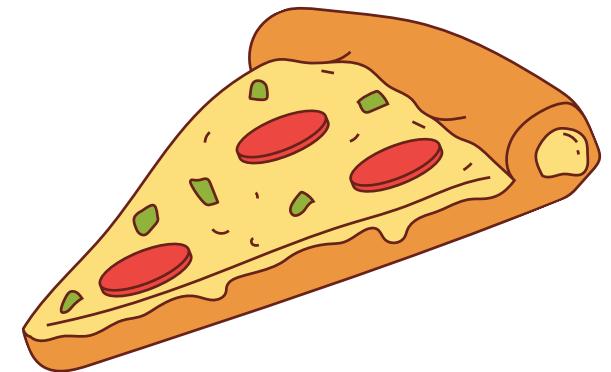
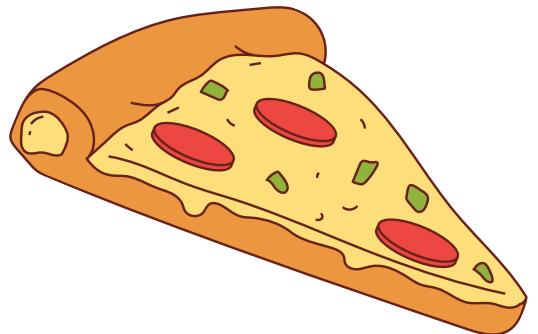


SELECT

```
ROUND(AVG(quantity), 0) as avg_quantity  
FROM  
(SELECT  
    orders.order_date, COUNT(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	avg_quantity
▶	136

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



SELECT

```
pizza_types.name,  
SUM(order_details.quantity * pizzas.price) AS revenue
```

FROM

```
pizza_types  
JOIN  
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
JOIN  
order_details ON order_details.pizza_id = pizzas.pizza_id
```

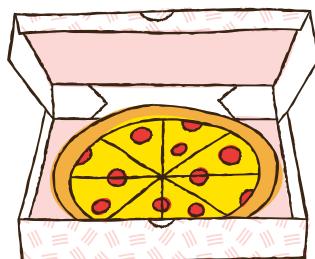
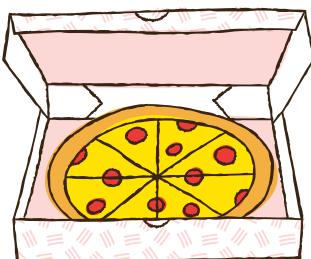
GROUP BY pizza\_types.name

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.

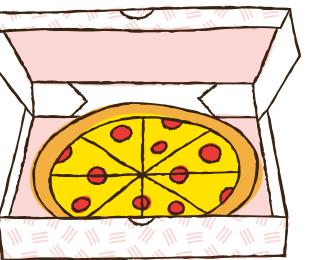
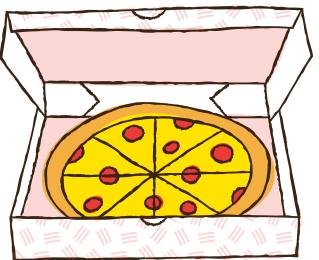


SELECT

```
    pizza_types.category,  
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT  
        SUM(order_details.quantity * pizzas.price) AS total_sales  
    FROM  
        order_details  
        JOIN  
        pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100),  
    2) AS ptr  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category
```

	category	ptr
→	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

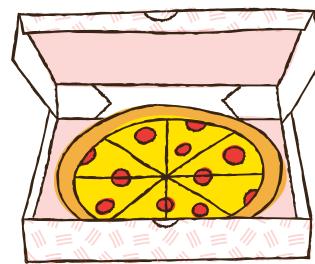
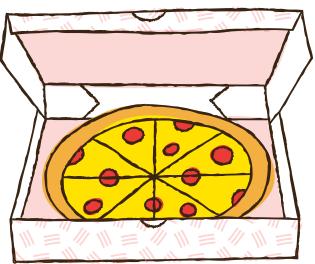
# Analyze the cumulative revenue generated over time.



```
select order_date,  
round(sum(revenue) over(order by order_date),2) as cum_revenue  
from  
(select orders.order_date,  
round(sum(order_details.quantity * pizzas.price),2) as revenue  
from orders join order_details on orders.order_id = order_details.order_id  
join pizzas on order_details.pizza_id = pizzas.pizza_id  
group by orders.order_date) as sales;
```

	order_date	cum_revenue
▶	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65

# 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 rn
from
(select pizza_types.category , pizza_types.name,
round(sum(order_details.quantity * pizzas.price),2) as revenue
from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category , pizza_types.name) as a) as b
where rn <=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

# THANKS !

