

PIZZA Sales Analysis

SQL Project

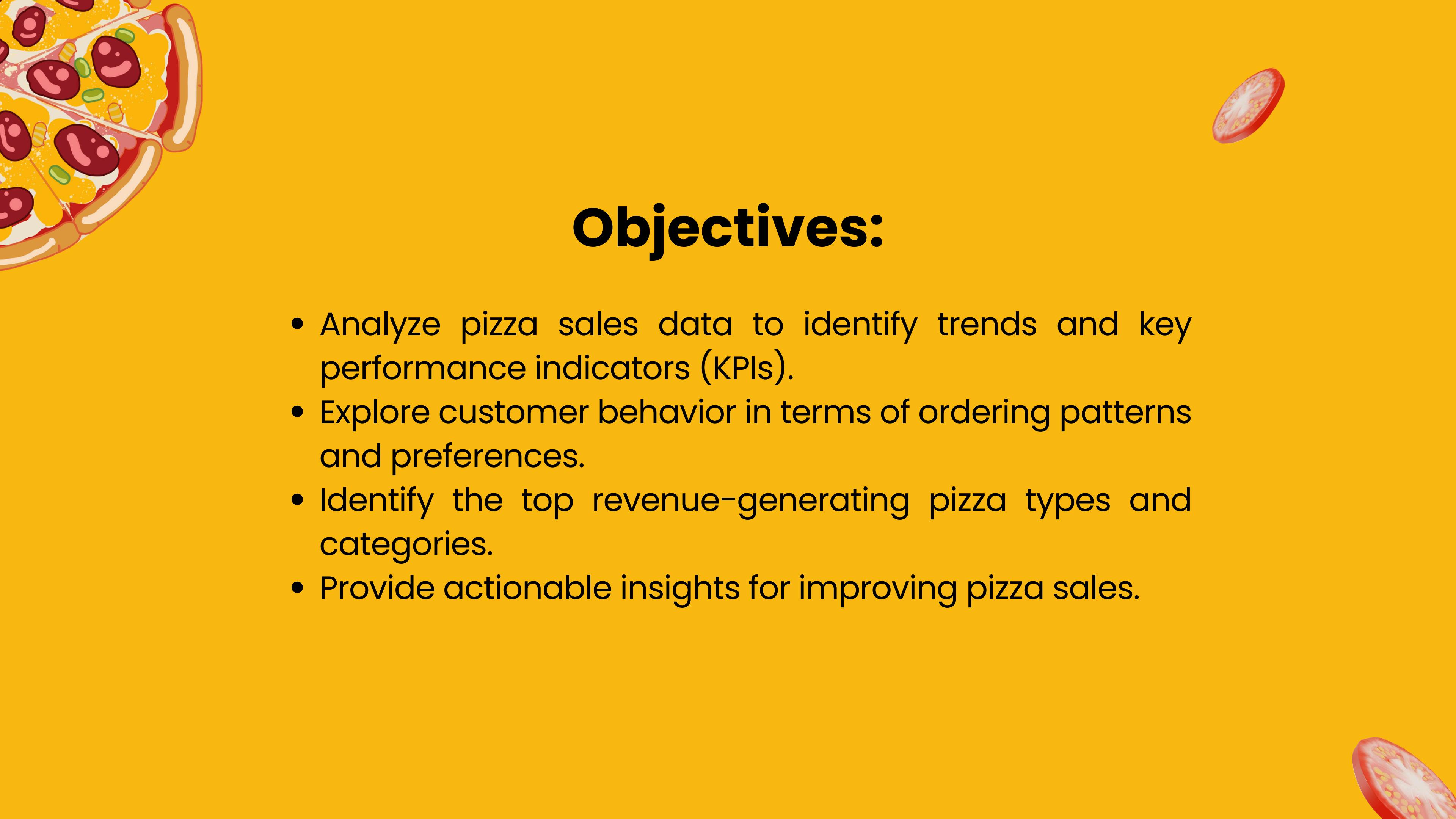
Presented By: **Vidhi Saxena**





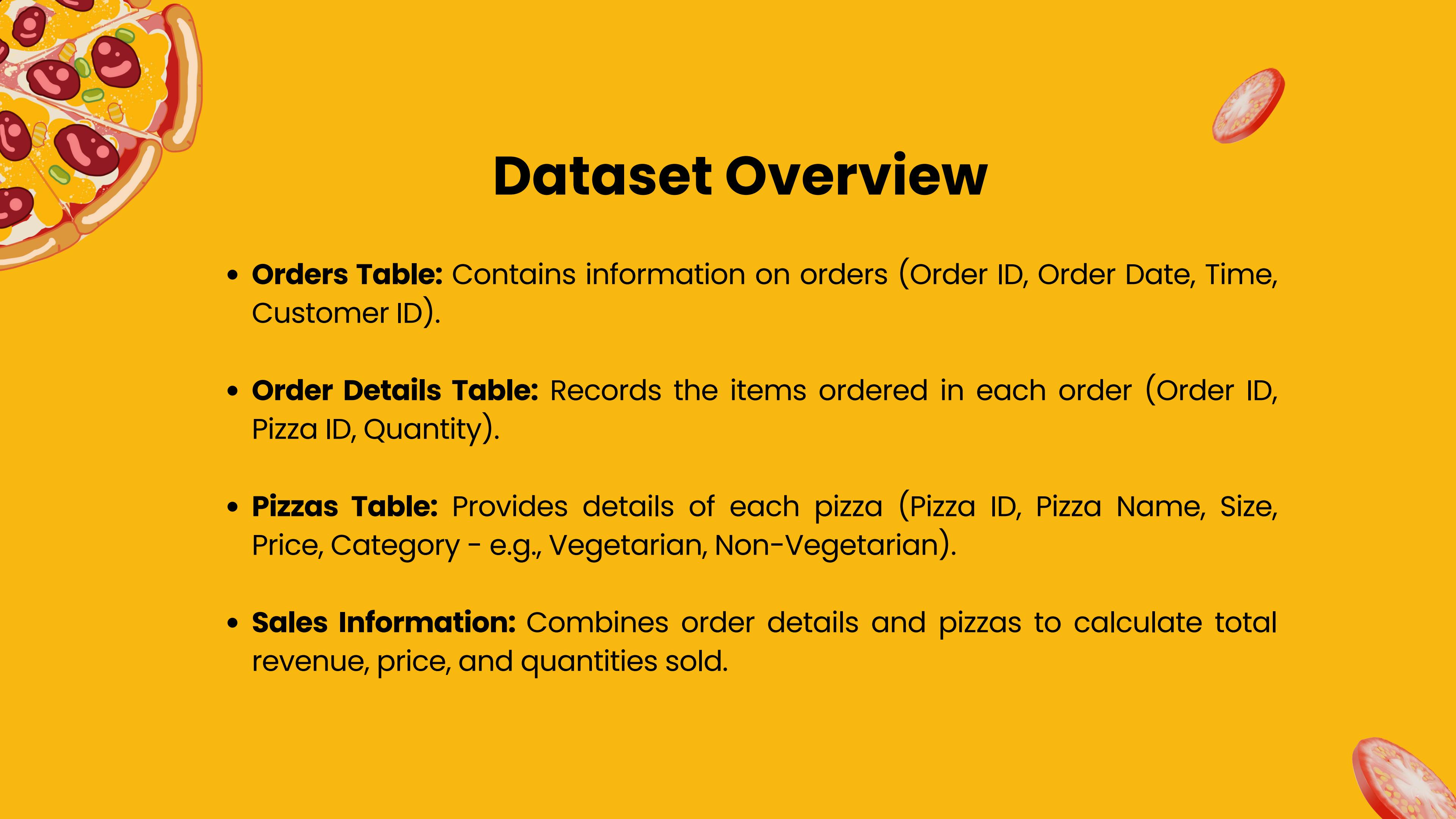
Project Overview

This data analysis project focuses on exploring and analyzing pizza sales data using SQL queries. The goal is to gain insights into the business performance, and customer preferences, and identify areas for potential growth. The analysis is divided into three levels: Basic, Intermediate, and Advanced.



Objectives:

- Analyze pizza sales data to identify trends and key performance indicators (KPIs).
- Explore customer behavior in terms of ordering patterns and preferences.
- Identify the top revenue-generating pizza types and categories.
- Provide actionable insights for improving pizza sales.



Dataset Overview

- **Orders Table:** Contains information on orders (Order ID, Order Date, Time, Customer ID).
- **Order Details Table:** Records the items ordered in each order (Order ID, Pizza ID, Quantity).
- **Pizzas Table:** Provides details of each pizza (Pizza ID, Pizza Name, Size, Price, Category - e.g., Vegetarian, Non-Vegetarian).
- **Sales Information:** Combines order details and pizzas to calculate total revenue, price, and quantities sold.

Recommended Analysis

→ Retrieve the total number of orders placed.

SQL Query:

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

Output :

total_orders
21350



→ Calculate the total revenue generated from pizza sales.

SQL Query:

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS total_sales
FROM
    orders_details
    JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Output:

total_sales
817860.05



→ Identify the highest-priced pizza.

SQL Query:

```
SELECT
    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
LIMIT 1;
```

Output :

name	price
The Greek Pizza	35.95



→ Identify the most common pizza size ordered.

SQL Query:

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

Output:

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.

SQL Query:

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

Output :



name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2317

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

SQL Query:

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

Output :

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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



Output :

SQL Query:

```
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.



Output :

SQL Query:

```
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.



SQL Query:

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_order_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Output :



avg_pizza_order_per_day
138



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



SQL Query:

```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) 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;
```

Output:



category	COUNT (name)
The Thai Chicken Pizza	6
The Barbecue Chicken Pizza	8
The California Chicken Pizza	9



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



SQL Query:

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

Output:



category	COUNT (name)
Chicken	26.91
Classic	25.46
Supreme	23.96
Veggie	23.68



Analyze the cumulative revenue generated over time.



SQL Query:

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



Output:

	order_date	cum_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
	2015-01-09	21526.4
	2015-01-10	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003
	2015-01-14	32358.70000000004
	2015-01-15	34343.50000000001
	2015-01-16	36937.65000000001



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



SQL Query:

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



Output:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.7000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Conclusion

This pizza sales data analysis provides insights into customer preferences, sales patterns, and revenue generation. Key findings include the most popular pizza sizes and types, sales performance by category, and high-revenue pizza varieties. These insights can help inform strategic decisions to optimize inventory, marketing, and customer engagement strategies.



Thank You!

