

|  |  |
| --- | --- |
| Project: Building an Azure Data Warehouse for Bike Share Data Analytics | |
|  |  |
| Jun 9 2024 |
| Dahi Nemutlu | |

Contents

[1. Project Overview 3](#_Toc169916960)

[2. Create Azure Resources 5](#_Toc169916961)

[2.1. Create Azure Database for PostgreSQL 5](#_Toc169916962)

[2.2. Create Azure Synapse Workspace 6](#_Toc169916963)

[3. Create the data in PostgreSQL 6](#_Toc169916964)

[4. Extract the data from PostgreSQL 7](#_Toc169916965)

[5. Load the data into external tables in the data warehouse 10](#_Toc169916966)

[6. Transform the data to the star schema 10](#_Toc169916967)

[7. Analyze and visualize data with Power BI 11](#_Toc169916968)

|  |
| --- |
| 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:   Create Azure ResourcesCreate Azure Database for PostgreSQL  Create Azure Synapse Workspace  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.   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.       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.   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.   Analyze and visualize data with Power BI Connect to the SQL database using Serverless SQL endpoint and import star schema tables into Power BI. |
| In the Model view, set the relationship between fact and dimension tables.    Explore and analyze the data by creating visualizations. |