

Pizza Sales In Sql



Project: Pizza Sales Analysis using SQL

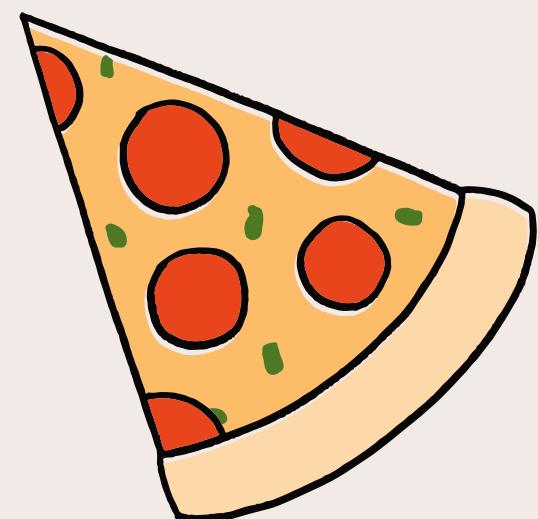
Overview: Completed an in-depth analysis of pizza sales data, utilizing SQL for data manipulation and insights.

Key Learnings:

- Mastered SQL queries ranging from basic to advanced levels.
- Applied various SQL techniques including:
 - Joins: Combined data from multiple tables to create comprehensive reports.
 - Subqueries: Used nested queries for complex data retrieval.
 - Common Table Expressions (CTEs): Simplified complex queries and improved readability.
 - DENSE_RANK: Ranked data without gaps for detailed analysis.

Worked on solving these queries to enhance my understanding and application of SQL.

- 1 Basic:
 - 2 Retrieve the total number of orders placed.
 - 3 Calculate the total revenue generated from pizza sales.
 - 4 Identify the highest-priced pizza.
 - 5 Identify the most common pizza size ordered.
 - 6 List the top 5 most ordered pizza types along with their quantities.
- 7
- 8
- 9 Intermediate:
 - 0 Join the necessary tables to find the total quantity of each pizza category ordered.
 - 1 Determine the distribution of orders by hour of the day.
 - 2 Join relevant tables to find the category-wise distribution of pizzas.
 - 3 Group the orders by date and calculate the average number of pizzas ordered per day.
 - 4 Determine the top 3 most ordered pizza types based on revenue.
- 5
- 6 Advanced:
 - 7 Calculate the percentage contribution of each pizza type to total revenue.
 - 8 Analyze the cumulative revenue generated over time.
 - 9 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

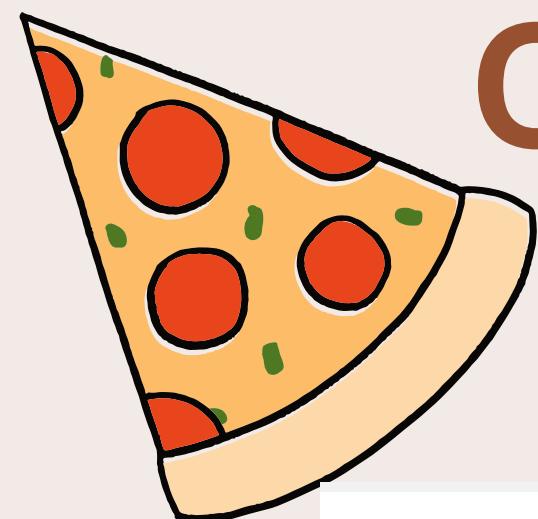


Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_number_of_orders_placed  
FROM  
    table_orders
```

	total_number_of_orders_placed
	21350





Calculate the total revenue generated from pizza sales

```
SELECT  
    ROUND(SUM(p.price * d.Quantity), 2) AS Total_Sales  
FROM  
    pizza.pizzas p  
    JOIN  
        order_details d ON d.pizza_id = p.pizza_id
```

	Total_Sales
▶	817860.05





Identify the highest-priced pizza.

SELECT

pi.name, p.price, p.size

FROM

pizza_types pi

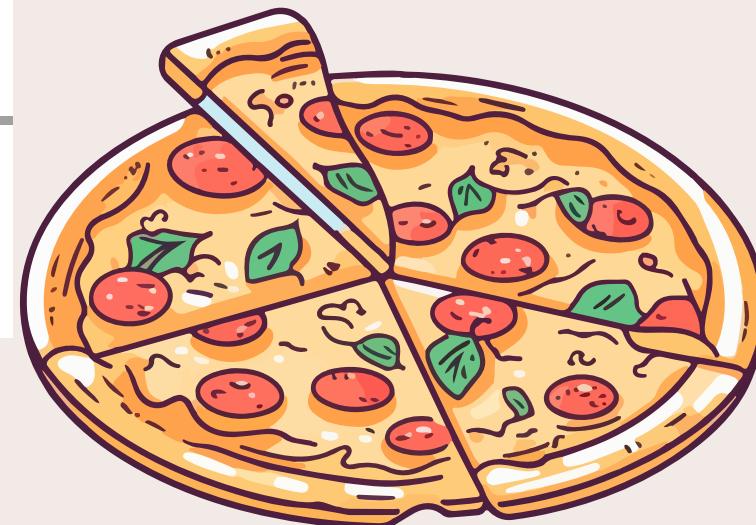
JOIN

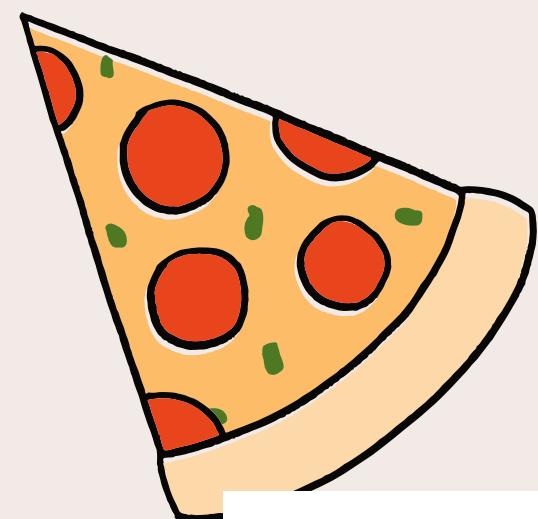
pizzas p **ON** p.pizza_type_id = pi.pizza_type_id

ORDER BY p.price **DESC**

LIMIT 1;

name	price	size
The Greek Pizza	35.95	XXL

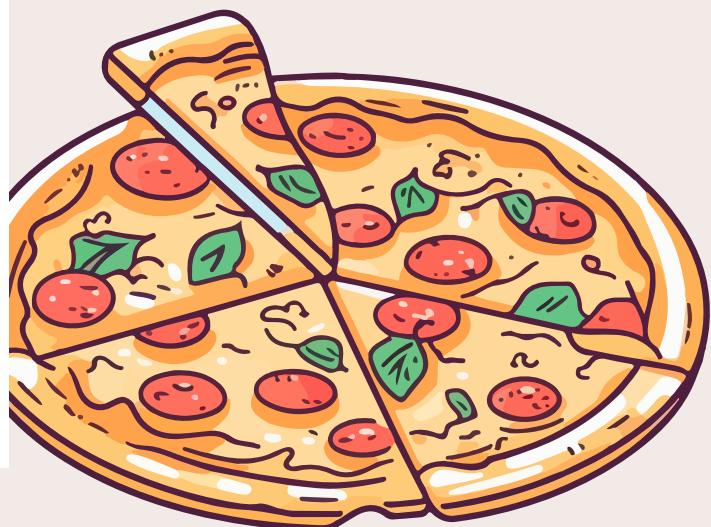




Identify the most common pizza size ordered.

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

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



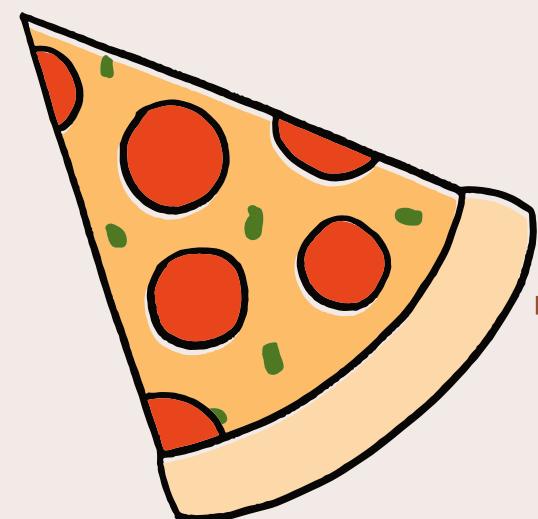


List the top 5 most ordered pizza types along with their quantities.

```
SELECT  
    pi.name, SUM(o.quantity) AS quantity  
FROM  
    pizza_types pi  
        JOIN  
    pizzas p ON pi.pizza_type_id = p.pizza_type_id  
        JOIN  
    order_details o ON p.pizza_id = o.pizza_id  
GROUP BY pi.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

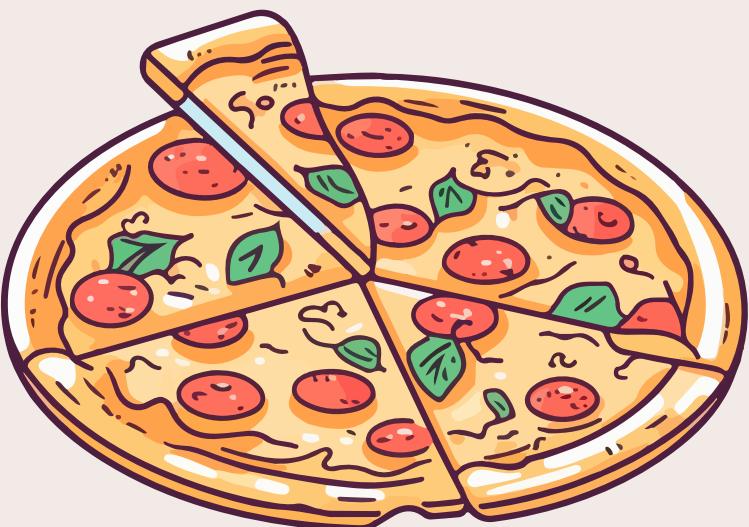


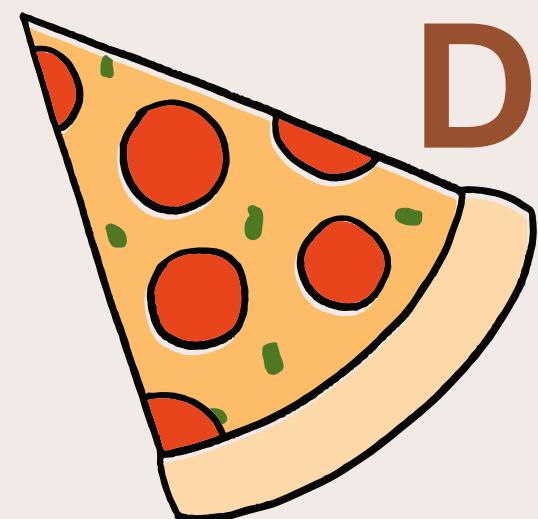


Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT  
    pi.category, SUM(o.quantity) AS quantity  
FROM  
    pizza_types pi  
        JOIN  
    pizzas p ON pi.pizza_type_id = p.pizza_type_id  
        JOIN  
    order_details o ON p.pizza_id = o.pizza_id  
GROUP BY pi.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 orders  
FROM  
    pizza.table_orders  
GROUP BY Hour;
```

	Hour	orders
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

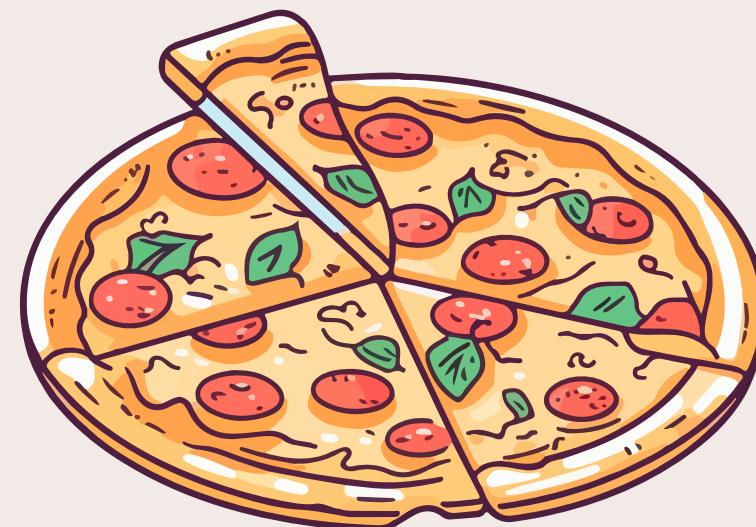


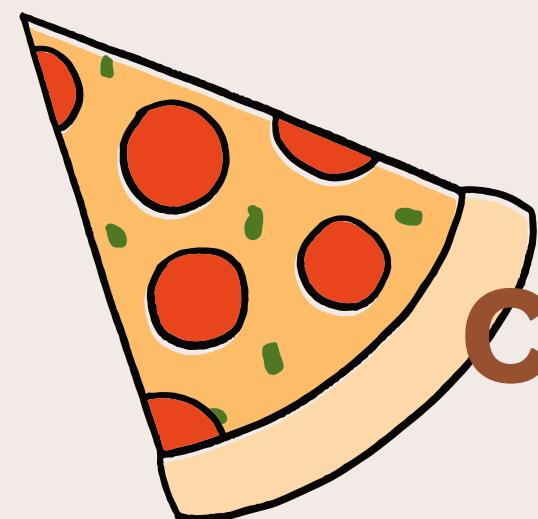


Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT  
    category, COUNT(name) AS Distribution  
FROM  
    pizza_types  
GROUP BY category;
```

	category	Distribution
▶	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** Quantity

FROM

(**SELECT**

t.order_dATE, COUNT(o.quantity) **AS** quantity

FROM

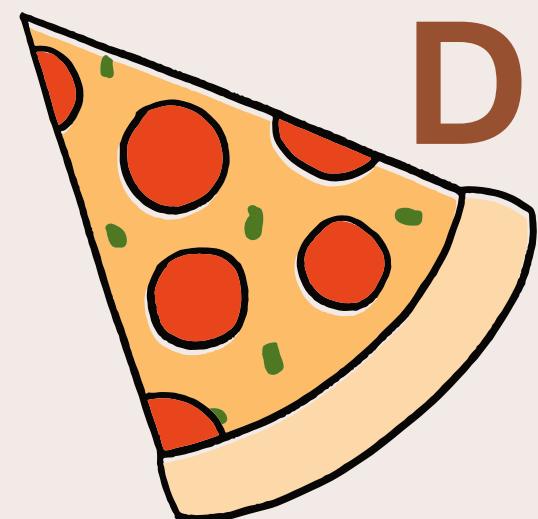
table_orders t

JOIN order_details o **ON** o.order_id = t.order_id

GROUP BY t.order_dATE) **AS** order_quantity;

	Quantity
▶	136





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

```
SELECT  
    Pi.name, ROUND(SUM((o.quantity * p.price)), 2) AS revenue  
FROM  
    pizza_types pi  
        JOIN  
    pizzas p ON p.pizza_type_id = pi.pizza_type_id  
        JOIN  
    order_details o ON p.pizza_id = o.pizza_id  
GROUP BY Pi.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

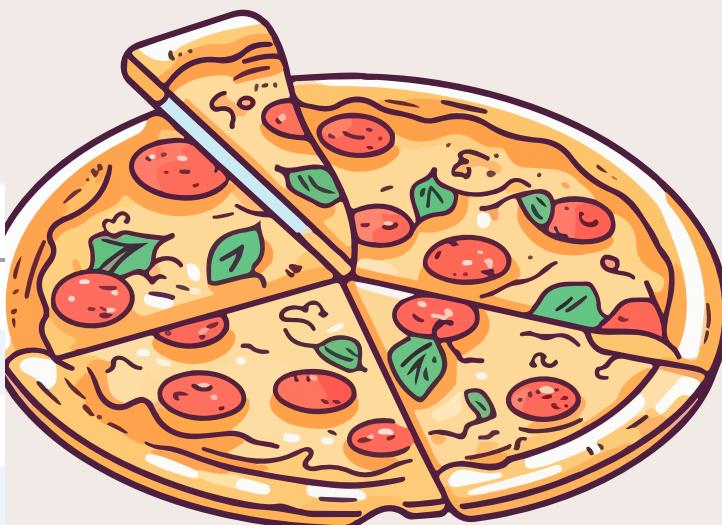


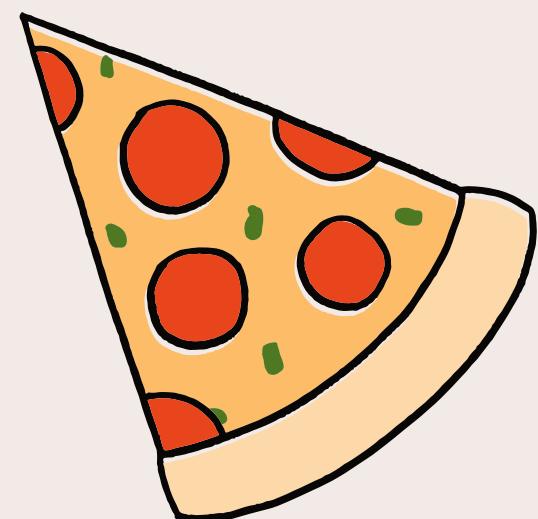


Calculate the percentage contribution of each pizza type to total revenue.

```
WITH RevenueCTE AS (
    SELECT
        pi.category,
        SUM(o.quantity * p.price) AS revenue
    FROM
        pizza_types pi
    JOIN pizzas p
        ON p.pizza_type_id = pi.pizza_type_id
    JOIN order_details o
        ON p.pizza_id = o.pizza_id
    GROUP BY
        pi.category
)SELECT
    category,
    ROUND((revenue / (SELECT SUM(revenue) FROM RevenueCTE)) * 100, 2) AS percentage_contribution
FROM RevenueCTE
ORDER BY revenue DESC;
```

	category	percentage_contribution
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

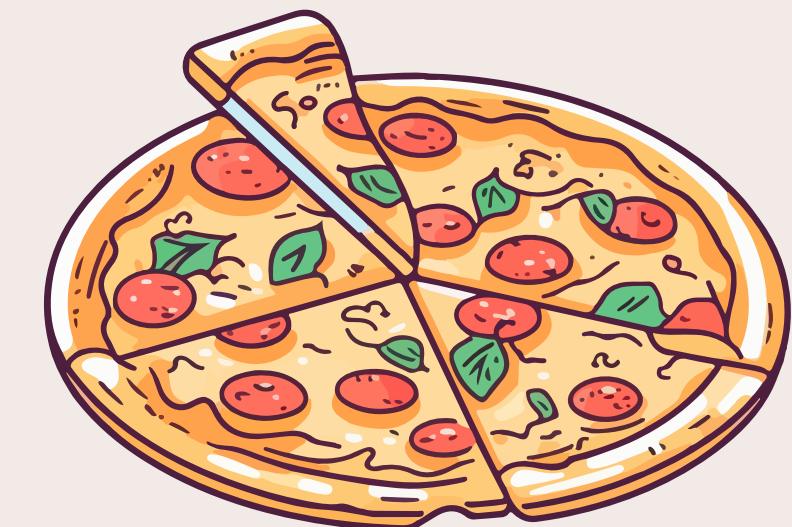




Analyze the cumulative revenue generated over time.

```
Select order_dATE, Round(sum(revenue) over (order by order_dATE), 2) as cum_revenue  
(Select t.order_date, ROUND(SUM(o.quantity * p.price)), 2) AS revenue  
from order_details o  
Join table_orders t  
on t.order_id=o.order_id  
join pizzas p
```

	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





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

```
With CTE2 as
(With CTE1 as
(Select pi.name, pi.category ,
Sum( o.quantity*p.price) as revenue
from pizza_types pi
Join pizzas p on p.pizza_type_id=pi.pizza_type_id
Join order_details o
on o.pizza_id=p.pizza_id
Group by pi.name, pi.category)Select name, category , revenue ,
dense_rank() over (partition by category Order By revenue desc) as rn from CTE1)
Select category ,name , revenue from CTE2 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





Thank you for reviewing my project. I welcome any feedback and would be happy to make improvements.