SQL PROJECT ON

PIZZA SALES

ANALYZING SALES DATA WITH STRUCTURED QUERIES

Presented By:Swati

Email: swatirana23112gmail.com

INTRODUCTION

Welcome to my SQL project on Pizza Sales, where I analyze a rich dataset to uncover valuable insights about pizza orders, types, and sales trends.
This project aims to:

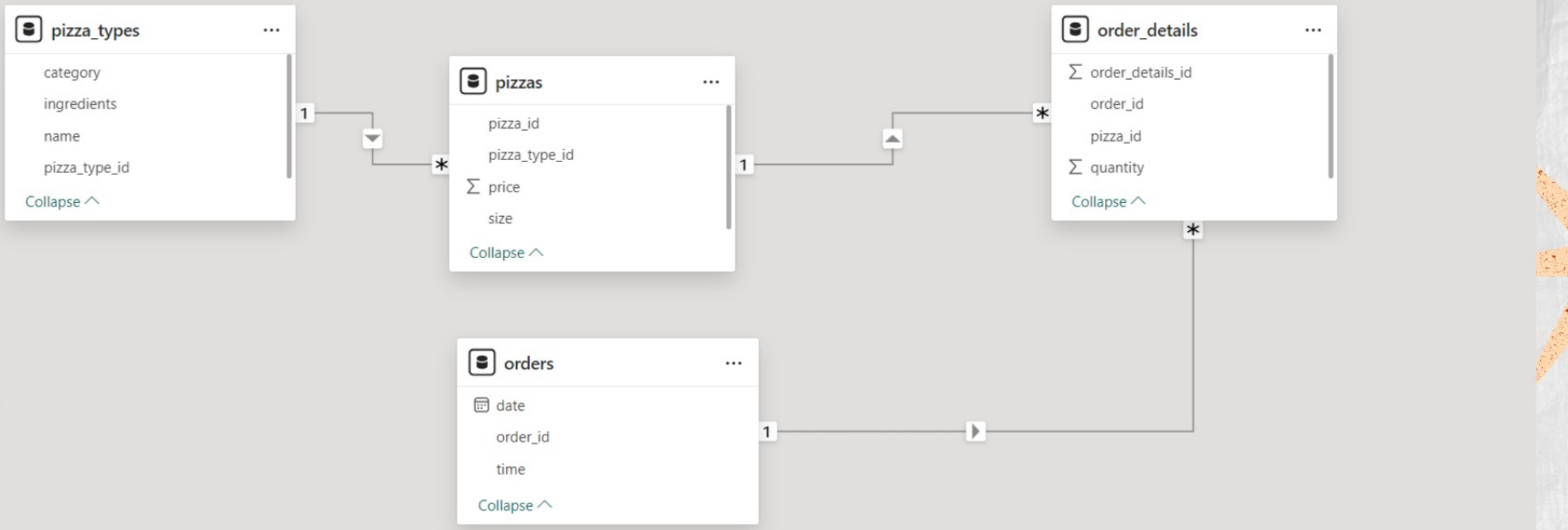
- Understand the sales performance of different pizzas and categories.
- Identify customer preferences and sales patterns.
- Gain hands-on experience in SQL for data analysis.

The dataset consists of four tables:

- Pizzas: Information about pizza_id, sizes and price.
- Pizza Types: Details about pizza name,categories and ingredients.
- Orders: Order-specific data like order_id, date and time.
- Order Details: Details like quantity .

By executing 13 SQL queries (5 basic, 5 intermediate, and 3 advanced), I extracted actionable insights and honed my SQL skills. This project showcases how structured query language can be a powerful tool for data-driven decision-making

UNDERSTANING THE DATASET



TOOL USED

- SQL SERVER MANAGEMENT STUDIO

METHODOLOGY:

- DATA EXPLORATION.
- WRITING AND RUNNING QUERIES.
- INTERPRETING THE RESULTS.

QUERIES

- **BASIC**

1) Retrieve the total number of ordes placed
2) Calculate the total revenue generated from Pizza sales
3) Identify the highest priced Pizza
4) Identify the most common pizza size ordered
5) List the Top 5 most ordered pizza types along with their quantities

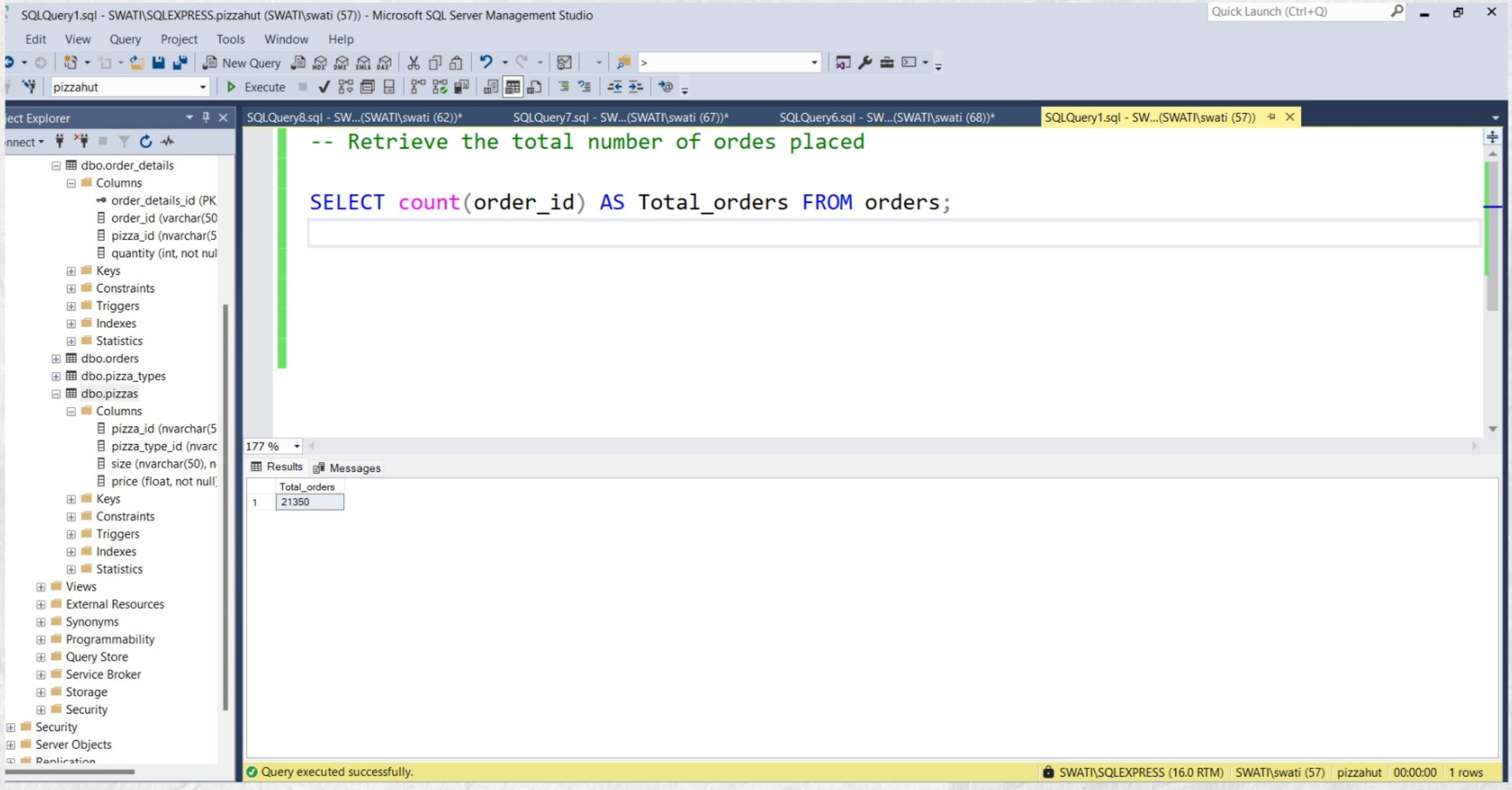
- **INTERMEDIATE**

6) Join the necessary tables to find the total quantity of each pizza category ordered.
7) Determine the distribution of the orders by hour of the day
8) Join relevant tables to find out the category wise distribution of pizzas
9) Group the orders By Date and calculate the average number of pizzas ordered per day.
10) Determne the Top 3 most ordered pizza types based on revenue

- **ADVANCE**

11) Calculate the percentage contribution of each pizza type to total revenue.
12) Analyze the cumulative Revenue generated over time
13) Determine the Top 3 most ordered pizza types based on Revenue for each pizza Category

BASIC QUERY 1



BASIC QUERY 2

SQLQuery6.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (68))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- dbo.order_details
  - Columns
    - order_details_id (PK
    - order_id (varchar(50
    - pizza_id (nvarchar(5
    - quantity (int, not nul
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.orders
- dbo.pizza_types
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views
- External Resources
- Synonyms
- Programmability
- Query Store
- Service Broker
- Storage
- Security

Security

Server Objects

Replication

SQLQuery8.sql - SW...(SWATI\swati (62))* | SQLQuery7.sql - SW...(SWATI\swati (67))* | SQLQuery6.sql - SW...(SWATI\swati (68))* | SQLQuery1.sql - SW...(SWATI\swati (57))

```
-- Calculate the total revenue generated from Pizza sales

SELECT ROUND(SUM(p.price * od.quantity), 2) AS Total_Revenue
FROM pizzas AS p INNER JOIN order_details AS od
ON p.pizza_id = od.pizza_id;
```

177 %

Results | Messages

	Total_Revenue
1	817860.05

Query executed successfully.

SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (68) | pizzahut | 00:00:00 | 1 rows

BASIC QUERY 3

SQLQuery7.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (67))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- dbo.order_details
  - Columns
    - order_details_id (PK
    - order_id (varchar(50
    - pizza_id (nvarchar(5
    - quantity (int, not nul
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.orders
- dbo.pizza_types
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views
- External Resources
- Synonyms
- Programmability
- Query Store
- Service Broker
- Storage
- Security

Security
Server Objects
Replication

SQLQuery8.sql - SW...(SWATI\swati (62))* | SQLQuery7.sql - SW...(SWATI\swati (67))* | SQLQuery6.sql - SW...(SWATI\swati (68))* | SQLQuery1.sql - SW...(SWATI\swati (57))

```
--Identify the highest priced Pizza

SELECT TOP 1 pt.name, p.price AS Highest_priced_pizza
FROM pizzas AS p  join pizza_types AS pt
ON p.pizza_type_id = pt.pizza_type_id
ORDER BY p.price desc;
```

177 %

Results | Messages

	name	Highest_priced_pizza
1	The Greek Pizza	35.9500007629395

Query executed successfully.

SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (67) | pizzahut | 00:00:00 | 1 rows

BASIC QUERY 4

SQLQuery8.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (62))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query | Execute | pizzahut

Object Explorer

Connect

- dbo.order_details
  - Columns
    - order_details_id (PK
    - order_id (varchar(50
    - pizza_id (nvarchar(5
    - quantity (int, not nul
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.orders
- dbo.pizza_types
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views
- External Resources
- Synonyms
- Programmability
- Query Store
- Service Broker
- Storage
- Security

Security

Server Objects

SQLQuery8.sql - SW...(SWATI\swati (62))* | SQLQuery7.sql - SW...(SWATI\swati (67))* | SQLQuery6.sql - SW...(SWATI\swati (68))* | SQLQuery1.sql - SW...(SWATI\swati (57))

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

SELECT Top 1 p.size,COUNT(od.order_details_id) AS order_count
FROM pizzas AS p join order_details AS od
ON p.pizza_id = od.pizza_id
group by p.size order by order_count desc;
```

177 %

Results | Messages

	size	order_count
1	L	18526

Query executed successfully.

SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (62) | pizzahut | 00:00:00 | 1 rows

BASIC QUERY 5

SQLQuery10.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (63)) - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- Columns
  - order_id (PK, varcha
  - date (date, not null)
  - time (time(7), not nu
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery13.sql - S...(SWATI\swati (59))* | SQLQuery12.sql - S...(SWATI\swati (60))* | SQLQuery10.sql - S...(SWATI\swati (63)) | SQLQuery8.sql - SW...(SWATI\swati (62))

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

SELECT TOP 5 pt.name , Sum(od.quantity) AS quantity
from pizza_types AS pt join pizzas AS p
ON pt.pizza_type_id = p.pizza_type_id
join order_details AS od ON od.pizza_id=p.pizza_id
group by pt.name
order by quantity desc;
```

177 %

Results | Messages

	name	quantity
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

Query executed successfully. | SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (63) | pizzahut | 00:00:00 | 5 rows

INTERMEDIATE QUERY1

SQLQuery12.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (60))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | Execute | pizzahut

Object Explorer

Connect

- Columns
  - order_id (PK, varcha
  - date (date, not null)
  - time (time(7), not nu
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery13.sql - S...(SWATI\swati (59))* | SQLQuery12.sql - S...(SWATI\swati (60))* | SQLQuery10.sql - S...(SWATI\swati (63)) | SQLQuery8.sql - SW...(SWATI\swati (62))

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

SELECT pt.category , sum(od.quantity) AS quantity
FROM pizza_types AS pt join pizzas AS p
ON pt.pizza_type_id = p.pizza_type_id
join order_details AS od ON p.pizza_id = od.pizza_id
group by pt.category;
```

177 %

Results | Messages

	category	quantity
1	Chicken	11050
2	Classic	14888
3	Supreme	11987
4	Veggie	11649

Query executed successfully. SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (60) | pizzahut | 00:00:00 | 4 rows

INTERMEDIATE QUERY2

SQLQuery13.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (59))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- dbo.order_details
  - Columns
    - order_details_id (PK
    - order_id (varchar(50
    - pizza_id (nvarchar(5
    - quantity (int, not nul
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.orders
  - Columns
    - order_id (PK, varcha
    - date (date, not null)
    - time (time(7), not nu
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5

SQLQuery13.sql - S...(SWATI\swati (59))* | SQLQuery12.sql - S...(SWATI\swati (60)) | SQLQuery10.sql - S...(SWATI\swati (63)) | SQLQuery8.sql - SW...(SWATI\swati (62))

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

select DATEPART(HOUR,time) AS Time , COUNT(order_id) AS Orders_COUNT
From orders
group by DATEPART(HOUR,time)
order By time ;
```

177 %

Results | Messages

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

Query executed successfully. | SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (59) | pizzahut | 00:00:00 | 15 rows

INTERMEDIATE QUERY3

SQLQuery14.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (66))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- Indexes
- Statistics
- dbo.orders
  - Columns
    - order_id (PK, varcha
    - date (date, not null)
    - time (time(7), not nu
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views

SQLQuery14.sql - S...(SWATI\swati (66))* | SQLQuery13.sql - S...(SWATI\swati (59)) | SQLQuery12.sql - S...(SWATI\swati (60)) | SQLQuery10.sql - S...(SWATI\swati (63))

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

SELECT pt.category , COUNT(p.pizza_id)
from pizza_types AS pt join pizzas AS p
on pt.pizza_type_id = p.pizza_type_id
group by pt.category;
```

177 %

Results | Messages

	category	(No column name)
1	Chicken	18
2	Classic	26
3	Supreme	25
4	Veggie	27

Query executed successfully. SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (66) | pizzahut | 00:00:00 | 4 rows

INTERMEDIATE QUERY4

SQLQuery15.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (71))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- order_id (varchar(50
- pizza_id (nvarchar(5
- quantity (int, not nul
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo.orders
  - Columns
    - order_id (PK, varcha
    - date (date, not null)
    - time (time(7), not nu
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null

SQLQuery15.sql - S...(SWATI\swati (71))* | SQLQuery14.sql - S...(SWATI\swati (66)) | SQLQuery13.sql - S...(SWATI\swati (59)) | SQLQuery12.sql - S...(SWATI\swati (60))

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

SELECT AVG(quantity) AS orders_per_day
from (SELECT o.Date , SUM(od.quantity) AS quantity
FROM orders AS o  join order_details AS od
ON o.order_id = od.order_id
Group by Date )AS Orders_quantity;
```

177 %

Results | Messages

	orders_per_day
1	138

Query executed successfully.

SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (71) | pizzahut | 00:00:00 | 1 rows

INTERMEDIATE QUERY5

SQLQuery16.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (65))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- Indexes
- Statistics
- dbo.orders
  - Columns
    - order_id (PK, varcha
    - date (date, not null)
    - time (time(7), not nu
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- Views

SQLQuery16.sql - S...(SWATI\swati (65))* | SQLQuery15.sql - S...(SWATI\swati (71)) | SQLQuery14.sql - S...(SWATI\swati (66)) | SQLQuery13.sql - S...(SWATI\swati (59))

```
--Determne the Top 3 most ordered pizza types based on revenue

SELECT TOP 3 pt.name , sum(p.price* od.quantity) AS Total_Revenue
from pizza_types AS pt join pizzas AS p
ON pt.pizza_type_id = p.pizza_type_id
join order_details AS od
On p.pizza_id=od.pizza_id
Group By pt.name
order by Total_Revenue desc;
```

177 %

Results | Messages

	name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

Query executed successfully. SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (65) | pizzahut | 00:00:00 | 3 rows

ADVANCE QUERY 1

SQLQuery17.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (60))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- dbo.order_details
  - Columns
    - order_details_id (PK
    - order_id (varchar(50
    - pizza_id (nvarchar(5
    - quantity (int, not nul
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.orders
  - Columns
    - order_id (PK, varcha
    - date (date, not null)
    - time (time(7), not nu
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(5
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5

SQLQuery17.sql - S...(SWATI\swati (60))* | SQLQuery16.sql - not connected | SQLQuery15.sql - not connected | SQLQuery14.sql - not connected | SQLQuery13.sql - not connected

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

SELECT pt.category, Round(sum(od.quantity * p.price)/(select sum(od.quantity*p.price) from
order_details AS od join pizzas AS p ON od.pizza_id = p.pizza_id)*100,2) AS Total_Revenue
from pizza_types AS pt join pizzas AS p
ON pt.pizza_type_id = p.pizza_type_id
join order_details AS od
ON od.pizza_id = p.pizza_id
Group by pt.category
order by Total_Revenue desc;
```

177 %

Results | Messages

	category	Total_Revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

Query executed successfully. | SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (60) | pizzahut | 00:00:00 | 4 rows

ADVANCE QUERY 2

SQLQuery18.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (61))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo.orders
  - Columns
    - order_id (PK, varcha
    - date (date, not null)
    - time (time(7), not nu
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizza_types
  - Columns
    - pizza_type_id (nvarc
    - name (nvarchar(50),
    - category (nvarchar(
    - ingredients (nvarcha
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
- dbo.pizzas
  - Columns
    - pizza_id (nvarchar(5
    - pizza_type_id (nvarc
    - size (nvarchar(50), n
    - price (float, not null
  - Keys
  - Constraints
  - Triggers

SQLQuery18.sql - S...(SWATI\swati (61))* | SQLQuery17.sql - S...(SWATI\swati (60)) | SQLQuery16.sql - not connected | SQLQuery15.sql - not connected | SQLQuery14.sql - not connected

```sql
--Analyze the cumulative Revenue generated over time

SELECT date, sum(Revenue) over(order by date) AS cum_Revenue
FROM
(SELECT o.date, sum(od.Quantity* p.price) AS Revenue
FROM order_details AS od join pizzas AS p
ON p.pizza_id =od.pizza_id
join orders AS o
ON o.order_id = od.order_id
GROUP BY o.date ) AS Sales;
```

161 %

Results | Messages

	date	cum_Revenue
1	2015-01-01	2713.85000228882
2	2015-01-02	5445.7500038147
3	2015-01-03	8108.15000724792
4	2015-01-04	9863.60000801086
5	2015-01-05	11929.5500087738
6	2015-01-06	14358.5000114441
7	2015-01-07	16560.700012207
8	2015-01-08	19399.0500183105
9	2015-01-09	21526.4000225067
10	2015-01-10	23990.350025177
11	2015-01-11	25862.6500263214
12	2015-01-12	27781.7000274658
13	2015-01-13	29831.3000278473
14	2015-01-14	32358.7000293732
15	2015-01-15	34343.5000324249
16	2015-01-16	36937.6500339508
17	2015-01-17	39001.7500343323

Query executed successfully. | SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (61) | pizzahut | 00:00:00 | 358 rows

ADVANCE QUERY 3

SQLQuery19.sql - SWATI\SQLEXPRESS.pizzahut (SWATI\swati (58))* - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

New Query | pizzahut | Execute

Object Explorer

Connect

- SWATI\SQLEXPRESS (SQL Server 16.0
  - Databases
    - System Databases
    - Database Snapshots
    - pizzahut
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.order_details
          - Columns
            - order_details_id (PK
            - order_id (varchar(50
            - pizza_id (nvarchar(5
            - quantity (int, not nul
          - Keys
          - Constraints
          - Triggers
          - Indexes
          - Statistics
        - dbo.orders
          - Columns
            - order_id (PK, varcha
            - date (date, not null)
            - time (time(7), not nu
          - Keys
          - Constraints
          - Triggers
          - Indexes
          - Statistics
        - dbo.pizza_types
          - Columns

SQLQuery15.sql - not connected | SQLQuery16.sql - not connected | SQLQuery17.sql - not connected | SQLQuery18.sql - not connected | SQLQuery19.sql - S...(SWATI\swati (58))*

```sql
--Determine the Top 3 most ordered pizza types based on Revenue for each pizza Category
SELECT category,name,Revenue FROM
(SELECT category,name, Revenue ,rank() over
(partition by category order by Revenue desc) AS rank
FROM
(SELECT  pt.category ,pt.name, sum(od.quantity*p.price) AS Revenue
FROM pizza_types AS pt join pizzas AS p
ON pt.pizza_type_id =p.pizza_type_id
join order_details AS od On od.pizza_id = p.pizza_id
group By pt.category ,pt.name) AS a ) AS b
where rank<=3;
```

161 %

Results | Messages

	category	name	Revenue
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.5
10	Veggie	The Four Cheese Pizza	32265.7010040283
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5

Query executed successfully. | SWATI\SQLEXPRESS (16.0 RTM) | SWATI\swati (58) | pizzahut | 00:00:01 | 12 rows

KEY INSIGHTS

KEY INSIGHTS FROM PIZZA SALES ANALYSIS

Top-Selling Pizza Types:
- The classic deluxe pizza is most ordered pizza type by quantity and The Thai Chicken pizza is most ordered pizza type by the revenue.

Sales Trends:
- 12pm to 1pm are the peak sales hours of the day while 9am to 10am are the least sales hours of the day.

Customer Preferences:
- People ordered pizzas from the Classic pizza category the most and Chicken pizza category the least by Quantity.

Revenue Drivers:
- Classic pizza category contributed 26.91% ,Supreme 25.46%,Chicken 23.96% , Veggie 23.68% to the Total Revenue.

Order Insights:
- L is the most common pizza size ordered and Average number of Pizzas ordered per day is 138.

Overall Sales Performance:
- Total Revenue: 817860.05
- Number of Orders placed: 21350
- Number of Pizzas Sold: 49574
- Average order Value:38.31

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

- This project provided valuable insights into the pizza sales dataset by leveraging SQL for data analysis.
- Key achievements include identifying top-selling pizzas, understanding customer preferences, and analyzing sales trends.
- The 13 queries (5 basic, 5 intermediate, and 3 advanced) demonstrated a progressive understanding of SQL concepts, from basic data exploration to advanced analytics.

FUTURE SCOPE:

- Use tools like Power BI or Tableau to create interactive dashboards for better data representation.

THANK YOU!

Feel Free to Ask About Anything

Presented by: **Swati**
Email: swatirana2311@gmail.com
LinkedIn Profile:
https://www.linkedin.com/in/swati-rana-5a8500222?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app