

Marti Rosenbeck

Final Written Report

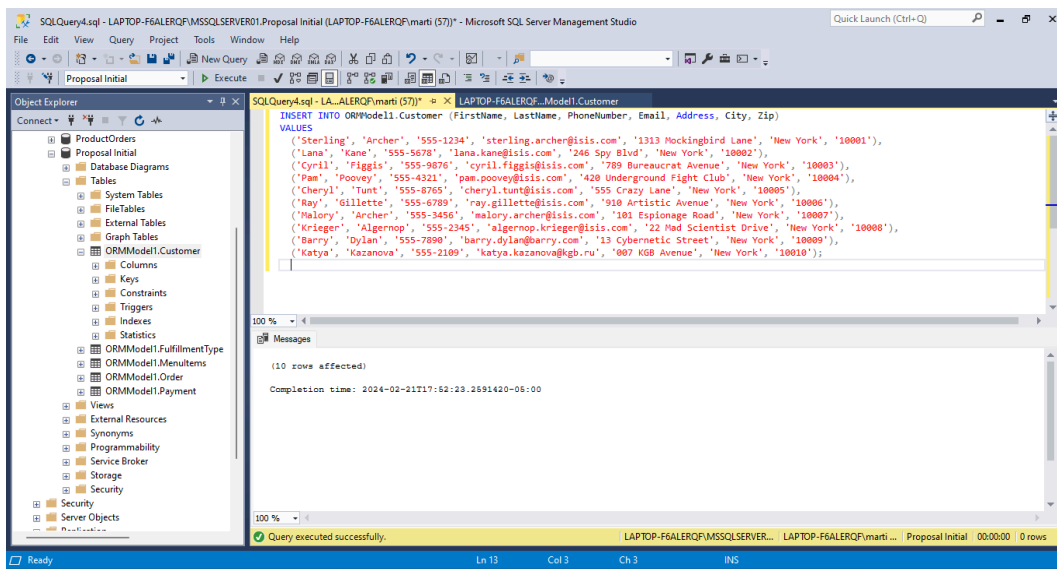
IFT 300

2/25/2024

Introduction

This report explores customer order habits for online ordering within the restaurant industry. Centered around essential entities such as customers, orders, menu items, fulfillment types, and payments, the objective is to uncover valuable insights that can enhance and refine the online food service experience. Specifically, I provide insights into order volumes, menu preferences, and customer behaviors.

Inserting 10 customers into the customer table



```
INSERT INTO ORMModel1.Customer (FirstName, LastName, PhoneNumber, Email, Address, City, Zip)
VALUES
('Sterling', 'Archer', '555-1234', 'sterling.archer@isis.com', '1313 Mockingbird Lane', 'New York', '10001'),
('Lana', 'Kane', '555-5678', 'lana.kane@isis.com', '246 Spy Blvd', 'New York', '10002'),
('Cyril', 'Figgis', '555-9876', 'cyril.figgis@isis.com', '789 Bureaucrat Avenue', 'New York', '10003'),
('Pam', 'Poovey', '555-4321', 'pam.poovey@isis.com', '420 Underground Fight Club', 'New York', '10004'),
('Cheryl', 'Tunt', '555-8765', 'cheryl.tunt@isis.com', '555 Crazy Lane', 'New York', '10005'),
('Ray', 'Gillette', '555-6789', 'ray.gillette@isis.com', '910 Artistic Avenue', 'New York', '10006'),
('Malory', 'Archer', '555-3456', 'malory.archer@isis.com', '101 Espionage Road', 'New York', '10007'),
('Krieger', 'Algernop', '555-2345', 'algernop.krieger@isis.com', '22 Mad Scientist Drive', 'New York', '10008'),
('Barry', 'Dylan', '555-7890', 'barry.dylan@barry.com', '13 Cybernetic Street', 'New York', '10009'),
('Katya', 'Kazanova', '555-2109', 'katya.kazanova@kqb.ru', '007 KGB Avenue', 'New York', '10010');
```

100 %

Messages

(10 rows affected)

Completion time: 2024-02-21T17:52:23.2591420-05:00

100 %

Query executed successfully.

Customer data in table

SQLQuery5.sql - LAPTOP-F6ALERQF\MSSQLSERVER01.Proposal Initial (LAPTOP-F6ALERQF\marti) - Microsoft SQL Server Management Studio

```

/***** Script for SelectTopRows command from SSIS *****/
SELECT TOP (10) [CustomerID]
      [FirstName]
      [LastName]
      [PhoneNumber]
      [Email]
      [Address]
      [City]
      [Zip]
FROM [Proposal Initial].[ORMModel1].[Customer]
  
```

customerID	FirstName	LastName	PhoneNumber	Email	Address	City	Zip
1	Sterling	Archer	555-1234	sterling.archer@sis.com	1313 Mockingbird Lane	New York	10001
2	Lana	Kane	555-5678	lana.kane@sis.com	246 Spy Blvd	New York	10002
3	Cyril	Figgs	555-9876	cyril.figgs@sis.com	789 Bureaucrat Avenue	New York	10003
4	Pam	Poovey	555-4321	pam.poovey@sis.com	420 Underground Fight Club	New York	10004
5	Cheryl	Turt	555-8765	cheryl.turt@sis.com	555 Crazy Lane	New York	10005
6	Ray	Gilette	555-6789	ray.gilette@sis.com	910 Artistic Avenue	New York	10006
7	Malory	Archer	555-3456	malory.archer@sis.com	101 Espionage Road	New York	10007
8	Krieger	Algemop	555-2345	algemop.krieger@sis.com	22 Mad Scientist Drive	New York	10008
9	Bary	Dylan	555-7890	bary.dylan@bary.com	13 Cybernetic Street	New York	10009
10	Katya	Kazanova	555-2109	katya.kazanova@gb.ru	007 KGB Avenue	New York	10010

Query executed successfully. LAPTOP-F6ALERQF\MSSQLSERVER... LAPTOP-F6ALERQF\marti ... Proposal Initial | 00:00:00 | 10 rows

Inserting fulfillment type data into FulfillmentType table

SQLQuery11.sql - LAPTOP-F6ALERQF\MSSQLSERVER01.Proposal Initial (LAPTOP-F6ALERQF\marti) - Microsoft SQL Server Management Studio

```

INSERT INTO ORMModel1.FulfillmentType (FulfillmentTypeName, orderID, customerID, deliveryName, pickupName)
VALUES
('Delivery', 1, 1, '123 Main St, Apt 4', NULL),
('Pickup', 2, 2, NULL, 'The Grill on Main'),
('Delivery', 3, 3, '456 Broad St, Suite 7', NULL),
('Pickup', 4, 4, NULL, 'The Grill on Main'),
('Delivery', 5, 5, '789 Side St, Unit 12', NULL),
('Pickup', 6, 6, NULL, 'The Grill on Main'),
('Delivery', 7, 7, '101 Penthouse Plaza', NULL),
('Pickup', 8, 8, NULL, 'The Grill on Main'),
('Delivery', 9, 9, '007 Secret St, KGB HQ', NULL),
('Pickup', 10, 10, NULL, 'The Grill on Main');
  
```

Completion time: 2024-02-22T14:33:01.3600015-05:00

Query executed successfully. LAPTOP-F6ALERQF\MSSQLSERVER... LAPTOP-F6ALERQF\marti ... Proposal Initial | 00:00:00 | 0 rows

FulfillmentType data in table

SQLQuery12.sql - LAPTOP-F6ALERQF\MSSQLSERVER01.Proposal Initial (LAPTOP-F6ALERQF\marti (55)) - Microsoft SQL Server Management Studio

Object Explorer

- Graph Tables
- ORMModel1.Customer
 - Columns
 - customerID (PK, int, not null)
 - FirstName (nvarchar(50), r...)
 - LastName (nvarchar(50), r...)
 - PhoneNumber (nvarchar(15), r...)
 - Email (nvarchar(150), not null)
 - Address (nvarchar(125), not null)
 - City (nvarchar(50), null)
 - Zip (int, null)
 - Keys
 - Customer_PK
 - Constraints
 - Triggers
 - Indexes
 - Statistics
- ORMModel1.FulfillmentType
 - Columns
 - fulfillmentTypeName (PK, nvarchar(50), not null)
 - orderID (int, not null)
 - customerID (int, not null)
 - deliveryName (nvarchar(100), not null)
 - pickUpName (nvarchar(100), not null)
 - Keys
 - FulfillmentType_PK
 - FulfillmentType_UC4
 - Constraints

SQLQuery12.sql - L...ALERQF\marti (55) *

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [fulfillmentTypeName]
, [orderID]
, [customerID]
, [deliveryName]
, [pickUpName]
FROM [Proposal Initial].[ORMModel1].[FulfillmentType]
  
```

Results

	fulfillmentTypeName	orderID	customerID	deliveryName	pickUpName
1	Delivery	1	1	123 Main St, Apt 4	NULL
2	Pickup	2	2	NULL	The Grill on Main
3	Delivery	3	3	456 Broad St, Suite 7	NULL
4	Pickup	4	4	NULL	The Grill on Main
5	Delivery	5	5	789 Side St, Unit 12	NULL
6	Pickup	6	6	NULL	The Grill on Main
7	Delivery	7	7	101 Penthouse Plaza	NULL
8	Pickup	8	8	NULL	The Grill on Main
9	Delivery	9	9	007 Secret St, KGB HQ	NULL
10	Pickup	10	10	NULL	The Grill on Main

Query executed successfully. LAPTOP-F6ALERQF\MSSQLSERVER... LAPTOP-F6ALERQF\marti ... Proposal Initial 00:00:00 10 rows

Inserting 20 menu items into the MenuItems table

SQLQuery20.sql - LAPTOP-F6ALERQF\MSSQLSERVER01.Proposal Initial (LAPTOP-F6ALERQF\marti (55)) - Microsoft SQL Server Management Studio

Object Explorer

- Proposal Initial
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - ORMModel1.Customer
 - ORMModel1.FulfillmentType
 - ORMModel1.MenuItems
 - ORMModel1.Order
 - ORMModel1.Payment
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Always On High Availability
 - Management
 - Integration Services Catalogs
 - SQL Server Agent (Agent XPs disabled)
 - XEvent Profiler

SQLQuery20.sql - L...ALERQF\marti (55) *

```

INSERT INTO ORMModel1.MenuItems (menuItemsID, menuItemsName, price)
VALUES
(1, 'Grilled Chicken Salad', 10.99),
(2, 'Vegetarian Wrap', 8.99),
(3, 'Quinoa Bowl', 12.49),
(4, 'Avocado Toast', 6.99),
(5, 'Spinach and Feta Omelette', 9.99),
(6, 'Mango Tango Smoothie', 4.99),
(7, 'Sweet Potato Fries', 5.49),
(8, 'Caprese Panini', 11.99),
(9, 'Berry Blast Salad', 7.99),
(10, 'Greek Yogurt Parfait', 3.99),
(11, 'Quinoa Avocado Salad', 9.49),
(12, 'Pesto Pasta', 13.99),
(13, 'Hummus Platter', 8.49),
(14, 'Vegan Chocolate Cake', 6.99),
(15, 'Cucumber Mint Lemonade', 3.49),
(16, 'Margherita Pizza', 12.99),
(17, 'Coconut Curry Tofu', 10.49),
(18, 'Chia Seed Pudding', 4.49),
(19, 'Green Goddess Smoothie', 5.99),
(20, 'Black Bean Burger', 8.99);
  
```

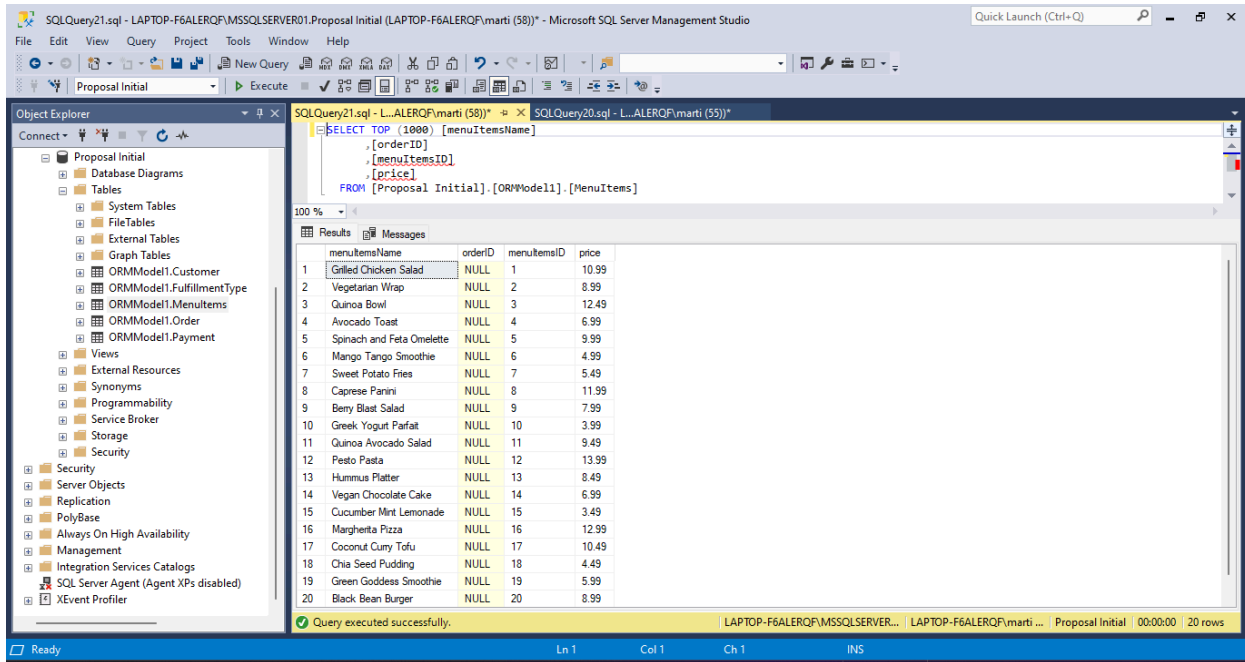
Messages

(20 rows affected)

Completion time: 2024-02-22T16:01:23.9827599-05:00

Query executed successfully. LAPTOP-F6ALERQF\MSSQLSERVER... LAPTOP-F6ALERQF\marti ... Proposal Initial 00:00:00 0 rows

Menu items in table



The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'Proposal Initial', including tables like 'MenuItems'. The main window shows a SQL query executed successfully, returning a list of 20 menu items with their respective IDs and prices.

```
SELECT TOP (1000) [menuItemsName]
, [orderID]
, [menuItemsID]
, [price]
FROM [Proposal Initial].[ORMModel1].[MenuItems]
```

menuItemsName	orderID	menuItemsID	price
Grilled Chicken Salad	NULL	1	10.99
Vegetarian Wrap	NULL	2	8.99
Quinoa Bowl	NULL	3	12.49
Avocado Toast	NULL	4	6.99
Spinach and Feta Omelette	NULL	5	9.99
Mango Tango Smoothie	NULL	6	4.99
Sweet Potato Fries	NULL	7	5.49
Caprese Panini	NULL	8	11.99
Berry Blast Salad	NULL	9	7.99
Greek Yogurt Parfait	NULL	10	3.99
Quinoa Avocado Salad	NULL	11	9.49
Pesto Pasta	NULL	12	13.99
Hummus Platter	NULL	13	8.49
Vegan Chocolate Cake	NULL	14	6.99
Cucumber Mint Lemonade	NULL	15	3.49
Margherita Pizza	NULL	16	12.99
Coconut Curry Tofu	NULL	17	10.49
Chia Seed Pudding	NULL	18	4.49
Green Goddess Smoothie	NULL	19	5.99
Black Bean Burger	NULL	20	8.99

Inserting payment data into Payment table

SQLQuery22.sql - LAPTOP-F6ALERQF\MSSQLSERVER01.Proposal Initial (LAPTOP-F6ALERQF\marti (55)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

SQLQuery22.sql - L...ALERQF\marti (55) *

```

INSERT INTO ORMModel1.Payment (CardHolderName, CVCNumber, CardNumber, BillingAddress, CustomerID)
VALUES
('Sterling Archer', '123', '1111222233334444', '1313 Mockingbird Lane', 1),
('Lana Kane', '456', '2222333344445555', '246 Spy Blvd', 2),
('Cyril Figgis', '789', '3333444455556666', '789 Bureaucrat Avenue', 3),
('Pam Poovey', '321', '4444555566667777', '420 Underground Fight Club', 4),
('Cheryl Tunt', '654', '5555666677778888', '555 Crazy Lane', 5),
('Ray Gillette', '987', '6666777788889999', '910 Artistic Avenue', 6),
('Malory Archer', '234', '7777888899990000', '101 Espionage Road', 7),
('Krieger Algernop', '567', '8888999900001111', '22 Mad Scientist Drive', 8),
('Barry Dylan', '890', '9999000011112222', '13 Cybernetic Street', 9),
('Katya Kazanova', '109', '0000111122223333', '007 KGB Avenue', 10);

```

100 %

Messages

(10 rows affected)

Completion time: 2024-02-22T15:14:46.703382-05:00

100 %

Query executed successfully. LAPTOP-F6ALERQF\MSSQLSERVER... LAPTOP-F6ALERQF\marti ... Proposal Initial 00:00:00 0 rows

Ready Ln 13 Col 1 Ch 1 INS

Payment data in table

SQLQuery23.sql - LAPTOP-F6ALERQF\MSSQLSERVER01.Proposal Initial (LAPTOP-F6ALERQF\marti (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

SQLQuery23.sql - L...ALERQF\marti (58) * SQLQuery22.sql - L...ALERQF\marti (55) *

```

/***** Script for SelectTopNRows command from SMS *****/
SELECT TOP (1000) [paymentID]
,[CardHolderName]
,[CVCNumber]
,[CardNumber]
,[BillingAddress]
,[CustomerID]
FROM [Proposal Initial].[ORMModel1].[Payment]

```

100 %

Results Messages

paymentID	CardHolderName	CVCNumber	CardNumber	BillingAddress	CustomerID
1	Sterling Archer	123	1111222233334444	1313 Mockingbird Lane	1
2	Lana Kane	456	2222333344445555	246 Spy Blvd	2
3	Cyril Figgis	789	3333444455556666	789 Bureaucrat Avenue	3
4	Pam Poovey	321	4444555566667777	420 Underground Fight Club	4
5	Cheryl Tunt	654	5555666677778888	555 Crazy Lane	5
6	Ray Gillette	987	6666777788889999	910 Artistic Avenue	6
7	Malory Archer	234	7777888899990000	101 Espionage Road	7
8	Krieger Algernop	567	8888999900001111	22 Mad Scientist Dnve	8
9	Bary Dylan	890	9999000011112222	13 Cybernetic Street	9
10	Katya Kazanova	109	0000111122223333	007 KGB Avenue	10

100 %

Query executed successfully. LAPTOP-F6ALERQF\MSSQLSERVER... LAPTOP-F6ALERQF\marti ... Proposal Initial 00:00:00 10 rows

Ready Ln 1 Col 1 Ch 1 INS

Inserting order data into Order table

The screenshot shows Microsoft SQL Server Management Studio with a query window open. The query is an INSERT INTO statement for the Order table in the ORMModel1 database. The query inserts 10 rows of data, each with identical values for all columns: orderID, customerID, paymentID, fulfillmentTypeID, and menuItemsID.

```
INSERT INTO ORMModel1.[Order] (customerID, paymentID, fulfillmentTypeID, menuItemsID)
VALUES
(1, 1, 1, 1),
(2, 2, 2, 2),
(3, 3, 3, 3),
(4, 4, 4, 4),
(5, 5, 5, 5),
(6, 6, 6, 6),
(7, 7, 7, 7),
(8, 8, 8, 8),
(9, 9, 9, 9),
(10, 10, 10, 10);
```

The Messages pane below the query window shows the execution results: (10 rows affected) and Completion time: 2024-02-22T15:34:08.7776172-05:00. The status bar at the bottom indicates the query executed successfully.

Order data in table

The screenshot shows Microsoft SQL Server Management Studio with a query window open. The query is a SELECT TOP statement that retrieves the first 10 rows of data from the Order table in the ORMModel1 database. The results are displayed in a table format in the Results pane.

```
/****** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [orderID]
,[customerID]
,[paymentID]
,[fulfillmentTypeID]
,[menuItemsID]
FROM [Proposal Initial].[ORMModel1].[Order]
```

orderID	customerID	paymentID	fulfillmentTypeID	menuItemsID
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10

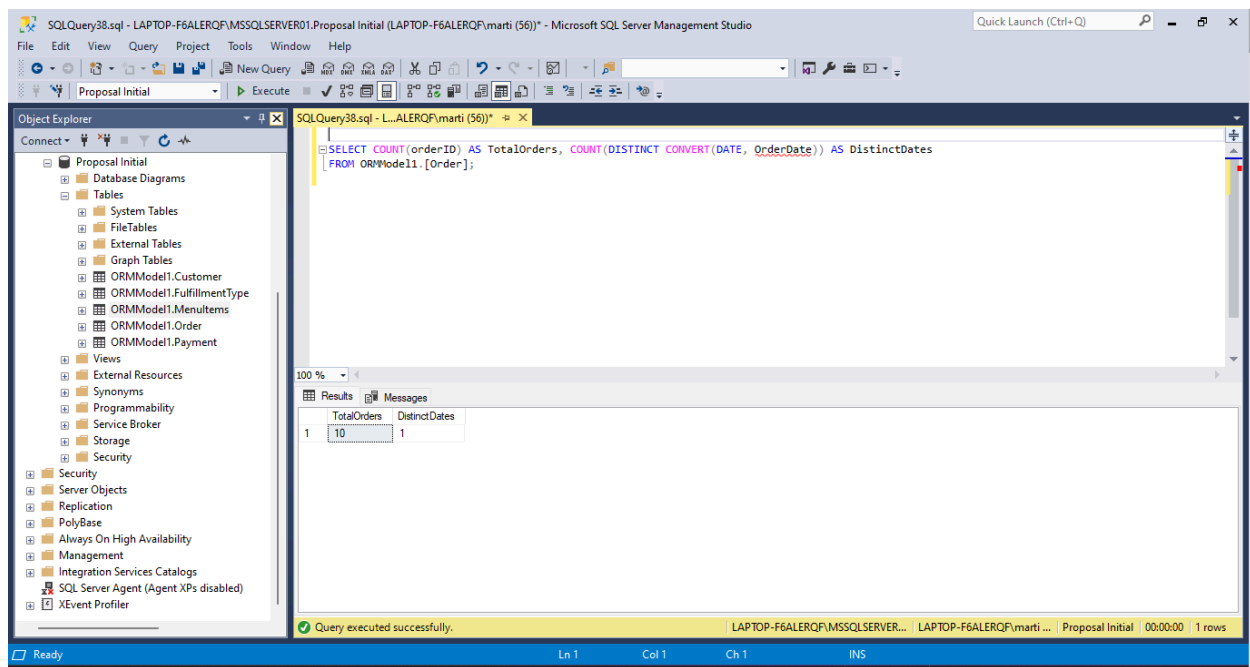
The status bar at the bottom indicates the query executed successfully and that 10 rows were returned.

Answering Incisive Questions Through Statements

Question #1

What is the average daily order volume processed through the online platform?

To answer this question, I used “SELECT” to retrieve data from the Order table. COUNT(orderID) was used to count the total orders in the Order table, and “AS” assigns it the alias “TotalOrders”. “COUNT(DISTINCT CONVERT(DATE, OrderDate)) AS DistinctDates” counts the number of distinct dates by converting the OrderDate to a date data type and then counts them. It is assigned the alias "DistinctDates. As shown in the screenshot, the output of this query is a single row with two columns: “TotalOrder” and “DistinctDates”. I only had 10 orders in my database and the date is the same for each, so the average daily of volume is $10/1=10$ orders per day.

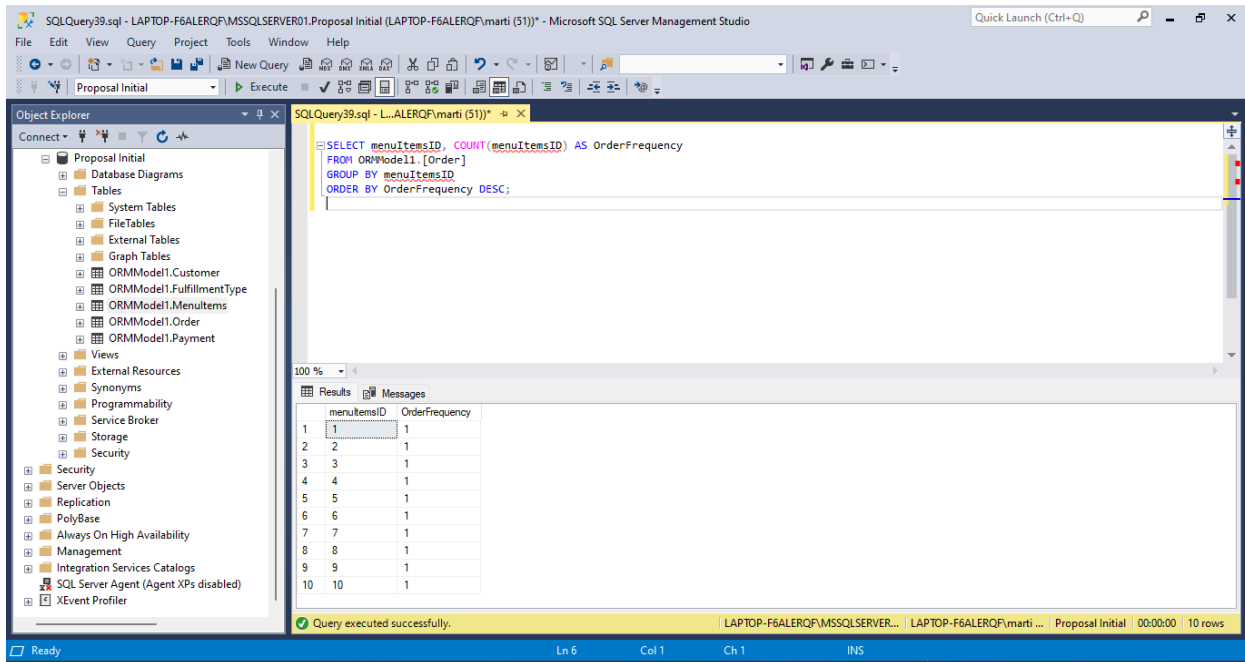


Question #2

Which menu items are most frequently ordered?

I answered this question by using “SELECT menuItemID, COUNT(menuItemID) AS OrderCount” to select the menuItemID and count the occurrences of each menu item ID in the Order table, giving it the alias "OrderCount”. I selected this data from the Order table and used “GROUP BY menuItemID” and “ORDER BY OrderCount DESC” to group the results by the menuItemID and to display the results in descending order based on the OrderCount, so the

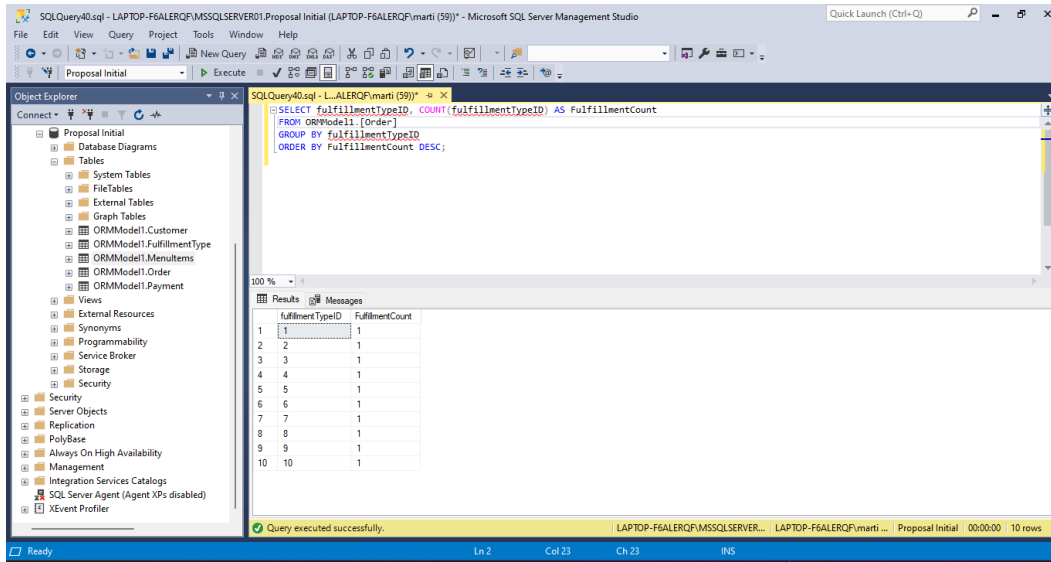
most frequently items are displayed first. In my small sample size, every one of my 10 customers ordered a different menu item, so not one menu item was more popular than the other.



Question #3

Is pick-up or delivery more popular?

I used the statements “SELECT fulfillmentTypeID, COUNT(fulfillmentTypeID) AS FulfillmentCount” to select the fulfillmentTypeID and count the occurrences of each fulfillment type ID in the Order table, giving it the alias "FulfillmentCount." I used “GROUP BY fulfillmentTypeID” to group the results by the fulfillmentTypeID and “ORDER BY FulfillmentCount DESC” to order the results in descending order based on the FulfillmentCount, so the most frequently chosen fulfillment type is shown first. There were an equal number of delivery and pickup orders.



Summary

In this project, I designed and implemented a database model for an online food ordering system within the restaurant industry. The database includes tables for Customers, Fulfillment Types, Menu Items, Orders, and Payments. Each table is populated with realistic data, reflecting the diverse preferences of customers, as inspired by characters from the TV show Archer.

I addressed incisive questions such as determining the average daily order volume, identifying frequently ordered menu items, and analyzing the popularity of pick-up versus delivery. Through SQL queries and data analysis, I provided valuable insights into the customer behavior within the online ordering system. The knowledge showcased includes the ability to create a well-organized database structure, populate tables with meaningful data, and derive insights through SQL queries.

Conclusion

In conclusion, this project represents an initial exploration of an online food ordering system, providing insights such as average daily order volumes and menu preferences. Future analysis could delve deeper into specific customer behaviors, such as peak ordering times and busiest days of the week. This showcases SQL's practicality in uncovering patterns of customer behavior

in the online food service realm. Moving forward, this foundation sets the stage for more nuanced investigations into user preferences and online ordering software optimizations.