**4 – Data flows: Code-free transformation**

**Pre-requisite: Create Datasets**

Download this CSV file and upload to your Azure storage account folder as we’ll use it for these labs on data flows: <https://github.com/kromerm/adfdataflowdocs/blob/master/sampledata/moviesDB2.csv>

Source Dataset:

1. Create an ADLS Gen2 Linked Service and a Delimited Text dataset
2. Name the dataset as “MoviesCSV”
3. Point to the CSV file above and set it to be comma delimited, include header on first row
4. Create a new data flow and point to the MoviesCSV dataset from step 2 above in the source transformation
5. Turn on your debug session
6. Go to the Source Project tab and Detect Data Types
7. Once you have your projection set, you can continue to the 2 scenarios below

Sink Dataset:

1. Using that same Linked Service from above, create a new delimited text dataset
2. Do not set a filename, just point a folder in your storage account, or type in a new folder name for ADF to create on the fly
3. Do not import any schemas
4. Call the dataset “folderout”

**Scenario 1: Big Data Analytics**

Objectives: Find the average rating of movies in each year in the dataset for any move that is categorized as a comedy. Then find the highest and lowest rated movie (across all genres) for each year. Last, filter all of the original source rows to show only your favorite year from the movies source data. Add a new column to this stream called “Favorite” as a Boolean and set it to true. Do this all in a single data flow. Sink the results of each separate stream into an ADLS Gen2 folder (use the folderout dataset from the sink dataset above). Make sure to format your outputs in a way that will be easy to read for business users.

Hints:

1. Use the Aggregate transformation for functions like min, max, and avg
2. You can use a Filter or avgIf() for the conditional average problem
3. Aggregates modify the outgoing schema and condense the stream down to just the columns used in that transformation. Therefore, you’ll need multiple streams and multiple sinks (outputs).
4. You can either add multiple Source transformations or use New Branch to add more streams from the same source.

Page Break

Each solution will look slightly different. Here is an example of a valid solution:

Timeline

Description automatically generated with low confidence

<https://github.com/kromerm/adfdataflowdocs/blob/master/sampledata/MovieAnalytics.zip>

**Scenario 2: Update changing data in the Lake**

Objectives: Take the MoviesCSV dataset source from above, create a new Delta Lake from it and then build the logic to updated rating for 1988 movies to 1, delete all movies from 1950 and insert new movies for 2021 by duplicating the movies from 1960.

Hints:

1. Create a source for the MoviesCSV, set the data types, and sink it directly into an Inline dataset type of Delta
2. Delta is an inline dataset type. You will need to point to your ADLS Gen2 storage account.
3. You’ll need to execute just that simple loading data flow from a debug pipeline first to form the Delta Lake
4. Next, you can create another data flow that will update/insert into the Lake
5. Use the MoviesCSV again as a source and detect data types
6. Only allow movie rows that match the 3 years you are going to work with which will be 1950, 1988, and 1960. Use a Filter transformation directly after your Source for this.
7. Update ratings for each 1988 movie to 1 in a Derived Column.
8. In that same Derived Column, create movies for 2021 by taking an existing year and change the year to 2021. Let’s pick 1960.
9. Your logic should look something like this in your Derived Column:

Graphical user interface, application, table, email

Description automatically generated

1. Update / insert / delete policies are created in the Alter Row transform. Add an Alter Row after your Derived Column.
2. Your Alter Row policies should look like this:
3. Graphical user interface

   Description automatically generated
4. Now that you’ve set the proper policy for each Alter Row type, check that the proper update rules have been set on the Sink:
5. Graphical user interface, application

   Description automatically generated
6. Here we are using the Delta Lake sink to your ADLS Gen2 data lake and allowing inserts, updates, deletes.
7. Note that the Key Columns is a composite key made up of the Movie PK and Year. This is because we created fake 2021 movies by duplicating the 1960 rows. This avoids collisions when looking up the existing rows by providing uniqueness.

Here is a sample solution for the Delta pipeline with a data flow for update/delete rows in the Lake: <https://github.com/kromerm/adfdataflowdocs/blob/master/sampledata/DeltaPipeline.zip>