TASK-1 CREATING DATAFLOW

1. Create a data stream named CleaningGenreRomance and perform data cleansing on the Genre column using Derived Column and case expression. (While collecting data it was observed that some genres have spelling mistakes like romance, Romence for Romance, comedy, Comdy for Comedy.)

STEPS

Before creating dataflow, resource group needs to be created.

1. Creating Resource Group named **ETLAssignment** as shown in ss.

Graphical user interface, text, application, email

Description automatically generated

1. Creating DataFactory named **ETLDataFactoryWorkspace** for starting pipeline

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

1. Creating ADLS storage account for storing input data

Graphical user interface, text, application

Description automatically generated

Enabling hierarchical namespace for creating gen2 account

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, email

Description automatically generated

1. Creating (container) inputdataflow and (folder) input.

Uploading input data movies.csv inside it(inputdataflow/input)

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Once Storage Account created and stored. Create a data flow and add source by creating new dataset ADLS gen2 with delimited text format and using linked service to connect with etladlsgen2 storage account for input data

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Connecting with new linked service with below information

A screenshot of a computer

Description automatically generated

Graphical user interface, application

Description automatically generated

Under Source settings, below options are chosen Graphical user interface, text, application, email

Description automatically generated

The input data location is selected in wildcard path under source option

Graphical user interface, text, application

Description automatically generated

The datatype and format of each column should be mentioned accurately in projection

A screenshot of a computer

Description automatically generated

Graphical user interface, text, email

Description automatically generated

The input data can be viewed from Data Preview if needed.

Graphical user interface, text, application, email

Description automatically generated

* Adding another pipeline from sourceMovie named **derivedColumn1**

The below condition is used in settings for getting cleansed data

Case(Genre ==’romance’, ‘Romance’, Genre == ‘Romence’, ‘Romance’, Genre == ‘Comdy’ , ‘Comedy’)

Graphical user interface, application

Description automatically generated

The output of derivedColumn1 will be cleansed data with 8 column. It can be viewed in datapreview

A screenshot of a computer

Description automatically generated

* Storing output of derivedColumn1 in sink with output stream name **CleaningGenreRomance**

Graphical user interface, application

Description automatically generated

2. Create a data stream named CountMoviesBasedOnGenre that can calculate number of films for each genre and store it as a separate dataset in ADLS under folder name “solution/genreCount”

Graphical user interface, application

Description automatically generated

1. Cloning derivedColumn1 from previous one and considering that cleansed data as a input for counting number of films

Graphical user interface, text, application, email

Description automatically generated

1. Adding pipeline named Aggregate1 which will calculate the number of films for each genre.

Under group by => genre is selected

Under Aggregate => column name is film and expression is count(film).

1. For storing the result in separate dataset in ADLS , creating ADLS gen1 account named **etladlshen2**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Creating new linked service with storage account name etladlsgen1

Graphical user interface, text, application, email

Description automatically generated

To mention the destination path for gen1 account Dataset > ADLSGen1 > under connection mention file path as shown below

Graphical user interface, text

Description automatically generated

Creating a sink with outputStream name **CountMovieBasedonGenre**

Graphical user interface, application

Description automatically generated

Selecting a destination dataset ADLSgen1

Under settings, filename options is Output to single file and the name is MovieCount

Graphical user interface, text, application, email

Description automatically generated

======================

3. Create a new stream named JoinMovieCountWithCleanData. Perform join operation on CountMoviesBasedOnGenre with CleaningGenreRomance stream and store the same in the Azure SQL Database.

Performing join operation where left stream is CleaningGenreRomance and right stream is aggregate1

Join type is InnerJoin and condition is CleaningGenreRomance@Genre == genre

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

After Performing Join operation, output will have two genre column. To remove duplicate column, adding another pipe named **RemoveColumn** where in derived column settings, the output stream name is MapDrifted1, incoming stream is RemoveColumn1

Condition is represented as **toString(byName(‘Genre’))**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Creating a sink with output stream name JoinMovieCountWithCleanData, incoming is RemoveColumn1

The output should be stored in SQL Database, so need to create new SQL db and server and use it in linked service

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Creating SQL DB named **ETLDB** and creating new server for this resource group.

Server name – etladmin

Pass – Visi@2022

Graphical user interface, application

Description automatically generated

Configuring as shown below

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

After deployment gets completed, adding client IP in Firewall settings for connecting with Studio

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Connecting server with studio for opening Database in studio and to create table

Below Query is used for creating table Movie\_Clean\_Data

Graphical user interface, text, application

Description automatically generated

For Using in Linked Service, need to give YES in Allow Azure service and resource to access server

Graphical user interface, text, application

Description automatically generated

Using the newly creating server information in linked service and mentioning Database name

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Under sink dataset, create a destination sql table name

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated