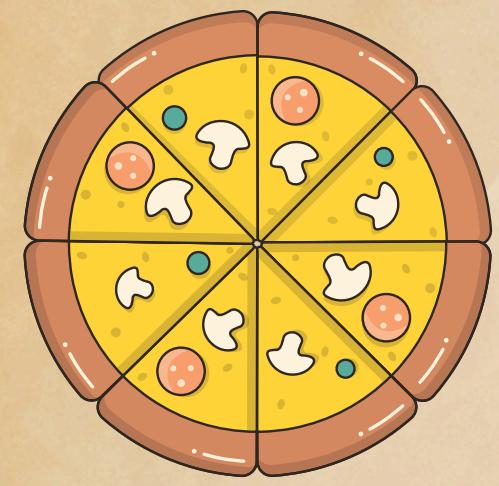


# Pizza Sales Analysis with SQL



By Rahul Sewate



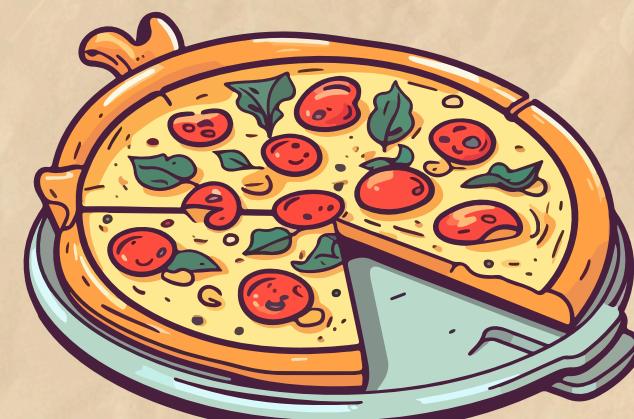
**In this project, I have utilized SQL queries to solve  
questions related to pizza sales."**



# Q1 – Retrieve the total number of orders placed.

```
select count(order_id) as total_orders from orders
```

	total_ord...
1	21350



# Q2 – Calculate the total revenue generated from pizza sales.



```
select
    round(sum(order_details.quantity * pizzas.price),2) as total_sales
from
    order_details join pizzas
on
    pizzas.pizza_id = order_details.pizza_id
```

A colorful illustration of a whole pizza with a golden-yellow crust. The pizza is topped with many red pepperoni slices, green basil leaves, and small green herbs. A slice is being lifted from the bottom left corner. The background is a light beige color.

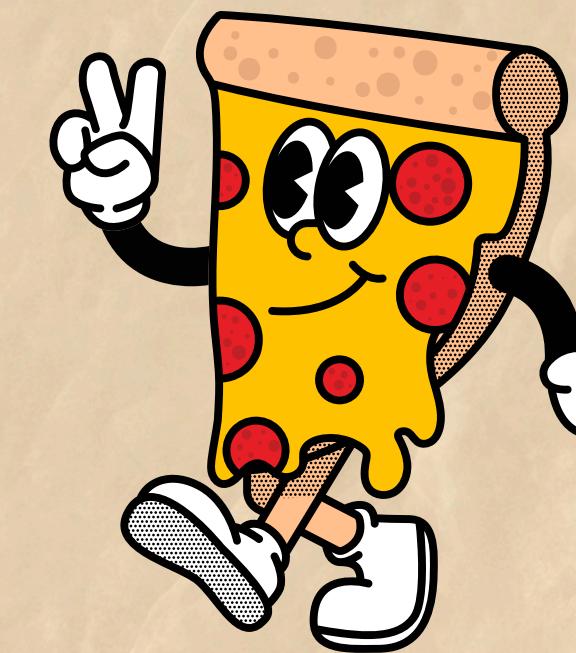
	total_sales
1	275473.15



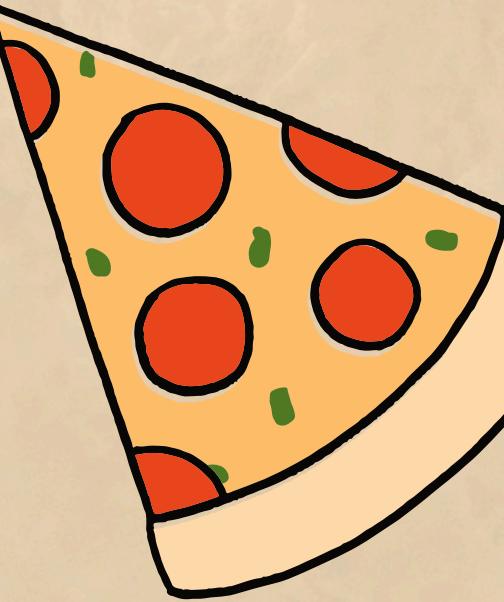
# Q3 - Identify the highest-priced pizza.

```
select top 1
pizza_types.name, pizzas.price
from
pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc
```

	name	price
1	The Greek Pizza	35.95



# Q4 – 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;
```

Results    Messages

	size	order_co...
1	L	6254
2	M	5167
3	S	4758
4	XL	192
5	XXL	12

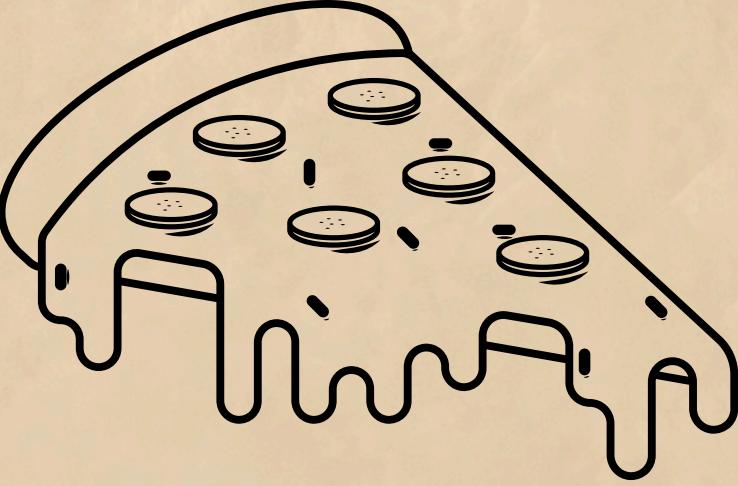


# Q5 – List the top 5 most ordered pizza types along with their quantities.

```
select top 5
    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 order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by quantity desc
```

	name	quantity
1	The Barbecue Chicken Pizza	853
2	The Hawaiian Pizza	824
3	The Pepperoni Pizza	805
4	The Classic Deluxe Pizza	788
5	The Thai Chicken Pizza	786

# Q6 – 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
    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
ORDER BY
    quantity DESC;
```



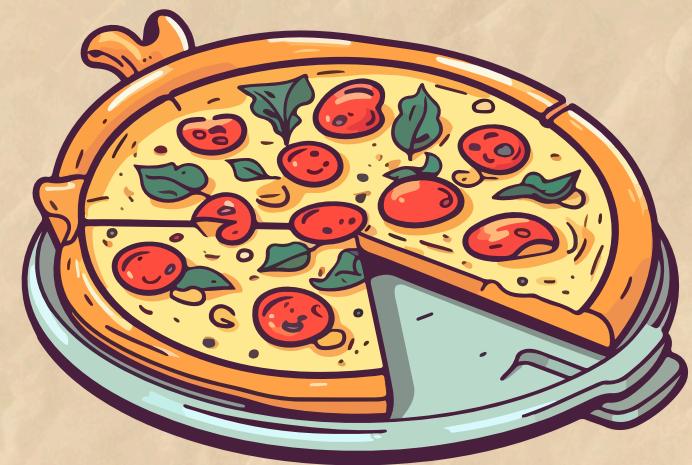
	category	quant...
1	Classic	4943
2	Supreme	4028
3	Veggie	3988
4	Chicken	3729



## Q7 – Determine the distribution of orders by hour of the day.

```
SELECT DATEPART(hour, time)as hour, COUNT(order_id) as order_count
FROM orders
GROUP BY DATEPART(hour, time)
ORDER BY order_count desc;
```

	hour	order_count
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009
6	16	1920
7	20	1642
8	14	1472
9	15	1468
10	11	1231
11	21	1198
12	22	663
13	23	28
14	10	8
15	9	1



## Q8 – Join relevant tables to find the category-wise distribution of pizzas.



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

	category	(No column name)
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



# Q9 – Group the orders by date and calculate the average number of pizzas ordered per day.

```
select round(avg(quantity),0) as avg_pizza_ord_perDay
  from
  (select orders.date, sum(order_details.quantity) as quantity
from
  orders join order_details
  on orders.order_id = order_details.order_id
  group by orders.date) as OQuantity;
```

	avg_pizza_ord_perDay
1	138





# Q10 – Determine the top 3 most ordered pizza types based on revenue.

```
select top 3 pizza_types.name,  
sum(order_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name order by revenue desc
```

	name	revenue
1	The Barbecue Chicken Pizza	15087.75
2	The Thai Chicken Pizza	14313.5
3	The California Chicken Pizza	13666.5





## Q11 – Calculate the percentage contribution of each pizza type to total revenue.

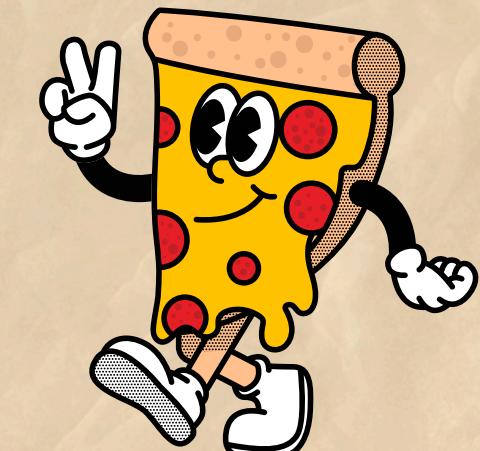
```
select pizza_types.category,  
    round(sum(order_details.quantity*pizzas.price) / (select  
        ROUND (SUM(order_details.quantity * pizzas.price),2)  
    AS total_sales  
from  
    order_details  
    join  
    pizzas ON pizzas.pizza_id= order_details.pizza_id) *100,2) as revenue  
from  
    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 order by revenue desc;
```

	category	reven...
1	Classic	26.56
2	Supreme	25.37
3	Veggie	24.05
4	Chicken	24.02

# Q12 – Analyze the cumulative revenue generated over time.

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

id	date	revenue
11	2015-01-11 00:00:00.000	20002.00
12	2015-01-12 00:00:00.000	27781.7
13	2015-01-13 00:00:00.000	29831.3
14	2015-01-14 00:00:00.000	32358.7
15	2015-01-15 00:00:00.000	34343.5
16	2015-01-16 00:00:00.000	36937.65
17	2015-01-17 00:00:00.000	39001.75
18	2015-01-18 00:00:00.000	40978.6
19	2015-01-19 00:00:00.000	43365.75
20	2015-01-20 00:00:00.000	45763.65
21	2015-01-21 00:00:00.000	47804.2

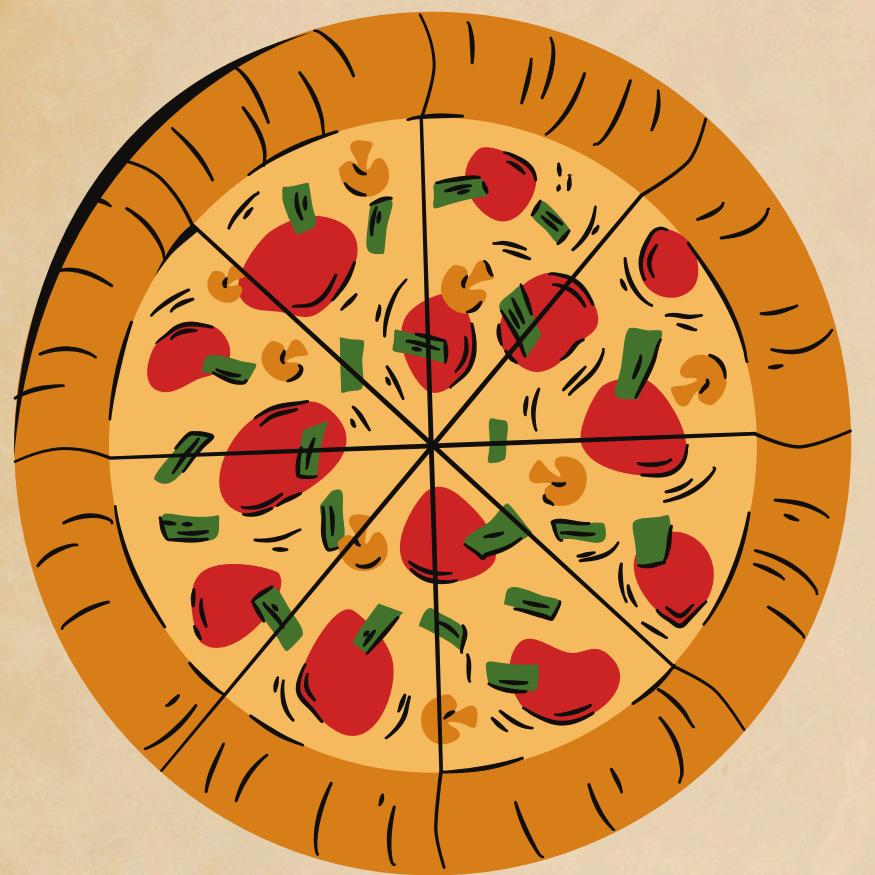


# Q13 – Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
SELECT 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,
           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.category, pizza_types.name
  ) AS a
) AS b
WHERE rn <= 3;
```

	name	revenue
1	The Barbecue Chicken Pizza	15087.75
2	The Thai Chicken Pizza	14313.5
3	The California Chicken Pizza	13666.5
4	The Classic Deluxe Pizza	12242
5	The Hawaiian Pizza	10886
6	The Pepperoni Pizza	10035
7	The Spicy Italian Pizza	11418
8	The Italian Supreme Pizza	11320.75
9	The Sicilian Pizza	10453.75
10	The Four Cheese Pizza	11015.95
11	The Mexicana Pizza	9078.75
12	The Five Cheese Pizza	9009.5





# THANK YOU

