Using Microsoft Power BI with AllegroGraph

There are multiple methods to integrate AllegroGraph SPARQL results into Microsoft Power BI. In this document we describe two best practices to automate queries and refresh results if you have a production AllegroGraph database with new streaming data:

The first method uses Python scripts to feed Power BI. The second method issues SPARQL queries directly from Power BI using POST requests.

**Method 1: Python Script:**

Assuming you know Python and have it installed locally, this is definitely the easiest way to incorporate SPARQL results into Power BI. The basic idea of the method is as follows: First, the Python script enables a connection to your desired AllegroGraph repository. Then we utilize AllegroGraph’s Python API within our script to run a SPARQL query and return it as a Pandas dataframe. When running this script within Power BI Desktop, the Python scripting service recognizes all unique dataframes created, and allows you to import the dataframe into Power BI as a table, which can then be used to create visualizations.

**Requirements:**

1. You must have the AllegroGraph Python API installed. If you do not, installation instructions are here: https://franz.com/agraph/support/documentation/current/python/install.html
2. Python scripting must be enabled in Power BI Desktop. Instructions to do so are here: https://docs.microsoft.com/en-us/power-bi/connect-data/desktop-python-scripts
a) As mentioned in the article, pandas and matplotlib must be installed. This can be done with ‘pip install pandas’ and ‘pip install matplotlib’ in your terminal.

The Process:

Once these requirements have been met, create a Python file with whatever script editor you usually use. The following code will create a connection to your desired repository. For this example, we will be using the Kennedy dataset that is available with the AllegroGraph distribution (See the ‘Tutorial’ directory). Load the Kennedy.ntriples file into your running AllegroGraph. (Replace the ‘****’ in the code with your corresponding username and password.)

```python
#the necessary imports
import os

from franz.openrdf.connect import ag_connect
from franz.openrdf.query.query import QueryLanguage

import pandas as pd

#connect to your agraph repository

def setup_env_var(var_name, value, description):
    os.environ[var_name] = value
    print("{}: {}".format(description, value))

setup_env_var('AGRAPH_HOST', 'localhost', 'Hostname')
setup_env_var('AGRAPH_PORT', '10035', 'Port')
setup_env_var('AGRAPH_USER', '****', 'Username')
```
setup_env_var('AGRAPHP_PASSWORD', '****', 'Password')
conn = ag_connect('kennedy', create=False, clear=False)

2. We then want to create a query. For this example, we will first show what our data looks like, what the visual query of the information is, and what the written query looks like. With the following query we want every person’s first and last names, as well as their birth years. Here is a small portion of the data visualized in Gruff, and then the visualization of the query:
3. Then add the written query to the python script as a variable string (we added an additional line to the query to sort on birth year). Next use the API functionality to simply execute the query and turn the results into a pandas dataframe.

```python
query = """select ?person ?first_name ?last_name ?birth_year
where
{ ?person <http://www.franz.com/simple#first-name> ?first_name ;
  <http://www.franz.com/simple#birth-year> ?birth_year ;
  rdf:type <http://www.franz.com/simple#person> ;
  <http://www.franz.com/simple#last-name> ?last_name .
}
order by desc(?birth_year)""

with conn.executeTupleQuery(query) as result:
    df = result.toPandas()
```

When looking at the result, we see that we have a DataFrame!

![DataFrame](image)

4. Now we will use this script in Power BI. When in Power BI Desktop, go to ‘Get Data’ and look for the python script option. Then simply copy and paste your entire script into the
text box, and run the script. In this case, our output looks like this:

5. Next simply ‘Load’ the data, and then you can use the Power BI Desktop interface to create whatever visualizations you want! If you do have a lot of additional operations to perform on your dataframe, we recommend doing these in your python script.

**Method 2: POST Request:**

For the SPARQL query via POST requests to work you need to url-encode the query. Every modern programming language will support that, but in our example we will be using Python again. This method is better for when you do not have python locally installed or prefer a different programming language.

It is possible to send a GET request from Power BI, but once the results from the query reach a certain size, a POST request is required, which is confusing to do within the Power BI Desktop interface. The following steps will show you how to do SPARQL Queries using POST requests. It looks a bit odd but it works well.

**The Process:**

1. In your AG WebView create an ‘anonymous’ user. (Go to admin -> Users -> [add a user] -> and add ‘anonymous’ as
username without adding a password). You can use these settings:

2. Go to your desired repository in WebView and Click on ‘Queries’ -> ‘New’

3. Write a simple SPARQL query, and run it to make sure you get the correct response back.

4. In python create the following script: (Assuming your AllegroGraph is on your localhost port 10035 and your repo is called ‘kennedy’)

   ```python
   import urllib
   def CreatePOSTquery(query):
       response = start + urllib.parse.quote(query)
       return response
   ```

   This function url-encodes the query and attaches it to the POST request. Replace the ‘localhost:10035’ and ‘kennedy’ strings in the start variable with your corresponding data. Then, using the same query as our previous example, we create
our url-encoded POST query:

query = """select ?person ?first_name ?last_name ?birth_year where 
{ ?person <http://www.franz.com/simple#first-name> ?first_name ;
  <http://www.franz.com/simple#birth-year> ?birth_year ;
  rdf:type <http://www.franz.com/simple#person> ;
  <http://www.franz.com/simple#last-name> ?last_name .
} order by desc(?birth_year)"""

result = CreatePOSTquery(query)
print(result)

This gives us the following result:

```
[16] result
```

5. Within Power BI Desktop we go to ‘Get data’ and create a ‘Blank query’ and go into the ‘Advanced Editor’ window. Using the following format we will get our desired results (please note that due to the length of the url-encoded request, it did not all fit in the image. Copy and pasting into the url field works fine. The ‘url’ variable needs to be in quotes and have a comma at the end):
We see the following results:

<table>
<thead>
<tr>
<th>Column3</th>
<th>Column2</th>
<th>Column1</th>
</tr>
</thead>
<tbody>
<tr>
<td>first_name</td>
<td>last_name</td>
<td>dob</td>
</tr>
<tr>
<td>Harry</td>
<td>Skuld</td>
<td>1972</td>
</tr>
<tr>
<td>Molly</td>
<td>Stark</td>
<td>1958</td>
</tr>
<tr>
<td>Barry</td>
<td>Kennedy</td>
<td>1998</td>
</tr>
<tr>
<td>Mildred</td>
<td>Kennedy</td>
<td>1997</td>
</tr>
<tr>
<td>Mark</td>
<td>Bailey</td>
<td>1987</td>
</tr>
<tr>
<td>Amarda</td>
<td>Smith</td>
<td>1967</td>
</tr>
<tr>
<td>Alfred</td>
<td>Tolkien</td>
<td>1972</td>
</tr>
<tr>
<td>Frank</td>
<td>Kennedy</td>
<td>1962</td>
</tr>
<tr>
<td>Carolynn</td>
<td>Rosewater</td>
<td>1956</td>
</tr>
<tr>
<td>Carl</td>
<td>Head</td>
<td>1960</td>
</tr>
<tr>
<td>Hana</td>
<td>You</td>
<td>1985</td>
</tr>
<tr>
<td>Anthony</td>
<td>Shriver</td>
<td>1985</td>
</tr>
<tr>
<td>Alina</td>
<td>Mulcahy</td>
<td>1945</td>
</tr>
<tr>
<td>Matthew</td>
<td>Kennedy</td>
<td>1945</td>
</tr>
<tr>
<td>Mark</td>
<td>Shriver</td>
<td>1999</td>
</tr>
<tr>
<td>Victoria</td>
<td>Hayes</td>
<td>1994</td>
</tr>
<tr>
<td>Frances</td>
<td>Kennedy</td>
<td>1963</td>
</tr>
<tr>
<td>Christopher</td>
<td>Kennedy</td>
<td>1963</td>
</tr>
</tbody>
</table>

6. One last step is to turn the top row into the column names, which can be achieved by pressing the ‘Use first row as headers’:

The best part about both of these methods is that once the query has been created, Power BI can refresh the visuals using the same queries if your data changed. This can be achieved by
Franz Inc. Named an AI 50 Company by KMWorld

AllegroGraph Powering Intelligent Knowledge Graph Solutions

Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology for Knowledge Graph Solutions, today announced that it has been named to The AI 50 — Companies Empowering Intelligent Knowledge Management Companies by KMWorld. The annual list reflects the urgency felt among many organizations to provide a timely flow of targeted information. Among the more prominent initiatives is the use of AI and cognitive computing, as well as related capabilities such as machine learning, natural language processing, and text analytics. This list recognizes companies based on their presence, execution, vision and innovation in delivering products and services to the marketplace.

“As the drive for digital transformation becomes an imperative for companies seeking to compete and succeed in all industry sectors, intelligent tools and services are being leveraged to enable speed, insight, and accuracy,” said Tom Hogan, Group Publisher at KMWorld. “To showcase organizations that are incorporating AI and an assortment of related
technologies—including natural language processing, machine learning, and computer vision—into their offerings, KMWorld created the “AI 50: The Companies Empowering Intelligent Knowledge Management.”

“Franz Inc. has a rich history in AI and we are honored to receive this acknowledgement for our efforts in delivering AI Knowledge Graph Solutions,” said Dr. Jans Aasman, CEO, Franz Inc. “In the past year, we have seen demand for Intelligent Data Fabrics take off across industries along with recognition from top technology analyst firms that Knowledge Graphs provide the critical foundation for Enterprise Wide Data Fabrics. Our recent launch of AllegroGraph 7 with FedShard, a breakthrough that allows infinite data integration to unify all data and siloed knowledge into an Entity-Event Knowledge Graph solution will catalyze Data Fabric deployments across the Enterprise.”

Gartner’s Top 10 Trends in Data and Analytics for 2020 noted “Relationships form the foundation of data and analytics value. By 2023, graph technologies will facilitate rapid contextualization for decision making in 30% of organizations worldwide. Graph analytics is a set of analytic techniques that allows for the exploration of relationships between entities of interest such as organizations, people and transactions. Data and analytics leaders need to evaluate opportunities to incorporate graph analytics into their analytics portfolios and applications to uncover hidden patterns and relationships. In addition, consider investigating how graph algorithms and technologies can improve your AI and ML initiatives.” (Source: Gartner, Top 10 Trends in Data and Analytics for 2020, June 9, 2020).

“Graph databases and knowledge graphs are now viewed as a must-have by enterprises serious about leveraging AI and predictive analytics within their organization,” said Dr. Aasman “We are working with organizations across a broad range of industries to deploy large-scale, high-performance Entity-
Event Knowledge Graphs that serve as the foundation for AI-driven Data Fabrics for personalized medicine, predictive call centers, digital twins for IoT, predictive supply chain management and domain-specific Q&A applications – just to name a few.”

**Forrester Shortlists AllegroGraph**

AllegroGraph was shortlisted in the February 3, 2020 Forrester Now Tech: Graph Data Platforms, Q1 2020 report, which recommends that organizations “Use graph data platforms to accelerate connected-data initiatives.” Forrester states, “You can use graph data platforms to become significantly more productive, deliver accurate customer recommendations, and quickly make connections to related data.”

**Bloor Research covers AllegroGraph with FedShard**

Bloor Research Analyst, Daniel Howard noted “With the 7.0 release of AllegroGraph, arguably the most compelling new capability is its ability to create what Franz refers to as “Entity-Event Knowledge Graphs” (or EEKGs) via its patented FedShard technology.” Mr. Howard goes on to state “Franz clearly considers this a major release for AllegroGraph. Certainly, the introduction of an explicit entity-event graph is not something I’ve seen before. The newly introduced text to speech capabilities also seem highly promising.”

**AllegroGraph Named to DBTA’s 100 Companies That Matter Most in Data**

AllegroGraph was also recently named to DBTA’s 100 Companies That Matter Most in Data. The DBTA 100 showcases organizations that delivering solutions for customers to meet the need for real-time, data-driven insights.

**Franz Knowledge Graph Technology and Services**

Franz’s Knowledge Graph Solution includes both technology and
services for building industrial strength Entity-Event Knowledge Graphs based on best-of-class tools, products, knowledge, skills and experience. At the core of the solution is Franz’s graph database technology, AllegroGraph with FedShard, which is utilized by dozens of the top F500 companies worldwide and enables businesses to extract sophisticated decision insights and predictive analytics from highly complex, distributed data that cannot be uncovered with conventional databases.

Franz delivers the expertise for designing ontology and taxonomy-based solutions by utilizing standards-based development processes and tools. Franz also offers data integration services from siloed data using W3C industry standard semantics, which can then be continually integrated with information that comes from other data sources. In addition, the Franz data science team provides expertise in custom algorithms to maximize data analytics and uncover hidden knowledge.

About Franz Inc.

Franz Inc. is an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology with expert knowledge in developing and deploying Knowledge Graph solutions. The foundation for Knowledge Graphs and AI lies in the facets of semantic technology provided by AllegroGraph with FedShard and Allegro CL. The ability to rapidly integrate new knowledge is the crux of the Knowledge Graph and Franz Inc. provides the key technologies and services to address your complex challenges. Franz Inc. is your Knowledge Graph technology partner.

All trademarks and registered trademarks in this document are the properties of their respective owners.