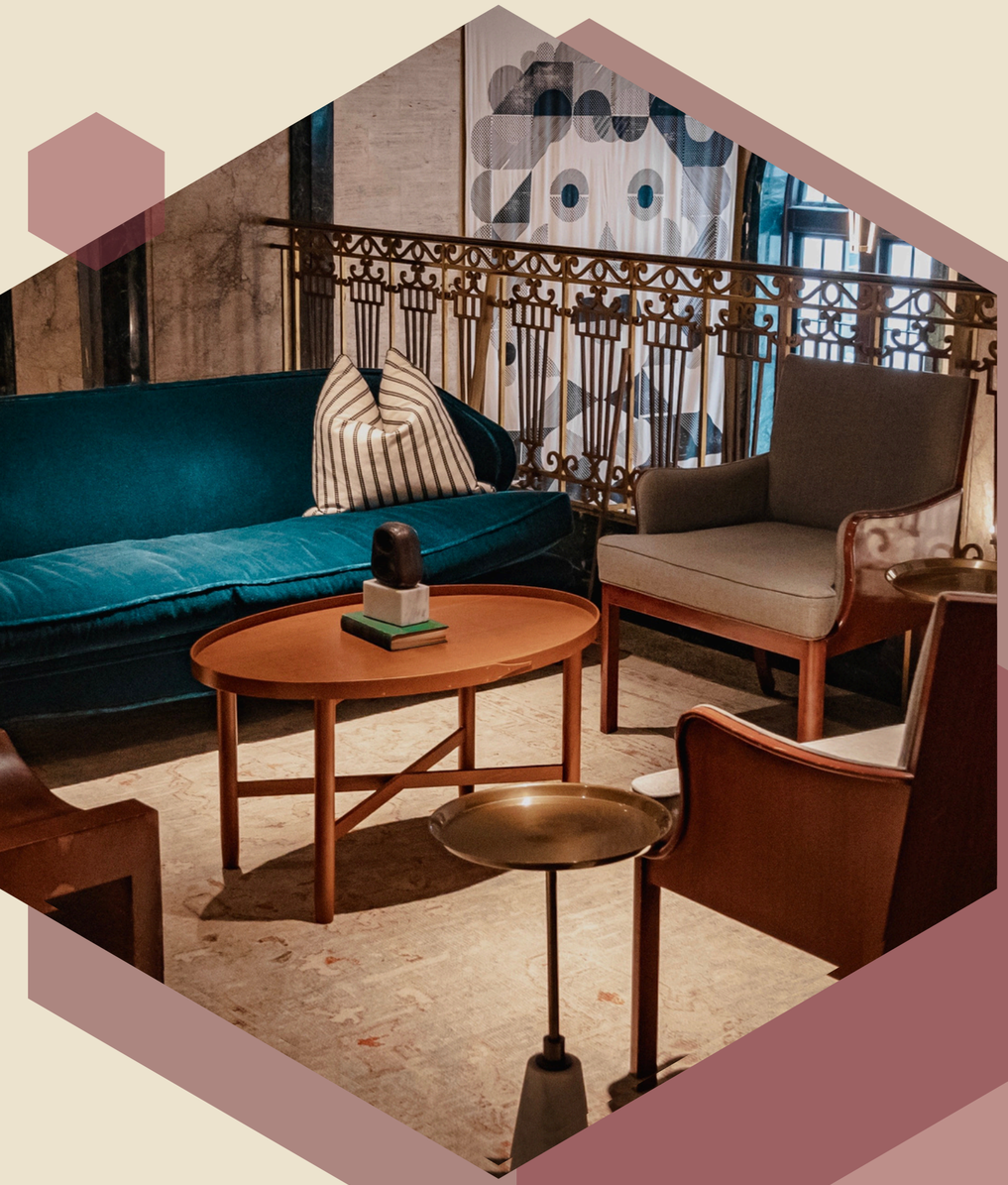
PIZZA SALES REPORT

5 May, 2024



INTRODUCTION

Welcome to our Pizza Sales Report Presentation. Today, we delve into a comprehensive overview of our sales performance, exploring the highs, challenges, and strategic insights that have shaped our journey. This presentation is more than just numbers, it's a narrative of our collective efforts, showcasing the impact of our sales strategies and the pathways to future success.

BASIC

01 Retrieve the total number of orders placed.

02 Calculate the total revenue generated from pizza sales.

03 Identify the highest-priced pizza.

04 Identify the most common pizza size ordered

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

INTERMEDIATE

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

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

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

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

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

ADVANCED

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

02 Analyze the cumulative revenue generated over time.

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

SUMMARY

Our project focuses on analyzing pizza sales data using SQL queries to derive valuable insights for a pizza business. We start by retrieving fundamental metrics such as the total number of orders placed and the total revenue generated from pizza sales. Next, we identify key attributes such as the highest-priced pizza and the most common pizza size ordered. Additionally, we list the top 5 most ordered pizza types along with their quantities to understand customer preferences.

Moving into the intermediate phase, we delve deeper into the data by joining tables to find the total quantity of each pizza category ordered. We also analyze the distribution of orders by hour of the day and the category-wise distribution of pizzas. Furthermore, we group orders by date to calculate the average number of pizzas ordered per day and determine the top 3 most ordered pizza types based on revenue.

In the advanced stage, we calculate the percentage contribution of each pizza type to total revenue, providing insights into the relative importance of different pizza varieties. We analyze the cumulative revenue generated over time to understand sales trends and patterns. Finally, we identify the top 3 most ordered pizza types based on revenue for each pizza category, allowing for targeted marketing strategies and menu optimization.

SQLQuery2.sql - A...HEK_PC\Abhis (71))* | SQLQuery1.sql - A...HEK_PC\Abhis (58))*

```sql
--Retrieve the total number of orders placed.

select count(order_id) as Total_order from orders
```

165 %

Results | Messages

	Total_order
1	21350

SQLQuery4.sql - A...HEK_PC\Abhis (51))* | SQLQuery3.sql - A...HEK_PC\Abhis (56))* | SQLQuery2.sql - A...HEK_PC\Abhis (71))* | SQLQuery1.sql - A...HEK_PC\Abhis (58))*

```sql
--Calculate the total revenue generated from pizza sales.

select round(sum(order_details.quantity * pizzas.price),2) as Total_sales from pizzas
join order_details
on pizzas.pizza_id= order_details.pizza_id
```

165 %

Results | Messages

	Total_sales
1	817860.05

SQLQuery4.sql - A...HEK_PC\Abhis (51))* | SQLQuery3.sql - A...HEK_PC\Abhis (56))* | SQLQuery2.sql - A...HEK_PC\Abhis (71))* | SQLQuery1.sql - A...HEK_PC\Abhis (58))*

```sql
--Identify the highest-priced pizza.

select top(1) round(pizzas.price,2) as Max_price ,pizza_types.name as Name from pizzas
join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc
```

165 %

Results | Messages

	Max_price	Name
1	35.95	The Greek Pizza

SQLQuery5.sql - A...HEK_PC\Abhis (54))* | SQLQuery4.sql - A...HEK_PC\Abhis (51))* | SQLQuery3.sql - A...HEK_PC\Abhis (56))* | SQLQuery2.sql - A...HEK_PC\Abhis (71))* | SQLQuery1.sql - A...HEK_PC\Abhis (58))*

```sql
-- Identify the most common pizza size ordered.

select pizzas.size , count(order_details.order_details_id) as Order_count from  pizzas
join order_details
on pizzas.pizza_id= order_details.pizza_id
group by pizzas.size
```

165 %

Results | Messages

	size	Order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

SQLQuery6.sql - A...HEK_PC\Abhis (70))* | SQLQuery5.sql - A...HEK_PC\Abhis (54))* | SQLQuery4.sql - A...HEK_PC\Abhis (51))* | SQLQuery3.sql - A...HEK_PC\Abhis (56))* | SQLQuery2.sql - A...HEK_PC\Abhis (71))* | SQLQuery1.sql - A...HEK_PC\

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

select top 5 pizza_types.name as Name, sum(order_details.quantity) as 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 Quantity_ordered desc
```

165 %

Results | Messages

	Name	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

SQLQuery5.sql - A...HEK_PC\Abhis (54))* | SQLQuery7.sql - A...HEK_PC\Abhis (65))* | SQLQuery4.sql - A...HEK_PC\Abhis (51))* | SQLQuery3.sql - A...HEK_PC\Abhis (56))* | SQLQuery2.sql - A...HEK_PC\Abhis (71))* | SQLQuery1.sql - A...HE

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

select pizza_types.category as Category, sum(order_details.quantity) as 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.Category
order by Quantity_ordered desc
```

165 %

Results | Messages

	Category	Quantity_ordered
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

SQLQuery10.sql - A...HEK_PC\Abhis (82)) SQLQuery9.sql - A...HEK_PC\Abhis (80)) SQLQuery8.sql - A...HEK_PC\Abhis (69))* SQLQuery7.sql - A...HEK_PC\Abhis (65))* SQLQuery5.sql - A...HEK_PC\Abhis (54))*

```sql
-- Determine the distribution of orders by hour of the day.

select datename ( HOUR ,time) as Hour,count(order_id) as Total_order from orders
group by datename ( HOUR ,time)
order by count(order_id) desc
```

65 %

Results Messages

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

SQLQuery10.sql - A...HEK_PC\Abhis (82)) | SQLQuery9.sql - A...HEK_PC\Abhis (80))* | SQLQuery8.sql - A...HEK_PC\Abhis (69))* | SQLQuery7.sql - A...HEK_PC\Abhis (65))* | SQLQuery5.sql - A...HEK_

```sql
-- Find the category-wise distribution of pizzas.

select category as Category ,count(name) as Name from pizza_types
group by category

select category as Category ,name as Name from pizza_types
group by category,name
```

165 %

Results | Messages

	Category	Name
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

	Category	Name
1	Chicken	The Barbecue Chicken Pizza
2	Chicken	The California Chicken Pizza
3	Chicken	The Chicken Alfredo Pizza
4	Chicken	The Chicken Pesto Pizza
5	Chicken	The Southwest Chicken Pizza
6	Chicken	The Thai Chicken Pizza
7	Classic	The Big Meat Pizza
8	Classic	The Classic Deluxe Pizza
9	Classic	The Greek Pizza
10	Classic	The Hawaiian Pizza
11	Classic	The Italian Capocollo Pizza
12	Classic	The Napolitana Pizza
13	Classic	The Pepperoni Pizza

SQLQuery10.sql - A...HEK_PC\Abhis (82))* | SQLQuery9.sql - A...HEK_PC\Abhis (80))* | SQLQuery8.sql - A...HEK_PC\Abhis (69))* | SQLQuery7.sql - A...HEK_PC\Abhis (65))* | SQLQuery5.sql - A...HEK_PC\Abhis (54))

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

SELECT ROUND(AVG(Total_count),2) AS Avg_No_pizza_per_day FROM
(select orders.date AS Day_No ,sum(order_details.quantity) As Total_count from order_details
join orders
on order_details.order_id =orders.order_id
group by orders.date) AS SUB
```

165 %

Results | Messages

	Avg_No_pizza_per_day
1	138.47

SQLQuery11.sql - A...HEK_PC\Abhis (72))* | SQLQuery10.sql - A...HEK_PC\Abhis (82))* | SQLQuery9.sql - A...HEK_PC\Abhis (80))* | SQLQuery8.sql - A...HEK_PC\Abhis (69))* | SQLQuery5.sql - A...HEK_PC\Abhis (54))* | SQLQuery1.sql - A...H

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

select Top 3 pizza_types.name as 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.name
order by Revenue desc;
```

165 %

Results | Messages

	Name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The Californa Chicken Pizza	41409.5

SQLQuery12.sql - A...HEK_PC\Abhis (53))* | SQLQuery13.sql - A...HEK_PC\Abhis (68))* | SQLQuery11.sql - A...HEK_PC\Abhis (72))* | SQLQuery8.sql - A...HEK_PC\Abhis (69))* | SQLQuery5.sql - A...HEK_PC\Abhis (54))*

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

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

165 %

Results | Messages

	Name	Revenue
1	Veggie	23.68
2	Chicken	23.96
3	Supreme	25.46
4	Classic	26.91

SQLQuery12.sql - A...HEK_PC\Abhis (53))* | SQLQuery13.sql - A...HEK_PC\Abhis (68))* | SQLQuery11.sql - A...HEK_PC\Abhis (72))* | SQLQuery8.sql - A...HEK_PC\Abhis (69))* | SQLQuery4.sql - A...HEK_PC\Abhis (51))* | SQLQuery1

```sql
-- Analyze the cumulative revenue generated over time.

select date as Date, round(sum(revenue) over (order by date),0) as Cum_Revenue from
(select orders.date as Date ,sum(pizzas.price * order_details.quantity) as Revenue from pizzas
join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date) as sales;
```

165 %

Results | Messages

	Date	Cum_Revenue
1	2015-01-01	2714
2	2015-01-02	5446
3	2015-01-03	8108
4	2015-01-04	9864
5	2015-01-05	11930
6	2015-01-06	14359
7	2015-01-07	16561
8	2015-01-08	19399
9	2015-01-09	21526
10	2015-01-10	23990
11	2015-01-11	25863
12	2015-01-12	27782
13	2015-01-13	29831
14	2015-01-14	32359
15	2015-01-15	34344
16	2015-01-16	36938
17	2015-01-17	39002
18	2015-01-18	40979
19	2015-01-19	43366

SQLQuery14.sql - A...HEK_PC\Abhis (76))* | SQLQuery12.sql - A...HEK_PC\Abhis (53))* | SQLQuery13.sql - A...HEK_PC\Abhis (68))* | SQLQuery11.sql - A...HEK_PC\Abhis (72))* | SQLQuery8.sql - A...HEK_PC\Abhis (69))* | SQLQuery1.sql - A...HEK_PC

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

Select Name , Round(Revenue,0),Category, Ranking from
(select Name, revenue,Category,Rank() over(partition by category order by Revenue desc) as Ranking
from
(select pizza_types.name as Name ,sum(pizzas.price * order_details.quantity) as Revenue,
pizza_types.category as Category  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.name,pizza_types.category) as Sales) as Rankk
where ranking <= 3 ;
```

165 %

Results | Messages

	Name	(No column name)	Category	Ranking
1	The Thai Chicken Pizza	43434	Chicken	1
2	The Barbecue Chicken Pizza	42768	Chicken	2
3	The California Chicken Pizza	41410	Chicken	3
4	The Classic Deluxe Pizza	38181	Classic	1
5	The Hawaiian Pizza	32273	Classic	2
6	The Pepperoni Pizza	30162	Classic	3
7	The Spicy Italian Pizza	34831	Supreme	1
8	The Italian Supreme Pizza	33477	Supreme	2
9	The Sicilian Pizza	30941	Supreme	3
10	The Four Cheese Pizza	32266	Veggie	1
11	The Mexicana Pizza	26781	Veggie	2
12	The Five Cheese Pizza	26067	Veggie	3

THANK YOU

05 May, 2024