sas
THE POWER TO KNOW®

Preparing Data in Hadoop for Analysis and Reporting

Course Notes

Preparing Data in Hadoop for Analysis and Reporting Course Notes was developed by Johnny Starling. Editing and production support was provided by the Curriculum Development and Support Department.

Preparing Data in Hadoop for Analysis and Reporting Course Notes

---

Book code E71119, course code ATEPDHAR, prepared date 09Aug2017. ATEPDHAR_001

Table of Contents

To learn more…

For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the web at http://support.sas.com/training/ as well as in the Training Course Catalog.

For a list of other SAS books that relate to the topics covered in this course notes, USA customers can contact the SAS Publishing Department at 1-800-727-3228 or send e-mail to sasbook@sas.com. Customers outside the USA, please contact your local SAS office.

Also, see the SAS Bookstore on the web at http://support.sas.com/publishing/ for a complete list of books and a convenient order form.

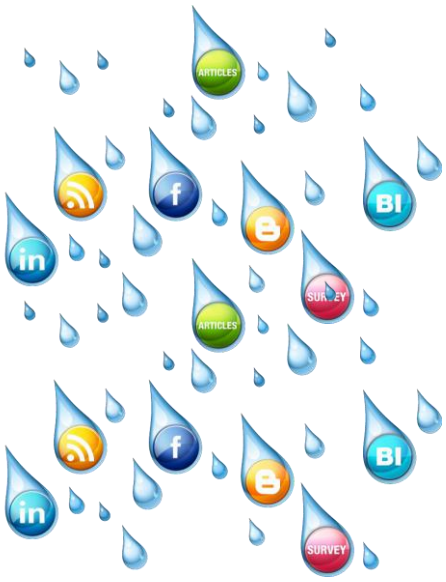
Chapter 1 Data Preparation Using SAS® Data Loader for Hadoop

1.1 Analytical Data Preparation in Hadoop

Big Data Is Everywhere…

… and much of it is ***streaming***.

- social media clicks
- customer interactions
- bank transactions
- financial feeds
- telecommunications
- sensor measurements
- intelligent devices (Internet of Things)



4

In today's world, only 10% of big data comes from relational databases. The other 90% comes from social media streams such as Twitter, Facebook, Netflix, and so on. With the Internet of Things (IoT) continuing to grow, data volume, velocity, and variety are exploding everywhere. The challenge for many organizations is finding ***value*** in the data.

Data Preparation to Support Analytics

Several questions come to mind when considering analytics and the data required to support the analysis:

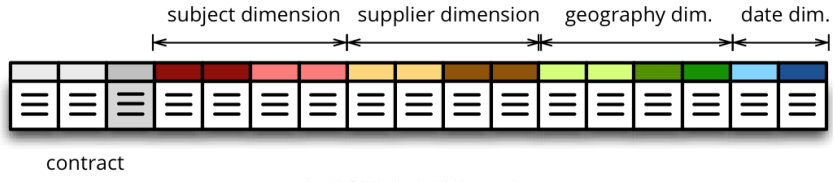
- Why is preparing data to support analytics important?
- What business questions can analytical data help answer?
- What tools are available to make it easy to prepare that data for analysis?

5

Data Requirements to Support Analytics

One-Row-per-Subject requires that all information pertaining to a subject entity, like CUSTOMER_ID, be on a single row. This paradigm supports statistical and data mining analytical methods such as these:

- regression analysis
- analysis-of-variance (ANOVA)
- neural networks
- decision trees (random forests)
- cluster and survival analysis



Regression analysis - estimate the relationship between variables

Analysis-of-variance (ANOVA) - compare the mean of several groups or variables

Neural networks – organized layers that communicate to each other

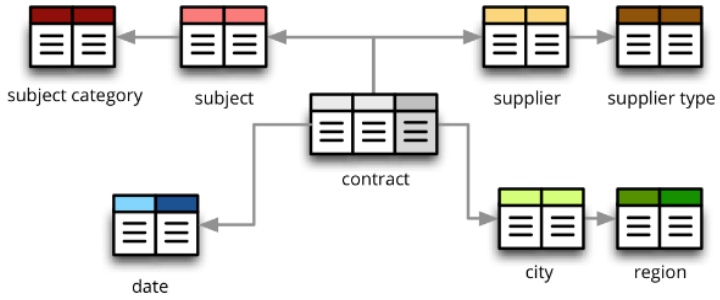
Decision trees (random forests) – learning method for classification, regression, and other tasks based on a multitude of decision trees

Cluster and survival analysis – analysis of data in the form of time from a well-defined origin until its end, such as disease infection data

Data Requirements to Support Analytics

Multiple-rows-per-Subject has a one-to-many relationship between the subject and the multiple observations that are relevant to the subject. This paradigm supports repeated measurements over time and hierarchical relationship analytical methods, such as the following:

- time series analysis
- association analysis
- sequence analysis
- link analysis



7

Time series analysis comprises methods for analyzing ***time series*** data in order to extract meaningful statistics and other characteristics of the data.

Time series forecasting is the use of a model to predict future values based on previously observed values.

Sequence analysis in marketing is often used in analytical customer relationship management applications. In sociology, sequence methods are increasingly used to study life-course and career trajectories, patterns of organizational and national development, conversation and interaction structure, and the problem of work and family synchrony.

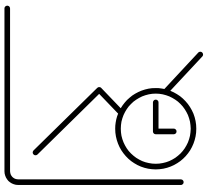
Link analysis is a data analysis technique used in network theory that is used to evaluate the relationships or connections between network nodes. These relationships can be between various types of objects (nodes), including people, organizations, and even transactions.

Association analysis is used to help retailers identify new opportunities for cross-selling their products to the customers. Besides market basket data, association analysis is also applicable to other application domains such as bioinformatics, medical diagnosis, web mining, and scientific data analysis.

Business Questions

One-Row-per-Subject data supports analyses to answer some of the following questions:

- ***prediction of events*** such as, campaign response, contract renewals, insurance claims
- ***prediction of value*** such as customer churn, time until next purchase, profit for next period
- ***segmentation and clustering of customers*** such as customer segmentation, text document clustering, and clustering based on sociodemographic



8

Business Questions

Multiple-Rows-per-Subject data helps answer some of the following questions:

- ***demand forecast*** for a ***product*** in the next 18 months
- ***products*** that are ***frequently purchased together*** (market basket analysis)
- ***Internet user paths*** (clickstream analysis)
- ***trends*** of patient medical information ***over time***, such as blood pressure

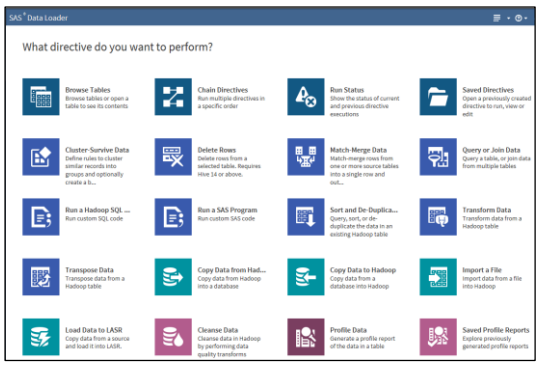
9

SAS Data Loader for Hadoop

SAS Data Loader for Hadoop is

- purpose-built to solve the big data challenges
- a ***web-based, wizard-driven user interface*** that minimizes the need for training and improves the productivity of business analysts and data scientists who work with Hadoop.

Self-service big data preparation for business users



SAS Data Loader is designed to provide business users access to data in the Hadoop cluster, without requiring them to be programmers. This greatly improves productivity and potentially gives them access to data and information that was not available to them in the past. The self-service nature of SAS Data Loader for Hadoop frees business users from their dependence on IT for the data that they need to make key business decisions.

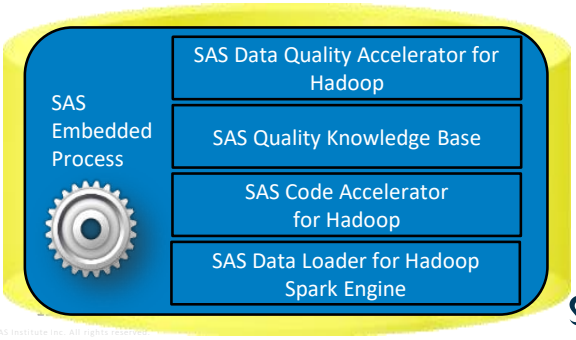
Some of the benefits for using the SAS Data Loader for Hadoop include the following considerations:

- empowers business users in a highly scalable manner
- no required coding or scripting
- no required, specialized skills
- empowers existing resource talents
- uses SAS In-Database Processing
- can execute existing SAS programs

Interaction with Other SAS Technology

SAS Data Loader for Hadoop leverages the following SAS technologies:

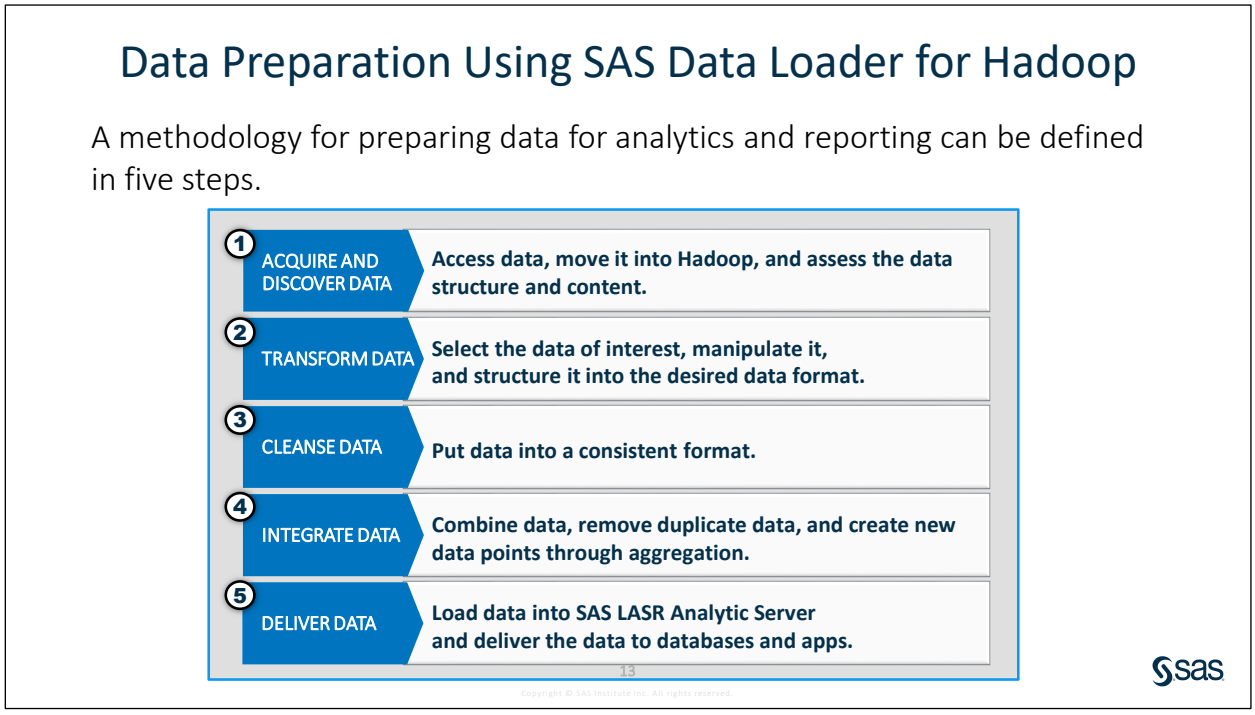
- *SAS/ACCESS* engines
- *SAS Code Accelerator* to execute SAS DS2 code
- *SAS Data Quality Accelerator* to leverage SAS QKB functions
- *SAS Data Loader for Hadoop Spark Engine* to execute data quality code in Hadoop memory
- *SAS LASR Analytic Server* to analyze Hadoop data in SAS Visual Analytics or SAS Visual Statistics



SAS software is also deployed to each node in the Hadoop cluster. SAS software on the cluster includes the following components:

- SAS Quality Knowledge Base, which supports data cleansing definitions
- SAS Embedded Process software, which supports in-database code execution using the SAS Code Accelerator
- SAS Data Quality Accelerator, which supports SAS DS2 methods for data cleansing in-database
- SAS Data Loader for Hadoop Spark Engine, which enables you to execute data integration and data quality tasks in Apache Spark (in-memory) on a Hadoop cluster
- SAS LASR Analytic Server, which enables you to load data from Hadoop into LASR for further analysis and reporting

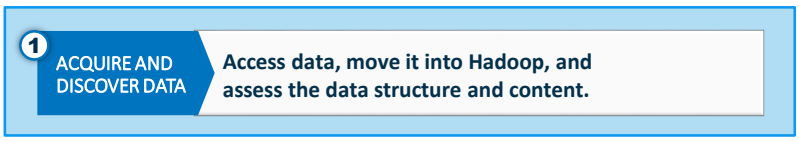
1.2 Acquire and Discover Data



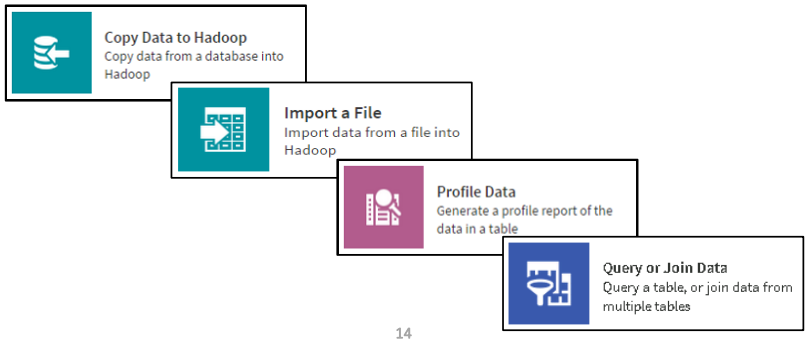
The process for preparing data for analytics and reporting can be defined in five steps:

1. Acquire the data that you believe is relevant. Discover what structure and inconsistencies exist in the data.
2. Transform the data into a desired format.
3. Cleanse the data of anomalies and inconsistencies.
4. Integrate the data with other sources to provide data that is fit-for-purpose and ready for analysis and reporting.
5. Deliver the data as new tables and views to the appropriate application for analysis or another third-party reporting tool.

Methodology Step 1 (Review)



The first step of the methodology deals with acquiring and discovering data.



Copy Data to Hadoop Directive

SAS libraries configured in the SAS Metadata using the SAS/ACCESS Interface technology are sources available for use in the Copy Data to Hadoop directive.

- Cloud Analytic Services server (CAS tables)
- Base SAS data sets
- Hadoop
- Oracle
- Teradata
- … and more

When you access source or target table locations, any libraries that are configured in SAS Metadata that are available to the SAS Workspace Server are also available for use by SAS Data Loader for Hadoop. These connections use the SAS/ACCESS Interface technologies. The user can also configure a Sqoop connection in SAS Metadata to use a JDBC connection to a relational database.

Importing a File Directive

The Import a File directive provides these capabilities:

- load a delimited file from the SAS Workspace Server to HDFS using *PROC HADOOP* and the *HDFS COPYFROMLOCAL* command
- create a table definition in Hive and assign an HDFS file using the *LOAD DATA INPATH* option
- use the first row to define the column definitions
- generate default column definitions for the target based on a sample of the data

16

The Import a File directive enables you to perform various operations when you copy a delimited text file into Hadoop for subsequent processing.

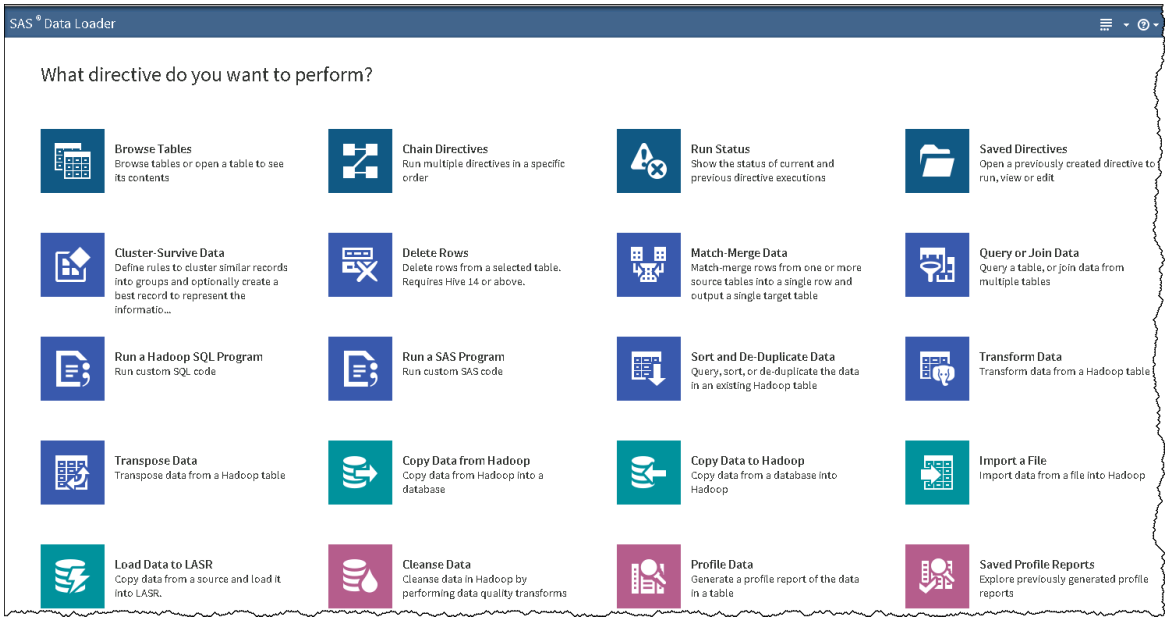
Note: In the source file, the delimiter must consist of a single character or symbol. Input delimiters must have ASCII character codes that range from 0 to 255, or octal values that range from \000 to \177.

Copy a SAS Data Set into Hadoop

This demonstration reviews how a user would copy the SAS data set **Customers** from a SAS library location that is specified in SAS Metadata into Hadoop for further processing and analysis using additional SAS Data Loader directives.

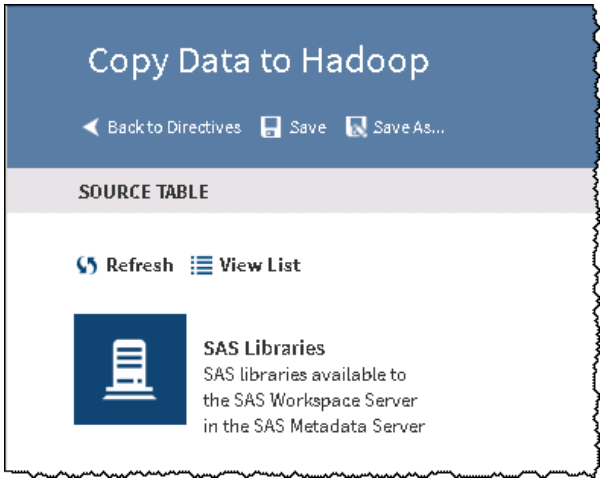
The location is specified in SAS Metadata.

1. Open an Internet browser.
2. Type ***<your SAS mid-tier server name>*:7980/SASDataLoader**.
3. Type valid credentials in the **User ID** and **Password** fields.
4. Verify that you are at the Data Loader console.

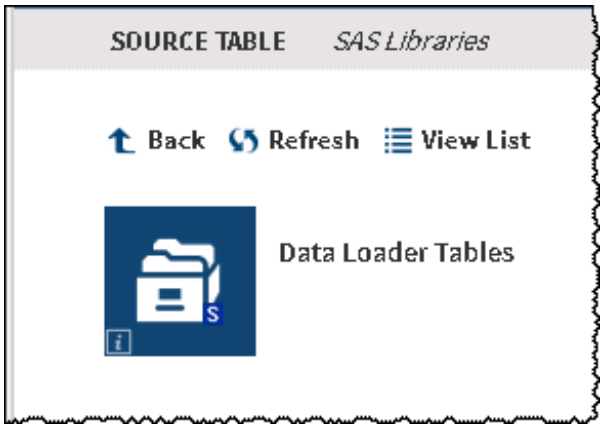


5. Select the **Saved Directives** directive.
   a. Select the **Copy Customers to Hadoop** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
6. Select the **Copy Data to Hadoop** directive.

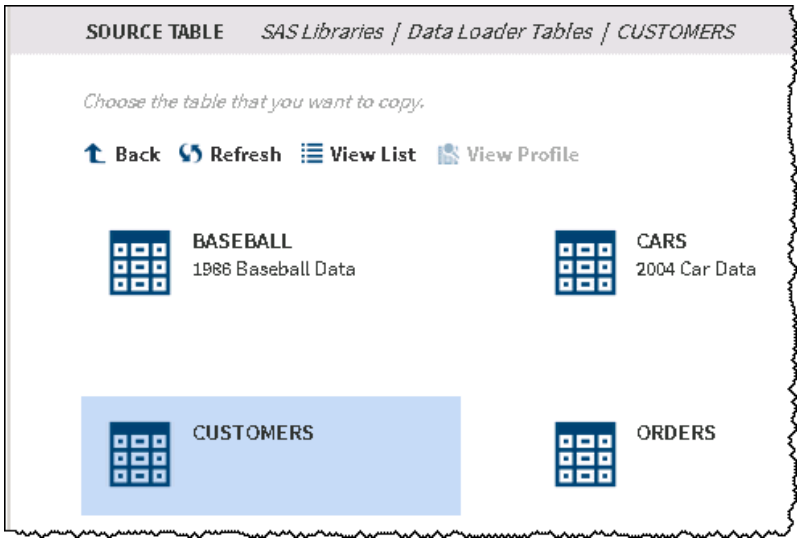
7. Select **SAS Libraries**.



8. Select **Data Loader Tables**.

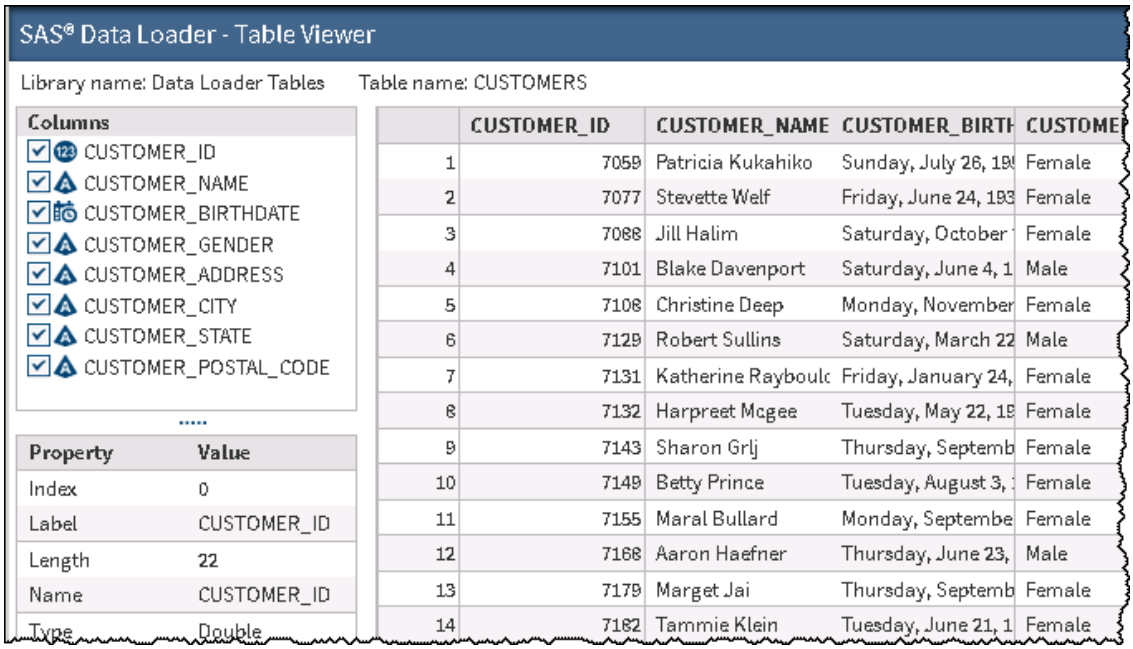


9. Select the **CUSTOMERS** table.

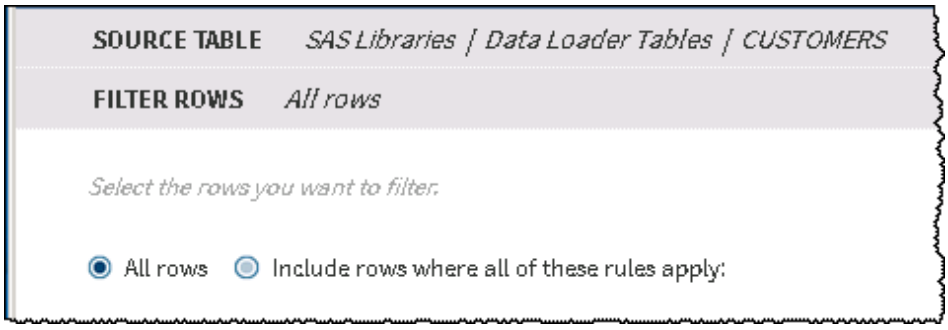


10. Click (the **Table Viewer** icon) in the upper right corner.

Note: The table is surfaced on a separate tab in the browser.

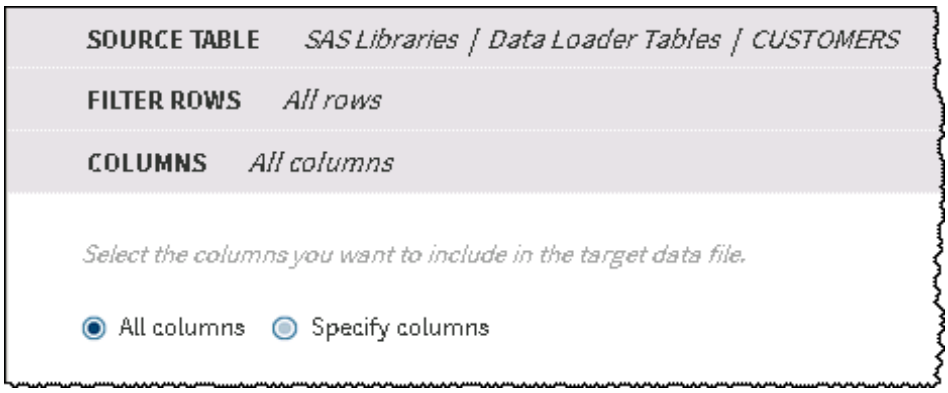


11. Close the **Table Viewer** tab.
12. Click **Next**.
13. Select the **All rows** radio button.



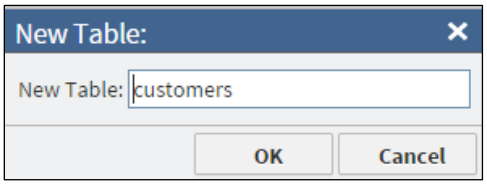
14. Click **Next**.

15. Accept the default (**All columns**).



16. Click **Next**.
17. Click **New Table** to create a new target table.

a. Enter **customers** in the **New Table** field.



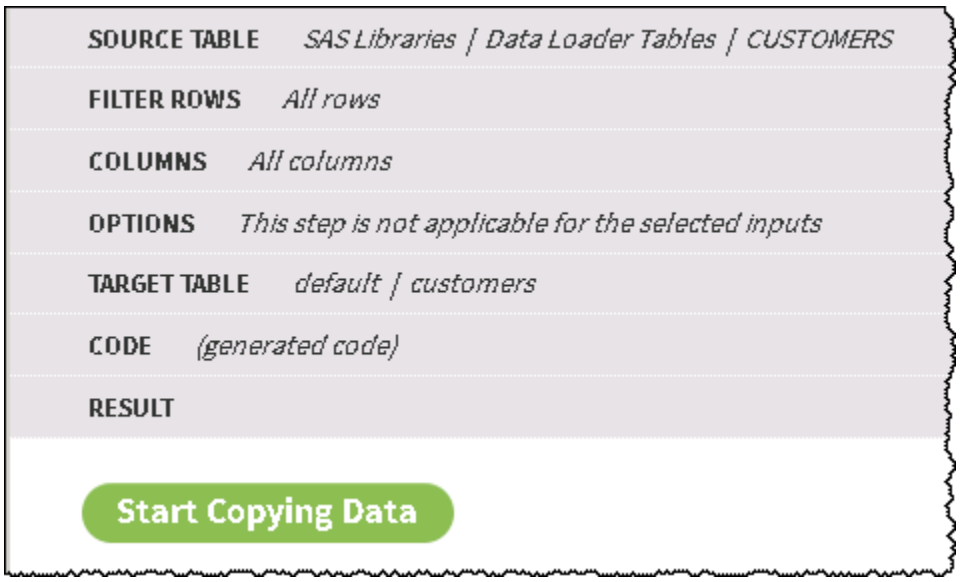
b. Click **OK**.

18. Click **Next** to view the generated LIBNAME and PROC SQL code.

```
1    LIBNAME sasdl BASE "/workshop/DL31HD/sas_data";
2  ;
3      libname WY2FR3PO HADOOP
4       port=10000
5       schema=default
6       pwd="{sas002}00403C1F0021646C0E1E965D3A1DC093"
7       transcode_fail=warning
8       host="server2.demo.sas.com"
9       PROPERTIES="mapreduce.job.queuename=default"
10      user=student;
11
12 PROC SQL;
13 CREATE TABLE WY2FR3PO.customers
14 AS
15 SELECT *
16 FROM sasdl.CUSTOMERS table0;
17 QUIT;
```

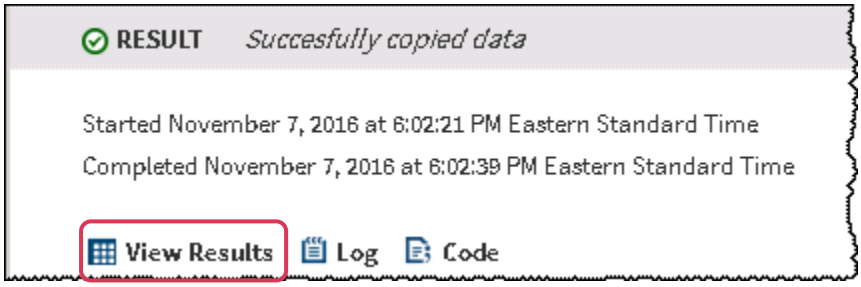
Note: Click the **Edit Code** button to modify the generated code if necessary.

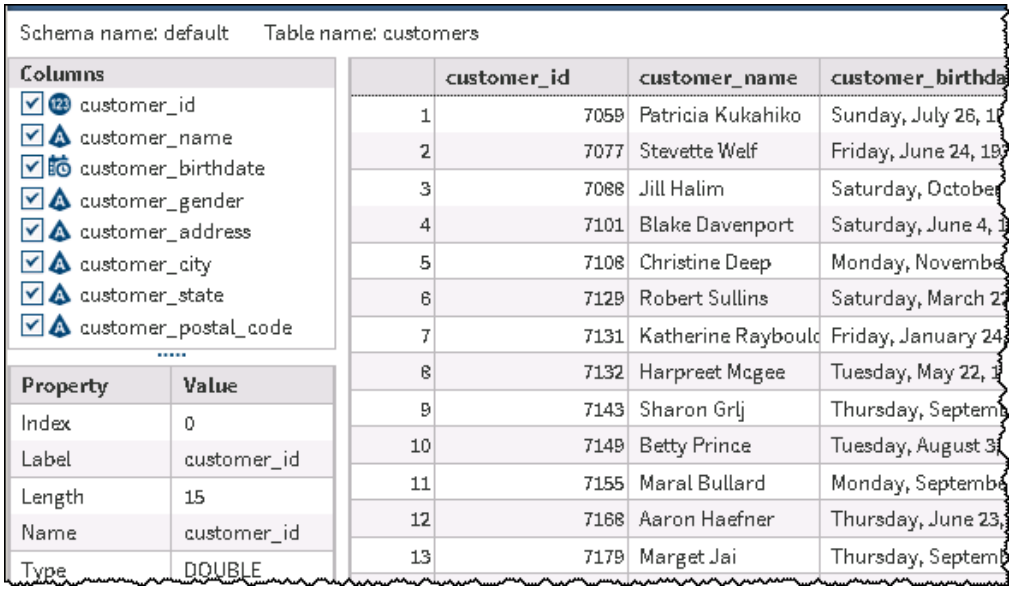
19. Click **Next**.
20. Click **Start Copying Data**.



Note: Clicking any task above the Start Copying Data button enables you to edit that task. Any changes to the task might require stepping through the subsequent tasks to cascade the change and its impact through the directive.

21. Select **View Results** to preview the results that were loaded to the target table.





22. Close the Table Viewer tab.
23. Click **Save As** to save the directive to the SAS folders.
    a. Enter **Copy Customers** in the **Directive name** field.
    b. Click **OK**.
24. Click **Back to Directives** to return to the main SAS Data Loader interface.

End of Demonstration

Discover Data Using Data Profiling

Data Profile	Provides the ability to inspect data for errors, inconsistencies, redundancies, and incomplete information.

Data profiles provide the following advantages:

- improve the understanding of existing databases
- aid in identifying issues early in the data management process, when they are easier and less expensive to manage
- help determine which steps need to be taken to address data problems
- enable you to make better business decisions about your data

18

Profile Data Capabilities

The Profile Data directive provides the following capabilities:

- profiles data based on specific locale definitions
- generates basic statistics, recognizes patterns, identifies scarcity, and calculates frequency
- reviews and evaluates profiled data trends over time
- identifies redundant and cross-column dependencies
- saves profile reports to SAS folder locations for sharing

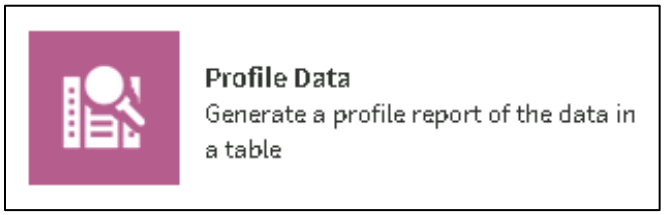
19

Using the Profile Data in Hadoop directive enables you to check the data for inconsistencies and anomalies.

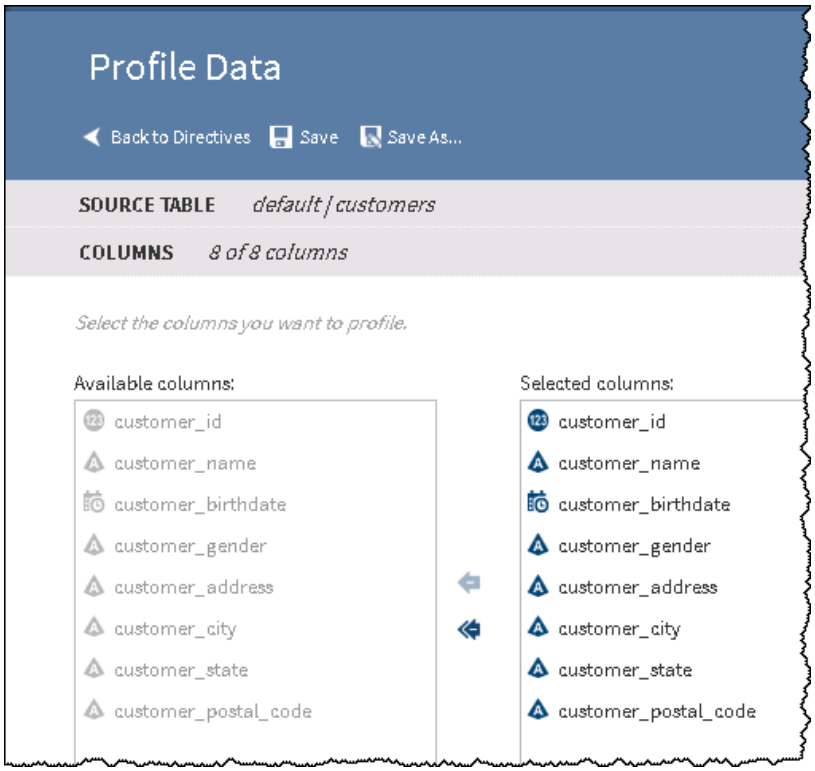
Creating and Exploring a Data Profile

This demonstration reviews the steps that are necessary to analyze the Hadoop table **CUSTOMERS** for inconsistencies and anomalies using a data profile. The results are written and reviewed in a profile report that is stored in a SAS folders location that is defined in SAS Metadata.

1. Verify that you are viewing the Data Loader console.
2. Select the **Saved Directives** directive.
   a. Select the **Profile Customers** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
3. Select the **Profile Data** directive from the SAS Data Loader console.

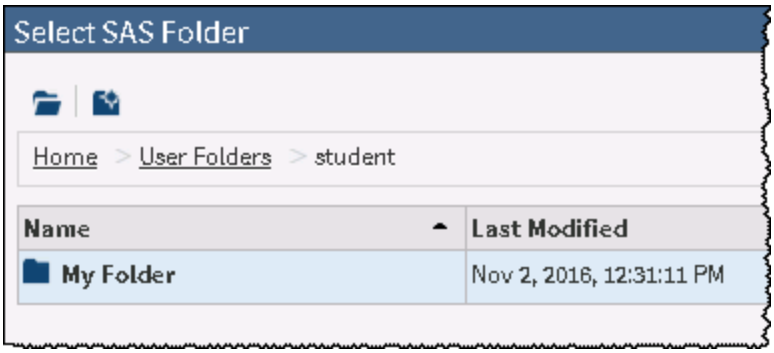


4. Select the **customers** source table.
5. Click **Next**.



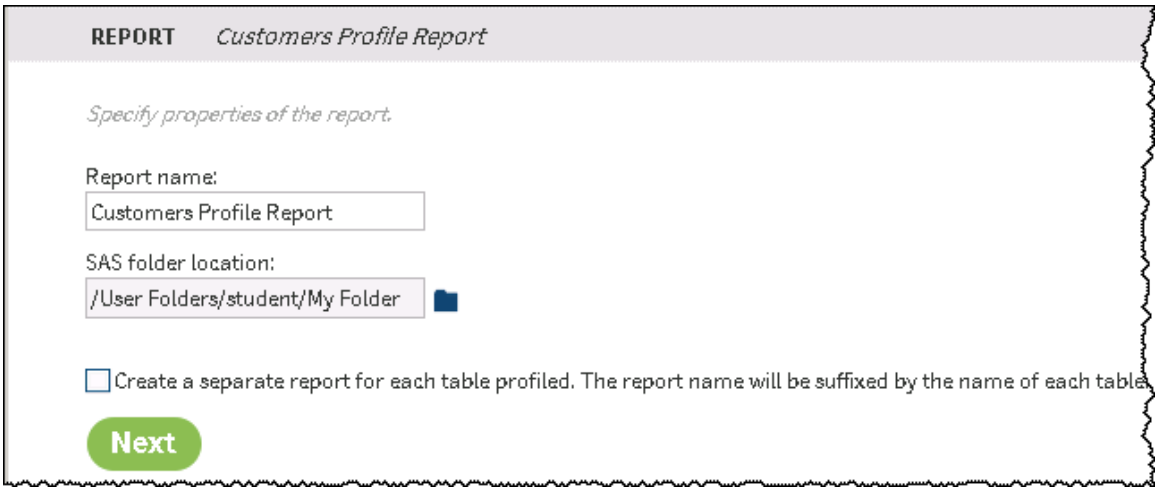
6. Click **Next** to accept the default and profile all the columns.
7. Enter **Customers Profile Report** in the **Report name** field.

a. Click (**Folder Selection** icon) to select a SAS folder location for the profile report.

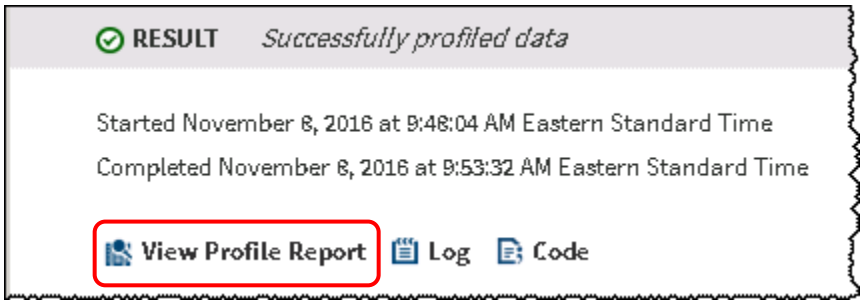


b. Select **My Folder**.

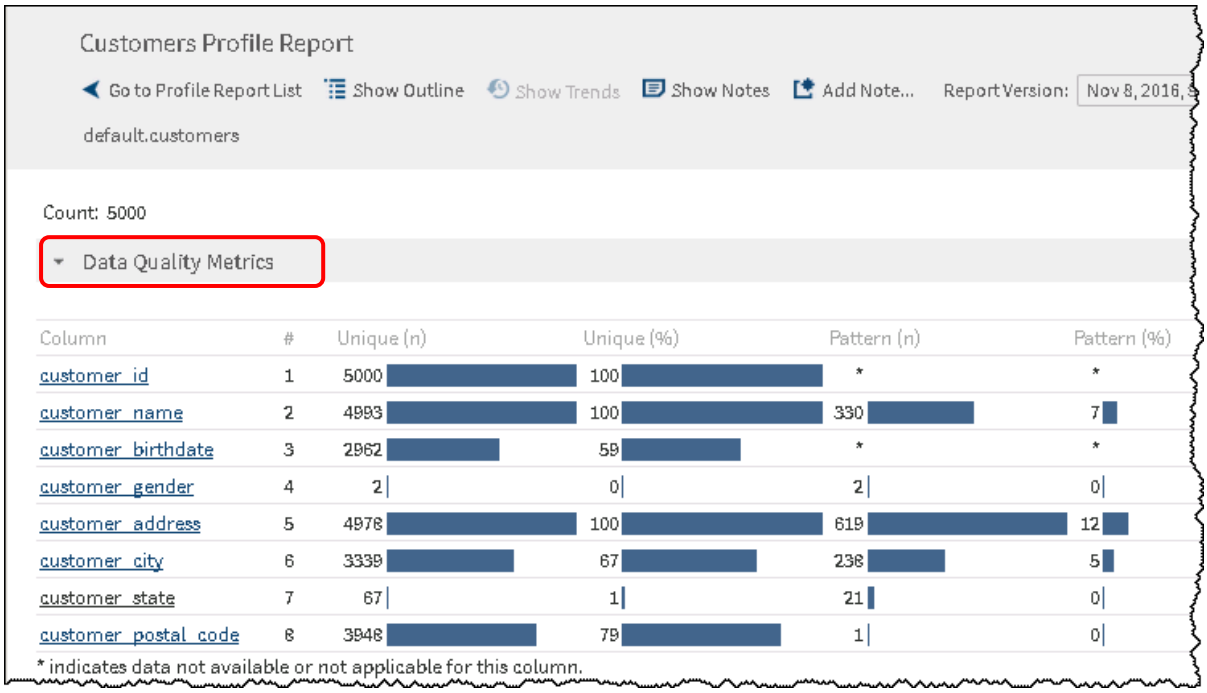
c. Click **OK**.



8. Click **Next**.
9. Click **Create Profile Report**.
10. Verify that the profile was successful.



11. Click **Save As**.
    a. Enter **Profile Customers** in the **Directive name** field.
    b. Click **OK**.
12. Click **View Profile Report** to open the report on a new browser tab.
13. Verify that the profile report was generated with **5000** total records and that the following values are in each report:



Customers Profile Report

Go to Profile Report List | Show Outline | Show Trends | Show Notes | Add Note... | Report Version: Nov 8, 2016, 3

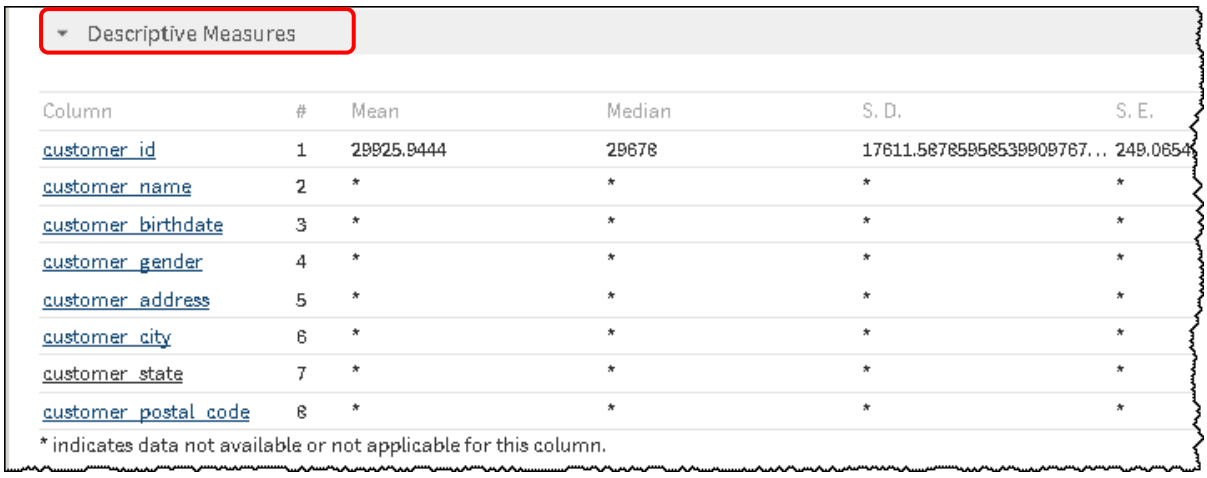
default.customers

Count: 5000

Data Quality Metrics

Column	#	Unique (n)	Unique (%)	Pattern (n)	Pattern (%)
customer_id	1	5000	100	*	*
customer_name	2	4993	100	330	7
customer_birthdate	3	2962	59	*	*
customer_gender	4	2	0	2	0
customer_address	5	4978	100	619	12
customer_city	6	3339	67	238	5
customer_state	7	67	1	21	0
customer_postal_code	8	3948	79	1	0

* indicates data not available or not applicable for this column.



Descriptive Measures

Column	#	Mean	Median	S. D.	S. E.
customer_id	1	29925.9444	29678	17611.587859585399097 67...	249.0654
customer_name	2	*	*	*	*
customer_birthdate	3	*	*	*	*
customer_gender	4	*	*	*	*
customer_address	5	*	*	*	*
customer_city	6	*	*	*	*
customer_state	7	*	*	*	*
customer_postal_code	8	*	*	*	*

* indicates data not available or not applicable for this column.

Metadata Measures

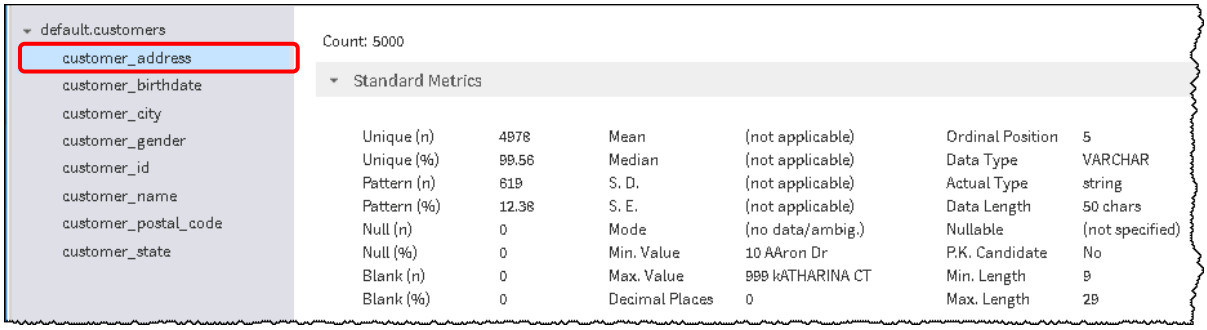
Column	#	Data Type	Actual Type	Data Length
customer_id	1	DOUBLE	*	8
customer_name	2	VARCHAR	string	30
customer_birthdate	3	TIMESTAMP	*	19
customer_gender	4	VARCHAR	string	6
customer_address	5	VARCHAR	string	50
customer_city	6	VARCHAR	string	30
customer_state	7	VARCHAR	string	25
customer_postal_code	8	VARCHAR	integer	10

* indicates data not available or not applicable for this column.

Note: Charts show summary graphics that provide information about the uniqueness and incompleteness of column values. Is the column a candidate for being a primary key? If it is incomplete, it probably is not a candidate.

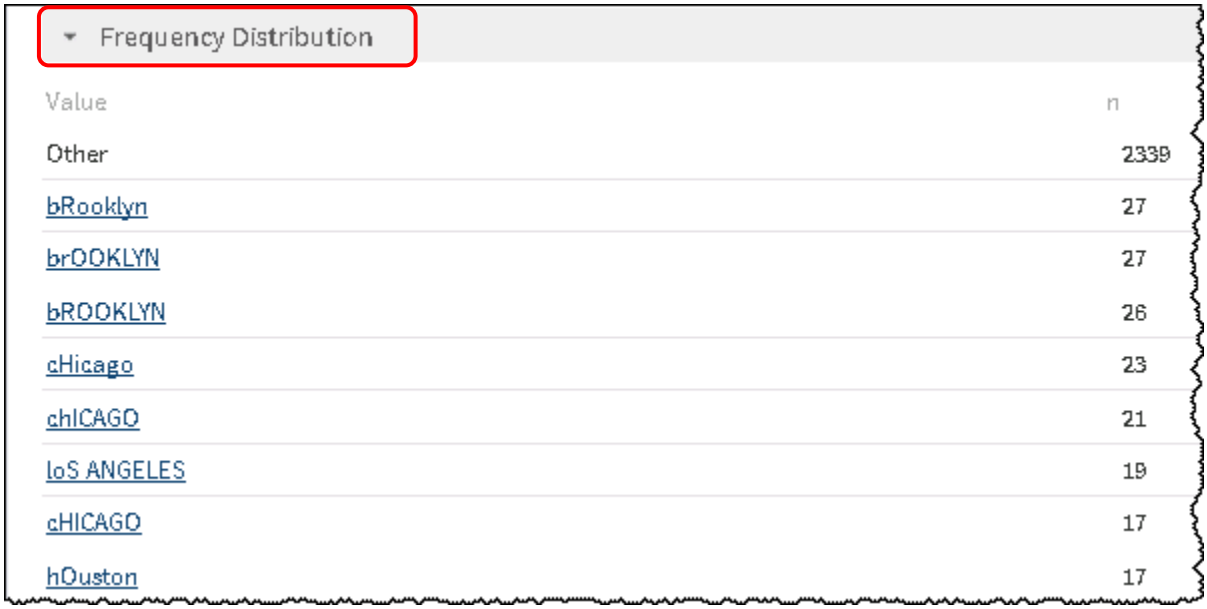
14. Click **Show Outline**.

15. Select **customer_address**.



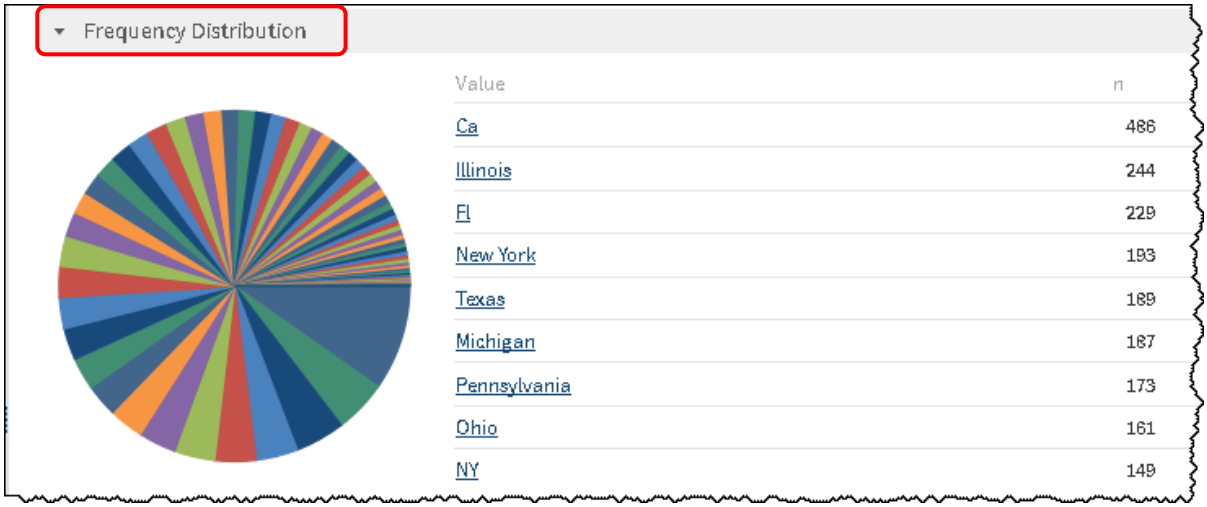
Note: The addresses under Frequency Distribution are in mixed case. As a result, the data must be standardized.

16. Select **customer_city**.



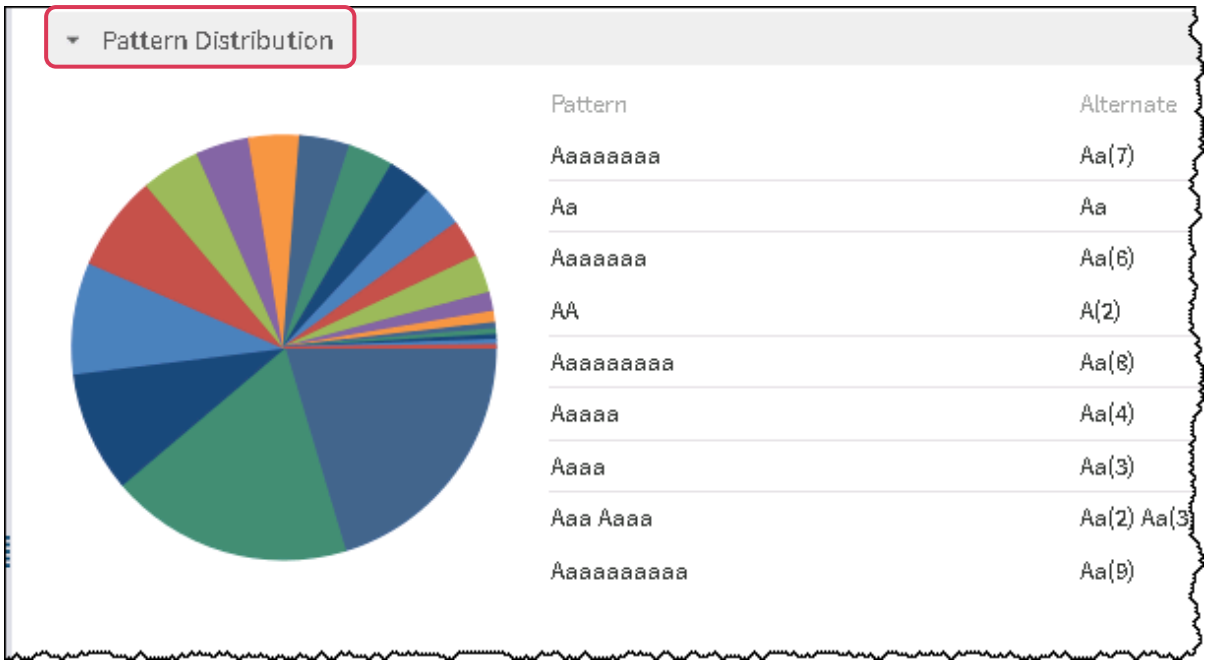
Note: The cities are in mixed case and require standardization. Notice the multiple occurrences of the same city due to mixed case.

17. Select **customer_state**.



Note: The number of unique states and the **customer_state** data include both full and abbreviated names. Both cases indicate that the data needs to be standardized.

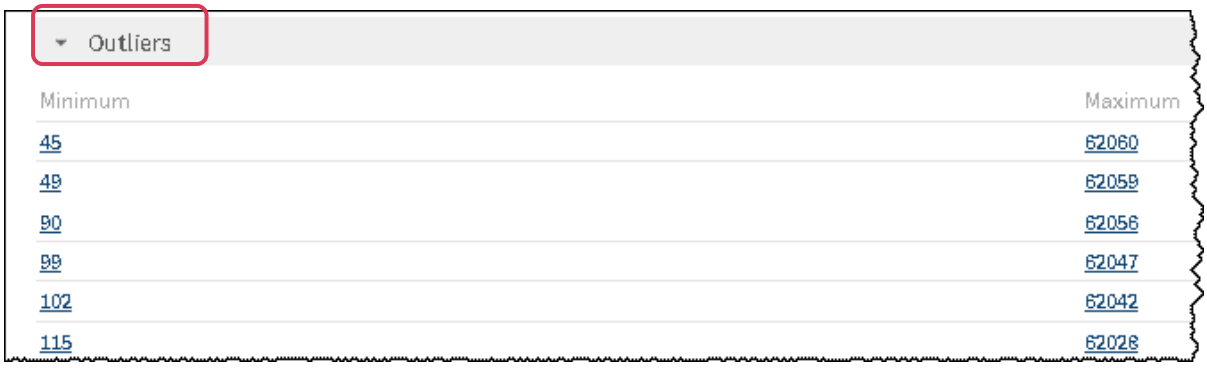
18. Scroll down and review the pattern distribution.



Note: The various patterns for **customer_state** indicate an inconsistency in the names. This might lead to unpredictable results.

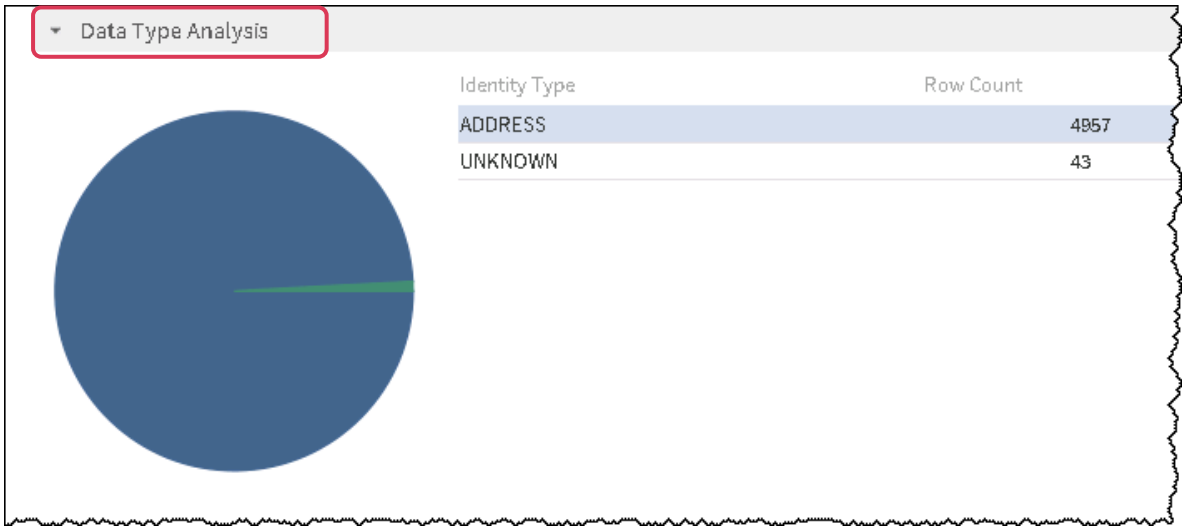
19. Select **customer_id**.

20. Scroll down and review the Outliers report.



Note: A review of the Outliers report for **customer_id** might indicate where expected **customer_id** values are out of range and unacceptable.

21. Select **customer_address**.
22. Scroll down and review the Data Type Analysis report.



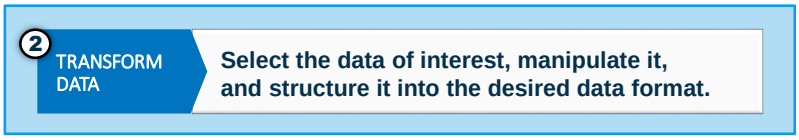
Note: The UNKNOWN type is an indication of an address that might be invalid.

23. Close the **Profile Report** tab in the web browser by clicking **X**.
24. Click **Back to Directives** to return to the Data Loader console.
25. Click **OK** in the Unsaved Directive dialog box.

End of Demonstration

1.3 Transform and Transpose Data

Methodology Step 2 (Review)



The second step of the methodology involves restructuring the data into a desired format. The following directives are available:

22

Step 2 involves restructuring the data to the desired format. This might include dropping columns, filtering rows, or using aggregation and summarization to create new columns. It might also include restructuring the data for analytical consumption, which involves transposing the data into a one-row-per-subject table format.

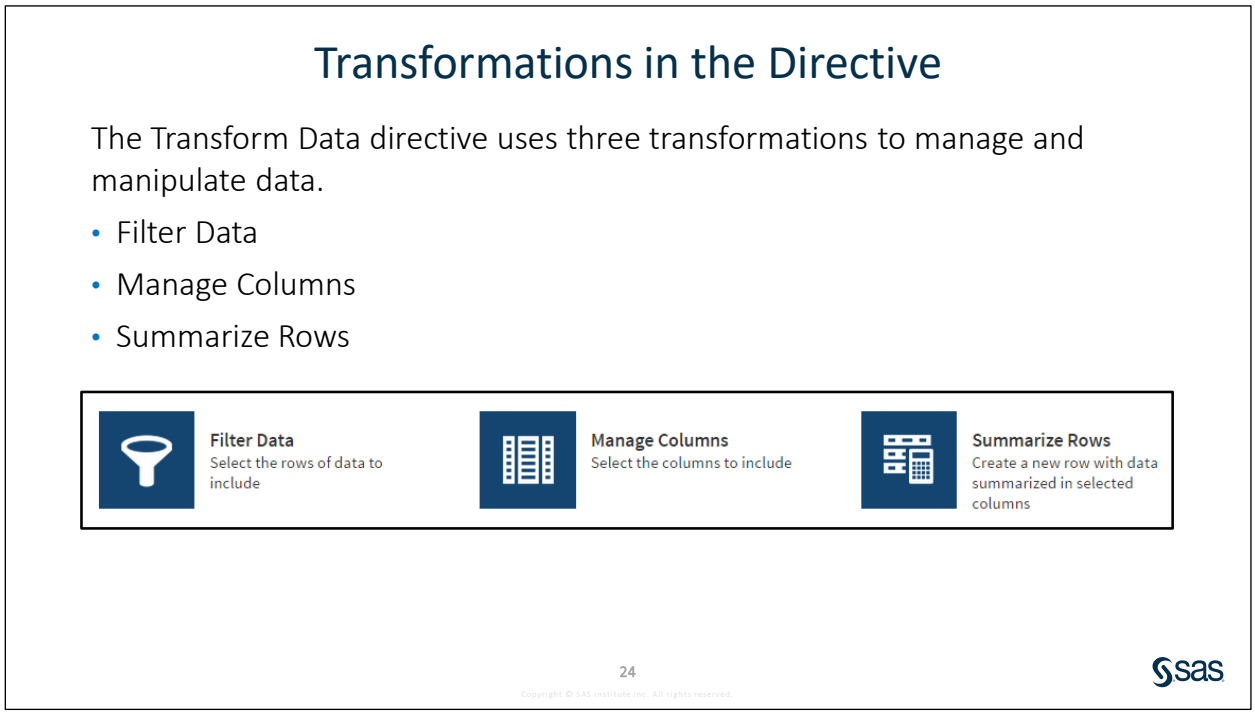
Why Transform Data in Hadoop?

The Transform Data directive enables you to do the following actions:

- select data of interest
- manipulate data
- modify the data structure as required for the desired reports, analyses, or both

23

The Transform Data directive enables you to create a table that contains only the data that you need in the structure that is required for the analytical or reporting objectives. This ensures that additional processing of unwanted data is not necessary, which increases processing efficiency.



These transformations give you the ability to perform the following actions:

- select columns
- apply filters
- map columns
- sort or order
- calculate columns
- aggregate data

Filter Data Transformation Capabilities

The *Filter Data* transformation enables you to do the following actions:

- define one or more rules or expressions to filter rows in a table
- create compound conditions using AND or OR
- view distinct data values for columns that are profiled

Filter condition operators include Equal to, Not Equal, In, Not In, Like, Null, Not Null, Contains, and Not Contains.

25

The Filter Data transformation enables you to remove unwanted data from your result set. As you build the filters, you can use either filter rules or expressions. Expressions are defined in the SQL editor, which has a list of functions and a Help resource to aid in their creation.

If the source table is profiled, then distinct values can be viewed when you create filter rules.

Manage Columns Transformation Capabilities

The *Manage Columns* transformation enables you to perform the following actions:

- select columns, order columns, set column metadata
- rename columns
- map columns
- create new columns with calculated values using SAS DS2 or DataFlux Expression Engine Language code

Note: SAS reserved words are flagged as invalid column names.

26

The Manage Columns transformation provides a robust set of features that enable you to manage, manipulate, and create columns for the target table. These columns are suitable for subsequent processing and reporting.

Summarize Rows Transformation

The *Summarize Rows* transformation enables you to perform the following actions:

- group rows based on values in one or more columns
- define column names for new aggregations
- choose between **Count**, **Count Distinct**, **Max**, **Min**, and **Sum** to aggregate values

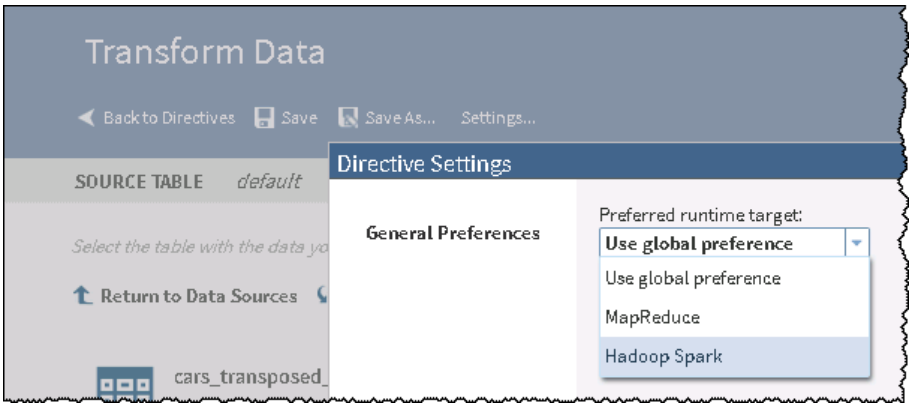
27

The Summarize Rows transformation gives you the ability to group one or many rows of data based on columns. From these groupings, you can create aggregations on numeric values with a predefined set of aggregations.

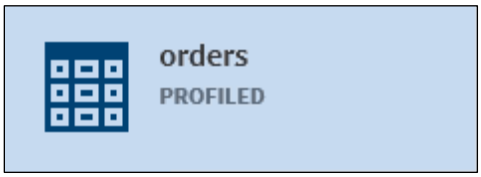
Transforming Data for Reporting

This demonstration reviews the use of the Transform Data directive and its associated transformations to get a list of customers with product purchases of more than five items and the associated costs per product identification to help determine customer discounting policies.

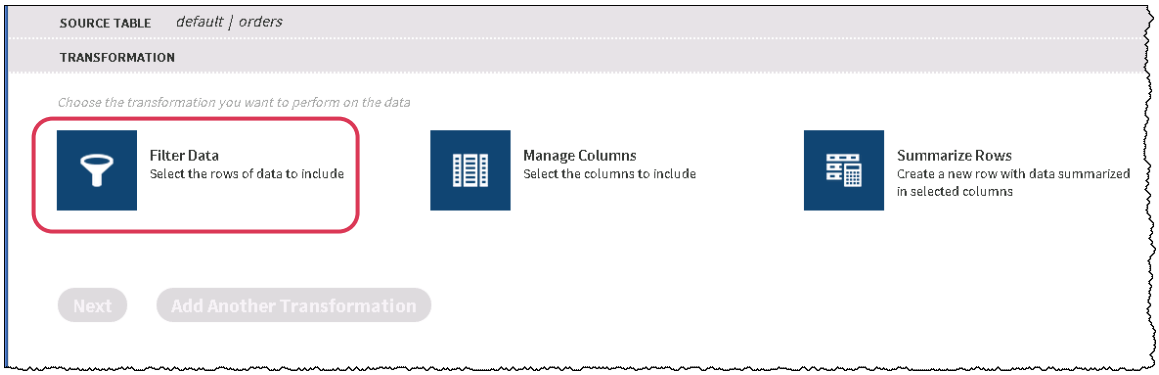
1. Verify that you are in the SAS Data Loader console.
2. Select the **Saved Directives** directive.
   a. Select the **Transform Orders** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
3. Click the **Transform Data** directive.
4. Click **Settings** in the Transform Data heading.
   a. Click the **Preferred runtime** target menu arrow.
   b. Select **Hadoop Spark** to execute the directive in Hadoop memory.



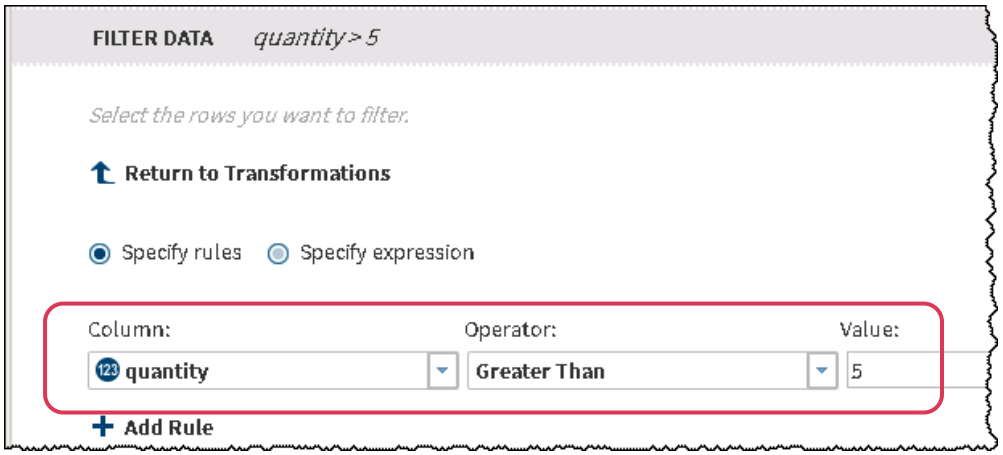
   c. Click OK.
5. Select the **Orders** table.



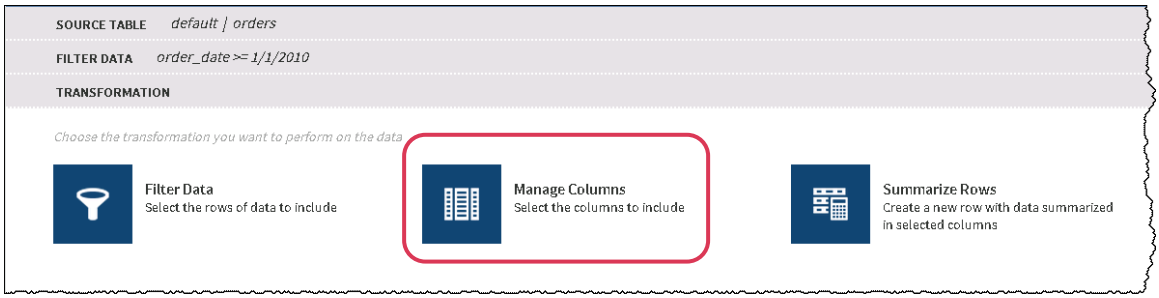
6. Click **Next**.
7. Select the **Filter Data** transformation.



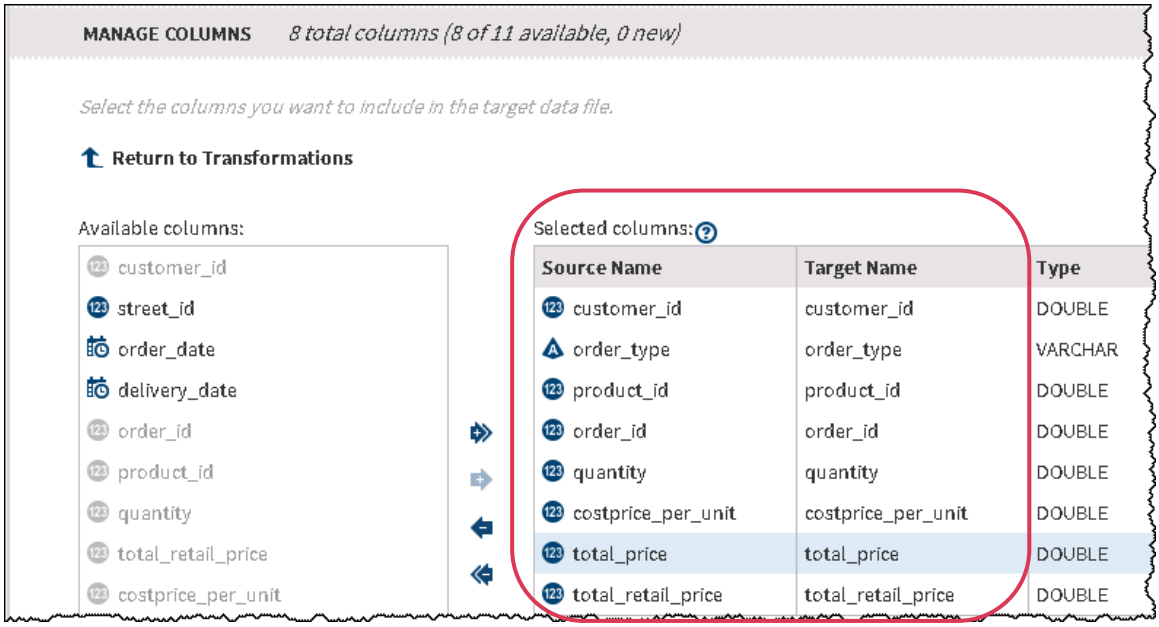
8. Click **Specify rules**.
9. Create the rule below by selecting the column **quantity** and the operator **Greater Than**.
10. Enter **5** in the **Value** field.



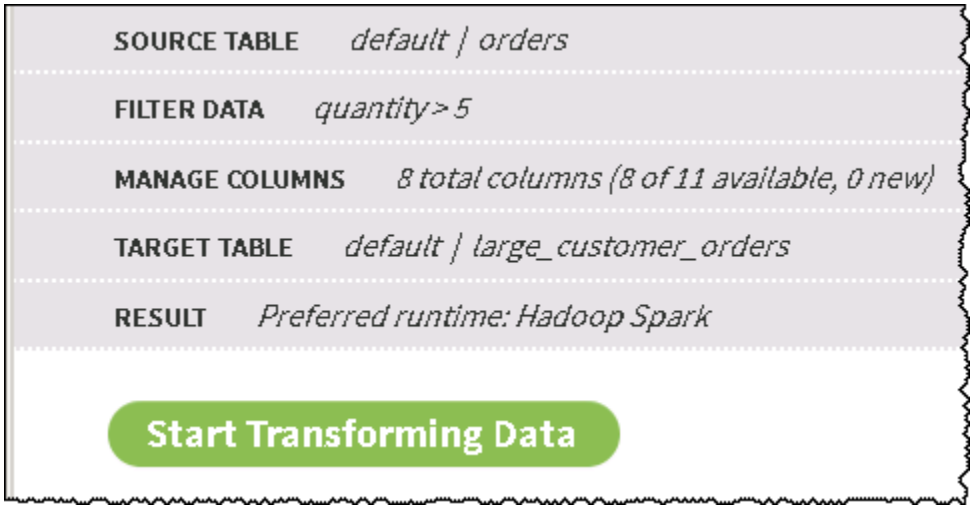
11. Click **Add Another Transformation**.
12. Select **Manage Columns**.



13. Remove all the columns from the Selected columns list box, except **customer_id**, **order_type**, **product_id**, **order_id**, **quantity**, **costprice_per_unit**, **total_price**, and **total_retail_price**.



14. Click **Next**.
15. Click **New Table**.
    a. Enter **large_customer_orders** in the **New Table** field.
    b. Click **OK**.
16. Click **Next**.



17. Click **Start Transforming Data**.
18. Click **View Results** to verify the data in the target table.

customer_id	order_type	product_id	order_id	quantity
1455	Catalog Sale	220100100418	1234288404	6
1912	Retail Sale	220100500026	1231956960	6
4648	Retail Sale	220100100500	1232004024	6
4971	Catalog Sale	220200200066	1234269858	6
5128	Retail Sale	220101400021	1234348928	6
6062	Retail Sale	220101400223	1234290932	6
7583	Retail Sale	220101000002	1231071662	6
8282	Retail Sale	220100100231	1233483019	6
8419	Retail Sale	220200200045	1231856410	6

19. Close the Table Viewer.
20. Click **Save As**.
    a. Type **transform orders** in the **Directive name** field.
    b. Click **OK**.
21. Click **Back to Directives** to return to the SAS Data Loader console.

End of Demonstration

Why Transpose Data in Hadoop?

Important questions to answer are these: Why transpose data? What is the purpose of transposing columns into rows and vice versa?

Here is a short list of reasons:

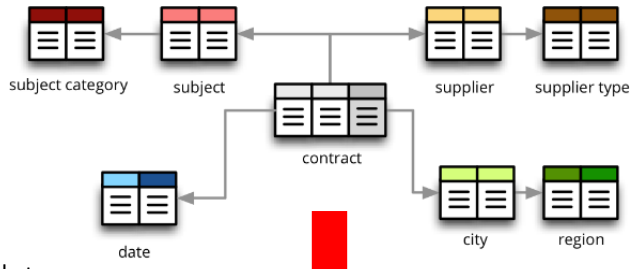
- Many multivariate data mining and predictive modeling methods ***require*** data in the form of one row per subject and multiple variables per row.
- The ability to transpose data is a critical success factor for being able to answer business questions with ***as many variables as possible***.
- All the necessary information for having good predictors is available for a target event or value.

29

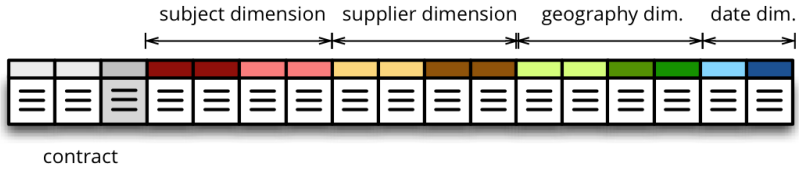
This slide discusses some of the basic reasons for transposing data. The most important reason is that the analytical analysis of data, such as data mining, requires that as many variables per subject as possible be available for analysis.

Fit-for-Purpose Data Structures

- efficient storage
- fast retrieval
- defined schema



- ***wide*** tables or time series data
- iteration (build, test, repeat)
- schema-less



sas

To gather all of the necessary data for a report, several passes over the data might be required. This is very inefficient and time consuming. You can transpose that data using group by variables to cluster the data.

Transpose Using BY Variables

Obs	ptid	sex	visit	sbp
1	01	F	1	142
2	01	F	2	141
3	01	F	3	131
4	02	F	1	107
5	02	F	2	.
6	02	F	3	111
7	03	M	1	135
8	03	M	3	128

Transpose systolic blood pressure values and group by patient ID.

Transpose by

Obs	ptid	sex	systolic1	systolic2	systolic3
1	01	F	142	141	131
2	02	F	107	.	111
3	03	M	135	.	128

31

sas

The bottom diagram shows the significance of transposing ***by*** a variable or variables. You can see very quickly how this is an invaluable process where you ultimately need to join this data to data that uses the patient ID as the common key.

The variables in the BY statement define how the data set is structured. In this case, each row represents a patient ID and each column represents a visit and the systolic pressure taken during the visit.

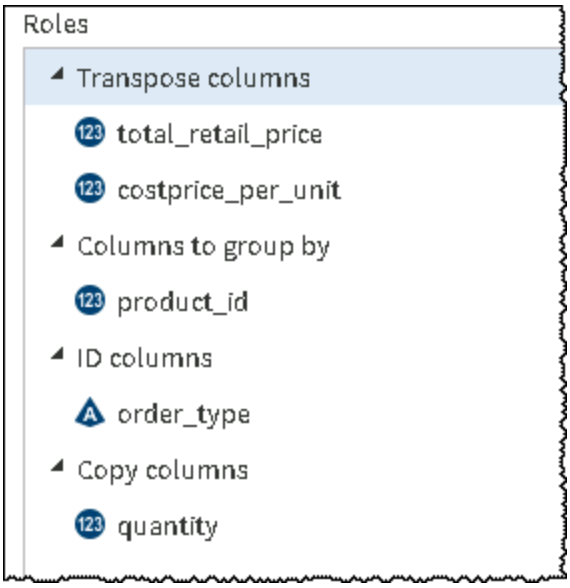
The ***values*** of the variables in the ID statement reflect the names of new columns. In this case, the visit number is part of the systolic columns.

Transposing Data in Hadoop

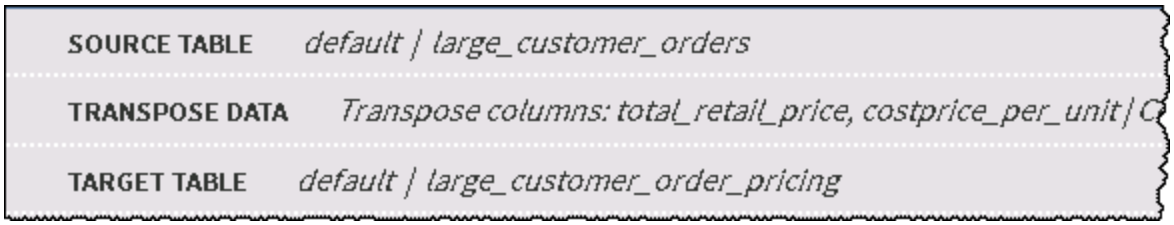
This demonstration reviews how to transpose the **LARGE_CUSTOMER_ORDERS** table by **cost_per_unit** and **total_retail_price** per **order_type** using the Transpose Data directive. The table is grouped by the product identification number.

1. Select the **Saved Directives** directive.
   a. Select the **Transpose Large_Customer_Orders** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
2. Select the **Transpose Data** in Hadoop directive.

3. Click **default** for the Hadoop schema, if it is not already selected.
4. Click **large_customer_orders**.
5. Click **Next**.
6. Assign the columns that are indicated below to the designated roles.
7. Click the **Transpose columns** role.
   a. Double-click **total_retail_price** in the Columns list box.
   b. Double-click **cost_per_unit** in the Columns list box.
8. Click the **Columns to group** by role.
   Double-click **product_id** in the Columns list box.
9. Click the **ID columns** role.
   Double-click **order_type** in the Columns list box.
10. Click **Copy columns**.
    Double-click **quantity**.



11. Click **Next**.
12. Click the **default** schema.
13. Click **New Table**.
    a. Type **large_customer_order_pricing** in the **New Table** field.
    b. Click **OK**.
14. Click **Next**.



15. Click **Start Transposing Data**.
16. Click **View Results** to verify the target table data.

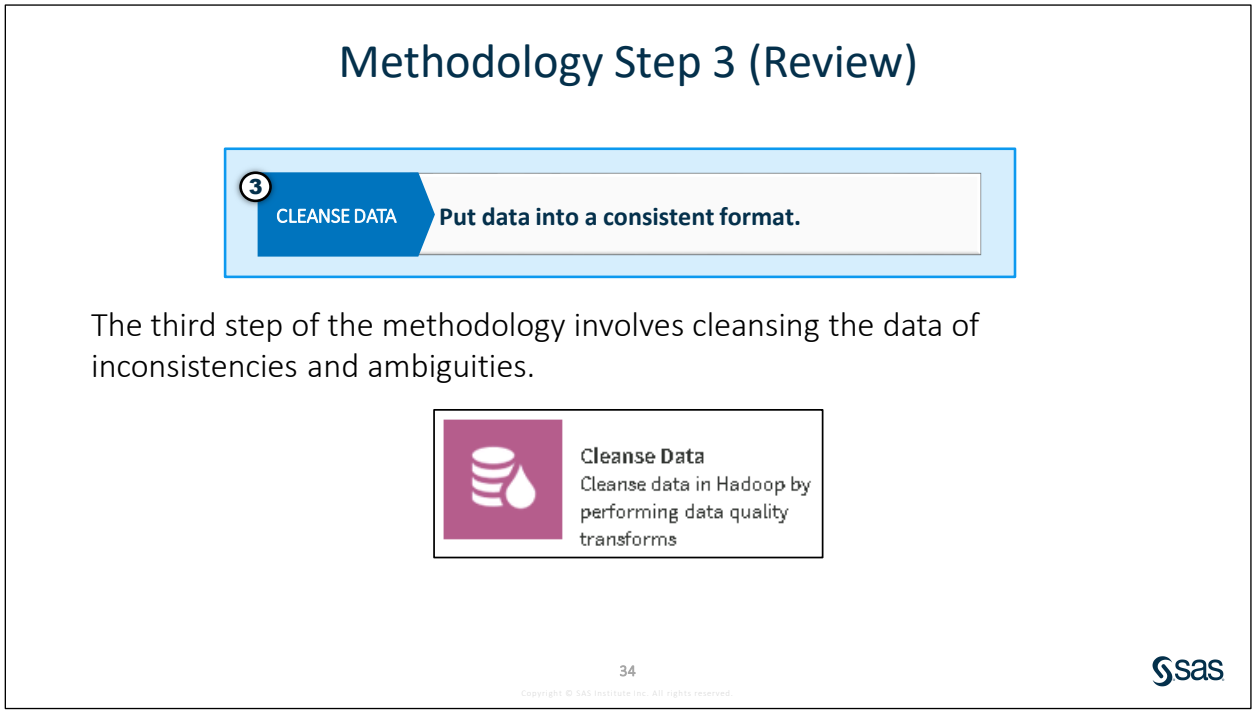
product_id	dl_name_	quantity	catalog_sale	internet_sale	retail_sale
220100100016	total_retail_price	6	null	null	85.1999999999995
220100100016	costprice_per_unit	6	null	null	7.8
220100100038	total_retail_price	6	null	null	118.2
220100100038	costprice_per_unit	6	null	null	9.95
220100100043	total_retail_price	6	null	null	108.6
220100100043	costprice_per_unit	6	null	null	7.8
220100100044	total_retail_price	6	null	null	612.6
220100100044	costprice_per_unit	6	null	null	48.65
220100100064	total_retail_price	6	null	null	139.2
220100100064	costprice_per_unit	6	null	null	11.7

17. Close the Table Viewer.

18. Click **Save As**.
    a. Type **transpose large_customer_orders** in the **Directive name** field.
    b. Click **OK**.
19. Click **Back to Directive** to return to the SAS Data Loader console.

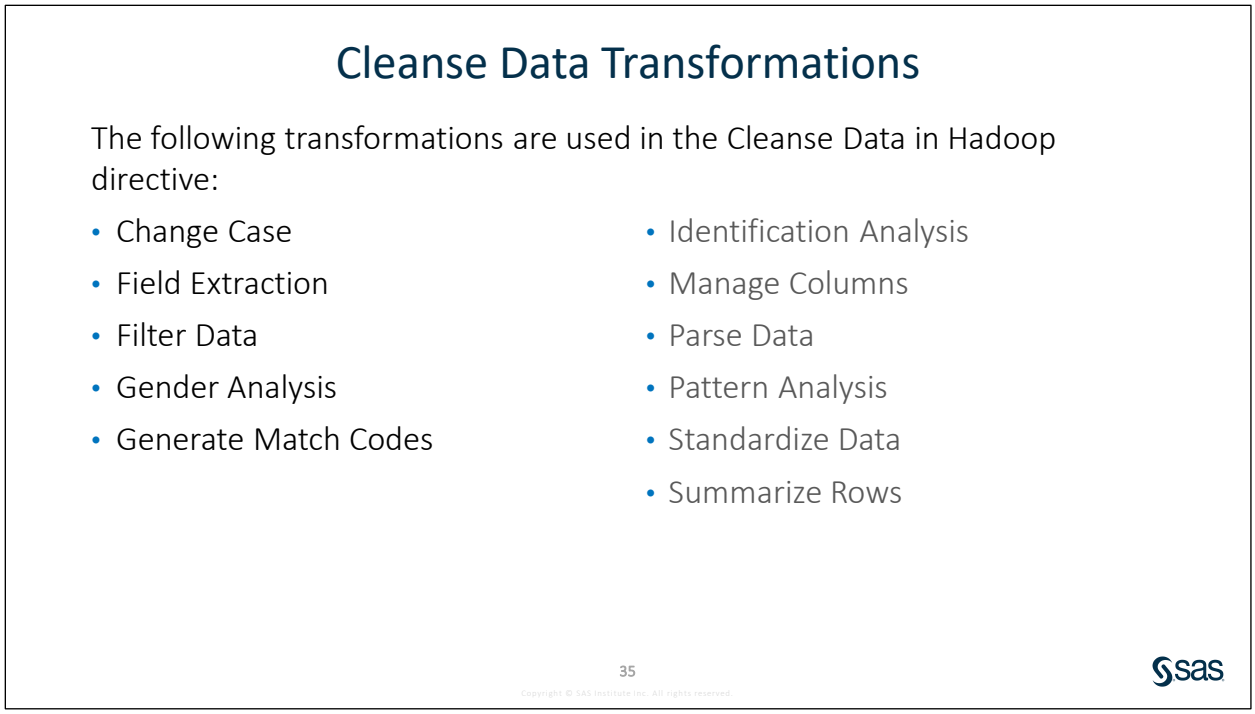
End of Demonstration

1.4 Cleanse Data



The third step of the methodology, Cleanse Data, involves the following types of activities:

- change the case of data values
- perform gender analysis from a name field
- generate fuzzy logic “match codes” to represent a text string
- identify the type of data in a field by performing identification analysis
- parse data elements into individual “tokens”
- perform pattern analysis on a data value
- standardize data values to a consistent representation
- …. and more

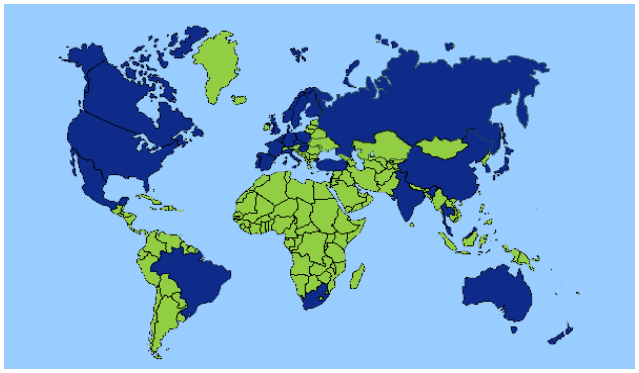


A number of transformations are available in the Cleanse Data in Hadoop directive. They enable you to perform tasks, such as the following:

- **Change Case** – enables you to change the case of data to a desired standard casing.
- **Field Extraction** – enables you to extract fields of data from a column.
- **Gender Analysis** – enables you to identify the gender of a person based on the first name.
- **Generate Match Codes** – enables you to create match codes for selected values in the table.
- **Identification Analysis** – enables you to identify the semantic data type of text in a column.
- **Parse Data** – enables you to break a text string into its individual components called “tokens.”
- **Pattern Analysis** – enables you to examine the pattern of characters in a string.
- **Standardize Data** – enables you to ensure a consistent representation of text strings.
- **Filter Data** – enables you to select the rows of data to include.
- **Summarize Rows** – creates a new row with data that is summarized in selected columns.
- **Manage Columns** – selects columns to include in a target table.

What Is a Quality Knowledge Base (QKB)?

Quality Knowledge Base	Collection of files that perform the "cleansing" of data, which could involve parsing, standardization, matching, and more



Locale Support

The QKB is a collection of files and algorithms that perform a wide variety of data cleansing tasks. The algorithms in the QKB are surfaced to the end user in the form of "definitions." There are specific definitions for the different data cleansing tasks that you might want to perform.

- Parsing
- Standardization
- Fuzzy matching
- Identification Analysis
- Gender Analysis
- … and more.

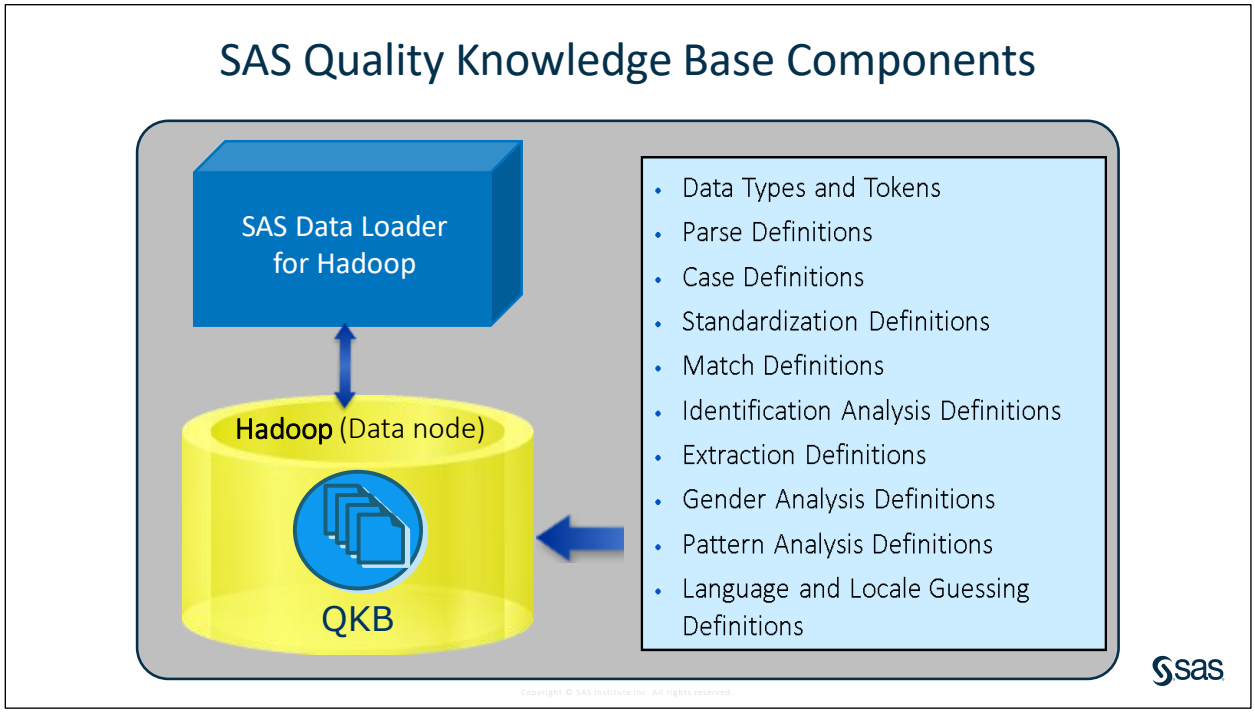
The definitions in the QKB are organized by the type of data they are designed to cleanse. For example, the QKB for Contact Information supports management of commonly used contact information for individuals and organizations, including the following categories:

- name
- address
- company name
- phone number
- email
- … and more.

The definitions in the QKB are geography-specific as well as language-specific, creating what is referred to as a "locale". Available locales are indicated by the darker regions in the map above. Every locale supported by the QKB has its own set of unique definitions for that geographic location and language(s).

Locales are hierarchical in nature. For example, Latin 1 is the parent of English, and English is the parent of English (United States) and English (Great Britain). This means that definitions in English are accessible from both English (United States) and English (Great Britain) locales, but would typically not be available in non-English locales (for example, the Spanish (Mexico) locale).

Note: When a child locale has a definition of the same name and type as the parent, the child's definition overrides the parent's, so the parent's definition is no longer accessible from the child.



The QKB is a set of files that contain rules, expressions, and reference data that are combined to analyze and transform text data in various SAS products. The QKB can be modified in order to create custom data quality logic for operations such as custom parsing or matching rules.

In the context of the QKB, a *data type* is an object that represents the semantic nature of a data value. A data type serves as a container (or grouping) for metadata that is used to define data cleansing algorithms (called definitions). Many data types are provided in the QKB, such as address, city, data, name, organization, phone, email, state/province, and so on.

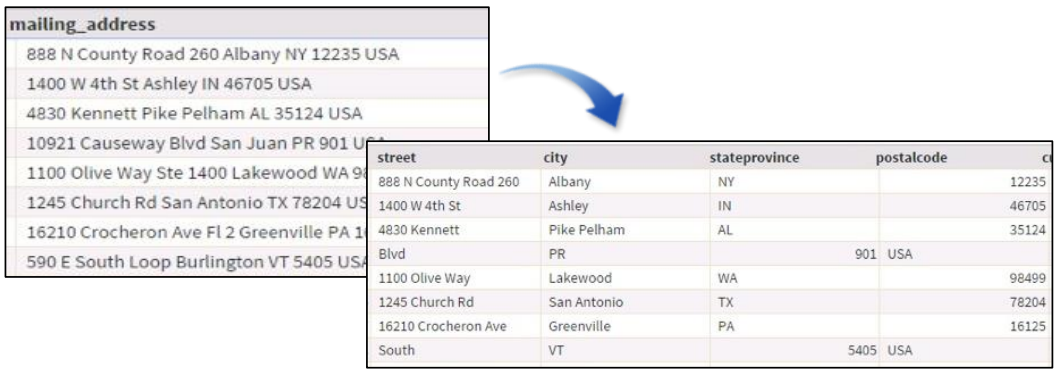
Each data type consists of one or more tokens. A *token* in the context of the QKB is an "atomically semantic" component of a data type. In other words, tokens represent the smallest pieces of a data type that have some distinct semantic meaning. A token does not necessarily need to contain only one word. For example, the “street name” token might contain multiple words (for example, New Hope Church, Martin Luther King, Dutch Creek). Tokens are beneficial throughout the data quality process. Many of the definitions for a given data type use the tokens as input. (For example, the tokens coming out of an ***Address Parse definition*** are used as input by the ***Address Match definition***.)

A *definition* is a set of steps for processing a piece of data. Several types of definitions can be created in the QKB. Generally, a definition is tied to a particular data type. Certain definitions use a data type's tokens. Some might use only a subset of the data type's tokens. Some types of definitions use other definitions as well.

Parse Data Transformation

The Parse Data transformation provides these capabilities:

- extracts tokens from a source column and adds the token to a new column
- uses tokens that represent a meaningful subset of the data value that provides a basis for analysis and reporting



mailing_address
888 N County Road 260 Albany NY 12235 USA
1400 W 4th St Ashley IN 46705 USA
4830 Kennett Pike Pelham AL 35124 USA
10921 Causeway Blvd San Juan PR 901 US
1100 Olive Way Ste 1400 Lakewood WA 9
1245 Church Rd San Antonio TX 78204 US
16210 Crocheron Ave Fl 2 Greenville PA 1
590 E South Loop Burlington VT 5405 USA

street	city	stateprovince	postalcode	c
888 N County Road 260	Albany	NY	12235	
1400 W 4th St	Ashley	IN	46705	
4830 Kennett	Pike Pelham	AL	35124	
Blvd	PR	901	USA	
1100 Olive Way	Lakewood	WA	98499	
1245 Church Rd	San Antonio	TX	78204	
16210 Crocheron Ave	Greenville	PA	16125	
South	VT	5405	USA	

The Parse Data directive enables you to parse data values in the Hadoop data table. Using this directive, you can parse data elements into individual tokens and store them as new columns in the Hadoop tables.

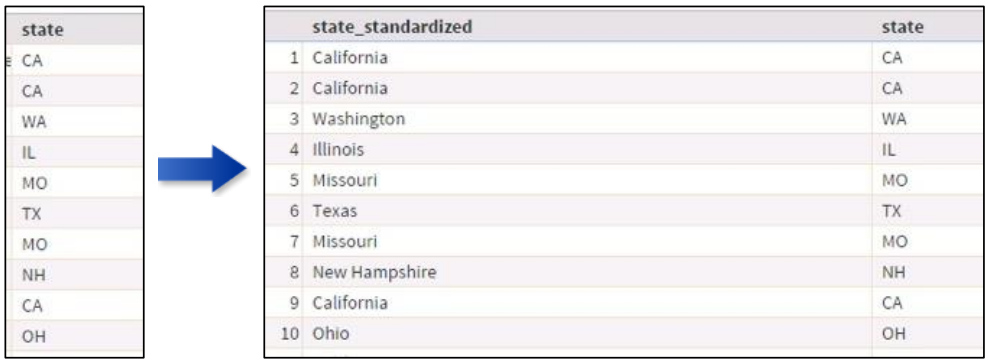
Note: A *token* is the smallest representative (semantically atomic) portion of a data string. (For example, a person's name might be divided into Name Prefix, Given Name, Middle Name, Family Name/Surname, Name Suffix, and Name Appendage tokens.)

Note: The fact that tokens can be written to new columns means that they can potentially be used in analysis and reports.

Standardize Data Transformation

The Standardize Data transformation provides these capabilities:

- converts values in a column to a consistent format
- uses the QKB definitions for a given locale



state
CA
CA
WA
IL
MO
TX
MO
NH
CA
OH

	state_standardized	state
1	California	CA
2	California	CA
3	Washington	WA
4	Illinois	IL
5	Missouri	MO
6	Texas	TX
7	Missouri	MO
8	New Hampshire	NH
9	California	CA
10	Ohio	OH

39

The Standardize Data transformation enables you to standardize data values in the Hadoop cluster. Using this transformation, you can transform data elements into a standard representation, and create consistency across data values in your Hadoop tables.

Standardizing data values before analysis and reporting greatly improves the accuracy and usefulness of the analysis and reports.

The Standardize Data transformation uses standardization definitions from the QKB to transform data.

Generate Match Codes Transformation

The Generate Match Codes transformation provides the following capabilities:

- generates a similar match code value when two strings are not exact character matches
- uses a data type definition and match sensitivity setting to determine the degree of similarity between strings to generate a match code value

Sensitivity 50

Bob Smith	→	4B~$$$$$$$$$$$$$$$$$$$$$$
Robert Smith	→	4B~$$$$$$$$$$$$$$$$$$$$$$$

Sensitivity 95

Bob Smith	→	4B7~2$$$$$$$$$$M@M$$$$$$$$
Robert Smith	→	4B7~2$$$$$$$$$$Y@M_Y~$$$$$$

The match code has the following attributes:

- presents an encoded representation of a text string
- enables you to identify similar, but not identical, data values within and across data sources
- is consistent across data sources and time
- can be used for house holding, de-duplicating data, surrogate-key joins, SQL lookups, and more

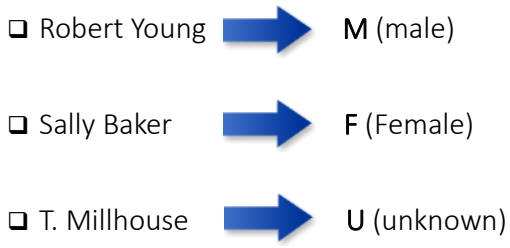
Match codes are especially useful when you have two data elements that are very similar, but not exactly the same. Although it might be easy for humans to understand that these two things are the same, it is very difficult for a computer to know that they are the same. Match codes are unique strings that are based on the actual data value and a fuzzy-matching sensitivity setting.

The sensitivity is used to express how exact you want to be when you generate the match code.

Gender Analysis Transformation

The Gender Analysis transformation provides the following capabilities:

- determines the gender of an individual based on the specified name field
- identifies **male**, **female**, or "**unknown**" based on the name field
- creates a new data element that can be used in analysis and reporting



41

The Gender Analysis transformation enables you to determine the gender from a column that contains name data. In the example, the appropriate gender analysis definition was chosen from the English (United States) locale to analyze gender on a column that contains people's names. The value that is returned from the analysis is written to a new column in the target table. The table can then be used for analysis and reporting.

Identification Analysis Transformation

The Identification Analysis transformation provides the following capabilities:

- determines the *type of data in a specified column*
- uses QKB definition types based on locale and data types for
  - contact info
  - phone (validation of format)
  - date (validation of format)
  - email address (validation of format)
  - email address (country identification)
  - offensive words
  - field name.

42

Identification analysis is especially useful when you need to determine the type of data that is contained in a column. Using identification analysis definitions in the QKB, the Identification Analysis transformation reads in a data string and uses provided algorithms to determine the most likely data type.

Change Case Transformation

The Change Case transformation provides the following capabilities:

- standardizes the case of column data into a consistent format
- uses a QKB definition based on locale and data type
- creates a new column to contain the cased output data

❑ **lower** – Change all characters values to lowercase values.

Example: John Smith ⇨ john smith

❑ **UPPER** – Change all character values to uppercase values.

Example: John Smith ⇨ JOHN SMITH

❑ **Proper** – Change the first character of the words in a string to uppercase.

Example: john smith ⇨ John Smith

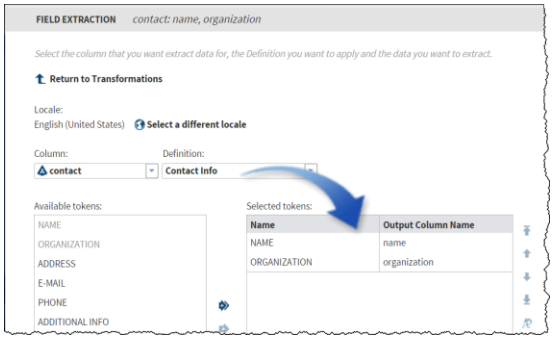
43

sas

Field Extraction Transformation

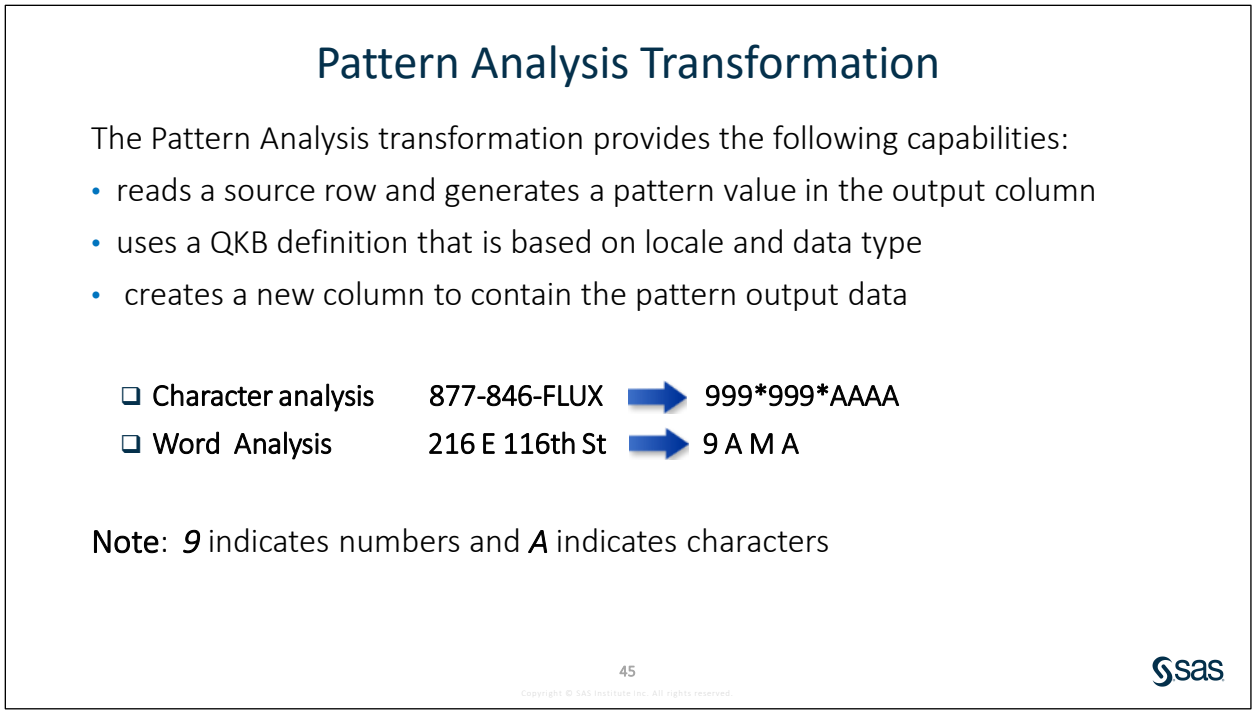
The Field Extraction transformation provides the following capabilities:

- It is used to copy tokens from the source column to a new target column based on a QKB definition.
- Tokens represent the types of content that can be extracted using an extraction definition in the QKB.
- Extraction definitions are based on locale and data type.



44

sas



The following pattern analysis definitions are supported:

Character – generates patterns that represent the types of each character in the source. ***A*** indicates uppercase, ***a*** indicates lowercase, ***9*** indicates numbers, and ***** indicates other (punctuation, and so on). Blanks in the source are replicated as blanks in the pattern.

- 877-846-FLUX 999*999*AAAA

Character (Script Identification) – generates patterns that identify the Unicode character set of each character in the source.

- (7F, SAS Institute）スズキイチロウ *9L* LLL LIIIIIII*アアアアアアア

Word – generates patterns that represent the types of words in the source. ***A*** represents alphabetic words, ***9*** numeric, ***M*** mixed, and ***** other.

- 216 E 116th St 9 A M A

Word (Script Identification) – generates patterns that represent the Unicode character set of each word in the source. ***W*** indicates a potentially invalid word that contains multiple character sets.

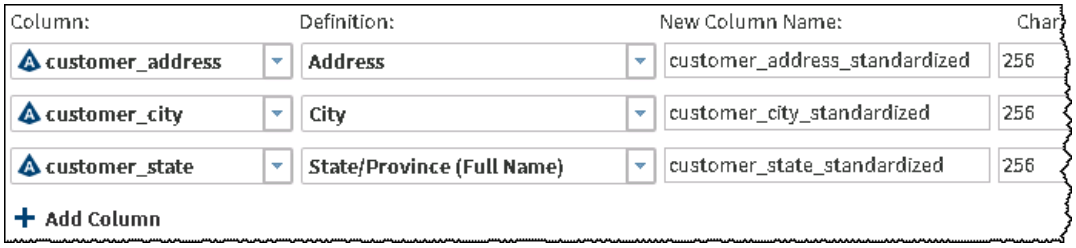
- （7F, SAS Institute）スズキイチロウ *9L* L L*ア

Cleanse Data Using the Parse and Standardize Transformations

This demonstration illustrates the use of the Parse and Standardize Data transformations to cleanse the **CUSTOMERS** Hadoop table. Using these transformations, you separate the data into meaningful fields and transform inconsistent data values into a reliable, consistent forms in a target table. To parse and standardize data in a Hadoop table, perform the following steps:

1. Verify that you are viewing the Data Loader console.
2. Select the **Saved Directives** directive.
   a. Select the **Cleanse Customers** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
3. Click the **Cleanse Data in Hadoop** directive.
4. Click **Settings**.
   a. Select **Hadoop Spark** in the **Preferred runtime target** pull-down menu.
   b. Click **OK**.
5. Click the **Customers** table, which was previously profiled and shown to contain inconsistent data.
6. Click **Next**.
7. Click the **Parse Data** transformation.
   a. Select **customer_name** in the **Column** pull-down menu.
   b. Select **Name** in the **Definition** pull-down menu.
      1) Double-click **Given Name** to move it from **Available tokens** to **Selected tokens**.
         Type **First_Name** in the **Output Column Name** field.
      2) Double-click **Family Name**.
         Type **Last_Name** in the **Output Column Name** field.
   c. Click **Add Another Transformation**.
8. Click **Standardize Data** transformation.
   a. Select **customer_address** in the **Column** pull-down menu.
      Select **Address** in the **Definition** pull-down menu.
   b. Click **Add** Column.
   c. Select **customer_city** in the **Column** pull-down menu.
      Select **City** in the **Definition** pull-down menu.
   d. Click **Add Column**.
      Select **customer_state** in the **Column** pull-down menu.

Select **State/Province(Full Name)** in the **Definition** pull-down menu.

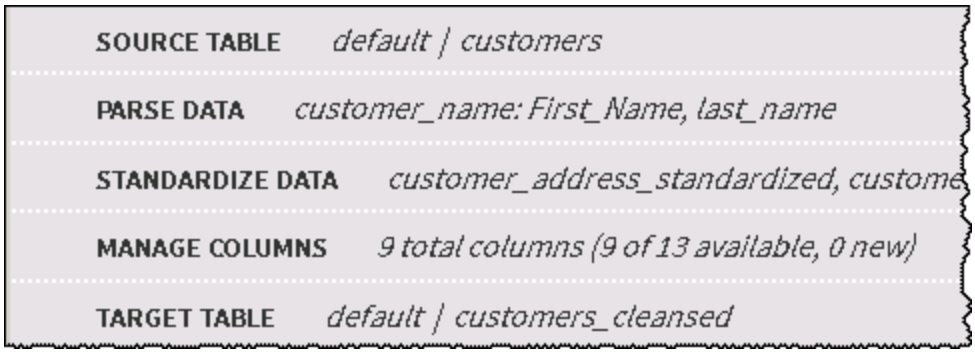


9. Click **Next**.
10. Click **Add Another Transformation**.
11. Click **Manage Columns**.
    a. Click (double arrows) to clear the default Selected Columns table.

       Click **OK** on the Delete columns dialog box to remove all columns.
    b. Double-click the following columns in the table to add them back to the Selected Columns table:

Source Name	Target Name
customer_id	customer_id
first_name	first_name
last_name	last_name
customer_address_standardized	address
customer_city_standardized	city
customer_state_standardized	state
customer_postal_code	postal_code
customer_gender	gender
customer_birthday	birthday

12. Click **Next**.
13. Click **New Table**.
    a. Type **Customers_Cleansed** in the **New table** field.
    b. Click **OK**.
14. Click **Next**.



15. Click **Start Transforming Data**.
16. Click **View Results** and verify that the data is parsed as expected.

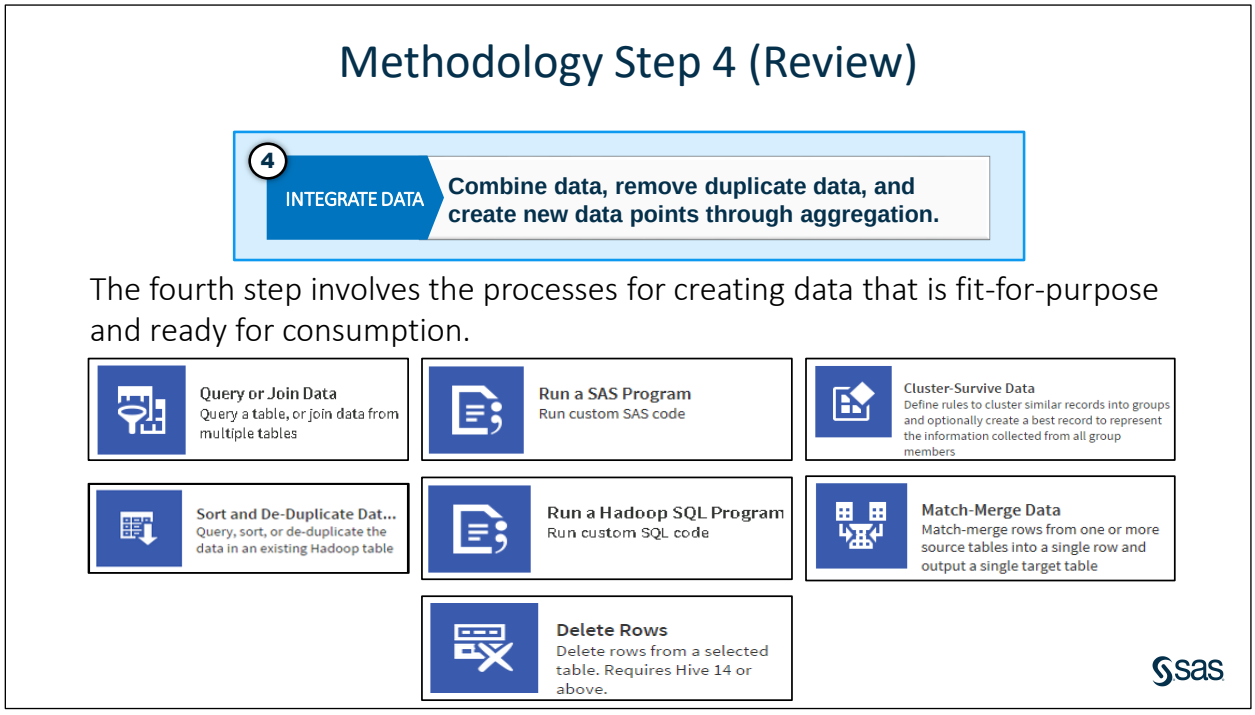
customer_id	first_name	last_name	address	city	state	po
7059	Patricia	Kukahiko	20 Circle (On the G	Vestal	New York	
7077	Welf	Stevette	139 Circlebank Dr	Vestal	New York	
7088	Jill	Halim	113 East St	Binghamton	New York	
7101	Blake	Davenport	927 Harps Mill Rd	Endicott	New York	
7108	Christine	Deep	68 Hedgerow Dr	Castle Creek	New York	
7129	Robert	Sullins	832 Antique Ln	Fort Lauderdale	Florida	
7131	Katherine	Raybould	375 Apache Ln	Pompano Beach	Florida	
7132	Harpreet	Mcgee	19 Appleton Dr	Fort Lauderdale	Florida	
7143	Sharon	Grlj	64 Arbor View Dr	Fort Lauderdale	Florida	
7149	Betty	Prince	204 Arbordale Ct	Pompano Beach	Florida	

Note: The customer_name has been parsed into two fields and address/city/state have been standardized.

17. Close the Table Viewer.
18. Click **Save As**.
    a. Type **cleanse customers** in the **Directive name** field.
    b. Click **OK**.
19. Click **Back to Directives**.

End of Demonstration

1.5 Integrate Data

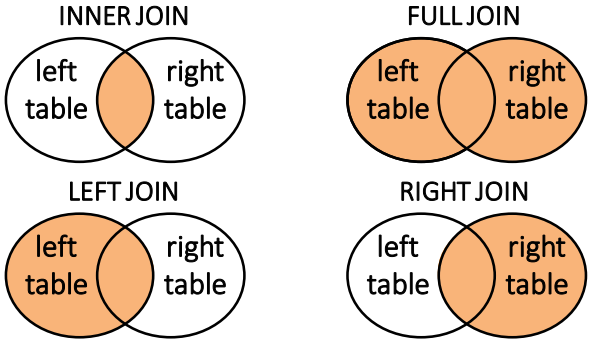


Step 4 is the process of creating data that is fit-for-purpose. In this step, you might want to join data sources to support an analytical exploration or output report format. Any data in the existing tables that is not required for a subsequent operation should be removed from the final table that is loaded to the LASR Analytic Server or copied to a SAS data set or relational database. To perform these integration processes, the Query or Join Data in Hadoop, Sort and De-Duplicate Data in Hadoop, and Delete Rows directives are available.

Joining Data in Hadoop Capabilities

The Query or Join Data directive provides these capabilities when you join tables:

- combine source tables in Hadoop
- create one or many "join-on" column statements
- intuitive joining based on source table column names



49

sas

What Is Clustering Data?

Clustering data provides the following capabilities:

- identify unique entities after you determine which fields can be used to ***identify related records***
- define a set of cluster rules to ***partition records into related sets of entities*** with a unique cluster ID based on those rules

name	email1	email2	
Alice	alice@alice.net	alice@alice.com	SET 1
Alice	(null)	Alice	
Bob	bob@bob.com		SET 2
Bob	robert@robert.com		

sas

Entities is a term that is used to identify the group of combined rows of records based on fields and rules.

What Is Survivorship?

Survivorship processing provides the following capability:

- ***create a survivor record*** after the clustering process identifies a set of records that are logically related

Name	Street	City	Updated	Email
Alice	101 Main Street	Albuquerque	June 26, 2015	(null)
Alice	205 North Avenue	Tucson	May 15, 2012	alice@alice.net

Name	Street	City	Updated	Email
Alice	101 Main Street	Albuquerque	June 26, 2015	alice@alice.net

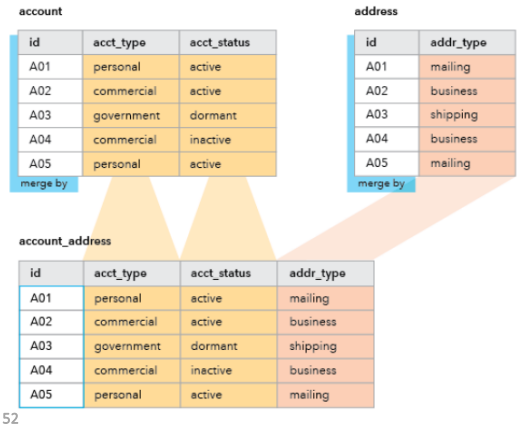
51

Note: Leading and trailing spaces are trimmed from fields when survivorship rules are evaluated.

Match-Merge Directive Capabilities

Match-merge provides these capabilities:

- combine numeric or character columns, common to all sources, into a single target table using values of one or more matched columns
- merge rows in two or more source tables when values match in a specified merge-by column



52

Run a SAS Program Directive

The Run a SAS Program directive provides the following capabilities:

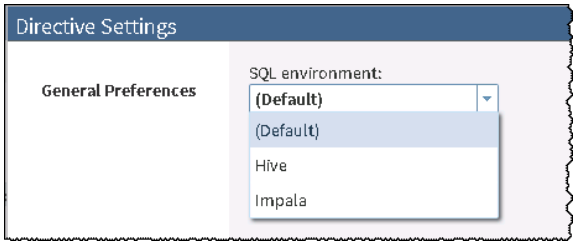
- This directive executes all SAS programs in Hadoop while leveraging the SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop.
- Programs are written with ultra-efficient SAS DS2 language elements.
- DS2 elements combine the extensive capabilities of SAS DATA step processing with the ability to execute in parallel in the Hadoop cluster.
- The directive supports common SAS coding statements and procedures such as the LIBNAME statement and PROC SQL.

53

Run a Hadoop SQL Program Directive

The Run a Hadoop SQL Program directive provides the following capabilities:

- uses the Hive SQL editor that contains HiveQL function resources to generate Hive query (HiveQL) programs to run in Hadoop
- supports copying and pasting existing code into the SQL editor
- leverages user credentials that are configured in the Hadoop Configuration panel during the submission of code
- supports the Cloudera Impala SQL environment, if configured, for performance gains

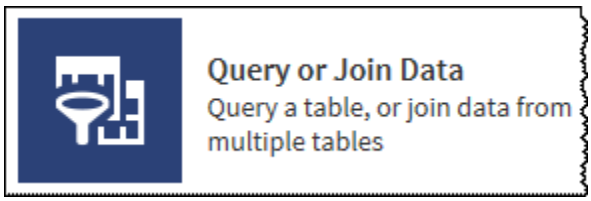


54

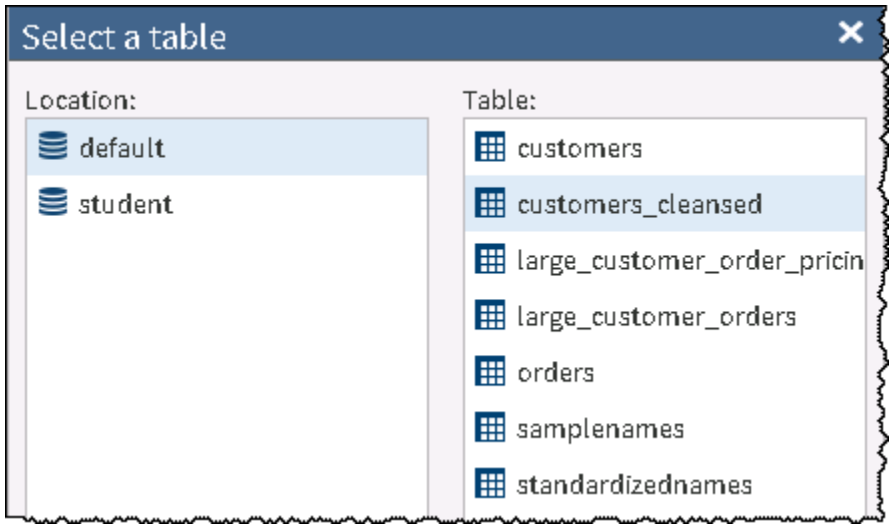
Integrate Data in Hadoop

The following demonstration reviews the use of the Query and Join Data directive to perform an inner join with the **CUSTOMERS_CLEANSED** and **LARGE_CUSTOMER_ORDERS** Hadoop tables on the **Customer_ID** field to build a customer sales order table for large orders.

1. Select the **Saved Directives** directive.
   a. Select the **Join Customers_Cleansed and Large_Customer_Orders** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
2. Select the **Query or Join Data** directive.



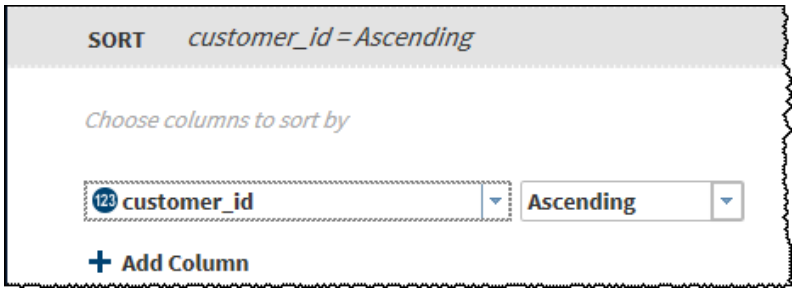
3. Click ... (ellipsis) to open the table selector.
   a. Select **customers_cleansed**.



   b. Click **OK**.
4. Click **Add join.**
5. Click ... (ellipsis) to select a table on which to join.
   a. Select **large_customer_orders**.
   b. Click **OK**.
6. Select **default.customers_cleansed.customer_id** in the first **Join on** field.

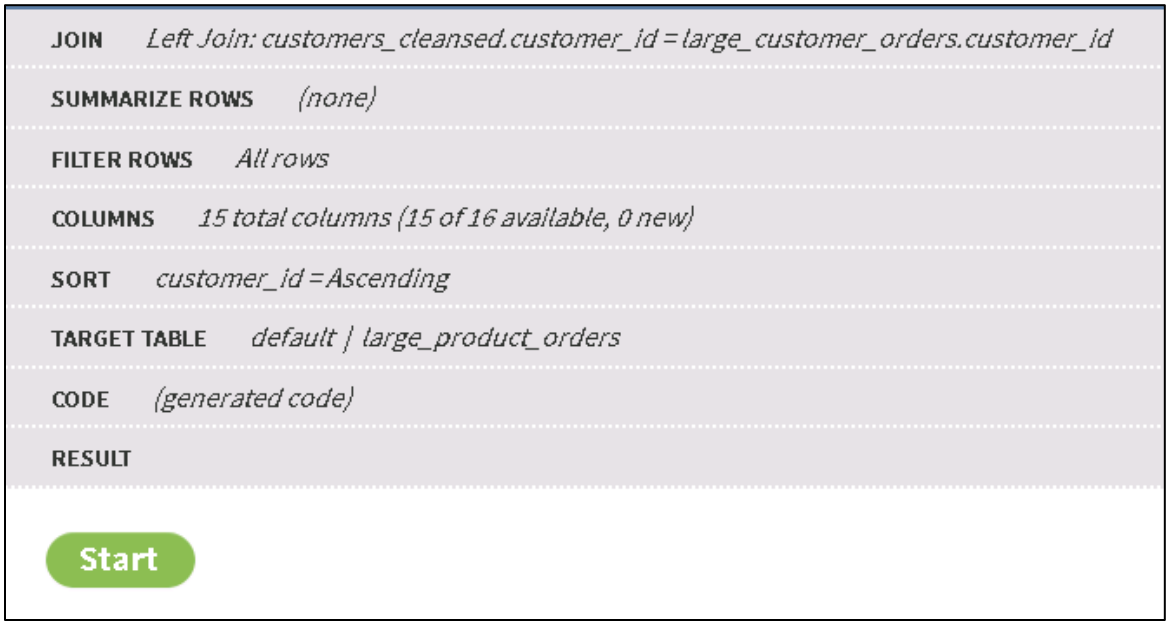
7. Select **default.large_customer_orders.customer_id** in the second **Join on** field.
8. Select **Left Join**.

9. Click **Next**.
10. Click **Next** in the SUMMARIZE ROWS task.
11. Click **No duplicate rows** on the FILTER ROWS task.
12. Click **Next**.
13. Click **Next** in the COLUMNS task.
14. Select **customer_id** and **Ascending** in the SORT task.

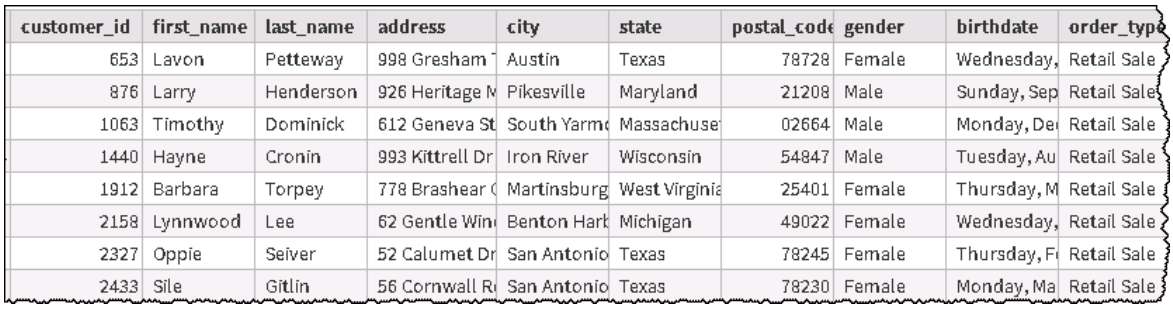


15. Click **Next**.
16. Click **New Table**.
    a. Enter **large_product_orders** in the **New table** field.
    b. Click **OK**.
17. Click **Next**.

18. Click **Next** in the CODE task.



19. Click **Start**.
20. Click **View Results** after the directive successfully runs to view the results in the Table Viewer.

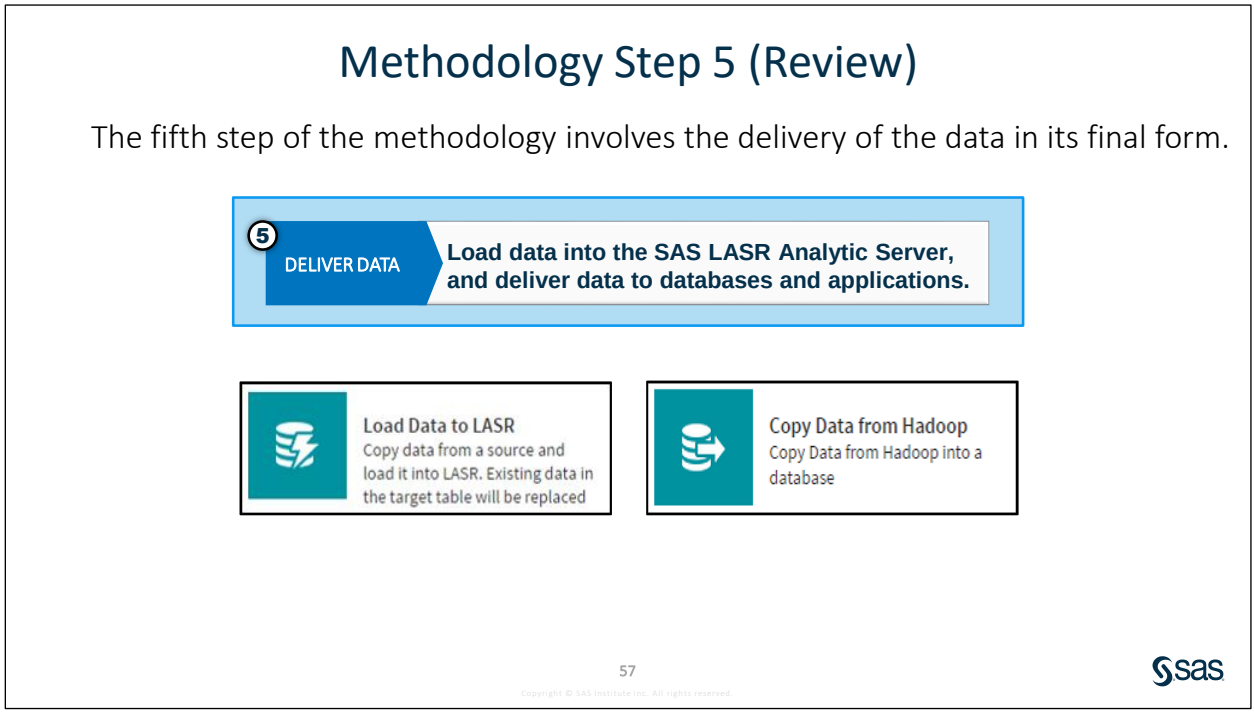


customer_id	first_name	last_name	address	city	state	postal_code	gender	birthdate	order_type
653	Lavon	Petteway	998 Gresham	Austin	Texas	78728	Female	Wednesday,	Retail Sale
876	Larry	Henderson	926 Heritage M	Pikesville	Maryland	21208	Male	Sunday, Sep	Retail Sale
1063	Timothy	Dominick	612 Geneva St	South Yarmo	Massachuse	02664	Male	Monday, De	Retail Sale
1440	Hayne	Cronin	993 Kittrell Dr	Iron River	Wisconsin	54847	Male	Tuesday, Au	Retail Sale
1912	Barbara	Torpey	778 Brashear (	Martinsburg	West Virginia	25401	Female	Thursday, M	Retail Sale
2158	Lynnwood	Lee	62 Gentle Win	Benton Harb	Michigan	49022	Female	Wednesday,	Retail Sale
2327	Oppie	Seiver	52 Calumet Dr	San Antonio	Texas	78245	Female	Thursday, F	Retail Sale
2433	Sile	Gitlin	56 Cornwall R	San Antonio	Texas	78230	Female	Monday, Ma	Retail Sale

21. Close the Table Viewer.
22. Click **Save As**.
    a. Type **Join customers_cleanse and large_customer_orders** in the **Directive name** field.
    b. Click **OK**.
23. Click **Back to Directives**.

End of Demonstration

1.6 Deliver Data



Step 5 is the delivery of the data in its final form for analysis and reporting to other applications. It currently consists of two directives, Load Data to LASR and Copy Data from Hadoop. A common scenario might be to load large amounts of data into Hadoop for cleansing and transformation. Then the Load Data to LASR directive can load the modified data into a preconfigured SAS Visual Analytics LASR Server for analysis and reporting. The Copy Data from Hadoop directive could be used to load the modified data into SAS data sets or relational databases, such as Oracle or Teradata.

Load Data to LASR Directive Capabilities

The Load Data to LASR directive provides the following capabilities:

- supports any SAS LASR Analytic Server that is configured in SAS Metadata
  - loads a single table into a SAS LASR Analytic Server that is optimized for symmetric multi-processing (SMP)
  - loads a single table into a multi-node SAS LASR Analytic Servers grid that is configured and optimized for massively parallel processing (MPP)
- creates a new table or replaces an existing table in a SAS folder location that is defined in SAS Metadata

58

The steps for loading Hadoop data to a preconfigured SAS LASR Analytic Server are very simple. You select your source and specify the LASR server. In SAS Data Loader, more than one LASR server can be configured, such as one for development, one for testing, and one for production. Lastly, define the target table name and select **Start loading data**.

Copy Data from Hadoop Directive

The Copy Data from Hadoop directive provides the following capabilities:

- copies data from Hadoop to any configured SAS target location that is configured in SAS Metadata, including SAS libraries and relational databases
- supports parallel copy processing for specific target locations, such as Oracle relational databases that use Hadoop Sqoop JDBC connections
- uses the SAS Workspace Server to connect to SAS data sets and to copy the Hadoop table to the SAS library location

Note: SAS target tables need to be registered in SAS Metadata after loading to a SAS library location.

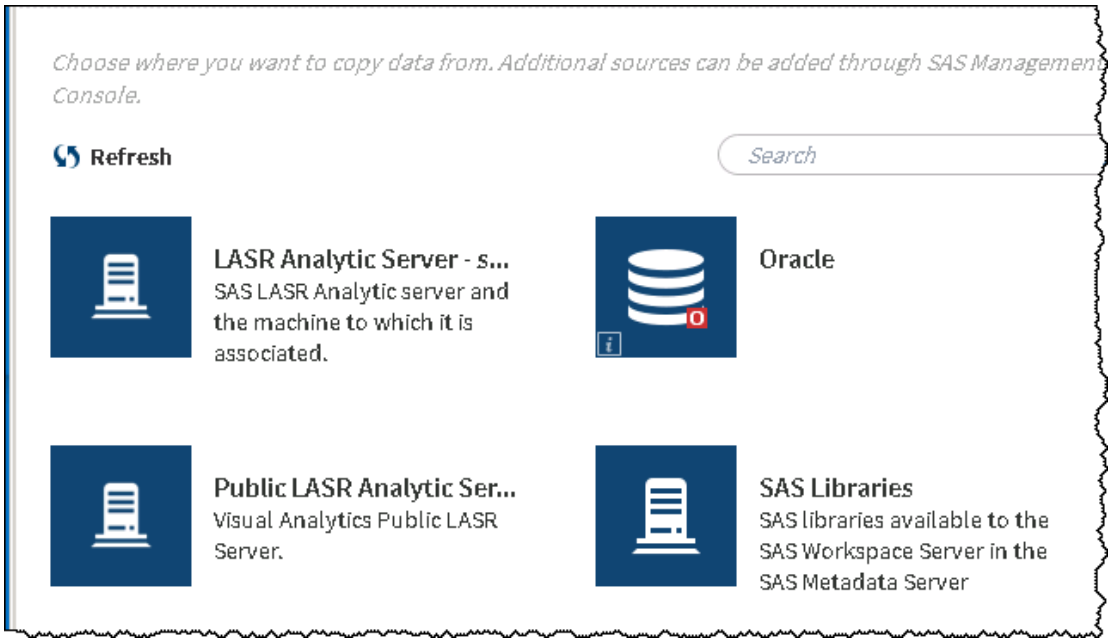
59

When SAS data sets are specified, Sqoop is not used as it is when using a relational database such as Oracle. Copying data from Hadoop to SAS uses the SAS Workspace Server, a LIBNAME statement, and PROC SQL code.

Deliver Data for Analytical Analysis and Reporting

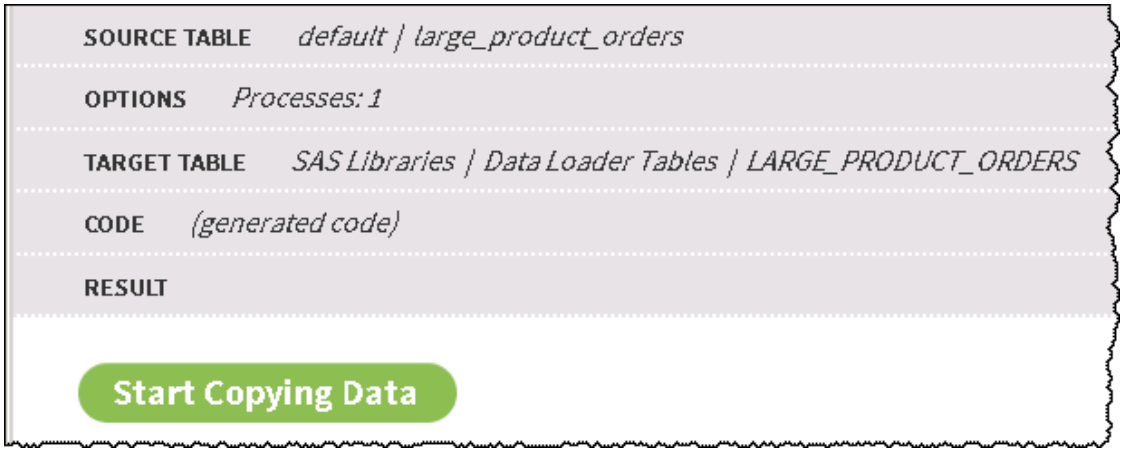
This demonstration illustrates the use of the Copy Data from Hadoop directive to copy a Hadoop table to a SAS table.

1. Verify that you are viewing the Data Loader console.
2. Select the Saved Directives directive.
   a. Select the **Copy Large_Product_Orders from Hadoop** directive.
   b. Review the tasks in the directive. Below are the steps to create the custom directive.
3. Select the **Copy Data from Hadoop** directive.
4. Select the **large_product_orders** table.
5. Click **Next**.
6. Select the **SAS Libraries** database.



7. Click **Data Loader Tables**.
8. Click **New Table**.
   a. Enter **large_product_orders** in the **New table** field.
   b. Click **OK**.
9. Click **Next**.

10. Click **Next** in the CODE task.



11. Click **Start copying data**.
12. Click **View Results** and verify the SAS target table.

customer_i	first_name	last_name	address	city	state	postal_code	gender
653	Lavon	Petteway	998 Greshan	Austin	Texas	78728	Female
876	Larry	Henderson	926 Heritage	Pikesville	Maryland	21208	Male
1063	Timothy	Dominick	612 Geneva	South Yarmo	Massachuse	02664	Male
1440	Hayne	Cronin	993 Kittrell D	Iron River	Wisconsin	54847	Male
1912	Barbara	Torpey	778 Brashea	Martinsburg	West Virginia	25401	Female
2158	Lynnwood	Lee	62 Gentle W	Benton Harb	Michigan	49022	Female
2327	Oppie	Seiver	52 Calumet	San Antonio	Texas	78245	Female
2433	Sile	Gitlin	56 Cornwall	San Antonio	Texas	78230	Female

13. Close the Table Viewer.
14. Click **Save As**.
    a. Type **copy large_product_orders to SAS data set** in the **Directive name** field.
    b. Click **OK**.
15. Click **Back to Directives**.

End of Demonstration