Thanks for checking out the SQL Join tool! If you don’t yet have the macro downloaded, you can find it here: https://community.alteryx.com/t5/Community-Gallery/SQL-Join/ta-p/1204659

Hard requirements:
Unfortunately, due to the build of this tool, there are a couple of mandatory requirements for use. Namely:

- Version 2023.1 (or later)
- An existing, working In-DB connection

Background:
At the moment, the native Join tool in Alteryx Designer only allows users to conduct very strict, like-for-like joins i.e.:

- Table1.A = Table2.A
- Table1.B = Table2.B

If we want to do more advanced operations that mimic those we can perform in SQL, such as:

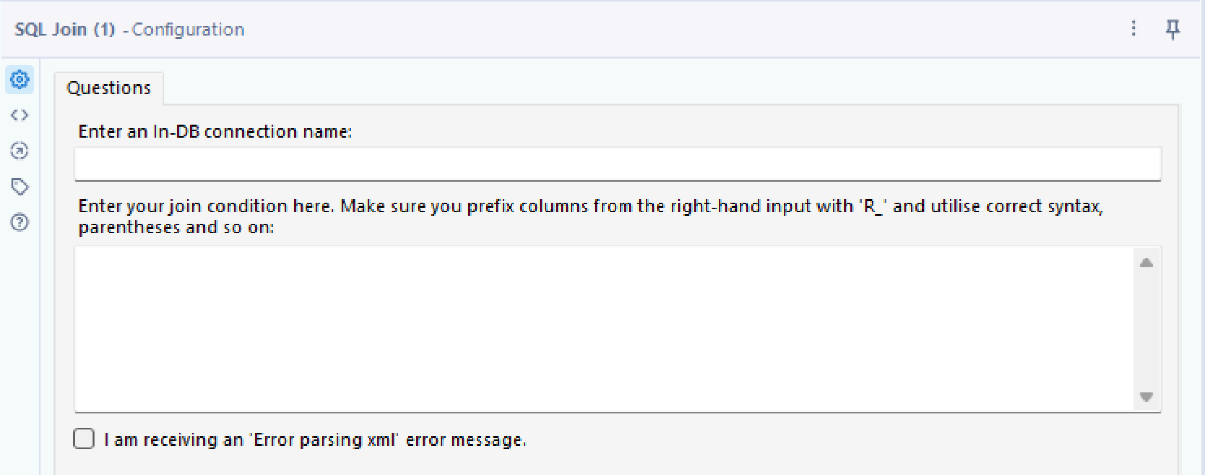
- Table1.A <= Table2.A
- Table1.A BETWEEN Table2.A AND Table2.B
- Table1.A LIKE Table2.A+’%’

Then we need to start blowing up our data by means of appending records (which gives a cartesian product), or creating a scaffold that gives us range values to conduct a like-for-like join on. In both of these cases, the result is an explosion of data which, if avoidable, is bad practice and can lead to a degradation in performance.

Purpose:
The purpose of this tool is to enable users to conduct flexible, SQL-like joins - like those mentioned above - within Alteryx Designer. Users can provide any join condition that satisfies the syntax and operability of the database to which they are connecting and the macro will return the left, inner and outer join products of their two tables.

How does it work?
There will be a full blog detailing the exact workings of each step coming soon. However, in short, the tool utilises an established In-DB connection provided by the user to write two temporary tables. A Text Box then takes end-user input (the join condition) and combines this with parsed temp table names in order to compile a custom SQL query, which is then passed to a Dynamic Input In-DB tool. After this is processed, data is streamed back into the underlying workflow before being sorted and used to create left & right join products against the initial input.

Using the SQL Join tool:
Upon placing the macro in their workflow, users will be presented with the following configuration:



SQL Join (1) - Configuration

Questions

Enter an In-DB connection name:

Enter your join condition here. Make sure you prefix columns from the right-hand input with 'R_' and utilise correct syntax, parentheses and so on:

□ I am receiving an 'Error parsing xml' error message.

There are 3 aspects to this:

1) **Enter an In-DB connection name:**
    a) Type in the name of an existing connection (**case sensitive**). These connections can be managed by navigating through ‘Options’ > ‘Advanced Options’ > ‘Manage In-DB Connections’
2) **Enter your join condition here:**
    a) Type out the join condition here. This needs to adhere to the syntax of whichever database you are connecting to from your In-DB connection.
    b) **Important:** Fields coming from the right-hand table need to be prefixed with ‘R_’ when typing out the join condition i.e. “Product” = “R_Product”. This is to prevent errors caused by ambiguous field names.
3) **I am receiving an ‘Error parsing xml’ error message:**
    a) When running the macro, if users come up against this type of error message, they can select the Check Box and try again. Some drivers do not properly cleanse inputs and therefore special XML characters like < and > need to be escaped at times. For example, during testing, “A” <= “B” did not work when querying Snowflake and needed to be written as “A” &lt;= “B”.