SQL Project Report

Project Name: PIZZA RESTRO

By: Loveleen Arora

Project Objective

The objective of this project is to build a comprehensive SQL-based analytics solution for a Pizza Restaurant. The project focuses on cleaning and modifying data tables, performing sales and customer analysis, deriving product insights, identifying customer behavior patterns, and implementing advanced SQL techniques such as window functions, views, and CTEs (Common Table Expressions).

Table of Contents

1.Create Data Base and Use It

```
CREATE DATABASE PIZZA_RESTO; SHOW DATABASES; USE pizza_resto;
```

2.Show Tables

```
SHOW TABLES;
```

3.Customers Table Modification

```
DESC pizza_cx;
ALTER TABLE pizza_cx MODIFY customer_id VARCHAR(100) NOT NULL;
ALTER TABLE pizza_cx MODIFY COLUMN Customer_Name VARCHAR(100) NOT NULL;
ALTER TABLE pizza_cx MODIFY COLUMN Phone_Number VARCHAR(15) NOT NULL;
```

4.Category Table Modification

```
DESC pizza_cet;
ALTER TABLE pizza_cet MODIFY pizza_id VARCHAR(100) NOT NULL;
ALTER TABLE pizza_cet MODIFY pizza_name VARCHAR(100) NOT NULL;
ALTER TABLE pizza_cet MODIFY category VARCHAR(100) NOT NULL;
ALTER TABLE pizza_cet MODIFY quantity INT NOT NULL;
ALTER TABLE pizza_cet MODIFY customer_id VARCHAR(100) NOT NULL;
```

5.Order Table Modification

```
DESC pizza_ord;
ALTER TABLE pizza_ord MODIFY Order_ID VARCHAR(100) NOT NULL;
ALTER TABLE pizza_ord MODIFY Order_amount DECIMAL(20,3) NOT NULL;
ALTER TABLE pizza_ord MODIFY order_date DATE NOT NULL;
ALTER TABLE pizza_ord MODIFY customer_ID VARCHAR(100) NOT NULL;
```

6.Verified Indexes

```
SHOW INDEX FROM pizza_cx;
SHOW INDEX FROM pizza_ord;
SHOW INDEX FROM pizza_cet;
```

7. View All Data from Tables

```
SELECT * FROM pizza_cet;
SELECT * FROM pizza_cx;
SELECT * FROM pizza_ord;
```

8. Sale & Customer Analysis Queries

```
 -- Total Customers --
SELECT COUNT(*) AS total_customers FROM pizza_cx;

--Total Orders --
SELECT COUNT(*) AS total_orders FROM pizza_ord;

 --Total Revenue --
SELECT SUM(order_amount) AS total_revenue FROM pizza_ord;

--Total Pizzas Sold --
SELECT SUM(quantity) AS total_quantity FROM pizza_cet;

-- Average Revenue --
SELECT AVG(order_amount) AS average_revenue FROM pizza_ord;
```

9.Product Insights

-- Top 5 Best-Selling Pizzas--
SELECT pizza_name, SUM(quantity) AS total_sold FROM pizza_cet GROUP BY pizza_name ORDER BY total_sold DESC LIMIT 5;

-- Most Ordered Category --
SELECT category, SUM(quantity) AS total_quantity FROM pizza_cet GROUP BY category ORDER BY total_quantity DESC LIMIT 1;

-- Least Selling Pizzas –
SELECT pizza_name, SUM(quantity) AS total_quantity FROM pizza_cet GROUP BY pizza_name ORDER BY total_quantity ASC LIMIT 1;

-- Revenue by Pizzas --
SELECT pizza_cet.pizza_name, SUM(pizza_ord.order_amount) AS total_revenue FROM pizza_cet INNER JOIN pizza_ord ON pizza_cet.customer_id = pizza_ord.customer_id GROUP BY pizza_cet.pizza_name ORDER BY total_revenue DESC;

-- Revenue by Category --
SELECT pizza_cet.category, SUM(pizza_ord.order_amount) AS total_revenue FROM pizza_cet INNER JOIN pizza_ord ON pizza_cet.customer_id = pizza_ord.customer_id GROUP BY pizza_cet.category ORDER BY total_revenue DESC;

10.Customer Behavior Analysis

-- Highest Spending Customers --
SELECT pizza_cx.customer_name, SUM(pizza_ord.order_amount) AS total_revenue FROM pizza_cx INNER JOIN pizza_ord ON pizza_cx.customer_id = pizza_ord.customer_id GROUP BY pizza_cx.customer_name ORDER BY total_revenue DESC LIMIT 1;

-- Lowest Spending Customers --
SELECT pizza_cx.customer_name, SUM(pizza_ord.order_amount) AS total_revenue FROM pizza_cx INNER JOIN pizza_ord ON pizza_cx.customer_id = pizza_ord.customer_id GROUP BY pizza_cx.customer_name ORDER BY total_revenue ASC LIMIT 1;

-- Repeat Customers (More than 1 order) ---
SELECT order_id, COUNT(order_id) AS order_count FROM pizza_ord GROUP BY order_id HAVING COUNT(order_id) > 1;

11.Advance SQL Analysis

-- Rank Customers by Total Spend ---

```
SELECT pizza_cx.customer_name, SUM(pizza_ord.order_amount) AS total_spend, RANK() OVER (ORDER BY SUM(pizza_ord.order_amount) DESC) AS customer_rank FROM pizza_cx JOIN pizza_ord ON pizza_cx.customer_id = pizza_ord.customer_id GROUP BY pizza_cx.customer_name;

-- Rank Pizzas by Quantity Sold--
 SELECT pizza_name, SUM(quantity) AS total_quantity, DENSE_RANK() OVER (ORDER BY SUM(quantity) DESC) AS customer_rank FROM pizza_cet GROUP BY pizza_name;
```

11.Views & CETs

```
-- Create View for Pizza-wise Revenue --
CREATE VIEW pizza_revenue_view AS SELECT pizza_cet.pizza_name, SUM(pizza_ord.order_amount) AS total_revenue FROM pizza_cet JOIN pizza_ord ON pizza_cet.customer_id = pizza_ord.customer_id GROUP BY pizza_name;

-- Revenue by Month –
SELECT MONTH(order_date) AS order_month, SUM(order_amount) AS monthly_revenue FROM pizza_ord GROUP BY MONTH(order_date) ORDER BY order_month;

-- CTE: Repeat Customers (More than 1 Order) --
WITH order_count AS ( SELECT customer_id, COUNT(order_id) AS total_orders FROM pizza_ord GROUP BY customer_id )
SELECT pizza_cx.customer_name, pizza_ord.order_id

FROM order_count JOIN pizza_cx ON pizza_cx.customer_id = order_count.customer_id JOIN pizza_ord ON pizza_ord.customer_id = pizza_cx.customer_id WHERE order_count.total_orders > 1;
```

Project Conclusion

This SQL project demonstrates the end-to-end process of data preparation, query optimization, and advanced business analytics on a restaurant database. From creating and modifying tables to generating meaningful insights using joins, aggregations, ranking, and CTEs, the project provides a solid foundation for real-world data analysis using SQL. It showcases the ability to identify customer trends, product performance, and sales behaviors effectively.