**2-1– In Database**

**In-database processing** enables blending and analysis against large sets of data without moving the data out of a database. This allows users to apply ETL and calculations inside the source system and utilize its own resources. In contrast, a standard input tool that sources from a database will download all of the specified input data to a temporary file on the local machine. If the table consists of millions or even billions of records, the download process can take a long time especially when experiencing high network latency issues. Performing analysis in the database can save processing time. Using Designer as the interface, in-database workflows integrate easily with standard workflows for additional data blending and analysis.

In-database processing requires 64-bit Alteryx with 64-bit database drivers. The following sources can support the In-Database function:

* Amazon Redshift
* Apache Spark ODBC
* Cloudera Impala
* Databricks
* EXASOL
* Hive
* IBM Netezza
* Microsoft Analytics Platform System
* Microsoft Azure SQL Database
* Microsoft Azure SQL Data Warehouse
* Microsoft SQL Server 2008, 2012, 2014, 2016
* MySQL
* Oracle
* Pivotal Greenplum
* PostgreSQL
* SAP Hana
* Snowflake
* Teradata
* Vertica

The In-Database Tools in designer allows users to visually build their SQL query which is processed in the database. These tools also use the SQL syntax for creating filters and formulas and does not work with Alteryx specific functions. The first step in creating an in-database process is to manage its connection. This is possible by using the Connect In-DB Tool. **Connect In-DB is used to create an in-database connection in a workflow. This tool is used to either connect to a new or existing connection.**

From the In-Database Palette, drag Connect In-DB and place it into the workflow. To create a connection, click the dropdown under connection name and select “manage connections”. If you already have a saved connection, you can use the other option to “Open File Location”. In the new window for managing in-db connections, the first configuration is to set the Data Source. The list of data sources available on the dropdown are same as the list that we’ve shown earlier. If your data source is not on the list, you can attempt to create a connection using the “Generic ODBC” option. This does not guarantee a successful connection to unsupported sources, but if your data source is similar to Microsoft SQL Server, it has a better chance of success. For this connection, we are going to connect to our local MySQL database, so we will choose “MySQL” on the list. Next is the connection type. There are 3 possible selections. For this local designer workflow, only 2 options are available. “User” for creating a connection that only you can use. “File” which saves a database connection as a .indbc file so it can be packaged with a workflow. The last option “System” is only visible if you have alteryx designer as an administrator. System creates a connection that can be shared. We are going to use the workflow for a local connection only so we are going to choose “User” as its connection type. Next is to set the connection. We do not have an existing connection in-DB so click the button “New” to enable the rest of the configuration. Under Connection Name type “MySQL in-DB”. After naming the connection, set the password encryption. **There are 4 options; “Hide” will hide the password with minimal encryption. This is the best option to use if you are going to schedule this workflow to run on any machine. “Encrypt for Machine” will make it possible for any users on this computer to fully use the workflow. “Encrypt for User” will let the logged in user use the workflow on any computer, and finally “Allow Decryption of Password” decrypts the password and passes it in the metadata. This option is only used in conjunction with In-DB predictive tools.** Select “Encrypt for Machine” and proceed to the read tab. The driver is automatically set to MySQL ODBC since we set the data source prior. Next is to set the connection string. Click the dropdown and select “New database connection”. We’ve already set up a MySQL data source called “MYSQL-indb” which connects to the sakila schema by default, but you can also create your own clicking ODBC admin and clicking add to create a new source. Once the ODBC connection is ready, we need to type in the User Name and Password then hit OK to finish the connection string.

For the write tab, we are going to retain the default “Same as Read Driver” so it can use the same connection string. If you want to use a different connection string, simply select the other option, in this case “MySQL ODBC” to let you create a different connection but for the same data source type. The default transaction size for writing database is also set to 10k by default so might need to increase this if your workflow will handle a lot more records. Once all connection settings are done, click apply and OK. A new window for the query builder should open up if the connection was built successfully. This window is similar to the input data query builder where in the schemas are listed on the right and you can apply sort, aggregations and filters using the bottom window. For this example, we are going to use the sakila default mysql schema and its customer table. Double click on the schema “sakila” then double click on the table “customer”. We are going to use all columns so tick the \* to select all. When you check the script that was automatically created under the tab “SQL Editor” it should have the query “select sakila.customer.\* from sakila.customer”. Then close the window to show this expression on the Connect in-Db tool’s query box. If you need to apply additional changes, simply click the query builder button to open the visual query editor window. Once we run the workflow, you will not see any records on the results window since we simply created a connection and all of the action were done on the backend or on the database.

If you wanted to show the records from the in-db process, we need to insert a Browse In-Db tool. Unlike an ordinary alteryx workflow, in-DB workflows do not show its processed records on its anchor and is not compatible with the regular browse tool. **Browse in-DB** is used to view your data at any point in an In-DB workflow. Use the Browse In-DB tool as you build an In-DB workflow to ensure data is passing through the way you intend it to. It allows you to view results at any point in the In-DB workflow. But do take note that each Browse In-DB tool triggers a database query and can impact performance so you need to use it sparingly.

Drag a Browse In-DB tool and connect it to our first tool. Notice how the In-DB tools differ as compared to other tools in alteryx designer. First, it used a blue square database anchor to connect to another in database tool, unlike other tools such as input data and browse tool which uses a green arrow anchor. 2nd is the connection line. In-DB tools uses a double line to display its connection. 3rd, due to the nature of in-database processing, the connection progress is not displayed. Since they are of different processes, you cannot simply connect an In-DB tool to other alteryx tools. We will discuss more on this topic on the next lesson.

Let’s configure the browse in-db tool. First you need to specify the number of records to display when the workflow is run. By default, it is set to 100, but you can set a different number of records such as 50 or 1000. Entering 0 will display the maximum number of records which can be up to 2 billion. If there are more than 2 billion records in the data stream, there will be a message in the Data View statistics. We will leave this to the default 100 and proceed with the next setting. If you check “Enable Caching” the tool will cache the number of records returned when the workflow is run. If you did not change the number of records in its browse first number of records for the next run, it will simply display the data in the cache which saves processing time. You can also clear the cache by clicking the “Clear Cache” button. Run the workflow to show the first 100 records from the in-database connection to the customer table. Now we can view the first 100 records. Same with that of the browse tool, you can export this records to a file by clicking the actions button, select export > all records and set a filename and type on the new window.