Pizza Sales Analysis

Shubham kumar

JULY 2024

Introduction

In this project, I am leveraging SQL to analyze sales data for a pizza-selling company. The primary objective is to gain insights into various aspects of the business, including sales trends, customer preferences, and operational efficiency. By querying the sales database, I aim to uncover patterns that can inform strategic decisions to enhance profitability and customer satisfaction.

1. Sales Performance: Evaluating overall sales metrics and identifying peak sales periods.

2.Customer Behavior: Analyzing order patterns to understand customer preferences and purchasing habits.

3.Product Analysis: Assessing the performance of different pizza types and toppings to determine best-sellers.

This project will provide valuable insights that can help the company optimize its menu, improve customer service, and drive growth.

About business

The pizza business is a lively and competitive part of the food industry. Everyone loves pizza, making it a favorite in many homes worldwide. This business includes small local pizza shops and big chains, all working hard to attract customers with delicious options and great service. Pizzerias offer a variety of pizzas, from classic ones like Margherita and Pepperoni to new choices like vegan and gluten-free pizzas.

Success in the pizza business depends on running things smoothly, making customers happy, and keeping up with new trends. Smooth operations mean taking orders quickly, managing supplies well, and delivering pizzas on time. Making customers happy with tasty ingredients, customizable pizzas, and excellent service helps build loyalty. It's also important to follow market trends, like online ordering and contactless delivery. By looking at sales data, businesses can make better decisions, improve their menu and services, and serve their customers better. This project uses SQL to find useful information that helps the pizza business grow and succeed.

Business update 1

Retrieve the total number of orders placed.

```sql
select count(*) as total_orders from orders;
```

Result Grid

	total_orders
▶	21350

Business update 2

Calculate the total revenue generated from pizza sales.

```
select sum(order_details.quantity*pizzas.price) as Total_Revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	Total_Revenue
▶	817860.04999993

Business update 3

Identify the highest-priced pizza.

```
select pizzas.price, pizza_types.name
from pizzas join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc
limit 1;
```

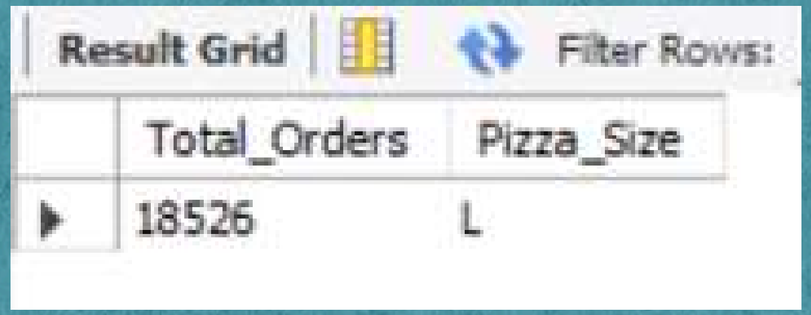
Result Grid | Filter Ro

	price	name
▶	35.95	The Greek Pizza

Business update 4

Identify the most common pizza size ordered.

```sql
select count(order_details.order_details_id) as Total_Orders, pizzas.size as Pizza_Size
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by Total_Orders desc
limit 1;
```



Business update 5

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

```
select pizza_types.name as Pizza_Name,sum(order_details.quantity) as Total_Quantity_Ordered,
count(order_details.order_details_id) as Number_Of_Orders
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 Number_Of_Orders desc
limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

Pizza_Name	Total_Quantity_Ordered	Number_Of_Orders
The Classic Deluxe Pizza	2453	2416
The Barbecue Chicken Pizza	2432	2372
The Hawaiian Pizza	2422	2370
The Pepperoni Pizza	2418	2369
The Thai Chicken Pizza	2371	2315

Business update 6

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

```
select pizza_types.category as Pizza_category, sum(order_details.quantity) as Total_Quantity
from pizzas
join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by Pizza_Category
```

Result Grid | Filter Rows:

Pizza_category	Total_Quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Business update 7

Determine the distribution of orders by hour of the day.

```
select hour(orders.order_time)
as Hour_Of_Day,
count(order_details.order_details_id) as Total_Orders
from orders
join order_details
on orders.order_id = order_details.order_id
group by Hour_Of_Day
order by Total_Orders desc;
```

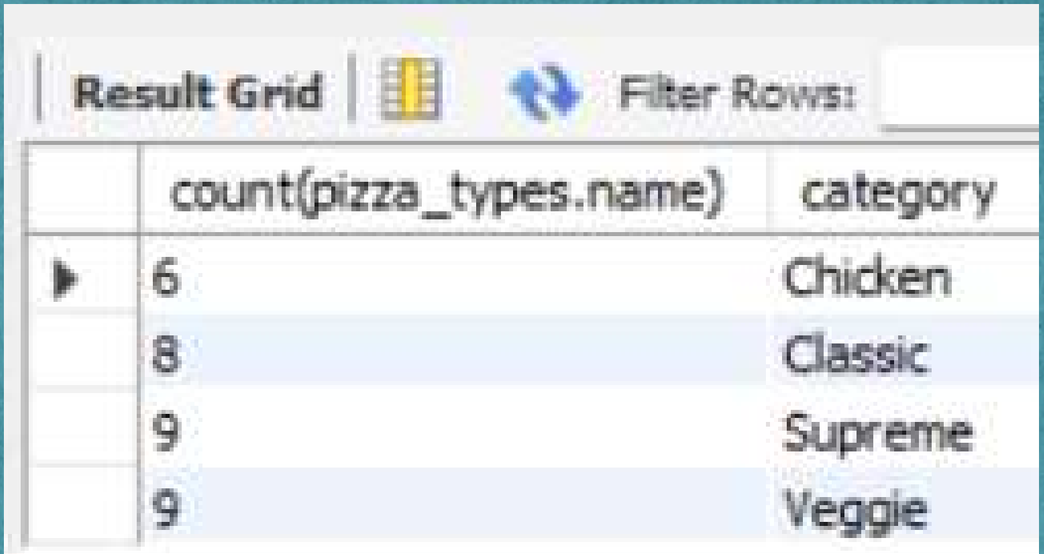
Result Grid | Filter Rows:

Hour_Of_Day	Total_Orders
12	6543
13	6203
18	5359
17	5143
19	4350
16	4185
14	3521

Business update 8

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

```
select count(pizza_types.name), pizza_types.category
from pizza_types
group by category
```



Result Grid | Filter Rows:

count(pizza_types.name)	category
6	Chicken
8	Classic
9	Supreme
9	Veggie

Business update 9

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

```
select pizza_types.name, sum(pizzas.price*order_details.quantity) as total_revenue_per_PizzaTypes
from pizzas join pizza_types
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_per_PizzaTypes desc
limit 3;
```

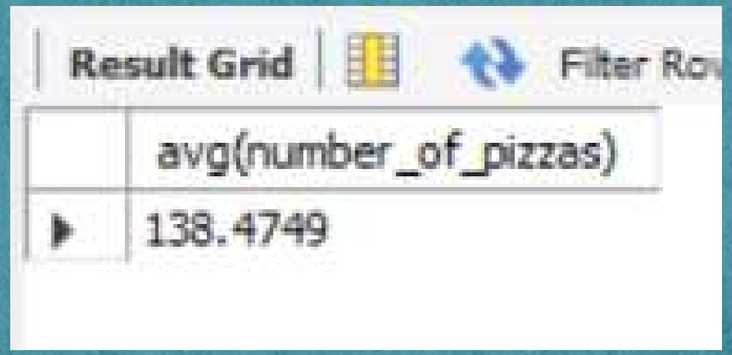
Result Grid | Filter Rows: | Export:

name	total_revenue_per_PizzaTypes
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Business update 10

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

```
select avg(number_of_pizzas) from
(select orders.order_date ,sum(order_details.quantity) as Number_of_pizzas
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as total_quantity
```



Business update 11

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

```
select pizza_types.category, (round(sum(pizzas.price*order_details.quantity),2)*100/
(select  sum(pizzas.price*order_details.quantity)
from pizzas
join order_details on pizzas.pizza_id = order_details.pizza_id)) as revenue_percent
from pizzas join pizza_types
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.category
order by revenue_percent desc;
```

Result Grid | Filter Rows:

category	revenue_percent
Classic	26.905960255669893
Supreme	25.45631126009906
Chicken	23.955137556847493
Veggie	23.682590927384418

Business update 12

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(order_details.quantity*pizzas.price) as revenue
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders on
order_details.order_id = orders.order_id
group by orders.order_date
order by orders.order_date) as sales;
```

Result Grid | Filter Rows:

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

Business update 13

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

```sql
select category, name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rnk
from
(select pizza_types.category, pizza_types.name, sum(pizzas.price*order_details.quantity) as revenue
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
join pizza_types on
pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category, pizza_types.name) as A) as B
where rnk<=3;
```

Contact

+91 6206096054

shubhamkesharibxr@gmail.com

Thank you!