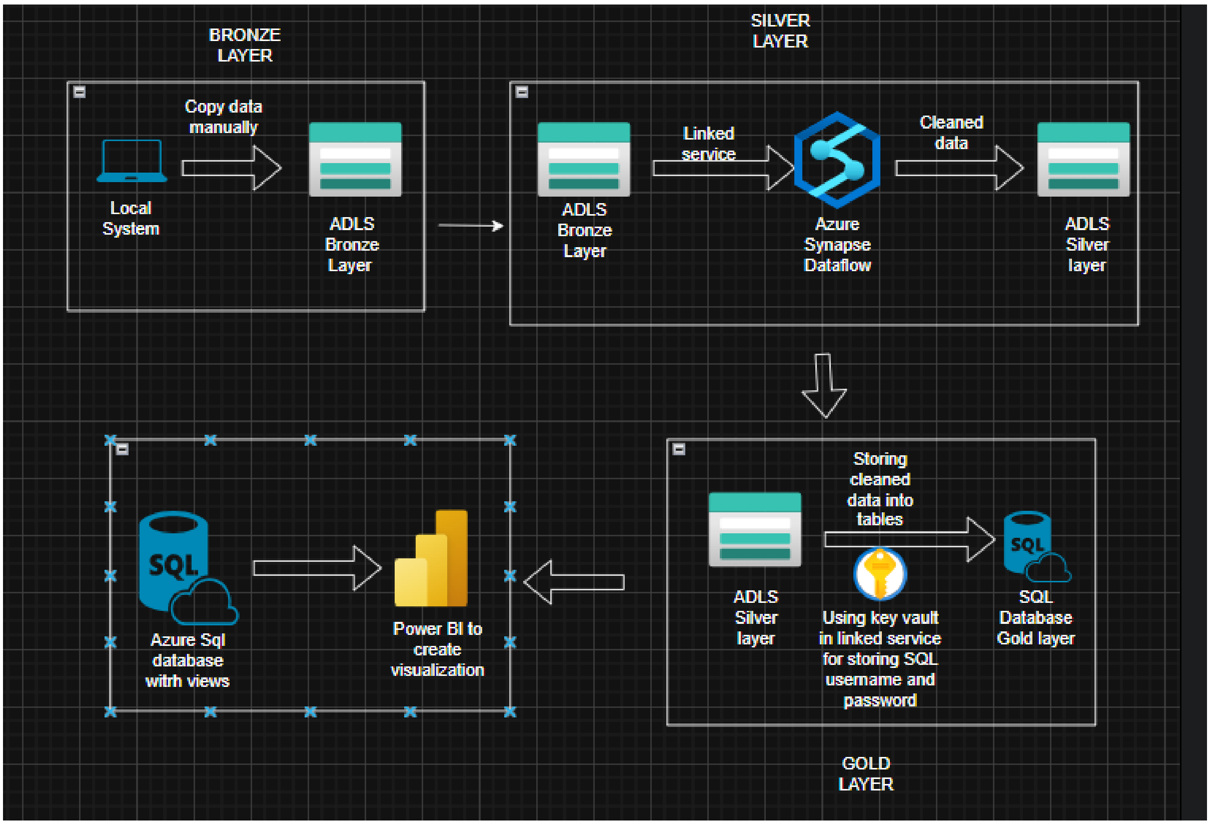
Customer 360 Data Integration

Objective:

A retail business wants to build a unified Customer 360 view by integrating data from multiple sources, including online transactions, in-store purchases, customer service interactions, and loyalty programs. This project uses a mix of fact and dimension tables to ensure a clean, scalable structure.

Architecture:



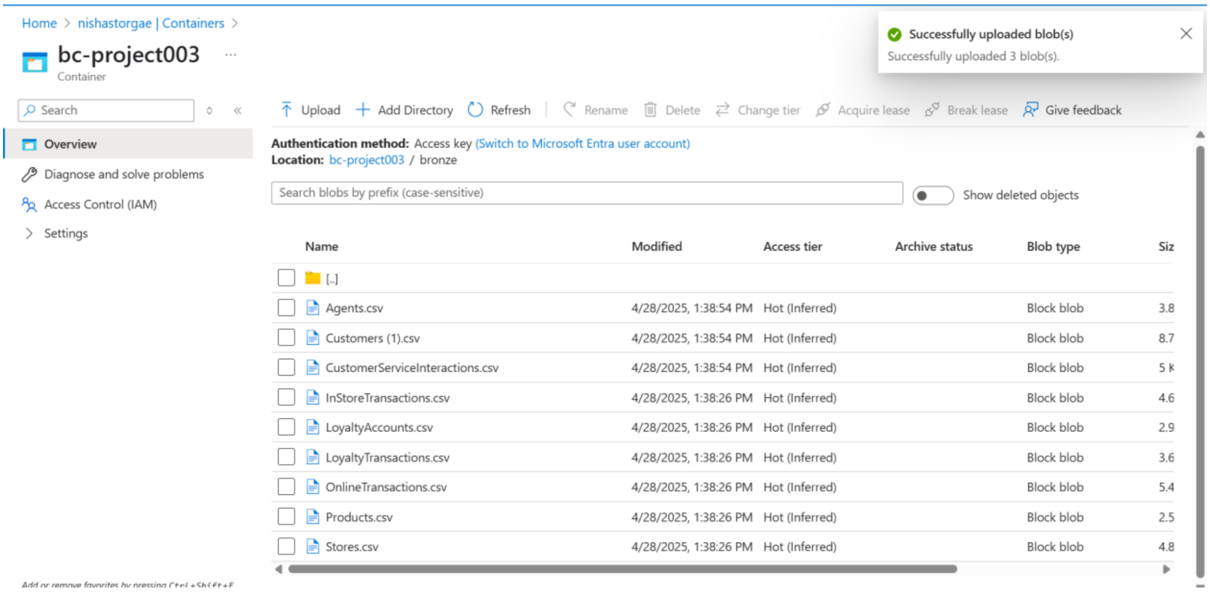
Tools and Technologies:

- Azure Synapse Analytics
- Azure Data Lake Storage (ADLS)
- Azure SQL Database
- Power BI
- Key Vault
- Draw.io

Step-by-Step Process:

- **Bronze layer**

Copying data from local system to Azure data lake storage account manually.



Home > nishastorgae | Containers >
bc-project003 ...
Container
Successfully uploaded blob(s)
Successfully uploaded 3 blob(s).
Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Upload
Add Directory
Refresh
Rename
Delete
Change tier
Acquire lease
Break lease
Give feedback
Search blobs by prefix (case-sensitive)
Show deleted objects
Add or remove favorites by pressing Ctrl+Shift+F

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: bc-project003 / bronze

	Name	Modified	Access tier	Archive status	Blob type	Siz
□	[..]					
□	Agents.csv	4/28/2025, 1:38:54 PM	Hot (Inferred)		Block blob	3.8
□	Customers (1).csv	4/28/2025, 1:38:54 PM	Hot (Inferred)		Block blob	8.7
□	CustomerServiceInteractions.csv	4/28/2025, 1:38:54 PM	Hot (Inferred)		Block blob	5 K
□	InStoreTransactions.csv	4/28/2025, 1:38:26 PM	Hot (Inferred)		Block blob	4.6
□	LoyaltyAccounts.csv	4/28/2025, 1:38:26 PM	Hot (Inferred)		Block blob	2.9
□	LoyaltyTransactions.csv	4/28/2025, 1:38:26 PM	Hot (Inferred)		Block blob	3.6
□	OnlineTransactions.csv	4/28/2025, 1:38:26 PM	Hot (Inferred)		Block blob	5.4
□	Products.csv	4/28/2025, 1:38:26 PM	Hot (Inferred)		Block blob	2.5
□	Stores.csv	4/28/2025, 1:38:26 PM	Hot (Inferred)		Block blob	4.8

- ## Silver Layer

Creating a dataflow in Azure synapse to get data from bronze layer, clean data (remove duplicates, null) and load that data to Azure SQL database.

Before creating Dataflows for files, we need to create tables as our destination is in Azure SQL database.

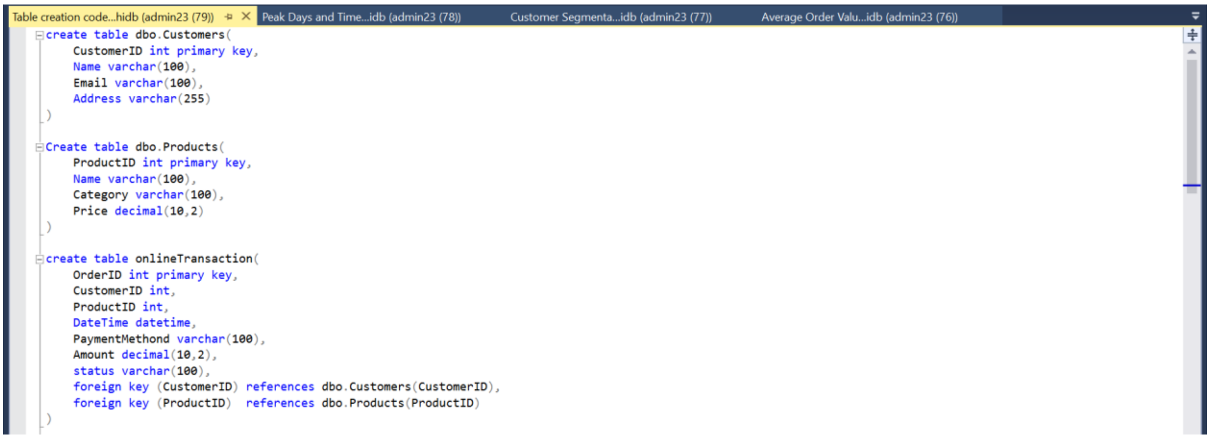
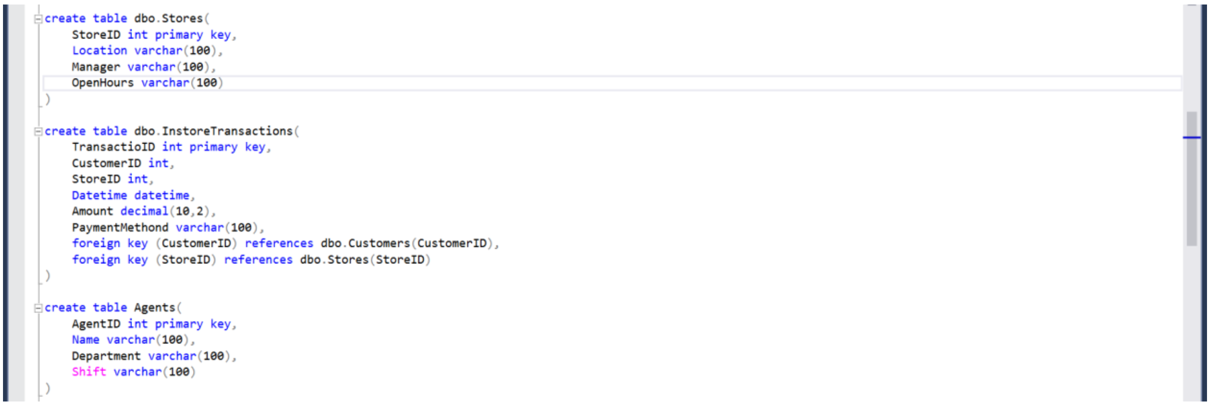


Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
Average Order Valu...idb (admin23 (76))

```
create table dbo.Customers(
    CustomerID int primary key,
    Name varchar(100),
    Email varchar(100),
    Address varchar(255)
)

Create table dbo.Products(
    ProductID int primary key,
    Name varchar(100),
    Category varchar(100),
    Price decimal(10,2)
)

create table onlineTransaction(
    OrderID int primary key,
    CustomerID int,
    ProductID int,
    DateTime datetime,
    PaymentMethond varchar(100),
    Amount decimal(10,2),
    status varchar(100),
    foreign key (CustomerID) references dbo.Customers(CustomerID),
    foreign key (ProductID)  references dbo.Products(ProductID)
)
```



```
create table dbo.Stores(
    StoreID int primary key,
    Location varchar(100),
    Manager varchar(100),
    OpenHours varchar(100)
)

create table dbo.InstoreTransactions(
    TransactioID int primary key,
    CustomerID int,
    StoreID int,
    Datetime datetime,
    Amount decimal(10,2),
    PaymentMethond varchar(100),
    foreign key (CustomerID) references dbo.Customers(CustomerID),
    foreign key (StoreID) references dbo.Stores(StoreID)
)

create table Agents(
    AgentID int primary key,
    Name varchar(100),
    Department varchar(100),
    Shift varchar(100)
)
```

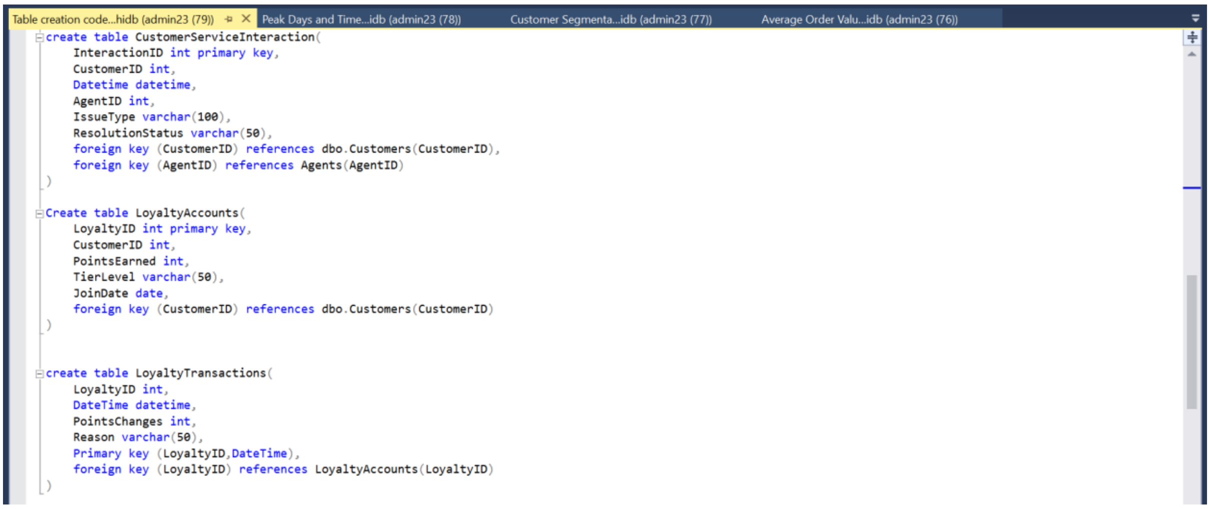
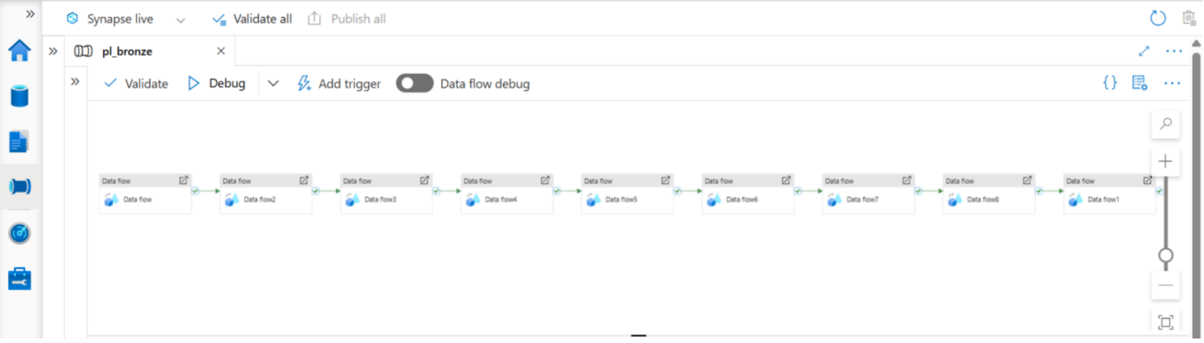


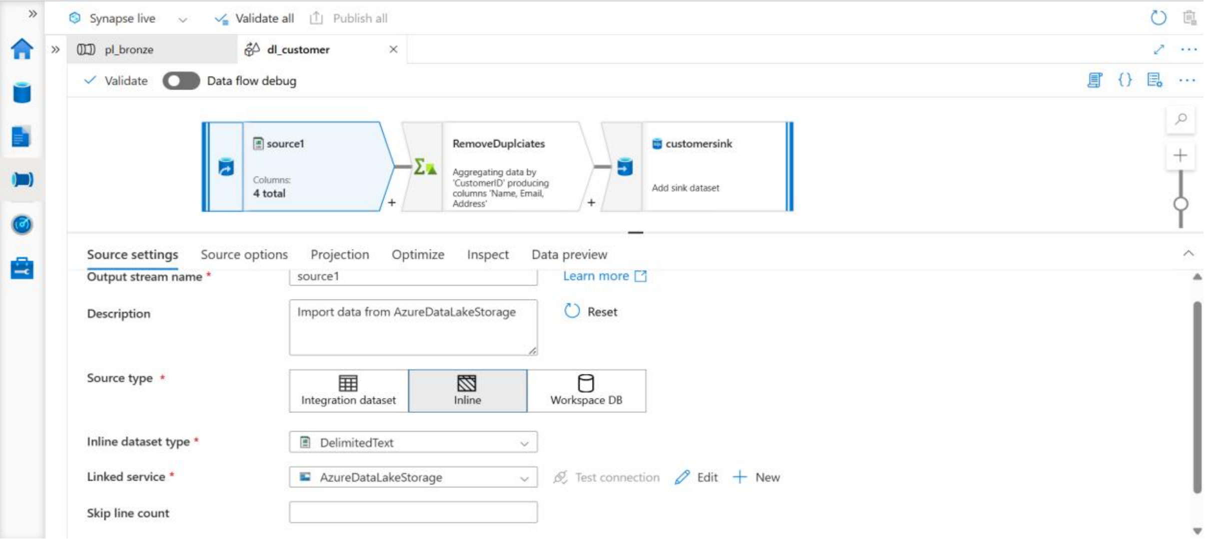
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
Average Order Valu...idb (admin23 (76))

```
create table CustomerServiceInteraction(
    InteractionID int primary key,
    CustomerID int,
    Datetime datetime,
    AgentID int,
    IssueType varchar(100),
    ResolutionStatus varchar(50),
    foreign key (CustomerID) references dbo.Customers(CustomerID),
    foreign key (AgentID) references Agents(AgentID)
)

Create table LoyaltyAccounts(
    LoyaltyID int primary key,
    CustomerID int,
    PointsEarned int,
    TierLevel varchar(50),
    JoinDate date,
    foreign key (CustomerID) references dbo.Customers(CustomerID)
)


create table LoyaltyTransactions(
    LoyaltyID int,
    DateTime datetime,
    PointsChanges int,
    Reason varchar(50),
    Primary key (LoyaltyID,DateTime),
    foreign key (LoyaltyID) references LoyaltyAccounts(LoyaltyID)
)
```

Create dataflows for each file to be cleaned and stored in Azure SQL database.



Dataflow for Customers File:



Output stream name * source1

Description: Import data from AzureDataLakeStorage

Source type *: Inline

Inline dataset type *: DelimitedText

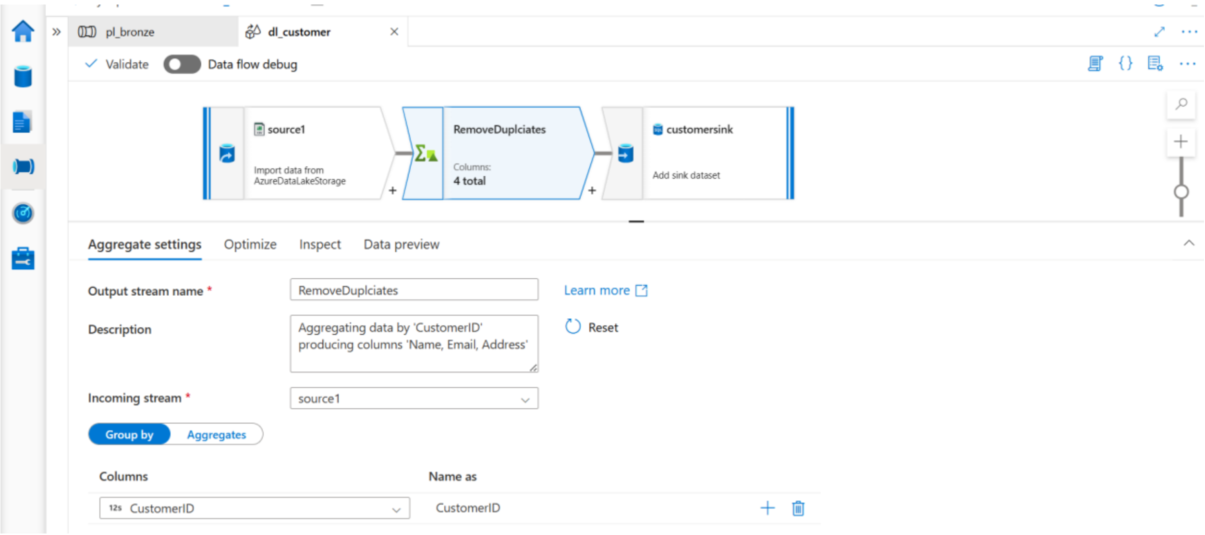
Linked service *: AzureDataLakeStorage

Skip line count

Source:

Our source is from Azure data lake storage gen 2, CSV file so we opt for delimited text file and in source option we give the path from where the file need to be copied, then projection import schema.

Next, we add Aggregate.

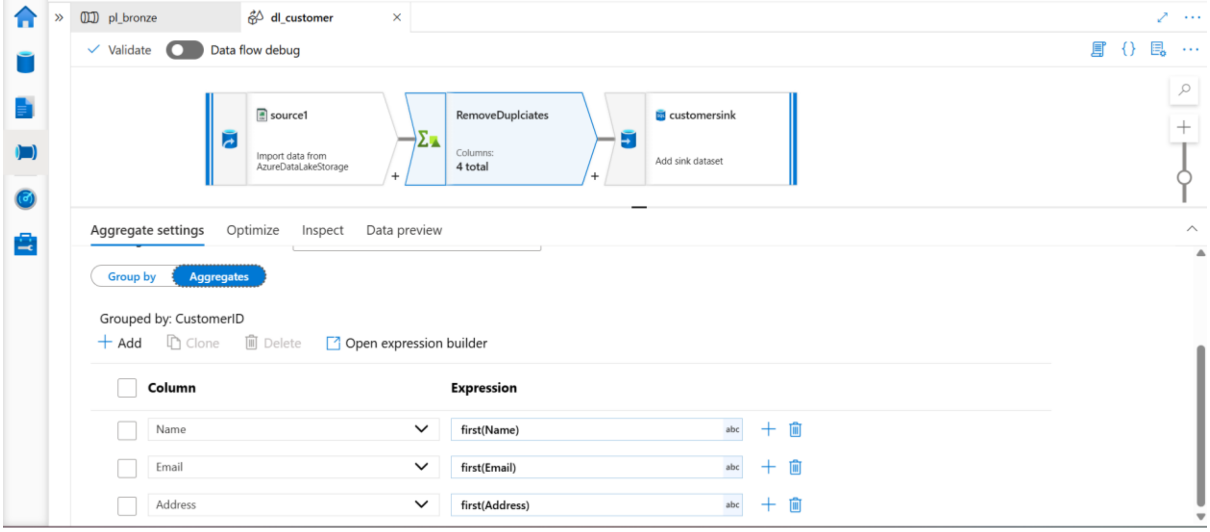


Output stream name * RemoveDuplciates

Description: Aggregating data by 'CustomerID' producing columns 'Name, Email, Address'

Incoming stream * source1

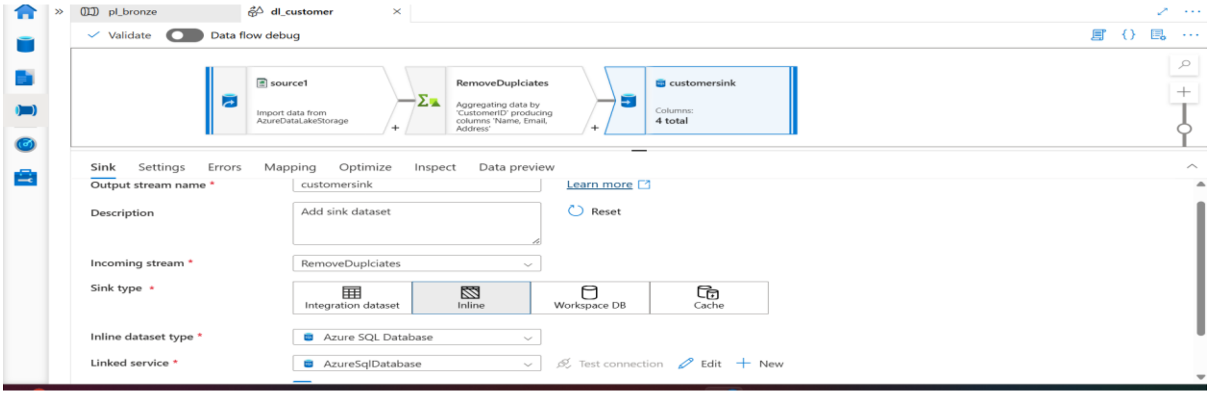
Columns	Name as
CustomerID	CustomerID



Grouped by: CustomerID

Column	Expression
Name	first(Name)
Email	first(Email)
Address	first(Address)

Used Groupby to remove duplicates based on primary key and used first () in aggregates.



Output stream name * customersink

Description: Add sink dataset

Incoming stream * RemoveDuplciates

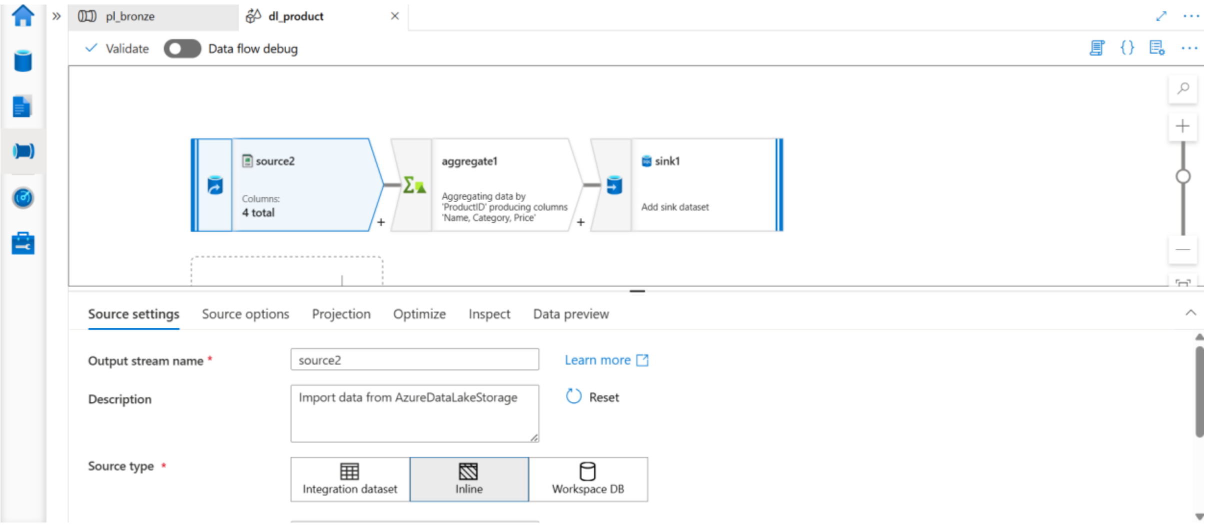
Sink type * Inline

Inline dataset type * Azure SQL Database

Linked service * AzureSqlDatabase

Sink, our destination is Azure SQL database, in settings tab select schema and table in which we want to store the values.

Dataflow for products file:



source2
Columns:
4 total

aggregate1
Aggregating data by 'ProductID' producing columns 'Name, Category, Price'

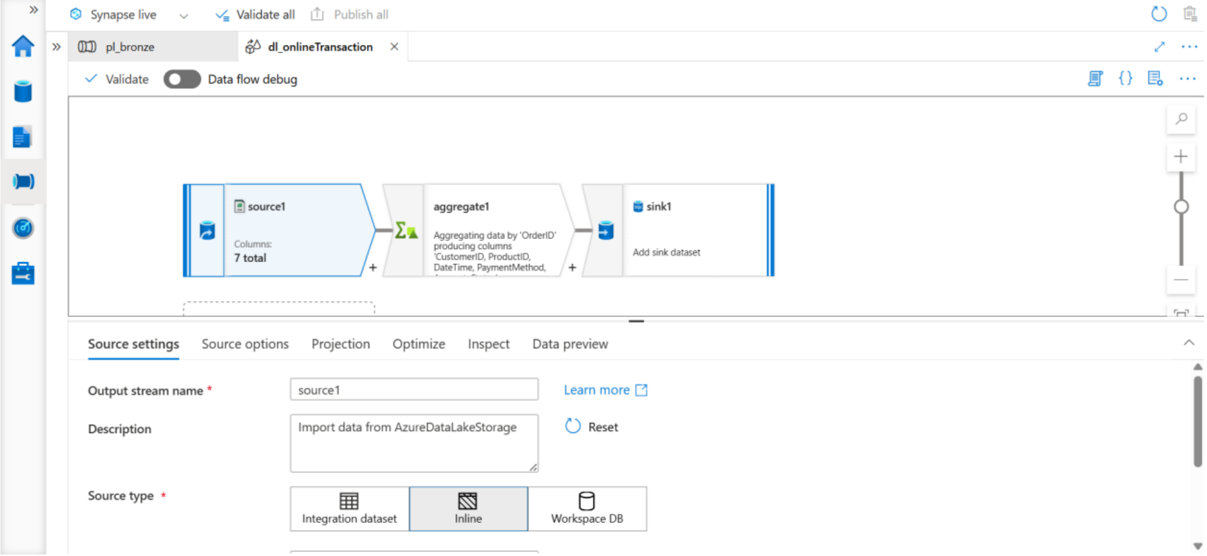
sink1
Add sink dataset

Output stream name * source2

Description: Import data from AzureDataLakeStorage

Source type * □ Integration dataset [✓ Inline] □ Workspace DB

Dataflow for onlineTransaction:



source1
Columns:
7 total

aggregate1
Aggregating data by 'OrderID' producing columns 'CustomerID, ProductID, DateTime, PaymentMethod,

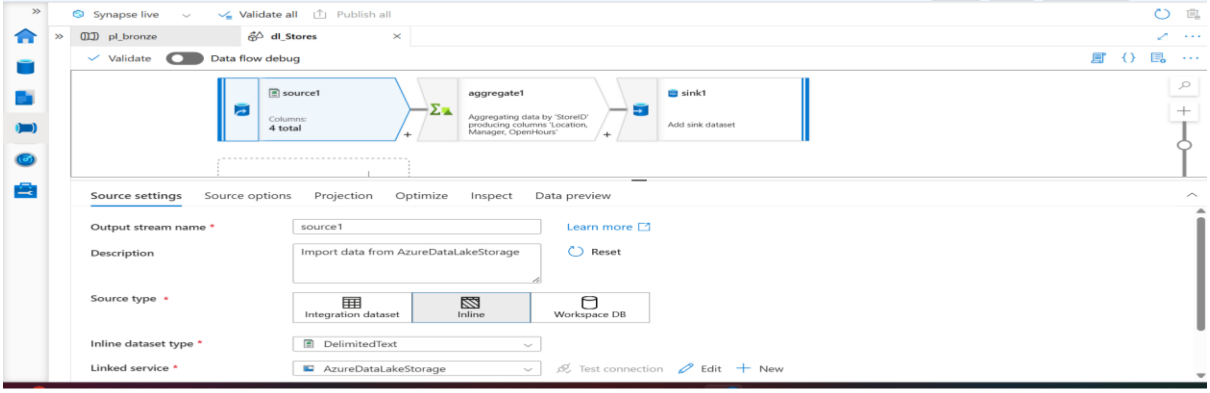
sink1
Add sink dataset

Output stream name * source1

Description: Import data from AzureDataLakeStorage

Source type * □ Integration dataset [✓ Inline] □ Workspace DB

Dataflow for stores:



source1
Columns:
4 total

aggregate1
Aggregating data by 'StoreID' producing columns 'Location, Manager, OpenHours'

sink1
Add sink dataset

Output stream name * source1

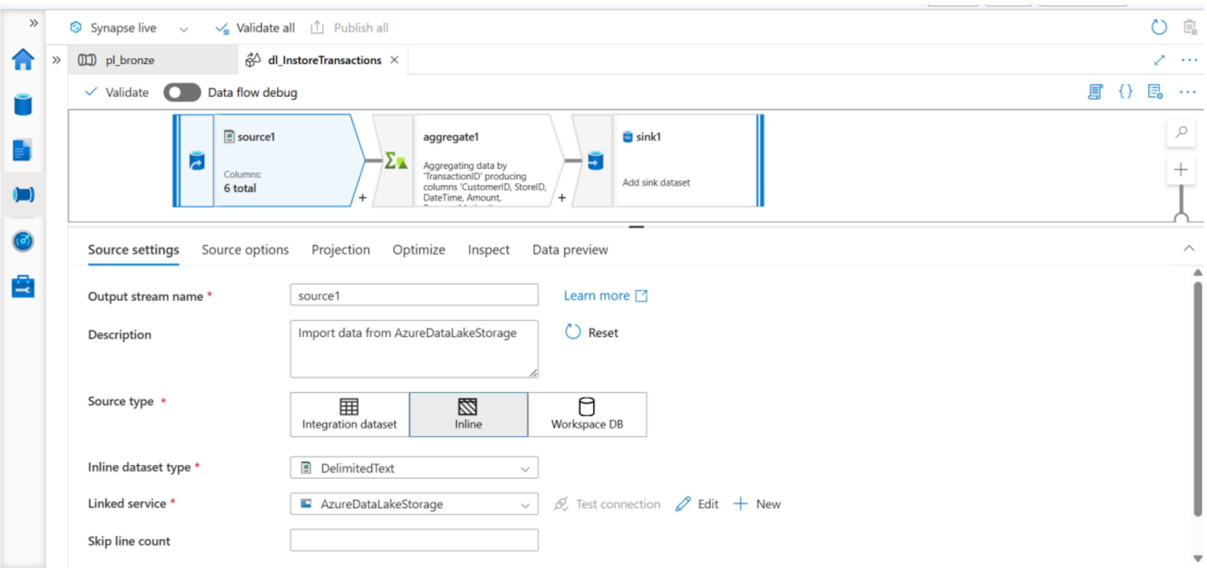
Description: Import data from AzureDataLakeStorage

Source type * □ Integration dataset [✓ Inline] □ Workspace DB

Inline dataset type * DelimitedText

Linked service * AzureDataLakeStorage

Dataflow for InstoreTransactions:



Output stream name * source1

Description: Import data from AzureDataLakeStorage

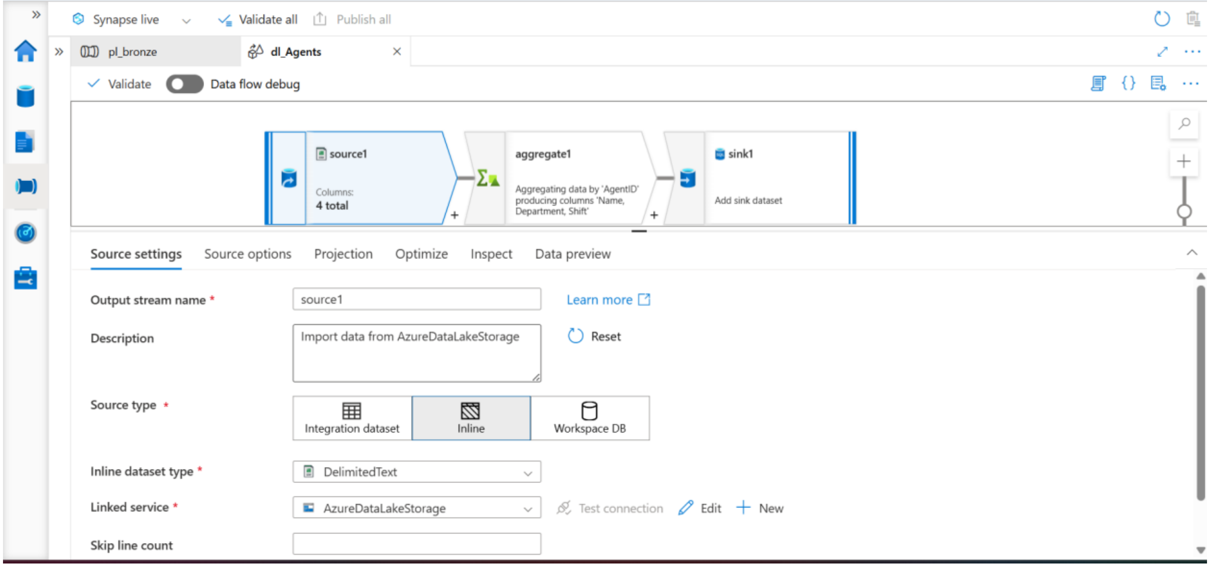
Source type *: Integration dataset | Inline | Workspace DB

Inline dataset type *: DelimitedText

Linked service *: AzureDataLakeStorage

Skip line count

Dataflow for Agents:



Output stream name * source1

Description: Import data from AzureDataLakeStorage

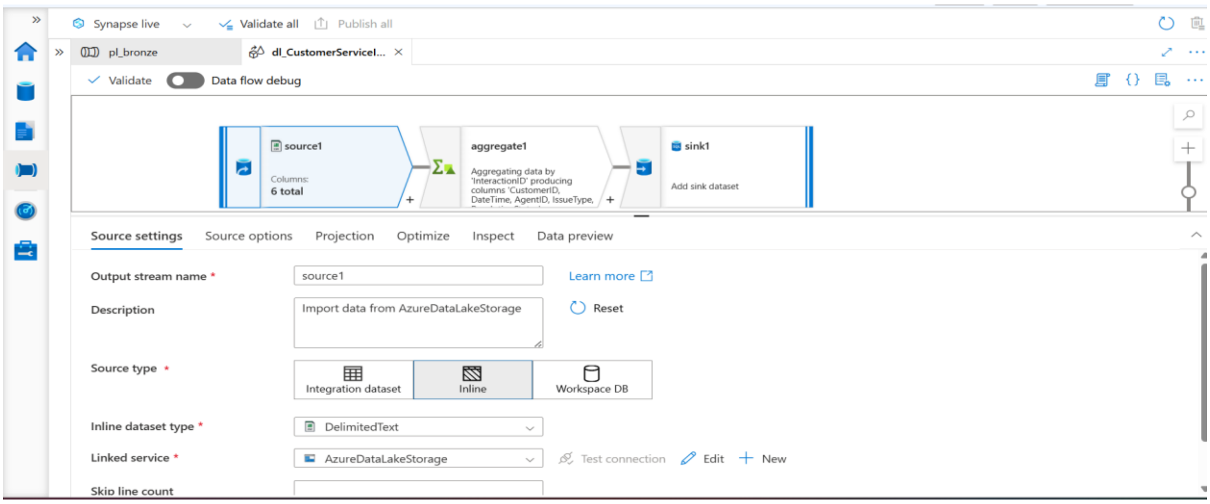
Source type *: Integration dataset | Inline | Workspace DB

Inline dataset type *: DelimitedText

Linked service *: AzureDataLakeStorage

Skip line count

Dataflow for CustomerServiceInteraction:



Output stream name * source1

Description: Import data from AzureDataLakeStorage

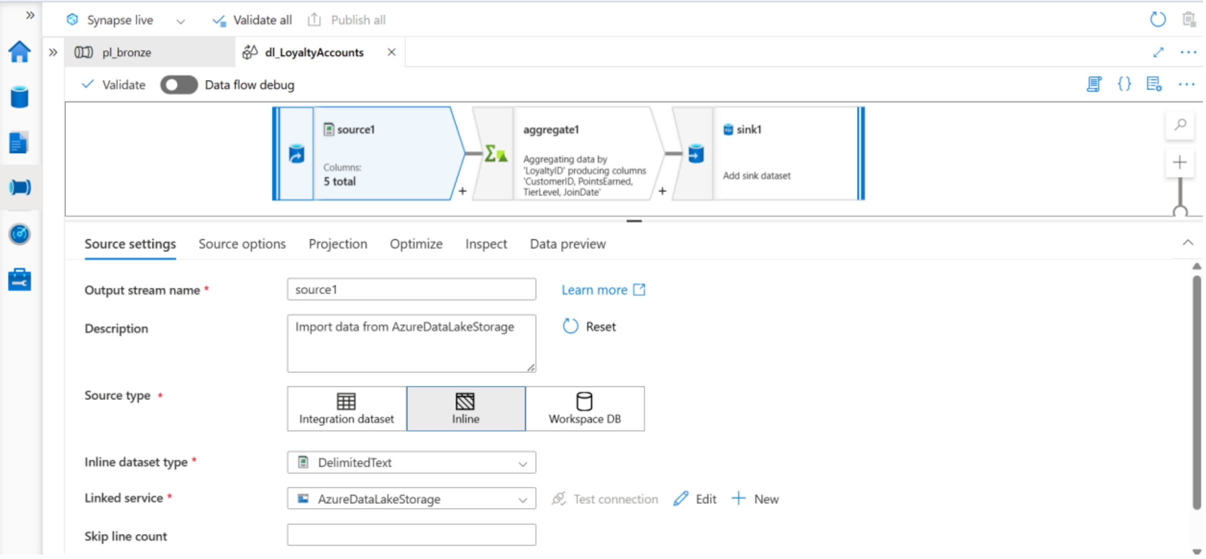
Source type *: Integration dataset | Inline | Workspace DB

Inline dataset type *: DelimitedText

Linked service *: AzureDataLakeStorage

Skip line count

Dataflow for LoyaltyAccounts:



Output stream name *: source1

Description: Import data from AzureDataLakeStorage

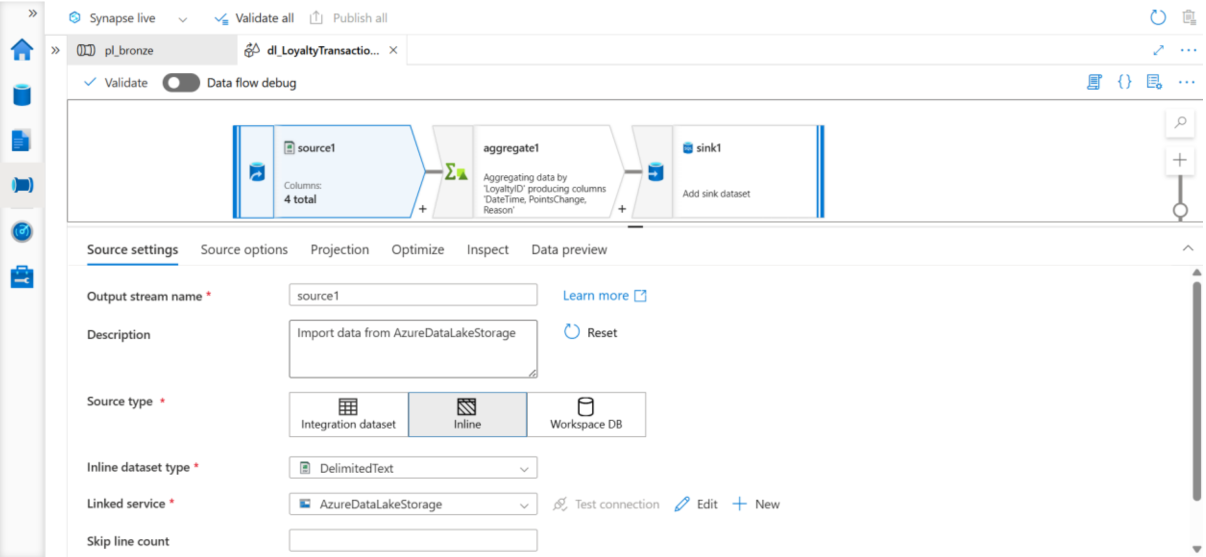
Source type *: Integration dataset / Inline / Workspace DB

Inline dataset type *: DelimitedText

Linked service *: AzureDataLakeStorage

Skip line count:

Dataflows for LoyaltyTransactions:



Output stream name *: source1

Description: Import data from AzureDataLakeStorage

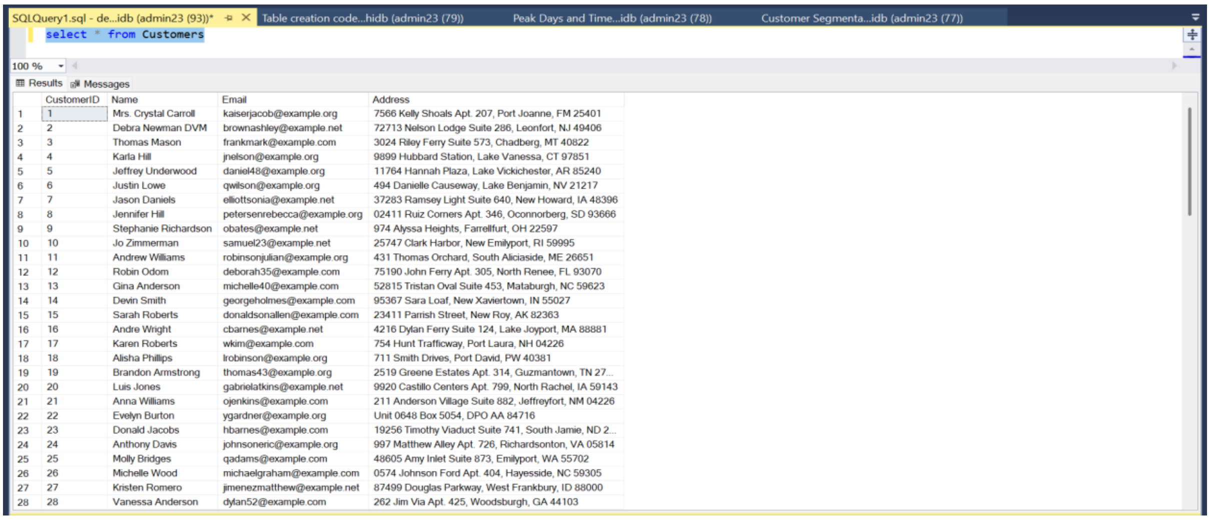
Source type *: Integration dataset / Inline / Workspace DB

Inline dataset type *: DelimitedText

Linked service *: AzureDataLakeStorage

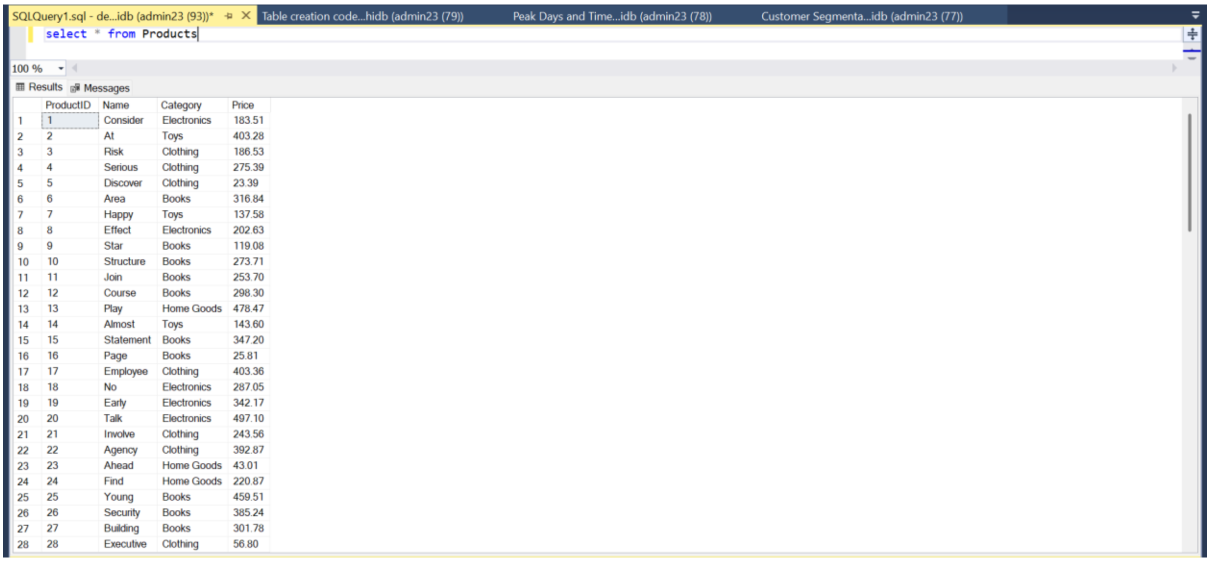
Skip line count:

Data in tables:


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

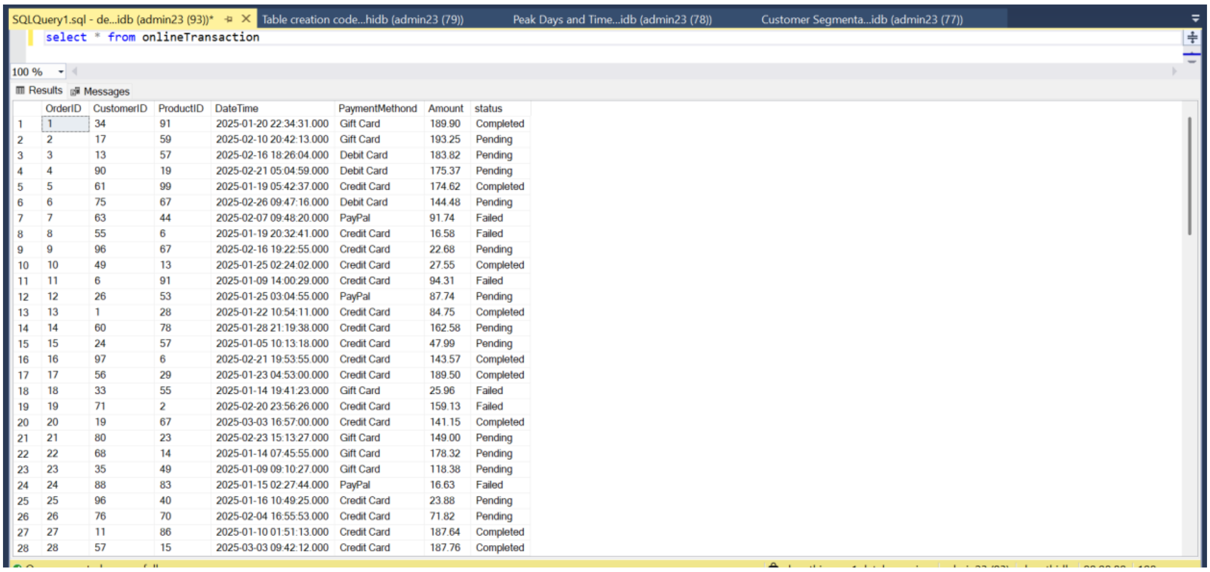
```sql
select * from Customers
```

	CustomerID	Name	Email	Address
1	1	Mrs. Crystal Carroll	kaiserjacob@example.org	7566 Kelly Shoals Apt. 207, Port Joanne, FM 25401
2	2	Debra Newman DVM	brownashley@example.net	72713 Nelson Lodge Suite 286, Leonfort, NJ 49406
3	3	Thomas Mason	frankmark@example.com	3024 Riley Ferry Suite 573, Chadberg, MT 40822
4	4	Karla Hill	jnelson@example.org	9899 Hubbard Station, Lake Vanessa, CT 97851
5	5	Jeffrey Underwood	daniel48@example.org	11764 Hannah Plaza, Lake Vickichester, AR 85240
6	6	Justin Lowe	qwilson@example.org	494 Danielle Causeway, Lake Benjamin, NV 21217
7	7	Jason Daniels	elliottsonia@example.net	37283 Ramsey Light Suite 640, New Howard, IA 48396
8	8	Jennifer Hill	petersenrebecca@example.org	02411 Ruiz Corners Apt. 346, Oconnorberg, SD 93666
9	9	Stephanie Richardson	obates@example.net	974 Alyssa Heights, Farrellfurt, OH 22597
10	10	Jo Zimmerman	samuel23@example.net	25747 Clark Harbor, New Emilyport, RI 59995
11	11	Andrew Williams	robinsonjulian@example.org	431 Thomas Orchard, South Aliciaside, ME 26651
12	12	Robin Odom	deborah35@example.com	75190 John Ferry Apt. 305, North Renee, FL 93070
13	13	Gina Anderson	michelle40@example.com	52815 Tristan Oval Suite 453, Mataburgh, NC 59623
14	14	Devin Smith	georgeholmes@example.com	95367 Sara Loaf, New Xaviertown, IN 55027
15	15	Sarah Roberts	donaldsonallen@example.com	23411 Parrish Street, New Roy, AK 82363
16	16	Andre Wright	cbarnes@example.net	4216 Dylan Ferry Suite 124, Lake Joyport, MA 88881
17	17	Karen Roberts	wkim@example.com	754 Hunt Trafficway, Port Laura, NH 04226
18	18	Alisha Phillips	lrobinson@example.org	711 Smith Drives, Port David, PW 40381
19	19	Brandon Armstrong	thomas43@example.org	2519 Greene Estates Apt. 314, Guzmantown, TN 27...
20	20	Luis Jones	gabrielatkins@example.net	9920 Castillo Centers Apt. 799, North Rachel, IA 59143
21	21	Anna Williams	ojenkins@example.com	211 Anderson Village Suite 882, Jeffreyfort, NM 04226
22	22	Evelyn Burton	ygardner@example.org	Unit 0648 Box 5054, DPO AA 84716
23	23	Donald Jacobs	hbarnes@example.com	19256 Timothy Viaduct Suite 741, South Jamie, ND 2...
24	24	Anthony Davis	johnsoneric@example.org	997 Matthew Alley Apt. 726, Richardsonton, VA 05814
25	25	Molly Bridges	qadams@example.com	48605 Amy Inlet Suite 873, Emilyport, WA 55702
26	26	Michelle Wood	michaelgraham@example.com	0574 Johnson Ford Apt. 404, Hayesside, NC 59305
27	27	Kristen Romero	jimenezmatthew@example.net	87499 Douglas Parkway, West Frankbury, ID 88000
28	28	Vanessa Anderson	dylan52@example.com	262 Jim Via Apt. 425, Woodsburgh, GA 44103


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

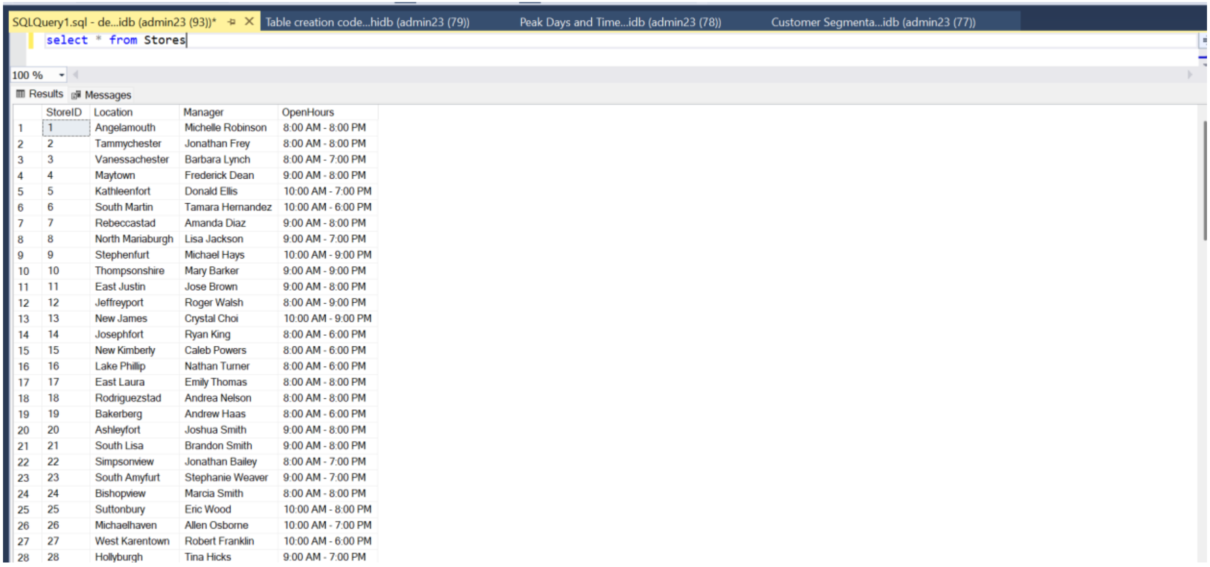
```sql
select * from Products
```

	ProductID	Name	Category	Price
1	1	Consider	Electronics	183.51
2	2	At	Toys	403.28
3	3	Risk	Clothing	186.53
4	4	Serious	Clothing	275.39
5	5	Discover	Clothing	23.39
6	6	Area	Books	316.84
7	7	Happy	Toys	137.58
8	8	Effect	Electronics	202.63
9	9	Star	Books	119.08
10	10	Structure	Books	273.71
11	11	Join	Books	253.70
12	12	Course	Books	298.30
13	13	Play	Home Goods	478.47
14	14	Almost	Toys	143.60
15	15	Statement	Books	347.20
16	16	Page	Books	25.81
17	17	Employee	Clothing	403.36
18	18	No	Electronics	287.05
19	19	Early	Electronics	342.17
20	20	Talk	Electronics	497.10
21	21	Involve	Clothing	243.56
22	22	Agency	Clothing	392.87
23	23	Ahead	Home Goods	43.01
24	24	Find	Home Goods	220.87
25	25	Young	Books	459.51
26	26	Security	Books	385.24
27	27	Building	Books	301.78
28	28	Executive	Clothing	56.80


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

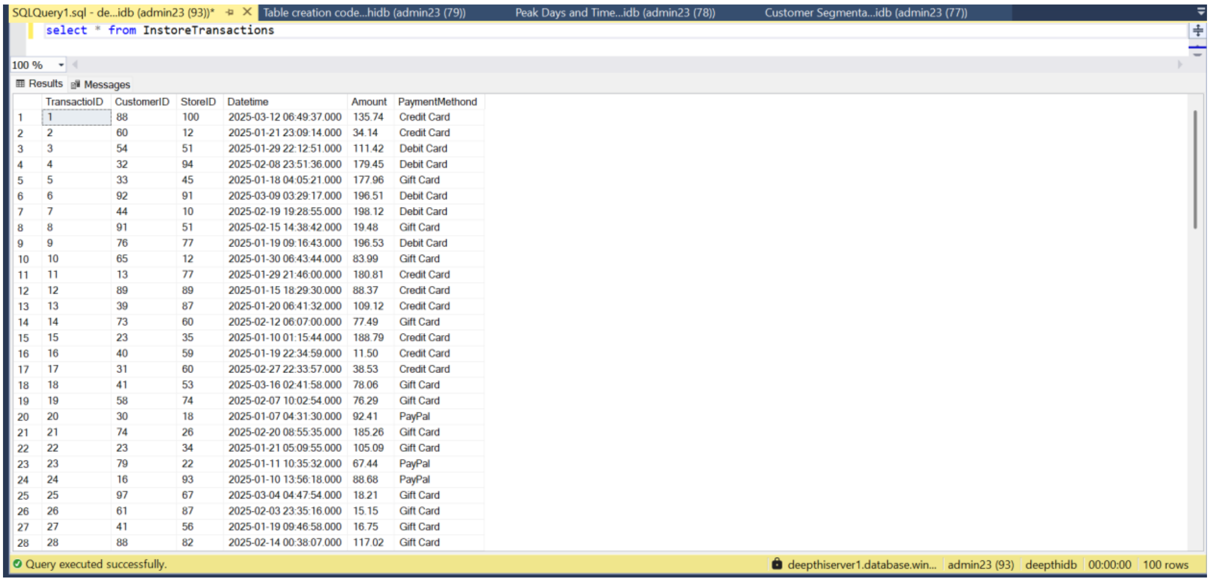
```sql
select * from onlineTransaction
```

	OrderID	CustomerID	ProductID	DateTime	PaymentMethond	Amount	status
1	1	34	91	2025-01-20 22:34:31.000	Gift Card	189.90	Completed
2	2	17	59	2025-02-10 20:42:13.000	Gift Card	193.25	Pending
3	3	13	57	2025-02-16 18:26:04.000	Debit Card	183.82	Pending
4	4	90	19	2025-02-21 05:04:59.000	Debit Card	175.37	Pending
5	5	61	99	2025-01-19 05:42:37.000	Credit Card	174.62	Completed
6	6	75	67	2025-02-26 09:47:16.000	Debit Card	144.48	Pending
7	7	63	44	2025-02-07 09:48:20.000	PayPal	91.74	Failed
8	8	55	6	2025-01-19 20:32:41.000	Credit Card	16.58	Failed
9	9	96	67	2025-02-16 19:22:55.000	Credit Card	22.68	Pending
10	10	49	13	2025-01-25 02:24:02.000	Credit Card	27.55	Completed
11	11	6	91	2025-01-09 14:00:29.000	Credit Card	94.31	Failed
12	12	26	53	2025-01-25 03:04:55.000	PayPal	87.74	Pending
13	13	1	28	2025-01-22 10:54:11.000	Credit Card	84.75	Completed
14	14	60	78	2025-01-28 21:19:38.000	Credit Card	162.58	Pending
15	15	24	57	2025-01-05 10:13:18.000	Credit Card	47.99	Pending
16	16	97	6	2025-02-21 19:53:55.000	Credit Card	143.57	Completed
17	17	56	29	2025-01-23 04:53:00.000	Credit Card	189.50	Completed
18	18	33	55	2025-01-14 19:41:23.000	Gift Card	25.96	Failed
19	19	71	2	2025-02-20 23:56:26.000	Credit Card	159.13	Failed
20	20	19	67	2025-03-03 16:57:00.000	Credit Card	141.15	Completed
21	21	80	23	2025-02-23 15:13:27.000	Gift Card	149.00	Pending
22	22	68	14	2025-01-14 07:45:55.000	Gift Card	178.32	Pending
23	23	35	49	2025-01-09 09:10:27.000	Gift Card	118.38	Pending
24	24	88	83	2025-01-15 02:27:44.000	PayPal	16.63	Failed
25	25	96	40	2025-01-16 10:49:25.000	Credit Card	23.88	Pending
26	26	76	70	2025-02-04 16:55:53.000	Credit Card	71.82	Pending
27	27	11	86	2025-01-10 01:51:13.000	Credit Card	187.64	Completed
28	28	57	15	2025-03-03 09:42:12.000	Credit Card	187.76	Completed


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

```sql
select * from Stores
```

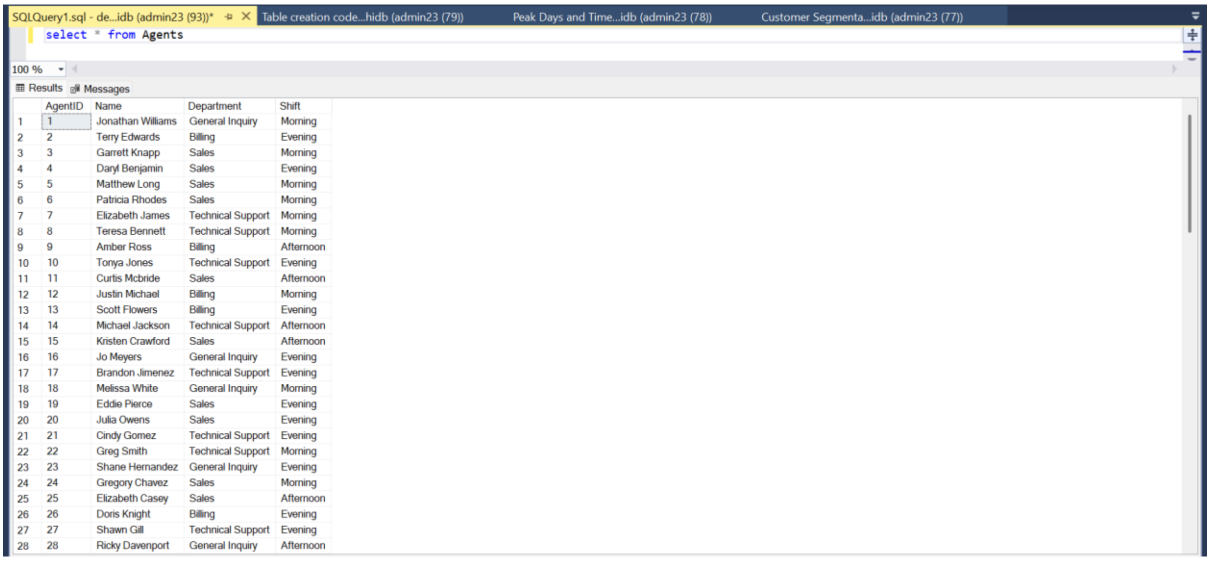
	StoreID	Location	Manager	OpenHours
1	1	Angelamouth	Michelle Robinson	8:00 AM - 8:00 PM
2	2	Tammychester	Jonathan Frey	8:00 AM - 8:00 PM
3	3	Vanessachester	Barbara Lynch	8:00 AM - 7:00 PM
4	4	Maytown	Frederick Dean	9:00 AM - 8:00 PM
5	5	Kathleenfort	Donald Ellis	10:00 AM - 7:00 PM
6	6	South Martin	Tamara Hernandez	10:00 AM - 6:00 PM
7	7	Rebeccastad	Amanda Diaz	9:00 AM - 8:00 PM
8	8	North Mariaburgh	Lisa Jackson	9:00 AM - 7:00 PM
9	9	Stephenfurt	Michael Hays	10:00 AM - 9:00 PM
10	10	Thompsonshire	Mary Barker	9:00 AM - 9:00 PM
11	11	East Justin	Jose Brown	9:00 AM - 8:00 PM
12	12	Jeffreyport	Roger Walsh	8:00 AM - 9:00 PM
13	13	New James	Crystal Choi	10:00 AM - 9:00 PM
14	14	Josephfort	Ryan King	8:00 AM - 6:00 PM
15	15	New Kimberly	Caleb Powers	8:00 AM - 6:00 PM
16	16	Lake Phillip	Nathan Turner	8:00 AM - 6:00 PM
17	17	East Laura	Emily Thomas	8:00 AM - 8:00 PM
18	18	Rodriguezstad	Andrea Nelson	8:00 AM - 8:00 PM
19	19	Bakerberg	Andrew Haas	8:00 AM - 6:00 PM
20	20	Ashleyfort	Joshua Smith	9:00 AM - 8:00 PM
21	21	South Lisa	Brandon Smith	9:00 AM - 8:00 PM
22	22	Simpsonview	Jonathan Bailey	8:00 AM - 7:00 PM
23	23	South Amyfurt	Stephanie Weaver	9:00 AM - 7:00 PM
24	24	Bishopview	Marcia Smith	8:00 AM - 8:00 PM
25	25	Suttonbury	Eric Wood	10:00 AM - 8:00 PM
26	26	Michaelhaven	Allen Osborne	10:00 AM - 7:00 PM
27	27	West Karentown	Robert Franklin	10:00 AM - 6:00 PM
28	28	Hollyburgh	Tina Hicks	9:00 AM - 7:00 PM


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages
deepthiserver1.database.win...
admin23 (93)
deepthidb
00:00:00
100 rows

```
select * from InstoreTransactions
```

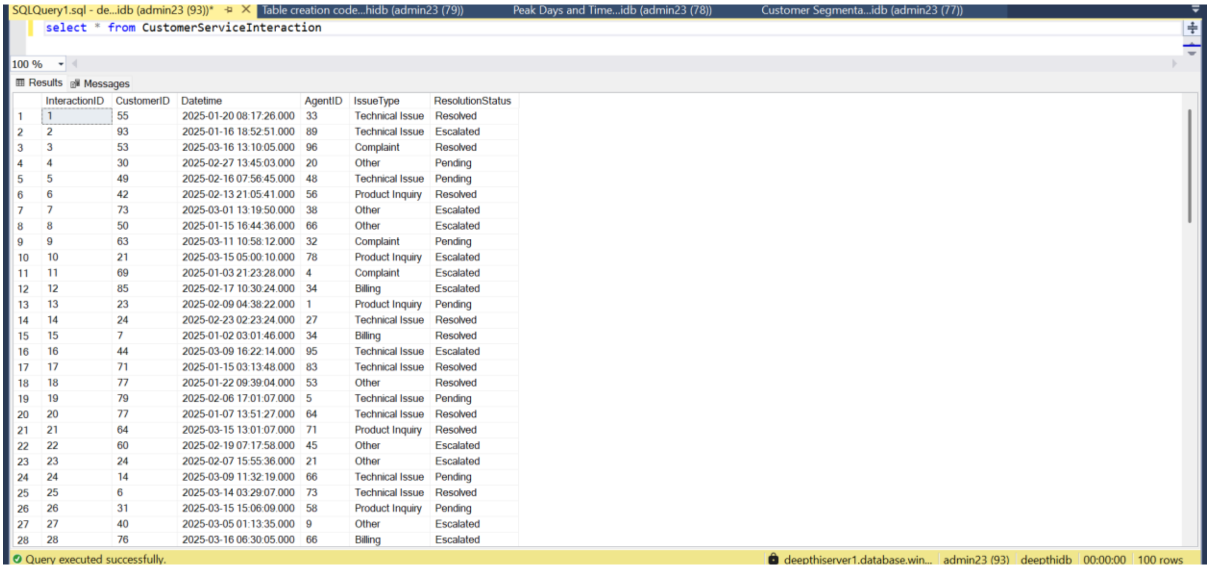
	TransactioID	CustomerID	StoreID	Datetime	Amount	PaymentMethond
1	1	88	100	2025-03-12 06:49:37.000	135.74	Credit Card
2	2	60	12	2025-01-21 23:09:14.000	34.14	Credit Card
3	3	54	51	2025-01-29 22:12:51.000	111.42	Debit Card
4	4	32	94	2025-02-08 23:51:36.000	179.45	Debit Card
5	5	33	45	2025-01-18 04:05:21.000	177.96	Gift Card
6	6	92	91	2025-03-09 03:29:17.000	196.51	Debit Card
7	7	44	10	2025-02-19 19:28:55.000	198.12	Debit Card
8	8	91	51	2025-02-15 14:38:42.000	19.48	Gift Card
9	9	76	77	2025-01-19 09:16:43.000	196.53	Debit Card
10	10	65	12	2025-01-30 06:43:44.000	83.99	Gift Card
11	11	13	77	2025-01-29 21:46:00.000	180.81	Credit Card
12	12	89	89	2025-01-15 18:29:30.000	88.37	Credit Card
13	13	39	87	2025-01-20 06:41:32.000	109.12	Credit Card
14	14	73	60	2025-02-12 06:07:00.000	77.49	Gift Card
15	15	23	35	2025-01-10 01:15:44.000	188.79	Credit Card
16	16	40	59	2025-01-19 22:34:59.000	11.50	Credit Card
17	17	31	60	2025-02-27 22:33:57.000	38.53	Credit Card
18	18	41	53	2025-03-16 02:41:58.000	78.06	Gift Card
19	19	58	74	2025-02-07 10:02:54.000	76.29	Gift Card
20	20	30	18	2025-01-07 04:31:30.000	92.41	PayPal
21	21	74	26	2025-02-20 08:55:35.000	185.26	Gift Card
22	22	23	34	2025-01-21 05:09:55.000	105.09	Gift Card
23	23	79	22	2025-01-11 10:35:32.000	67.44	PayPal
24	24	16	93	2025-01-10 13:56:18.000	88.68	PayPal
25	25	97	67	2025-03-04 04:47:54.000	18.21	Gift Card
26	26	61	87	2025-02-03 23:35:16.000	15.15	Gift Card
27	27	41	56	2025-01-19 09:46:58.000	16.75	Gift Card
28	28	88	82	2025-02-14 00:38:07.000	117.02	Gift Card

Query executed successfully.


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

```
select * from Agents
```

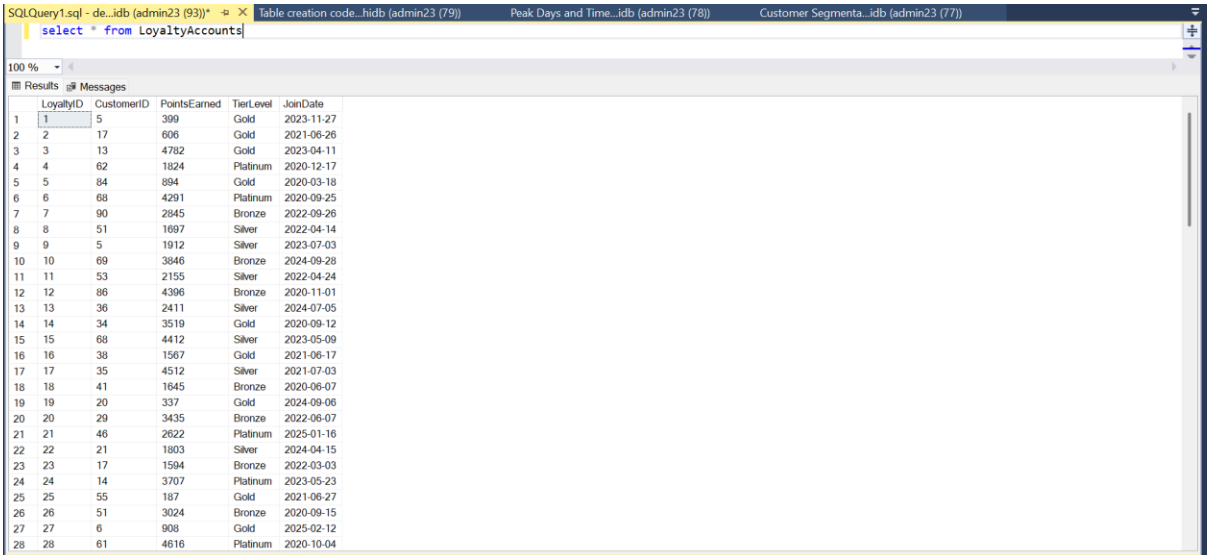
	AgentID	Name	Department	Shift
1	1	Jonathan Williams	General Inquiry	Morning
2	2	Terry Edwards	Billing	Evening
3	3	Garrett Knapp	Sales	Morning
4	4	Daryl Benjamin	Sales	Evening
5	5	Matthew Long	Sales	Morning
6	6	Patricia Rhodes	Sales	Morning
7	7	Elizabeth James	Technical Support	Morning
8	8	Teresa Bennett	Technical Support	Morning
9	9	Amber Ross	Billing	Afternoon
10	10	Tonya Jones	Technical Support	Evening
11	11	Curtis Mcbride	Sales	Afternoon
12	12	Justin Michael	Billing	Morning
13	13	Scott Flowers	Billing	Evening
14	14	Michael Jackson	Technical Support	Afternoon
15	15	Kristen Crawford	Sales	Afternoon
16	16	Jo Meyers	General Inquiry	Evening
17	17	Brandon Jimenez	Technical Support	Evening
18	18	Melissa White	General Inquiry	Morning
19	19	Eddie Pierce	Sales	Evening
20	20	Julia Owens	Sales	Evening
21	21	Cindy Gomez	Technical Support	Evening
22	22	Greg Smith	Technical Support	Morning
23	23	Shane Hernandez	General Inquiry	Evening
24	24	Gregory Chavez	Sales	Morning
25	25	Elizabeth Casey	Sales	Afternoon
26	26	Doris Knight	Billing	Evening
27	27	Shawn Gill	Technical Support	Evening
28	28	Ricky Davenport	General Inquiry	Afternoon


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages
deepthiserver1.database.win...
admin23 (93)
deepthidb
00:00:00
100 rows

```
select * from CustomerServiceInteraction
```

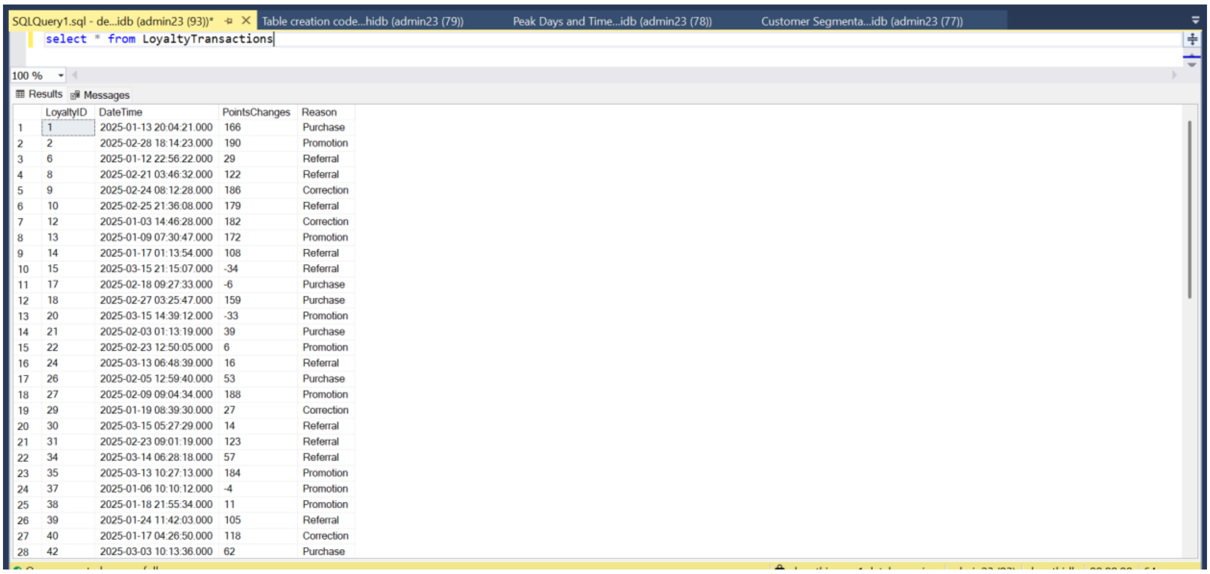
	InteractionID	CustomerID	Datetime	AgentID	IssueType	ResolutionStatus
1	1	55	2025-01-20 08:17:26.000	33	Technical Issue	Resolved
2	2	93	2025-01-16 18:52:51.000	89	Technical Issue	Escalated
3	3	53	2025-03-16 13:10:05.000	96	Complaint	Resolved
4	4	30	2025-02-27 13:45:03.000	20	Other	Pending
5	5	49	2025-02-16 07:56:45.000	48	Technical Issue	Pending
6	6	42	2025-02-13 21:05:41.000	56	Product Inquiry	Resolved
7	7	73	2025-03-01 13:19:50.000	38	Other	Escalated
8	8	50	2025-01-15 16:44:36.000	66	Other	Escalated
9	9	63	2025-03-11 10:58:12.000	32	Complaint	Pending
10	10	21	2025-03-15 05:00:10.000	78	Product Inquiry	Escalated
11	11	69	2025-01-03 21:23:28.000	4	Complaint	Escalated
12	12	85	2025-02-17 10:30:24.000	34	Billing	Escalated
13	13	23	2025-02-09 04:38:22.000	1	Product Inquiry	Pending
14	14	24	2025-02-23 02:23:24.000	27	Technical Issue	Resolved
15	15	7	2025-01-02 03:01:46.000	34	Billing	Resolved
16	16	44	2025-03-09 16:22:14.000	95	Technical Issue	Escalated
17	17	71	2025-01-15 03:13:48.000	83	Technical Issue	Resolved
18	18	77	2025-01-22 09:39:04.000	53	Other	Resolved
19	19	79	2025-02-06 17:01:07.000	5	Technical Issue	Pending
20	20	77	2025-01-07 13:51:27.000	64	Technical Issue	Resolved
21	21	64	2025-03-15 13:01:07.000	71	Product Inquiry	Resolved
22	22	60	2025-02-19 07:17:58.000	45	Other	Escalated
23	23	24	2025-02-07 15:55:36.000	21	Other	Escalated
24	24	14	2025-03-09 11:32:19.000	66	Technical Issue	Pending
25	25	6	2025-03-14 03:29:07.000	73	Technical Issue	Resolved
26	26	31	2025-03-15 15:06:09.000	58	Product Inquiry	Pending
27	27	40	2025-03-05 01:13:35.000	9	Other	Escalated
28	28	76	2025-03-16 06:30:05.000	66	Billing	Escalated

Query executed successfully.


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

```sql
select * from LoyaltyAccounts
```

	LoyaltyID	CustomerID	PointsEarned	TierLevel	JoinDate
1	1	5	399	Gold	2023-11-27
2	2	17	606	Gold	2021-06-26
3	3	13	4782	Gold	2023-04-11
4	4	62	1824	Platinum	2020-12-17
5	5	84	894	Gold	2020-03-18
6	6	68	4291	Platinum	2020-09-25
7	7	90	2845	Bronze	2022-09-26
8	8	51	1697	Silver	2022-04-14
9	9	5	1912	Silver	2023-07-03
10	10	69	3846	Bronze	2024-09-28
11	11	53	2155	Silver	2022-04-24
12	12	86	4396	Bronze	2020-11-01
13	13	36	2411	Silver	2024-07-05
14	14	34	3519	Gold	2020-09-12
15	15	68	4412	Silver	2023-05-09
16	16	38	1567	Gold	2021-06-17
17	17	35	4512	Silver	2021-07-03
18	18	41	1645	Bronze	2020-06-07
19	19	20	337	Gold	2024-09-06
20	20	29	3435	Bronze	2022-06-07
21	21	46	2622	Platinum	2025-01-16
22	22	21	1803	Silver	2024-04-15
23	23	17	1594	Bronze	2022-03-03
24	24	14	3707	Platinum	2023-05-23
25	25	55	187	Gold	2021-06-27
26	26	51	3024	Bronze	2020-09-15
27	27	6	908	Gold	2025-02-12
28	28	61	4616	Platinum	2020-10-04


SQLQuery1.sql - de...idb (admin23 (93))*
Table creation code...hidb (admin23 (79))
Peak Days and Time...idb (admin23 (78))
Customer Segmenta...idb (admin23 (77))
100 %
Results
Messages

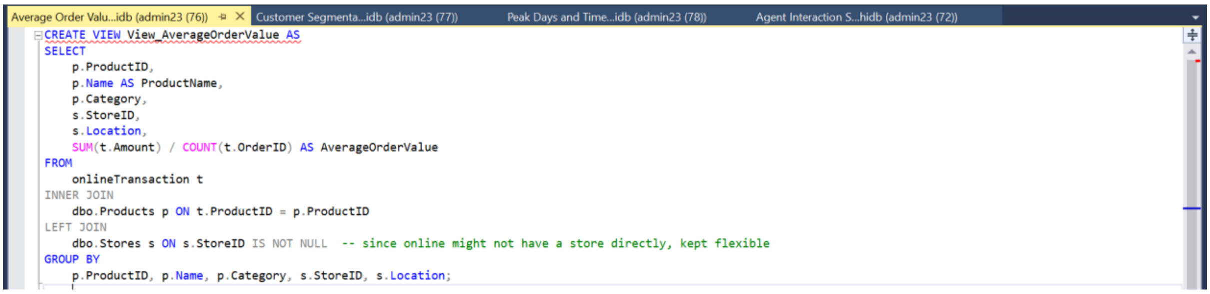
```sql
select * from LoyaltyTransactions
```

	LoyaltyID	DateTime	PointsChanges	Reason
1	1	2025-01-13 20:04:21.000	166	Purchase
2	2	2025-02-28 18:14:23.000	190	Promotion
3	6	2025-01-12 22:56:22.000	29	Referral
4	8	2025-02-21 03:46:32.000	122	Referral
5	9	2025-02-24 08:12:28.000	186	Correction
6	10	2025-02-25 21:36:08.000	179	Referral
7	12	2025-01-03 14:46:28.000	182	Correction
8	13	2025-01-09 07:30:47.000	172	Promotion
9	14	2025-01-17 01:13:54.000	108	Referral
10	15	2025-03-15 21:15:07.000	-34	Referral
11	17	2025-02-18 09:27:33.000	-6	Purchase
12	18	2025-02-27 03:25:47.000	159	Purchase
13	20	2025-03-15 14:39:12.000	-33	Promotion
14	21	2025-02-03 01:13:19.000	39	Purchase
15	22	2025-02-23 12:50:05.000	6	Promotion
16	24	2025-03-13 06:48:39.000	16	Referral
17	26	2025-02-05 12:59:40.000	53	Purchase
18	27	2025-02-09 09:04:34.000	188	Promotion
19	29	2025-01-19 08:39:30.000	27	Correction
20	30	2025-03-15 05:27:29.000	14	Referral
21	31	2025-02-23 09:01:19.000	123	Referral
22	34	2025-03-14 06:28:18.000	57	Referral
23	35	2025-03-13 10:27:13.000	184	Promotion
24	37	2025-01-06 10:10:12.000	-4	Promotion
25	38	2025-01-18 21:55:34.000	11	Promotion
26	39	2025-01-24 11:42:03.000	105	Referral
27	40	2025-01-17 04:26:50.000	118	Correction
28	42	2025-03-03 10:13:36.000	62	Purchase

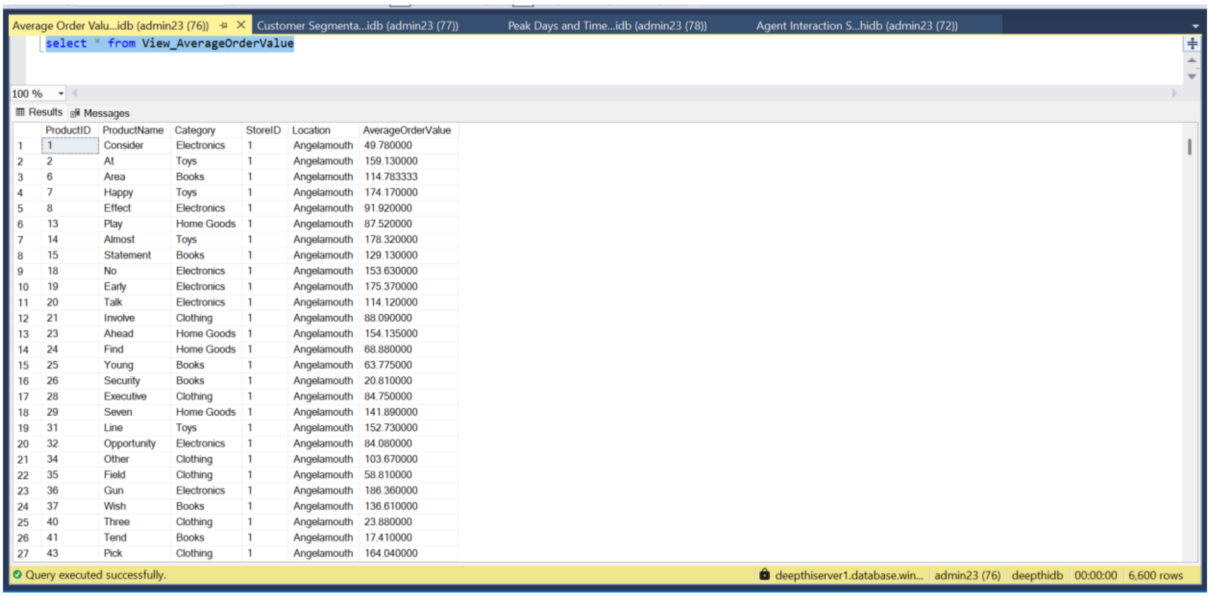
Creating Views:

1. View 1 - for Average Order Value (AOV)

SUM(Amount) / COUNT(OrderID) per product, category, and location.


Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))

```sql
CREATE VIEW View_AverageOrderValue AS
SELECT
    p.ProductID,
    p.Name AS ProductName,
    p.Category,
    s.StoreID,
    s.Location,
    SUM(t.Amount) / COUNT(t.OrderID) AS AverageOrderValue
FROM
    onlineTransaction t
INNER JOIN
    dbo.Products p ON t.ProductID = p.ProductID
LEFT JOIN
    dbo.Stores s ON s.StoreID IS NOT NULL  -- since online might not have a store directly, kept flexible
GROUP BY
    p.ProductID, p.Name, p.Category, s.StoreID, s.Location;
```

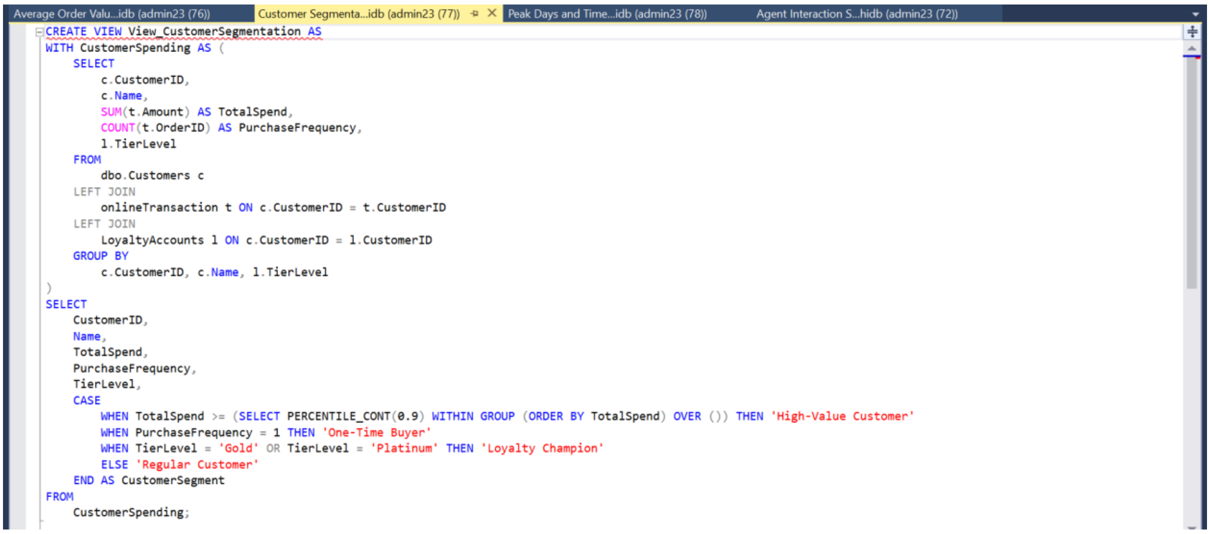

Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))
100 %
Results
Messages

```sql
select * from View_AverageOrderValue
```

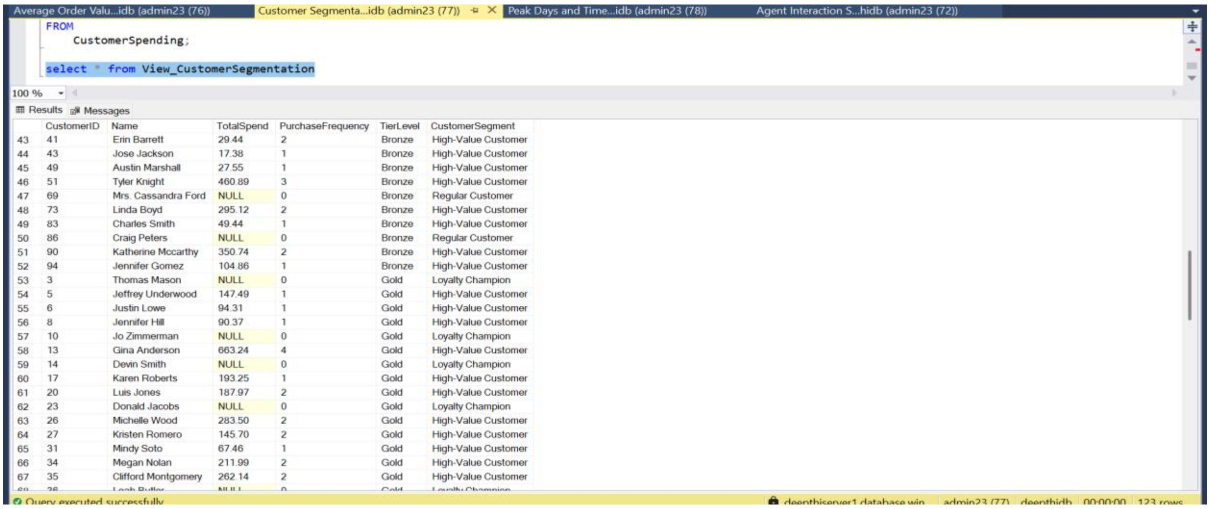
	ProductID	ProductName	Category	StoreID	Location	AverageOrderValue
1	1	Consider	Electronics	1	Angelamouth	49.780000
2	2	At	Toys	1	Angelamouth	159.130000
3	6	Area	Books	1	Angelamouth	114.783333
4	7	Happy	Toys	1	Angelamouth	174.170000
5	8	Effect	Electronics	1	Angelamouth	91.920000
6	13	Play	Home Goods	1	Angelamouth	87.520000
7	14	Almost	Toys	1	Angelamouth	178.320000
8	15	Statement	Books	1	Angelamouth	129.130000
9	18	No	Electronics	1	Angelamouth	153.630000
10	19	Early	Electronics	1	Angelamouth	175.370000
11	20	Talk	Electronics	1	Angelamouth	114.120000
12	21	Involve	Clothing	1	Angelamouth	88.090000
13	23	Ahead	Home Goods	1	Angelamouth	154.135000
14	24	Find	Home Goods	1	Angelamouth	68.880000
15	25	Young	Books	1	Angelamouth	63.775000
16	26	Security	Books	1	Angelamouth	20.810000
17	28	Executive	Clothing	1	Angelamouth	84.750000
18	29	Seven	Home Goods	1	Angelamouth	141.890000
19	31	Line	Toys	1	Angelamouth	152.730000
20	32	Opportunity	Electronics	1	Angelamouth	84.080000
21	34	Other	Clothing	1	Angelamouth	103.670000
22	35	Field	Clothing	1	Angelamouth	58.810000
23	36	Gun	Electronics	1	Angelamouth	186.360000
24	37	Wish	Books	1	Angelamouth	136.610000
25	40	Three	Clothing	1	Angelamouth	23.880000
26	41	Tend	Books	1	Angelamouth	17.410000
27	43	Pick	Clothing	1	Angelamouth	164.040000

```
Query executed successfully.    deepthiserver1.database.win...  admin23 (76)  deepthidb  00:00:00  6,600 rows
```

2. View 2 - for Segment customers based on total spend, purchase frequency, and loyalty tier (LoyaltyAccounts.TierLevel).
   Example: "High-Value Customers" (Top 10% spenders), "One-Time Buyers," "Loyalty Champions."


Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))

```sql
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
        onlineTransaction 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;
```

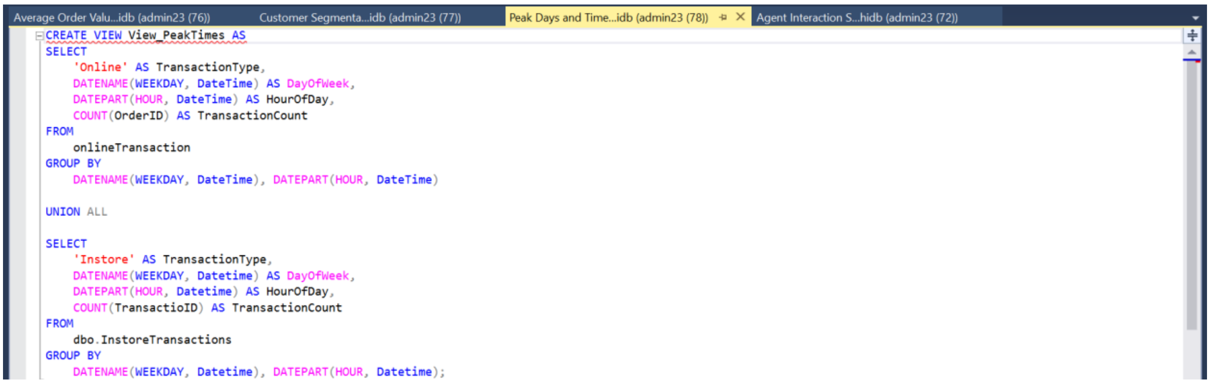

Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))
100 %
Results
Messages

```sql
FROM
    CustomerSpending;

select * from View_CustomerSegmentation
```

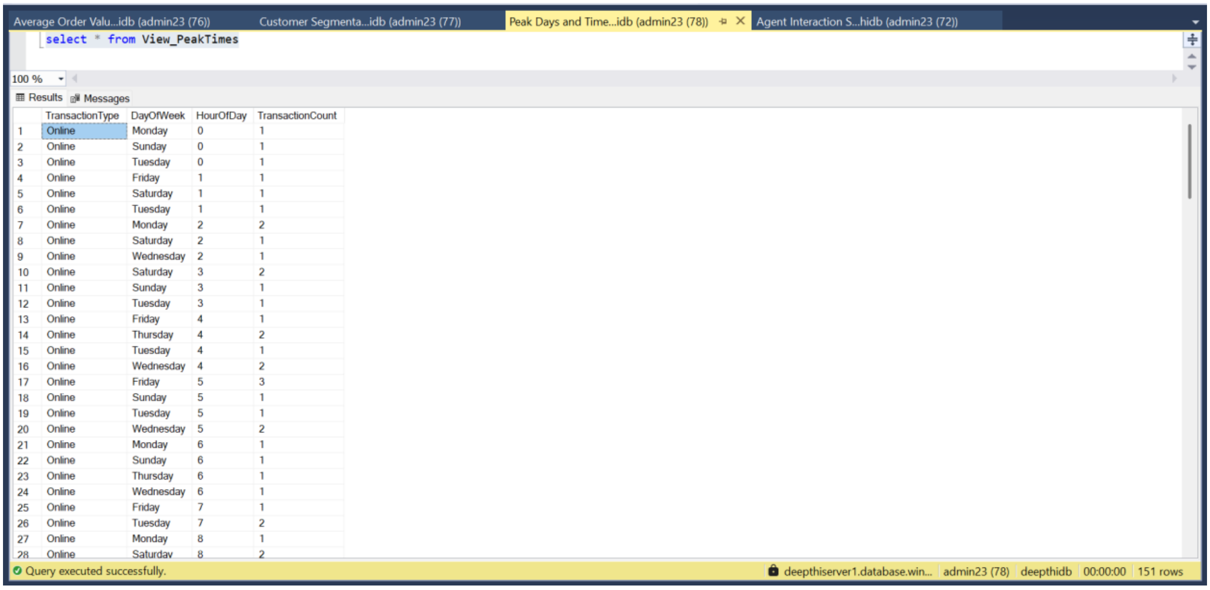
	CustomerID	Name	TotalSpend	PurchaseFrequency	TierLevel	CustomerSegment
43	41	Erin Barrett	29.44	2	Bronze	High-Value Customer
44	43	Jose Jackson	17.38	1	Bronze	High-Value Customer
45	49	Austin Marshall	27.55	1	Bronze	High-Value Customer
46	51	Tyler Knight	460.89	3	Bronze	High-Value Customer
47	69	Mrs. 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
50	86	Craig Peters	NULL	0	Bronze	Regular Customer
51	90	Katherine Mccarthy	350.74	2	Bronze	High-Value Customer
52	94	Jennifer Gomez	104.86	1	Bronze	High-Value Customer
53	3	Thomas Mason	NULL	0	Gold	Loyalty Champion
54	5	Jeffrey Underwood	147.49	1	Gold	High-Value Customer
55	6	Justin Lowe	94.31	1	Gold	High-Value Customer
56	8	Jennifer Hill	90.37	1	Gold	High-Value Customer
57	10	Jo Zimmerman	NULL	0	Gold	Loyalty Champion
58	13	Gina Anderson	663.24	4	Gold	High-Value Customer
59	14	Devin Smith	NULL	0	Gold	Loyalty Champion
60	17	Karen Roberts	193.25	1	Gold	High-Value Customer
61	20	Luis Jones	187.97	2	Gold	High-Value Customer
62	23	Donald Jacobs	NULL	0	Gold	Loyalty Champion
63	26	Michelle Wood	283.50	2	Gold	High-Value Customer
64	27	Kristen Romero	145.70	2	Gold	High-Value Customer
65	31	Mindy Soto	67.46	1	Gold	High-Value Customer
66	34	Megan Nolan	211.99	2	Gold	High-Value Customer
67	35	Clifford Montgomery	262.14	2	Gold	High-Value Customer
68	38	Leah Butler	NULL	0	Gold	Loyalty Champion

```
Query executed successfully.    deepthiserver1.database.win...  admin23 (77)  deepthidb  00:00:00  123 rows
```

3. View 3 - for Analyze DateTime to find peak days and times in-store vs. online.


Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))

```
CREATE VIEW View_PeakTimes AS
SELECT
    'Online' AS TransactionType,
    DATENAME(WEEKDAY, DateTime) AS DayOfWeek,
    DATEPART(HOUR, DateTime) AS HourOfDay,
    COUNT(OrderID) AS TransactionCount
FROM
    onlineTransaction
GROUP BY
    DATENAME(WEEKDAY, DateTime), DATEPART(HOUR, DateTime)

UNION ALL

SELECT
    'Instore' AS TransactionType,
    DATENAME(WEEKDAY, Datetime) AS DayOfWeek,
    DATEPART(HOUR, Datetime) AS HourOfDay,
    COUNT(TransactioID) AS TransactionCount
FROM
    dbo.InstoreTransactions
GROUP BY
    DATENAME(WEEKDAY, Datetime), DATEPART(HOUR, Datetime);
```

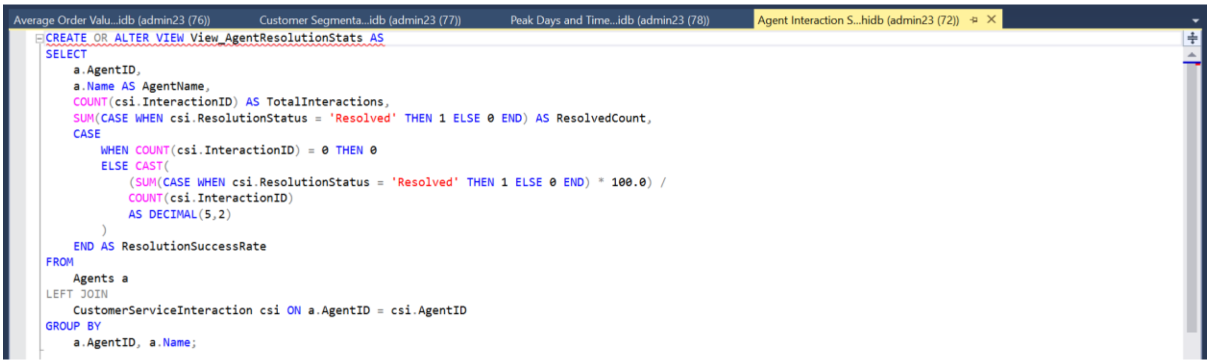

Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))
100 %
Results
Messages

```
select * from View_PeakTimes
```

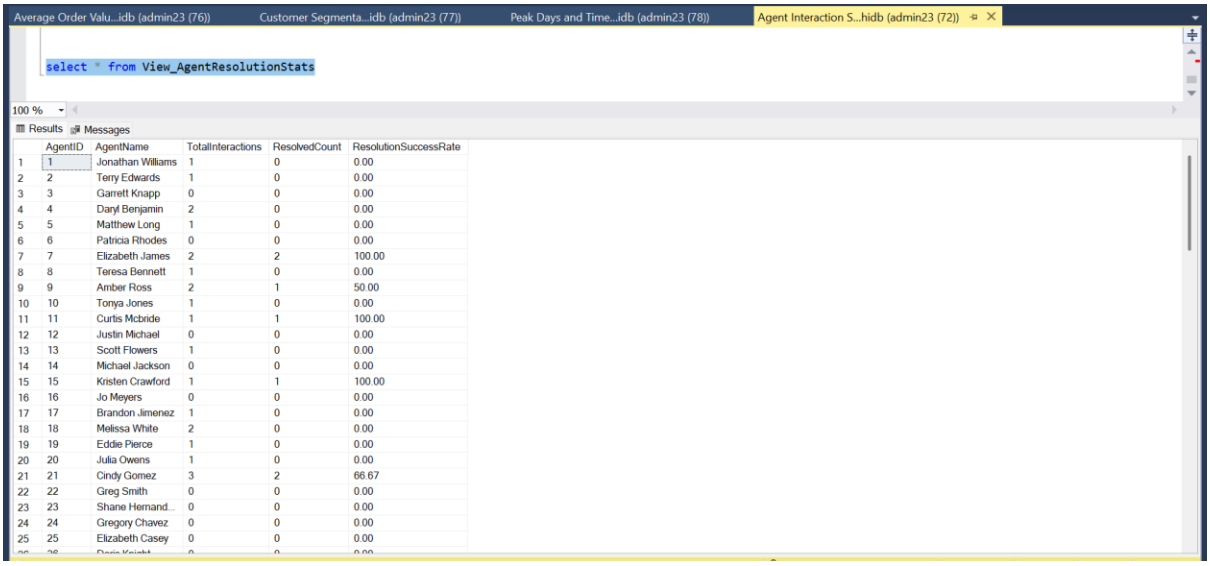
	TransactionType	DayOfWeek	HourOfDay	TransactionCount
1	Online	Monday	0	1
2	Online	Sunday	0	1
3	Online	Tuesday	0	1
4	Online	Friday	1	1
5	Online	Saturday	1	1
6	Online	Tuesday	1	1
7	Online	Monday	2	2
8	Online	Saturday	2	1
9	Online	Wednesday	2	1
10	Online	Saturday	3	2
11	Online	Sunday	3	1
12	Online	Tuesday	3	1
13	Online	Friday	4	1
14	Online	Thursday	4	2
15	Online	Tuesday	4	1
16	Online	Wednesday	4	2
17	Online	Friday	5	3
18	Online	Sunday	5	1
19	Online	Tuesday	5	1
20	Online	Wednesday	5	2
21	Online	Monday	6	1
22	Online	Sunday	6	1
23	Online	Thursday	6	1
24	Online	Wednesday	6	1
25	Online	Friday	7	1
26	Online	Tuesday	7	2
27	Online	Monday	8	1
28	Online	Saturday	8	2

Query executed successfully. | deepthiserver1.database.win... | admin23 (78) | deepthidb | 00:00:00 | 151 rows

4. View 4 - for Number of interactions and resolution success rates per agent (ResolutionStatus).


Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))

```
CREATE OR ALTER VIEW View_AgentResolutionStats AS
SELECT
    a.AgentID,
    a.Name AS AgentName,
    COUNT(csi.InteractionID) AS TotalInteractions,
    SUM(CASE WHEN csi.ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) AS ResolvedCount,
    CASE
        WHEN COUNT(csi.InteractionID) = 0 THEN 0
        ELSE CAST(
            (SUM(CASE WHEN csi.ResolutionStatus = 'Resolved' THEN 1 ELSE 0 END) * 100.0) /
            COUNT(csi.InteractionID)
            AS DECIMAL(5,2)
        )
    END AS ResolutionSuccessRate
FROM
    Agents a
LEFT JOIN
    CustomerServiceInteraction csi ON a.AgentID = csi.AgentID
GROUP BY
    a.AgentID, a.Name;
```

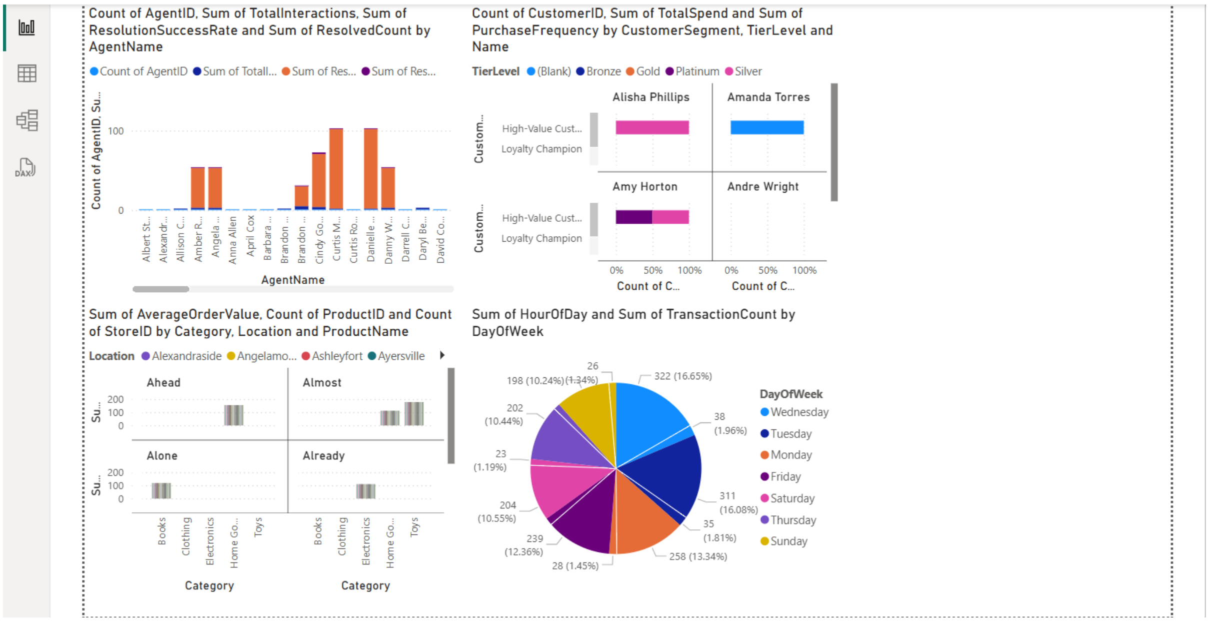


Average Order Valu...idb (admin23 (76))
Customer Segmenta...idb (admin23 (77))
Peak Days and Time...idb (admin23 (78))
Agent Interaction S...hidb (admin23 (72))
100 %
Results
Messages

```
select * from View_AgentResolutionStats
```

	AgentID	AgentName	TotalInteractions	ResolvedCount	ResolutionSuccessRate
1	1	Jonathan Williams	1	0	0.00
2	2	Terry Edwards	1	0	0.00
3	3	Garrett Knapp	0	0	0.00
4	4	Daryl Benjamin	2	0	0.00
5	5	Matthew Long	1	0	0.00
6	6	Patricia Rhodes	0	0	0.00
7	7	Elizabeth James	2	2	100.00
8	8	Teresa Bennett	1	0	0.00
9	9	Amber Ross	2	1	50.00
10	10	Tonya Jones	1	0	0.00
11	11	Curtis Mcbride	1	1	100.00
12	12	Justin Michael	0	0	0.00
13	13	Scott Flowers	1	0	0.00
14	14	Michael Jackson	0	0	0.00
15	15	Kristen Crawford	1	1	100.00
16	16	Jo Meyers	0	0	0.00
17	17	Brandon Jimenez	1	0	0.00
18	18	Melissa White	2	0	0.00
19	19	Eddie Pierce	1	0	0.00
20	20	Julia Owens	1	0	0.00
21	21	Cindy Gomez	3	2	66.67
22	22	Greg Smith	0	0	0.00
23	23	Shane Hernand...	0	0	0.00
24	24	Gregory Chavez	0	0	0.00
25	25	Elizabeth Casey	0	0	0.00
26	26	Doris Knight	0	0	0.00

Power BI Visualization:



Count of AgentID, Sum of TotalInteractions, Sum of ResolutionSuccessRate and Sum of ResolvedCount by AgentName

Count of CustomerID, Sum of TotalSpend and Sum of PurchaseFrequency by CustomerSegment, TierLevel and Name

Sum of AverageOrderValue, Count of ProductID and Count of StoreID by Category, Location and ProductName

Sum of HourOfDay and Sum of TransactionCount by DayOfWeek