FME UC 2022:
Advanced Attribute and List Handling Exercises

Resources:

Sample data and exercises

- `C:\FMEData2022\Resources\FMEUC22\Advanced Attribute and List Handling\Workshop Exercises"`

POP TEST: Date & Time

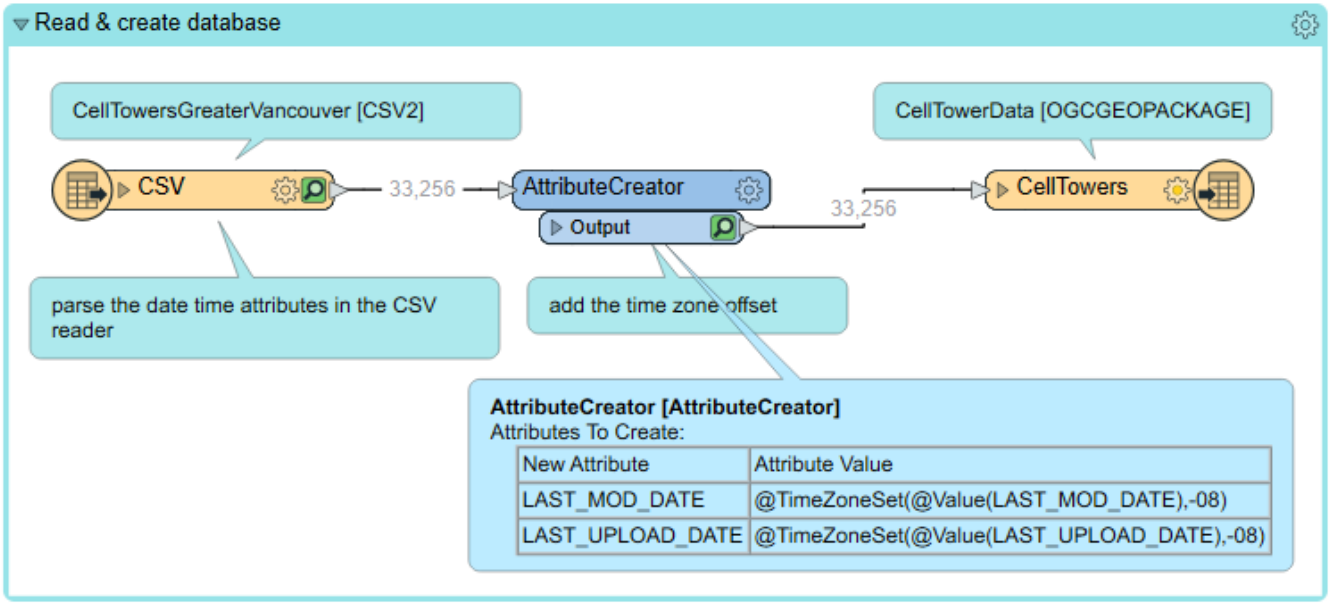
Quick Exercise - 3min:

- read the CSV cell signal data. **CellTowersGreaterVancouver.csv**
- parse the two date fields (**LAST_MOD_DATE**, **LAST_UPLOAD_DATE**) into FME date/time format
  - Hint: %d-%b-%Y %H:%M:%S under the CSV Reader Parameters -> Schema Generation -> Date Input Format
- set this as a Preset on the CSV Reader so you can use this configuration again tomorrow
- set a UTC offset (-08:00) using a datetime function, i.e. @TimeZoneSet() to the two date fields
- write it all to GeoPackage

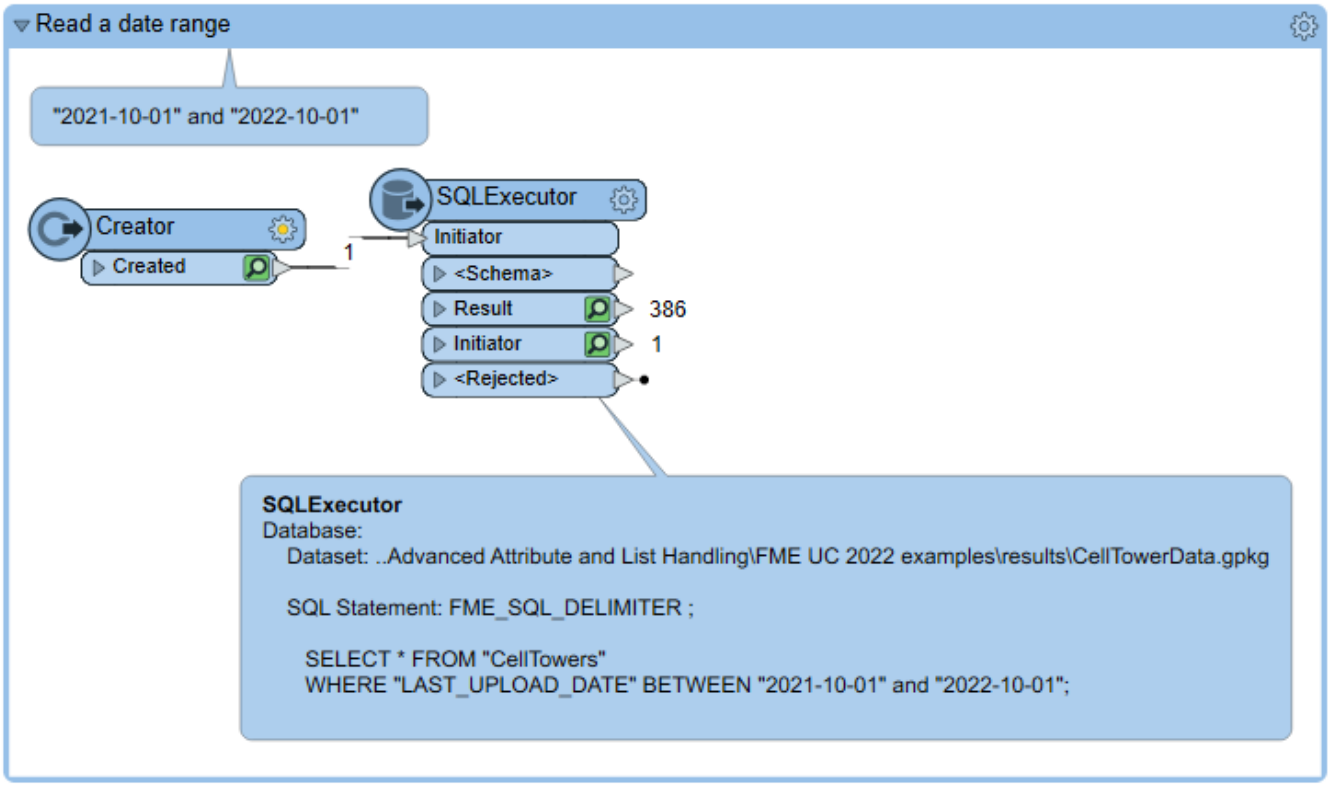
Bonus:

- In the same workspace: use a SQLExecutor to query the Geopackage for a date range - say between "2021-10-01" and "2022-10-01".
  - **Hint:** that we have to use the SQLite date/time format for this query not the FME format SELECT * FROM "CellTowers" WHERE "LAST_UPLOAD_DATE" BETWEEN "2021-10-01" and "2022-10-01";
- **Hint**: have a look at the Geopackage date formats FME Data Inspector and DB Browser and compare the formats

Your workspace will look something like this:



And the bonus SQL:



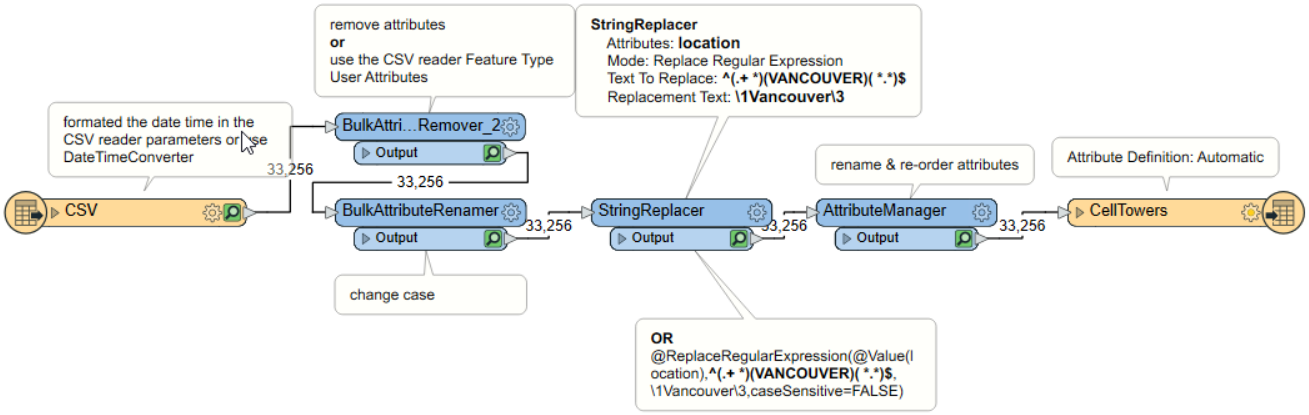
EXERCISE: Attribute Handling

Scenario: CSV cell tower data that requires some data clean up before writing to a Geopackage.

- CSV Reader: Read the cell tower data: **CellTowersGreaterVancouver.csv**
  ensure dates are converted to datetime data type in GeoPackage
  - **Hint**: two ways you can do this
    - CSV reader Schema Generation or
    - DateTimeConverter.
  - Source format is: 30-Mar-2019 22:31:01
  - **Hint**: here's the date formatting string: %d-%b-%Y %H:%M:%S
- OGC GeoPackage writer
  - use Table Definition: Automatic.
  - Choose an appropriate table name and geometry type.
  - Reproject coordinates to UTM84-10N for the GeoPackage output
- Remove *ALL* the attributes starting with TX_
  - **Hint**: Bulk up or use the CSV reader feature type parameters
- Convert *ALL* the attribute names to lowercase.
  - **Hint**: Bulk-up
- In the LOCATION attribute, *Vancouver* has been included as "VANCOUVER"
  - Use regex to replace these with "Vancouver"
    - **Hint**: try **StringReplacer** or @ReplaceRegularExpression()
    - **Hint**: regex example: `^(.+ *)(VANCOUVER)( *.*)$`
      … use https://regexr.com/ to find out what this means
    - Replacement Text: `\1Vancouver\3`
- rename NEW_ACCOUNT & NEW_LICNO -> account and licenceno
- reorder the output attributes: recordid, account, licenceno, licensee, location, prov, …

Pop Quiz: Why the WARNs on the GeoPackage writer?

You workspace looks something like this:



EXERCISE: List Attribute Handling

Scenario: We need to do a quick analysis of the CSV cell tower data for our service technicians.

A. at each location (LOCATION), create a list of antenna models (TX_ANT_MODEL) and a count of how many different models on each tower. Then the service technicians know which parts to bring and how many.

B. We need to identify towers where the antenna manufacturer (TX_MFR & TX_ANT_MFR) is unknown but is using excess “effective isotropic radiated power (EIRP)” (TX_PWR) at each location (LOCATION)
   - manufacturer is unknown if the MFR code is “ZZ”

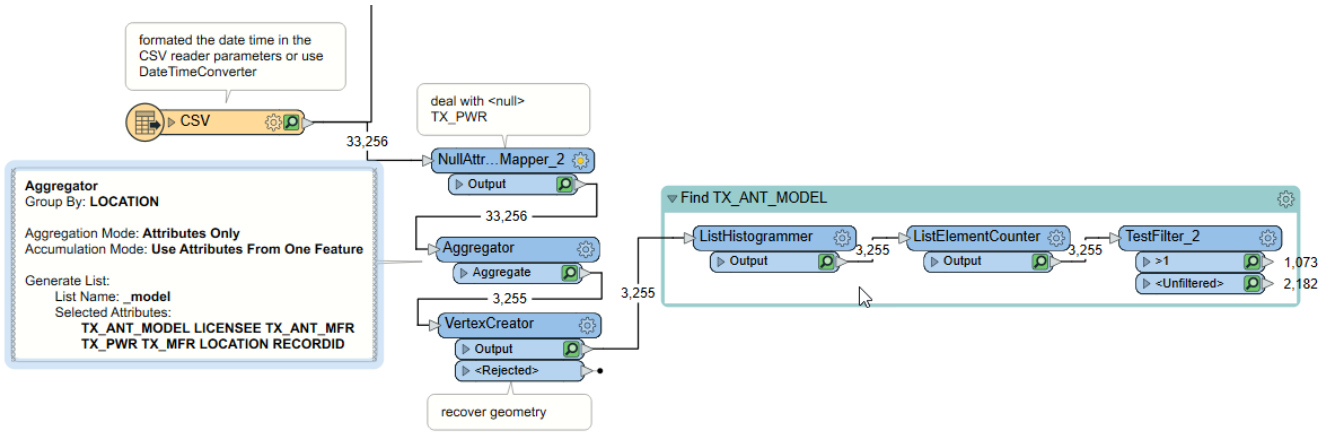
Using the workspace from the previous exercise *or* start with
“Exercise2-Begin-ListAttributeHandling.fmw” (has some additional hints)

- Aggregate the cell towers by **LOCATION** - Aggregator transformer
  - Use Attribute Accumulation to preserve attributes
  - Generate list attributes:
    - Only some attributes are needed in the list:
      TX_ANT_MODEL TX_ANT_MFR TX_PWR TX_MFR LOCATION LICENSEE RECORDID
- Create a single point for each cell tower location
- Replace any <null> TX_PWR values with zero (0)

Part A: antenna models

- Create an attribute list of the unique TX_ANT_MODEL’s and the number of models at each tower LOCATION.
  - **Hint**: start with *ListHistogrammer*
  - Optional - create an Excel report - use ListExploder

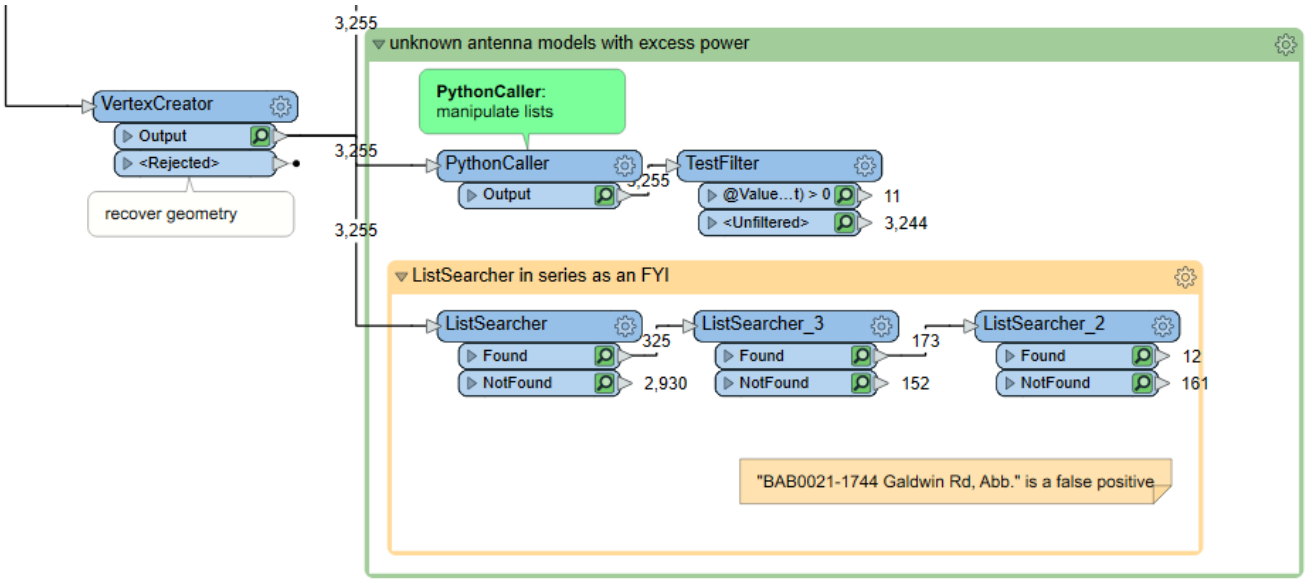
New additions to your workspace will look something like:



Part B: unknown manufacturer

- For each tower **LOCATION**, create a list of antenna models (TX_ANT_MODEL) where the antenna manufacture is unknown (TX_MFR **and** TX_ANT_MFR = ZZ) **and** the EIRP (TX_PWR) is greater than 5000
    - **Hint:** use multiple List Searchers *OR*
    - PythonCaller
      Hints in: “Exercise2-Begin-ListAttributeHandling.fmw” or below
- **Note**: the results of the PythonCaller are a list that is only visible in the Visual Preview Feature Information window…

Your part B additions look something like this:



Here’s an example of the Python:

```
import fme
import fmeobjects

# create a list of LOCATIONS where the antenna manufacturer is unknown
# and the  effective isotropic radiated power (EIRP) >= 50000

def processFeature(feature):
    # initialize the python lists with FME lists
    _TX_MODEL = feature.getAttribute('_model{}.TX_ANT_MODEL')
    _TX_ANT_MFR = feature.getAttribute('_model{}.TX_ANT_MFR')
    _TX_MFR = feature.getAttribute('_model{}.TX_MFR')
    _TX_PWR = feature.getAttribute('_model{}.TX_PWR')
    _tx_model_unknown = []
    _tx_count = 0

    # test for "MFR" and PWR
    for i in range(len(_TX_PWR)) :
        if  ( _TX_MFR[i] == "ZZ" and _TX_ANT_MFR[i] == "ZZ" and
float(_TX_PWR[i]) >= 5000.0 ):
            if (_TX_MODEL[i] not in _tx_model_unknown ) :
```

```
            _tx_model_unknown.append(_TX_MODEL[i])
        _tx_count = _tx_count + 1

# output the FME lists
feature.setAttribute('_ant{}.model', _tx_model_unknown)
feature.setAttribute('_ant_count', _tx_count)
```