HELLO

THIS IS

Pizza Sales Analysis SQL Project

ABOUT ME

Hello, i’m kirti Lakhdharia
I’m driven by the thrill of uncovering insights hidden in data an transforming raw numbers into compelling stories that drive business success. Every query and dataset is a new puzzle, and I’m excited by the challenge of solving it.

ABOUT PROJECT

- Purpose: To analyze pizza sales data to uncover trends,optimize inventor management, and enhance marketing strategies.
- Goal: Improve sales performance and customer satisfaction through data-driven insights.
- Data Sources: Sales transactions,customer demographics, and seasonal trends.
- Key Metrics: Total sales, sales by pizza type, peak sales times, customer purchase patterns.

METHODOLOGY

- Data Collection: Aggregated sales data from point-of-sale systems.
- Data Analysis: Utilized SQL queries to extract and aggregate data, identified key trends and patterns.
- Techniques: Implemented statistical analysis and visualization techniques to interpret data effectively.

KEY FINDINGS:

- Sales Trends: Identified peak sales periods and popular pizza types.
- Customer Insights: Discovered preferences based on demographics and purchasing behavior.
- Operational Efficiency: Highlighted inventory needs and areas for potential cost savings.

IMPACT:

- Business Decisions: Provided actionable insights for marketing campaigns and menu adjustments.
- Recommendations: Suggested inventory changes and promotional strategies to boost sales and customer satisfaction.

-- Retrive the total number of orders placed.

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

Result Grid

	total_orders
▶	21350

-- CALCULATED THE TOTAL REVENUE GENERATED FROM PIZZA SALES:

```
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;
```

total_sales
817860.05

-- find the highest price of the pizza

```
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;
```

Result Grid | Filter Ro

	name	price
▶	The Greek Pizza	35.95

-- Identify the most common pizza size ordered..

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

Result Grid | Filter Rows:

size	count(order_details_id)
M	15385
L	18526
S	14137
XL	544
XXL	28

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

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

Result Grid | Filter

pizza_type_id	count
classic_dlx	2416
bbq_ckn	2372
hawaiian	2370
pepperoni	2369
thai_ckn	2315

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

```
SELECT pizza_types.category,
COUNT(orders_details.quantity) AS count 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;
```

Result Grid

category	count
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

-- 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)
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.

```sql
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.

```
SELECT ROUND(AVG(quantity), 0) 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;
```

Result Grid | Filter R

round(avg(quantity),0)
138

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

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

Result Grid | Filter Rows:

name	revenu
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.25

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

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

Result Grid

category	revenu
Classic	27
Supreme	25
Veggie	24
Chicken	24

-- Analyze the cumulative revenue generated over time.

```sql
select order_date,sum(revenu) over(order by order_date) as cum_revenu from
(select orders.order_date,sum(pizzas.price * orders_details.quantity) as revenu 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;
```

Result Grid Filter Rows:

order_date	cum_revenu
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

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

Result Grid Filter Rows: Ex

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	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

THANK YOU