

Customer 360 Data Integration



Contents

Project Overview	3
GitHub link	3
Tools required	3
Architecture	4
Bronze/Raw Layer	4
Silver Layer	5
Gold Layer	12
PowerBI	17

Project Overview

The Customer 360 Data Integration project aims to create a unified view of customers for a retail business by integrating data from online transactions, in-store purchases, customer service interactions, and loyalty programs. This comprehensive view helps the business better understand customer behavior, personalize experiences, and support data-driven decisions. The project uses an end-to-end data pipeline built on the Azure ecosystem, including Azure Synapse Analytics, Azure Data Lake Storage (ADLS), Azure SQL Database, and Power BI. It follows a layered architecture—from raw data ingestion to staging, analytics modeling, and final visualization.

Outcomes

Raw Layer: Source data stored in ADLS containers.

Curated Layer (Silver): Cleaned and joined data in Synapse Analytics.

Analytics Layer (Gold): Aggregated tables and customer insights in Azure SQL Database.

Power BI Dashboard: Interactive Customer 360 dashboard visualizing purchasing behaviour, service interactions, and loyalty metrics.

GitHub link

https://github.com/DeepthiChethi/Azure_DE/upload/Madhumita/Project3.

Tools required

ADLSGEN2

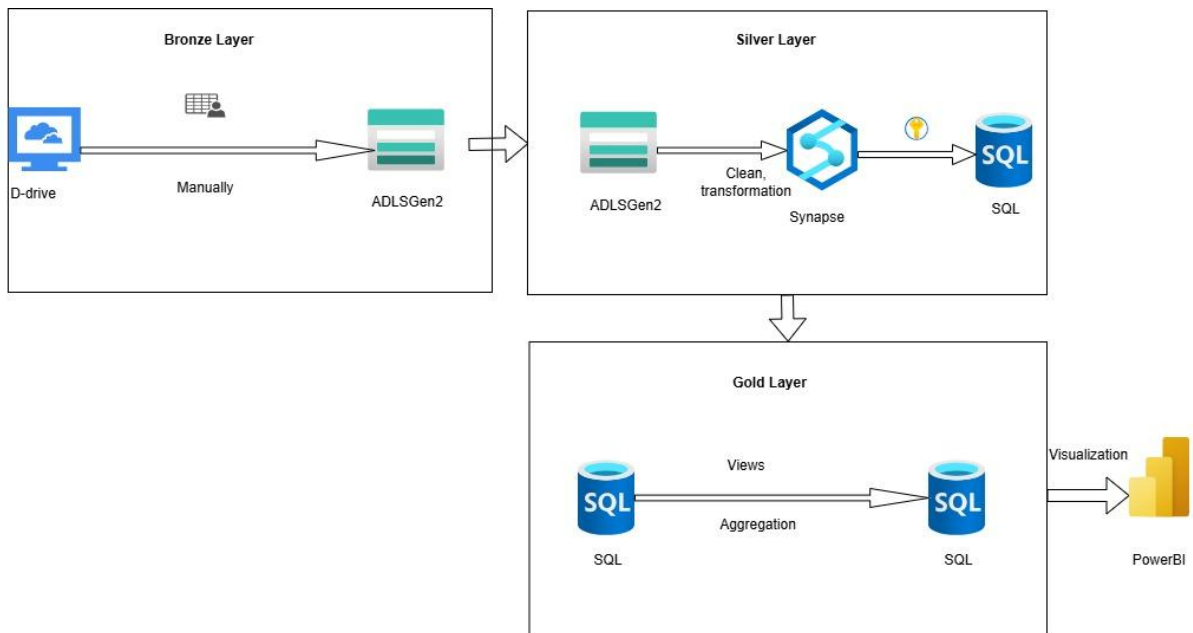
Azure Synapse Analytics

Key Vault

Azure SQL Database

PowerBI

Architecture



Bronze/Raw Layer

Download the 9 files from [Kaggle's Customer 360 Data](#) and save them to the D-drive. Then, manually upload all 9 CSV files to Azure Data Lake Storage Gen2 (ADLS Gen2).

Object Explorer

Connect

- System Tables
- External Tables
- Graph Tables
- dbo.Agents
- dbo.BuildVersion
- dbo.Customers
- dbo.CustomerServiceInteracti...
- dbo.ErrorLog
- dbo.InStoreTransactions
- dbo.LoyaltyAccounts
- dbo.LoyaltyTransactions
- dbo.OnlineTransactions
- dbo.Products
- dbo.Stores
- SalesLT.Address
- SalesLT.Customer
- SalesLT.CustomerAddress
- SalesLT.Product
- SalesLT.ProductCategory
- SalesLT.ProductDescription
- SalesLT.ProductModel
- SalesLT.ProductModelProduct
- SalesLT.SalesOrderDetail
- SalesLT.SalesOrderHeader
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery6.sql - nc..b (ncplmadhu (80))*

```

CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Address VARCHAR(255)
);

Select * from Customers

delete from Customers

CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    Name VARCHAR(100),
    Category VARCHAR(50),
    Price DECIMAL(10, 2)
);

```

99 %

Results Messages

CustomerID	Name	Email	Address
1	Mrs. Crystal Carroll	kaiserjacob@example.org	7566 Kelly Shoals Apt. 207, Port Joanne, FM 25401
2	Debra Newman DVM	brownashley@example.net	72713 Nelson Lodge Suite 286, Leonfort, NJ 49406
3	Thomas Mason	frankmark@example.com	3024 Riley Ferry Suite 573, Chadberg, MT 40822
4	Karla Hill	jnelson@example.org	8899 Hubbard Station, Lake Vanessa, CT 07851
5	Jeffrey Underwood	daniel48@example.org	11764 Hannah Plaza, Lake Vickichester, AR 85240
6	Justin Lowe	qwilson@example.org	494 Danielle Causeway, Lake Benjamin, NV 21217
7	Jason Daniels	elliottsonia@example.net	37283 Ramsey Light Suite 640, New Howard, IA 48396
8	Jennifer Hill	petersenrebecca@example.org	02411 Ruiz Corners Apt. 346, Oconnorberg, SD 93666
9	Stephanie Richardson	obates@example.net	974 Alyssa Heights, Fanellfurt, OH 22597
10	Jo Zimmerman	samuel23@example.net	25747 Clark Harbor, New Emlyport, RI 59995
11	Andrew Williams	robinsonjulan@example.org	431 Thomas Orchard, South Aliciaside, ME 26651
12	Robin Odom	deborah35@example.com	75190 John Ferry Apt. 305, North Renee, FL 93070
13	Gina Anderson	michelle40@example.com	52815 Tristan Oval Suite 453, Malaburgh, NC 59623
14	Devin Smith	georoholmes@example.com	95367 Sara Loaf, New Xavertown, IN 55027

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (78) ncplsqladb 00:00:00 100 rows

Object Explorer

Connect

- System Tables
- External Tables
- Graph Tables
- dbo.Agents
- dbo.BuildVersion
- dbo.Customers
- dbo.CustomerServiceInteracti...
- dbo.ErrorLog
- dbo.InStoreTransactions
- dbo.LoyaltyAccounts
- dbo.LoyaltyTransactions
- dbo.OnlineTransactions
- dbo.Products
- dbo.Stores
- SalesLT.Address
- SalesLT.Customer
- SalesLT.CustomerAddress
- SalesLT.Product
- SalesLT.ProductCategory
- SalesLT.ProductDescription
- SalesLT.ProductModel
- SalesLT.ProductModelProduct
- SalesLT.SalesOrderDetail
- SalesLT.SalesOrderHeader
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery6.sql - nc..b (ncplmadhu (80))*

```

CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    Name VARCHAR(100),
    Category VARCHAR(50),
    Price DECIMAL(10, 2)
);

Select * from Products

CREATE TABLE OnlineTransactions (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    DateTime DATETIME,
    PaymentMethod VARCHAR(50),
    Amount DECIMAL(10, 2),
    Status VARCHAR(20)
);

```

99 %

Results Messages

ProductID	Name	Category	Price
1	Consider	Electronics	183.51
2	At	Toys	403.28
3	Risk	Clothing	186.53
4	Serious	Clothing	275.39
5	Discover	Clothing	23.39
6	Area	Books	316.84
7	Happy	Toys	137.58
8	Effect	Electronics	202.63
9	Star	Books	119.08
10	Structure	Books	273.71
11	Join	Books	253.70
12	Course	Books	298.30
13	Play	Home Goods	478.47
14	Almost	Tovs	143.60

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (74) ncplsqladb 00:00:00 100 rows

Object Explorer

Connect

- System Tables
- External Tables
- Graph Tables
- dbo.Agents
- dbo.BuildVersion
- dbo.Customers
- dbo.CustomerServiceInteracti...
- dbo.ErrorLog
- dbo.InStoreTransactions
- dbo.LoyaltyAccounts
- dbo.LoyaltyTransactions
- dbo.OnlineTransactions
- dbo.Products
- dbo.Stores
- SalesLT.Address
- SalesLT.Customer
- SalesLT.CustomerAddress
- SalesLT.Product
- SalesLT.ProductCategory
- SalesLT.ProductDescription
- SalesLT.ProductModel
- SalesLT.ProductModelProduct
- SalesLT.SalesOrderDetail
- SalesLT.SalesOrderHeader
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery6.sql - nc..b (ncplmadhu (80))*

```

CREATE TABLE OnlineTransactions (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    DateTime DATETIME,
    PaymentMethod VARCHAR(50),
    Amount DECIMAL(10, 2),
    Status VARCHAR(20),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

select * from OnlineTransactions

CREATE TABLE Stores (
    StoreID INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(50),
    State VARCHAR(50),
    Zip VARCHAR(10)
);

```

99 %

Results Messages

OrderID	CustomerID	ProductID	DateTime	PaymentMethod	Amount	Status
1	34	91	2025-01-20 22:34:31.000	Gift Card	189.90	Completed
2	17	59	2025-02-10 20:42:13.000	Gift Card	193.25	Pending
3	13	57	2025-02-16 18:26:04.000	Debit Card	183.82	Pending
4	90	19	2025-02-21 05:04:59.000	Debit Card	175.37	Pending
5	61	99	2025-01-19 05:42:37.000	Credit Card	174.62	Completed
6	75	67	2025-02-26 09:47:16.000	Debit Card	144.48	Pending
7	63	44	2025-02-07 09:48:20.000	PayPal	91.74	Failed
8	55	6	2025-01-19 20:32:41.000	Credit Card	16.58	Failed
9	96	67	2025-02-16 19:22:55.000	Credit Card	22.88	Pending
10	49	13	2025-01-25 02:24:02.000	Credit Card	27.55	Completed
11	6	91	2025-01-09 14:00:29.000	Credit Card	94.31	Failed
12	26	53	2025-01-25 03:04:55.000	PayPal	87.74	Pending
13	1	28	2025-01-22 10:54:11.000	Credit Card	84.75	Completed
14	60	78	2025-01-28 21:19:38.000	Credit Card	162.58	Pending

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (74) ncplsqladb 00:00:00 100 rows

Object Explorer

- System Tables
- External Tables
- Graph Tables
- dbo.Agents
- dbo.BuildVersion
- dbo.Customers
- dbo.CustomerServiceInteracti
- dbo.ErrorLog
- dbo.InStoreTransactions
- dbo.LoyaltyAccounts
- dbo.LoyaltyTransactions
- dbo.OnlineTransactions
- dbo.Products
- dbo.Stores
- SalesLT.Address
- SalesLT.Customer
- SalesLT.CustomerAddress
- SalesLT.Product
- SalesLT.ProductCategory
- SalesLT.ProductDescription
- SalesLT.ProductModel
- SalesLT.ProductModelProduct
- SalesLT.SalesOrderDetail
- SalesLT.SalesOrderHeader
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery6.sql - nc...b (ncplmadhu (80))*

```

CREATE TABLE InStoreTransactions (
    TransactionID INT PRIMARY KEY,
    CustomerID INT,
    StoreID INT,
    DateTime DATETIME,
    Amount DECIMAL(10, 2),
    PaymentMethod VARCHAR(50),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (StoreID) REFERENCES Stores(StoreID)
);

select * from InStoreTransactions
  
```

SQLQuery7.sql - nc...b (ncplmadhu (78))*

```

CREATE TABLE Agents (
    AgentID INT PRIMARY KEY,
    Name VARCHAR(100),
    Department VARCHAR(50),
    Shift VARCHAR(50)
);

select * from Agents
  
```

SQLQuery3.sql - nc...b (ncplmadhu (74))*

99 %

TransactionID	CustomerID	StoreID	DateTime	Amount	PaymentMethod
1	88	100	2025-03-12 06:49:37.000	135.74	Credit Card
2	60	12	2025-01-21 23:09:14.000	34.14	Credit Card
3	54	51	2025-01-29 22:12:51.000	111.42	Debit Card
4	32	94	2025-02-08 23:51:36.000	179.45	Debit Card
5	33	45	2025-01-18 04:05:21.000	177.96	Gift Card
6	92	91	2025-03-09 03:29:17.000	196.51	Debit Card
7	44	10	2025-02-19 19:28:55.000	198.12	Debit Card
8	91	51	2025-02-15 14:38:42.000	19.48	Gift Card
9	76	77	2025-01-19 09:16:43.000	196.53	Debit Card
10	65	12	2025-01-30 06:43:44.000	83.99	Gift Card
11	13	77	2025-01-29 21:46:00.000	180.81	Credit Card
12	89	89	2025-01-15 18:29:30.000	88.37	Credit Card
13	39	87	2025-01-20 06:41:32.000	109.12	Credit Card
14	73	60	2025-02-12 06:07:00.000	77.49	Gift Card

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (74) ncplsqladb 00:00:00 100 rows

Object Explorer

- System Tables
- External Tables
- Graph Tables
- dbo.Agents
- dbo.BuildVersion
- dbo.Customers
- dbo.CustomerServiceInteracti
- dbo.ErrorLog
- dbo.InStoreTransactions
- dbo.LoyaltyAccounts
- dbo.LoyaltyTransactions
- dbo.OnlineTransactions
- dbo.Products
- dbo.Stores
- SalesLT.Address
- SalesLT.Customer
- SalesLT.CustomerAddress
- SalesLT.Product
- SalesLT.ProductCategory
- SalesLT.ProductDescription
- SalesLT.ProductModel
- SalesLT.ProductModelProduct
- SalesLT.SalesOrderDetail
- SalesLT.SalesOrderHeader
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery6.sql - nc...b (ncplmadhu (80))*

```

CREATE TABLE Stores (
    StoreID INT PRIMARY KEY,
    Location VARCHAR(100),
    Manager VARCHAR(100),
    OpenHours VARCHAR(50)
);

select * from Stores
  
```

SQLQuery7.sql - nc...b (ncplmadhu (78))*

```

CREATE TABLE InStoreTransactions (
    TransactionID INT PRIMARY KEY,
    CustomerID INT,
    StoreID INT,
    DateTime DATETIME,
    Amount DECIMAL(10, 2),
    PaymentMethod VARCHAR(50)
);

select * from InStoreTransactions
  
```

SQLQuery3.sql - nc...b (ncplmadhu (74))*

99 %

StoreID	Location	Manager	OpenHours
1	Anglemouth	Michelle Robinson	8:00 AM - 8:00 PM
2	Tammychester	Jonathan Fry	8:00 AM - 8:00 PM
3	Vanesschester	Barbara Lynch	8:00 AM - 7:00 PM
4	Maytown	Frederick Dean	9:00 AM - 8:00 PM
5	Kathleenfort	Donald Ellis	10:00 AM - 7:00 PM
6	South Martin	Tamara Hernandez	10:00 AM - 6:00 PM
7	Rebeccastad	Amanda Diaz	9:00 AM - 8:00 PM
8	North Mariaburgh	Lisa Jackson	9:00 AM - 7:00 PM
9	Stephenfurt	Michael Hays	10:00 AM - 9:00 PM
10	Thompsonshire	Mary Barker	9:00 AM - 9:00 PM
11	East Justin	Jose Brown	9:00 AM - 8:00 PM
12	Jeffreyport	Roger Walsh	8:00 AM - 9:00 PM
13	New James	Crystal Choi	10:00 AM - 9:00 PM
14	Josephfort	Ryan King	8:00 AM - 6:00 PM

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (74) ncplsqladb 00:00:00 100 rows

Object Explorer

- System Tables
- External Tables
- Graph Tables
- dbo.Agents
- dbo.BuildVersion
- dbo.Customers
- dbo.CustomerServiceInteracti
- dbo.ErrorLog
- dbo.InStoreTransactions
- dbo.LoyaltyAccounts
- dbo.LoyaltyTransactions
- dbo.OnlineTransactions
- dbo.Products
- dbo.Stores
- SalesLT.Address
- SalesLT.Customer
- SalesLT.CustomerAddress
- SalesLT.Product
- SalesLT.ProductCategory
- SalesLT.ProductDescription
- SalesLT.ProductModel
- SalesLT.ProductModelProduct
- SalesLT.SalesOrderDetail
- SalesLT.SalesOrderHeader
- Dropped Ledger Tables
- Views
- External Resources
- Synonyms
- Programmability

SQLQuery6.sql - nc...b (ncplmadhu (80))*

```

CREATE TABLE Agents (
    AgentID INT PRIMARY KEY,
    Name VARCHAR(100),
    Department VARCHAR(50),
    Shift VARCHAR(50)
);

select * from Agents
  
```

SQLQuery7.sql - nc...b (ncplmadhu (78))*

```

CREATE TABLE CustomerServiceInteractions (
    InteractionID INT PRIMARY KEY,
    CustomerID INT,
    DateTime DATETIME,
    AgentID INT,
    IssueType VARCHAR(50),
    ResolutionStatus VARCHAR(50),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (AgentID) REFERENCES Agents(AgentID)
);

select * from CustomerServiceInteractions
  
```

SQLQuery3.sql - nc...b (ncplmadhu (74))*

99 %

AgentID	Name	Department	Shift
1	Jonathan Williams	General Inquiry	Morning
2	Tony Edwards	Billing	Evening
3	Garrett Knapp	Sales	Morning
4	Daryl Benjamin	Sales	Evening
5	Matthew Long	Sales	Morning
6	Patricia Rhodes	Sales	Morning
7	Elizabeth James	Technical Support	Morning
8	Teresa Bennett	Technical Support	Morning
9	Amber Ross	Billing	Afternoon
10	Tonya Jones	Technical Support	Evening
11	Curtis McBride	Sales	Afternoon
12	Justin Michael	Billing	Morning
13	Scott Flowers	Billing	Evening
14	Michael Jackson	Technical Support	Afternoon

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (74) ncplsqladb 00:00:00 100 rows

Object Explorer

Connect

System Tables
External Tables
Graph Tables
dbo.Agents
dbo.BuildVersion
dbo.Customers
dbo.CustomerServiceInteractions
dbo.ErrorLog
dbo.InStoreTransactions
dbo.LoyaltyAccounts
dbo.LoyaltyTransactions
dbo.OnlineTransactions
dbo.Products
dbo.Stores
SalesLT.Address
SalesLT.Customer
SalesLT.CustomerAddress
SalesLT.Product
SalesLT.ProductCategory
SalesLT.ProductDescription
SalesLT.ProductModel
SalesLT.ProductModelProduct
SalesLT.SalesOrderDetail
SalesLT.SalesOrderHeader
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability

SQLQuery6.sql - nc..b (ncplmadhu (80))*

SQLQuery7.sql - nc..b (ncplmadhu (78))*

SQLQuery3.sql - nc..b (ncplmadhu (74))*

```

CREATE TABLE CustomerServiceInteractions (
    InteractionID INT PRIMARY KEY,
    CustomerID INT,
    DateTime DATETIME,
    AgentID INT,
    IssueType VARCHAR(50),
    ResolutionStatus VARCHAR(50),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (AgentID) REFERENCES Agents(AgentID)
);

select * from CustomerServiceInteractions

CREATE TABLE LoyaltyAccounts (
    LoyaltyID INT PRIMARY KEY,
    CustomerID INT,
    PointsEarned INT,
    TierLevel VARCHAR(20),
    JoinDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

select * from LoyaltyAccounts

CREATE TABLE LoyaltyTransactions (
    LoyaltyID INT,
    DateTime DATETIME,
    PointsChange INT,
    Reason VARCHAR(100),
    PRIMARY KEY (LoyaltyID, DateTime),
    FOREIGN KEY (LoyaltyID) REFERENCES LoyaltyAccounts(LoyaltyID)
);

select * from LoyaltyTransactions

```

99 %

Results Messages

InteractionID	CustomerID	DateTime	AgentID	IssueType	ResolutionStatus
1	55	2025-01-20 08:17:26.000	33	Technical Issue	Resolved
2	93	2025-01-16 18:52:51.000	89	Technical Issue	Escalated
3	53	2025-03-16 13:10:05.000	96	Complaint	Resolved
4	30	2025-02-27 13:45:03.000	20	Other	Pending
5	49	2025-02-16 07:56:45.000	48	Technical Issue	Pending
6	42	2025-02-13 21:05:41.000	56	Product Inquiry	Resolved
7	73	2025-03-01 13:19:50.000	38	Other	Escalated
8	50	2025-01-15 16:44:36.000	66	Other	Escalated
9	63	2025-03-11 10:58:12.000	32	Complaint	Pending
10	21	2025-03-15 05:00:10.000	78	Product Inquiry	Escalated
11	69	2025-01-03 21:23:28.000	4	Complaint	Escalated
12	85	2025-02-17 10:30:24.000	34	Billing	Escalated
13	23	2025-02-09 04:38:22.000	1	Product Inquiry	Pending
14	24	2025-02-23 02:23:24.000	27	Technical Issue	Resolved

Query executed successfully.

ncplmadhusqlb.database.win... ncplmadhu (74) ncplsqldb 00:00:00 100 rows

Object Explorer

Connect

System Tables
External Tables
Graph Tables
dbo.Agents
dbo.BuildVersion
dbo.Customers
dbo.CustomerServiceInteractions
dbo.ErrorLog
dbo.InStoreTransactions
dbo.LoyaltyAccounts
dbo.LoyaltyTransactions
dbo.OnlineTransactions
dbo.Products
dbo.Stores
SalesLT.Address
SalesLT.Customer
SalesLT.CustomerAddress
SalesLT.Product
SalesLT.ProductCategory
SalesLT.ProductDescription
SalesLT.ProductModel
SalesLT.ProductModelProduct
SalesLT.SalesOrderDetail
SalesLT.SalesOrderHeader
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability

SQLQuery6.sql - nc..b (ncplmadhu (80))*

SQLQuery7.sql - nc..b (ncplmadhu (78))*

SQLQuery3.sql - nc..b (ncplmadhu (74))*

```

CREATE TABLE LoyaltyAccounts (
    LoyaltyID INT PRIMARY KEY,
    CustomerID INT,
    PointsEarned INT,
    TierLevel VARCHAR(20),
    JoinDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

select * from LoyaltyAccounts

CREATE TABLE LoyaltyTransactions (
    LoyaltyID INT,
    DateTime DATETIME,
    PointsChange INT,
    Reason VARCHAR(100),
    PRIMARY KEY (LoyaltyID, DateTime),
    FOREIGN KEY (LoyaltyID) REFERENCES LoyaltyAccounts(LoyaltyID)
);

select * from LoyaltyTransactions

```

99 %

Results Messages

LoyaltyID	CustomerID	PointsEarned	TierLevel	JoinDate
1	5	399	Gold	2023-11-27
2	17	606	Gold	2021-06-26
3	13	4782	Gold	2023-04-11
4	62	1824	Platinum	2020-12-17
5	84	894	Gold	2020-03-18
6	68	4291	Platinum	2020-09-25
7	90	2845	Bronze	2022-09-26
8	51	1697	Silver	2022-04-14
9	5	1012	Silver	2023-07-03
10	69	3846	Bronze	2024-09-28
11	53	2155	Silver	2022-04-24
12	86	4396	Bronze	2020-11-01
13	36	2411	Silver	2024-07-05
14	34	3519	Gold	2020-09-12

Query executed successfully.

ncplmadhusqlb.database.win... ncplmadhu (74) ncplsqldb 00:00:00 100 rows

Object Explorer

Connect

System Tables
External Tables
Graph Tables
dbo.Agents
dbo.BuildVersion
dbo.Customers
dbo.CustomerServiceInteractions
dbo.ErrorLog
dbo.InStoreTransactions
dbo.LoyaltyAccounts
dbo.LoyaltyTransactions
dbo.OnlineTransactions
dbo.Products
dbo.Stores
SalesLT.Address
SalesLT.Customer
SalesLT.CustomerAddress
SalesLT.Product
SalesLT.ProductCategory
SalesLT.ProductDescription
SalesLT.ProductModel
SalesLT.ProductModelProduct
SalesLT.SalesOrderDetail
SalesLT.SalesOrderHeader
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability

SQLQuery6.sql - nc..b (ncplmadhu (80))*

SQLQuery7.sql - nc..b (ncplmadhu (78))*

SQLQuery3.sql - nc..b (ncplmadhu (74))*

```

CREATE TABLE LoyaltyTransactions (
    LoyaltyID INT,
    DateTime DATETIME,
    PointsChange INT,
    Reason VARCHAR(100),
    PRIMARY KEY (LoyaltyID, DateTime),
    FOREIGN KEY (LoyaltyID) REFERENCES LoyaltyAccounts(LoyaltyID)
);

select * from LoyaltyTransactions

```

99 %

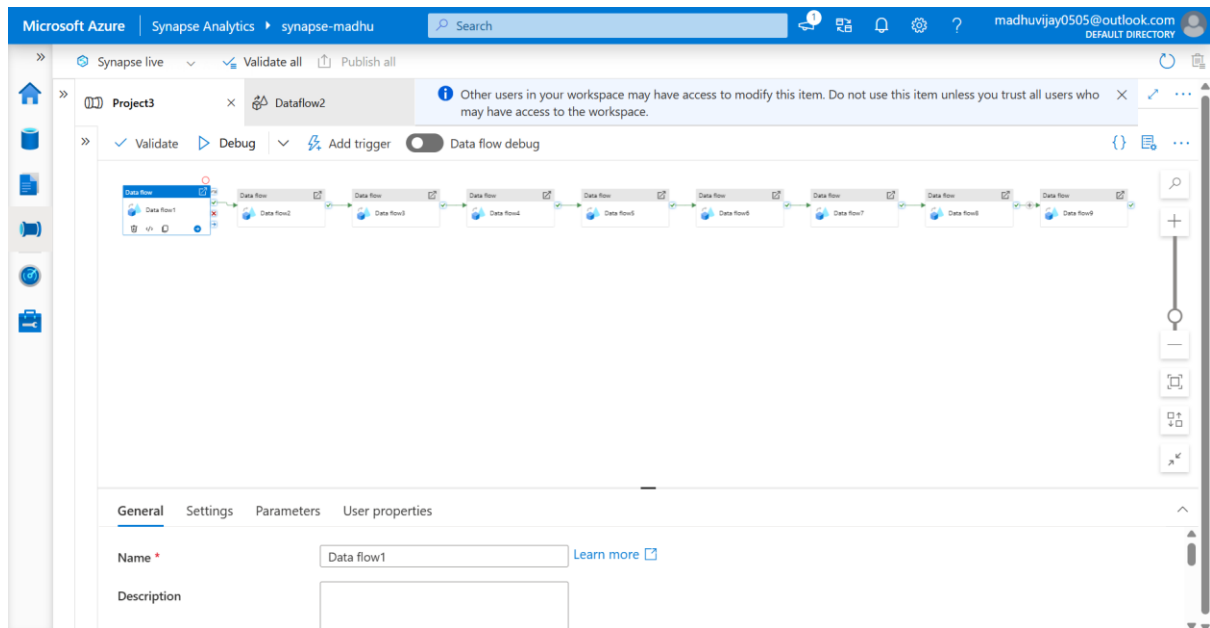
Results Messages

LoyaltyID	DateTime	PointsChange	Reason
1	2025-01-13 20:04:21.000	166	Purchase
2	2025-02-28 18:14:23.000	190	Promotion
3	2025-01-12 22:56:22.000	29	Referral
4	2025-02-21 03:46:32.000	122	Referral
5	2025-02-24 08:12:28.000	186	Correction
6	2025-02-25 21:36:08.000	179	Referral
7	2025-01-03 14:46:28.000	182	Correction
8	2025-01-09 07:30:47.000	172	Promotion
9	2025-01-17 01:13:54.000	108	Referral
10	2025-03-15 21:15:07.000	-34	Referral
11	2025-02-18 09:27:33.000	-6	Purchase
12	2025-02-27 03:25:47.000	159	Purchase
13	2025-03-15 14:39:12.000	-33	Promotion
14	2025-02-03 01:13:19.000	39	Purchase

Query executed successfully.

ncplmadhusqlb.database.win... ncplmadhu (74) ncplsqldb 00:00:00 64 rows

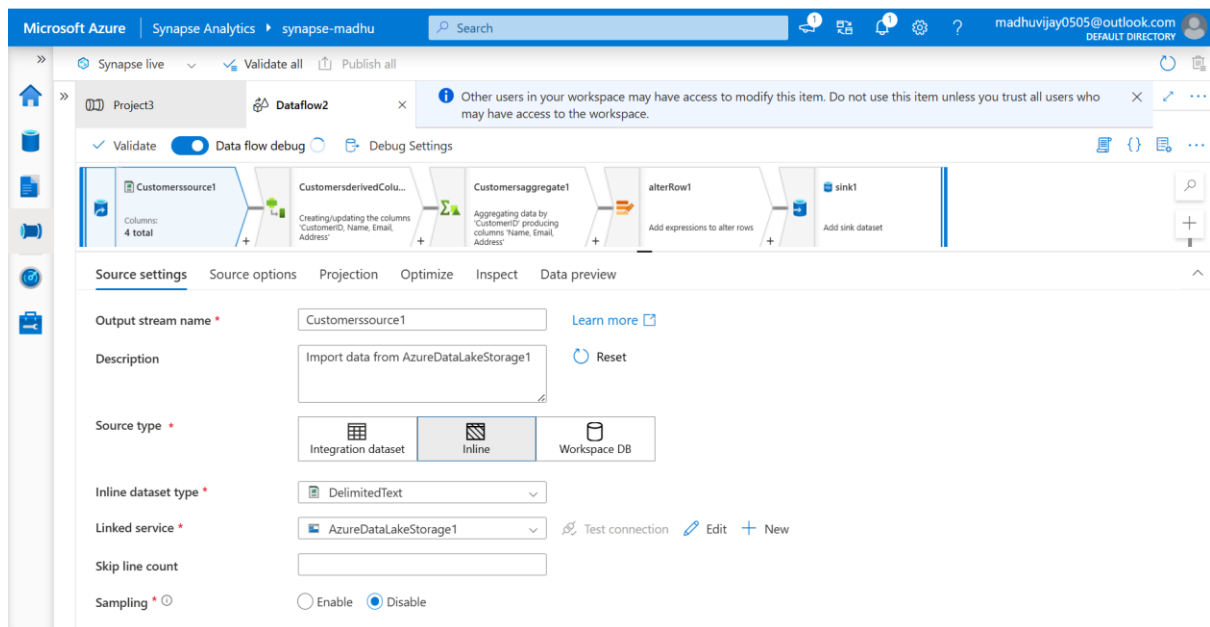
Create dataflows in Azure Synapse Analytics to get data from bronze layer and clean, transformation the data and load it into Azure SQL database.

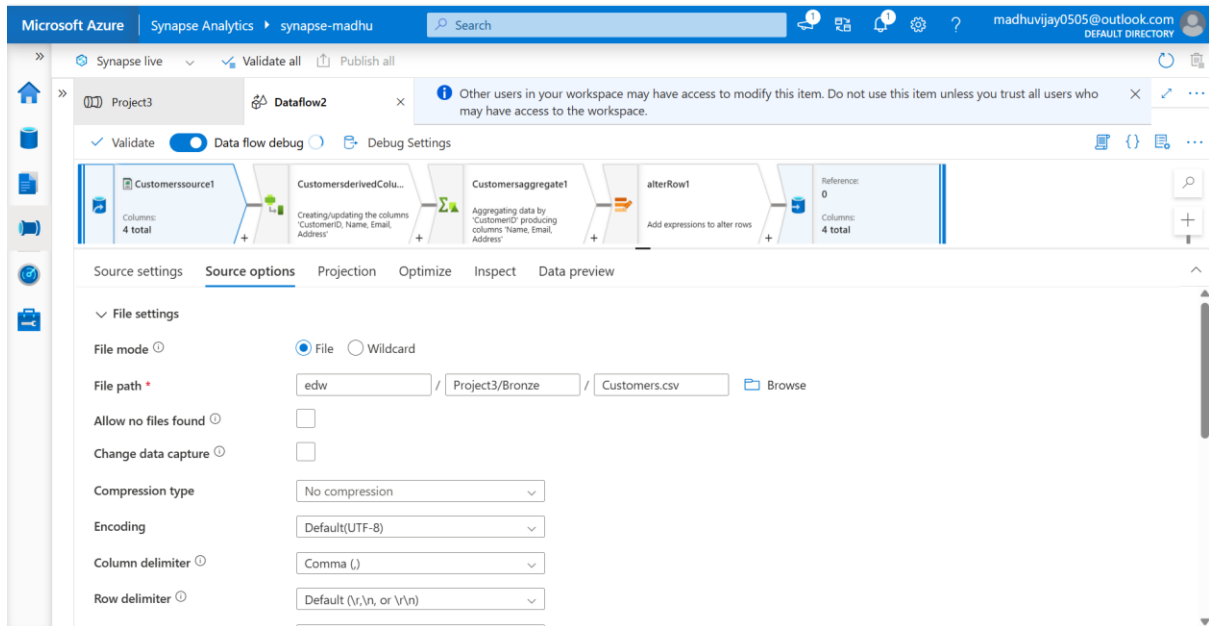


Customer File

Source

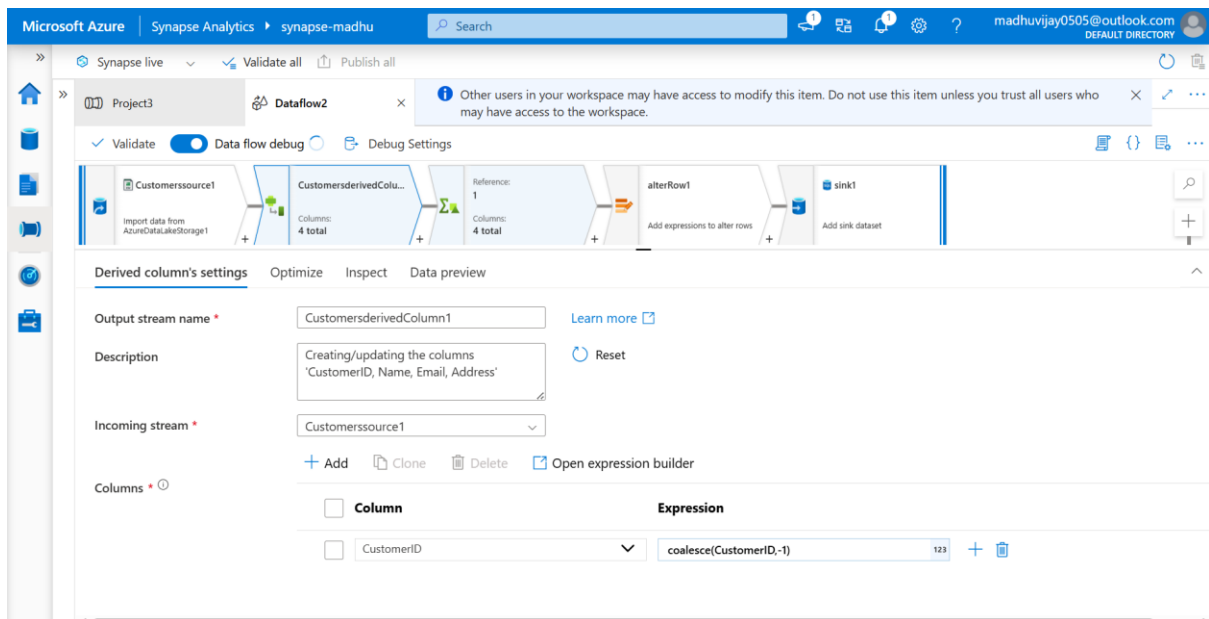
To get data from ADLSGen2, we use source activity





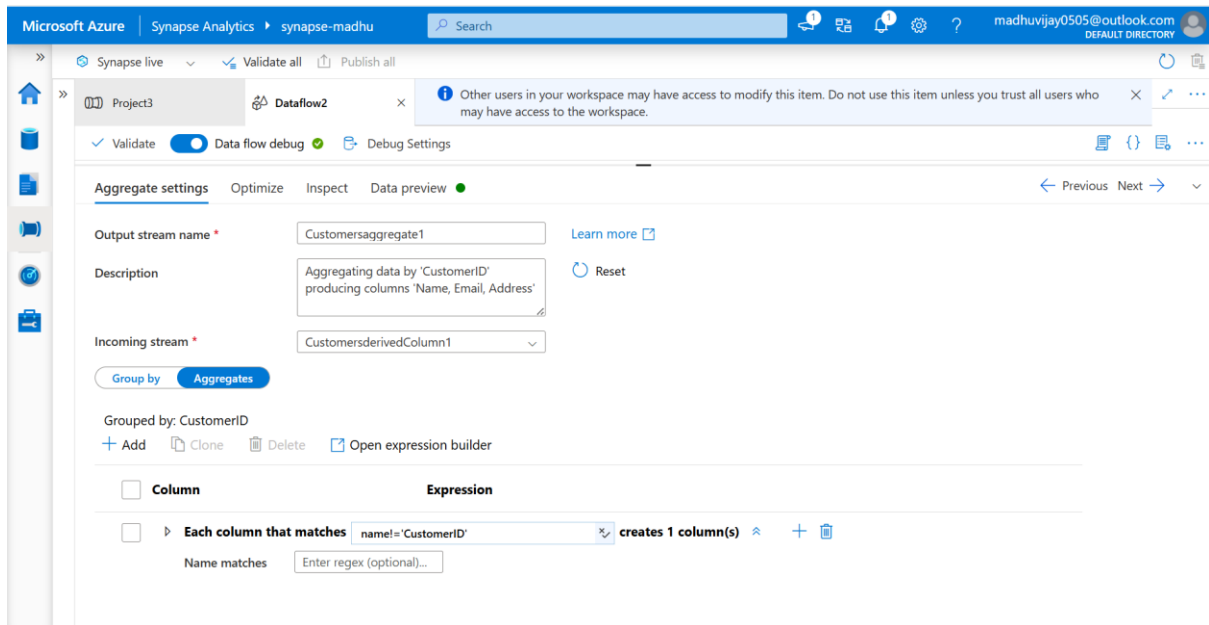
Derived Activity

To remove null values, we are using derived activity



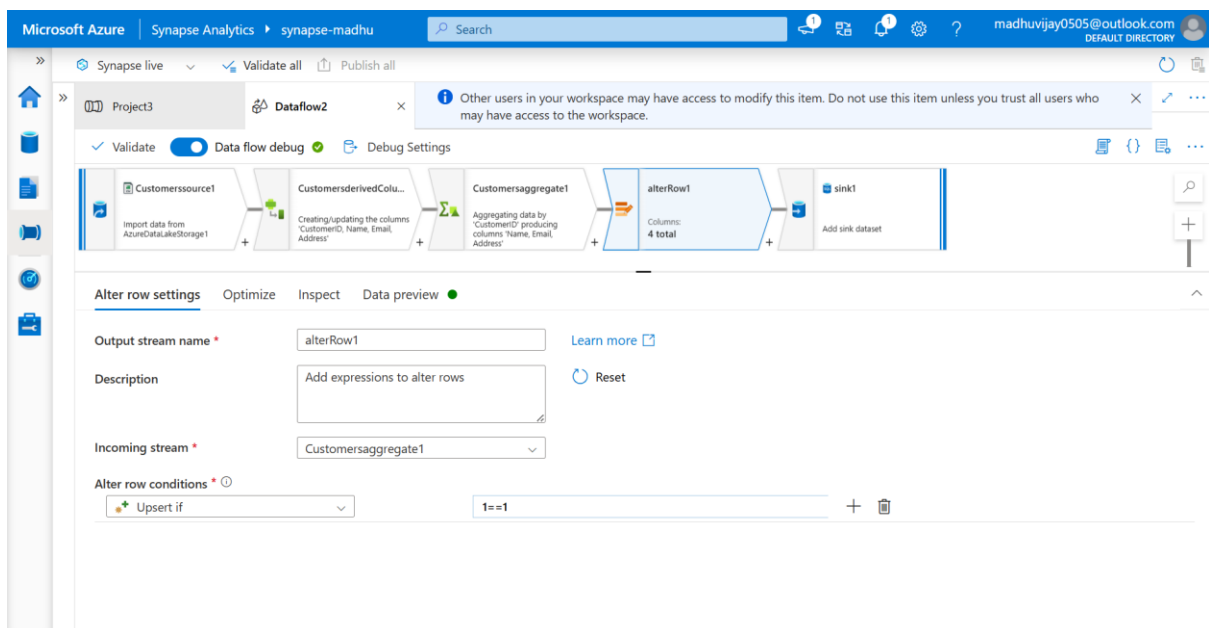
Aggregate Activity

To remove the duplicate values, we are using aggregate activity.



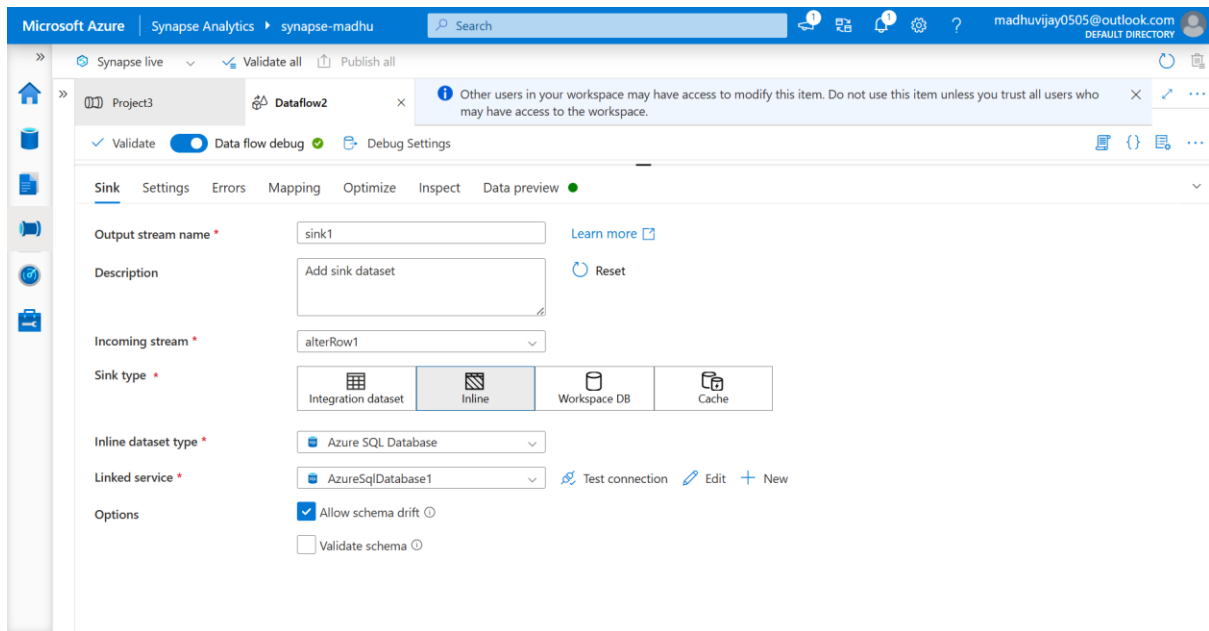
AlterRow

To set row policies, we use AlterRow



Sink

Loading data to SQL database, we use sink activity



Repeat the same steps for other files too.

Gold Layer

Views-1

```
CREATE VIEW View_AverageOrderValue AS
SELECT
    p.ProductID,
    p.Name AS ProductName,
    p.Category,
    'Online' AS Location,
    SUM(t.Amount)/ COUNT(t.OrderID) AS AverageOrderValue
FROM
    OnlineTransactions t
Inner JOIN
    Products p ON t.ProductID = p.ProductID
GROUP BY
    p.ProductID, p.Name, p.Category;
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure. The central query window contains the following SQL code:

```

SELECT
    p.ProductID,
    p.Name AS ProductName,
    p.Category,
    'Online' AS Location,
    SUM(t.Amount) / COUNT(t.OrderID) AS AverageOrderValue
FROM
    OnlineTransactions t
Inner JOIN
    Products p ON t.ProductID = p.ProductID
GROUP BY
    p.ProductID, p.Name, p.Category;

Select * from View_AverageOrderValue

Drop view View_AverageOrderValue

```

Below the query, the Results pane shows the output of the query, which is a table with 5 columns: ProductID, ProductName, Category, Location, and AverageOrderValue. The table contains 14 rows of data.

ProductID	ProductName	Category	Location	AverageOrderValue
1	Consider	Electronics	Online	49.780000
2	Alt	Toys	Online	159.130000
3	Area	Books	Online	114.783333
4	Happy	Toys	Online	174.170000
5	Effect	Electronics	Online	91.920000
6	Play	Home Goods	Online	87.520000
7	Almost	Toys	Online	178.320000
8	Statement	Books	Online	129.130000
9	No	Electronics	Online	153.630000
10	Early	Electronics	Online	175.370000
11	Talk	Electronics	Online	114.120000
12	Involve	Clothing	Online	88.090000
13	Ahead	Home Goods	Online	154.135000
14	Find	Home Goods	Online	68.880000

The status bar at the bottom indicates that the query was executed successfully and returned 66 rows.

VIEW 2

CREATE VIEW View_CustomerSegmentation AS

WITH CustomerSpending AS (

SELECT

c.CustomerID,

c.Name,

SUM(t.Amount) AS TotalSpend,

COUNT(t.OrderID) AS PurchaseFrequency,

l.TierLevel

FROM

dbo.Customers c

LEFT JOIN

OnlineTransactions t ON c.CustomerID = t.CustomerID

LEFT JOIN

LoyaltyAccounts l ON c.CustomerID = l.CustomerID

GROUP BY

c.CustomerID, c.Name, l.TierLevel

)

SELECT

CustomerID,

Name,

TotalSpend,

PurchaseFrequency,

TierLevel,

CASE

WHEN TotalSpend >= (SELECT PERCENTILE_CONT(0.9) WITHIN GROUP (ORDER BY TotalSpend) OVER ()) THEN 'High-Value Customer'

WHEN PurchaseFrequency = 1 THEN 'One-Time Buyer'

WHEN TierLevel = 'Gold' OR TierLevel = 'Platinum' THEN 'Loyalty Champion'

ELSE 'Regular Customer'

END AS CustomerSegment

FROM

CustomerSpending;

The screenshot shows a SQL Server Enterprise Manager interface. On the left is the Object Explorer showing a database structure with various tables and views. The main window displays a SQL query in a query editor. The query is a SELECT statement that retrieves customer information and segments them based on spending, purchase frequency, and tier level. Below the query editor, the 'Results' pane shows the output of the query as a table with 123 rows. The status bar at the bottom indicates the query was executed successfully.

```
SELECT
    CustomerID,
    Name,
    TotalSpend,
    PurchaseFrequency,
    TierLevel,
    CASE
        WHEN TotalSpend >= (SELECT PERCENTILE_CONT(0.9) WITHIN GROUP (ORDER BY TotalSpend) OVER ()) THEN 'High-Value Customer'
        WHEN PurchaseFrequency = 1 THEN 'One-Time Buyer'
        WHEN TierLevel = 'Gold' OR TierLevel = 'Platinum' THEN 'Loyalty Champion'
        ELSE 'Regular Customer'
    END AS CustomerSegment
FROM
    CustomerSpending;
```

CustomerID	Name	TotalSpend	PurchaseFrequency	TierLevel	CustomerSegment	
36	1	Mrs. Crystal Carroll	84.75	1	Bronze	High-Value Customer
37	10	Jo Zimmerman	NULL	0	Bronze	Regular Customer
38	12	Robin Odom	315.70	2	Bronze	High-Value Customer
39	17	Karen Roberts	193.25	1	Bronze	High-Value Customer
40	28	Vanessa Anderson	NULL	0	Bronze	Regular Customer
41	29	Dennis Davis	NULL	0	Bronze	Regular Customer
42	34	Megan Nolan	211.99	2	Bronze	High-Value Customer
43	41	Erin Barrett	29.44	2	Bronze	High-Value Customer
44	43	Jose Jackson	17.38	1	Bronze	High-Value Customer
45	49	Austin Marshall	27.55	1	Bronze	High-Value Customer
46	51	Tyler Knight	460.89	3	Bronze	High-Value Customer
47	69	Mrs. Cassandra Ford	NULL	0	Bronze	Regular Customer
48	73	Linda Boyd	295.12	2	Bronze	High-Value Customer
49	83	Charles Smith	49.44	1	Bronze	High-Value Customer

VIEW 3

CREATE VIEW View_PeakTrafficAnalysis AS

SELECT

DATENAME(WEEKDAY, DateTime) AS DayOfWeek,

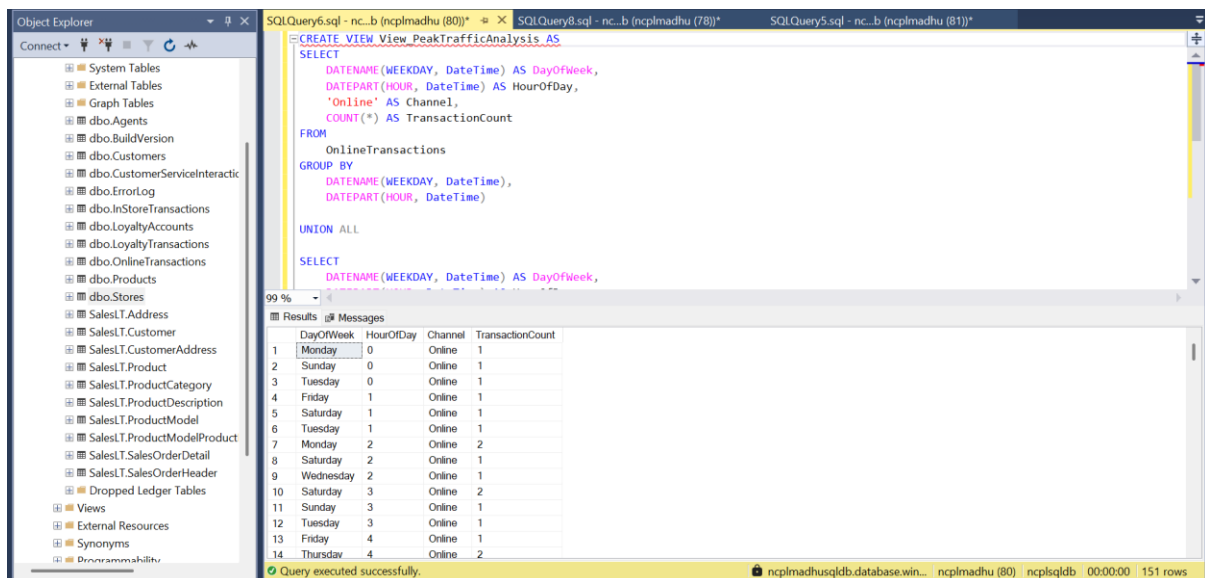
DATEPART(HOUR, DateTime) AS HourOfDay,

'Online' AS Channel,

```

COUNT(*) AS TransactionCount
FROM
    OnlineTransactions
GROUP BY
    DATENAME(WEEKDAY, DateTime),
    DATEPART(HOUR, DateTime)
UNION ALL
SELECT
    DATENAME(WEEKDAY, DateTime) AS DayOfWeek,
    DATEPART(HOUR, DateTime) AS HourOfDay,
    'In-Store' AS Channel,
    COUNT(*) AS TransactionCount
FROM
    InStoreTransactions
GROUP BY
    DATENAME(WEEKDAY, DateTime),
    DATEPART(HOUR, DateTime);

```



VIEW 4


```

CREATE VIEW View_AgentInteractionStats AS

SELECT

    a.AgentID,

    a.Name AS AgentName,

    a.Department,

    a.Shift,

    COUNT(csi.InteractionID) AS TotalInteractions,

    SUM(CASE WHEN csi.ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) AS
ResolvedCount,

    CAST(

        100.0 * SUM(CASE WHEN csi.ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END)

        / NULLIF(COUNT(csi.InteractionID), 0) AS DECIMAL(5,2)

    ) AS ResolutionRatePercent

FROM

    Agents a

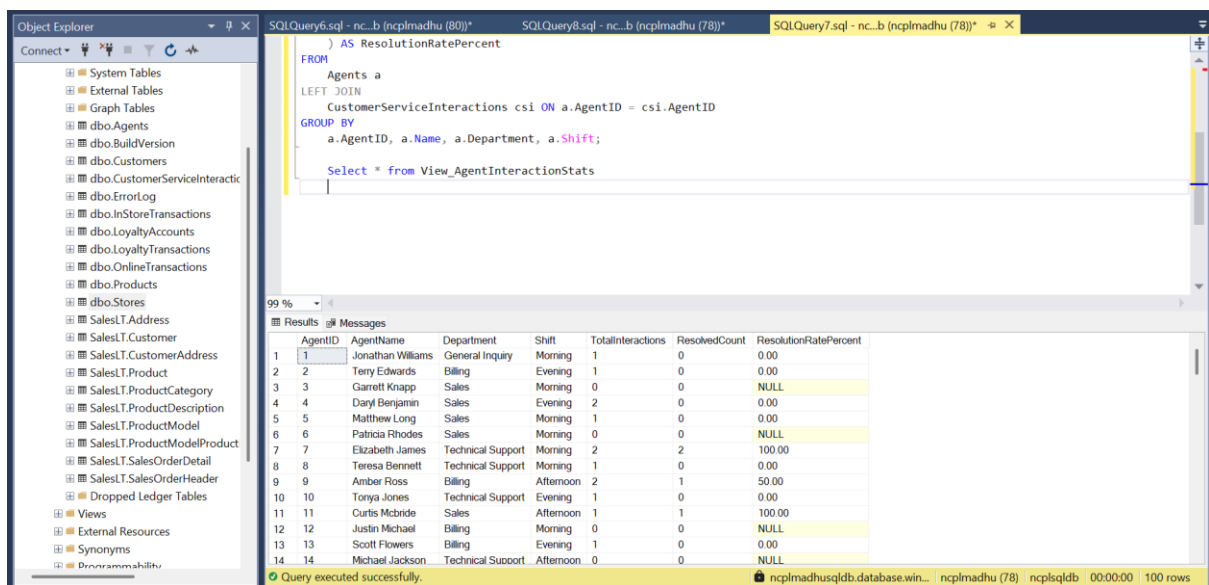
LEFT JOIN

    CustomerServiceInteractions csi ON a.AgentID = csi.AgentID

GROUP BY

    a.AgentID, a.Name, a.Department, a.Shift;

```



The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the database structure. The right pane shows the SQL query and the results grid.

SQL Query:

```

) AS ResolutionRatePercent
FROM
    Agents a
LEFT JOIN
    CustomerServiceInteractions csi ON a.AgentID = csi.AgentID
GROUP BY
    a.AgentID, a.Name, a.Department, a.Shift;
Select * from View_AgentInteractionStats

```

Results:

AgentID	AgentName	Department	Shift	TotalInteractions	ResolvedCount	ResolutionRatePercent
1	Jonathan Williams	General Inquiry	Morning	1	0	0.00
2	Terry Edwards	Billing	Evening	1	0	0.00
3	Garrett Knapp	Sales	Morning	0	0	NULL
4	Daryl Benjamin	Sales	Evening	2	0	0.00
5	Matthew Long	Sales	Morning	1	0	0.00
6	Patricia Rhodes	Sales	Morning	0	0	NULL
7	Elizabeth James	Technical Support	Morning	2	2	100.00
8	Teresa Bennett	Technical Support	Morning	1	0	0.00
9	Amber Ross	Billing	Afternoon	2	1	50.00
10	Tonya Jones	Technical Support	Evening	1	0	0.00
11	Curtis McBride	Sales	Afternoon	1	1	100.00
12	Justin Michael	Billing	Morning	0	0	NULL
13	Scott Flowers	Billing	Evening	1	0	0.00
14	Michael Jackson	Technical Support	Afternoon	0	0	NULL

Query executed successfully. ncplmadhusqladb.database.win... ncplmadhu (78) ncplsqladb 00:00:00 100 rows

PowerBI

