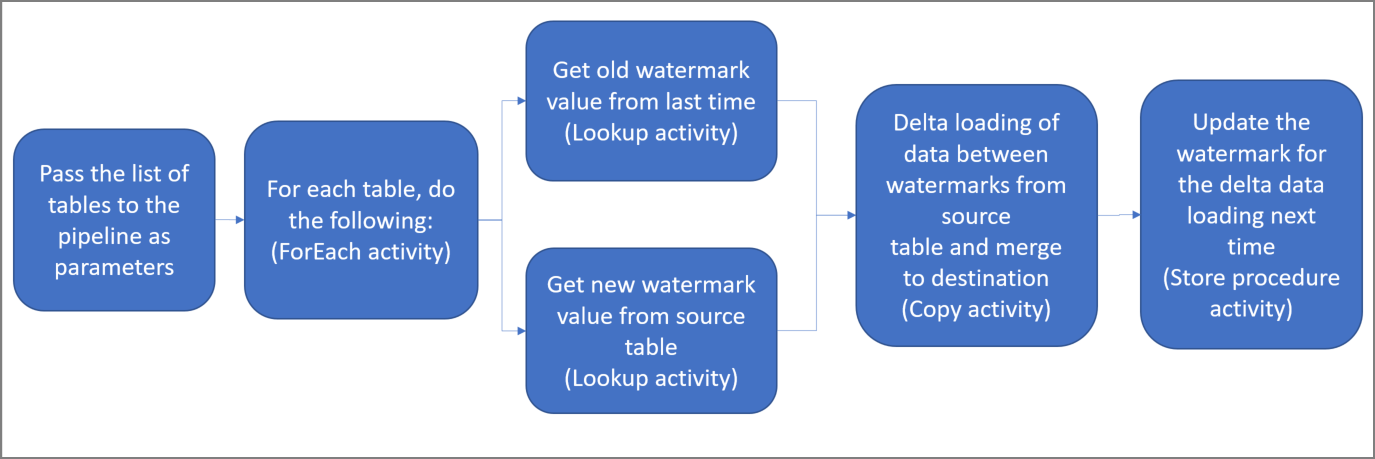
**Data Loading Scenarios**

1) Scenario1 – We have large data in a table in database and cannot perform data load on full data every time as it might take too long.

For this scenario, we can look into incremental data load approach but it also depends on the question Does input data has fields defined for watermark column (last modified time for specifying when each row of table was modified last). The approach to be followed in this is very similar to the one shown in diagram below



2) Scenario2 – We can perform data loading on full table data each time – In this full table data copy would be performed for each table

The above two ways could be mixed and matched depending on different data source types. For table with low data (in size), full loading could be done and for tables with large volumes of data incremental loading of data could be done.

**Configuration table**

Configuration table would be referred for checking configs related to incremental copy of data for each type of data source

Schema-

* 1. Source (varchar) - specifies the type of data source. For instance, specifying sql or filesystem for defining files for copying.
  2. Source\_table (varchar) - will specify name of table/ name of file
  3. Dest\_table (varchar) - specify name of the export file.
  4. Columns (varchar)- \* for all columns (number of columns to copy – default value \*)
  5. Watermark\_Column (varchar) – column name for the watermark column in the table
  6. Watermark\_Value (datetime) - The column value to be matched for checking which lines to load. Default value for this field will be very old date, for instance 1st jan 1800 so that for first time incremental loading will result in fully copying data.
  7. Enabled (bit) – to control enabling or disabling of copy data operation on specific table(s)
  8. Load Flag (varchar)– specifies if the full data or incremental data is to be loaded for the specified data source
  9. Status (varchar)– to update the status of the current data source.

Logging files for storing metadata information

A csv log file to be created in metadata directory for each data source. It will contain metadata information for the data source in the pipeline run.

Schema for log file -

data source, data destination, count of rows copied (for relational data), data loading duration and loading start and end time

Azure Data Factory pipeline implementation details

* Single pipeline for all data inputs (could be separated if needed)
* Initially it will load configurations for each DataSource (relational or file based). Configurations would have list of details of each of the different DataSources.
  + Configurations could be saved on a Azure SQL Database table (Basic DTU version of Azure SQL costs 5$/month for 2GB) or configs could also be saved on a csv file and be loaded from it as an alternative
* Steps of pipeline completion could be saved to a SQL database table (azure sql db cost will add as mentioned previously, alternative way would be with a csv file in adls). Logs and Metadata details for pipeline run can be saved in ADLS in separate metadata directory.

Implementation of Activity details for pipeline-

1. Lookup Activity (it will query config file and load all of the rows for defined data sources)
2. Two Filter activities to split
   1. SQL Filter activity - to filter the sql data sources and run pipeline on those sources
   2. ZIP files filter activity - Second filter activity for loading zip files
   3. For each activity to be performed for each SQL data sources -
      * Check if flag is “incremental”

If above condition True, perform: -

* + - * Get current watermark value from config table.
      * Retrieve records for modified date > watermark value
      * Load the incremental records to data lake
      * Get maximum date modified time from the file loaded
      * Update config table watermark value with the maximum date modified
      * Update config table status column to successful.

If above condition False, perform: -

* + - * Copy all the rows from source to data lake
  1. For each activity for zip files (in progress)

Azure Data Lake Storage details

* Different blob containers can be created in same storage account for Staging, Processed and Curated data for easy separation.