Qlik-to-Power BI Accelerator

Created by – Shreya D (Software Developer – DW Practice)

Table of Contents

1. Overview

Name: Qlik – to -Power BI Accelerator

2. Purpose:

Automatically converts Qlik scripts, measures and dashboards into production-ready Power BI artifacts (DAX, measures , PySpark Transformation Code, and Power BI files) with accurate naming conventions, support for complex functions and visuals (including custom visuals), and preservations of interactivity (slicers, bookmarks , filters , buttons).

3. Scope:

This Document covers three sub-models:

1. Dax Converter: Convert Qlik measures to DAX measures with table/column naming and syntactically – correct DAX expressions.
2. Back-end Converter: Convert Qlik transformations scripts to PySpark code that reproduce data transformation logic.
3. Visualization Converter: Convert Qlik image dashboards to Power BI file output preserving visuals, slicer, bookmarks, filters, custom visuals , multiple pages and interactivity.

4. Actors:

1. Business User: upload measures, scripts or dashboards, validate output.
2. Data Engineer: Review backend PySpark, tunes naming and integrating.
3. BI Developer: Validates DAX syntax, visuals and user experience in Power BI
4. System (Accelerator) : Service that performs the conversions.

5. Assumptions:

1. Input Qlik assets follow standard syntax and names for table/Column when available.
2. Pre data loaded to view visuals, handling custom visuals.

6. Non – Functional Requirements:

1. Accuracy: At least 95% correct conversion for expressions and tranformations in common enterprise workloads.
2. Performance: DAX conversions should complete within minutes for single measures; script conversions within minutes depending on script-size; full dashboards conversion within a reasonable time.

UI Requirements:

- **Two UI variants Must be supported:**
  a. **Streamlit Applications –** Frontend and backend both orchestrated through python for rapid prototyping and internal demon.
     - ➢ UI must be clean minimal and optimized for fast interactions.
     - ➢ Real time logs or progress indicator for conversions.

  b. **Production UI** (React + FastAPI)-
     - ➢ React server=s as the production-grade frontend with robust routing.
     - ➢ FastAPI handles high-performance backend processing and API orchestration.
     - ➢ Must be visually clean, and optimized for fast interactions.

7. High Level Flows:

1) DAX Converter:

- What it does: This translates the logic/formulas. It converts Qlik "Set Analysis" into Power BI DAX measures.
- It works in 3 steps:
  a. Validate: You upload a script to let the system understand your data model (tables and columns).

  b. Single Convert: You can paste a specific Qlik Set Expression to get the immediate DAX equivalent.
  c. Batch Convert: You can upload a CSV of multiple measures to convert them all at once.

Acceptance Criteria:

- No placeholder or temporary tables names in final DAX.

2) **Back-end Converted:**
   - What it does: This focuses on the data engineering side. You upload a Qlik Data Load script, and the app converts the proprietary Qlik script into PySpark code.
   - Output: It generates code compatible with Microsoft Fabric and a Semantic Model in JSON format. It supports two modes: "Transformation" and "Extraction".
   - It should handle complex logics like Apply map, Subquery , Date etc,.

3) **Visualization Converter:**
   - What it does: This automates the visual creation. You upload a screenshot (image) of your existing dashboard and provide a name.
   - Output: The AI analyzes the image to understand the layout and chart types, then generates a downloadable Power BI file that attempts to replicate the visual design of the original dashboard.

     Acceptance Criteria:
     - All report pages open in Power BI Desktop without errors.
     - Visualizations render and support slicers, filters , bookmarks , custom visuals.