Project: Building an Azure Data Warehouse for Bike Share Data Analytics

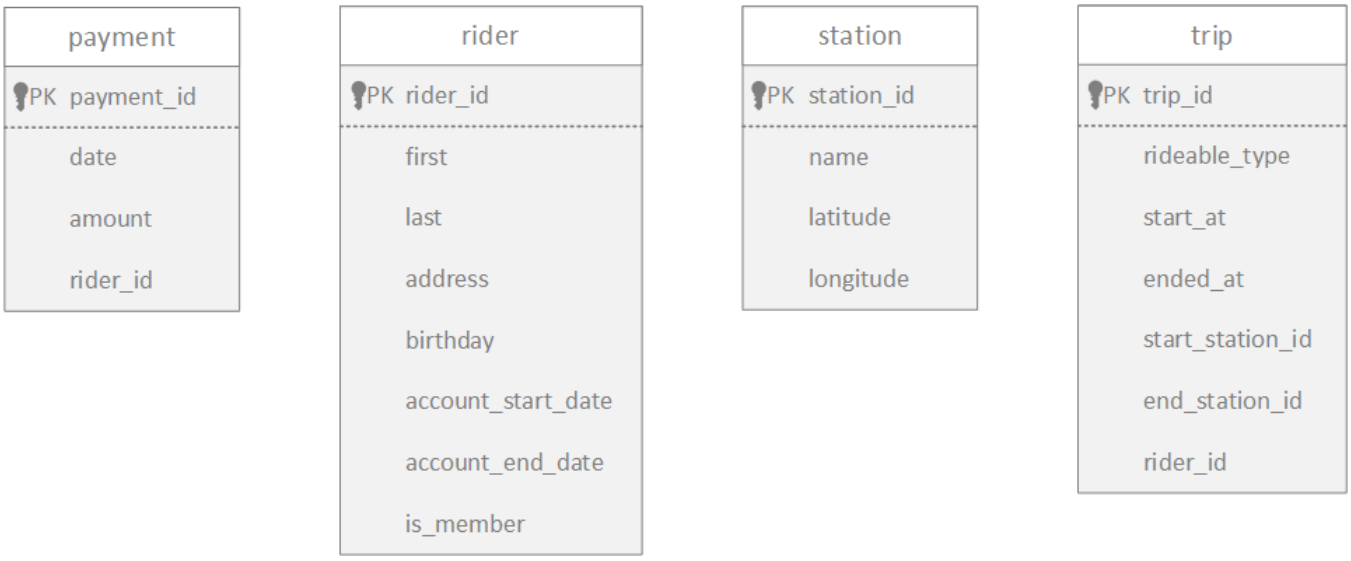
Jun 9 2024

Dahi Nemutlu

Contents

1. Project Overview

Divvy is a bike sharing program in Chicago, Illinois USA that allows riders to purchase a pass at a kiosk or use a mobile application to unlock a bike at stations around the city and use the bike for a specified amount of time. The bikes can be returned to the same station or to another station. The City of Chicago makes the anonymized bike trip data publicly available for projects like this where we can analyze the data. The dataset looks like this:



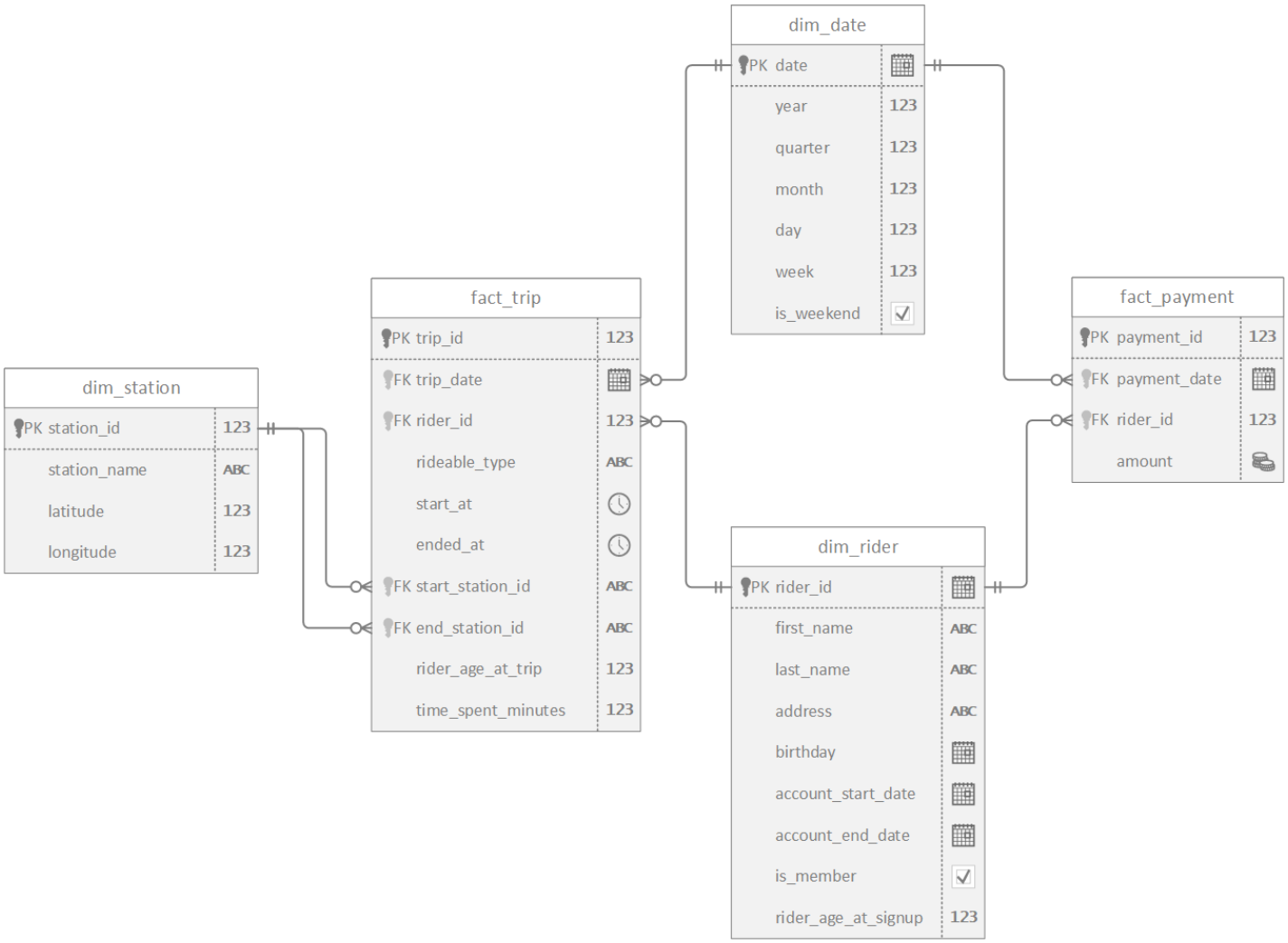
The goal of this project is to develop a data warehouse solution using Azure Synapse Analytics. We will:

- Design a star schema based on the business outcomes listed below;
- Import the data into Synapse;
- Transform the data into the star schema;
- and finally, view the reports using Power BI.

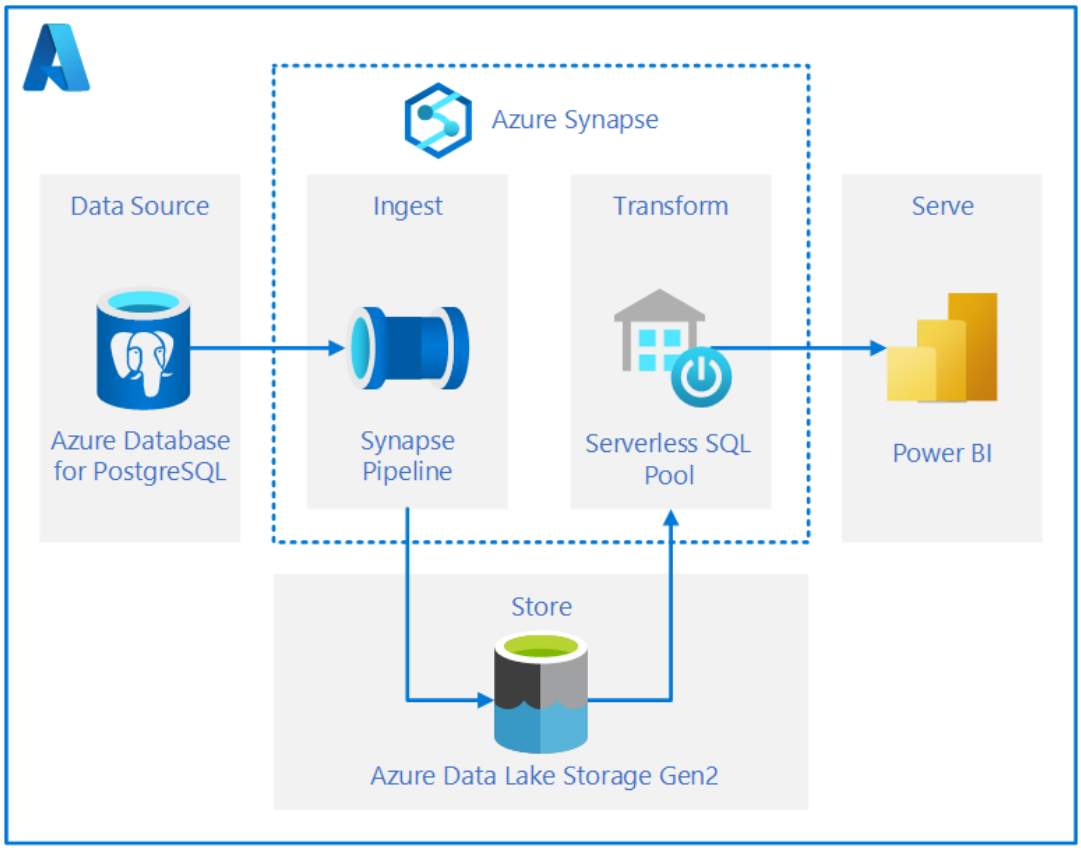
The business outcomes we are designing for are as follows:

1. Analyze how much time is spent per ride
    - Based on date and time factors such as day of week and time of day
    - Based on which station is the starting and / or ending station
    - Based on age of the rider at time of the ride
    - Based on whether the rider is a member or a casual rider
2. Analyze how much money is spent
    - Per month, quarter, year
    - Per member, based on the age of the rider at account start

Based on the provided business requirements for the data warehouse, we will create the following star schema using fact and dimension tables.

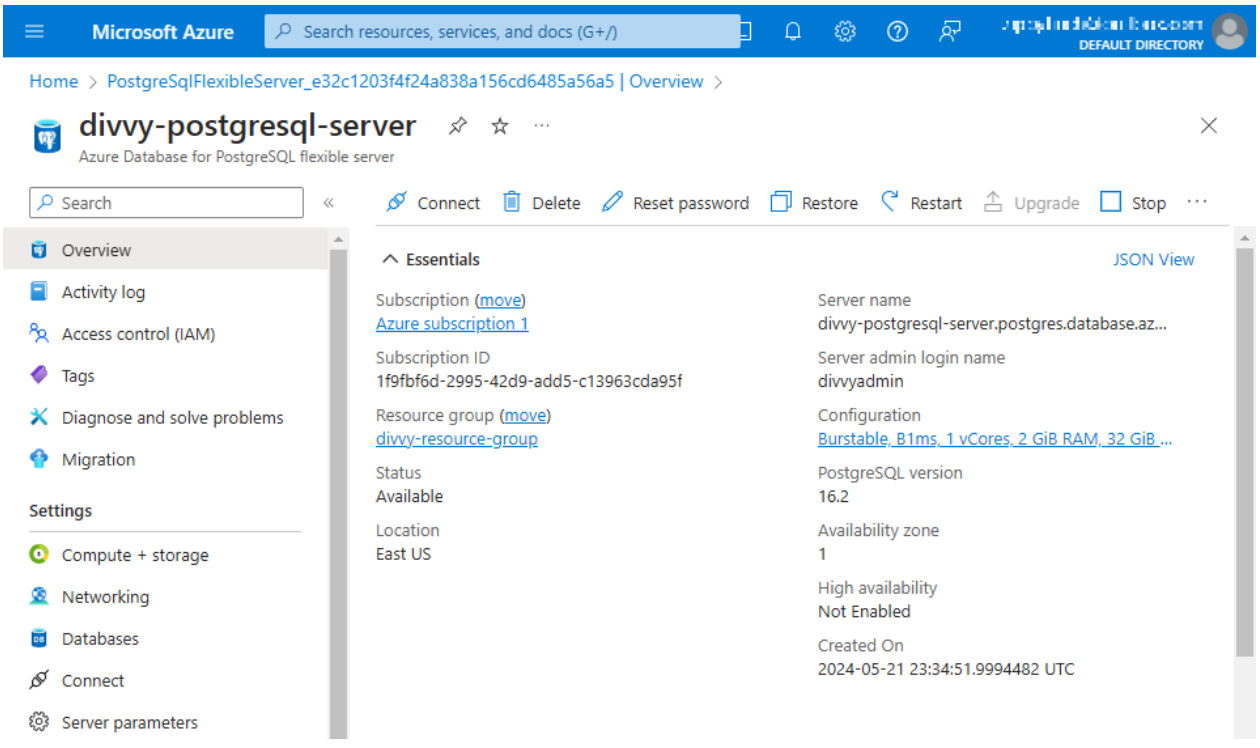


With this project, we will create an end-to-end Azure data warehousing solution, with the architecture as follows:

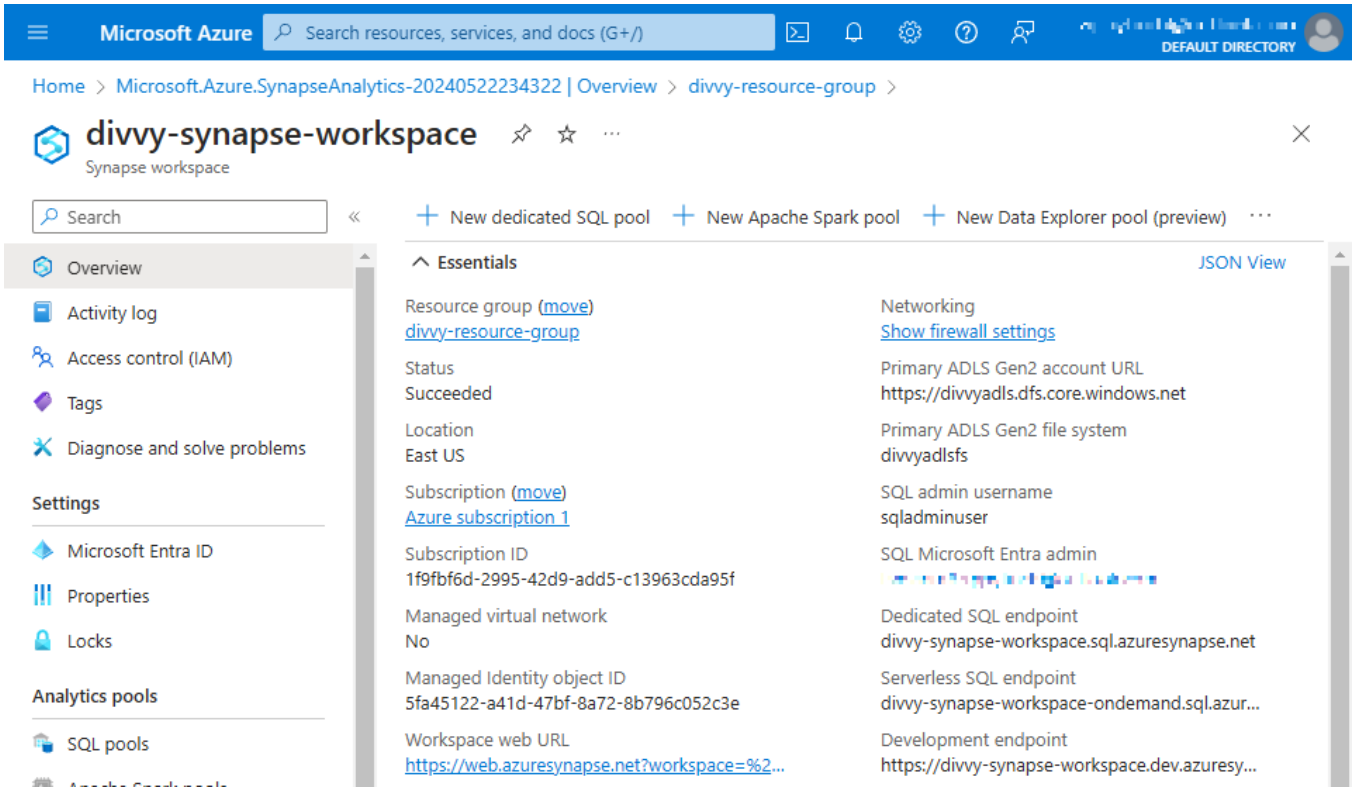


2. Create Azure Resources

2.1. Create Azure Database for PostgreSQL



2.2. Create Azure Synapse Workspace



3. Create the data in PostgreSQL

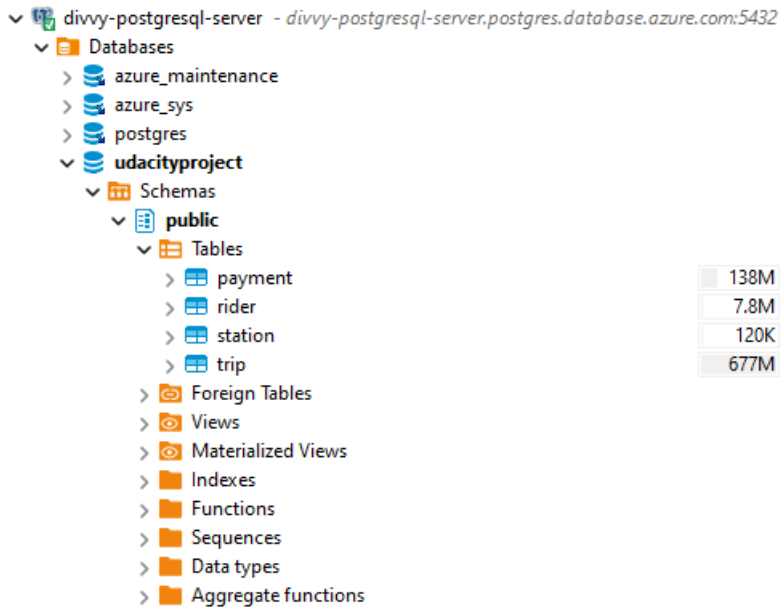
To prepare our environment for this project, we first must create the data in PostgreSQL. This will simulate the production environment where the data is being used in the OLTP system. This can be done using the Python script provided: *ProjectDataToPostgres.py*

Note: Add host, username, and password information for your PostgreSQL database to the script before running it.

```
   PS C:\Users\dnemu\Desktop\building-an-azure-data-warehouse-for-bike-share-data-
analytics> & C:/Users/dnemu/AppData/Local/Microsoft/WindowsApps/python3.11.exe
c:/Users/dnemu/Desktop/building-an-azure-data-warehouse-for-bike-share-data-
analytics/ProjectDataToPostgres.py
   Connection established
   Connection established
   Finished creating table rider
   Finished populating rider
   Finished creating table payment
   Finished populating payment
   Finished creating table station
   Finished populating station
```

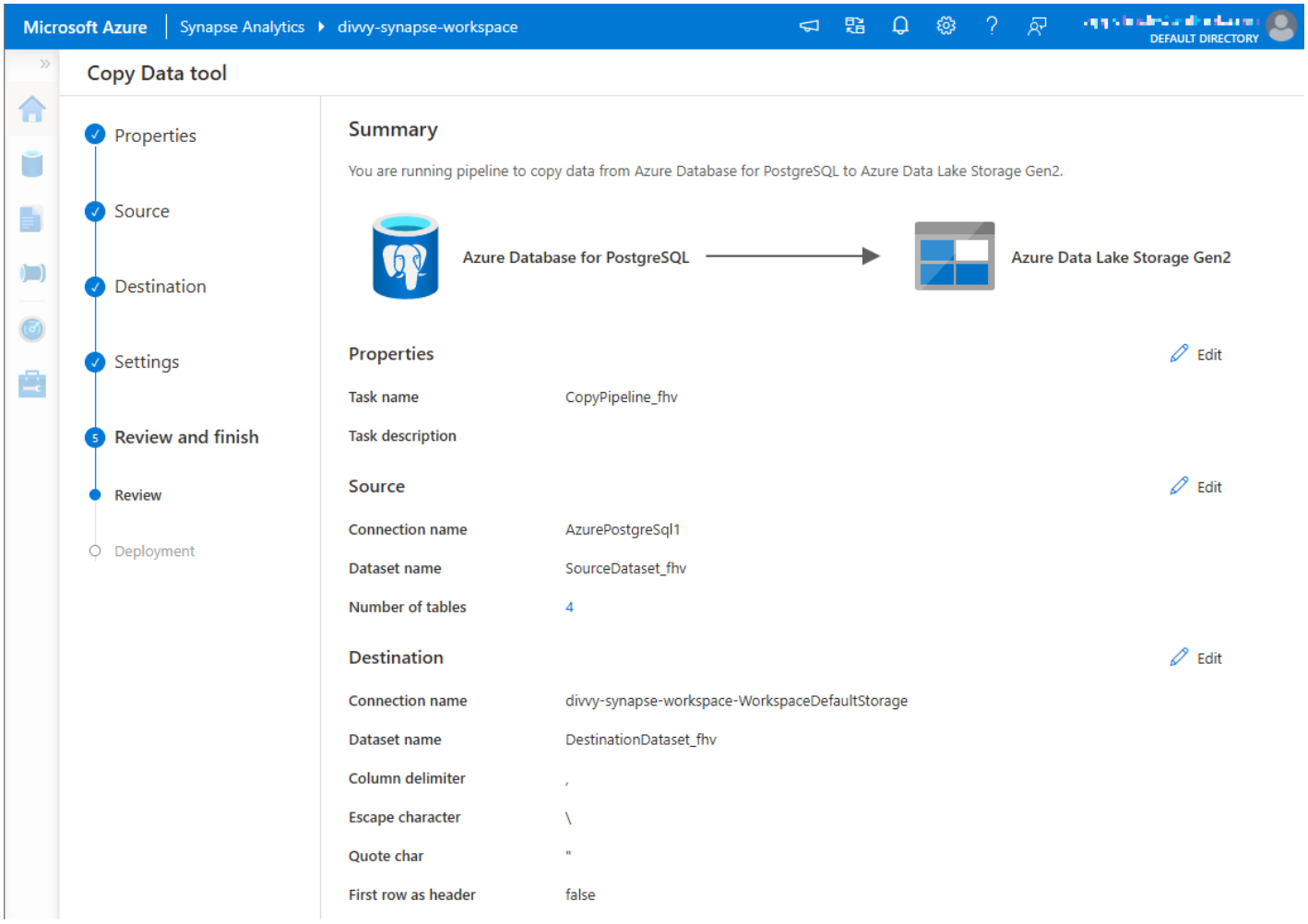
```
Finished creating table trip
Finished populating trip
All done!
```

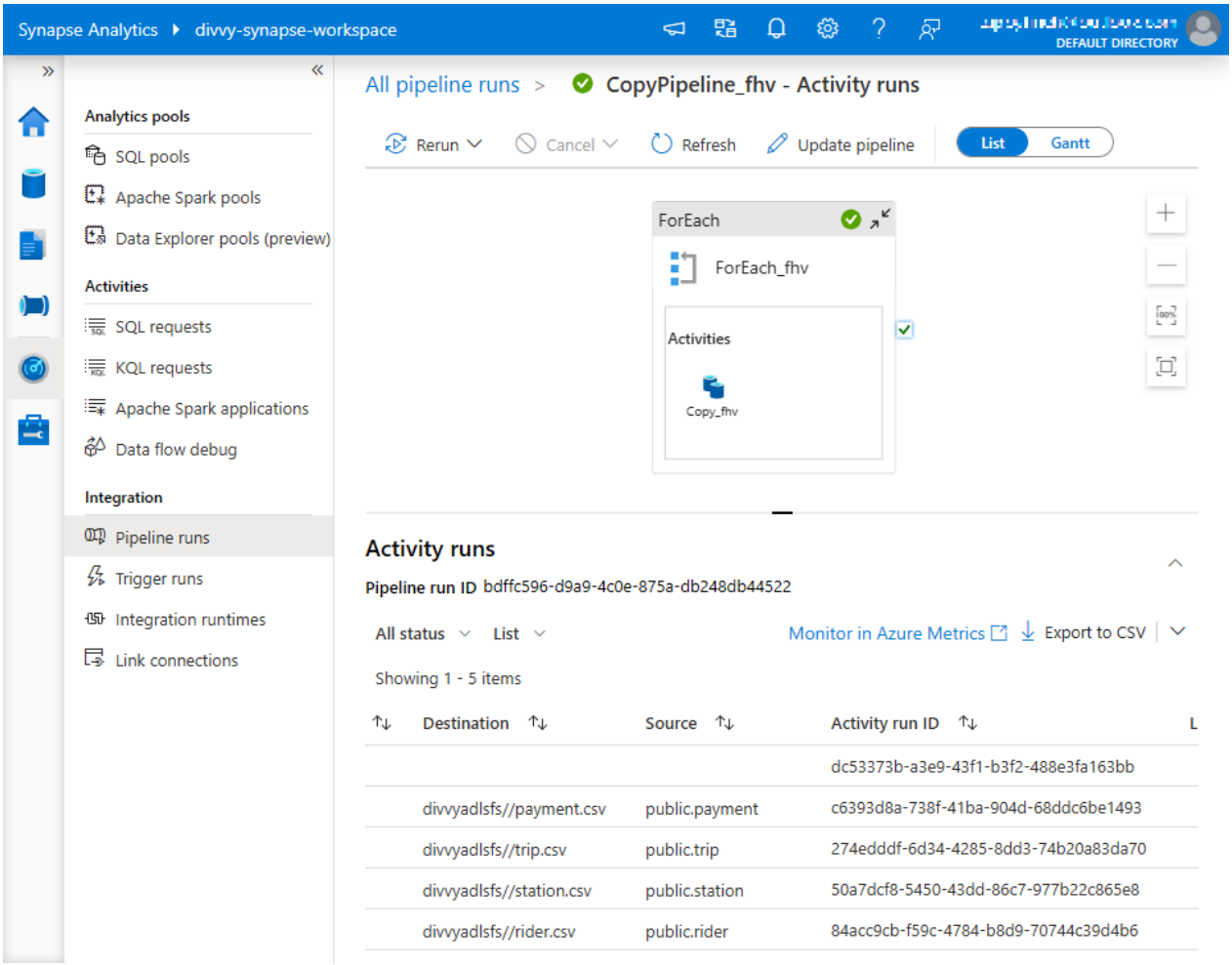
Then verify that the data exists using pgAdmin or a similar PostgreSQL data tool.

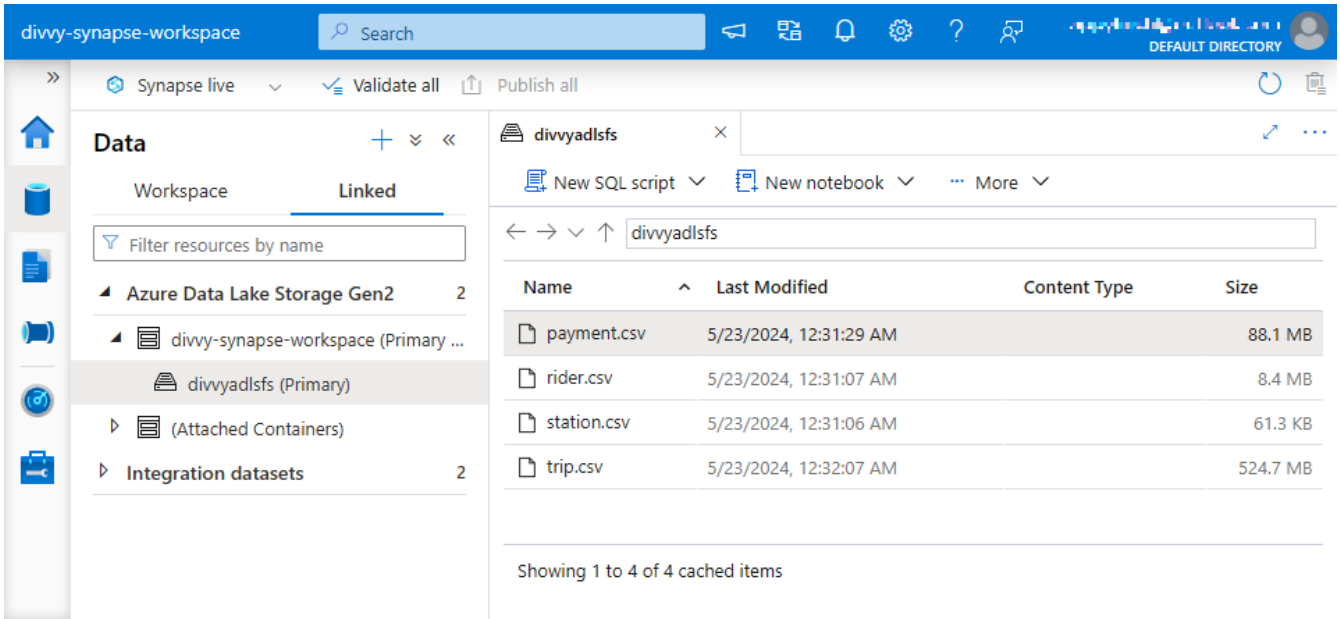


4. Extract the data from PostgreSQL

In our Azure Synapse workspace, we will use the ingest wizard to create a one-time pipeline that ingests the data from PostgreSQL into Azure Data Lake Gen 2 storage. This will result in all four tables being represented as text files in the storage, ready for loading into the data warehouse.

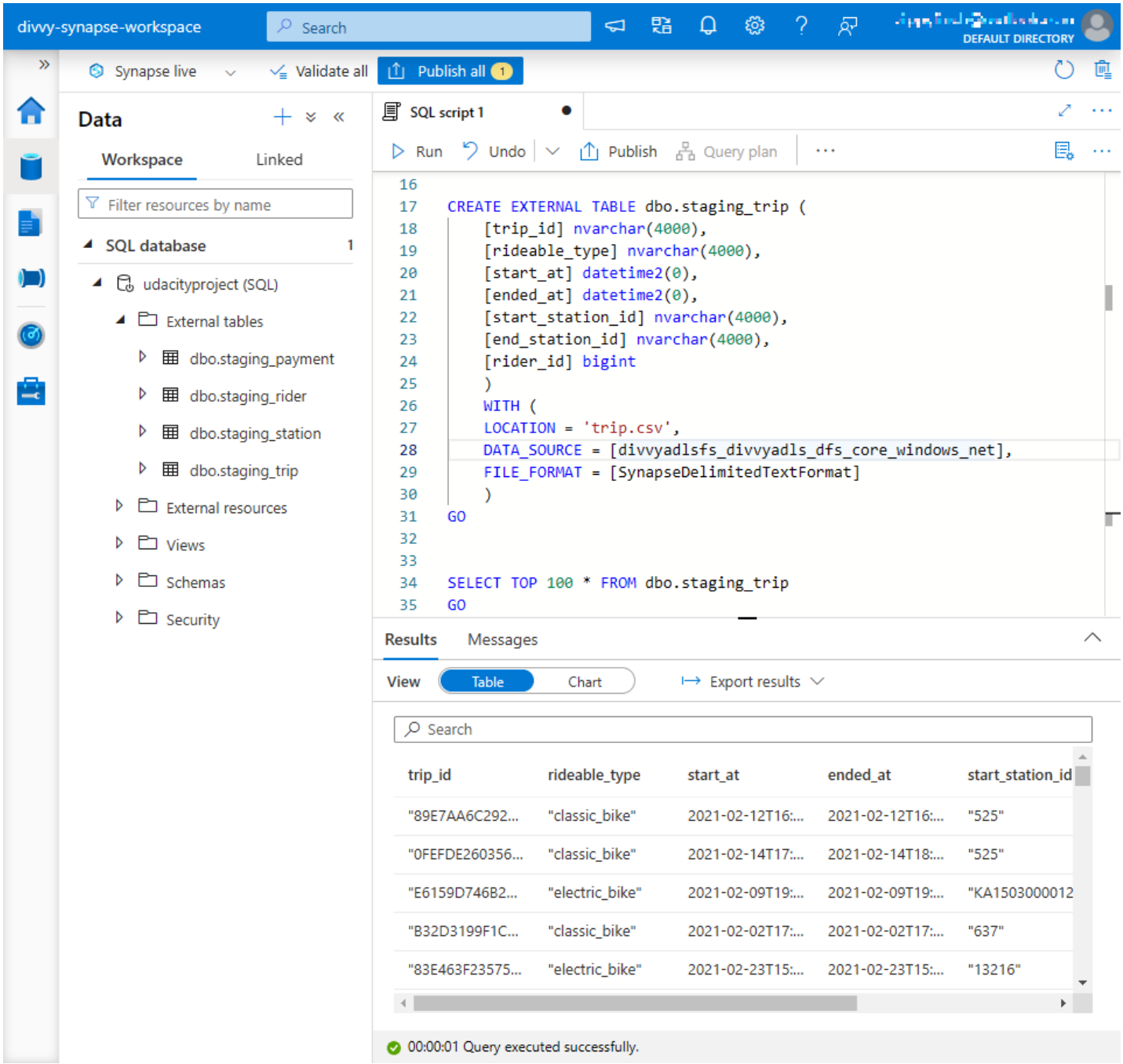

Microsoft Azure
Synapse Analytics
divvy-synapse-workspace
DEFAULT DIRECTORY
Copy Data tool
Properties
Source
Destination
Settings
Review and finish
Review
Deployment
Summary
You are running pipeline to copy data from Azure Database for PostgreSQL to Azure Data Lake Storage Gen2.
Azure Database for PostgreSQL
Azure Data Lake Storage Gen2
Properties
Edit
Task name
CopyPipeline_fhv
Task description
Source
Edit
Connection name
AzurePostgreSql1
Dataset name
SourceDataset_fhv
Number of tables
4
Destination
Edit
Connection name
divvy-synapse-workspace-WorkspaceDefaultStorage
Dataset name
DestinationDataset_fhv
Column delimiter
,
Escape character
\
Quote char
"
First row as header
false


Synapse Analytics
divvy-synapse-workspace
DEFAULT DIRECTORY
Analytics pools
SQL pools
Apache Spark pools
Data Explorer pools (preview)
Activities
SQL requests
KQL requests
Apache Spark applications
Data flow debug
Integration
Pipeline runs
Trigger runs
Integration runtimes
Link connections
All pipeline runs > CopyPipeline_fhv - Activity runs
Rerun
Cancel
Refresh
Update pipeline
List
Gantt
ForEach
ForEach_fhv
Activities
Copy_fhv
Activity runs
Pipeline run ID bdffc596-d9a9-4c0e-875a-db248db44522
All status
List
Monitor in Azure Metrics
Export to CSV
Showing 1 - 5 items
Destination
Source
Activity run ID
dc53373b-a3e9-43f1-b3f2-488e3fa163bb
divvyadlsfs//payment.csv
public.payment
c6393d8a-738f-41ba-904d-68ddc6be1493
divvyadlsfs//trip.csv
public.trip
274edddf-6d34-4285-8dd3-74b20a83da70
divvyadlsfs//station.csv
public.station
50a7dcf8-5450-43dd-86c7-977b22c865e8
divvyadlsfs//rider.csv
public.rider
84acc9cb-f59c-4784-b8d9-70744c39d4b6


divvy-synapse-workspace
Search
DEFAULT DIRECTORY
Synapse live
Validate all
Publish all
Data
Workspace
Linked
Filter resources by name
Azure Data Lake Storage Gen2
2
divvy-synapse-workspace (Primary ...
divvyadlsfs (Primary)
(Attached Containers)
Integration datasets
2
divvyadlsfs
New SQL script
New notebook
More
divvyadlsfs
Name
Last Modified
Content Type
Size
payment.csv
5/23/2024, 12:31:29 AM
88.1 MB
rider.csv
5/23/2024, 12:31:07 AM
8.4 MB
station.csv
5/23/2024, 12:31:06 AM
61.3 KB
trip.csv
5/23/2024, 12:32:07 AM
524.7 MB
Showing 1 to 4 of 4 cached items

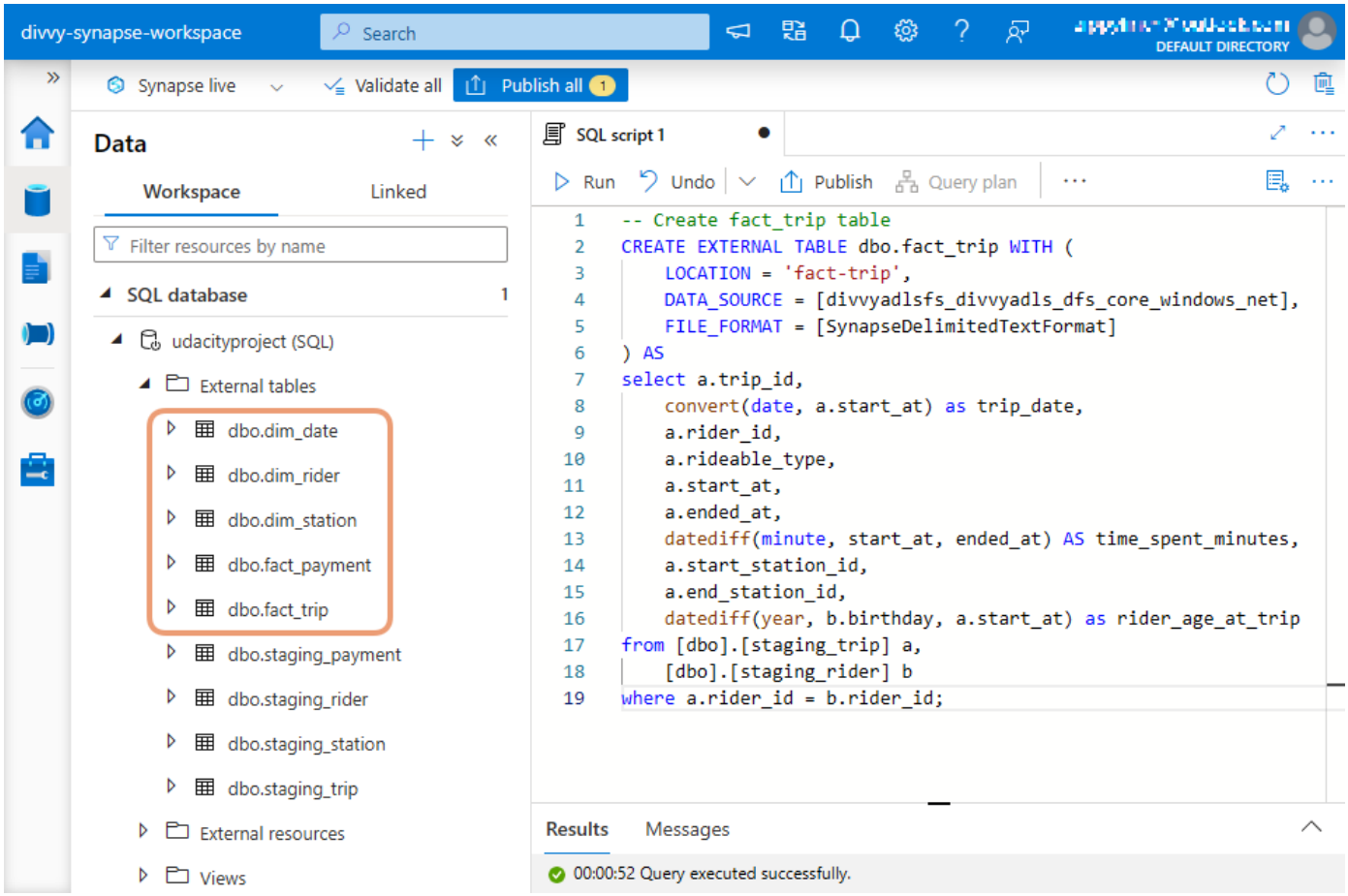
5. Load the data into external tables in the data warehouse

Once in Data Lake storage, the files will be shown in the data lake node in the Synapse Workspace. From here, we can use the script-generating function to load the data into external staging tables in the data warehouse you created using the serverless SQL Pool.



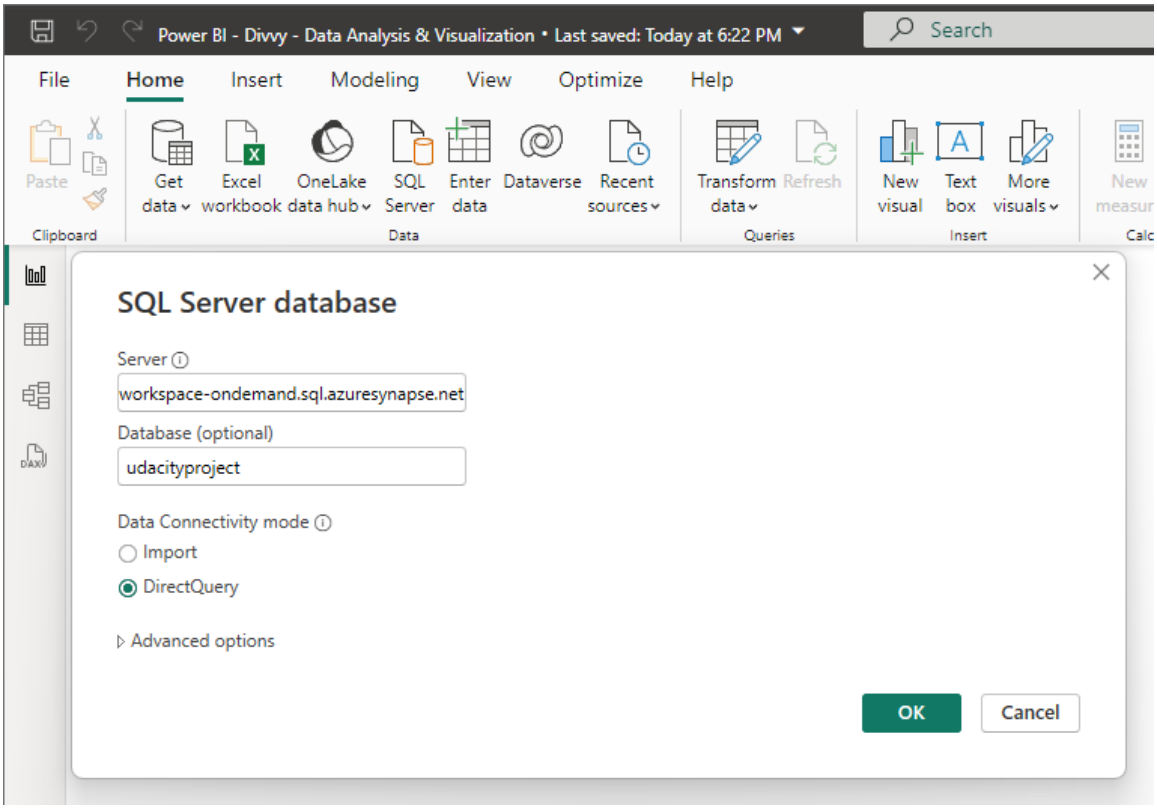
6. Transform the data to the star schema

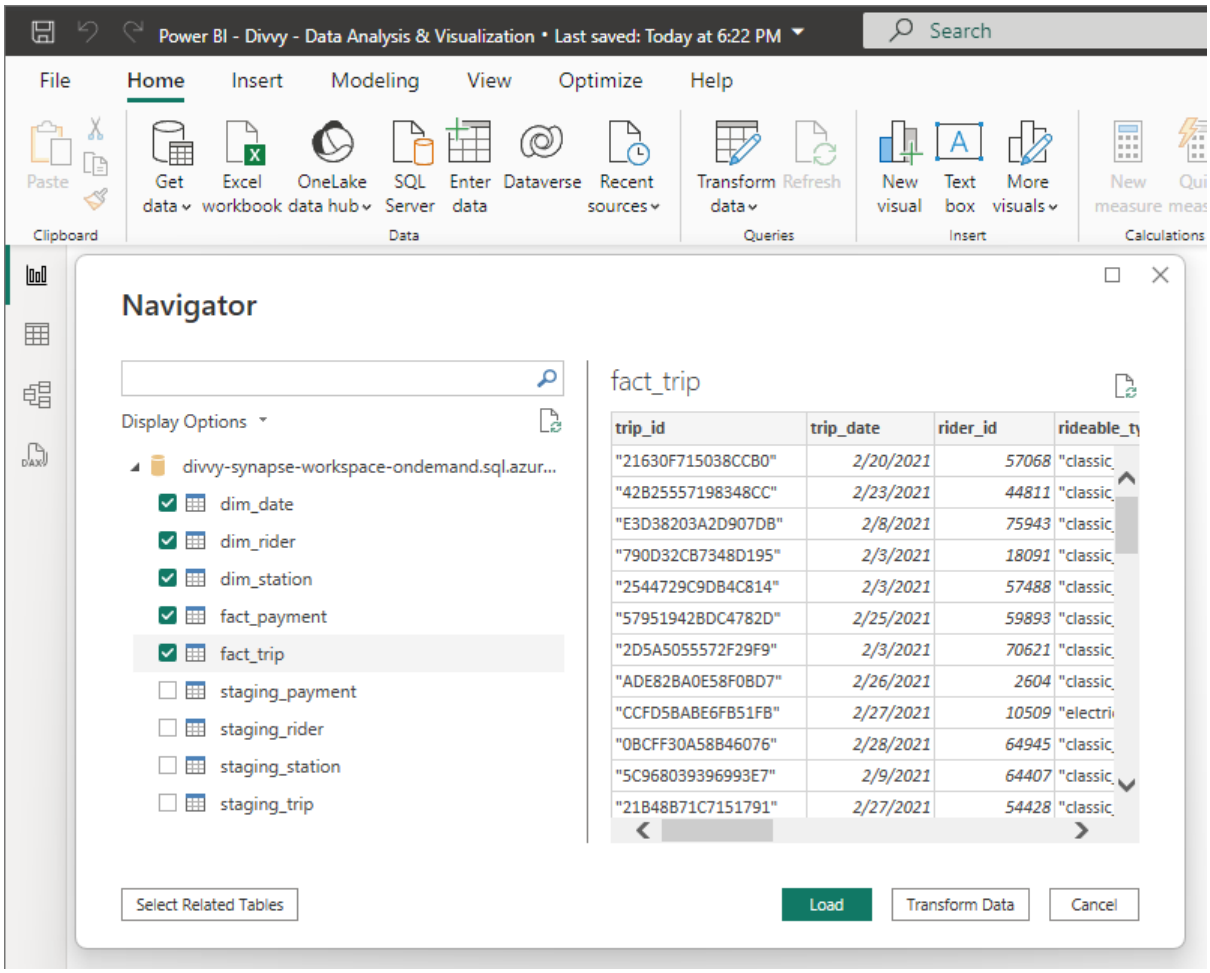
Now we will run SQL scripts to transform the data from the staging tables to the final star schema that we designed. The serverless SQL pool won't allow us to create persistent tables in the database, as it has no local storage. So, we will use CREATE EXTERNAL TABLE AS SELECT (CETAS) instead.



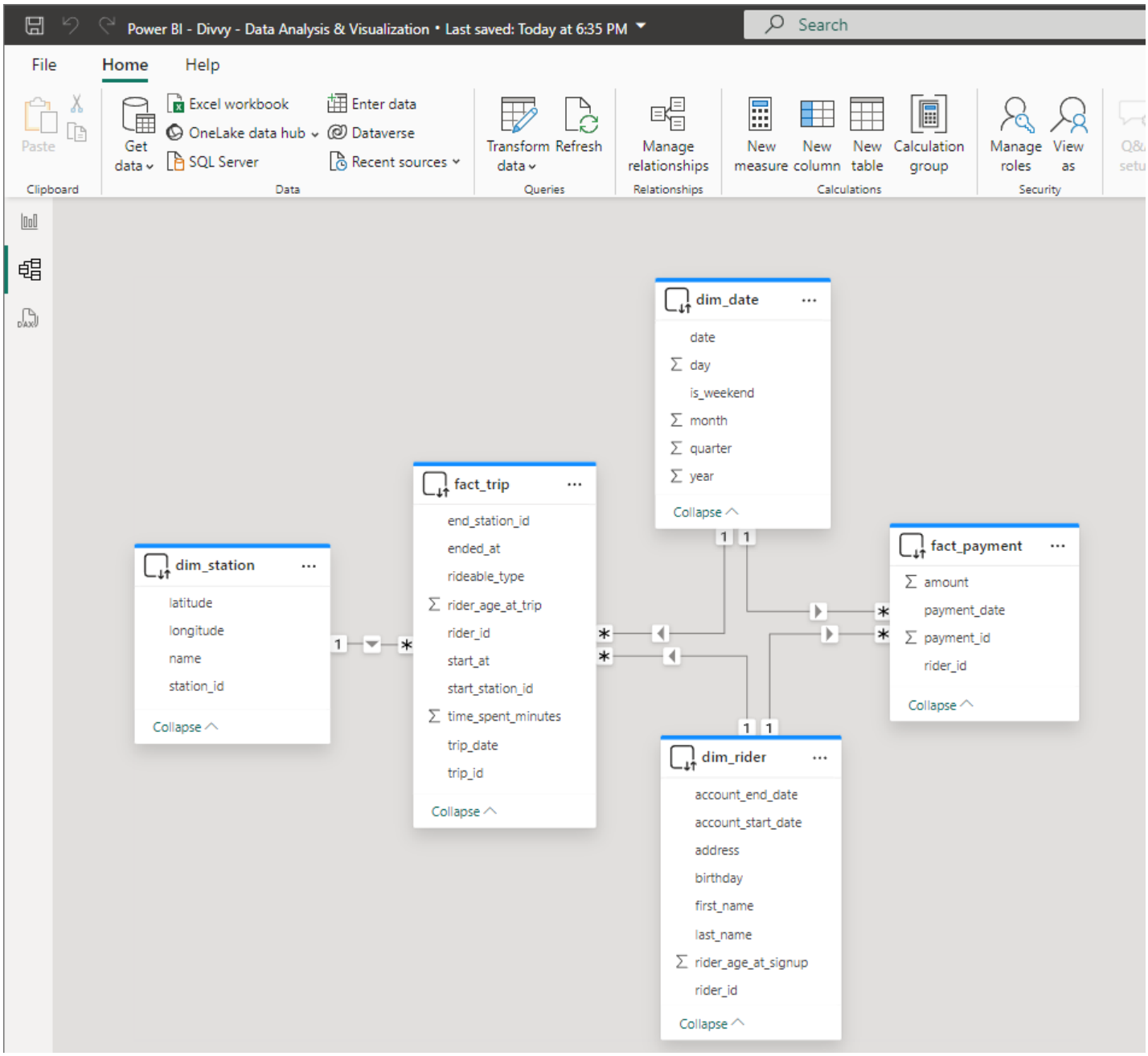
7. Analyze and visualize data with Power BI

Connect to the SQL database using Serverless SQL endpoint and import star schema tables into Power BI.

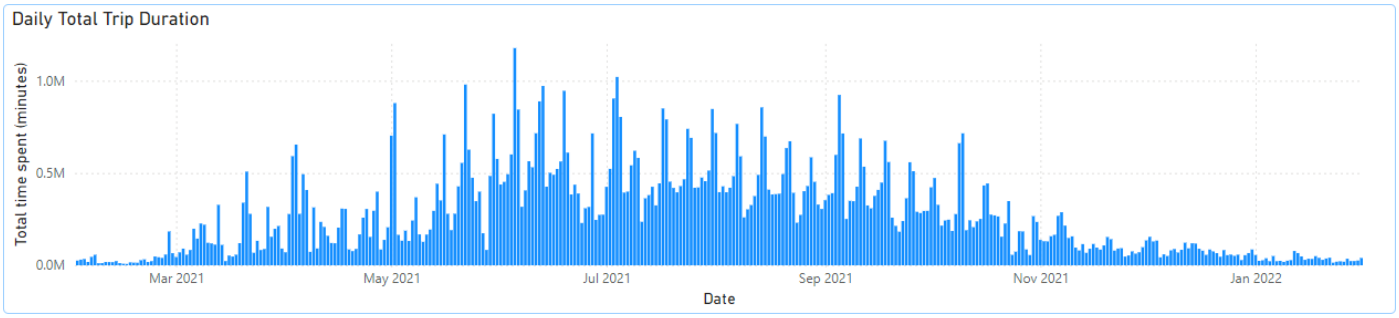

Power BI - Divvy - Data Analysis & Visualization • Last saved: Today at 6:22 PM
Search
File
Home
Insert
Modeling
View
Optimize
Help
Paste
Get data
Excel workbook
OneLake data hub
SQL Server
Enter data
Dataverse
Recent sources
Transform data
Refresh
New visual
Text box
More visuals
Clipboard
Data
Queries
Insert
SQL Server database
Server
workspace-ondemand.sql.azuresynapse.net
Database (optional)
udacityproject
Data Connectivity mode
Import
DirectQuery
Advanced options
OK
Cancel

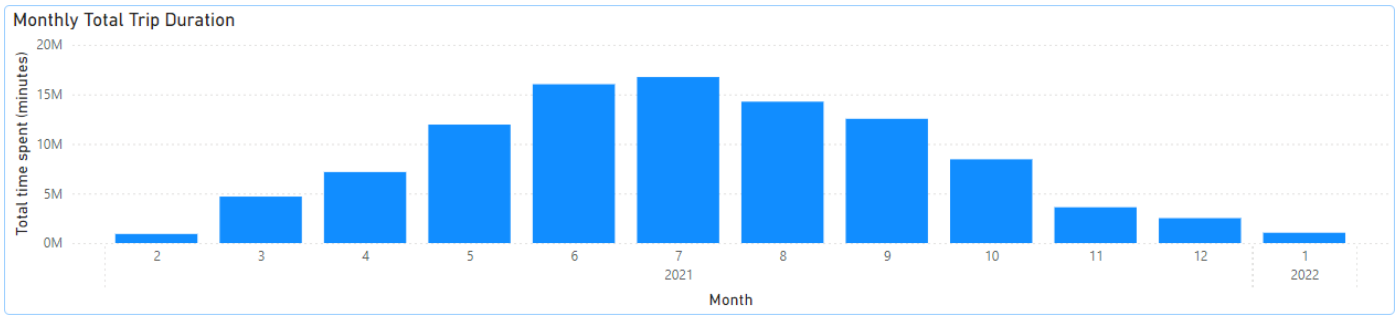

Power BI - Divvy - Data Analysis & Visualization • Last saved: Today at 6:22 PM
Search
File
Home
Insert
Modeling
View
Optimize
Help
Paste
Get data
Excel workbook
OneLake data hub
SQL Server
Enter data
Dataverse
Recent sources
Transform data
Refresh
New visual
Text box
More visuals
Clipboard
Data
Queries
Insert
Calculations
Navigator
Display Options
divvy-synapse-workspace-ondemand.sql.azur...
dim_date
dim_rider
dim_station
fact_payment
fact_trip
staging_payment
staging_rider
staging_station
staging_trip
fact_trip
trip_id
trip_date
rider_id
rideable_ty
"21630F715038CCB0" 2/20/2021 57068 "classic
"42B25557198348CC" 2/23/2021 44811 "classic
"E3D38203A2D907DB" 2/8/2021 75943 "classic
"790D32CB7348D195" 2/3/2021 18091 "classic
"2544729C9DB4C814" 2/3/2021 57488 "classic
"57951942BDC4782D" 2/25/2021 59893 "classic
"2D5A5055572F29F9" 2/3/2021 70621 "classic
"ADE82BA0E58F0BD7" 2/26/2021 2604 "classic
"CCFD5BABE6FB51FB" 2/27/2021 10509 "electri
"0BCFF30A58B46076" 2/28/2021 64945 "classic
"5C968039396993E7" 2/9/2021 64407 "classic
"21B48B71C7151791" 2/27/2021 54428 "classic
Select Related Tables
Load
Transform Data
Cancel

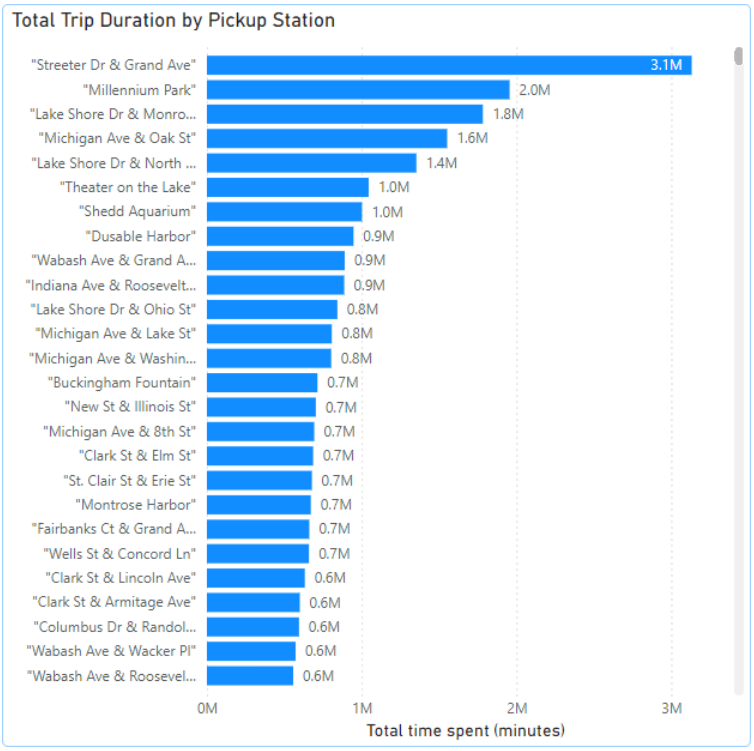
In the Model view, set the relationship between fact and dimension tables.

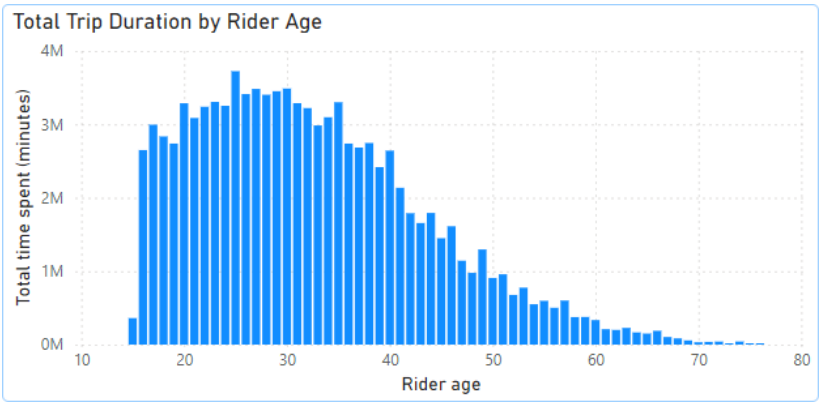


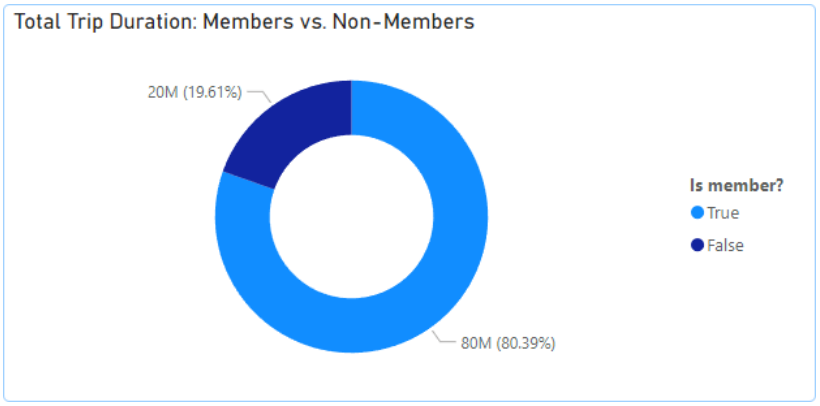
Explore and analyze the data by creating visualizations.


Daily Total Trip Duration
Total time spent (minutes)
1.0M
0.5M
0.0M
Mar 2021
May 2021
Jul 2021
Sep 2021
Nov 2021
Jan 2022
Date

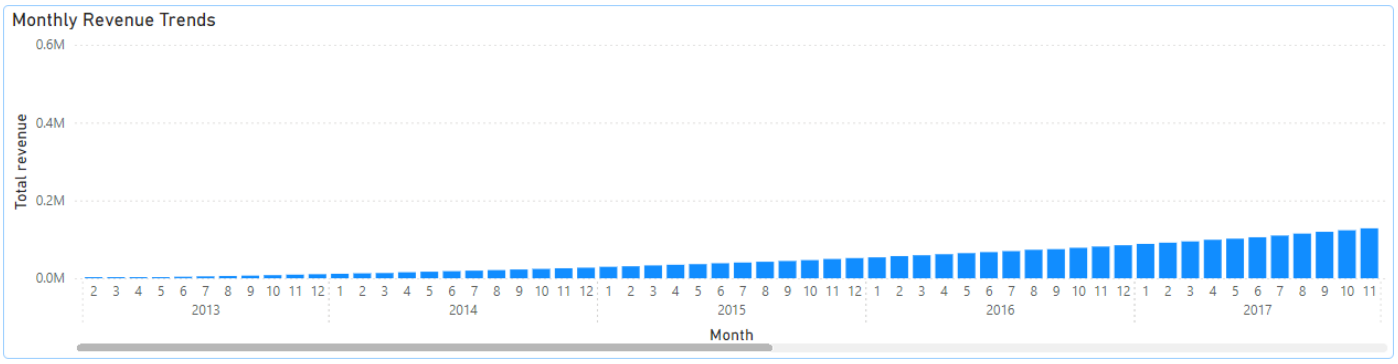

Monthly Total Trip Duration
Total time spent (minutes)
20M
15M
10M
5M
0M
2
3
4
5
6
7
2021
8
9
10
11
12
1
2022
Month

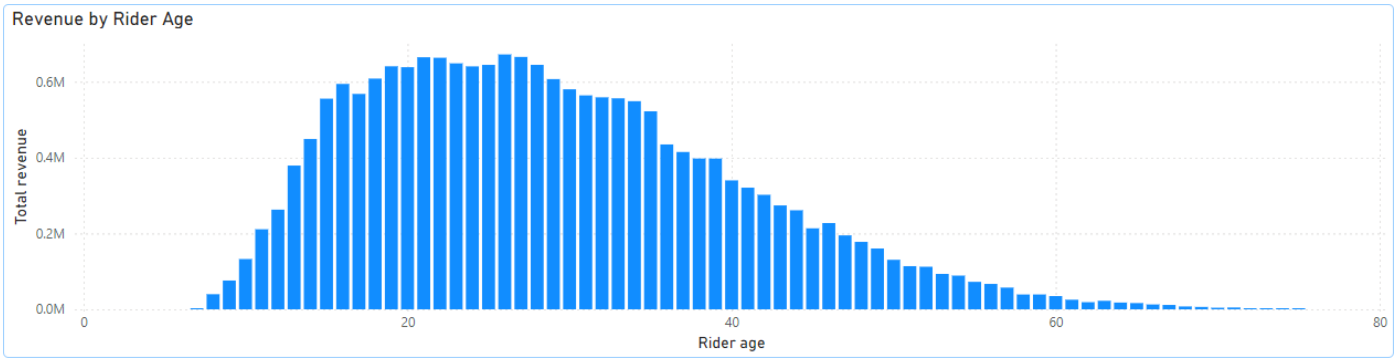

Total Trip Duration by Pickup Station
"Streeter Dr & Grand Ave" 3.1M
"Millennium Park" 2.0M
"Lake Shore Dr & Monro... 1.8M
"Michigan Ave & Oak St" 1.6M
"Lake Shore Dr & North ... 1.4M
"Theater on the Lake" 1.0M
"Shedd Aquarium" 1.0M
"Dusable Harbor" 0.9M
"Wabash Ave & Grand A... 0.9M
"Indiana Ave & Roosevelt... 0.9M
"Lake Shore Dr & Ohio St" 0.8M
"Michigan Ave & Lake St" 0.8M
"Michigan Ave & Washin... 0.8M
"Buckingham Fountain" 0.7M
"New St & Illinois St" 0.7M
"Michigan Ave & 8th St" 0.7M
"Clark St & Elm St" 0.7M
"St. Clair St & Erie St" 0.7M
"Montrose Harbor" 0.7M
"Fairbanks Ct & Grand A... 0.7M
"Wells St & Concord Ln" 0.7M
"Clark St & Lincoln Ave" 0.6M
"Clark St & Armitage Ave" 0.6M
"Columbus Dr & Randol... 0.6M
"Wabash Ave & Wacker Pl" 0.6M
"Wabash Ave & Roosevel... 0.6M
0M
1M
2M
3M
Total time spent (minutes)


Total Trip Duration by Rider Age
Total time spent (minutes)
4M
3M
2M
1M
0M
10
20
30
40
50
60
70
80
Rider age


Total Trip Duration: Members vs. Non-Members
20M (19.61%)
80M (80.39%)
Is member?
True
False


Quarterly Revenue Trends
Total revenue
2.0M
1.5M
1.0M
0.5M
0.0M
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year & Quarter


Monthly Revenue Trends
Total revenue
0.6M
0.4M
0.2M
0.0M
2013
2014
2015
2016
2017
Month


Revenue by Rider Age
Total revenue
0.6M
0.4M
0.2M
0.0M
0
20
40
60
80
Rider age