Pizza Sales Analysis

Business overview

A pizza sales store manager can utilize this SQL project to extract valuable insights and conduct detailed data analysis, facilitating informed decision-making and efficient management of the store's operations.

Use Case

- Sales Analysis: By querying the order details and pizzas tables, managers can identify the best-selling pizzas, assess revenue from different pizza sizes, and evaluate pricing strategies.
- Inventory Management: Analyzing the pizza types and their ingredients helps in managing inventory more efficiently, ensuring that ingredients are stocked according to demand and reducing waste.
- Customer Preferences: Through data gathered in the orders and pizzas tables, managers can track customer preferences over time, adjusting the menu to cater to popular choices and experimenting with new or seasonal offerings.
- Operational Efficiency: Date and time data from the orders table allow managers to assess peak hours and staff the store appropriately, ensuring operational efficiency and customer satisfaction.- Marketing Insights: Data analysis can also support targeted marketing campaigns, like promotions on specific types of pizzas that are popular or on days when sales are typically lower.

SQL Queries

Retrieve the total number of orders placed

```
Select Count(distinct order_id) As "Total Orders"
from orders;
```

O/P

Total Orders
21338

Calculate the total revenue generated from pizza sales

```
Select round(sum((order_details.quantity *
price)),2) as 'Total  Revenue' from order_details
join pizzas on pizzas.pizza_id =
order_details.pizza_id;
```

O/P

Total Revenue
817860.05

Identify the highest-priced pizza using TOP/Limit functions

```
Select pizza_types.name as pizza_name,pizzas.price
from pizzas join pizza_types on
pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc Limit 1;
```

O/P

Pizza_name
The Greek Pizza

Identify the most common pizza size ordered

```
select pizzas.size, count(order_details_id) as 'No of
Orders'from order_details join pizzas on
pizzas.pizza_id = order_details.pizza_idgroup by
pizzas.size order by count(distinct order_id) desc
limit 1;
```

O/P	Size	No of orders
	L	18526

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

```
select pizza_types.name as 'Pizza', sum(quantity) as 'Total Ordered’ from order_details join pizzas 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 sum(quantity) desc limit 5;
```

O/P

Pizza	Total ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Find the total quantity of each pizza category ordered

```
Select pizza_types.category,sum(quantity) as 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.category order by Quantity desc;
```

O/P

Category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```
Select hour(order_time) as 'Hour',count(distinct order_id)
order_count from ordersgroup by hour(order_time)order
by order_count desc;
```

O/P

Hour	Order_count
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

Find the category-wise distribution of pizzas

```
Select category, count(name) as 'No of pizzas'from
pizza_types group by category order by 'No of pizzas' desc;
```

O/P

category	No of pizzas
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) as 'Avg pizzas ordered per
day' from (select order_date ,sum(order_details.quantity)
as Quantity from orders join order_details on
order_details.order_id = orders.order_id group by
order_date) as Order_Quantity;
```

O/P

Avg pizzas ordered per day
139

Determine the top 3 most ordered pizza types based on revenue

```
Select pizza_types.name,sum(order_details.quantity *
pizzas.price) 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.name order by revenue desc limit 3;
```

O/P

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

Calculate the percentage contribution of each pizza type to total revenues

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

O/P

category	Total_revenue_pct
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

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

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

Output on next slide

category	name	revenue	rnk
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

Conclusion

This SQL project not only serves as a robust data management system but also as a strategic tool for business intelligence. By maintaining comprehensive data on every aspect of the store's operations, the database allows store managers to make precise adjustments to improve both customer experience and profitability. When presented on a blog, this project can provide practical insights into how structured SQL queries can be used to harness data for real business applications, making it an excellent resource for aspiring data analysts and business owners alike.

Thank You!