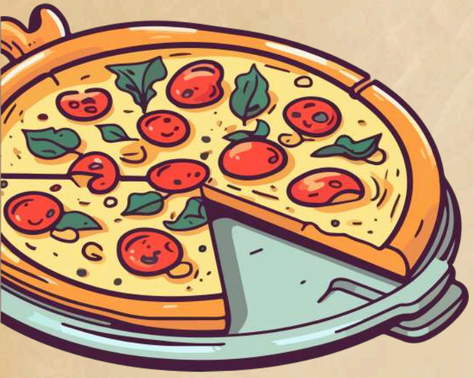
PIZZA SALES ANALYSIS

Using SQL

By : Mayank Yadav

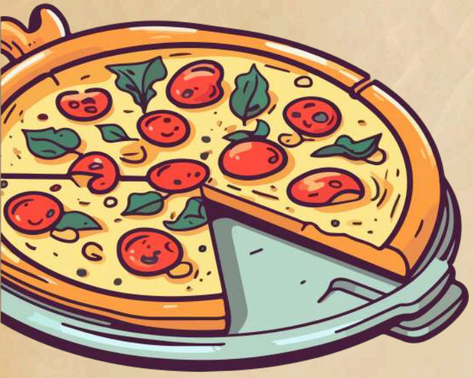


Hello Everyone!

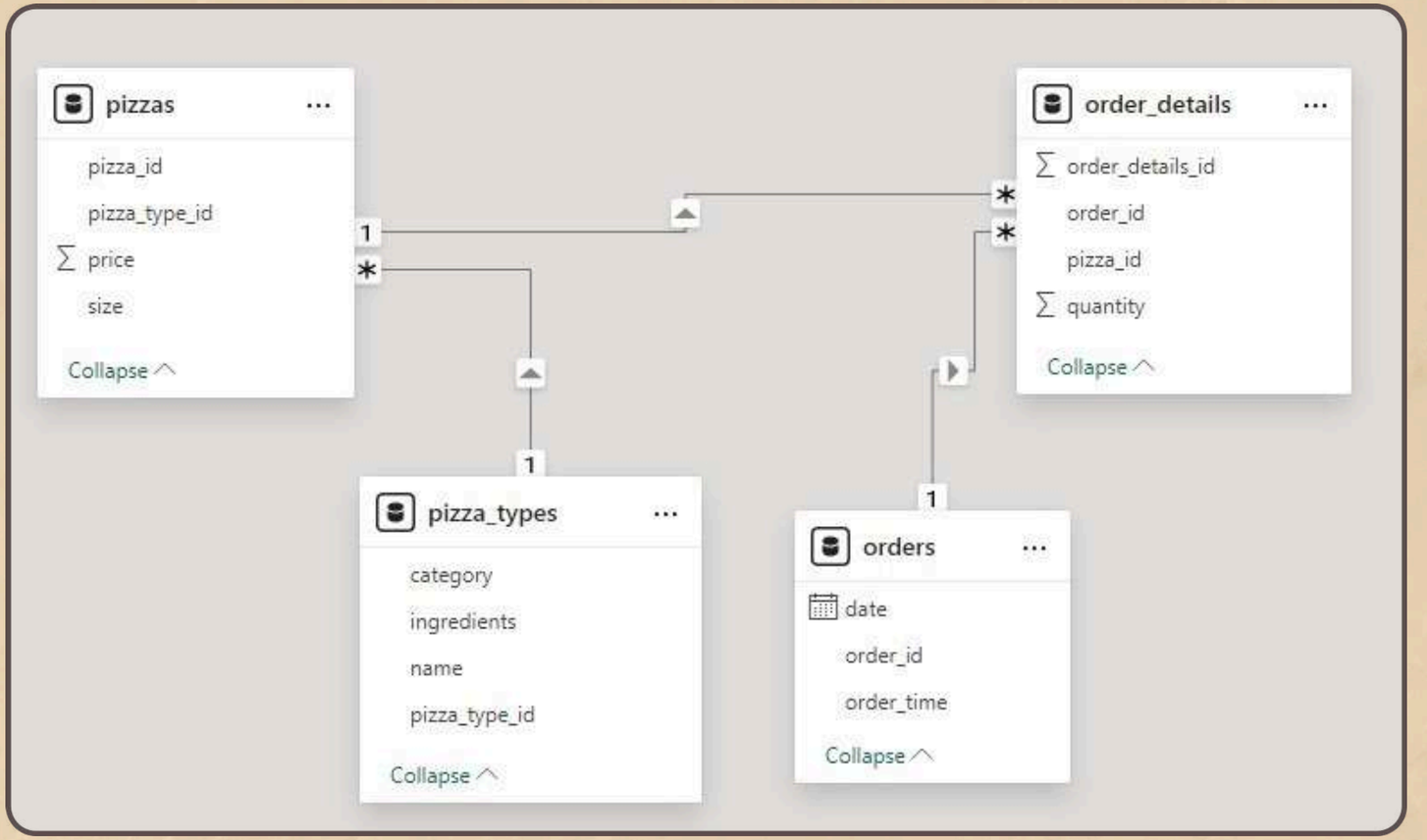
I embarked on an exciting SQL journey with a project focused on analyzing pizza sales data using **Microsoft SQL Server Management Studio**. This project dives deep into four comprehensive datasets, each revealing crucial insights about pizza sales performance. By addressing **13 strategic business questions**, I uncovered key trends related to:

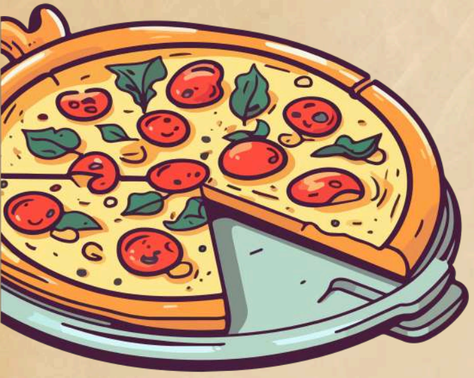
- Sales growth and patterns over time
- Customer preferences and popular choices
- Operational efficiency in delivering customer satisfaction

These insights helped shape a clearer understanding of overall business performance, customer behaviors, and future growth opportunities in the pizza industry.



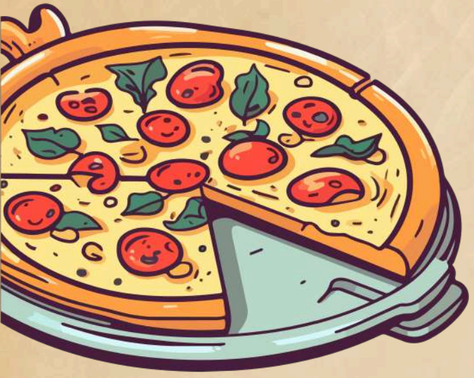
Schema


pizzas
pizza_id
pizza_type_id
price
size
Collapse
order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
orders
date
order_id
order_time
Collapse
1
*
1
*
*
1



Questions

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.
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.
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.

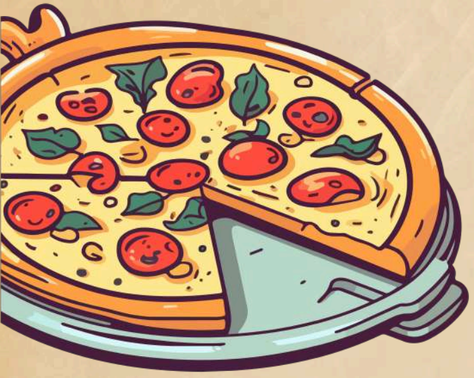


1. Retrieve the total number of orders placed.

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

Results | Messages

	Total_orders
1	21350

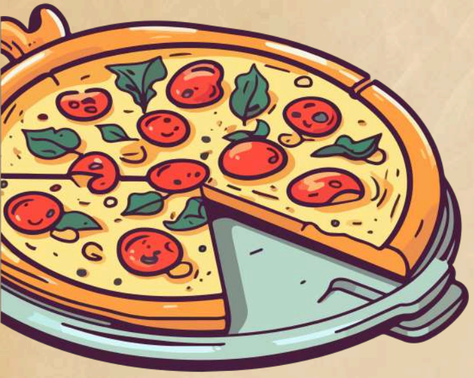


2. Calculate the total revenue generated from pizza sales.

```
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
```

Results | Messages

	Total_sales
1	817860.05

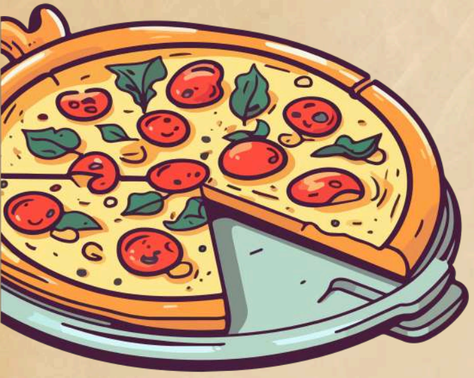


3. Identify the highest-priced pizza

```sql
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 price DESC
```

Results | Messages

	name	price
1	The Greek Pizza	35.95

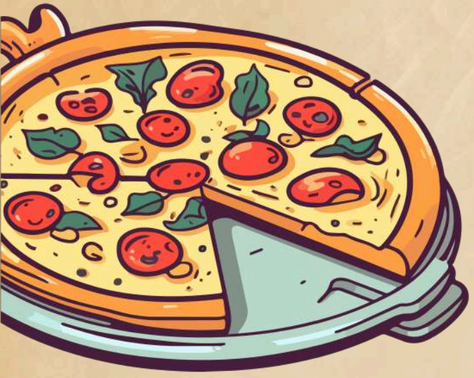


4. Identify the most common pizza size ordered.

```
SELECT
    TOP 1
        COUNT(order_details.order_details_id) AS No_of_orders,pizzas.size
FROM order_details
    JOIN pizzas
        ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
```

Results | Messages

	No_of_orders	size
1	18526	L

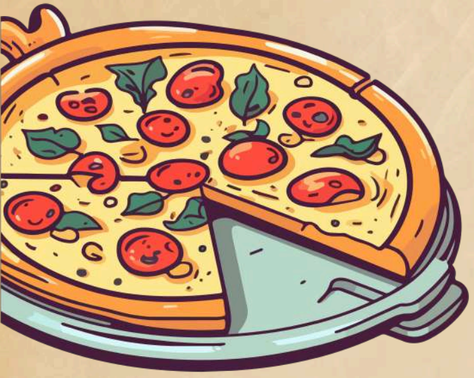


5. List the top 5 most ordered pizza types along with their quantities.

```sql
SELECT
    TOP 5 pizza_types.name,
        SUM(order_details.quantity) AS Total_quantity_ordered
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 Total_quantity_ordered DESC
```

Results | Messages

	name	Total_quantity_ordered
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

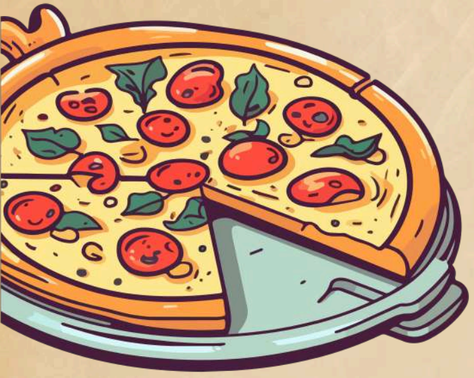


6. Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT pizza_types.category,
    SUM(order_details.quantity) AS ToTal_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 ToTal_quantity DESC
```

Results | Messages

	category	ToTal_quantity
1	Classic	13529
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050
5	Mushroom	1359

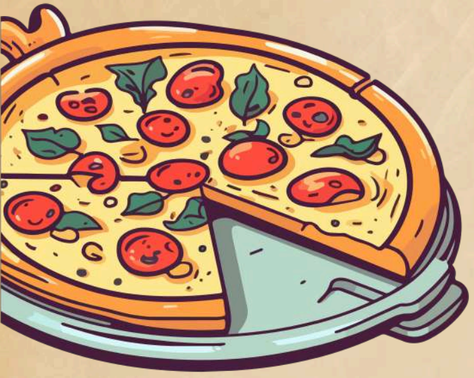


7. Determine the distribution of orders by hour of the day.

```sql
SELECT
    DATEPART(HOUR, order_time) AS Hour,
        COUNT(order_id) AS no_of_orders
FROM orders
GROUP BY DATEPART(HOUR, order_time)
ORDER BY Hour DESC
```

Results | Messages

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

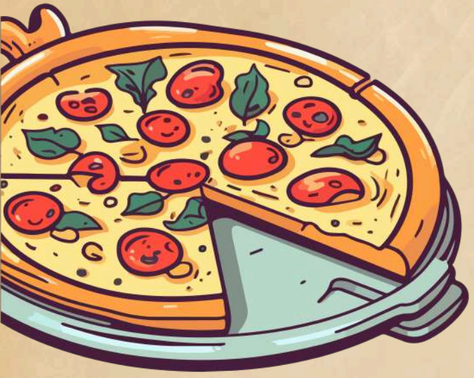


8. Join relevant tables to find the category-wise distribution of pizzas.

```sql
SELECT category,
    COUNT(name) AS No_of_pizzas
FROM pizza_types
GROUP BY category
ORDER BY No_of_pizzas DESC
```

Results | Messages

	category	No_of_pizzas
1	Supreme	9
2	Veggie	9
3	Classic	7
4	Chicken	6
5	Mushroom	1

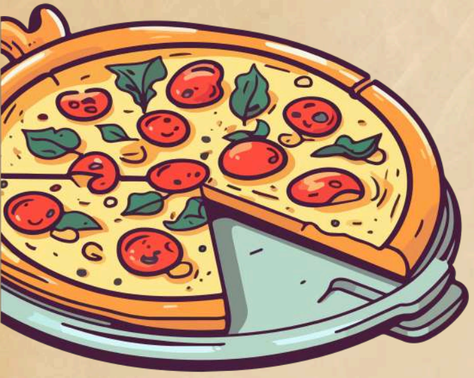


9. Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    AVG(Total_quantity) AS Avg_pizza_ordered_per_day
FROM
    (SELECT orders.date,
        SUM(order_details.quantity) AS Total_quantity
    FROM orders
        JOIN order_details
            ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS Ordered_quantity
```

Results | Messages

	Avg_pizza_ordered_per_day
1	138

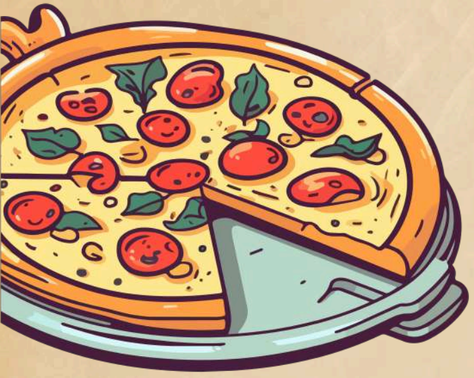


10. Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    TOP 3 pizza_types.name AS Pizza_Type,
        ROUND(
            SUM(order_details.quantity*pizzas.price),2) AS Total_revenue
FROM pizza_types
    JOIN pizzas
        ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details
        ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_revenue DESC
```

Results | Messages | Client Statistics

	Pizza_Type	Total_revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

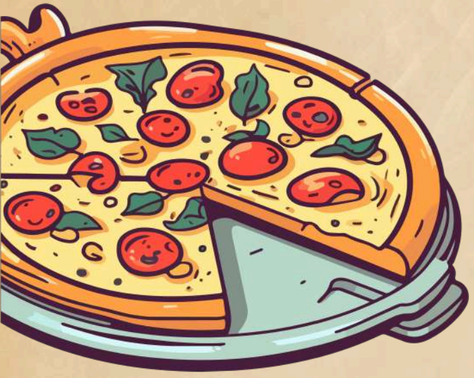


11. Calculate the percentage contribution of each pizza type to total revenue.

```
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 pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details
        ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue
```

Results | Messages

	category	revenue
1	Mushroom	2.3
2	Veggie	23.68
3	Chicken	23.96
4	Classic	24.6
5	Supreme	25.46

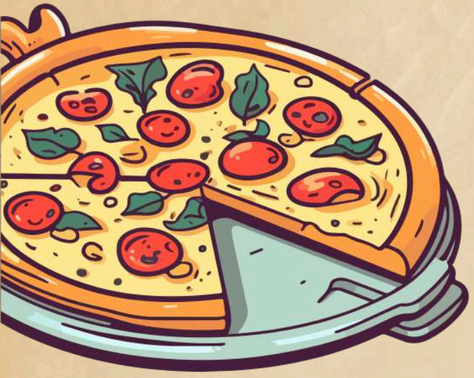


12. Analyze the cumulative revenue generated over time.

```sql
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 order_details.order_id = orders.order_id
    GROUP BY orders.date) AS sales
```

Results | Messages | C

	date	Cum_revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.35
11	2015-01-11	25862.65
12	2015-01-12	27781.7
13	2015-01-13	29831.3
14	2015-01-14	32358.7
15	2015-01-15	34343.5
16	2015-01-16	36937.65
17	2015-01-17	39001.75
18	2015-01-18	40978.6
19	2015-01-19	43365.75



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

```sql
SELECT name, sales,Rank
FROM
    (SELECT category, name, sales,
        RANK()
            OVER (PARTITION BY category ORDER BY sales DESC) AS Rank
    FROM
        (SELECT pizza_types.category, pizza_types.name,
            ROUND(
                SUM(order_details.quantity*pizzas.price),2)  AS sales
        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, pizza_types.name )
    AS A)
AS B
WHERE Rank<=3
```

Results | Messages | Client Statistics

	name	sales	Rank
1	*The Pepperoni	18834.5	1
2	The Thai Chicken Pizza	43434.25	1
3	The Barbecue Chicken Pizza	42768	2
4	The California Chicken Pizza	41409.5	3
5	The Classic Deluxe Pizza	38180.5	1
6	The Hawaiian Pizza	32273.25	2
7	The Pepperoni Pizza	30161.75	3
8	The Spicy Italian Pizza	34831.25	1
9	The Italian Supreme Pizza	33476.75	2
10	The Sicilian Pizza	30940.5	3
11	The Four Cheese Pizza	32265.7	1
12	The Mexicana Pizza	26780.75	2
13	The Five Cheese Pizza	26066.5	3

Thank You