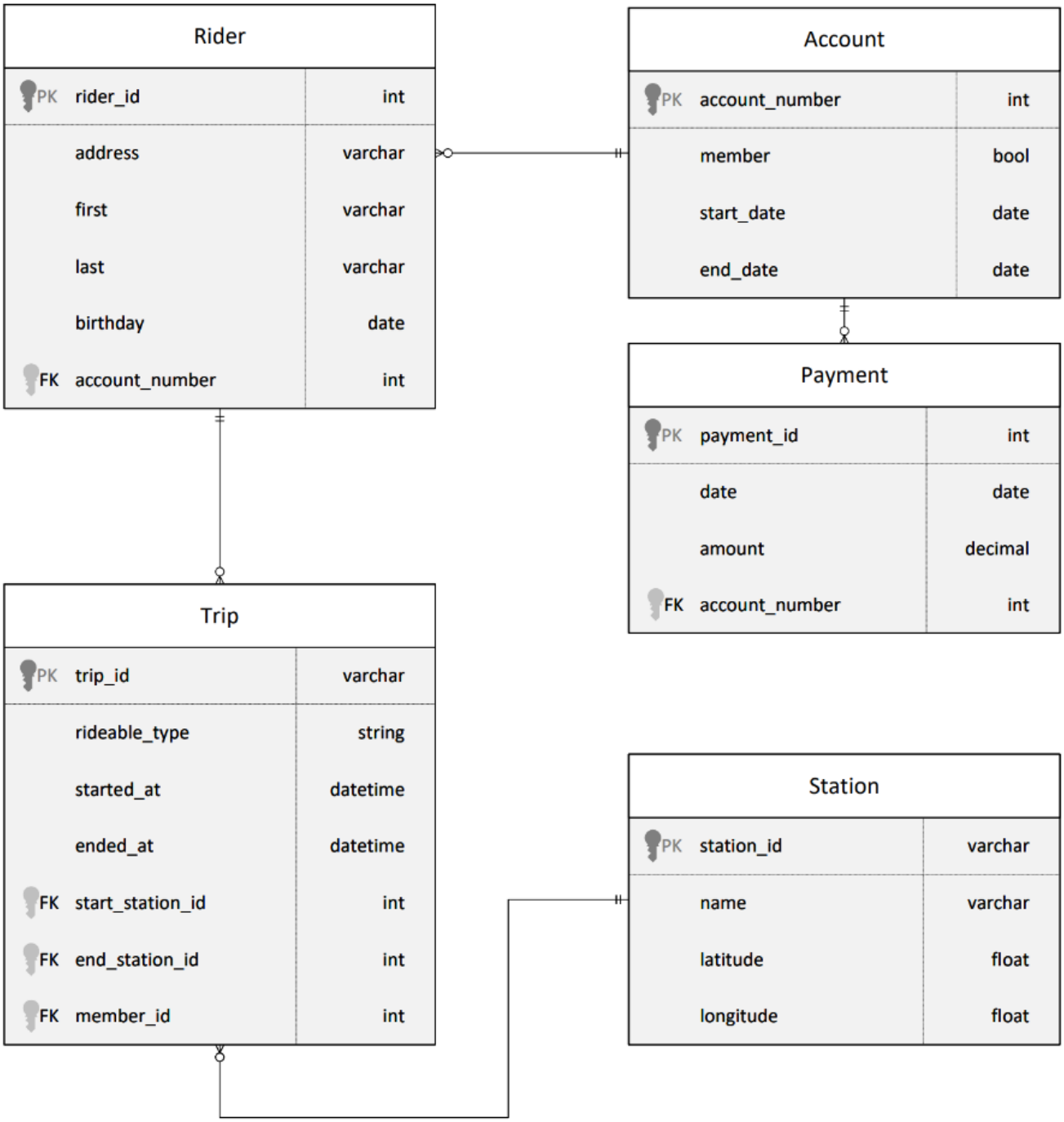
Building an Azure Data Warehouse for Bike Share Data Analytics

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:

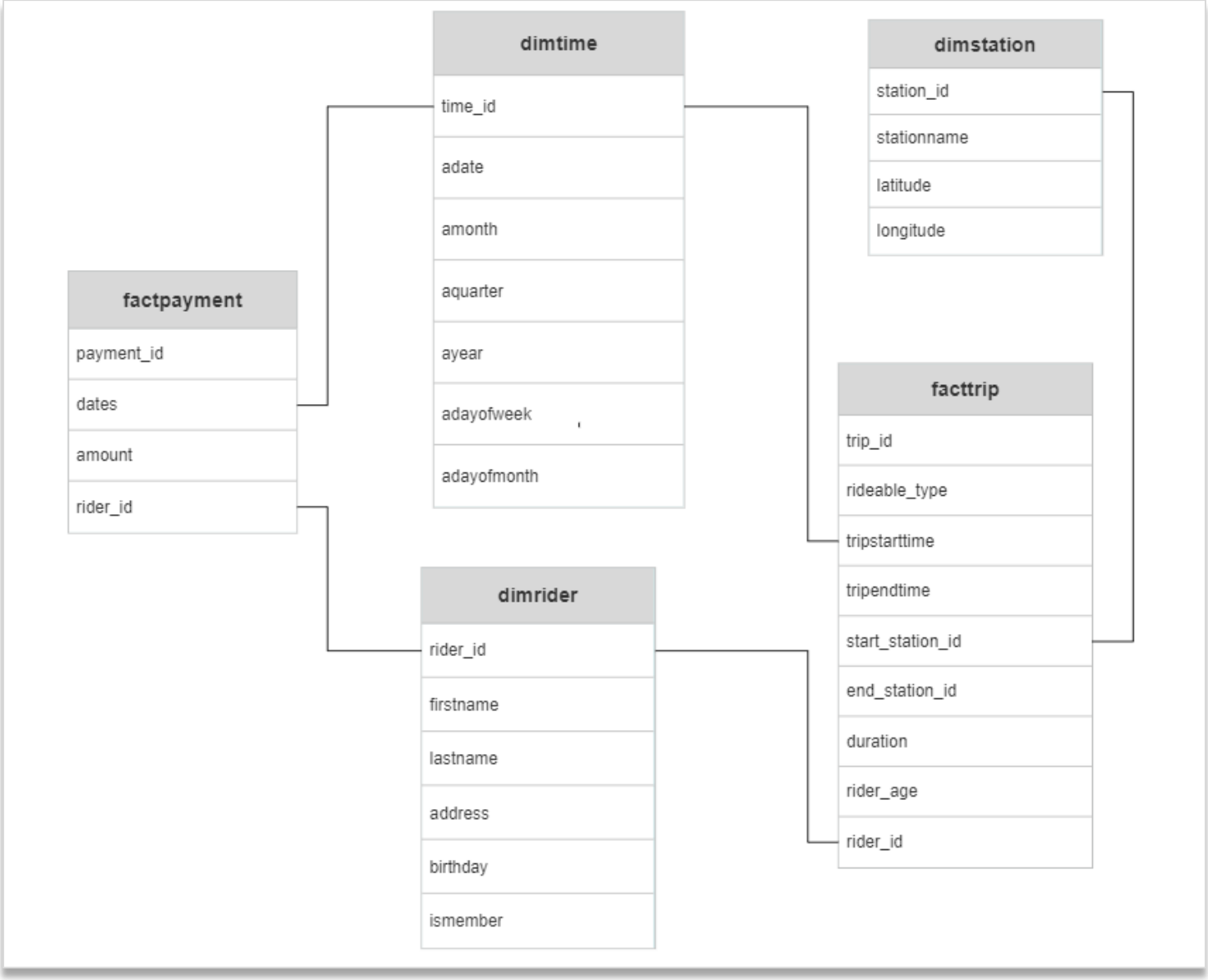
- Design a star schema based on the business outcomes listed below;
- Import the data into Synapse;
- Transform the data into the star schema;
- and finally, view the reports from Analytics.

The business outcomes you 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
3. Analyze how much money is spent per member
    - Based on how many rides the rider averages per month
    - Based on how many minutes the rider spends on a bike per month

Task 1: Design a star schema

Designing a Star Schema based on the relation diagram and the business problems outlined above:



Star Schema design comprising 2 fact tables, and 3 dimension tables:

FACT TABLES:

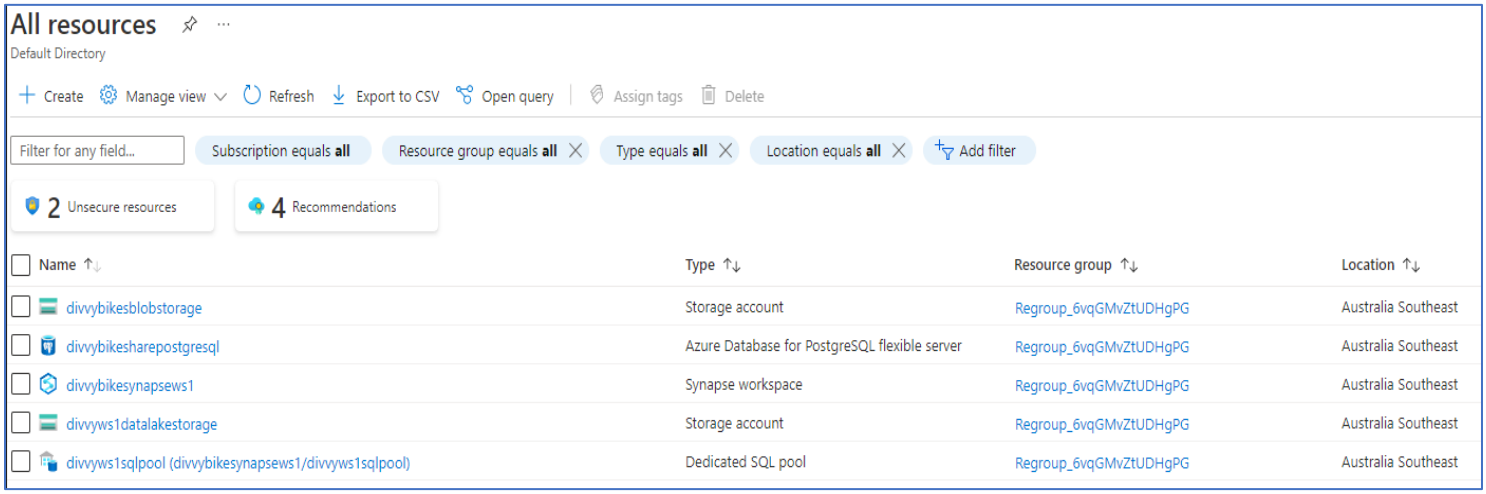
- FactTrip
- FactPayment

DIMENSION TABLES:

- Dimrider
- DimTime
- Dimstation

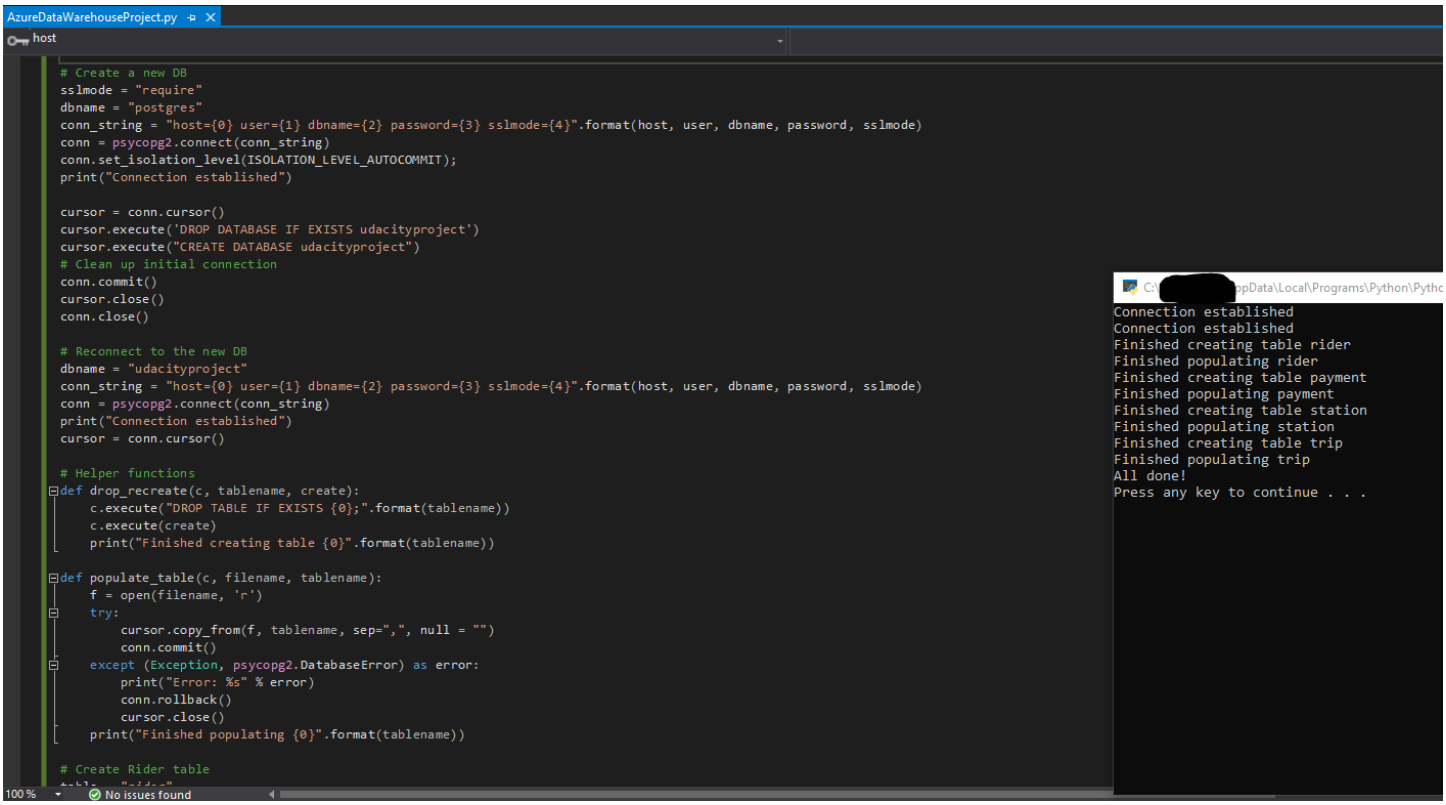
Task 2: Create Azure resources

- Azure PostgreSQL database
- Azure Blob Storage
- Azure Synapse workspace (comes with Azure DataLake Gen2 storage in the process)
- Dedicated SQL Pool and database within the Synapse workspace (Connected on Synapse studio)

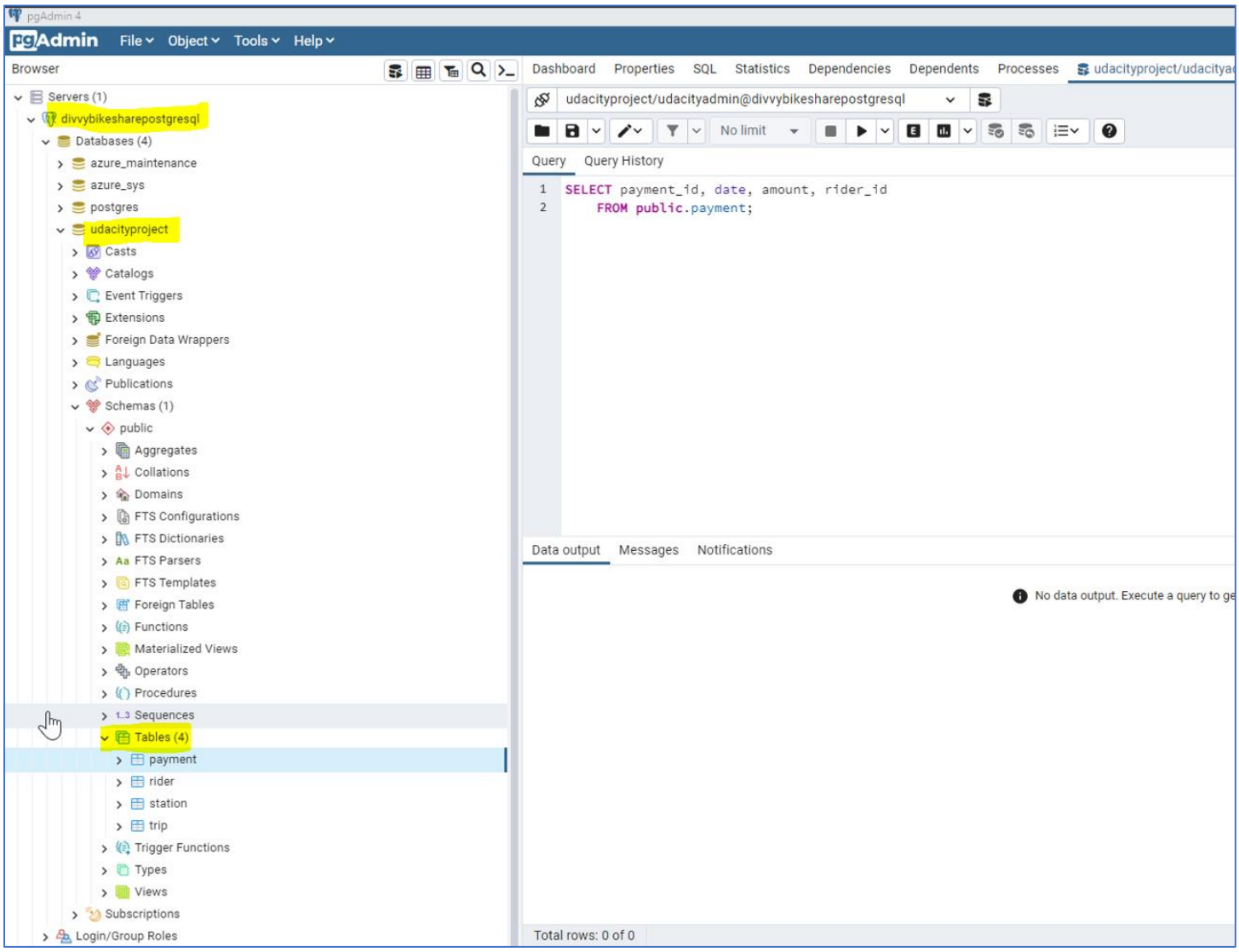


Task 3: Create the data in PostgreSQL

Downloaded **Github: ProjectDataToPostgres.py** and the **Data Files**

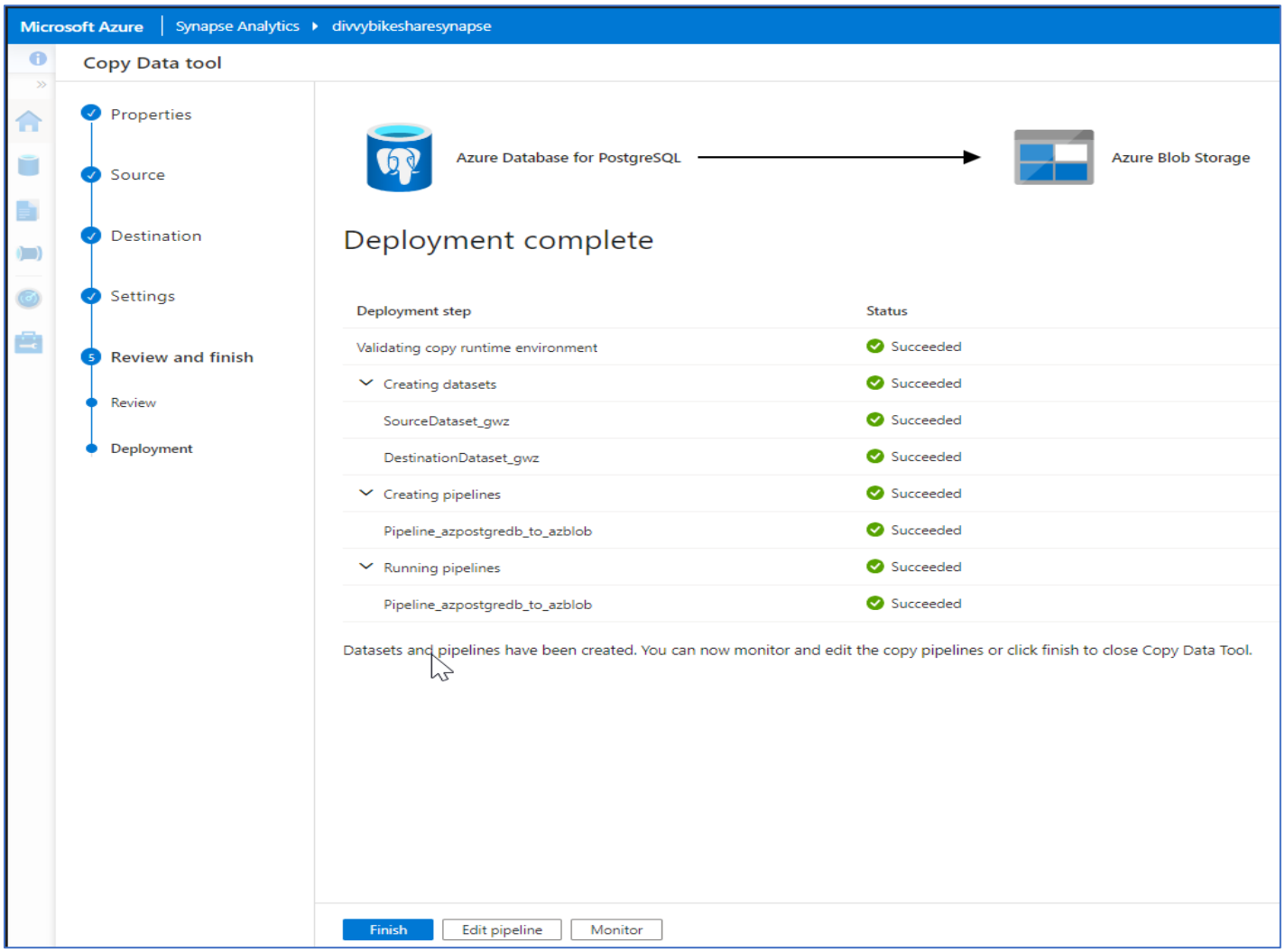


Uploaded all 4 data files using the Python script file in Visual Studio, by updating the host, username, and password information of the Azure Database for PostgreSQL resource. Verified this data exists by using pgAdmin4.

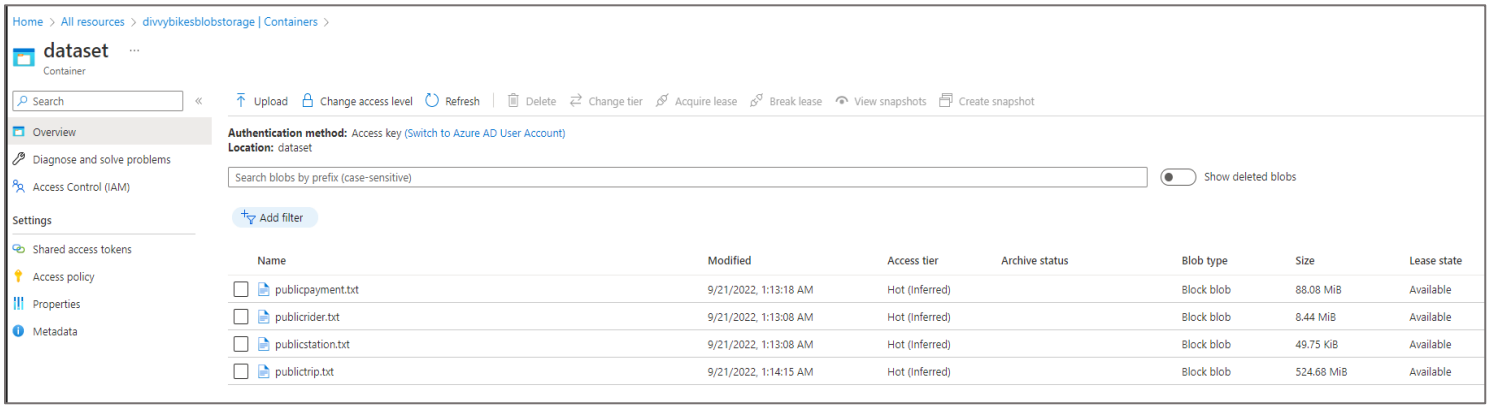


Task 4: EXTRACT the data from PostgreSQL

In Azure Synapse workspace, created a one-time pipeline that ingests the data from PostgreSQL into Azure Blob Storage. This will result in all four tables being represented as text files in Blob Storage, ready for loading into the data warehouse.



Proof of Extract: Azure Blob storage container with the 4 extracted files



Task 5: LOAD the data into external tables in the data warehouse

Once in Blob storage, the files will be shown in the data lake node in the Synapse Workspace. From here, you can use the script generating function to load the data from blob storage into external staging tables in the data warehouse you created using the Dedicated SQL Pool.

divvyws1datalakefile... | SQL script 1 | SQL script 2 | SQL script 3 | SQL script 4

Run | Undo | Publish | Query plan | Connect to divvyws1sqlpool | Use database divvyws1sqlpool

```sql

CREATE EXTERNAL TABLE staging_trip (
    [trip_id] nvarchar(4000),
    [rideable_type] nvarchar(4000),
    [tripstarttime] VARCHAR(50),
    [tripendtime] VARCHAR(50),
    [start_station_id] nvarchar(4000),
    [end_station_id] nvarchar(4000),
    [rider_id] bigint
    )
    WITH (
    LOCATION = 'publictrip.txt',
    DATA_SOURCE = [divvyws1datalakefilesys_divvyws1datalakestorage_dfs_core_windows_net],
    FILE_FORMAT = [SynapseDelimitedTextFormat]
    )
GO

```

divvyws1datalakefile... | SQL script 1 | SQL script 2

Run | Undo | Publish | Query plan | Connect to divvyws1sqlpool | Use database div

```sql

CREATE EXTERNAL TABLE staging_rider (
    [rider_id] bigint,
    [firstname] nvarchar(4000),
    [lastname] nvarchar(4000),
    [address] nvarchar(4000),
    [birthday] VARCHAR(50),
    [accountstartdate] VARCHAR(50),
    [accountenddate] VARCHAR(50),
    [currentmember] bit
    )
    WITH (
    LOCATION = 'publicrider.txt',
    DATA_SOURCE = [divvyws1datalakefilesys_divvyws1datalakestorage_dfs_core_windows_net],
    FILE_FORMAT = [SynapseDelimitedTextFormat]
    )
GO

```

```sql
CREATE EXTERNAL TABLE staging_payment (
    [payment_id] bigint,
    [dates] varchar(50),
    [amount] float,
    [rider_id] bigint
    )
    WITH (
    LOCATION = 'publicpayment.txt',
    DATA_SOURCE = [divvyws1datalakefilesys_divvyws1datalakestorage_dfs_core_windows_net],
    FILE_FORMAT = [SynapseDelimitedTextFormat]
    )
GO


SELECT TOP 100 * FROM dbo.staging_payment
GO
```

divvyws1datalakefile... | SQL script 1 | SQL script 2 | SQL script 3

Run | Undo | Publish | Query plan | Connect to divvyws1sqlpool | Use database divvyws1sqlp

```sql

CREATE EXTERNAL TABLE staging_station (
    [station_id] nvarchar(4000),
    [stationname] nvarchar(4000),
    [latitude] float,
    [longitude] float
    )
    WITH (
    LOCATION = 'publicstation.txt',
    DATA_SOURCE = [divvyws1datalakefilesys_divvyws1datalakestorage_dfs_core_windows_net],
    FILE_FORMAT = [SynapseDelimitedTextFormat]
    )
GO
```

Microsoft Azure | Synapse Analytics ▸ divvybikesynapsews1 | Search

Synapse live | Validate all | Publish all

Data

Workspace | Linked

Filter resources by name

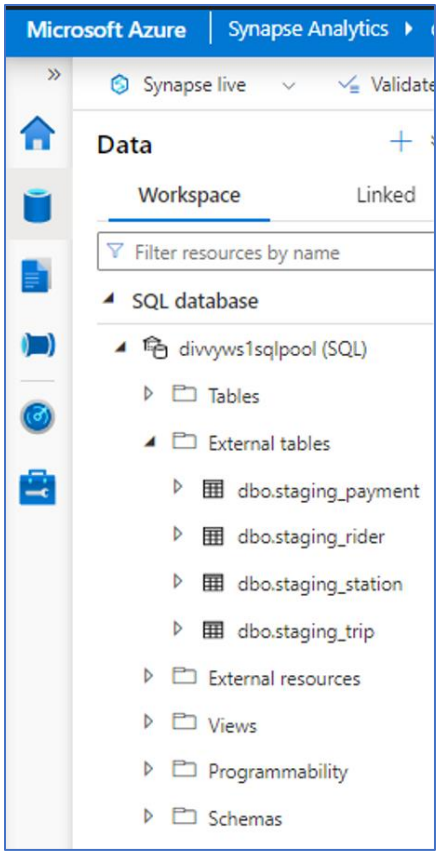
- Azure Blob Storage 1
  - azbloblink (divvybikesblobstorage)
    - dataset
- Azure Data Lake Storage Gen2 2
  - divvybikesynapsews1 (Primary - div...
    - divvyws1datalakefilesys (Primary)
  - (Attached Containers)

dataset | divvyws1datalakefile...

New SQL script | New notebook | Upload | Download | New folder | Select all | Rename

divvyws1datalakefilesys

Name	Last Modified
publicpayment.txt	9/23/2022, 12:24:06 PM
publicrider.txt	9/23/2022, 12:24:01 PM
publicstation.txt	9/23/2022, 12:24:16 PM
publictrip.txt	9/23/2022, 12:30:08 PM



Task 6: TRANSFORM the data to the star schema

We will write SQL scripts to transform the data from the staging tables to the final star schema designed.