P4 Work with Database

Project Team 5

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)

Topic: Urban Mobility Data Management System

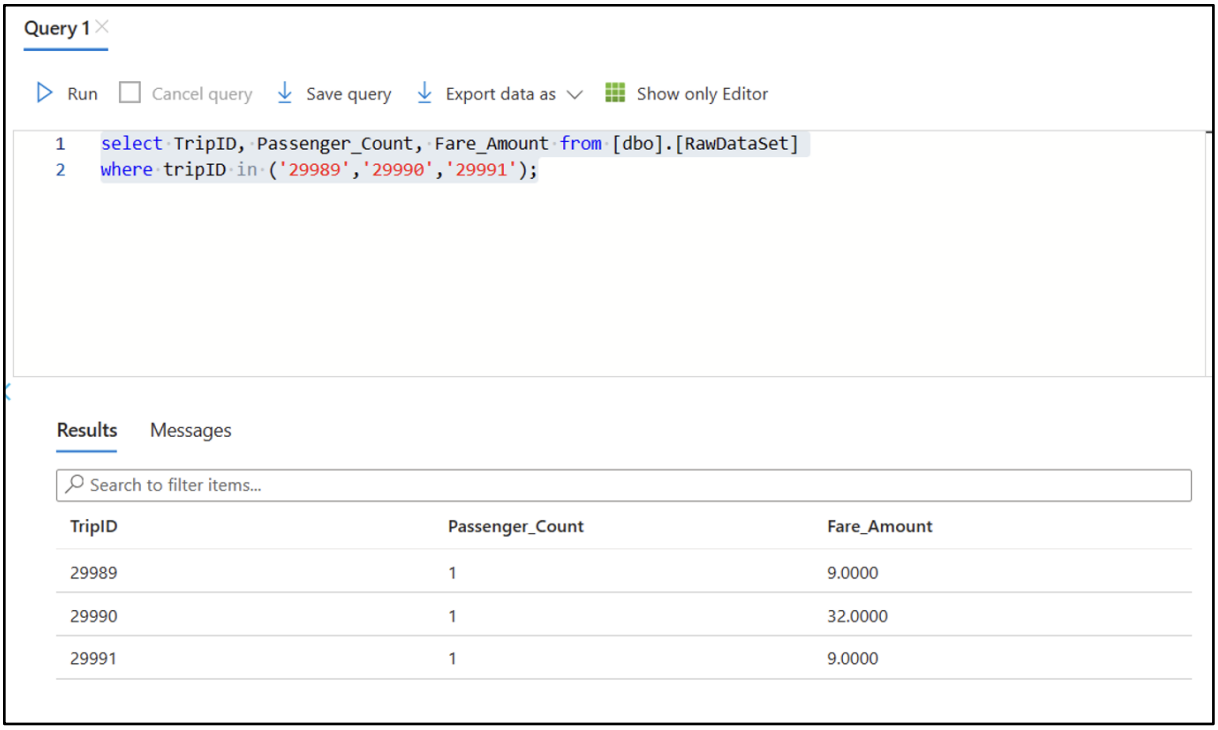
Step 1: Manually update the CSV file for selected rows and add new rows in the csv file in our Blob storage:
We have chosen the below TripIDs for UPDATE operation and changed their Passenger_Count and Fare_Amount values as highlighed:

29989
29990
29991

We have newly added the below rows with the highlighted TripIDs:

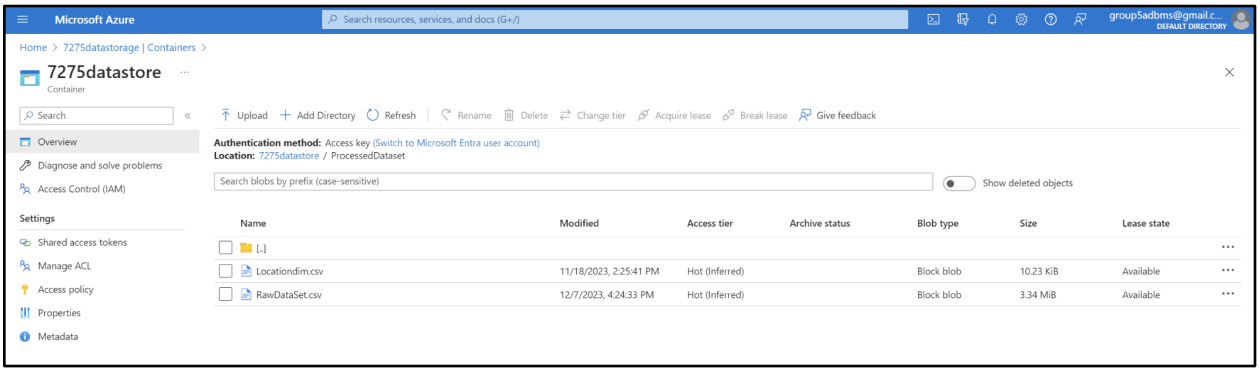
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Serviceprovider	Pickup_Datetime	Dropoff_Datetime	Passenger_Count	Trip_Distance	Ratecodeid	Store_And_Fwd_Flag	Pickup_Location	Droppff_Location	Payment_Type	Fare_Amount	Extra	Mta_Tax	Tip_Amount	Tolls_Amount
29980	29978	Uber	2022-03-29T13:00:00	2022-03-29T13:11:00	1	0.9	1	N	239	90	1	9	1	0.5	2.06	0
29981	29979	Uber	2022-03-03T16:06:00	2022-03-03T16:13:00	0	0.8	1	N	113	148	1	6.5	2.5	0.5	1.96	0
29982	29980	Uber	2022-03-26T00:27:00	2022-03-26T00:41:00	2	1.82	1	N	170	246	1	10.5	0.5	0.5	2.86	0
29983	29981	Uber	2022-11-16T09:43:00	2022-11-16T10:00:00	1	3.6	1	N	100	13	1	14	0	0.5	3.46	0
29984	29982	Juno	2022-03-04T21:57:00	2022-03-04T22:02:00	1	0.7	1	N	142	143	1	5	0.5	0.5	0.88	0
29985	29983	Uber	2022-06-27T10:49:00	2022-06-27T10:55:00	1	0.49	1	N	170	170	1	5.5	0	0.5	1	0
29986	29984	Lyft	2022-12-10T20:31:00	2022-12-10T20:44:00	1	1.8	1	N	229	236	2	9	3	0.5	0	0
29987	29985	Uber	2022-10-26T13:58:00	2022-10-26T14:09:00	1	1.31	1	N	163	236	1	8.5	0	0.5	2.36	0
29988	29986	Uber	2022-09-17T12:51:00	2022-09-17T13:13:00	2	6.5	1	N	263	232	1	22	2.5	0.5	5.05	0
29989	29987	Lyft	2022-04-11T08:21:00	2022-04-11T09:32:00	1	17.1	2	N	132	230	1	52	3.75	0.5	12.6	6.55
29990	29988	Lyft	2022-09-12T20:40:00	2022-09-12T20:47:00	2	1.48	1	N	158	234	1	7	0.5	0.5	1.5	0
29991	29989	Uber	2022-02-28T18:39:00	2022-02-28T18:50:00	3	1.7	1	N	236	237	1	12	3.5	0.5	2.65	0
29992	29990	Uber	2022-12-10T22:03:00	2022-12-10T22:40:00	3	9.56	1	N	142	97	2	33	0.5	0.5	0	0
29993	29991	Uber	2022-03-14T14:24:00	2022-03-14T14:35:00	3	1.41	1	N	186	163	2	19	0	0.5	0	0
29994	29992	Lyft	2020-03-10T14:24:00	2020-03-10T14:29:00	0	0.1	1	N	145	22	2	14	0	0.5	3.46	0
29995	29993	Lyft	2021-08-12T14:24:00	2021-08-12T14:44:00	1	1.26	2	N	123	43	1	5	0.5	0.5	0.88	0
29996	29994	Lyft	2022-05-19T14:24:00	2022-05-19T14:32:00	2	1.8	1	N	122	52	1	5.5	0	0.5	1	0
29997	29995	Juno	2020-03-19T10:29:00	2020-03-19T11:02:00	1	1.42	1	N	142	237	1	10	0	0.5	2.66	0
29998	29996	Juno	2021-06-06T17:10:00	2021-06-06T17:23:00	1	0.6	1	N	246	68	1	8.5	3.5	0.5	2.55	0
29999	29997	Via	2022-09-30T14:46:00	2022-09-30T14:58:00	1	2.37	1	N	87	66	2	10	0	0.5	0	0
30000	29998	Via	2022-05-11T15:24:00	2022-05-11T15:27:00	1	0.49	1	N	238	239	1	4.5	0	0.5	1.56	0
30001	29999	Via	2022-12-15T21:23:00	2022-12-15T21:51:00	1	3.2	1	N	79	163	1	18.5	3	0.5	4.45	0
30002	30000	Via	2022-12-23T00:07:00	2022-12-23T00:15:00	1	1.16	1	N	211	79	1	9.3	1	0.5	1	0
30003																
30004																
30005																
30006																

Before the Data Refresh:

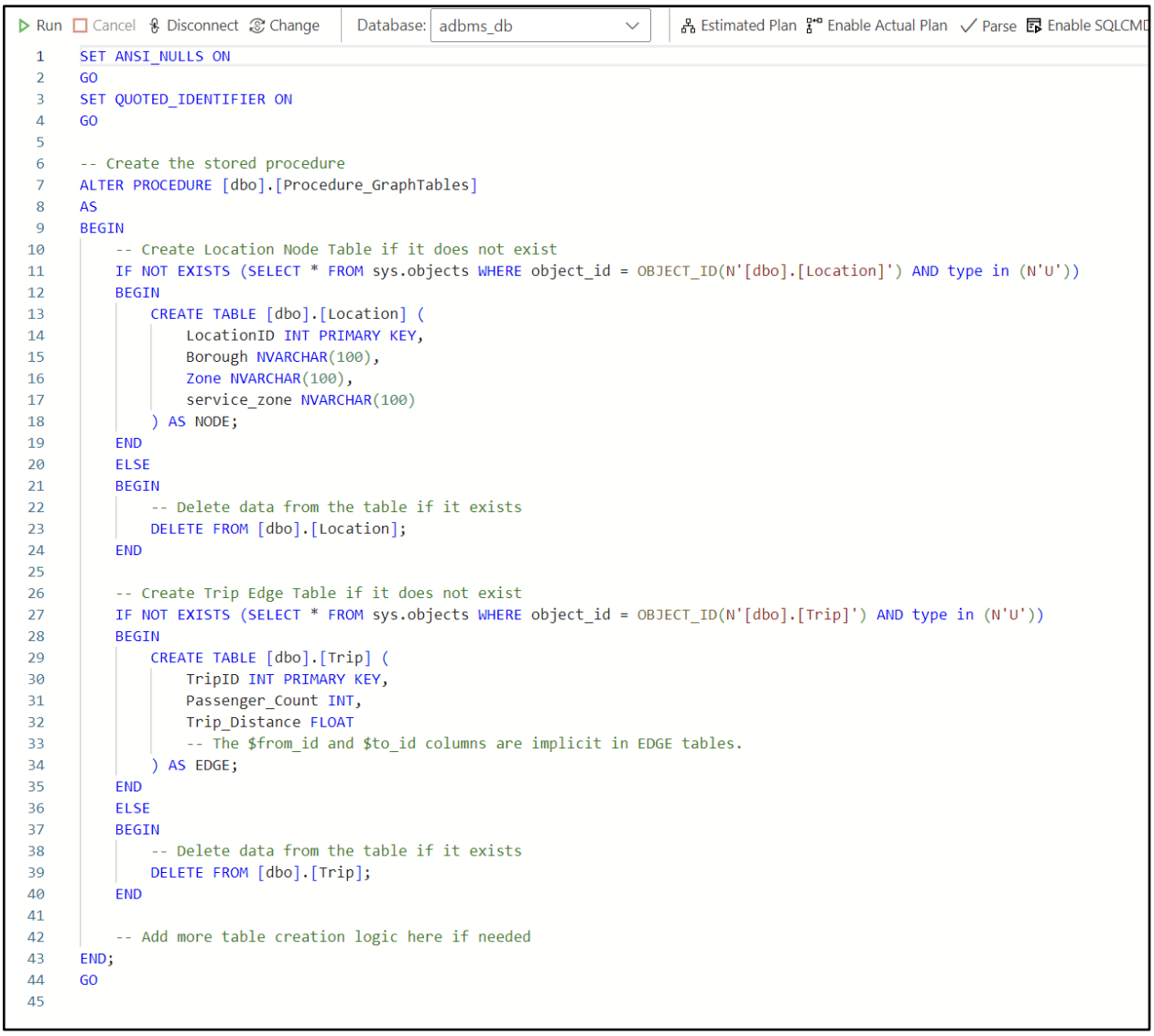


Step 2: The new CSV file is manually changed in the Blob Storage:

It's essential to handle the deletion of the old file in the blob storage before passing the new file. However, since there is no primary key in the source file and changes are made manually for data refresh execution, it becomes challenging to track changes added before pre-processing, hence we directly upload the new file to the blob storage.



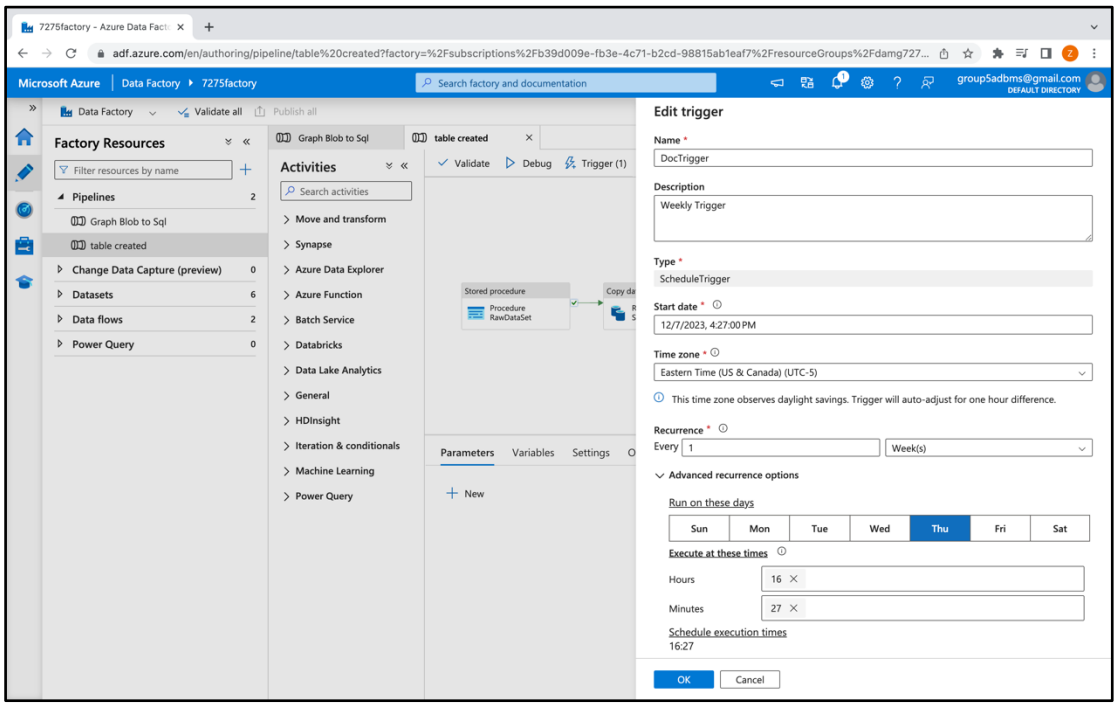
Prerequisite for Step 3: The pipelines should be designed to empty the tables before new data is inserted.



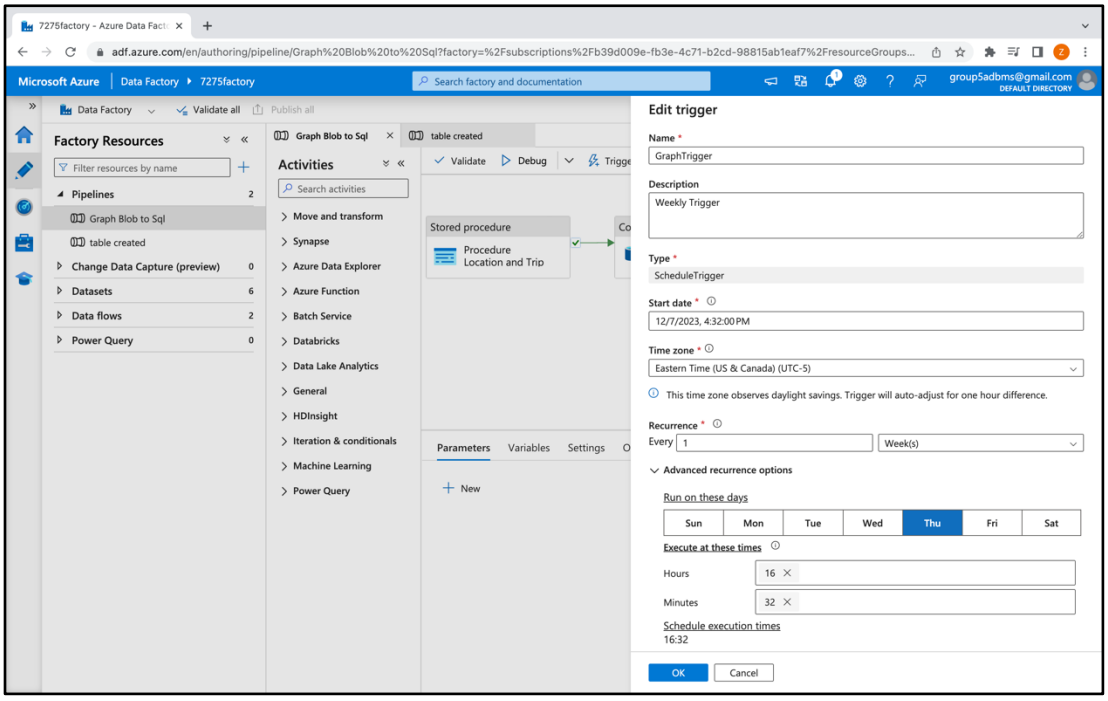

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- Create the stored procedure
ALTER 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
    BEGIN
        -- Delete data from the table if it exists
        DELETE FROM [dbo].[Location];
    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
            -- The $from_id and $to_id columns are implicit in EDGE tables.
        ) AS EDGE;
    END
    ELSE
    BEGIN
        -- Delete data from the table if it exists
        DELETE FROM [dbo].[Trip];
    END

    -- Add more table creation logic here if needed
END;
GO
```

Step 3: Now we schedule our two triggers to run the (document + relational) and (graph) pipelines, respectively, one after the other at intervals of 5 minutes on every Thursday on weekly basis at 04:27 PM and 04:32 PM.

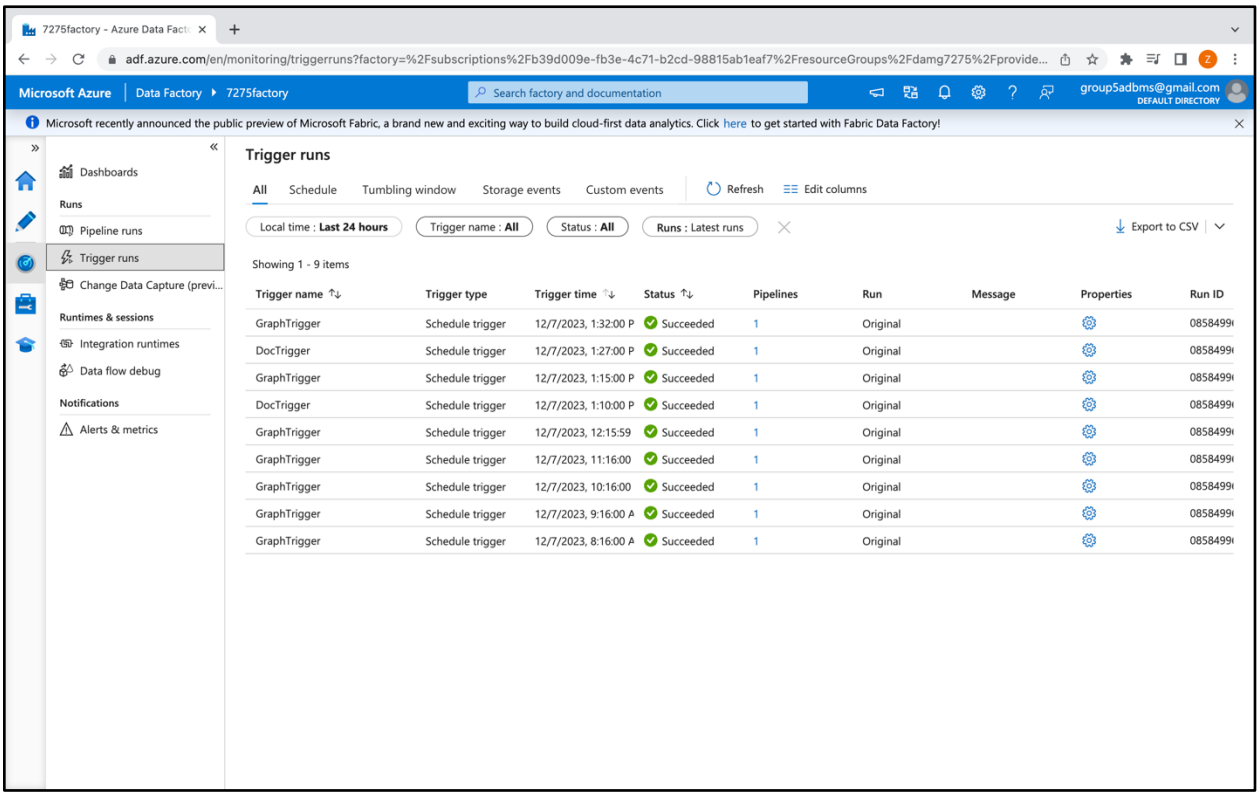
Trigger for Document and Relational Model Data Pipeline:



Trigger for Graph Model Data Pipeline:



Triggers are running successfully:



Step 4: After the successful completion of the pipelines, we check the output for the 'After' state of the updated rows and new rows:

Rows of tripID 29989, 29990, and 29991 have been successfully updated

SQLQuery_1 - adbmsg...(adbms)

Run | Cancel | Disconnect | Change | Database: adbms_db | Estimate

```
1  select TripID, Passenger_Count, Fare_Amount from [dbo].[RawDataSet]
2  where tripID in ('29989','29990','29991');
3
```

Results | Messages

	TripID	Passenger_Count	Fare_Amount
1	29989	3	12.0000
2	29990	3	33.0000
3	29991	3	19.0000

New rows are added in the output Table:

Run | Cancel | Disconnect | Change | Database: adbms_db | Estimated Plan | Enable Actual Plan | Parse | Enable SQLCMD | To Notebook

```
1  Select * from RawDataSet where TripID in ('29992','29993','29994','29995','29996','29997','29998','30000')
```

Results | Messages

	TripID	DateTimeID	TransactionID	RatecodeID	Serviceprovider	Pickup_Datetime	Dropoff_Datetime	Passenger_Count	Trip_Distance	Store_And_Fwd_Flag
1	29992	29992	29992	1	Lyft	2020-03-10T14:24:00	2020-03-10T14:29:00	0	0.1000	N
2	29993	29993	29993	2	Lyft	2021-08-12T14:24:00	2021-08-12T14:44:00	1	1.2600	N
3	29994	29994	29994	1	Lyft	2022-05-19T14:24:00	2022-05-19T14:32:00	2	1.8000	N
4	29995	29995	29995	1	Juno	2020-03-19T10:29:00	2020-03-19T11:02:00	1	1.4200	N
5	29996	29996	29996	1	Juno	2021-06-06T17:10:00	2021-06-06T17:23:00	1	0.6000	N
6	29997	29997	29997	1	Via	2022-09-30T14:46:00	2022-09-30T14:58:00	1	2.3700	N
7	29998	29998	29998	1	Via	2022-05-11T15:24:00	2022-05-11T15:27:00	1	0.4900	N
8	30000	30000	30000	1	Via	2022-12-23T00:07:00	2022-12-23T00:15:00	1	1.1600	N

Since our database is static, the above steps outline our approach to achieving data refresh.