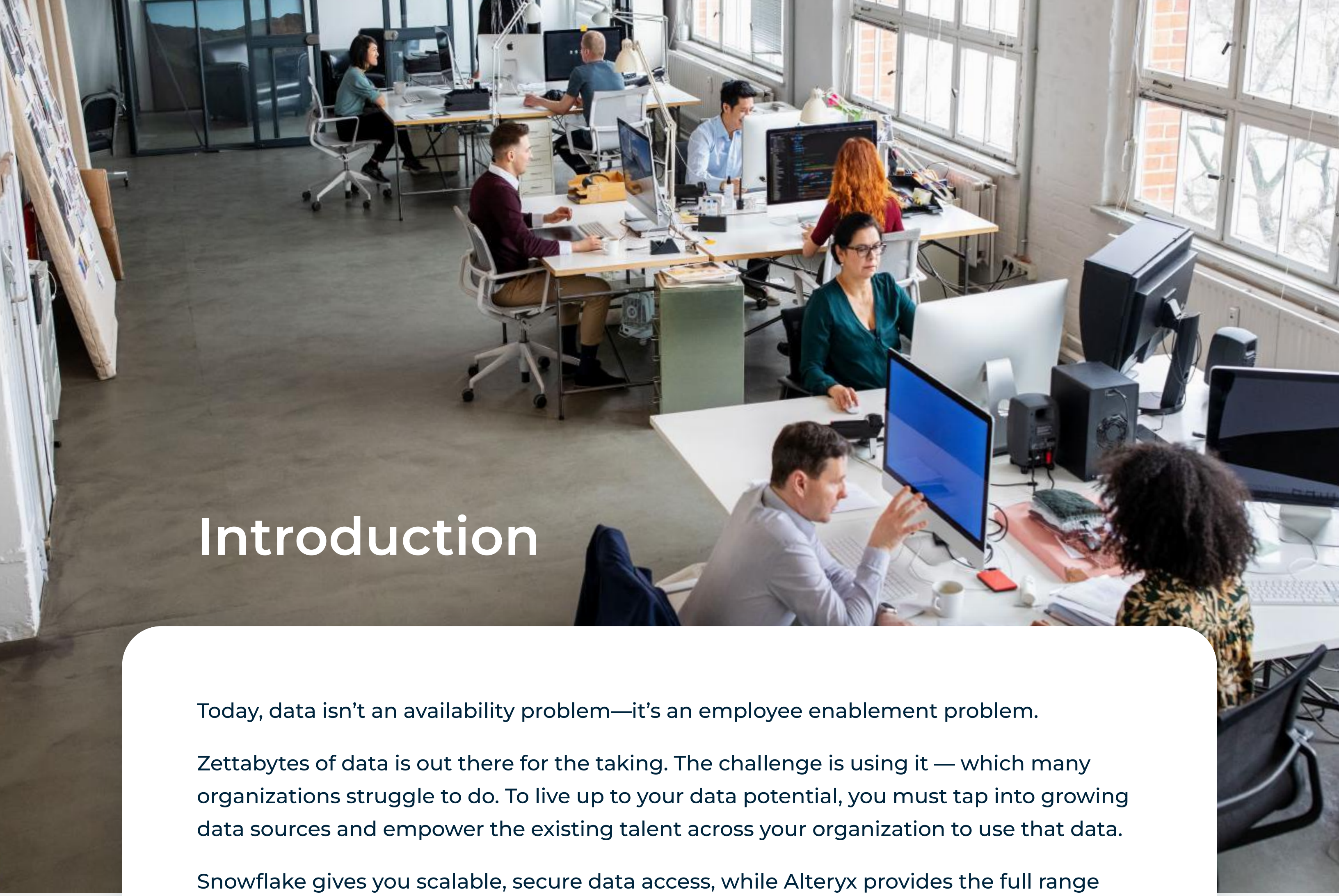
Best Practices Guide for Snowflake + Alteryx

Designer and Server

Table of Contents



Introduction

Today, data isn't an availability problem—it's an employee enablement problem.

Zettabytes of data is out there for the taking. The challenge is using it — which many organizations struggle to do. To live up to your data potential, you must tap into growing data sources and empower the existing talent across your organization to use that data.

Snowflake gives you scalable, secure data access, while Alteryx provides the full range of self-service analytics capabilities with the Alteryx Analytics Cloud Platform. Together, they speed up analytics, data engineering, and data science by making it easy for anyone in your organization to quickly prep and transform data for business insights.

This whitepaper will help you achieve your full potential by getting the most out of the capabilities unique to both Snowflake and Alteryx. This guide will focus on a deeper dive into Alteryx Designer and Server specifically.

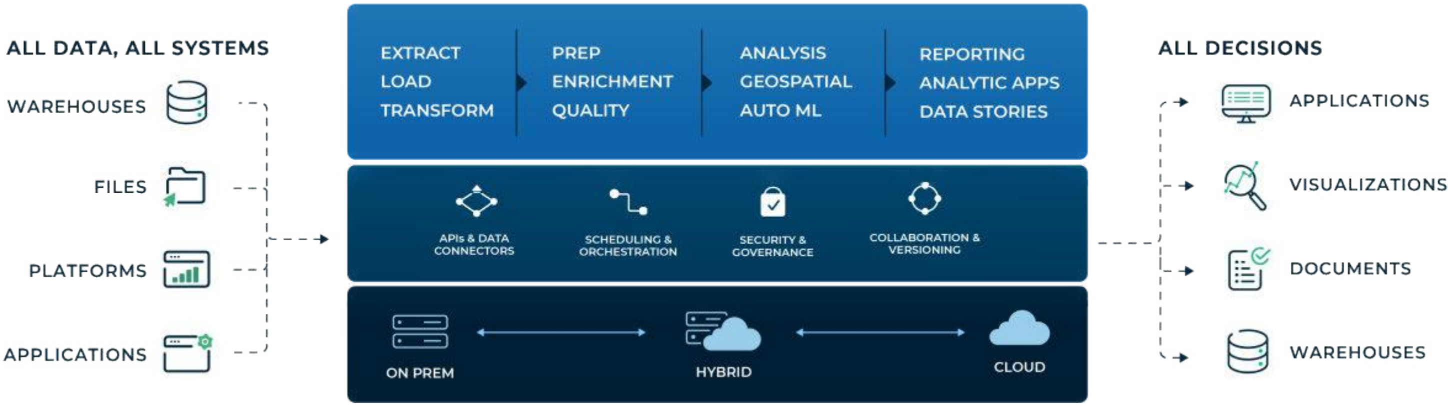
Topics covered will include:

- Key features of Alteryx and Snowflake and how to combine them for added value
- How to take advantage of pushdown processing in both Designer and Designer Cloud to leverage Snowflake's scalable compute resources
- Insights on developing your analytic workloads to take advantage of the unique capabilities of each platform with a focus on Alteryx Designer

What is the Alteryx Analytics Cloud Platform?

The Alteryx Analytics Cloud Platform supports an insight-driven organization with unified solutions from automated data preparation & analytics, approachable machine learning, and AI-generated insights. The Alteryx Analytics Cloud Platform delivers the right analytic capability to each user and allows collaboration across teams to support end-to-end enterprise analytics solutions.

alteryx Analytics Cloud Platform



What is Snowflake?

The Snowflake Data Cloud consists of two parts – platform and data.

The Data Cloud is a network that connects Snowflake customers, partners, data providers, and data service providers. It enables more organizations to derive value from sharing rapidly growing data sets with each other in secure, governed, compliant, and seamless ways to enable teams to act faster and smarter. In the Data Cloud, organizations can avoid copying or moving data. Instead, a data provider shares access to live, read-only data with any number of data consumers.

The Data Cloud is powered by Snowflake's platform and it enables customers to execute a number of critical workloads, including data engineering, data warehousing, data lakes, data science, data sharing, and building and operating data applications. Snowflake's platform scales instantly and near-infinitely and is a single, near-zero maintenance platform, as a service offered across all public clouds. By offering dedicated compute resources that are provisioned within seconds, there is no resource contention between users and workloads and you only pay for what you use.

What are Snowflake's Core Workloads?

Data Lakes

Using Snowflake as either a standalone data lake or as a means to augment an existing one, the platform delivers the best value in the market for storage, transformations, and data warehousing to serve all business needs.

Data Engineering

Snowflake streamlines data engineering, delivering performance and simplicity so organizations can focus on getting value from their data instead of managing the pipelines and infrastructure.

Data Sharing

Snowflake Data Marketplace enables instant, frictionless, secure sharing of live data within and between organizations. Unlike traditional data sharing methods such as email, FTP, cloud storage (Amazon S3, Box), and APIs, Snowflake eliminates data movement, does not require the data consumer to reconstruct data (ETL), and provides direct access to live data in a secure environment. Snowflake Data Marketplace allows companies to grant instant access to ready-to-use data to any number of data consumers without any data movement, copying, or complex pipelines.

Data Warehousing

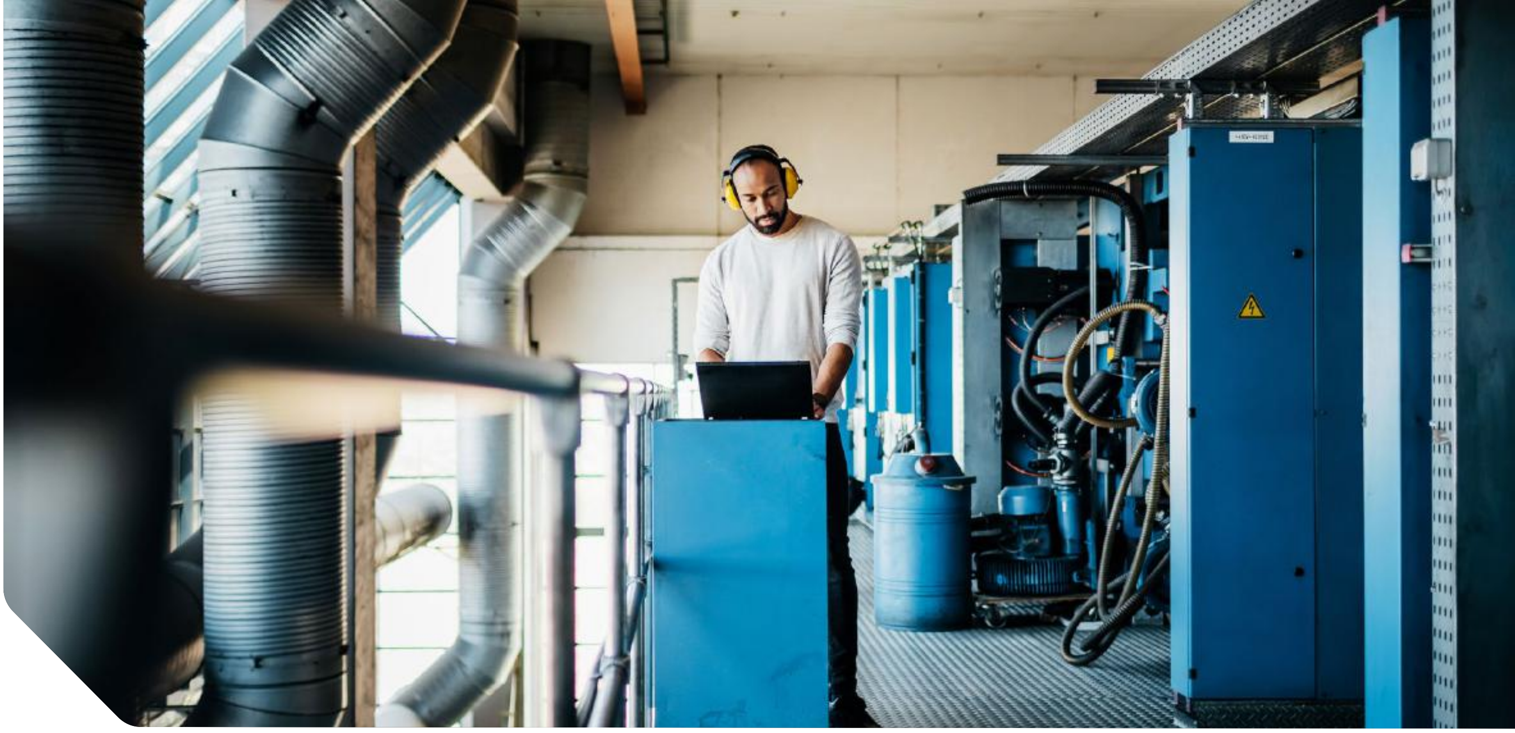
Snowflake's support for data warehousing and analytics provides a low-maintenance, cost-effective way for organizations to consolidate all their data silos into a single source of truth they can query to get quick results. By providing consistently fast queries, more users analyze more data and collaborate with their peers.

Data Science

Snowflake helps data scientists operate quickly and efficiently by providing a centralized source of high-performance data to a robust ecosystem of data science partners that handle modeling and training algorithms. Partner-provided output is fed back into Snowflake where it's easily accessible to technical and non-technical users.

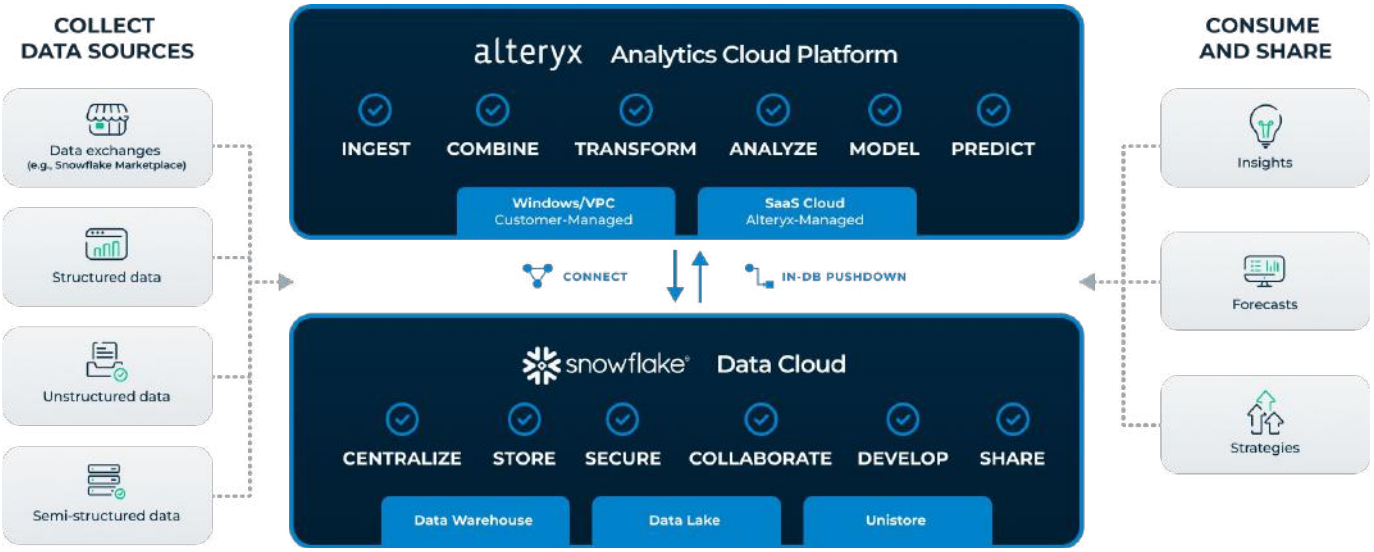
Data Applications

Snowflake provides a unique architecture that enables the development of modern applications without managing complex data infrastructure. Because Snowflake is a fully managed data platform with features such as high concurrency, scalability, automatic scaling, and support for ANSI SQL, developers can quickly deliver data applications that are fast and scalable.



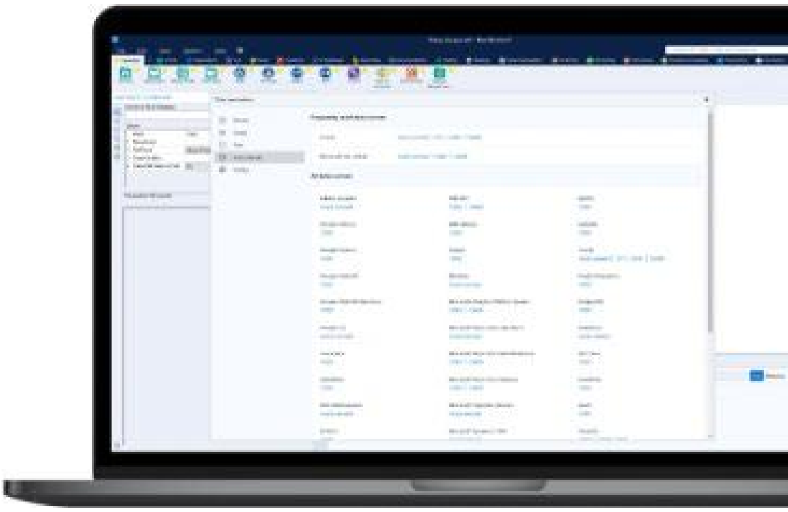
Benefits of Using Snowflake + Alteryx

With Snowflake and Alteryx, you get a unified stack that empowers everyone to access the full data pipeline quickly and securely. With its approachable drag-and-drop building blocks, Alteryx enables self-service analytics for anyone in the organization who needs it. At the same time, Snowflake enables self-service access to data. Below are some of the specific ways Alteryx combines with Snowflake to unlock value at each step of the data analytics process.



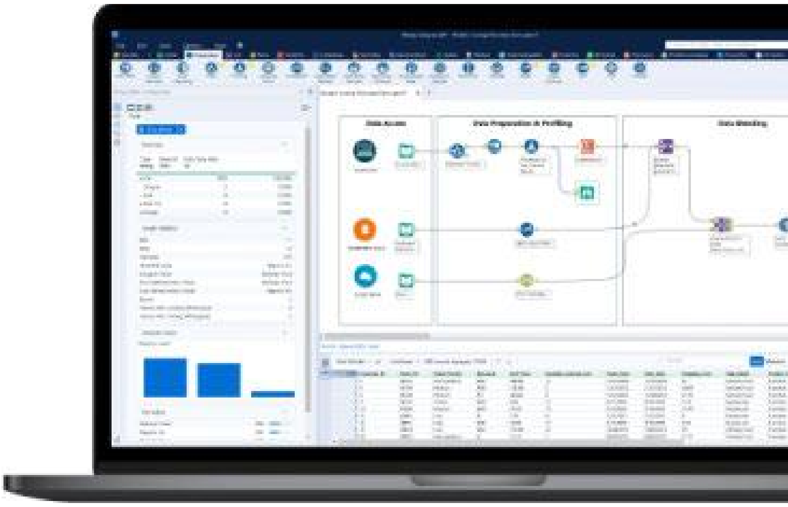
Combine Disparate Data Sources

Explore and connect data from on-prem databases, the cloud, big or small data sets and more. Alteryx features 80+ natively integrated data sources including Snowflake. Securely connect to an unlimited number of additional sources in Designer and the Alteryx Analytics Cloud Platform. If you can access the data, you can bring it into Alteryx and spend more time analyzing and less time searching.



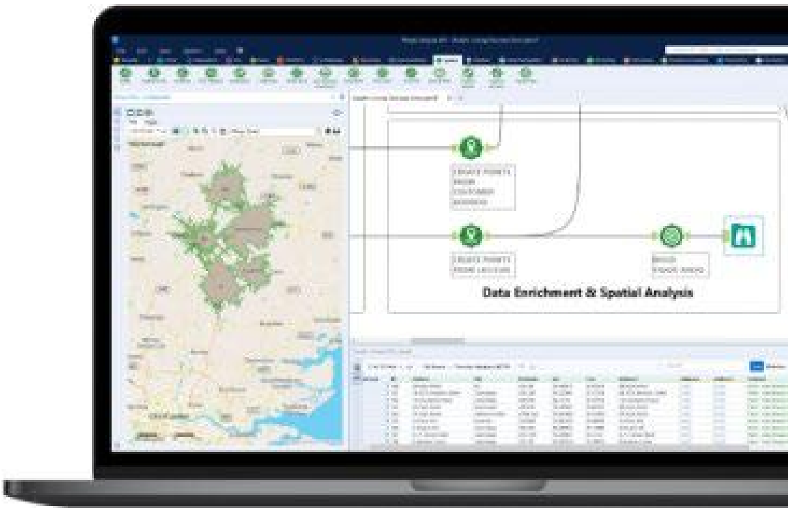
Transform Big Data with No-code Pushdown in Snowflake

Easily cleanse, prepare, and blend data from a variety of different sources with or without unique identifiers to deliver unified data profiles. Leverage the power of the cloud with in-database pushdown processing in Snowflake that turns big data into big insights.



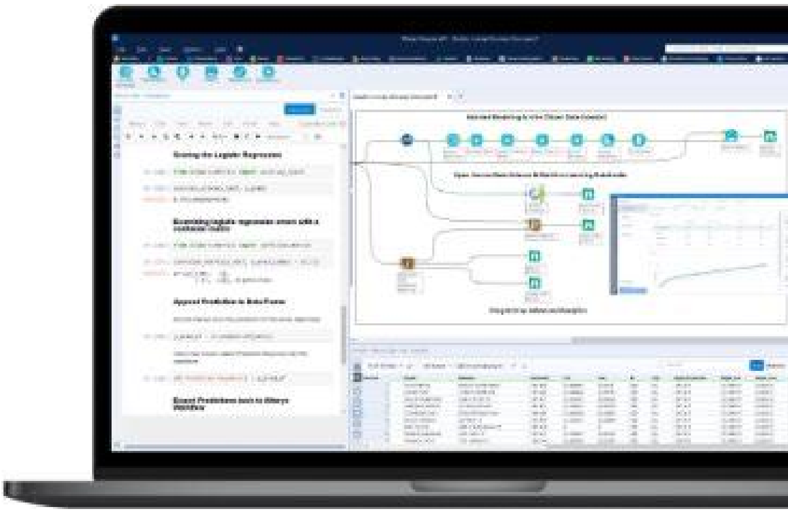
Drive Richer Insights with Spatial Analysis

Go beyond standard demographic data with behavioral and retail purchasing information to create an in-depth view and truly understand consumers. Enrich your analysis with maps, address solutions, drive time capabilities, and a deeper understanding of your customers and locations — because everything happens somewhere. Use the geospatial capabilities in Designer and Alteryx Location Intelligence to make location-based decisions with Snowflake data.


CREATE POINTS FROM CUSTOMER ADDRESS
CREATE POINTS FROM LOCATION
BUILD TRADE AREAS
Data Enrichment & Spatial Analysis

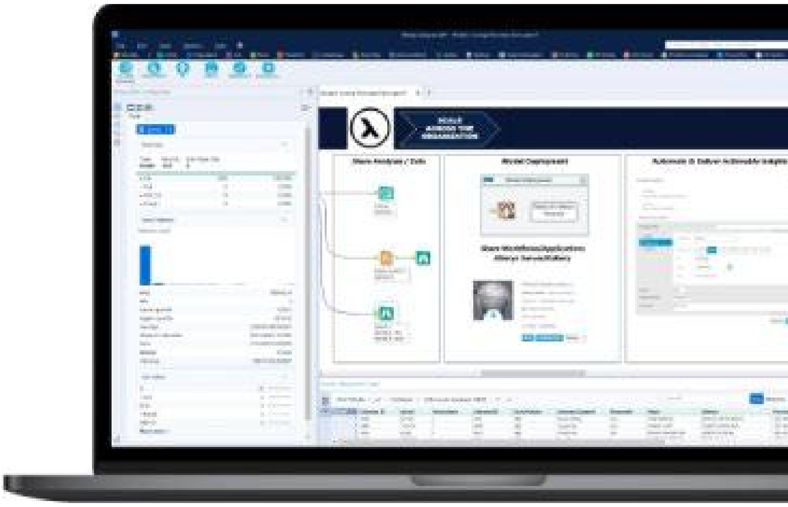
Make Intelligent, AI-powered Decisions

Upskill your team’s analytic output with step-by-step guides and assisted modeling to build models without coding or analytics expertise. Gain insights and better answers by using data for complex analytical methods, from sentiment analysis on unstructured data to building complex R-based models with little-to-no coding skills. Activate the intelligence layer with advanced analytics, such as Alteryx Machine Learning and Alteryx Auto Insights, to provide richer insights. Snowflake users can also take advantage of Alteryx AiDIN, the AI engine that infuses the power of generative AI & Machine Learning (ML) across the Alteryx Analytics Cloud Platform.



Automate Outcomes

Empower others to customize any analysis and make decisions with analytic apps. Share in a variety of formats such as writing back to the database or leveraging spreadsheets or an easy-to-consume report. Build once and automate forever. Effectively communicate answers and share them with stakeholders so they can take action, visualize outputs, or even create lightweight dashboards in Alteryx. Amplify human output and enable the perpetual upskilling of people with intelligent decisioning to deliver faster, better outcomes.


SCALE ACROSS THE ORGANIZATION
Share Analysis / Data
Model Deployment
Automate & Deliver Actionable Insights
Share Workflows/Applications
Alteryx Server/Gallery

Alteryx transforms business outcomes and workforces by quickly automating analytics and entire data-driven business processes that would otherwise require a myriad of building blocks and manual handoffs. The result: quick wins and fast ROI that scales across your organization.

Optimizing Your Alteryx Workflow with Snowflake

Alteryx gives you options for how and where you want to work with your data, depending on your business needs and data stack. You can choose to leverage the Alteryx AMP engine to process your data, and you can execute workflows via in-database pushdown in Snowflake. Alteryx also offers code-friendly capabilities with Snowflake, including Snowpark. Here's a rundown.

Standard Designer Workflow

Standard workflows provide a breadth of capabilities but do not leverage the computational power of an underlying data source. Used with Snowflake, standard building blocks would query Snowflake into Alteryx at runtime, then process, prepare and blend data used in sophisticated geospatial, data science, or advanced analytical models. These workflows could be automated with Alteryx Server or Alteryx Analytics Hub or designed into an Analytic App – empowering others to customize analysis and make decisions.

Standard building blocks are easy to use and work well for most use cases, but you will gain speed and execution efficiency by processing your Snowflake workloads with In-Database building blocks. Optimized workflows take advantage of In-Database and Orchestration techniques to place the processing as close to Snowflake as possible. Later sections will expand on these techniques.

Keep the following in mind when designing your standard workflows

- ✓ Use data reducing queries when using standard building blocks to reduce the volume of data loaded into memory.
- ✓ Use a SAMPLE SQL clause to cut down on data size for an initial analysis.

Pushdown Execution in Snowflake

While standard workflows are ubiquitous in the Alteryx community, they do not take full advantage of Snowflake's compute power and ability to scale elastically. That's where pushdown processing comes in. To help you leverage the Snowflake engine, Alteryx supports pushdown processing in Snowflake in multiple ways:

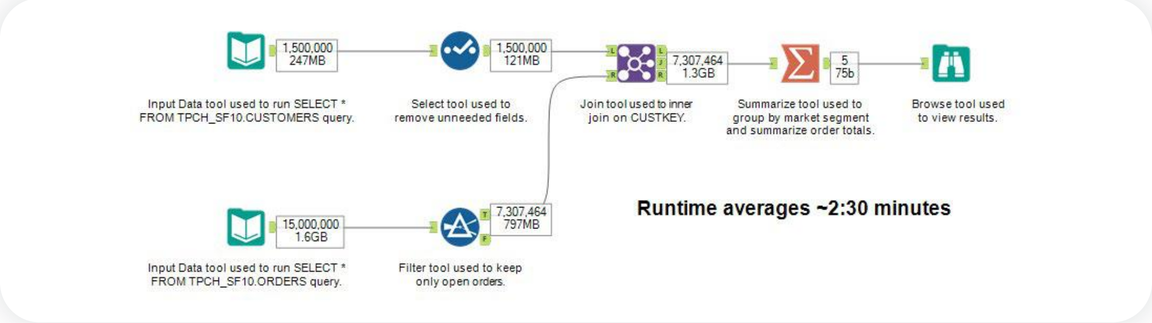
- **In-database:** Execute SQL pushdown with no-code tools in Designer
- **Automatic pushdown in Designer Cloud:** When connected to Snowflake in Designer Cloud, the workflow automatically executes in Snowflake.
- **Snowpark:** Define custom analytic building blocks in Alteryx that push processing directly to Snowflake

In-database Tools in Designer

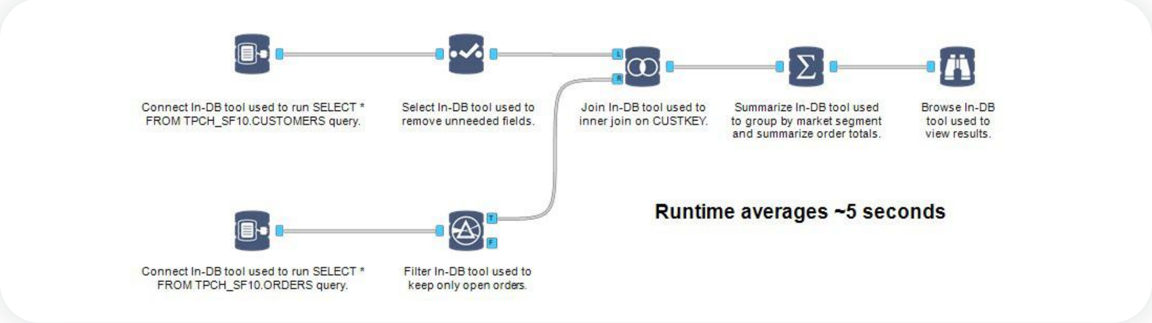
Alteryx In-Database building blocks make it easy to supercharge your analytics workloads by unlocking the near-unlimited scale, concurrency, and performance of Snowflake. In-Database building blocks push down the execution of SQL queries into your Snowflake Virtual Warehouse - eliminating loading your queried data into system memory. In-Database provides an effortless way to analyze large data sets without needing to limit or sample data.

The performance benefits of using In-Database vs. standard building blocks can be significant. For example, look at the performance difference between the following standard and In-DB workflows, which both produce the same outcome:

STANDARD WORKFLOW


1,500,000
247MB
Input Data tool used to run SELECT * FROM TPCH_SF10.CUSTOMERS query.
1,500,000
121MB
Select tool used to remove unneeded fields.
7,307,464
1.3GB
Join tool used to inner join on CUSTKEY.
5
75b
Summarize tool used to group by market segment and summarize order totals.
Browse tool used to view results.
15,000,000
1.6GB
Input Data tool used to run SELECT * FROM TPCH_SF10.ORDERS query.
7,307,464
797MB
Filter tool used to keep only open orders.
Runtime averages ~2:30 minutes

IN-DATABASE WORKFLOW


Connect In-DB tool used to run SELECT * FROM TPCH_SF10.CUSTOMERS query.
Select In-DB tool used to remove unneeded fields.
Join In-DB tool used to inner join on CUSTKEY.
Summarize In-DB tool used to group by market segment and summarize order totals.
Browse In-DB tool used to view results.
Connect In-DB tool used to run SELECT * FROM TPCH_SF10.ORDERS query.
Filter In-DB tool used to keep only open orders.
Runtime averages ~5 seconds

Before detailing the different In-Database building blocks, let us discuss a few best practices:

- Leverage In-Database building blocks early and often. Look to move prep and blend processes as early in your workflow as possible and execute them with In-Database building blocks.
- If your workflow does require standard building blocks, have the data set groomed, filtered, and prepped to minimize the data transfer query to Alteryx.
- Consider streaming the data back into Snowflake for final processing and storage.

Review this table to understand where an In-Database tool can be used in place of a standard tool.

IN-DB TOOL	TOOL DESCRIPTION	STANDARD TOOL ANALOG	CONSIDERATIONS
Browse In-DB	Review your data at any point in an In-DB stream. Each In-DB Browse Triggers a DB query and can impact overall workflow performance.	**Browse**	Use the Browse In-DB tool generously to see what is happening within the workflow. Take all the Browse Building blocks out before committing the workflow to production.
Connect In-DB	Establish a database connection for an In-DB stream.	**Input Data**	You can use pre and post SQL statements.
Data Stream In	Bring data from a standard workflow into an In-DB workflow.	**Output Data**	You can create a temporary or permanent table for the data.
Data Stream Out	Stream data from an In-DB stream to a standard workflow, with an option to sort the records.	**Input Data**	Easy to use tool. You do have the ability to sort the records before streaming out.
Dynamic Input In-DB	Take an In-DB Connection Name and Query fields from a standard data stream and input them into an In-DB data stream.	**None**	Reference Dynamic Select with In-DB Building Blocks article in the Alteryx Community for an example use case.
Dynamic Output In-DB	Output information about the In-DB workflow to a standard workflow.	**None**	Outputs metadata, such as the query and connection name, to the workflow. It will not allow you to output connection string information unless you select the "Allow Decryption of Password" option in Manage In-DB Connections. Performs fastest when filtering on dimensional filters where the micro partitions are clustered. Reference In-DB Query and the Dynamic Output tool for additional guidance.
Filter In-DB	Filter In-DB records with a Basic filter or with a Custom expression using the Snowflake SQL.	**Filter**	In-DB and standard building blocks have a similar configuration interface.
Formula In-DB	Create or update fields in an In-DB data stream with an expression using Snowflake SQL.	**Formula**	Some formulas in the Standard Formula tool may translate to different building blocks In-DB. For example, adding a field with a constant value can be placed in custom SQL in the connect In-DB tool

IN-DB TOOL	TOOL DESCRIPTION	STANDARD TOOL ANALOG	CONSIDERATIONS
Join In-DB	Combine two In-DB streams based on common fields by performing an INNER JOIN or OUTER JOIN.	**Join** **Join Multiple**	Behaves more like a database join than an Alteryx Join or Join Multiple tool. It can be configured to execute Inner Joins as well as Left Right and Full Outer Joins. Use as you would a SQL JOIN clause. Performance is better on dimensions that are cluster keys. For ultimate performance, move this join to the database as a Materialized View.
Macro Input In-DB	Create an In-DB input connection on a macro and populate it with placeholder values.	**Macro Input**	In-DB and standard tool have a similar configuration interface.
Macro Output In-DB	Create an In-DB output connection on a macro.	**Macro Output**	In-DB and standard tool have a similar configuration interface.
Sample In-DB	Limit the In-DB data stream to a number of records.	**Sample**	Will add a LIMIT “N” clause. Use when prototyping on large data sets to decrease processing time. Add Sort by any dimension that is being clustered to improve partition pruning.
Select In-DB	Select, deselect, reorder, and rename fields in an In-DB stream.	**Select**	Allows you to cut down on fields and rename them. Slightly different purpose than the select tool. Users that are looking to replicate the in-memory select tool’s ability to change data types, create a formula tool with a CAST or a TO_<datatype> statement.
Summarize In-DB	Summarize In-DB data by grouping, summing, counting and more. The output contains the result of the calculation.	**Summarize**	In-DB and standard tool have a similar configuration interface.
Union In-DB	Combine two or more In-DB data streams with similar structures based on field names or positions. In the output, each column will contain the data from each input.	**Union**	In-DB and standard tool have a similar configuration interface.
Write Data In-DB	Use an In-DB data stream to create or update a table directly in the database.	**Output Data**	More focused than the output data tool since its intent is to take In-DB data and write to a table in the same database

Using Code-Friendly Approaches with Snowflake

The capabilities of Alteryx are not limited to processing data in Snowflake or Alteryx. Some of the building blocks within Alteryx are code-friendly and enable users to leverage and manipulate other systems. This gives you the advantage of executing processes in the optimal system or location. Also, Alteryx gives you the ability to encapsulate this external process into a macro that can be called a single tool in future workflows.

Related to working with Snowflake, a few orchestration methods can be accessed through Python and CLI building blocks.

SnowSQL

SnowSQL is Snowflake's next-generation command line client for connecting to Snowflake to execute SQL queries and perform all DDL and DML operations, including loading and unloading data from database tables. It can be run as an interactive shell or in batch mode. For our purposes, you will be using SnowSQL exclusively in batch mode.

Using the Alteryx Run Command tool, you can create a workflow that can execute a SnowSQL script, giving you the ability to execute anything that SnowSQL supports. Executed scripts can be static or pre-existing SQL. You can also use a Formula tool to dynamically create the script and write the file via an Output Data tool.

Snowflake Python Library

The Snowflake Connector for Python provides an interface for developing Python applications that can connect to Snowflake and perform all standard operations. It provides a programming alternative to developing applications in Java or C/C++ using the Snowflake JDBC or ODBC drivers.

The Alteryx Python tool, combined with the Snowflake Connector for Python, provides a flexible way to interact with your Snowflake instance and drive custom outcomes with the full scope of Python capabilities.

Hybrid Workflows

Most use cases will require you to implement more than one of the above techniques. The workflow may start with In-DB processing and stream the data out to Alteryx, where it gets joined again back in Snowflake. Also, branches of the same workflow can use Python and CLI building blocks. It can be a challenge to determine the best approach.

For guidance, use these rules of thumb for creating a workflow when working with Snowflake data:

- ✓ Push as much transformation and processing as possible to the beginning of the workflow using In-DB techniques.
- ✓ Stream data back into Snowflake after processing data. You can create a target data set in Alteryx, stream the data to a temp table, and then join the data in Snowflake for a faster method of analysis.
- ✓ Create Macros and In-DB Macros for complex data functions in Snowflake. You may have a sophisticated process that may not be easily created by all your users.
- ✓ Anything goes when prototyping. Just like writing a document, you are better off getting your analytical ideas into a workflow any way you find easy. After you have proved out the concept, you can optimize the workflow to use the above techniques.

Connecting to Snowflake in Designer

Connections and Alteryx Designer

The first step when connecting to Snowflake is to download and install a Snowflake driver. The Snowflake Simba driver is recommended and can be downloaded from the Alteryx Product Downloads site. Alternatively, a driver can be downloaded directly from Snowflake.

When connecting to Snowflake, the only supported method for authentication is username-password. Other authentication types like OAuth, External Auth, and key pair are not currently supported.

There are three ways a Snowflake connection can be configured within an Alteryx tool – DSN, DSN-less, and In-DB Connection Files.

Types

DATA SOURCE NAME (DSN)

A data source name (DSN) is used to detail the specifics of a data source connection. Configure a DSN on your machine using the following steps:

1. Launch the ODBC Data Sources Administration Tool by searching on your Windows machine for "ODBC Data Sources (64-bit)".
2. Within the ODBC Data Sources Administration Tool:
   a. Click on the Drivers tab and verify that either the Simba (displayed as Simba Snowflake ODBC Driver) or Snowflake driver (displayed as SnowflakeDSIIDriver) is installed. If the driver isn't shown, install the driver.
   b. Create a new DSN by clicking the Add button within the User DSN or System DSN tabs.
   c. Select either Simba Snowflake ODBC Driver or SnowflakeDSIIDriver.
   d. Enter the connection parameters for the driver. It is recommended to enter Data Source, Server, Database, Schema, Warehouse, and Role.
   e. Click OK to create the DSN.

The DSN you created can now be selected as the data source within the Input Data, Output Data, Connect In-DB, and Data Stream In building blocks.

DATA SOURCE NAME (DSN)-LESS

To avoid creating a DSN, you can use a DSN-less connection string. The connection string specifies the connection parameters. The Input Data, Output Data, Connect In-DB, and Data Stream In building blocks support this method. It does not work when writing data using the Snowflake Bulk Load option.

The connection string is composed as a single statement containing the following parameters:

```
odbc:Driver={Simba Snowflake ODBC Driver OR
SnowflakeDSIIDriver};
UID={USERNAME};
PWD={PASSWORD};
DATABASE={DB_NAME};
SCHEMA={SCHEMA_NAME};
SERVER={SERVER_ADDRESS};
WAREHOUSE={WAREHOUSE_NAME};
ROLE={ROLE_NAME}
```

When writing data, you can specify an output table by adding |||TABLE_NAME to the end of the connection string.

Additional guidance can be found in How To: Format Common ODBC DSN-less Connection Strings.

IN-DB CONNECTION FILES (.INDBC FILES)

An Alteryx In-DB connection (.indbc) file saves database connection information so that it can be packaged with a workflow. Create an .indbc file using the following steps:

1. In Alteryx Designer, navigate to Options > Advanced Options > Manage In-DB Connections.
2. Select Snowflake from the Data Source dropdown.
3. Select File from the Connection Type dropdown.
4. Click the ellipses (...) button next to the Connection File input field.
5. Navigate to the location you want to store the .indbc file.
6. Type in a name for the .indbc file in the File name: text box.
7. Click Open.
8. Configure the Read and Write connections as you normally would using the DSN or DSN-less instructions.

An .indbc file will be created when you click OK. This method is supported by the Connect In-DB tool.

Refer to How To: Create an Alteryx In-DB Connection File for additional information on creating .indbc files.

BEST PRACTICES

A few best practices to keep in mind when connecting to Snowflake.

1. Use the most recent Simba Snowflake ODBC Driver provided on Alteryx Product Downloads.
2. Define the Database, Warehouse, and Role when configuring your connection.
3. If any ODBC connection parameter (e.g., Proxy) needs to be defined, it can be added using the tool configuration dialogue.

CONNECTIONS AND ALTERYX SERVER

Considerations for running a workflow on Alteryx Server that utilizes a connection to Snowflake:

- Installation of a Snowflake driver is required on Alteryx Server and worker nodes.
- It is recommended to use either DSN-less or In-DB Connection files when saving and/or scheduling a workflow.
- When using DSN based connections, ensure that the DSN is also set up on Alteryx Server and worker nodes.

CONNECTIONS AND ALTERYX ANALYTICS HUB

Considerations for running a workflow on Alteryx Analytics Hub that utilizes a connection to Snowflake:

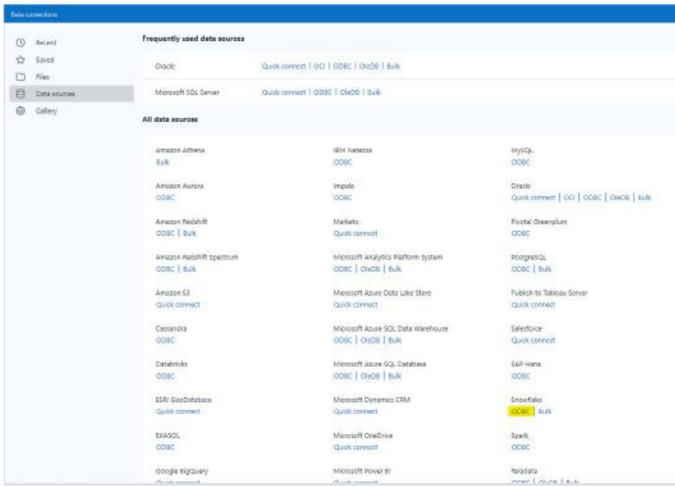
- Installation of a Snowflake driver is required on Alteryx Analytics Hub and worker nodes.
- In-DB Connection File is the only supported connection method.

Loading Data into Snowflake

Inserts

The baseline ODBC method for writing data is to insert a single record at a time. The Output Data and Connect In-DB (Write) building blocks support inserts. Limit your use of this method to times when you are working with small data sets or are otherwise unable to use the Bulk Loader.

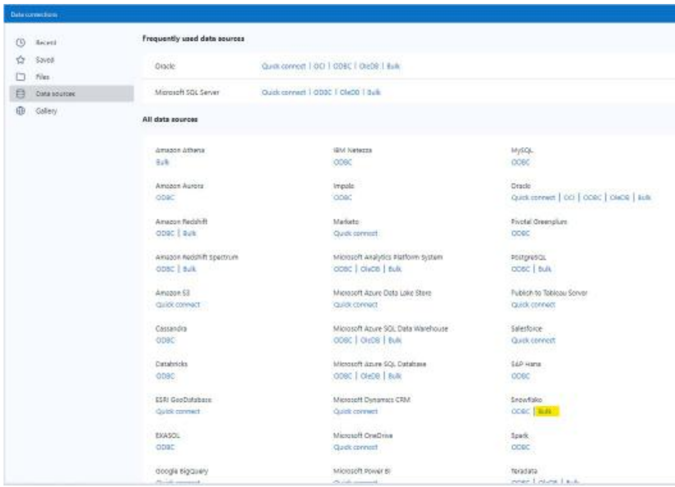
To use inserts to load data into Snowflake, select ODBC from the Data connections > Data sources menu within the Output Data tool configuration.



Bulk Loader

For most of your Snowflake workloads that include writing data to Snowflake, you should default to using the Snowflake Bulk Loader. This method loads batches of data from files in cloud storage or by copying data files from a local machine to an internal (i.e., Snowflake) cloud storage location before the data is loaded into tables using the COPY command. Alteryx supports all three internal stage types – user, table, and named. Alteryx also supports loading data from an Amazon S3 external stage.

To use bulk loading to load data into Snowflake, select Bulk from the Data connections > Data sources menu within the Output Data tool configuration.



Loading Big Datasets

When writing large datasets to Snowflake, a few things should be considered to improve the performance and resiliency of the process. Snowflake notes that to optimize bulk load performance,

> "The number of load operations that run in parallel cannot exceed the number of data files to be loaded. To optimize the number of parallel operations for a load, we recommend aiming to produce data files roughly 100-250 MB (or larger) in size compressed."

To improve the resiliency of your bulk load workflows you should implement a chunking strategy. Through a chunking strategy, if a bulk load fails, you will be able to retry loading on a chunk instead of running the entire load again. A method for doing this is to create a batch macro that manages the bulk load. The batch macro would be set up to ingest one chunk at a time and bulk load it to Snowflake using an Output Data tool.

For additional information reference Snowflake's Preparing Your Data Files.

Custom SQL

One of the big advantages of Alteryx is its code-free and code-friendly interface. Whether you can write either full SQL statements or snippets in some fashion gives you some abilities to do some special things.

The advantage of adding Custom SQL is you can simplify the workflow in instances where several Alteryx In-DB building blocks can be reduced down to some SQL in your Connect In-DB tool. Be aware that the downside to this approach is flexibility. Workflows that use In-DB building blocks can be easier to troubleshoot and repurpose. You can also create a more universal workflow by making it a macro or an app.

Here are locations where you can use Custom SQL within a workflow:

Input Data Tool

Table or Query – This text entry will allow you execute queries that will return a table for further processing in Alteryx. Companies that use Snowflake and Alteryx provided feedback, and said they find these query types useful:

- Time Travel – Add an AT or BEFORE clause to your query to enable Snowflake Time Travel. Reference the Snowflake documentation for query syntax details. Couple a time travel query with a standard query of the same table to create a workflow to analyze changes after an ETL job completes or over time.
- Where clauses, aggregations, and group by clauses – If your workflow requires you to take the data out of Snowflake, create your Input Data Tool query string to return only necessary data at the granularity required. This allows you to push the SQL processing to Snowflake while Alteryx does complex analyses and automation not achievable via SQL.
- Limit Clause – A LIMIT clause at the end of a SELECT statement is a quick way to speed up the workflow execution time for prototyping

Pre SQL & Post SQL Statements – This is a text box where you can have a SQL command execute before or after the input data query is executed. This statement can be used to resize the warehouse before the load is executed. By increasing or decreasing the size of the warehouse you can change the speed that queries are executed when pushed down to Snowflake.

Output Data Tool

Pre SQL & Post SQL Statements – This is a text box where you can have a SQL command execute before or after the output data query is executed. This statement can be used to resize the warehouse before the load is executed. By increasing or decreasing the size of the warehouse you can change the speed that queries are executed when pushed down to Snowflake.

Python Tool

This tool has the most flexibility in executing SQL commands to Snowflake. The Python tool does require you to import the Snowflake Connector module with the line:

IMPORT SNOWFLAKE.CONNECTOR
Since the capabilities are based upon the Python Library, use the Snowflake Connector for Python documentation as a resource to understand what can be achieved with this tool. There are a few interesting things that the Python tool can do within your workflow:

- Call Stored Procedures – use the .execute("call <<sprocname>>()")
- Change Warehouse sizes
- Set environment variables
- Asynchronous queries – use the "_no_results = True" after your query cursor.execute(data['querystring'], _no_results = True
- Special aggregations – not all aggregate functions are exposed in the Alteryx building blocks. Use SQL executed in Python to get aggregates like APPROX_COUNT_DISTINCT.

Connect In-DB Tool

Table or Query – This text entry will allow you execute queries that will return a table for further processing in Alteryx. Feedback from the field where companies are using Snowflake and Alteryx find these query types useful:

- Time Travel – Add an AT or BEFORE clause to your query to enable Snowflake Time Travel. Reference the Snowflake documentation for query syntax details. Couple a time travel query with a standard query of the same table to create a workflow to analyze changes after an ETL job completes or over time.
- Where clauses, aggregations, and group by clauses – If your workflow requires you to take the data out of Snowflake, create your Input Data Tool query string to return only necessary data at the grain required. This allows you to push the SQL processing to Snowflake while Alteryx does complex analyses and automation not achievable via SQL.
- Limit Clause – A LIMIT clause at the end of a SELECT statement is a quick way to speed up the workflow execution time for prototyping.

Formula In-DB Tool

The formula syntax is the difference between the regular Formula Tool and the Formula In-DB Tool. The Formula In-DB Tool looks more like SQL so the translation will be very straight forward for most calculations. Here is a short list of comparable calculations executed in Alteryx with a Formula Tool and Snowflake with a Formula In-DB Tool.

FORMULA	STANDARD FORMULA TOOL	IN-DB FORMULA TOOL FOR SNOWFLAKE
Create a column with a constant field	1	Select 1
Retrieve the left most character	Left([FIELD]),1)	select left("FIELD",1)
Strip currency characters from a string to create a numerical field ($ and + characters)	Replace(Replace ([CurrField],"$","") ,"+","")	select Cast(trim(split_part ("CurrField", '-',1),'$+') AS INT)
Conditional Statement	Switch([Field])],"Default" ,"Val1","A" ,"Val2","B" ,"Val3","C")	select case when "Field"= 'Val1' then ‘A’ when "Field"= 'Val2' then ‘B’ when "Field"= ‘Val3’ then ‘C’ else ‘Default' end

Data Type Considerations

Binary

The Snowflake binary data type is comparable to the Alteryx blob data type. Set your data type to blob in Alteryx if you want to write it as a binary data type in your target table. Note that the writing of blobs is only supported by the standard insert output method. Bulk loading of blob data types is not supported.

Semi-structured (VARIANT, OBJECT, ARRAY)

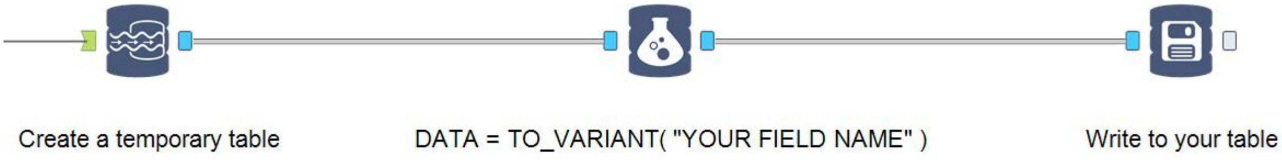
Alteryx does not directly support writing of semi-structured data types to a Snowflake table. There are a variety of ways to handle writing to these data types. One recommended method would be to:

1. Create a temporary table with the data loaded as a VARCHAR using the Data Stream In tool.
2. Using a Formula In-DB tool, cast the data to the semi-structured data type:
    a. TO_VARIANT(“YOUR FIELD NAME”)
    b. TO_OBJECT(“YOUR FIELD NAME”)
    c. TO_ARRAY(“YOUR FIELD NAME”)
3. Write to your target table using the Write Data In-DB tool.

Numeric

Snowflake and Alteryx handle some datatypes differently. In the case of numerical data, Alteryx will upsize numerical fields of datatype Number(38,0) to a higher size of FixedDecimal(40.0) or an Int64 datatype. You will need to use a select tool to convert the field back to a Number(38.0) when writing back to Snowflake tables.

For additional information reference this Knowledge Base article.



Date & Time

Snowflake has a few methods of using timestamps. It is important to know what kind you are using when analyzing data with Alteryx since there are some differences between the two platforms. The three types of timestamps are:

- **TIMESTAMP_NTZ:** this is the datatype for timestamps without a time zone (No Time Zone). This is also referred to as wall time since it is just like looking at a wall clock and writing it down.
- **TIMESTAMP_TZ:** is the datatype for timestamps with time zones. It is worth noting that Snowflake does not store the time zone associated with the timestamp. Instead, it stores the offset from UTC associated with the time zone for that particular timestamp.
- **TIMESTAMP_LTZ:** is the datatype for timestamps with the Local Time Zone of the current session of a Snowflake client.

Alteryx does not recognize timestamps with a time zone attached. You will have to develop additional logic for standard workflows that handle time zones from Snowflake.

Conclusion

With Snowflake and Alteryx handling your data, you can tailor your analytics infrastructure to meet the needs of your analytics strategy. Your organization can unlock the near-unlimited scale, concurrency, and performance of Snowflake using the Alteryx In-Database building blocks. Through Python, APIs, and CLI building blocks, you can easily merge large amounts of data into an automated platform that provides insights based on predictive, prescriptive, and spatial analytics. As your organization grows, collecting and storing more data, your solutions can scale to provide access to data across an organization while giving you the competitive advantage you need to stay ahead and make fast, data-driven decisions.

Next Steps

Take a Tour of the Alteryx Analytics Cloud Platform

Try the Snowflake Starter Kit

Learn More About Alteryx + Snowflake

Start a Free Trial

About Alteryx

As a global leader in analytic process automation (APA), Alteryx unifies analytics, data science and business process automation in one, end-to-end platform to accelerate digital transformation.

Organizations of all sizes, all over the world, rely on the Alteryx Analytic Process Automation Platform to deliver high-impact business outcomes and the rapid upskilling of their modern workforce.

Visit alteryx.com