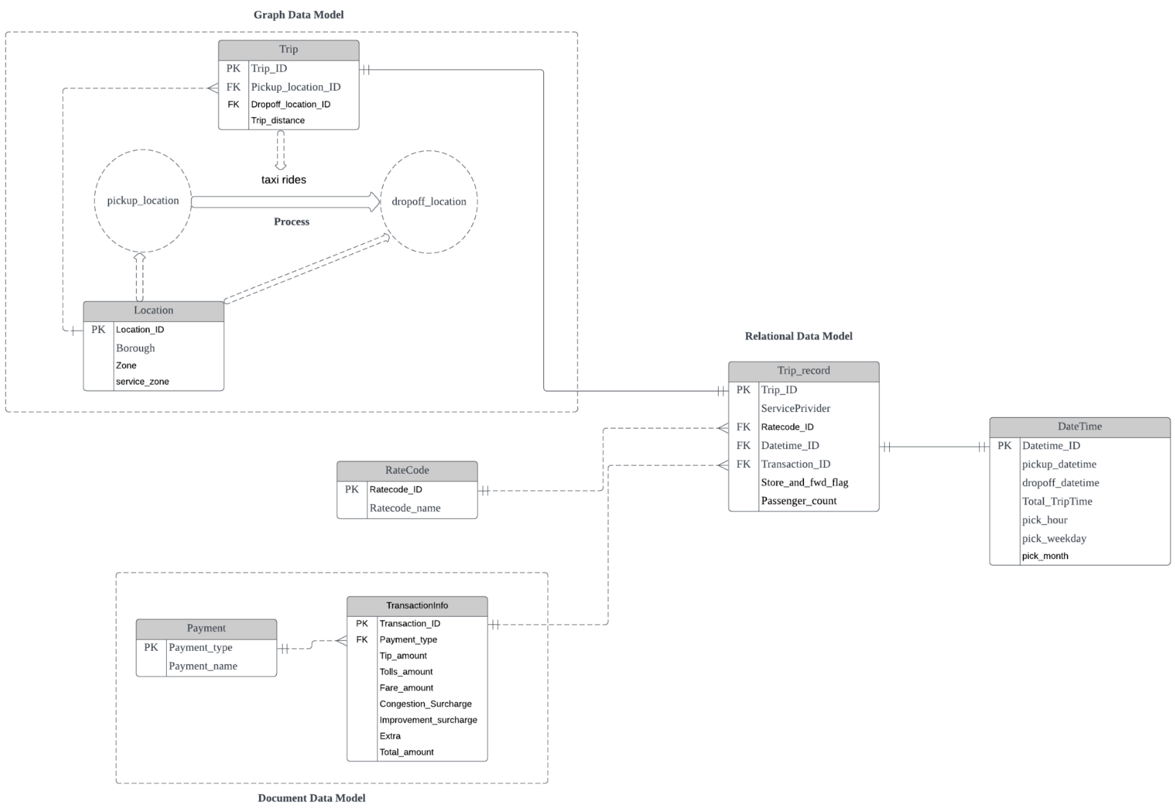
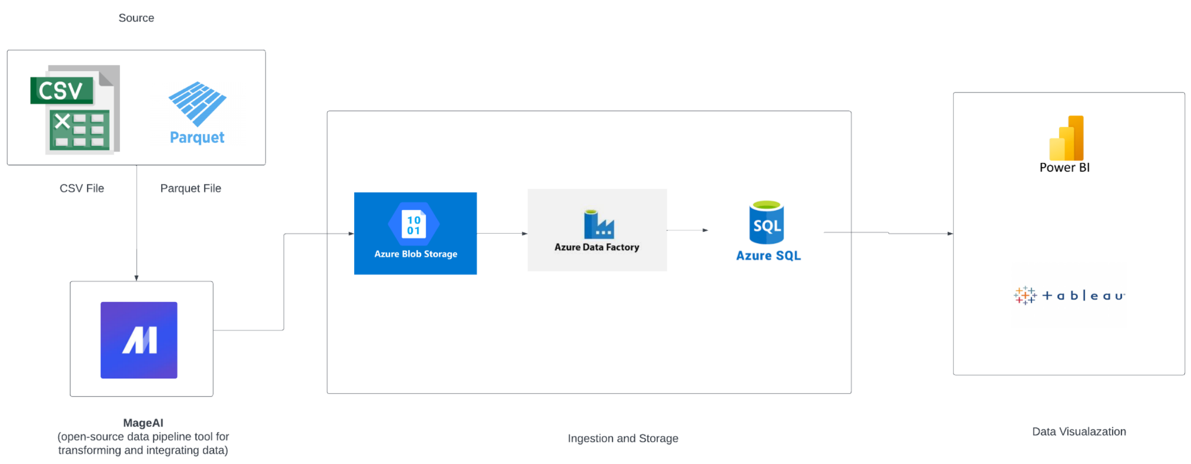
Urban Mobility Data Management System P3 using Azure SQL

Team5 Members:
1. Veenadharini Shukla (NU ID: 002704948)
2. Vikrant Satish Pawar (NU ID: 002772104)
3. Lokesh Mohan Jeswani (NU ID: 002795957)
4. Xin Shen (NU ID: 002728429)
5. Zequn Cao (NU ID: 002747196)

ER Diagram



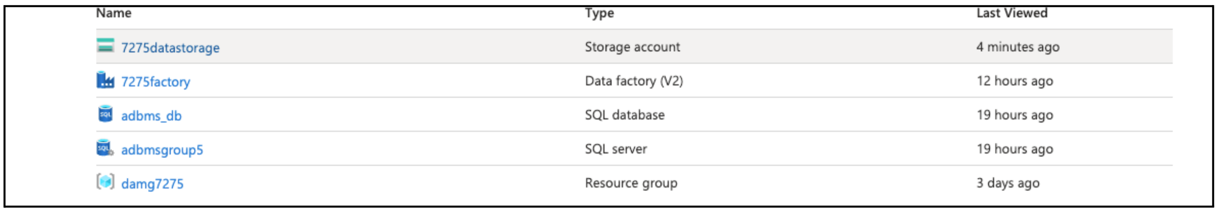
Architectural Diagram



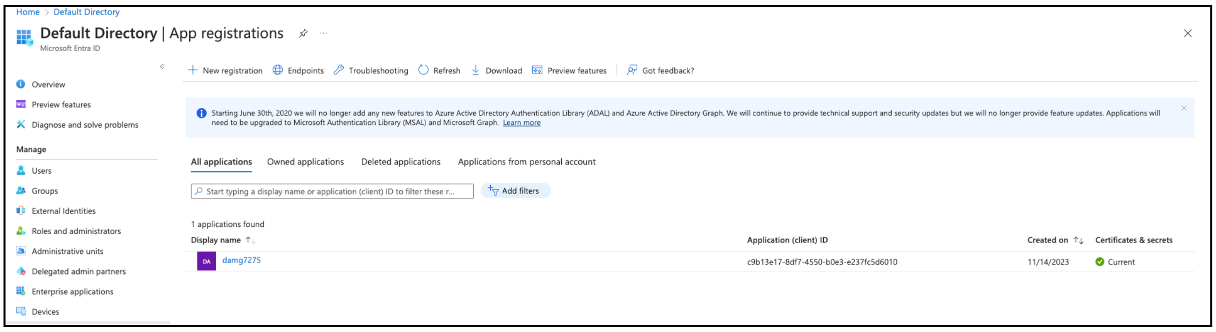
Implementation

Part 1: Azure Tools Deployment

7275darastorage: One blob storage to store cleaned dataset from MageAI
7275factory: One data factory for loading data into Azure Sql database
adbmsgroup5 & adbms_db: One SQL Server and SQL DataBase
damg7275: One application in Microsoft Entra ID which will be used for connect between Azure blob storage and MageAI



Name	Type	Last Viewed
7275datastorage	Storage account	4 minutes ago
7275factory	Data factory (V2)	12 hours ago
adbms_db	SQL database	19 hours ago
adbmsgroup5	SQL server	19 hours ago
damg7275	Resource group	3 days ago



Home > Default Directory

Default Directory | App registrations

Microsoft Entra ID

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Start typing a display name or application (client) ID to filter these r...

1 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
DA damg7275	c9b13e17-8df7-4550-b0e3-e237fc5d6010	11/14/2023	✓ Current

Part 2: Working on unmodified dataset

Converting a dataset from GitHub for use in MageAI, an open-source data pipeline tool, to execute various data preprocessing steps:

- removing column with excessive missing values
- filling missing values in multiple columns using statistical methods
- adjusting specific values in a column to a new standard value
- renaming several columns for better clarity
- adding some columns which will be used as primary keys for dimension tables
- converting data types

steps screenshots:

- ❖ data load

```
❖ import io
❖ import pandas as pd
❖ import requests
❖ if 'data_loader' not in globals():
❖     from mage_ai.data_preparation.decorators import data_loader
❖ if 'test' not in globals():
❖     from mage_ai.data_preparation.decorators import test
❖
❖
```

```python
@data_loader
def load_data_from_api(*args, **kwargs):
    """
    Template for loading data from API
    """
    url =
'https://raw.githubusercontent.com/xinwen88888/DAMG7275/main/final_dataset(n
ot_modified).csv'
    response = requests.get(url)

    return pd.read_csv(io.StringIO(response.text), sep=',')


@test
def test_output(output, *args) -> None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'
```

❖ data preprocessing

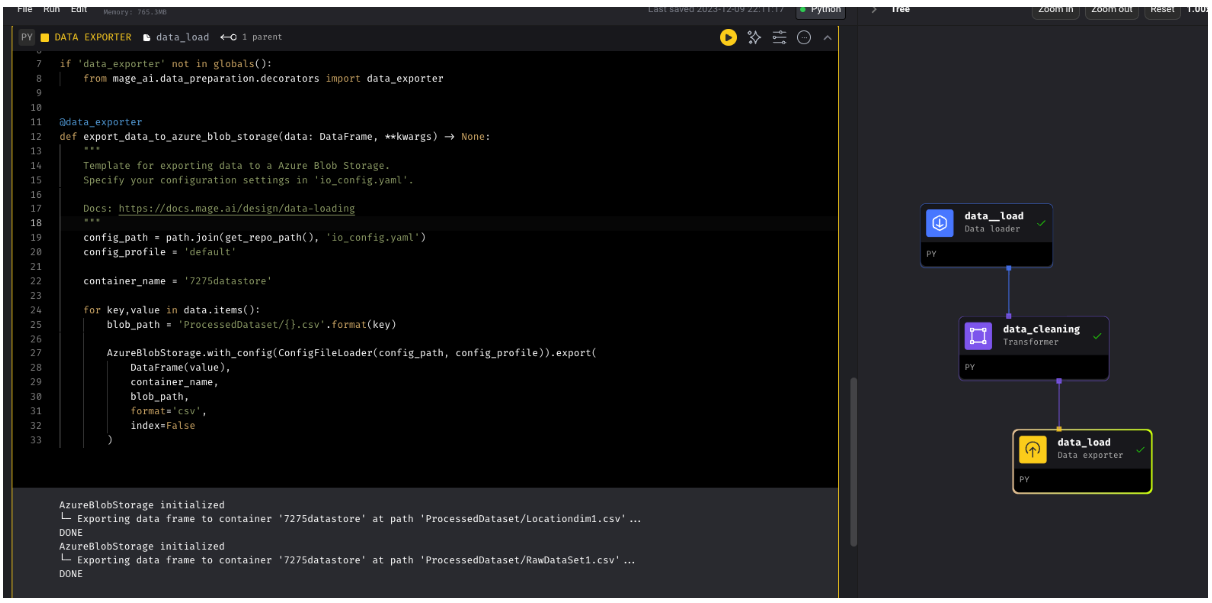
```python
if 'transformer' not in globals():
    from mage_ai.data_preparation.decorators import transformer
if 'test' not in globals():
    from mage_ai.data_preparation.decorators import test
import pandas as pd
import numpy as np
import requests
import io


@transformer
def transform(df, *args, **kwargs):
    df.drop(columns=['airport_fee'],inplace=True) #since airport has too
much NA values, so it will be eliminated from our futher analysis
    df.fillna({'passenger_count':round(df['passenger_count'].mean()),
           'RatecodeID':df['RatecodeID'].mode()[0],
           'store_and_fwd_flag':df['store_and_fwd_flag'].mode()[0],

'congestion_surcharge':round(df['congestion_surcharge'].mean(),2)},inplace=T
rue)

    df['RatecodeID'] = df['RatecodeID'].where(df['RatecodeID']!=99.0,6.0)

    del df['VendorID']

    ServiceProvider = ['Uber','Lyft','Juno','Via']
    proportions = [0.48, 0.32, 0.12, 0.08]
    df['ServiceProvider'] =
np.random.choice(ServiceProvider,size=len(df),p=proportions)

    df.rename(columns={
    'tpep_pickup_datetime':'pickup_datetime',
    'tpep_dropoff_datetime':'dropoff_datetime',
    'PULocationID':'pickup_location',
    'DOLocationID':'droppff_location'},inplace=True)

    df['pickup_datetime'] = pd.to_datetime(df['pickup_datetime'])
    df['dropoff_datetime'] = pd.to_datetime(df['dropoff_datetime'])
    df['RatecodeID'] = df['RatecodeID'].astype('int')
    df['passenger_count'] = df['passenger_count'].astype('int')

    cols = df.columns.tolist()
    cols.remove('ServiceProvider')
    cols.insert(0,'ServiceProvider')
    df = df[cols]
    df.columns = [col.title() for col in df.columns]

    df['TripID'] = df.index
    df['TransactionID'] = df.index
    df['DateTimeID'] = df.index

    url = 
'https://raw.githubusercontent.com/xinwen88888/DAMG7275/main/taxi%2B_zone_lookup.csv'
    response = requests.get(url)
    Location = pd.read_csv(io.StringIO(response.text), sep=',')
    Location.fillna('Unknown',inplace=True)

    return {'Locationdim':Location.to_dict(orient='dict'),
            'RawDataSet':df.to_dict(orient='dict')}


@test
def test_output(output, *args) -> None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'
```

Part 3 Export cleaned dataset to Azure Blob Storage in .csv type

```python
from mage_ai.settings.repo import get_repo_path
from mage_ai.io.azure_blob_storage import AzureBlobStorage
from mage_ai.io.config import ConfigFileLoader
from pandas import DataFrame
from os import path

if 'data_exporter' not in globals():
    from mage_ai.data_preparation.decorators import data_exporter


@data_exporter
def export_data_to_azure_blob_storage(data: DataFrame, **kwargs) -> None:
    """
    Template for exporting data to a Azure Blob Storage.
    Specify your configuration settings in 'io_config.yaml'.

    Docs: https://docs.mage.ai/design/data-loading
    """
    config_path = path.join(get_repo_path(), 'io_config.yaml')
```

```
●       config_profile = 'default'
●
●       container_name = '7275datastore'
●
●       for key,value in data.items():
●           blob_path = 'ProcessedDataset/{}.csv'.format(key)
●
●           AzureBlobStorage.with_config(ConfigFileLoader(config_path,
    config_profile)).export(
●               DataFrame(value),
●               container_name,
●               blob_path,
●               format='csv',
●               index=False
●           )
```

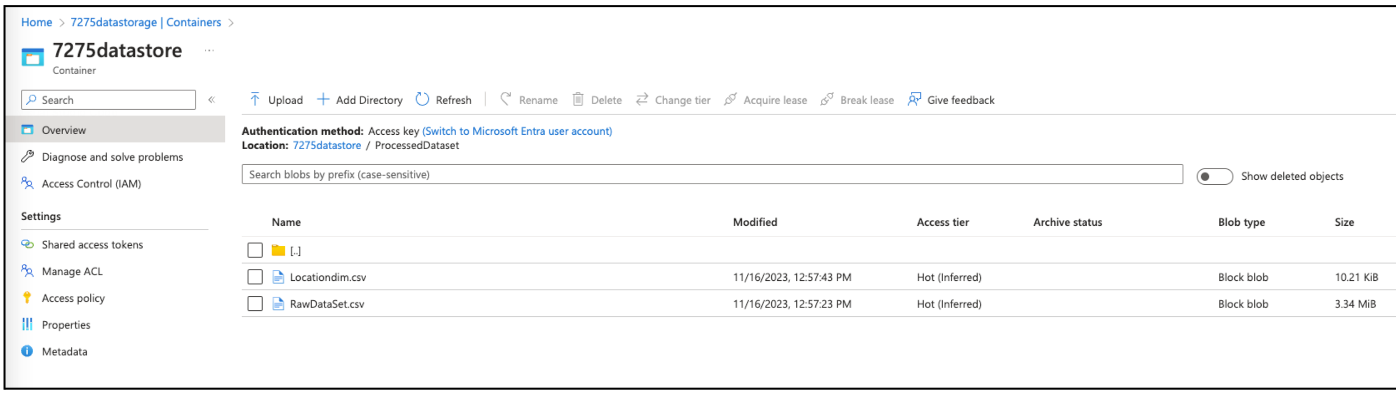
MageAI process screenshot:



```
if 'data_exporter' not in globals():
    from mage_ai.data_preparation.decorators import data_exporter


@data_exporter
def export_data_to_azure_blob_storage(data: DataFrame, **kwargs) -> None:
    """
    Template for exporting data to a Azure Blob Storage.
    Specify your configuration settings in 'io_config.yaml'.

    Docs: https://docs.mage.ai/design/data-loading
    """
    config_path = path.join(get_repo_path(), 'io_config.yaml')
    config_profile = 'default'

    container_name = '7275datastore'

    for key,value in data.items():
        blob_path = 'ProcessedDataset/{}.csv'.format(key)

        AzureBlobStorage.with_config(ConfigFileLoader(config_path, config_profile)).export(
            DataFrame(value),
            container_name,
            blob_path,
            format='csv',
            index=False
        )
```

```
AzureBlobStorage initialized
└ Exporting data frame to container '7275datastore' at path 'ProcessedDataset/Locationdim1.csv' ...
DONE
AzureBlobStorage initialized
└ Exporting data frame to container '7275datastore' at path 'ProcessedDataset/RawDataSet1.csv' ...
DONE
```

Result: two cleaned csv files are available in blob storage container



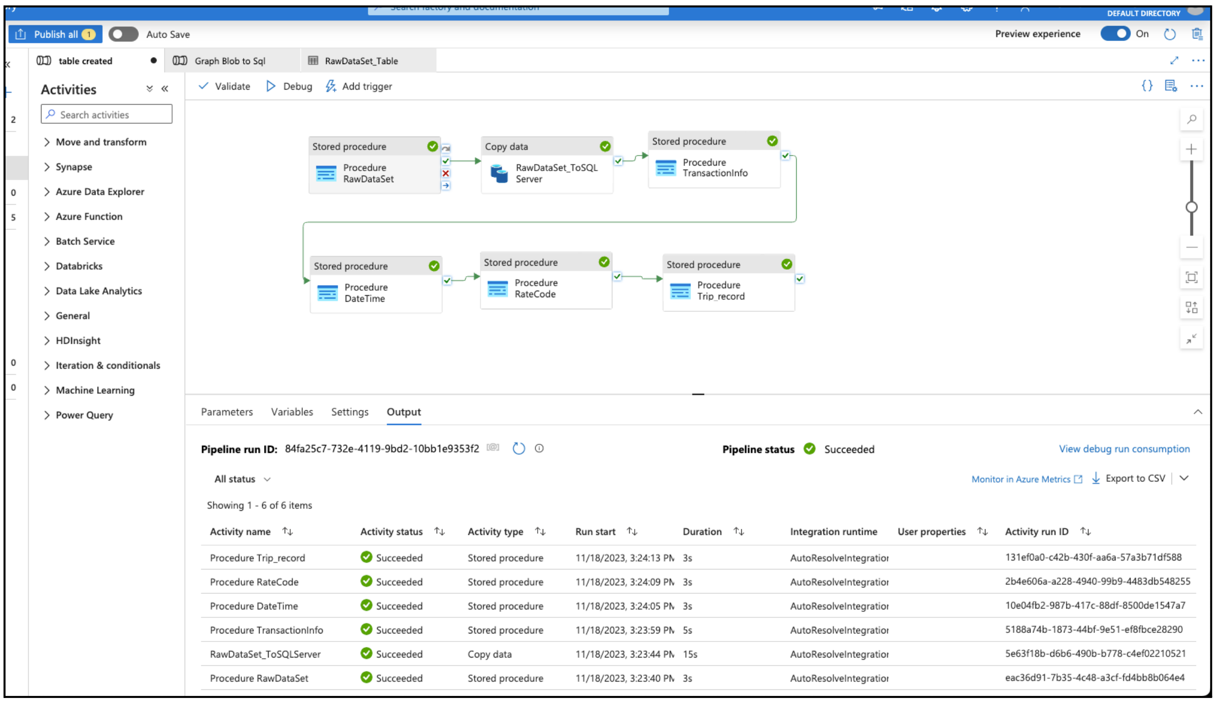
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: 7275datastore / ProcessedDataset

Name	Modified	Access tier	Archive status	Blob type	Size
[..]					
Locationdim.csv	11/16/2023, 12:57:43 PM	Hot (Inferred)		Block blob	10.21 KiB
RawDataSet.csv	11/16/2023, 12:57:23 PM	Hot (Inferred)		Block blob	3.34 MiB

Part 3: Data pipeline deployment

1) Document model and relational model deployment:

using activity ‘Stored procedure’ and ‘Copy’ to perform table creation and data insertion.



Pipeline run ID: 84fa25c7-732e-4119-9bd2-10bb1e9353f2

Pipeline status ✓ Succeeded

View debug run consumption

All status

Monitor in Azure Metrics · Export to CSV

Showing 1 - 6 of 6 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID
Procedure Trip_record	✓ Succeeded	Stored procedure	11/18/2023, 3:24:13 PM	3s	AutoResolveIntegratior		131ef0a0-c42b-430f-aa6a-57a3b71df588
Procedure RateCode	✓ Succeeded	Stored procedure	11/18/2023, 3:24:09 PM	3s	AutoResolveIntegratior		2b4e606a-a228-4940-99b9-4483db548255
Procedure DateTime	✓ Succeeded	Stored procedure	11/18/2023, 3:24:05 PM	3s	AutoResolveIntegratior		10e04fb2-987b-417c-88df-8500de1547a7
Procedure TransactionInfo	✓ Succeeded	Stored procedure	11/18/2023, 3:23:59 PM	5s	AutoResolveIntegratior		5188a74b-1873-44bf-9e51-ef8fbce28290
RawDataSet_ToSQLServer	✓ Succeeded	Copy data	11/18/2023, 3:23:44 PM	15s	AutoResolveIntegratior		5e63f18b-d6b6-490b-b778-c4ef02210521
Procedure RawDataSet	✓ Succeeded	Stored procedure	11/18/2023, 3:23:40 PM	3s	AutoResolveIntegratior		eac36d91-7b35-4c48-a3cf-fd4bb8b064e4

code:

```
CREATE PROCEDURE Procedure_RawDataSet
AS
BEGIN
    IF NOT EXISTS (
        SELECT *
        FROM sys.objects
        WHERE object_id = OBJECT_ID(N'[dbo].[RawDataSet]')
        AND type in (N'U')
    )
    BEGIN
        create table RawDataSet(
            TripID int primary key,
            DateTimeID int,
            TransactionID int,
            RatecodeID int,
            Serviceprovider varchar(50),
            Pickup_Datetime varchar(50),
            Dropoff_Datetime varchar(50),
            Passenger_Count int,
            Trip_Distance decimal(10,4),
            Store_And_Fwd_Flag varchar(50),
            Pickup_Location int,
            Droppff_Location int,
            Payment_Type int,
            Fare_Amount decimal(10,4),
            Extra decimal(10,4),
            Mta_Tax decimal(10,4),
            Tip_Amount decimal(10,4),
            Tolls_Amount decimal(10,4),
            Improvement_Surcharge decimal(10,4),
            Total_Amount decimal(10,4),
            Congestion_Surcharge decimal(10,4)
            );
    END
    ELSE
    BEGIN
        TRUNCATE TABLE [dbo].[RawDataSet];
    END
END

CREATE PROCEDURE Procedure_TransactionInfo
AS
BEGIN
    IF NOT EXISTS (
        SELECT *
        FROM sys.objects
        WHERE object_id = OBJECT_ID(N'[dbo].[TransactionInfo]')
        AND type in (N'U')
    )
    BEGIN
        -- if table not exists, then create
        CREATE TABLE TransactionInfo(
            TransactionID INT PRIMARY KEY,
            PaymentInfo NVARCHAR(MAX),
            CostInfo NVARCHAR(MAX)
        );
    END
    ELSE
    BEGIN
        -- Delete data from the table if it exists
        truncate table [dbo].[TransactionInfo];
    END

    -- insert action
    INSERT INTO TransactionInfo
    select c.TransactionID,
        (
            select Payment_Type as PaymentID,
                   case Payment_Type
                        when 0 then 'Unknown'
                        when 1 then 'Credit card'
                        when 2 then 'Cash'
                        when 3 then 'Dispute'
                        else 'No charge' end as PaymentName
                    from RawDataSet a
            where a.TransactionID = c.TransactionID
            for json path
        ) as PaymentInfo,
        (
            select Tip_Amount,
                   Tolls_Amount,
                   Improvement_Surcharge,
                   Total_Amount,
                   Congestion_Surcharge,
                   Fare_Amount,
                   Extra,
                   Mta_Tax from RawDataSet b
            where b.TransactionID = c.TransactionID
            for json path
        ) as CostInfo
        from RawDataSet c
END


CREATE PROCEDURE Procedure_DateTime
AS
BEGIN
    -- Create DateTime Table from RawDataSet with computed columns
    IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[DateTime]') AND type in (N'U'))
    BEGIN
        CREATE TABLE DateTime (
            DatetimeID INT PRIMARY KEY,
            pickup_datetime DATETIME,
            dropoff_datetime DATETIME,
            Total_TripTime AS DATEDIFF(MINUTE, pickup_datetime, dropoff_datetime),
            pick_hour AS DATEPART(HOUR, pickup_datetime),
            pick_weekday AS DATENAME(WEEKDAY, pickup_datetime),
            pick_month AS MONTH(pickup_datetime)
        );
    end
    ELSE
    BEGIN
        -- Delete data from the table if it exists
        truncate table [dbo].[DateTime];
    END

    -- Populate DateTime Table
    INSERT INTO DateTime (DatetimeID, pickup_datetime, dropoff_datetime)
    SELECT DISTINCT DateTimeID, CAST(Pickup_Datetime AS DATETIME), CAST(Dropoff_Datetime AS DATETIME)
    FROM RawDataSet
    WHERE DateTimeID IS NOT NULL;

END



CREATE PROCEDURE Procedure_RateCode
AS
BEGIN
```

```
•     -- Create RateCode Table from RawDataSet with specified names for each
  RatecodeID
•     IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
  OBJECT_ID(N'[dbo].[RateCode]') AND type in (N'U'))
•     BEGIN
•         CREATE TABLE RateCode (
•             RatecodeID INT PRIMARY KEY,
•             Ratecode_name NVARCHAR(255)
•         );
•     END
•     ELSE
•     BEGIN
•         -- Delete data from the table if it exists
•         truncate table [dbo].[RateCode];
•     END
•         -- Populate RateCode Table with names based on RatecodeID
•     INSERT INTO RateCode (RatecodeID, Ratecode_name)
•     SELECT DISTINCT
•         RatecodeID,
•         CASE
•             WHEN RatecodeID = 1 THEN 'Standard rate'
•             WHEN RatecodeID = 2 THEN 'JFK'
•             WHEN RatecodeID = 3 THEN 'Newark'
•             WHEN RatecodeID = 4 THEN 'Nassau or Westchester'
•             WHEN RatecodeID = 5 THEN 'Negotiated fare'
•             WHEN RatecodeID = 6 THEN 'Group ride'
•             ELSE 'Other'
•         END
•     FROM RawDataSet
•     WHERE RatecodeID IS NOT NULL;
•
• END
•
•
• CREATE PROCEDURE Procedure_Trip_record
• AS
• BEGIN
•     -- Create Trip_record Table from RawDataSet
•     IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
  OBJECT_ID(N'[dbo].[Trip_record]') AND type in (N'U'))
•     BEGIN
•         CREATE TABLE Trip_record (
•             TripID INT PRIMARY KEY,
•             Serviceprovider NVARCHAR(255),
•             RatecodeID INT,
•             DatetimeID INT,
•             TransactionID INT,
•             Store_and_fwd_flag CHAR(1),
•             Passenger_count INT
•         );
•     END
•     ELSE
•     BEGIN
•         -- Delete data from the table if it exists
•           truncate table [dbo].[Trip_record];
•       END
•
•           -- Populate Trip_record Table
•           INSERT INTO Trip_record (TripID, Serviceprovider, RatecodeID,
    DatetimeID, TransactionID, Store_and_fwd_flag, Passenger_count)
•           SELECT DISTINCT TripID, Serviceprovider, RatecodeID, DatetimeID,
    TransactionID, Store_and_fwd_flag, Passenger_count
•           FROM RawDataSet
•           WHERE TripID IS NOT NULL;
•   END
```

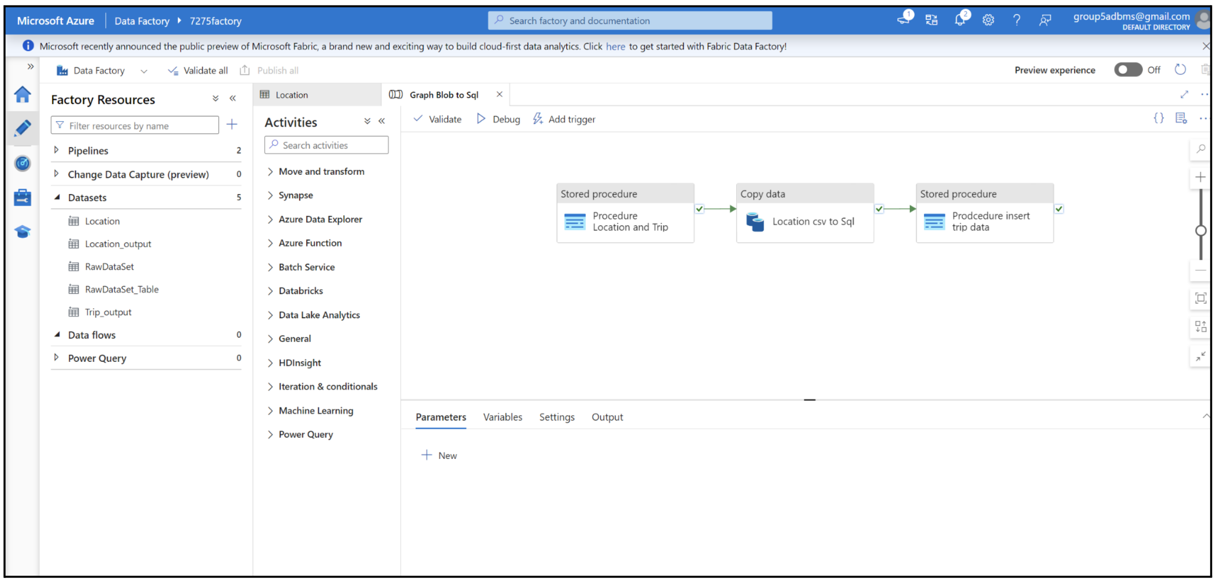
2) Graph Model deployment

Created a stored procedure for the creation of the graph tables using the below code:

```
    -- Drop the stored procedure if it already exists
•   IF EXISTS ( SELECT  *  FROM  sys.objects  WHERE  object_id =
    OBJECT_ID(N '[dbo].[Procedure_GraphTables]' )  AND  type  in  (N 'P'
    ))
•       DROP PROCEDURE  [dbo].[Procedure_GraphTables];
•   GO
•
•   --Create the stored procedure
•   CREATE PROCEDURE  [dbo].[Procedure_GraphTables]
•   AS
•   BEGIN
•       -- Create Location Node Table if it does not exist
•       IF  NOT  EXISTS ( SELECT  *  FROM  sys.objects  WHERE  object_id =
    OBJECT_ID(N '[dbo].[Location]' )  AND  type  in  (N 'U' ))
•       BEGIN
•           CREATE TABLE  [dbo].[Location] (
•               LocationID  INT PRIMARY KEY ,
•               Borough NVARCHAR(100),
•               Zone NVARCHAR(100),
•               service_zone NVARCHAR(100)
•           )  AS  NODE;
•       END
•           ELSE
o       BEGIN
o           -- Delete data from the table if it exists
o           DELETE FROM [dbo].[Location];
o       END
•
•
•
```

```

    -- Create Trip Edge Table if it does not exist
    IF  NOT  EXISTS ( SELECT  *  FROM  sys.objects  WHERE  object_id =
OBJECT_ID(N '[dbo].[Trip]' )  AND  type  in  (N 'U' ))
    BEGIN
        CREATE TABLE  [dbo].[Trip] (
            TripID  INT PRIMARY KEY ,
            Passenger_Count  INT ,
            Trip_Distance  FLOAT
             )  AS  EDGE;
    END

END ;
GO
```

Created an Azure Data Factory pipeline to run the stored procedures and populate data into the Location table and Trip table


Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

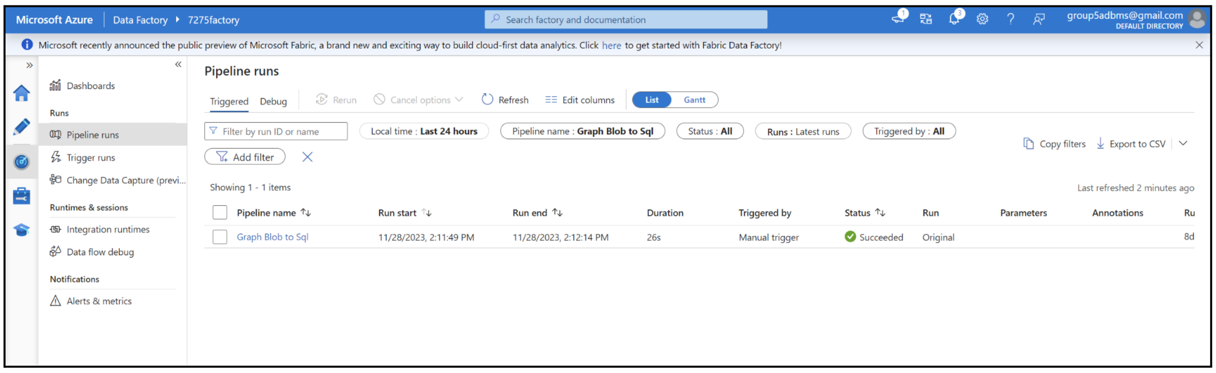
We first used the Stored Procedure for populating the Trip table and then implemented it in a dataflow activity in the pipeline

Stored procedure:

```
CREATE OR ALTER PROCEDURE  [dbo].[InsertTripData]
AS
BEGIN
    -- Insert data into the Trip table by joining RawDataset with
Location to get the $node_id
```

```
    INSERT INTO  [dbo].[Trip] ($from_id, $to_id, TripID,
Passenger_Count, Trip_Distance)
    SELECT
        locFrom.$node_id,
        locTo.$node_id,
        rd.TripID,
        rd.Passenger_Count,
        rd.Trip_Distance
    FROM
        [dbo].[RawDataset]rd
    INNER JOIN
        [dbo].[Location] locFrom  ON  rd.Pickup_Location =
locFrom.LocationID
    INNER JOIN
        [dbo].[Location] locTo  ON  rd.Droppff_Location =
locTo.LocationID;
END ;
GO
```

Successful execution of the pipeline with stored procedure:



Microsoft Azure | Data Factory ▸ 7275factory

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Pipeline runs

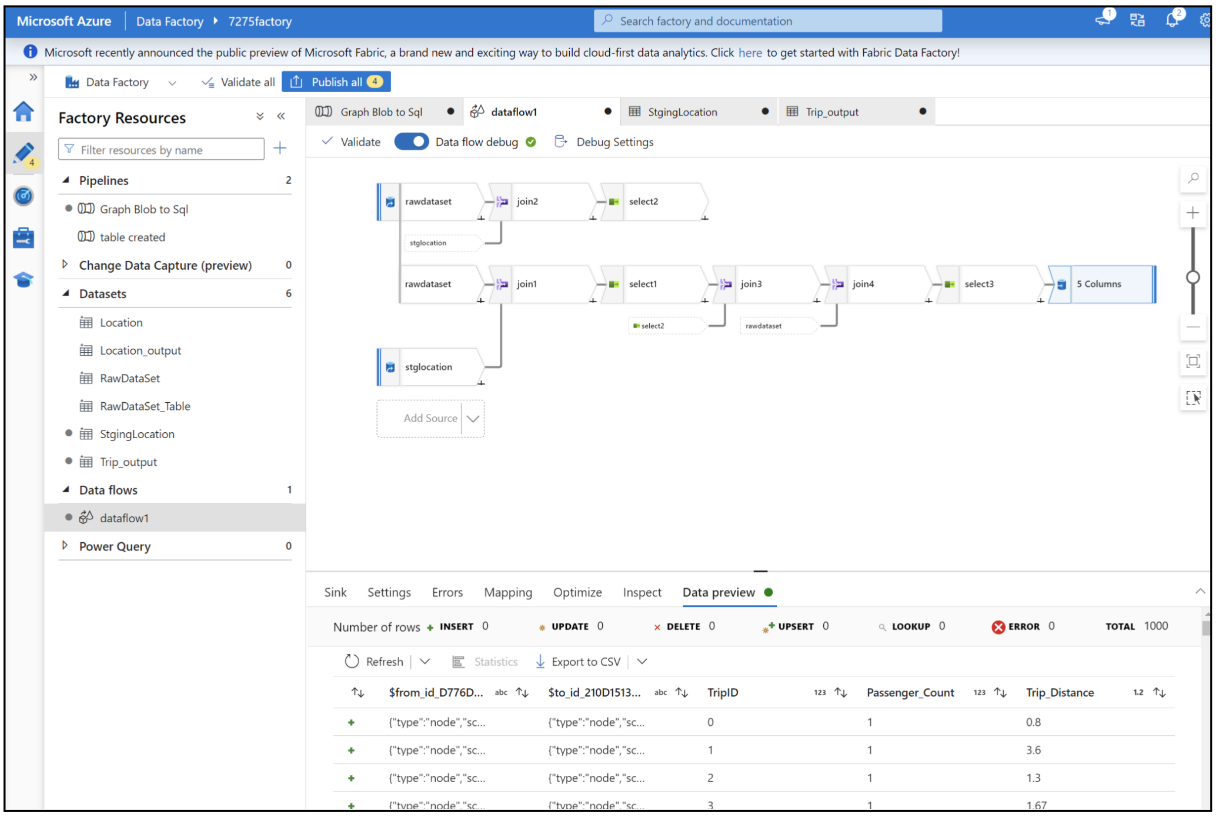
Local time : Last 24 hours | Pipeline name : Graph Blob to Sql | Status : All | Runs : Latest runs | Triggered by : All

Showing 1 - 1 items

Last refreshed 2 minutes ago

Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run	Parameters	Annotations	Ru
Graph Blob to Sql	11/28/2023, 2:11:49 PM	11/28/2023, 2:12:14 PM	26s	Manual trigger	Succeeded	Original			8d

Dataflow implementation:

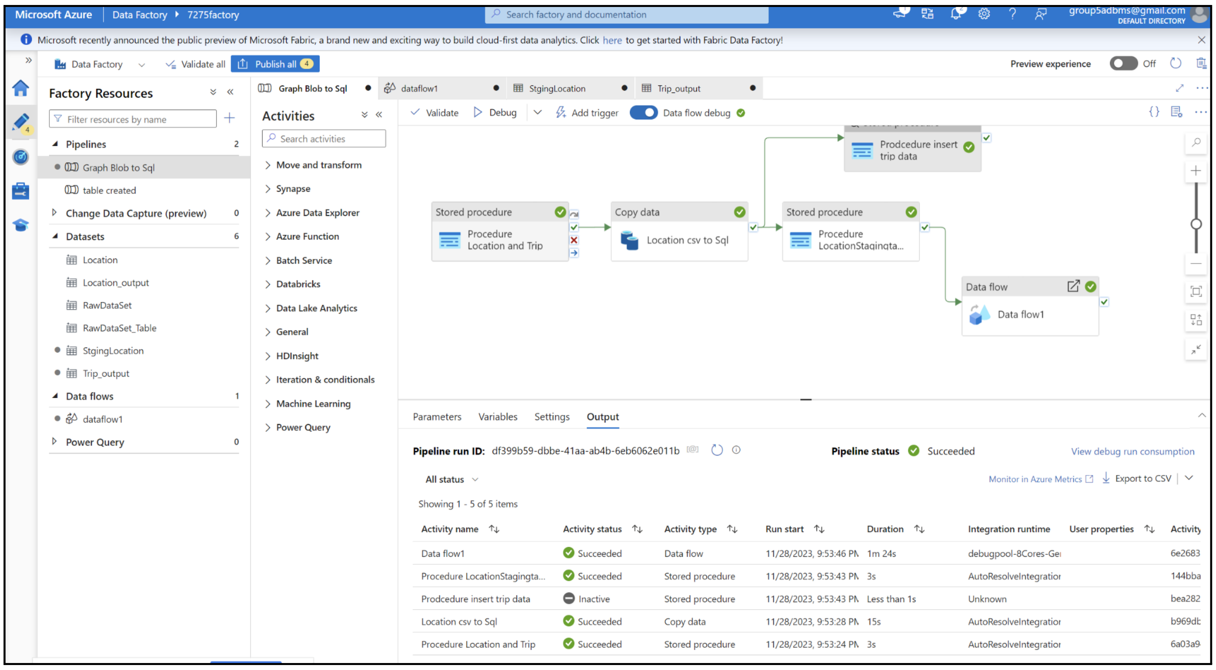


Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Number of rows: INSERT 0 · UPDATE 0 · DELETE 0 · UPSERT 0 · LOOKUP 0 · ERROR 0 · TOTAL 1000

	$from_id_D776D...	$to_id_210D1513...	TripID	Passenger_Count	Trip_Distance
+	{"type":"node","sc...	{"type":"node","sc...	0	1	0.8
+	{"type":"node","sc...	{"type":"node","sc...	1	1	3.6
+	{"type":"node","sc...	{"type":"node","sc...	2	1	1.3
+	{"type":"node","sc...	{"type":"node","sc...	3	1	1.67

Successful execution of the pipeline with the dataflow (Stored Procedure disabled):



Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Pipeline run ID: df399b59-dbbe-41aa-ab4b-6eb6062e011b

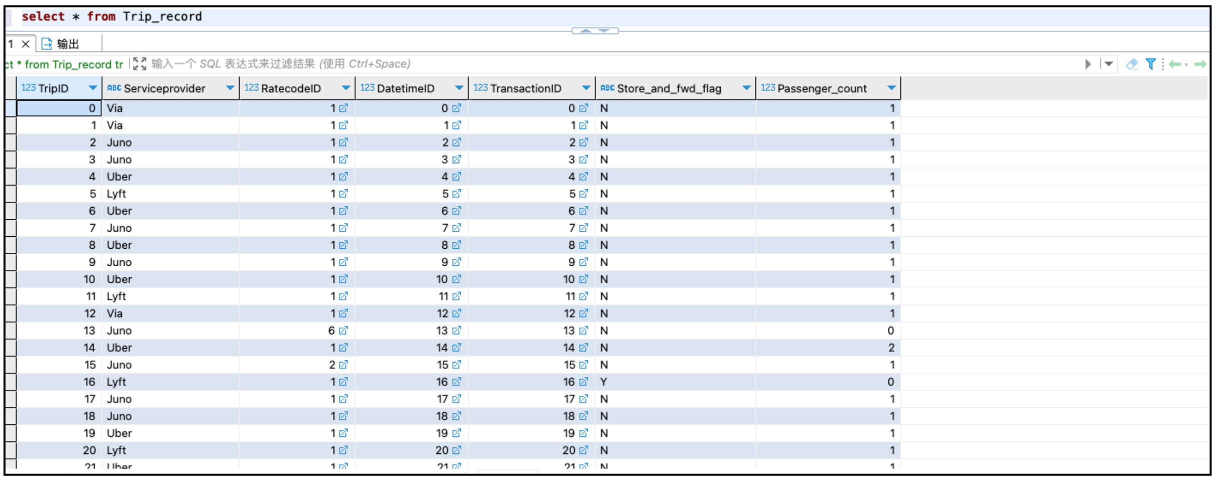
Pipeline status: Succeeded

Showing 1 - 5 of 5 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity
Data flow1	Succeeded	Data flow	11/28/2023, 9:53:46 PM	1m 24s	debugpool-8Cores-Ge		6e2683
Procedure LocationStagingta...	Succeeded	Stored procedure	11/28/2023, 9:53:43 PM	3s	AutoResolveIntegratior		144bba
Prodcedure insert trip data	Inactive	Stored procedure	11/28/2023, 9:53:43 PM	Less than 1s	Unknown		bea282
Location csv to Sql	Succeeded	Copy data	11/28/2023, 9:53:28 PM	15s	AutoResolveIntegratior		b969db
Procedure Location and Trip	Succeeded	Stored procedure	11/28/2023, 9:53:24 PM	3s	AutoResolveIntegratior		6a03a9

Part 4: tables created display in Azure SQL

Trip_record(relational):



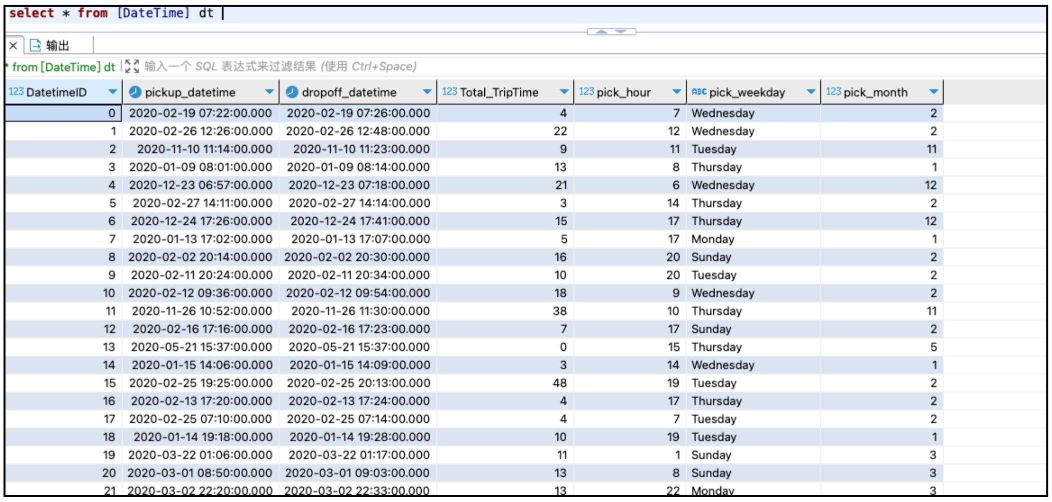
```sql
select * from Trip_record
```

1 × 输出

ct * from Trip_record tr | 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

TripID	Serviceprovider	RatecodeID	DatetimeID	TransactionID	Store_and_fwd_flag	Passenger_count
0	Via	1	0	0	N	1
1	Via	1	1	1	N	1
2	Juno	1	2	2	N	1
3	Juno	1	3	3	N	1
4	Uber	1	4	4	N	1
5	Lyft	1	5	5	N	1
6	Uber	1	6	6	N	1
7	Juno	1	7	7	N	1
8	Uber	1	8	8	N	1
9	Juno	1	9	9	N	1
10	Uber	1	10	10	N	1
11	Lyft	1	11	11	N	1
12	Via	1	12	12	N	1
13	Juno	6	13	13	N	0
14	Uber	1	14	14	N	2
15	Juno	2	15	15	N	1
16	Lyft	1	16	16	Y	0
17	Juno	1	17	17	N	1
18	Juno	1	18	18	N	1
19	Uber	1	19	19	N	1
20	Lyft	1	20	20	N	1
21	Uber	1	21	21	N	1

Datetime(relational):



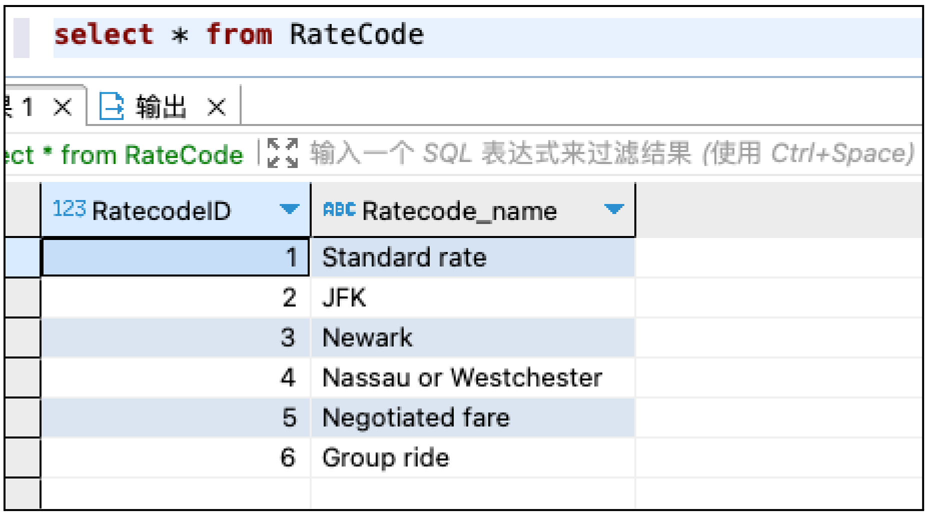
```sql
select * from [DateTime] dt
```

× 输出

from [DateTime] dt | 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

DatetimeID	pickup_datetime	dropoff_datetime	Total_TripTime	pick_hour	pick_weekday	pick_month
0	2020-02-19 07:22:00.000	2020-02-19 07:26:00.000	4	7	Wednesday	2
1	2020-02-26 12:26:00.000	2020-02-26 12:48:00.000	22	12	Wednesday	2
2	2020-11-10 11:14:00.000	2020-11-10 11:23:00.000	9	11	Tuesday	11
3	2020-01-09 08:01:00.000	2020-01-09 08:14:00.000	13	8	Thursday	1
4	2020-12-23 06:57:00.000	2020-12-23 07:18:00.000	21	6	Wednesday	12
5	2020-02-27 14:11:00.000	2020-02-27 14:14:00.000	3	14	Thursday	2
6	2020-12-24 17:26:00.000	2020-12-24 17:41:00.000	15	17	Thursday	12
7	2020-01-13 17:02:00.000	2020-01-13 17:07:00.000	5	17	Monday	1
8	2020-02-02 20:14:00.000	2020-02-02 20:30:00.000	16	20	Sunday	2
9	2020-02-11 20:24:00.000	2020-02-11 20:34:00.000	10	20	Tuesday	2
10	2020-02-12 09:36:00.000	2020-02-12 09:54:00.000	18	9	Wednesday	2
11	2020-11-26 10:52:00.000	2020-11-26 11:30:00.000	38	10	Thursday	11
12	2020-02-16 17:16:00.000	2020-02-16 17:23:00.000	7	17	Sunday	2
13	2020-05-21 15:37:00.000	2020-05-21 15:37:00.000	0	15	Thursday	5
14	2020-01-15 14:06:00.000	2020-01-15 14:09:00.000	3	14	Wednesday	1
15	2020-02-25 19:25:00.000	2020-02-25 20:13:00.000	48	19	Tuesday	2
16	2020-02-13 17:20:00.000	2020-02-13 17:24:00.000	4	17	Thursday	2
17	2020-02-25 07:10:00.000	2020-02-25 07:14:00.000	4	7	Tuesday	2
18	2020-01-14 19:18:00.000	2020-01-14 19:28:00.000	10	19	Tuesday	1
19	2020-03-22 01:06:00.000	2020-03-22 01:17:00.000	11	1	Sunday	3
20	2020-03-01 08:50:00.000	2020-03-01 09:03:00.000	13	8	Sunday	3
21	2020-03-02 22:20:00.000	2020-03-02 22:33:00.000	13	22	Monday	3

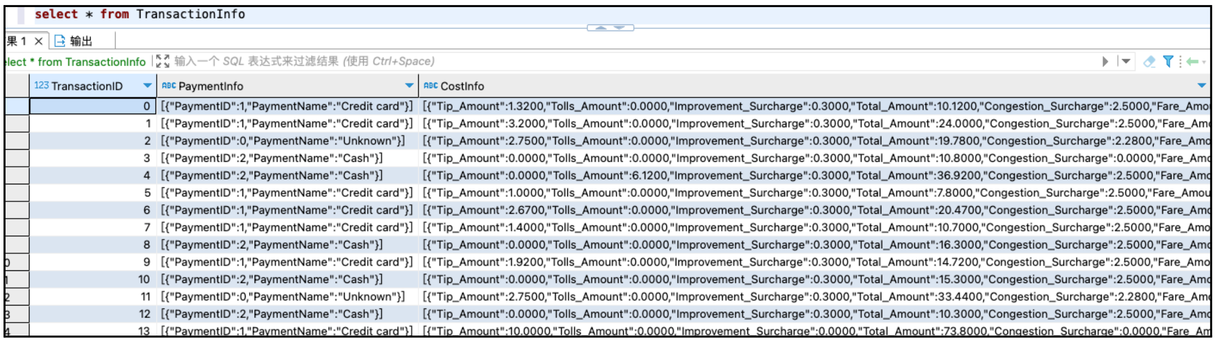
RateCode(relational):



```sql
select * from RateCode
```

RatecodeID	Ratecode_name
1	Standard rate
2	JFK
3	Newark
4	Nassau or Westchester
5	Negotiated fare
6	Group ride

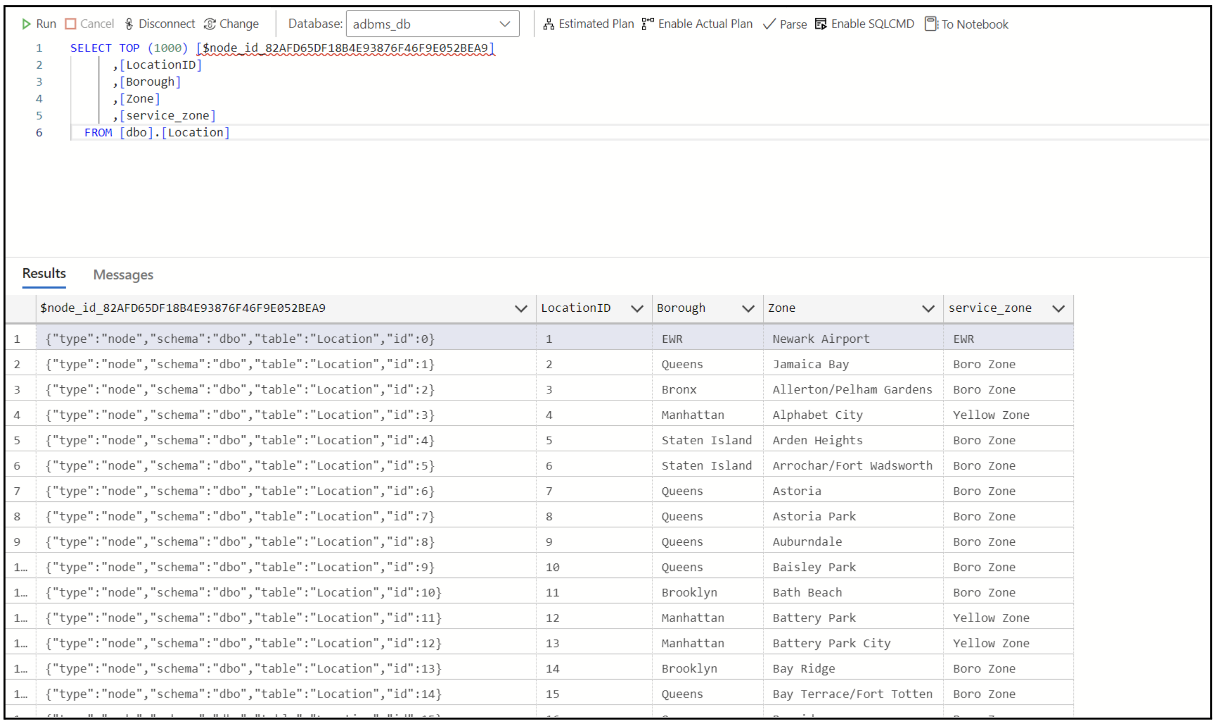
TransactionInfo(document):



```sql
select * from TransactionInfo
```

TransactionID	PaymentInfo	CostInfo
0	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":1.3200,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":10.1200,"Congestion_Surcharge":2.5000,"Fare_Amo
1	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":3.2000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":24.0000,"Congestion_Surcharge":2.5000,"Fare_Am
2	[{"PaymentID":0,"PaymentName":"Unknown"}]	[{"Tip_Amount":2.7500,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":19.7800,"Congestion_Surcharge":2.2800,"Fare_Amo
3	[{"PaymentID":2,"PaymentName":"Cash"}]	[{"Tip_Amount":0.0000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":10.8000,"Congestion_Surcharge":0.0000,"Fare_Amo
4	[{"PaymentID":2,"PaymentName":"Cash"}]	[{"Tip_Amount":0.0000,"Tolls_Amount":6.1200,"Improvement_Surcharge":0.3000,"Total_Amount":36.9200,"Congestion_Surcharge":2.5000,"Fare_Amo
5	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":1.0000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":7.8000,"Congestion_Surcharge":2.5000,"Fare_Amou
6	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":2.6700,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":20.4700,"Congestion_Surcharge":2.5000,"Fare_Amo
7	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":1.4000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":10.7000,"Congestion_Surcharge":2.5000,"Fare_Amo
8	[{"PaymentID":2,"PaymentName":"Cash"}]	[{"Tip_Amount":0.0000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":16.3000,"Congestion_Surcharge":2.5000,"Fare_Amo
9	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":1.9200,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":14.7200,"Congestion_Surcharge":2.5000,"Fare_Amo
10	[{"PaymentID":2,"PaymentName":"Cash"}]	[{"Tip_Amount":0.0000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":15.3000,"Congestion_Surcharge":2.5000,"Fare_Amo
11	[{"PaymentID":0,"PaymentName":"Unknown"}]	[{"Tip_Amount":2.7500,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":33.4400,"Congestion_Surcharge":2.2800,"Fare_Am
12	[{"PaymentID":2,"PaymentName":"Cash"}]	[{"Tip_Amount":0.0000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.3000,"Total_Amount":10.3000,"Congestion_Surcharge":2.5000,"Fare_Amo
13	[{"PaymentID":1,"PaymentName":"Credit card"}]	[{"Tip_Amount":10.0000,"Tolls_Amount":0.0000,"Improvement_Surcharge":0.0000,"Total_Amount":73.8000,"Congestion_Surcharge":0.0000,"Fare_Am

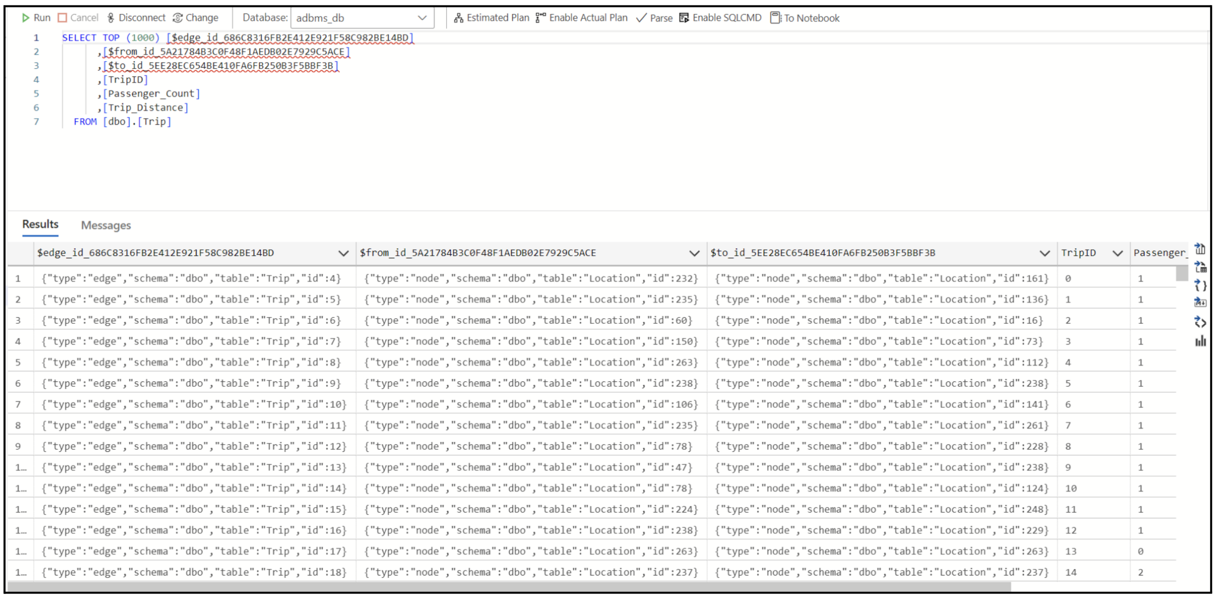
Location (node graph):



```sql
SELECT TOP (1000) [$node_id_82AFD65DF18B4E93876F46F9E052BEA9]
      ,[LocationID]
      ,[Borough]
      ,[Zone]
      ,[service_zone]
  FROM [dbo].[Location]
```

	$node_id_82AFD65DF18B4E93876F46F9E052BEA9	LocationID	Borough	Zone	service_zone
1	{"type":"node","schema":"dbo","table":"Location","id":0}	1	EWR	Newark Airport	EWR
2	{"type":"node","schema":"dbo","table":"Location","id":1}	2	Queens	Jamaica Bay	Boro Zone
3	{"type":"node","schema":"dbo","table":"Location","id":2}	3	Bronx	Allerton/Pelham Gardens	Boro Zone
4	{"type":"node","schema":"dbo","table":"Location","id":3}	4	Manhattan	Alphabet City	Yellow Zone
5	{"type":"node","schema":"dbo","table":"Location","id":4}	5	Staten Island	Arden Heights	Boro Zone
6	{"type":"node","schema":"dbo","table":"Location","id":5}	6	Staten Island	Arrochar/Fort Wadsworth	Boro Zone
7	{"type":"node","schema":"dbo","table":"Location","id":6}	7	Queens	Astoria	Boro Zone
8	{"type":"node","schema":"dbo","table":"Location","id":7}	8	Queens	Astoria Park	Boro Zone
9	{"type":"node","schema":"dbo","table":"Location","id":8}	9	Queens	Auburndale	Boro Zone
1...	{"type":"node","schema":"dbo","table":"Location","id":9}	10	Queens	Baisley Park	Boro Zone
1...	{"type":"node","schema":"dbo","table":"Location","id":10}	11	Brooklyn	Bath Beach	Boro Zone
1...	{"type":"node","schema":"dbo","table":"Location","id":11}	12	Manhattan	Battery Park	Yellow Zone
1...	{"type":"node","schema":"dbo","table":"Location","id":12}	13	Manhattan	Battery Park City	Yellow Zone
1...	{"type":"node","schema":"dbo","table":"Location","id":13}	14	Brooklyn	Bay Ridge	Boro Zone
1...	{"type":"node","schema":"dbo","table":"Location","id":14}	15	Queens	Bay Terrace/Fort Totten	Boro Zone

Trip (edge graph):



```sql
SELECT TOP (1000) [$edge_id_686C8316FB2E412E921F58C982BE14BD]
      ,[$from_id_5A21784B3C0F48F1AEDB02E7929C5ACE]
      ,[$to_id_5EE28EC654BE410FA6FB250B3F5BBF3B]
      ,[TripID]
      ,[Passenger_Count]
      ,[Trip_Distance]
  FROM [dbo].[Trip]
```

	$edge_id_686C8316FB2E412E921F58C982BE14BD	$from_id_5A21784B3C0F48F1AEDB02E7929C5ACE	$to_id_5EE28EC654BE410FA6FB250B3F5BBF3B	TripID	Passenger_
1	{"type":"edge","schema":"dbo","table":"Trip","id":4}	{"type":"node","schema":"dbo","table":"Location","id":232}	{"type":"node","schema":"dbo","table":"Location","id":161}	0	1
2	{"type":"edge","schema":"dbo","table":"Trip","id":5}	{"type":"node","schema":"dbo","table":"Location","id":235}	{"type":"node","schema":"dbo","table":"Location","id":136}	1	1
3	{"type":"edge","schema":"dbo","table":"Trip","id":6}	{"type":"node","schema":"dbo","table":"Location","id":60}	{"type":"node","schema":"dbo","table":"Location","id":16}	2	1
4	{"type":"edge","schema":"dbo","table":"Trip","id":7}	{"type":"node","schema":"dbo","table":"Location","id":150}	{"type":"node","schema":"dbo","table":"Location","id":73}	3	1
5	{"type":"edge","schema":"dbo","table":"Trip","id":8}	{"type":"node","schema":"dbo","table":"Location","id":263}	{"type":"node","schema":"dbo","table":"Location","id":112}	4	1
6	{"type":"edge","schema":"dbo","table":"Trip","id":9}	{"type":"node","schema":"dbo","table":"Location","id":238}	{"type":"node","schema":"dbo","table":"Location","id":238}	5	1
7	{"type":"edge","schema":"dbo","table":"Trip","id":10}	{"type":"node","schema":"dbo","table":"Location","id":106}	{"type":"node","schema":"dbo","table":"Location","id":141}	6	1
8	{"type":"edge","schema":"dbo","table":"Trip","id":11}	{"type":"node","schema":"dbo","table":"Location","id":235}	{"type":"node","schema":"dbo","table":"Location","id":261}	7	1
9	{"type":"edge","schema":"dbo","table":"Trip","id":12}	{"type":"node","schema":"dbo","table":"Location","id":78}	{"type":"node","schema":"dbo","table":"Location","id":228}	8	1
1...	{"type":"edge","schema":"dbo","table":"Trip","id":13}	{"type":"node","schema":"dbo","table":"Location","id":47}	{"type":"node","schema":"dbo","table":"Location","id":238}	9	1
1...	{"type":"edge","schema":"dbo","table":"Trip","id":14}	{"type":"node","schema":"dbo","table":"Location","id":78}	{"type":"node","schema":"dbo","table":"Location","id":124}	10	1
1...	{"type":"edge","schema":"dbo","table":"Trip","id":15}	{"type":"node","schema":"dbo","table":"Location","id":224}	{"type":"node","schema":"dbo","table":"Location","id":248}	11	1
1...	{"type":"edge","schema":"dbo","table":"Trip","id":16}	{"type":"node","schema":"dbo","table":"Location","id":238}	{"type":"node","schema":"dbo","table":"Location","id":229}	12	1
1...	{"type":"edge","schema":"dbo","table":"Trip","id":17}	{"type":"node","schema":"dbo","table":"Location","id":263}	{"type":"node","schema":"dbo","table":"Location","id":263}	13	0
1...	{"type":"edge","schema":"dbo","table":"Trip","id":18}	{"type":"node","schema":"dbo","table":"Location","id":237}	{"type":"node","schema":"dbo","table":"Location","id":237}	14	2