**Data visualization with Power BI using FHIR analytics pipelines for Breast Cancer Screening (BCS)**

With Power BI as Data visualization tool, this is an application to visualize FHIR Data using FHIR Analytics Pipeline. In pipeline new or updated FHIR data is moved to Azure Data Lake stores data in parquet files, from parquet files we create external tables and views in Azure Synapse Analytics database with Serverless SQL pool. On the Database we have stored procedures to calculate data as per quality measures, that stratified data is visualized on power BI desktop report.

**About FHIR to Data Lake pipeline:**

The FHIR Analytics Pipeline is an Azure Function that continually exports new and modified FHIR resources in specific time chunks to Azure Data Lake Storage in the form of Parquet files. By default, this function runs every 5 minutes and exports changed data in five-minute chunks (or windows). Each export window will only export the latest version of a resource and a single resource will exist in multiple windows as it's changed in the FHIR service.

Diagram

Description automatically generated

Figure 1 - Sample Data Flow for FHIR analytics pipeline and Power BI sample application

**Database in Synapse:**

A database is created in Synapse with serverless pool, database has External Tables and Views pointing to the parquet files in the Azure Data Lake Storage.

**About Power BI Dashboard:**

The Power BI dashboard demonstrated in this document uses the sample quality measure below to get the data from Serverless SQL tables/views using stored procedures to show data visualization in power BI.

**Quality Measure**:

Percentage of women 50-70 years of age who had a mammogram to screen for breast cancer in the 48 months prior to the end of the measurement period.

Note: This sample is not the same as the HEDIS eCQI digital quality measure. This is a basic sample to demonstrate capabilities of FHIR analytics Pipeline and Data Visualization with Power BI. This sample uses Synthea data comprising of SNOMED codes

Below is the prerequisite that we need for this application:

1. Microsoft work or school account
2. Azure Synapse Workspace with Serverless SQL Endpoint.

* The Serverless SQL Endpoint will be used to connect to Database from Power BI Desktop Application to create Power BI Dashboard/reports.

1. Power BI Desktop application

* It is available for download [here](https://www.microsoft.com/en-us/download/details.aspx?id=58494).

1. Power BI service account

* Login to power BI service [here](https://msit.powerbi.com/home) and create a workspace where Power Bi reports will be published from Power Bi Desktop App

1. Microsoft SQL Server Management Studio

* It is available for Download [here](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16).

## **Connecting to Database from SQL Server Management Studio**

To explore (view/edit) the stored procedures definition for any stored procedure used for Power BI dashboard report, connect to database from synapse using serverless SQL pool endpoint, Serverless SQL endpoint could be found in Synapse Workspace as highlighted below:

Graphical user interface, text, application

Description automatically generated

1. Open the “Microsoft SQL Server Management Studio”, from “Object Explorer” menu click on “Connect” and select “Database Engine” from list.

Graphical user interface, application

Description automatically generated

1. Enter serverless SQL endpoint from Synapse workspace into server name textbox, choose authentication method, provide username ad click connect:

Graphical user interface

Description automatically generated

1. After clicking connect you will be asked to provide authentication details for user, once user authentication is done, database connection will be done:

Graphical user interface, application

Description automatically generated

1. Once the connection is made you would be able to see “fhirdb” database, stored procedures should be available inside “Programmability => Stored Procedures” folder as shown below:

Graphical user interface, application

Description automatically generated

## **Checking and editing the dashboard in Power BI desktop application**

Before proceeding ahead with dashboard, please connect to “fhirdb” database using serverless SQL endpoint from Microsoft SQL Server Management Studio and make sure the stored procedures are created there.

1. Save the provided report file (.pbix file) to your computer.

Graphical user interface, application

Description automatically generated

1. Double click on report file(.pbix) to open it in Power BI Desktop application, the report file has multiple pages, “Main Dashboard” is the landing page, as shown in below image:

Graphical user interface, application

Description automatically generated

The report has below pages:

1. Main Dashboard

* It’s an entry point for a report, it shows all the graphs at one place with navigation buttons to detailed pages.

1. Data By Age

* This page shows graph with Age range and age wise

1. State wise Percentage

* This page shows the map with state wise percentage of Numerator population

1. Data By Race Category

* This page shows graph with count of Numerator population with each Race Category

1. Data By Ethnicity Category

* This page shows graph with count of Numerator population with each Ethnicity Category

1. Payor Details

* This page shows graph with percentage of Numerator population with payors

1. Tabular Data

* This page shows table of data for Numerator population

The Main Dashboard page in report shows all the visualizations listed below:

* Age range chart
* State wise Percentage
* Count by Race Category
* Count by Ethnicity Category
* Payor wise percentage
* Button to navigate to tabular data

Age Range, Race Category and Ethnicity charts get data from “NumeratorDetails” table.

State wise Percentages chart gets data from “StatewisePercentage” table.

Payor wise Percentage chart gets data from “PayorwisePercetage” table.

1. Below image shows the “Data” section of the report which has tables, on selecting specific table in the right-hand side pane, it will show the data in that table as shown in below image:

Graphical user interface, application, table

Description automatically generated

1. The “Model” section shows all the table and the relationships (if any) as shown in below image:

Graphical user interface, application

Description automatically generated

## **Editing the query to change measurement period date range in parameters**

1. Go to “Model” section, here one can see the table is available, and can view properties and fields of the table, when you click “More Options” button (Three dots in top right corner of the table Shape, rectangle

   Description automatically generated), it shows all the options available for the table, click on “Edit Query”

Graphical user interface, application, Word

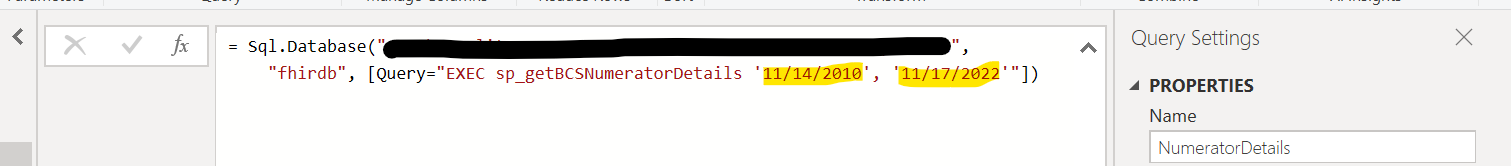
Description automatically generated

1. A new “Power Query Editor” window will open, click on expand arrow button as highlighted in below image which will show the query:

Table, Excel

Description automatically generated

1. Change the date value in query for measurement period start date and end date, first value is for start date and second value is for end date parameter respectively, so we have set 11/14/2010 as measurement period start date and 11/17/2022 as measurement period end date:



Change the serverless SQL pool URL(blacked out part),

After editing the values, click on right tick as highlighted in image below:

Table

Description automatically generated

1. There will be an “Edit permission” button, click on it.

Graphical user interface, application, Word

Description automatically generated

1. On new pop-up window click on “Run” button:

Graphical user interface, application

Description automatically generated

1. Query will be executed, latest result data will be loaded into the table, this will complete query editing, close the “Power Query Editor” window:

Table, Excel

Description automatically generated

1. In a similar manner date parameters can be changed for “StatewisePercentage” and “PayorwisePercentage” tables.

## **Publish the dashboard in Power BI Service**

To be able to view the dashboard in Power BI service we need to publish it from Power BI desktop application to Power BI Service.

1. If any changes are made to dashboard, save the changes, and then click on “Publish” option from “Home” menu as shown in image below:

Graphical user interface, application

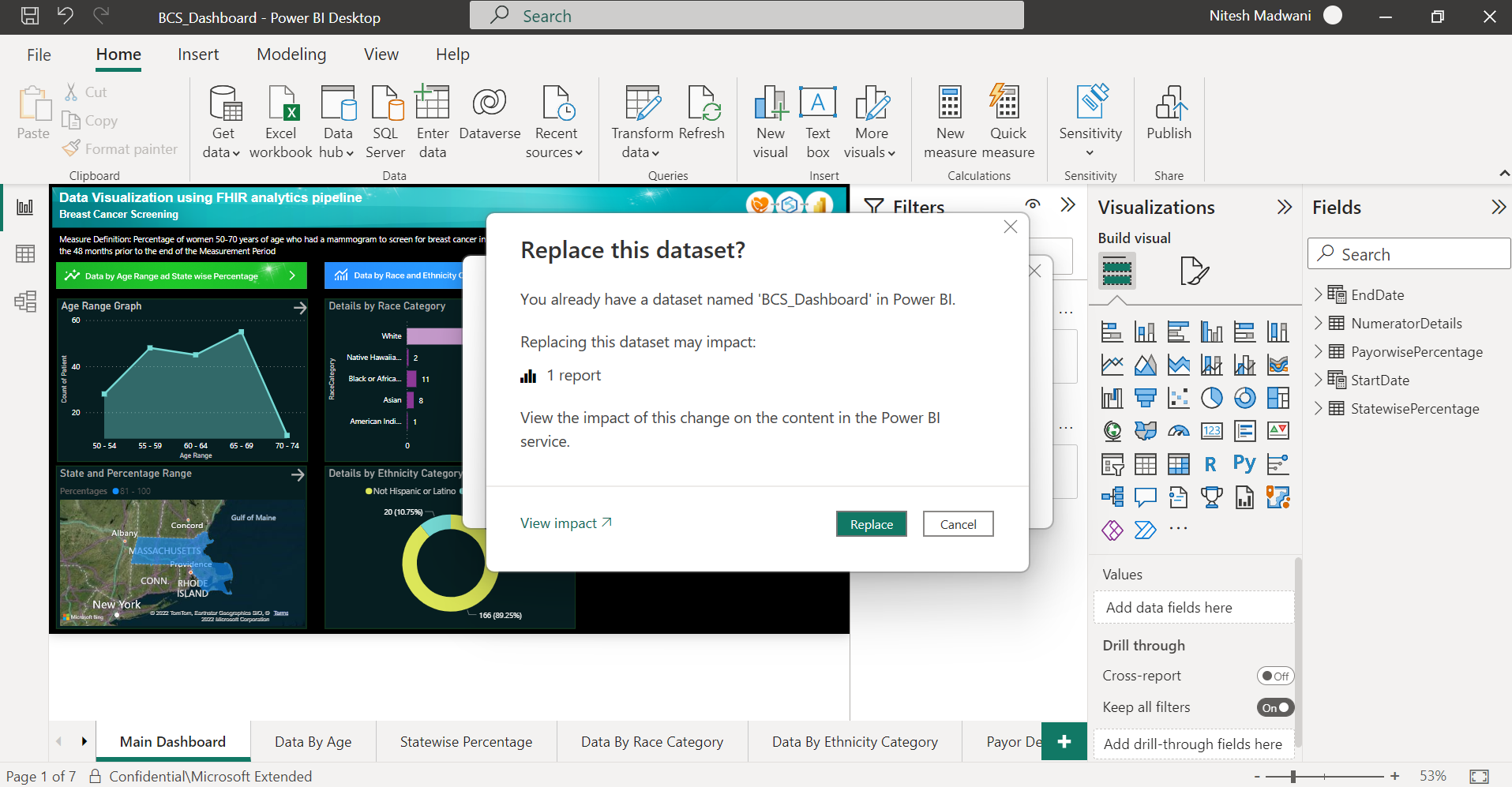
Description automatically generated

1. It will ask for the workspace in power BI service where you want to publish it, Select the specific workspace from dropdown list and click “Select”, in our case workspace name is “Quality Measure”, as shown in below:

Graphical user interface, text, application

Description automatically generated

1. If it is a fresh report, it will get published. In case you already have the report in workspace with same name, it will ask for confirmation to replace existing one. If you wish to replace the existing one click on replace, otherwise click cancel and save file with different name and follow steps again.



1. Once the report is published successfully, it will show pop-up with success message, click on “Got it” as below

Graphical user interface, text, application

Description automatically generated

## **View the dashboard in Power BI service**

1. Login to Power BI service and select your workspace in left-hand side pane, under workspaces scroll down and select your report from “Reports” section, once you select the report, it will open, and you can see all the pages listed in “Pages” as highlighted in below image:

Graphical user interface, website

Description automatically generated

1. The arrow sign button at top right corner of each graph in main dashboard will navigate to detailed page as shown below:

Graphical user interface, website

Description automatically generated

1. Each detailed page has a navigation button which navigates back to Main Dashboard page as shown below:

Graphical user interface

Description automatically generated