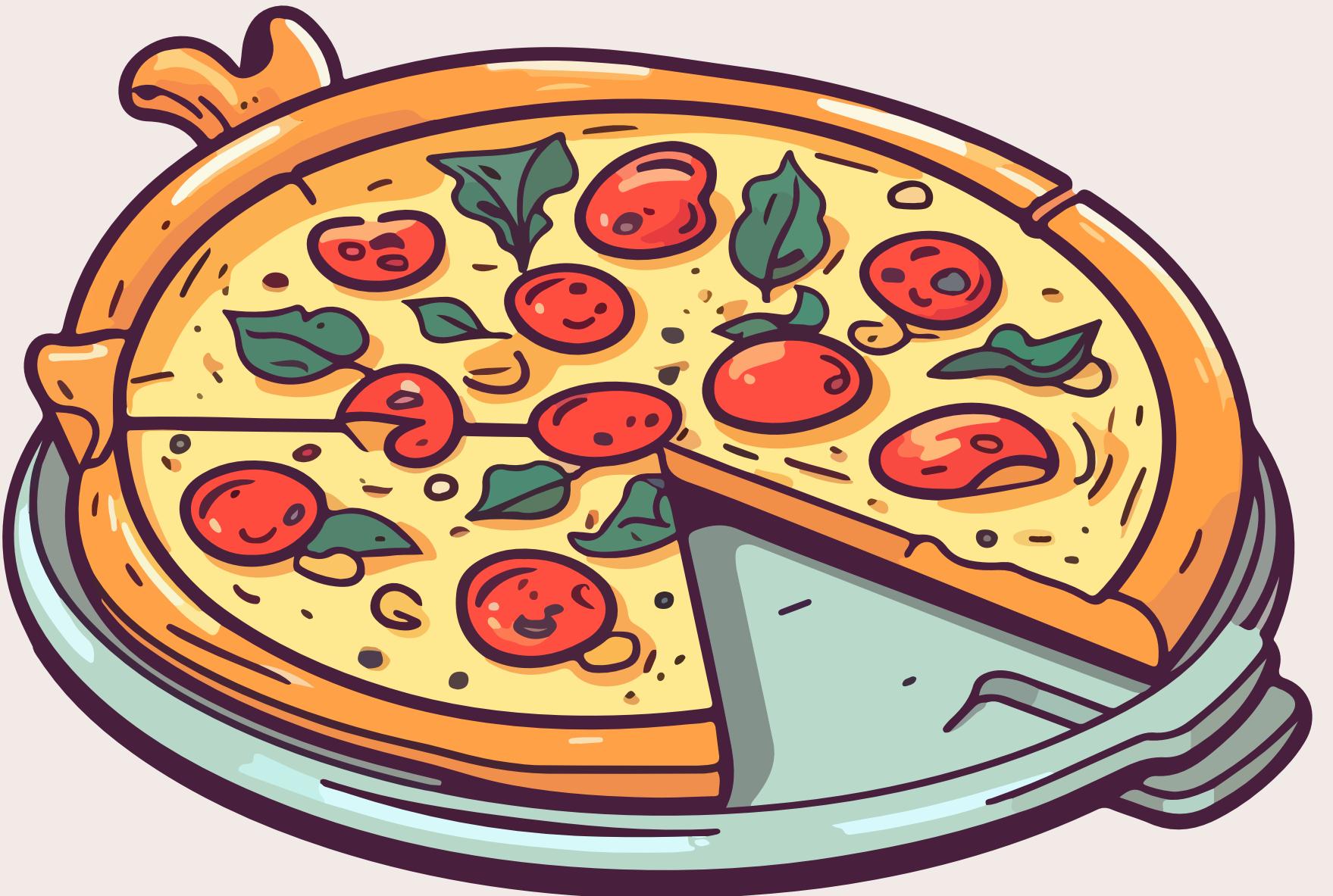
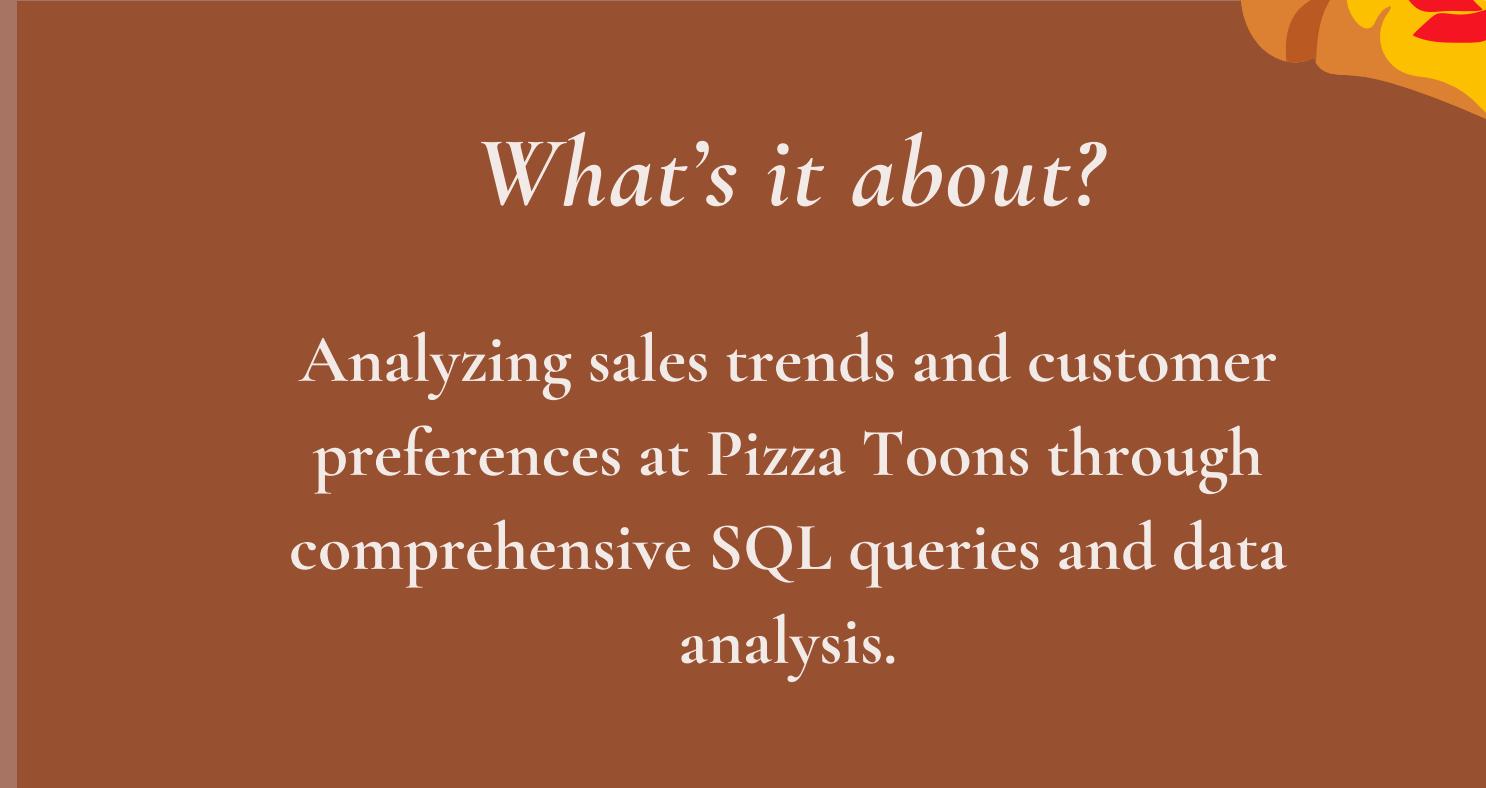


# Pizza Toons Sales

*An SQL project*



# Hi!!! Welcome to Pizza Toons Sales Project.



*What's it about?*

Analyzing sales trends and customer preferences at Pizza Toons through comprehensive SQL queries and data analysis.

*What are we going to get?*

This project aims to deliver actionable insights into sales performance, customer behavior, and product popularity to optimize business strategies at Pizza Toons.

# Key questions guiding this project include:

## *Basic :*

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

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

## *Intermediate :*

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

Determine the distribution of orders by hour of the day.

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

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

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

## *Advance:*

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

Analyze the cumulative revenue generated over time.

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



# Let's Get Started...





Basic :

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

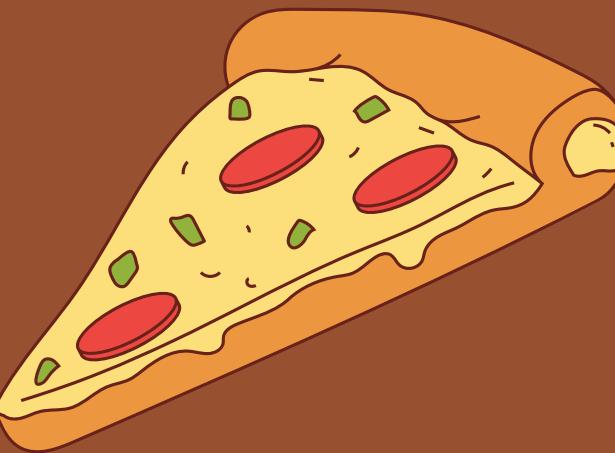
```
• SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

Result Grid

	total_orders
▶	21350



- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



```
SELECT  
    SUM(orders_details.quantity * pizzas.price) total_revenue  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.0499999993



- IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



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

Result Grid | Filter R

	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(orders_details.quantity) AS most_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY most_ordered DESC
LIMIT 5;
```



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

- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



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

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



# Intermediate:

- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

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



A decorative image of a pizza slice with various toppings like pepperoni, olives, and cheese.

	category	total_quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(order_time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

	HOUR(order_time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468



- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



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

Result Grid | Filter Rows

	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
```

```
    AVG(quantity)
```

```
FROM
```

```
(SELECT
```

```
    orders.order_date, SUM(orders_details.quantity) AS quantity
```

```
FROM
```

```
    orders
```

```
JOIN orders_details ON orders_details.order_id = orders.order_id
```

```
GROUP BY orders.order_date) AS order_per_day;
```



Result Grid	
	AVG(quantity)
▶	138.4749

- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



```
SELECT
    pizza_types.name,
    SUM(pizzas.price * orders_details.quantity) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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



# Advanced :

- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category,
    (SUM(orders_details.quantity * pizzas.price) / (SELECT
        SUM(orders_details.quantity * pizzas.price)
    FROM
        orders_details
    JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100 AS revenue
FROM
    pizza_types
    JOIN
        pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
        orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```



Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.905960255669903
	Supreme	25.45631126009884
	Chicken	23.955137556847493
	Veggie	23.682590927384783

- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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



	order_date	cumulative_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	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05

- 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)
as ranking
from
(select pizza_types.category, pizza_types.name,
round(sum(orders_details.quantity*pizzas.price), 2) as revenue
from pizza_types
join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id
join orders_details on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name
) as b) as a
where ranking <= 3;
    
```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap

	category	name	revenue
▶	Chicken	The Chicken Pesto Pizza	16701.75
	Chicken	The Chicken Alfredo Pizza	16900.25
	Chicken	The Southwest Chicken Pizza	34705.75
	Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
	Classic	The Big Meat Pizza	22968
	Classic	The Napolitana Pizza	24087
	Supreme	The Brie Carre Pizza	11588.5
	Supreme	The Spinach Supreme Pizza	15277.75

Result 14

THANK YOU

