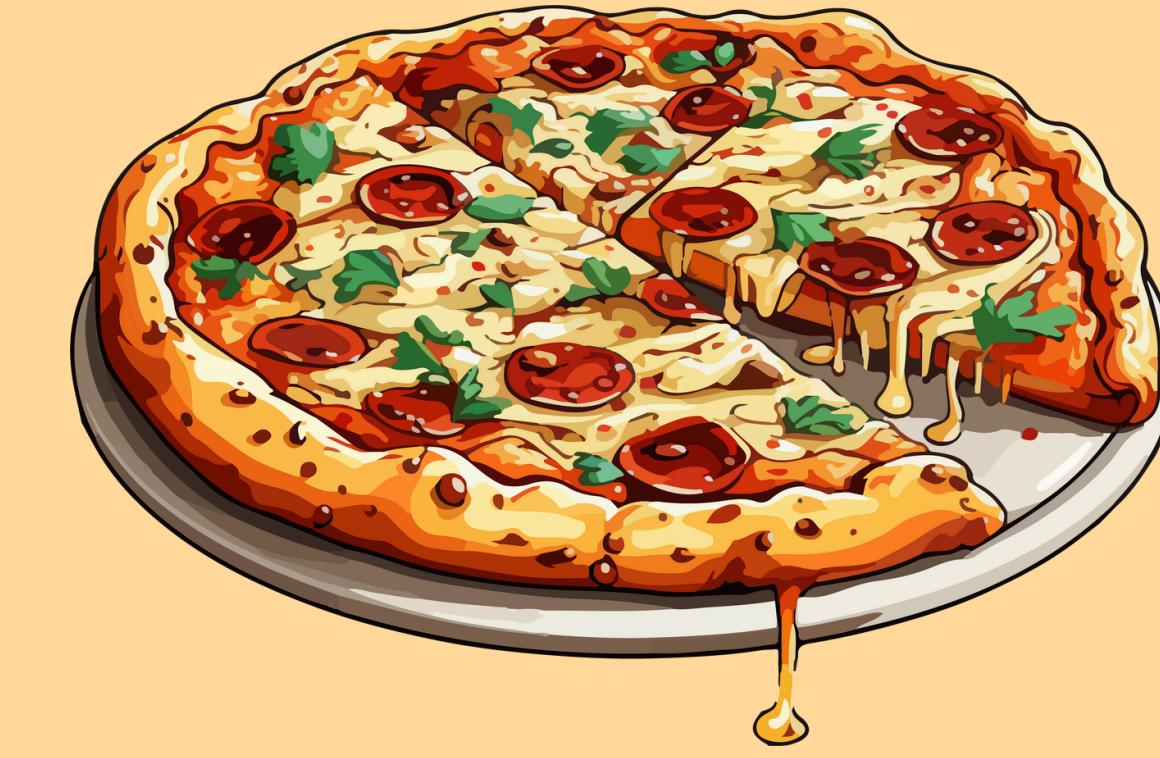




pizza sales analysis using SQL

PIZZA WORLD

ANALYSIS BY JATIN



OBJECTIVE

Analyzed pizza sales data to uncover key insights and trends within the pizza industry, leveraging SQL for data extraction, transformation, and analysis. The project aimed to enhance understanding of customer preferences, sales performance, and operational efficiencies to inform strategic decision-making.

Analysis Techniques

- Employed SQL for data cleaning, normalization, and transformation to ensure accuracy and consistency.
- Conducted exploratory data analysis (EDA) to identify sales trends, peak sales periods, and popular pizza varieties.
- Analyzed customer purchasing behavior to segment the market and tailor marketing efforts.
- Evaluated operational performance, including order processing times and delivery efficiency.

Q1 .Retrieve the total number of orders placed.

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

	count(order_id)
▶	21350

Q2 . Calculate the total revenue generated from pizza sales.

```
select  
round(sum(order_details.quantity * pizzas.price),0)  
as total_revenue  
from order_details join pizzas  
on pizzas.pizza_id = order_details.pizza_id
```

	total_revenue
▶	817860

Q3. Identify the highest-priced pizza.

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

	name	HIGHEST_PRICED_PIZZA
▶	The Greek Pizza	35.95

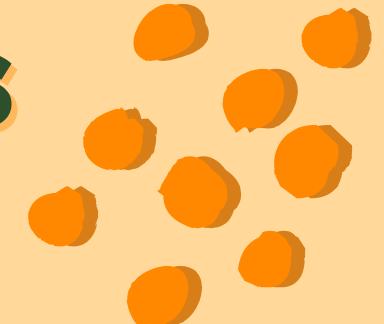
Q4 . Identify the most common pizza size ordered.

```
SELECT pizzas.size , COUNT(order_details.quantity)  
AS TOTAL_QUANTITY  
FROM pizzas JOIN order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size;
```

	size	TOTAL_QUANTITY
▶	M	15385
	L	18526
	S	14137
	XL	544
	XXL	28



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



```
select pizza_types.name ,  
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.name order by total_quantity  
desc limit 5 ;
```

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



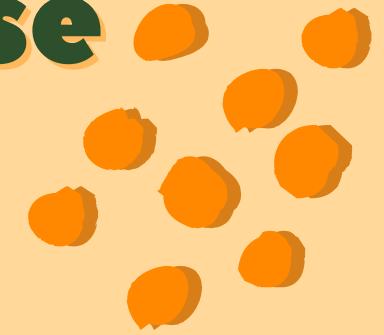
Q7.Determine the distribution of orders by hour of the day.

```
SSELECT  
    HOUR(ORDER_TIME) AS HOUR, COUNT(ORDER_ID) AS ORDER_COUNT  
FROM  
    ORDERS  
GROUP BY HOUR(ORDER_TIME);
```

	HOUR	ORDER_COUNT
	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



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



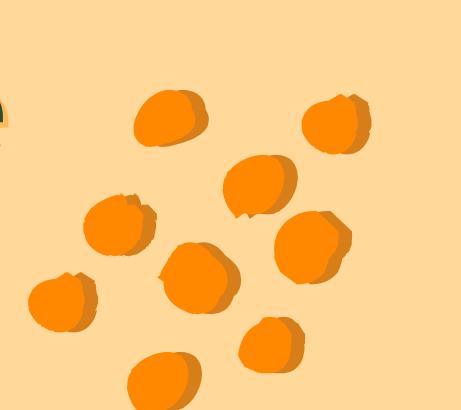
```
SELECT  
    category, COUNT(name) AS total_distribution  
FROM  
    pizza_types  
GROUP BY category
```

	category	total_distribution
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





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



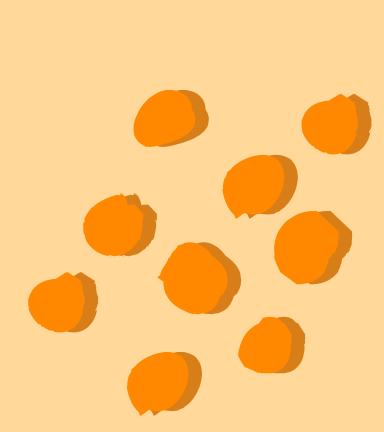
```
SELECT  
    ROUND(AVG(total_orders), 0) AS avg_order_per_day  
FROM  
(SELECT  
    orders.order_date, SUM(order_details.quantity) total_orders  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

	avg_order_per_day
▶	138





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



SELECT

 pizza_types.name,
 ROUND(SUM(order_details.quantity * pizzas.price),
 0) AS TOTAL_REVENUE

FROM

 pizza_types

 JOIN

 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

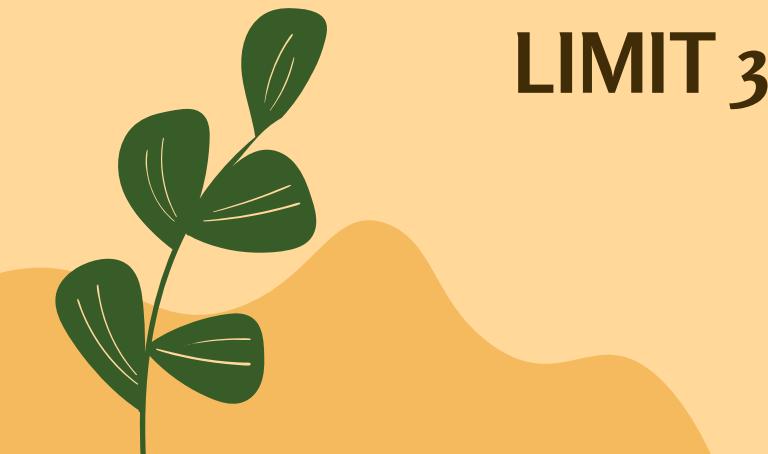
 JOIN

 order_details ON order_details.pizza_id = pizzas.pizza_id

GROUP BY pizza_types.name

ORDER BY TOTAL_REVENUE

LIMIT 3;



	name	TOTAL_REVENUE
▶	The Brie Carre Pizza	11588
	The Green Garden Pizza	13956
	The Spinach Supreme Pizza	15278



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

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * PRICE), 0) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) TOTAL_SALES
    )
FROM
    order_details
    JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100 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	REVENUE
▶	Classic	26.905948028638882
	Supreme	25.45631126009884
	Chicken	23.955198692001154
	Veggie	23.68253590574573

Outcome

The analysis provided actionable insights that can drive strategic decisions, improve customer experience, and increase overall sales performance within the pizza industry. The project demonstrates proficiency in **SQL** and data analysis, contributing valuable expertise to the field of data-driven decision-making.





pizza sales analysis by SQL

**THANK
YOU**

analysis by jatin