Paper SAS4356-2020

Data Catalog: The Binding Governance Tool for a Successful Digital Transformation

Vincent Rejany, SAS Institute Inc.

ABSTRACT

No one today challenges the power of data as a driver for growth and innovation. One after another, companies are starting their digital transformation to add value to their data and to build a data-driven strategy. Unfortunately, most organizations govern their data in an ad hoc or firefighting manner across different parts of the business, and most of the time only within IT. Mapping data by building a data catalog is one of the first steps toward more governance and sustainability. A data catalog is a repository of metadata, centralizing information about data sources, schemas, tables, and columns extended. In this paper, we discuss about the importance of data catalogs how to build a data catalog over the SAS®9 Platform using open-source technologies.

INTRODUCTION

Data catalogs, data catalogs, data catalogs! A new buzz word? What happened over the last few years to put metadata management back into the headlines and for it to climb the famous Gartner Hype Cycle for Data Management for becoming "the new black" according to the advisory firm? 451 Research has even stated that “There is a case to be made that the data catalog is the most important data management breakthrough to have emerged in the last decade” (Aslett 2018). Why is there such enthusiasm? Why are data catalogs the new data management rock star? Are they different from traditional metadata management and data dictionaries that millennial data scientists cannot know?

Let's immediately break the myth, technically speaking. They are not. It is only a question of exposure to a new audience and therefore, a less technical user experience. However, I assume that the major driver resides in the awareness that accumulating data without proper governance can only result in one obvious consequence: a B I G mess. A mess that includes an ever-increasing risk in terms of security, decision making process, and in the end, a lack of trust in data. For example, in the early 2000s, as the internet was growing exponentially, there was this empty space that Google took, and became the reference tool for searching, finding, and evaluating content for relevancy and in the end for telling us what we should look at. Similarly, as organizations today struggle to maximize the value derived from their ever-growing volumes of data, the focus is no longer on "having" data, but on "knowing" your data in order to break the 80-20 ratio between (lost) time spent in searching data and doing data preparation versus real analytics and decision making.

In this paper we look at explaining why data catalogs are data governance rock stars and work! It also explores what a data catalog is and what is does as well as everything you should know about accessing and exporting metadata from the SAS9 platform in order to create your data catalog.

DATA CATALOG SMELLS LIKE TEEN SPIRIT!

DON'T LOOK BACK IN ANGER CHIEF DATA OFFICER!

Yes Chief, don't look back and even in front of you in anger. The difficulties of data management have intensified at a steady pace over the past several years. Your organization is struggling to get and maximize the value from its data, and the following are three main reasons for this that explain why data catalogs have been emerging:

Figure 1. Data Catalog Adoption Drivers

- Data proliferation: Your organization has never managed so much data, and more data that is spread over multiple locations.
- Regulatory pressure: Your organization is now heavily scrutinized by industry, state, and national regulations (GDPR, CCPA, PIPA, PIPEDA, KVKK, and so on) that are asking for transparency and accountability.
- Data democratization: Your data consumers are requesting more and more data, but at the same time they want to know where it comes from, and how reliable it is. They ask for the end of tribal knowledge and for the advent of data democracy.

These three drivers explain why data catalogs have become so popular versus the former metadata management approach. End users are no longer able to spend more time looking for relevant, adequate, up to date, qualitative, and reliable data, than they spend analyzing data. Data catalogs are key in self-data service strategies by being the entry point for next valuable actions with data. On the other hand, by identifying sensitive data before it's applied to business analytics, data catalogs reduce the impact of potential breaches while meeting all industry and government regulations.

DO I WANNA KNOW DATA CATALOGS?

Data Catalogs! So, What Are They?

In its report, "Data Catalogs are the New Black in Data Management and Analytics," Gartner gives the following definition: "A data catalog maintains an inventory of data assets through the discovery, description, and organization of datasets. The catalog provides context to enable data analysts, data scientists, data stewards, and other data consumers to find and understand a relevant dataset for the purpose of extracting business value" (Gartner, Inc. 2017).

Gartner's definition does not really defer from historical metadata management as it does not focus on what makes data catalogs today so trendy: automation and collaboration. Excel-based or IT-driven data dictionaries are over, and the amount of data is too important and does require automation for scaling. Data consumers want to access data and to enrich, comment, and challenge the use and the quality of data.

Let's dare to give a definition: "A data catalog is an automated collection of metadata, combined with data management and search tools, that helps analysts and other data users to find the data that they need. It also serves as an inventory of available data and provides

information to evaluate the fitness of data for intended uses." In few words, a data catalog is your organization metadata social network!

Data catalogs are one of the main pillars of agile data governance as they allow organizations to create and make available for a non-technical audience a snapshot of their entire information ecosystem. Data cataloging accelerates analysis by minimizing the time and effort that analysts spend finding and preparing data. Anecdotally, it is said that 80% of self-service analysis without a data catalog is spent getting the data ready for analysis. Using the data catalog can cut that percentage from 80% to 20%. By providing a good understanding of the information present in an organization's data catalogs supports digital transformation strategies.

Data Catalogs' Objectives

Analytics can get you answers from data. However, only a data catalog will tell you where to find that data and everything you should know about it!

In some businesses where information is still too siloed, users have challenges when finding and identifying data they can trust and usually come with the following questions: Where can I find data? Who uses it? What are my goals? Are they of quality? By centralizing data knowledge and through a simple UI that allows you to search for data sets for reporting, analysis, integration, and data migration projects, data catalogs intend to do the following:

- allow data citizens to find the data they need in an efficient way
- empower organizations to quickly invent, discover, manage, and understand all their data
- move from tribal to centralized and crowdsource knowledge
- ingest new data sets and the use of new of data faster
- become the foundational layer for driving data governance, quality, and information security policies
- foster collaboration between business users and IT and to contribute to the shared understanding of the information

In terms of benefits, data catalogs contribute to increasing efficiency, as they allow analysts to short cut the time, they need to qualify the correct data. They also support data governance and risk mitigation by identifying personal and sensitive data, and by allowing you to establish and spread best practices in terms of data management and data quality. Finally, data management is simplified as new data sources can be onboarded more quickly and key assets can be easily identified and monitored, as redundant and untapped data can be detected and remediated. In the end, the data ecosystem gets rationalized and more agile.

START ME UP: DATA CATALOG CLASSIC FEATURES

Let's open the beast. What is there in a data catalog? Of course, there are different approaches of performing data cataloging depending on software vendors, but most of the existing solutions rely on the following four main components:

- A flexible data model for storing the metadata objects and their relationships
- A set of data discovery services that allow you to extract metadata from structured and unstructured data sources as well as enriching (discovering, scoring) metadata with additional information/insight
- Search and indexing services that allow you to make the information available as quick as possible and to formulate complex search queries

- An intuitive, easy to use, and collaborative user interface so that any kind of user can search and find what he or she needs

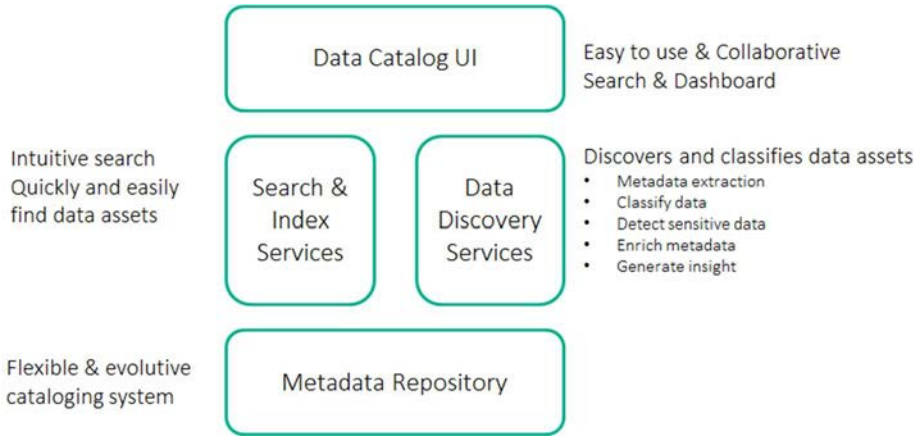


Figure 2. Data Catalog Components

Manage and Store Metadata in a Central and Agile Repository

Metadata must be stored and represented in a flexible format, allowing for performant searching and retrieval of content, a high volume of data, security, and versioning. Most of the data catalog solutions rely on a graph database as it allows you to put a high focus on the relationships between the data asset elements, and it facilitates the querying of the repository. The following is non-exhaustive list of the requirements or features that are expected from a data catalog in terms of metadata management:

- Metadata can either be extracted through metadata crawlers or created manually by data catalog administrators.
- Properties or attributes can be created by administrators on metadata objects A, which can be inherited from parent to children.
- Application domains (for example, HR, CRM, analytics, and so on) or departments for grouping data sources together can be created by administrators to organize the catalog.
- Purpose and appropriate use (provided by owners, stewards, or SMEs) can be defined by a data catalog user to indicate what the purpose of the asset is, and what its appropriate use would and would not be.
- Tables can be linked to other tables (that is, a fact table that is related to dimensions or reference data).

Extract Metadata and Discover New Insights

Technical Metadata Extraction

However, where do we get this metadata? Metadata about data sources that is structured or unstructured and connected to the platform must be extracted and made available to the end user. We will see later in this paper the different ways to get metadata from the SAS9 platform. Most of the data catalog solutions rely on content discovery techniques. They have different names (for example, crawlers, sniffers, bots, and connectors), but they have all the same objective: connecting to databases and either querying the database dictionary and system tables or running a set of queries for reading metadata. Depending on the size of the database or data lake storage, this step can take time. The refresh of metadata could be done automatically when a change is detected, scheduled, or executed manually.

Data management platforms and databases usually support metadata import/export capabilities through standards like CWM, SQL, DDL, or even specific text formatted files. It is often a time consuming and error prone approach, considering it is usually not automated. Finally, REST API integration as well as manual entries are also other options.

A data catalog must master metadata. If the same definition about a data asset (that is, a table) comes from multiple sources (for example, crawling, 3rd party, or manual entry), then it should be mastered and versioned in the data catalog, and the asset should only be presented one time.

The following tables illustrate some examples about the information that can be retrieved by content crawlers or imported into the data catalog.

• Unique identifier • Source or library name • Schema name • Table name / View name • Table label / View label • Alias • Description / Comments • Purpose • Owner(s) • Creator(s) • License • System table flag • Primary key(s) • Foreign key(s) and the tables related • Indexe(s) • Content provider	• Encoding • Language • Locale • Version of the metadata • Last modified date • Last modified by • Created on • Created by • Expires on / Retention • Table data last analyzed date and time • Tagging • Information privacy (not, private/personal, sensitive) • Information security (public, confidential, restricted) • Spatial coverage • Temporal coverage

Table 1. Non-exhaustive List of Table Metadata

• Unique identifier • Schema name • Table name • Column name • Ordinal position • Data type with length • Length of the column in bytes • Decimal precision • SAS format • Nullable flag • Column default value • Primary key flag	• Foreign key flag • Unique key flag • Sequence name • Check constraint flag • Column comments • Collation • Masked / Encrypted • Information privacy (not, private/personal, sensitive) • Information security (public, confidential, restricted)

Table 2. Non-exhaustive List of Column Metadata

Support Metadata Discovery

Getting technical metadata is one critical step but delivering insights about the data assets is also key for data catalogs. Data profiling metrics is a first level of analysis. Profiling is a discovery process of examining data by collecting statistics and information about that data to gain insight and uncover potential data quality issues. It involves gathering measurements for key metrics about specific data elements. Data profiling is a step in the assessment stage of the data quality lifecycle. It is not to be mistaken as a complete data quality assessment in that it does not explicitly determine whether a data defect exists. Instead, profiling enhances knowledge of the data and raises awareness of potential data quality issues that might require a thorough data quality assessment in order to determine the true quality status of the data in question. Table 3 presents a list of typical data profiling metrics.

Metadata Measures	**Descriptive Measures**
• Ordinal Position • Column Name • Data Type • Data Length • Actual Type • Decimal Places • Nullable • Primary Key Candidate • Minimum Length • Maximum Length • Format Name	• Statistical Dispersion • Mean • Median • Mode • Standard Deviation • Pearson Deviation • Standard Error • Minimum Value • Maximum Value • Interquartile Range • Classical Skewness • Robust Skewness

Data Quality Metrics	Other Analysis
• Count • Unique Count • Uniqueness • Pattern Count • Pattern % • Null Count • Percent of Null • Non-null Count • Percent of Non-null • Blank Count • Percent of Blank • Missing Count (Blank and Null) • Percent of Missing Values • Zero Count • Percent of Zero	• Frequency Distribution • Pattern Frequency Distribution • Percentiles • Outliers

Table 3. Non-exhaustive List of Column-Level Profiling Metrics

Depending on the information available, additional analysis can be orchestrated, such as the following:

- Data inventory/tagging: It aims at answering questions like "Do you have any data of type X?" To address this requirement, you need to be able to orchestrate field name and field content analysis in order to tag variables and tables. Such analysis can rely on machine learning, natural language processing identification analysis capabilities, regular expression, rules, or dictionaries. These technologies are often supported with a manual remediation process and/or collaborative work from the catalog community.
- Scoring: There are many metrics that can be calculated on a table:
    - data quality score by analyzing the completeness and the redundancy of the information.
    - ABT score, meaning how much one table is fit for the purpose for doing analytics.
    - privacy score by analyzing the number of variables that are identified as personal or sensitive data, their completeness, and the likelihood of reidentification
    - risk score by combining the privacy score, the information security metadata, and the degree of exposure of the table in the organization

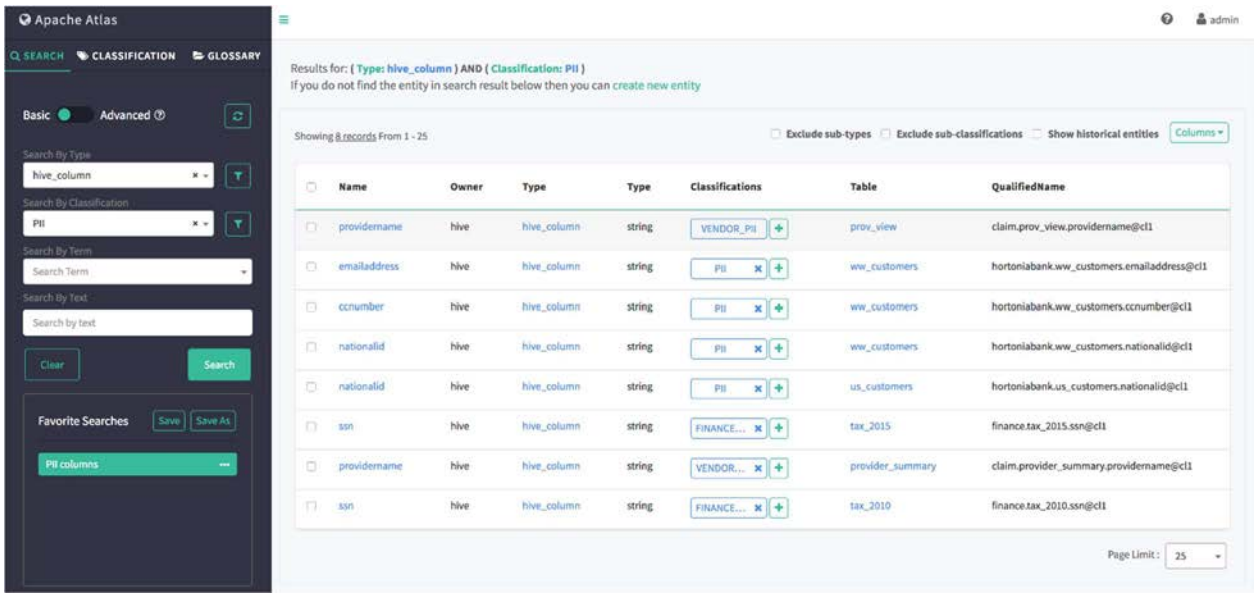
Support Smart and Easy Search Engine and UI

User interface complexity is one of the main reasons why metadata management has failed in the past. Now that personas have changed, users want to access a UI where they can search/interrogate a catalog of available assets. It must be as easy as possible. No technical knowledge should be required. The main objective is to facilitate the access to data by non-

technical people and to allow them to be autonomous. Searching in a data catalog must take a sub second and be as easy as searching on Google or Amazon.

Free text search on metadata through facets, keywords, or even natural language is a must have. All properties and attributes can be used for searching so that users could be formulating queries such as "revenue for French retail stores in 2018" or "tables containing sensitive data with more than 10000 records."

Most of the data catalog solutions rely on well-known search engines such as Elasticsearch, Solar, or Lucene and expose an API so that the search can be embedded into other applications

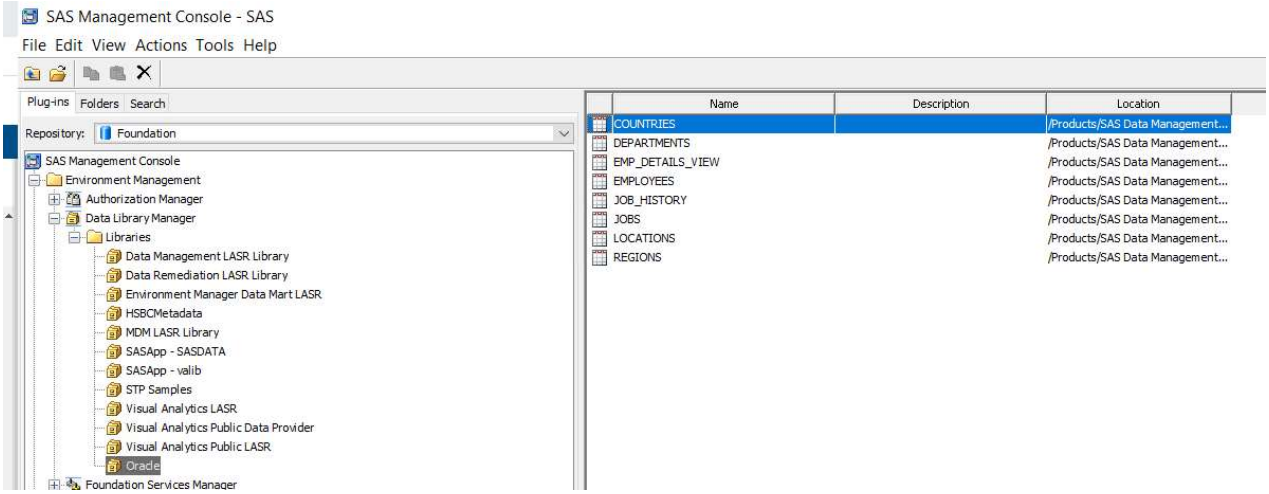


Display 1. Apache Atlas Basic Search

Let's put these capabilities to music using the SAS9 platform!

BORN TO RUN: EXTRACT METADATA FROM SAS9

There are of course multiple options for extracting metadata from SAS9. We will focus on extracting data assets (libraries), data sets (tables, views), and columns, as well as their properties. To illustrate these different options, we will use the Oracle HR database as an example. Display 2 presents the Oracle HR tables in SAS® Management Console.



Display 2. SAS Management Console Library View

YOU CAN'T ALWAYS GET WHAT YOU WANT?

There are two approaches when getting metadata from any data management platform. Either metadata is already accessible, and you only need to extract the content, or metadata is generated using a crawler. The two approaches are applicable to the SAS9 platform as it contains a metadata server that can drive one or multiple metadata repositories and delivers programmatic content for accessing and analyzing data sources. Table 4 presents the different options that are detailed in the following sections.

	Complexity 1 - 5	**Comments**	**Data Profiling Metrics**
Repository-based Metadata			
SAS® Metadata Server	3	Contains not only data assets, but all kinds of metadata objects. Multiple queries might be needed for getting detailed information.	No
SAS® Lineage Relationship Services	4	Focus is on relationships. Does not contain classic metadata attributes such as format and length.	No
Crawled Metadata			
SAS® Drivers for ODBC SAS® Drivers for JDBC	2	Relies on the program using the ODB/JDBC drivers. Data volumes might not be suitable with ODBC/JDB.	N/A
SAS Code-Based Crawler	1	Heavily configurable (sampling, tagging, and so on)	Yes
SAS® Data Quality Crawler	3	Heavily configurable but limited to ODBC-supported data sources and to medium data volumes.	Yes

Table 4. Summary of the Different Options for Getting Data Assets' Metadata from SAS9

KNOCKIN' ON SAS METADATA SERVER'S DOOR

What Is the SAS Metadata Server, and How to Access It?

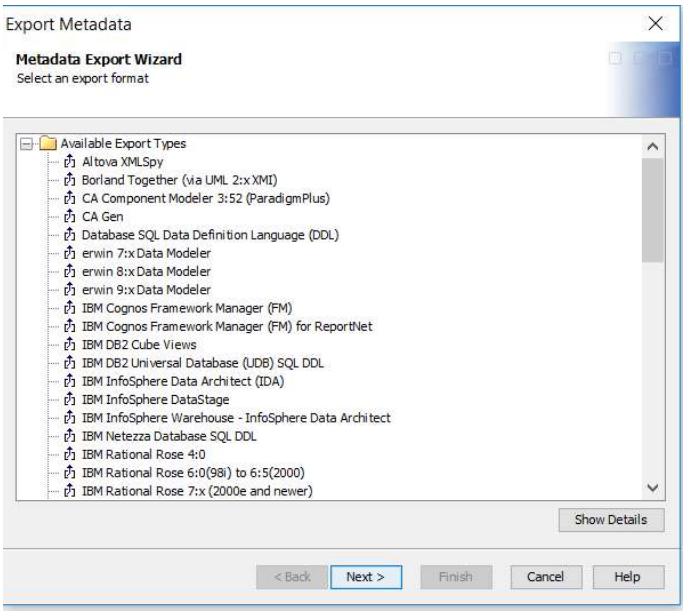
SAS Metadata Server is the most critical software component in the SAS9 platform. It provides common metadata services to applications. One metadata server supports all the SAS applications in your environment and can support hundreds of concurrent users. The SAS Metadata Server supports the exchange of metadata between applications, so that applications can work together more easily. It also provides centralized management of metadata resources. Because there is a common framework for creating, accessing, and updating metadata, it is easier to manage the applications that rely on this metadata. The SAS Metadata Server stores information about the following:

- enterprise data sources and data structures that are accessed by SAS applications
- resources that are created and used by SAS applications, including information maps, OLAP cubes, report definitions, stored process definitions, and scheduled jobs
- servers that run SAS processes

- users and groups of users that use the system, and the levels of access that users and groups have to resources

Export Metadata from SAS Metadata Server Using SAS® Management Console and DI Export Metadata Wizard

This feature relies on the SAS® Metadata Bridges that are third-party components from Meta Integration Technology, Inc. SAS Metadata Bridges enable you to export and import metadata to and from standard formats like common warehouse metamodel (CWM) or XMI, but also to exchange metadata with other software vendors like IBM, Informatica, SAP, or Oracle. Unfortunately, all the vendors do not use the same strategies and architecture for managing metadata, and these bridges try to address this challenge. Display 3 illustrates some of the SAS Metadata Bridges that are supported in SAS 9.4M6.



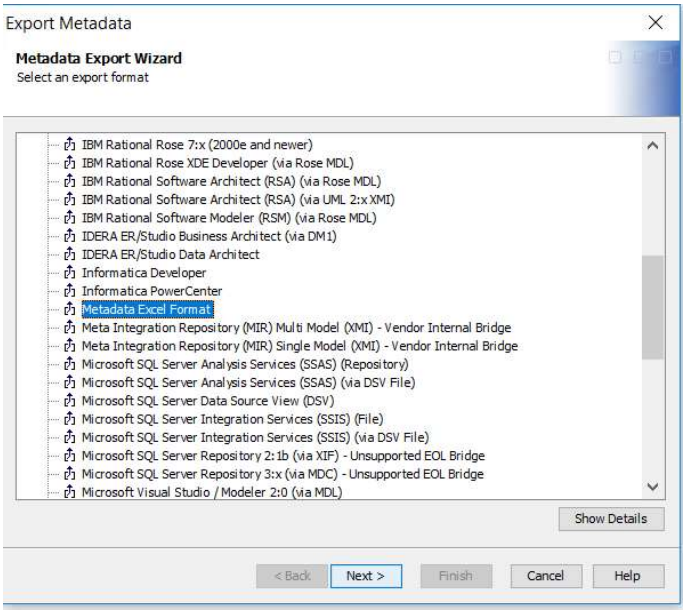
Display 3. Available Export Format in SAS Management Console

To import or export metadata in a format that is accessible with a SAS Metadata Bridge, you must license the appropriate bridge. The bridges that are appropriate for your site were probably installed along with other SAS software. For a list of the available bridges, see the SAS Metadata Bridges page at http://support.sas.com/software/bridges/.

Through the Export Metadata Wizard, you can import and export relational metadata in any format that is accessible with a SAS Metadata Bridge. Relational metadata includes the metadata for the following objects:

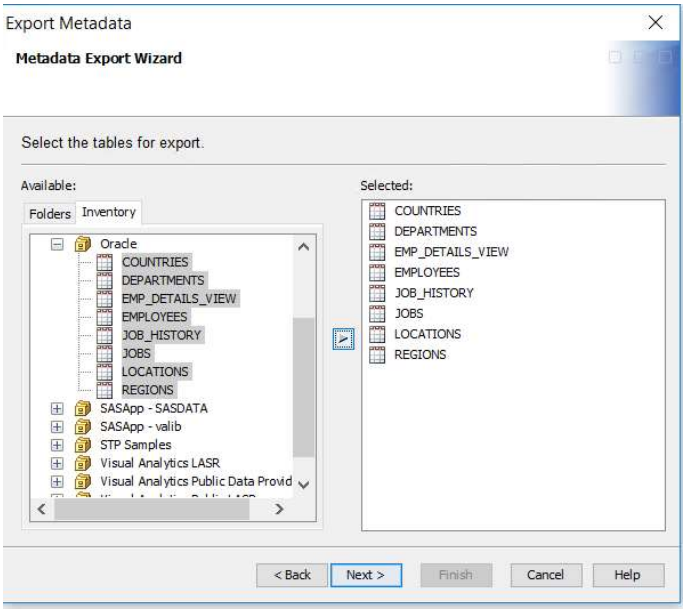
- data libraries
- tables
- columns
- indexes
- keys (including primary keys and foreign keys)

It can be invoked either from the SAS Management Console or from SAS® Data Integration Studio. Display 4 shows the selection of the Excel export format.



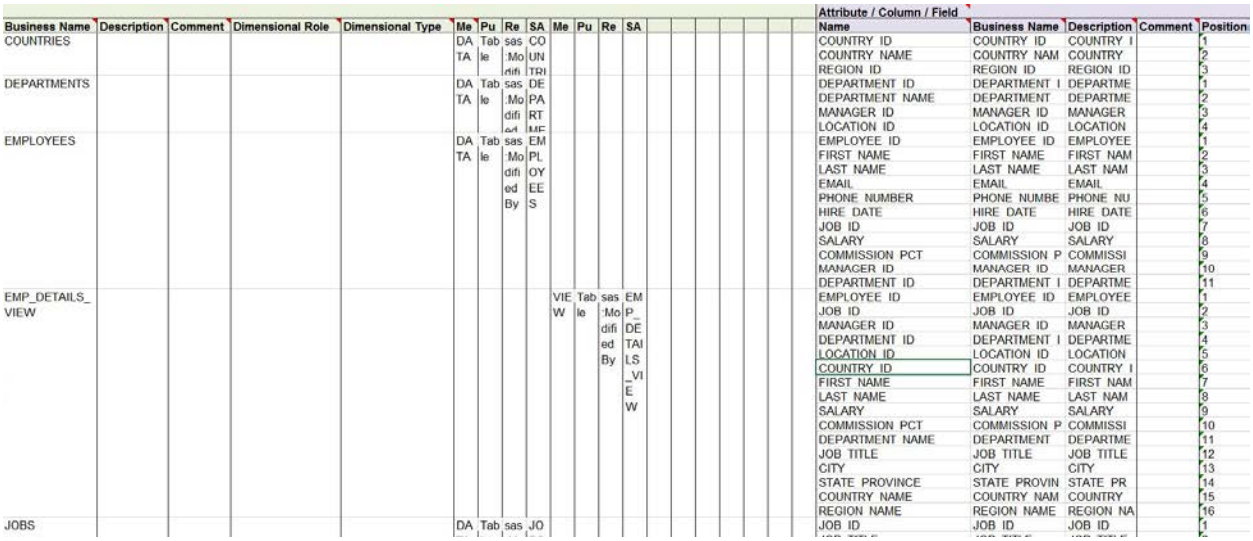
Display 4. Metadata Excel Format Selection

As presented in Display 5, the process is simple and only requires selecting the tables to be exported and the expected format to be delivered. Unfortunately, this process is not supported through a command line interface.



Display 5. Tables Selection for Export

The process generates an Excel spreadsheet with all the metadata related to the selected tables. Next, this spreadsheet can be reused for preparing metadata in order to ingest them into a third-party data catalog. Display 6 is a screen capture from an Excel export.



Business Name	Description	Comment	Dimensional Role	Dimensional Type	Me	Pu	Re	SA	Me	Pu	Re	SA
COUNTRIES					DATA	Table	sas.Modified	COUNTRY				
DEPARTMENTS					DATA	Table	sas.Modified	DEPARTMENT				
EMPLOYEES					DATA	Table	sas.Modified By	EMPLOYEES				
EMP_DETAILS_VIEW									VIEW	Table	sas.Modified By	EMP_DETAILS_VIEW
JOBS					DATA	Table	sas	JOBS				

Attribute / Column / Field Name	Business Name	Description	Comment	Position
COUNTRY_ID	COUNTRY_ID	COUNTRY_I		1
COUNTRY_NAME	COUNTRY_NAM	COUNTRY		2
REGION_ID	REGION_ID	REGION_ID		3
DEPARTMENT_ID	DEPARTMENT_I	DEPARTME		1
DEPARTMENT_NAME	DEPARTMENT	DEPARTME		2
MANAGER_ID	MANAGER_ID	MANAGER		3
LOCATION_ID	LOCATION_ID	LOCATION		4
EMPLOYEE_ID	EMPLOYEE_ID	EMPLOYEE		1
FIRST_NAME	FIRST_NAME	FIRST_NAM		2
LAST_NAME	LAST_NAME	LAST_NAM		3
EMAIL	EMAIL	EMAIL		4
PHONE_NUMBER	PHONE_NUMBE	PHONE_NU		5
HIRE_DATE	HIRE_DATE	HIRE_DATE		6
JOB_ID	JOB_ID	JOB_ID		7
SALARY	SALARY	SALARY		8
COMMISSION_PCT	COMMISSION_P	COMMISSI		9
MANAGER_ID	MANAGER_ID	MANAGER		10
DEPARTMENT_ID	DEPARTMENT_I	DEPARTME		11
EMPLOYEE_ID	EMPLOYEE_ID	EMPLOYEE		1
JOB_ID	JOB_ID	JOB_ID		2
MANAGER_ID	MANAGER_ID	MANAGER		3
DEPARTMENT_ID	DEPARTMENT_I	DEPARTME		4
LOCATION_ID	LOCATION_ID	LOCATION		5
COUNTRY_ID	COUNTRY_ID	COUNTRY_I		6
FIRST_NAME	FIRST_NAME	FIRST_NAM		7
LAST_NAME	LAST_NAME	LAST_NAM		8
SALARY	SALARY	SALARY		9
COMMISSION_PCT	COMMISSION_P	COMMISSI		10
DEPARTMENT_NAME	DEPARTMENT	DEPARTME		11
JOB_TITLE	JOB_TITLE	JOB_TITLE		12
CITY	CITY	CITY		13
STATE_PROVINCE	STATE_PROVIN	STATE_PR		14
COUNTRY_NAME	COUNTRY_NAM	COUNTRY		15
REGION_NAME	REGION_NAME	REGION_NA		16
JOB_ID	JOB_ID	JOB_ID		1

Display 6. Excel Export

Export Metadata from SAS Metadata Server using SAS Code

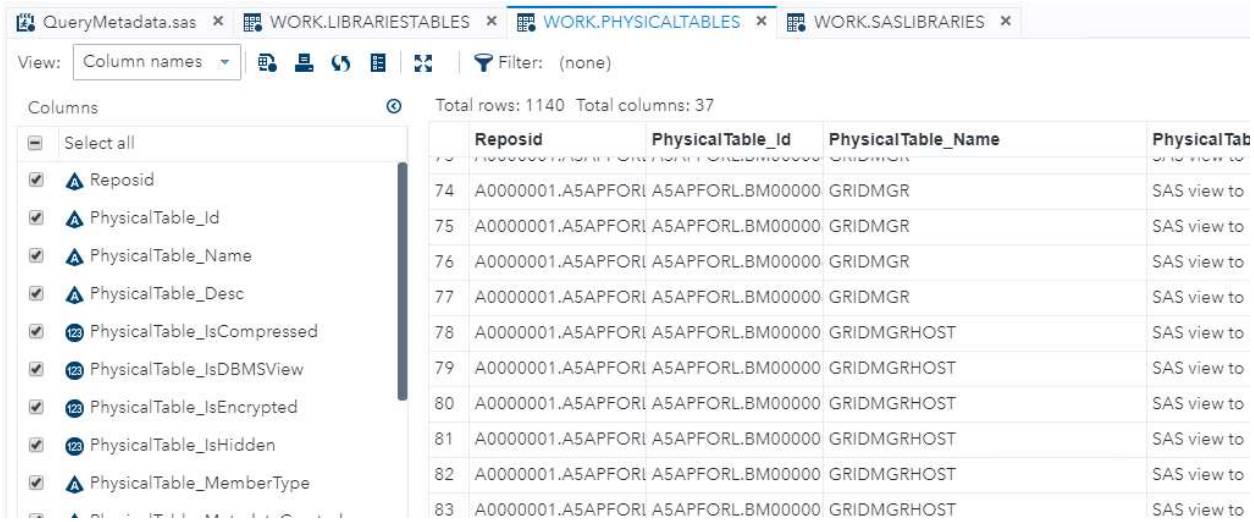
If there is a need to add other metadata objects or to automate the export of metadata from SAS Metadata Server, then the use of SAS code is judicious. The following code illustrates how to extract all library, table, and column information from the SAS Metadata Server. XML files are generated and then transformed using an XML map. This XML map is not presented in this paper. However, all the code examples are made available in the GitHub repository mentioned in the references.

```
filename query temp;
filename sastab "C:\pathToFiles\tables.xml";
filename saslib "C:\pathToFiles\libraries.xml";
filename  SXLEMAP 'C:\pathToMap\catalog.map';

/*Query SAS Metadata Server for getting registered SAS Libraries*/
data _null_;
   file query;
   input;
   put _infile_;
   datalines;
<GetMetadataObjects>
   <Reposid>$METAREPOSITORY</Reposid>
   <Type>SASLibrary</Type>
   <Objects/>
   <Ns>SAS</Ns>
   <Flags>2309</Flags>
      <Options>
      <Templates>
```

```
       <SASLibrary Id="" Name="" ChangeState="" Desc="" Engine=""
IsDBMSLibname="" IsHidden="" IsPreassigned="" Libref="" LockedBy=""
MetadataCreated="" MetadataUpdated="" PublicType="" UsageVersion=""/>
    </Templates>
   </Options>
</GetMetadataObjects>
;;
run;


proc metadata
     in=query
     out=saslib;
run;


/*Query SAS Metadata Server for getting registered Tables and Columns*/
data _null_;
   file query;
   input;
   put _infile_;
   datalines;
<GetMetadataObjects>
   <Reposid>$METAREPOSITORY</Reposid>
   <Type>PhysicalTable</Type>
   <Objects/>
   <Ns>SAS</Ns>
   <Flags>2309</Flags>
   <Options>
    <Templates>
<DatabaseSchema><UsedByPackages/></DatabaseSchema>
       <PhysicalTable/>
      <Column Name="" BeginPosition="" ColumnLength="" ColumnName=""
ColumnType="" Desc="" EndPosition="" IsDiscrete="" IsHidden="" IsNullable=""
MetadataCreated="" MetadataUpdated="" PublicType="" SASColumnLength=""
SASColumnName="" SASColumnType="" SASExtendedLength="" SASPrecision=""
SASScale="" UsageVersion=""/>
       <TablePackage/>
       <DatabaseSchema><UsedByPackages/></DatabaseSchema>
    </Templates>
   </Options>
</GetMetadataObjects>
```

```
;;
run;

proc metadata
     in=query
     out=sastab;
run;

/*Generates SAS Datasets based on XML output*/
libname   saslib xmlv2 xmlmap=SXLEMAP access=READONLY;
libname   sastab xmlv2 xmlmap=SXLEMAP access=READONLY;

data SASLibraries;
   set saslib.SASLibrary;
run;

data LibrariesTables;
   set sastab.LibTable;
run;

data PhysicalTables;
   set sastab.PhysicalTable;
run;
```

Display 7 illustrates the output of this program in SAS® Studio. Three tables are generated containing the libraries, the tables and their columns, and the relationships between the libraries and tables.



Display 7. SAS Studio Output

ROCK THE SAS LINEAGE/RELATIONSHIP SERVICE

SAS Lineage was made available in the first maintenance release of SAS 9.4, and is part of different SAS® Data Management bundles: SAS® Data Governance, SAS® Data Quality Advanced, and SAS Data Management Advanced. SAS Lineage relies on the SAS Relationship Service and the underlying relationship database. The relationship service collects and stores metadata about a variety of content from SAS and sources outside of SAS as well as processes that include resources used in data management, business intelligence, and data integration. It provides a common, standard way for applications to manage and discover relationships between objects to aid in lineage and impact analysis.

The SAS Relationship Service REST API provides an internal API for the storage and discovery of relationships and for the efficient retrieval of large networks of relationship data used by BI lineage.

From a data cataloging perspective, the relationship service could be interesting in terms of integration in order to present the visualization of assets within SAS Lineage. However, it does not offer an efficient way for extracting metadata objects and their attributes. Moreover, getting details about each object can become cumbersome compared to the other approaches presented in this paper. Therefore, only note that it is possible to extract libraries, tables, and columns using either the relationship-reporter batch tool or the SAS Relationship Service REST API.

Reporting About Metadata Using the Relationship Reporter Batch Tool

Authorized users can use the relationship reporter batch tool utility (sas-relationship-reporter) to create reports about the relationships among content objects in the SAS folders tree. The relationship reporter utility combines filtering options that are available with other command-line utilities with a new set of options for filtering relationships. Four standard reports (lineage, impact, direct dependencies, and indirect dependencies) can be written to the console or to a file. Note that the relationship reporter must be run for each library and each table. The following command line command enables you to export all the tables for one library:

```
\SASHome\SASPlatformObjectFramework\9.4\tools>sas-relationship-reporter -host
myhost -port 80 -user mysuer -password mypw "/Products/SAS Data
Management/Data Sources/Oracle/Oracle (Library)" -rel {-direction FROM}
```

Output 1 shows an example of the output of the relationship-reporter batch tool utility.

```
"/Products/SAS Data Management/Data Sources/Oracle/Oracle" (Library)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/COUNTRIES"
(Table)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/DEPARTMENTS"
(Table)
 Impacts: "/Products/SAS Data Management/Data
Sources/Oracle/EMP_DETAILS_VIEW" (Table)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/EMPLOYEES"
(Table)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/JOB_HISTORY"
(Table)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/JOBS" (Table)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/LOCATIONS"
(Table)
 Impacts: "/Products/SAS Data Management/Data Sources/Oracle/REGIONS"
(Table)
```

Output 1. Output From the sas-relationship-reporter Command Line

In order to extract metadata about columns (as shown in Output 2), another report must be generated for each table:

```
\SASHome\SASPlatformObjectFramework\9.4\tools>sas-relationship-reporter -host
myhost -port 80 -user mysuer -password mypw "/Products/SAS Data
Management/Data Sources/Oracle/COUNTRIES (Table)" -report lineage
```

```
"/Products/SAS Data Management/Data Sources/Oracle/COUNTRIES" (Table)
 Is dependent on: "/Products/SAS Data Management/Data
Sources/Oracle/Oracle" (Library)
 Is dependent on: "/Products/SAS Data Management/Data
Sources/Oracle/REGIONS" (Table)
 Is dependent on: "/Products/SAS Data Management/Data
Sources/Oracle/Oracle" (Library)
 Contains: "REGION_ID" (Column)
 Contains: "REGION_NAME" (Column)
 Contains: "COUNTRY_ID" (Column)
 Contains: "COUNTRY_NAME" (Column)
 Contains: "REGION_ID" (Column)
```

Output 2. Output From the sas-relationship-reporter Command Line

Querying the SAS Relationship REST API

We will just present here the classic REST requests for retrieving the different objects. Each JSON or XML response needs to be parsed and content reused for getting more information. Mind that the REST API does not allow you to get extended attributes about libraries, tables, and columns. Only relationships are presented.

Retrieve SAS Libraries

Note that the object ID for SAS Libraries is 31. The following REST call generates a list of all the libraries available (as shown in Output 3):

```
GET http://sasdm.demo.sas.com/SASWIPClientAccess/rest/relsvc/31
```

```
        {
"label": "Oracle",
"resourceId": 1326,
"id": "A5APFORL.B400000C/SASLibrary",
"objectType": 31,
"analysisDate": 1582277228519,
"resourceAttributes": {
    "resourceAttributes": []
},
"links": [
    {
        "method": "GET",
        "rel": "self",
        "href": "http://sasdm.demo.sas.com/SASWIPClientAccess/rest/relsvc/3
1/A5APFORL.B400000C-SASLibrary",
        "uri": "/relsvc/31/A5APFORL.B400000C-SASLibrary"
    },
    {
        "method": "GET",
        "rel": "get-relationships",
        "href": "http://sasdm.demo.sas.com/SASWIPClientAccess/rest/relsvc/3
1/A5APFORL.B400000C-SASLibrary/relationships",
        "uri": "/relsvc/31/A5APFORL.B400000C-SASLibrary/relationships"
    }
]
```

```
        },
```

Output 3. JSON Output Listing All the SAS Libraries

Retrieve Tables

Note that the object ID for Tables is 32. The following REST call generates a list of all the tables available in one library (as shown in Output 4):

```
GET
http://sasdm.demo.sas.com/SASWIPClientAccess/rest/relsvc/31/A5APFORL.B400000C
-SASLibrary/relationships?direction=3&objectTypes=32
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<relationshipModels>
    <relationshipModel version="1">
        <resource version="1">
<date>2020-02-21T04:27:08.519-05:00</date>
<id>A5APFORL.B400000C/SASLibrary</id>
<label>Oracle</label>
<links>
    <link uri="/relsvc/31/A5APFORL.B400000C-
SASLibrary" rel="self" method="GET" href="http://sasdm.demo.sas.com/SASWIPC
lientAccess/rest/relsvc/31/A5APFORL.B400000C-SASLibrary"/>
    <link uri="/relsvc/31/A5APFORL.B400000C-
SASLibrary/relationships" rel="get-
relationships" method="GET" href="http://sasdm.demo.sas.com/SASWIPClientAcc
ess/rest/relsvc/31/A5APFORL.B400000C-SASLibrary/relationships"/>
</links>
<objectType>31</objectType>
<properties/>
<resourceId>1326</resourceId>
        </resource>
        <relationships>
<relationship>
    <relationshipType>D</relationshipType>
    <direction>FROM</direction>
    <resource version="1">
        <date>2020-02-21T04:27:08.436-05:00</date>
        <id>A5APFORL.BM000017/PhysicalTable</id>
        <label>COUNTRIES</label>
        <links>
<link uri="/relsvc/32/A5APFORL.BM000017-
PhysicalTable" rel="self" method="GET" href="http://sasdm.demo.sas.com/SASW
IPClientAccess/rest/relsvc/32/A5APFORL.BM000017-PhysicalTable"/>
<link uri="/relsvc/32/A5APFORL.BM000017-
PhysicalTable/relationships" rel="get-
relationships" method="GET" href="http://sasdm.demo.sas.com/SASWIPClientAcc
ess/rest/relsvc/32/A5APFORL.BM000017-PhysicalTable/relationships"/>
        </links>
        <objectType>32</objectType>
        <properties/>
        <resourceId>1325</resourceId>
    </resource>
    <properties/>
</relationship>
```

Output 4. JSON Output Listing All the Tables in One SAS Library

Retrieve Columns

Note that the object ID for Columns is 33. The following REST call generates a list of all the columns available in one table (as shown in Output 5):

```
GET
http://sasdm.demo.sas.com/SASWIPClientAccess/rest/relsvc/32/A5APFORL.BM000017
-PhysicalTable/relationships?direction=3&objectTypes=33
```

```
<relationship>
    <relationshipType>I</relationshipType>
    <direction>TO</direction>
    <resource version="1">
        <date>2020-02-21T04:27:07.267-05:00</date>
        <id>A5APFORL.BO0000UC/Column</id>
        <label>REGION_ID</label>
        <links>
<link uri="/relsvc/33/A5APFORL.BO0000UC-
Column" rel="self" method="GET" href="http://sasdm.demo.sas.com/SASWIPClien
tAccess/rest/relsvc/33/A5APFORL.BO0000UC-Column"/>
<link uri="/relsvc/33/A5APFORL.BO0000UC-Column/relationships" rel="get-
relationships" method="GET" href="http://sasdm.demo.sas.com/SASWIPClientAcc
ess/rest/relsvc/33/A5APFORL.BO0000UC-Column/relationships"/>
        </links>
        <objectType>33</objectType>
        <properties/>
        <resourceId>1329</resourceId>
    </resource>
    <properties/>
</relationship>
```

Output 5. JSON Output Listing All the Columns in One Table

JUST CAN'T GET ENOUGH METADATA

In this section, we cover different methods for generating outputs containing the libraries, tables, and columns, but without going through the SAS Metadata Server.

Crawl Metadata with SAS Code

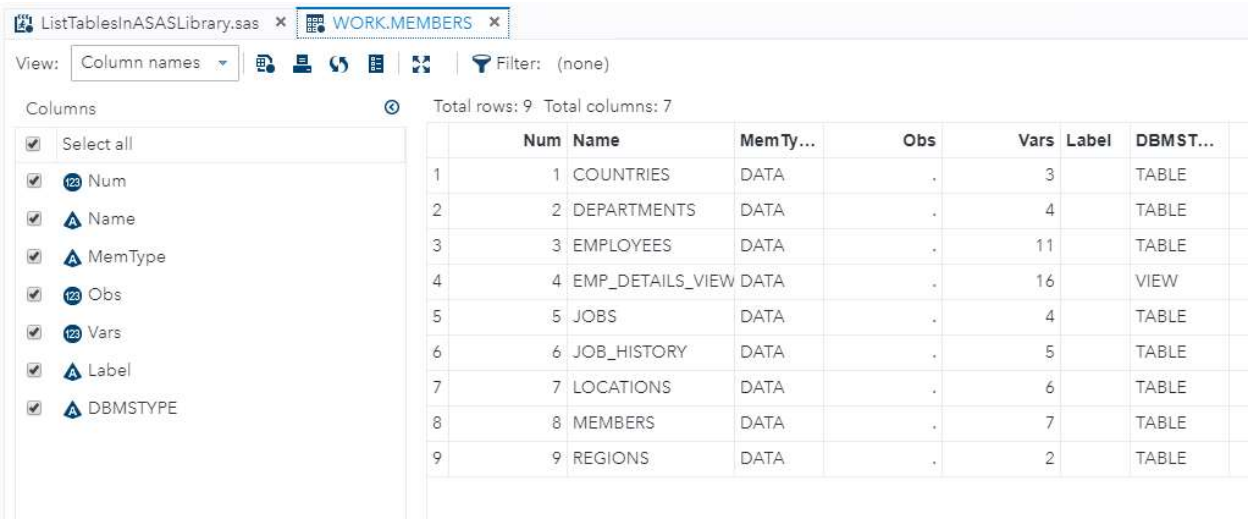
There are many ways to retrieve metadata from one library. We will present hereafter the different SAS procedures that can be used. For the following examples we will use the following defined SAS libname to Oracle:

```
LIBNAME ORACLE ORACLE PATH="(DESCRIPTION = (ADDRESS_LIST = (ADDRESS =
(PROTOCOL = TCP) (HOST = localhost) (PORT = 1521))) (CONNECT_DATA = (SID =
xe)))"

SCHEMA=HR USER=myuser PASSWORD="mypw";
```

List Tables in One SAS Library Using the DATASETS Procedure

The following code illustrates the use of the DATASETS procedure. It allows you to list the tables of one specific library (as presented in Display 8):

```
%let saslib=ORACLE;

ods output members=members;

proc datasets library=&saslib memtype=(data view) details;

run;
```

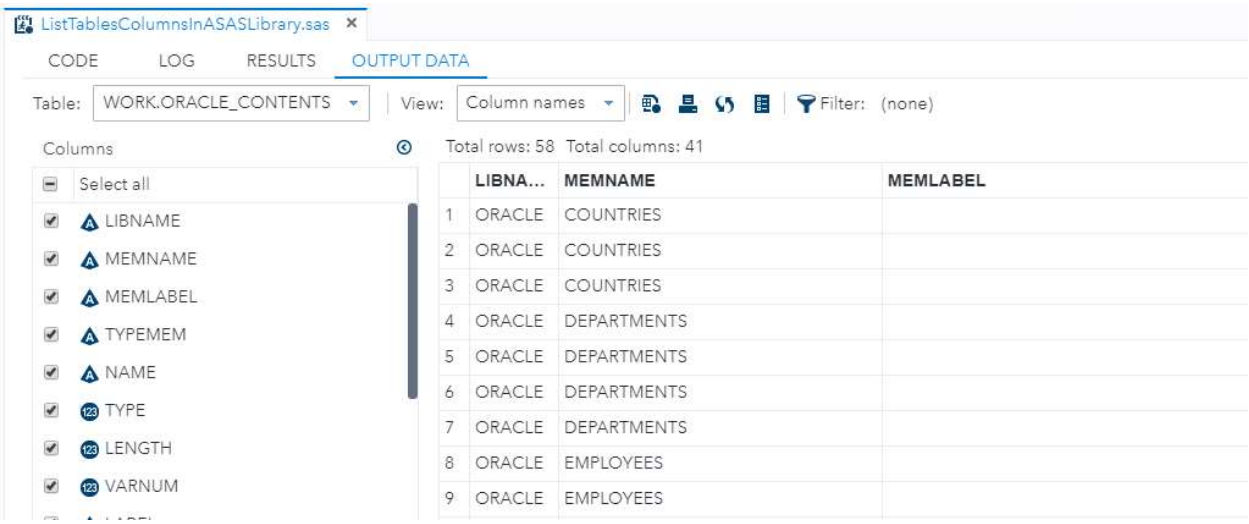


Display 8. SAS Studio Output – the DATASETS Procedure

List Tables in one SAS Library using the CONTENTS Procedure

The following code illustrates the use of the CONTENTS procedure. It allows you to list the tables of one specific library (as presented in Display 9):

```
%let saslib=ORACLE;

proc contents data=&saslib.._all_ noprint out=&saslib._contents;

run;
```



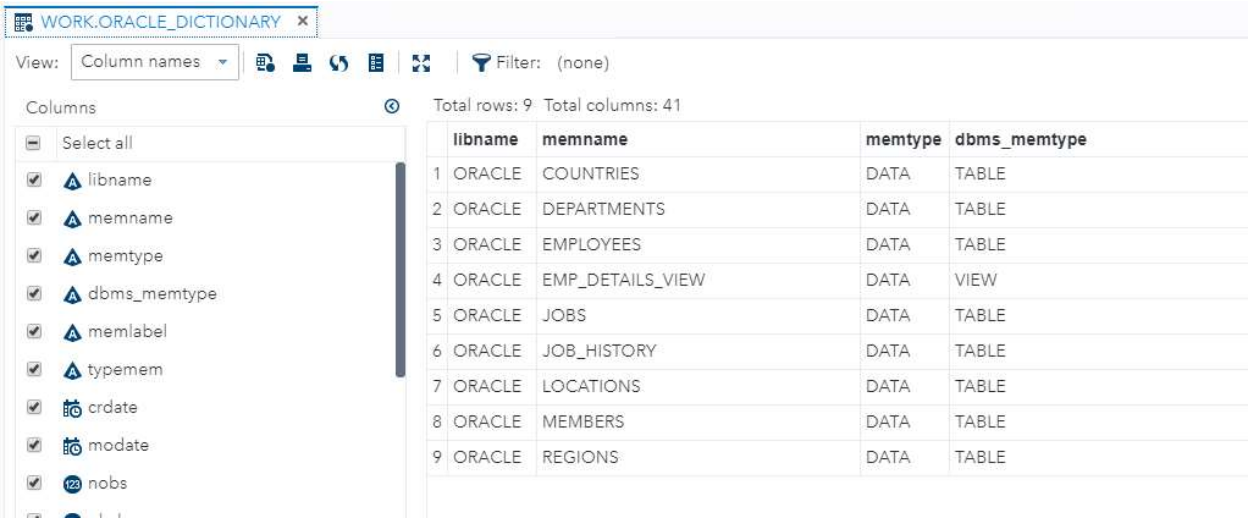
Display 9. SAS Studio Output – the CONTENTS Procedure

List Libraries, Tables, and Columns Using the SQL Procedure

The Sashelp library contains system tables that can expose information about libraries, tables, and columns that are registered or assigned in SAS session. The following examples illustrate the SQL queries to execute in order to get the information that is shown in Display 10.

```
%let saslib=ORACLE;


/*list tables in a library*/
```

```
proc sql;
create table &saslib._dictionary as select * from dictionary.tables
libname = "&saslib" order by memname ;
quit ;


Proc sql;
create table &saslib._vmember as select * from sashelp.vmember where libname
= "&saslib";
quit;


Proc sql;
create table &saslib._vtable as select * from sashelp.vtable where libname =
"&saslib";
quit;


/*List columns in SAS Library Tables*/
Proc sql;
create table &saslib._vcolumn as select * from sashelp.vcolumn where libname
= "&saslib";
quit;
```



	libname	memname	memtype	dbms_memtype
1	ORACLE	COUNTRIES	DATA	TABLE
2	ORACLE	DEPARTMENTS	DATA	TABLE
3	ORACLE	EMPLOYEES	DATA	TABLE
4	ORACLE	EMP_DETAILS_VIEW	DATA	VIEW
5	ORACLE	JOBS	DATA	TABLE
6	ORACLE	JOB_HISTORY	DATA	TABLE
7	ORACLE	LOCATIONS	DATA	TABLE
8	ORACLE	MEMBERS	DATA	TABLE
9	ORACLE	REGIONS	DATA	TABLE

Display 10. SAS Studio Output – the SQL Procedure

Generate Additional Statistical Metrics on Variables

If we want to go beyond the generation of metadata and run analysis such as data profiling, the code presented earlier has to be extended. The following example shows a crawler that allows you to analyze one Oracle library and extract metadata about the tables and columns that it contains, as well as to run the MEANS procedure and the UNIVARIATE procedure for each table. All the results are consolidated into three different tables:

```
%let saslib=ORACLE;
```

```
/*Profile Macro*/
%macro profile(table);
/*Apply Proc Means on one table*/
proc means data=&saslib..&table STACKODSOUTPUT;
  ods output summary=means_temp;
run;
data means_temp;
  set means_temp;
  MEMNAME = "&table";
run;

proc append data=means_temp base=&saslib._means force;
run;

/*Apply Proc Univariate on one table*/
proc univariate data=&saslib..&table outtable=univariate_temp NORMAL noprint;

data univariate_temp;
  set univariate_temp;
  MEMNAME = "&table";
run;

proc append data=univariate_temp base=&saslib._univariate force;
run;

proc delete data=means_temp univariate_temp;
run;

%mend profile;
/*End of Macro*/

proc contents data=&saslib.._all_ noprint out=&saslib._contents;
run;

/*Runs the profile macro on each table of the Library*/
data _null_;
  set &saslib._contents;
  by memname ;
```

```
  if first.memname;
  call execute(cats('%profile(',memname,')'));
run;
```

This basic crawler generates the following three tables:

- MYLIB_CONTENTS
- MYLIB_MEANS
- MYLIB_UNIVARIATE

Crawl with SAS ODBC or JDBC

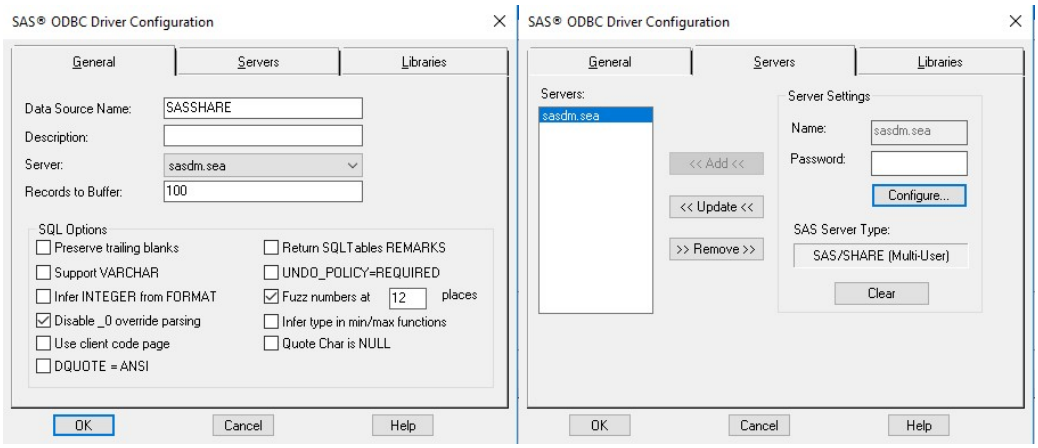
All the former SQL queries can be reused through an SQL client that is connected to SAS through an ODBC or JDBC driver to SAS.

First, SAS/SHARE must be licensed and installed. Next, SAS/SHARE must be started with the server procedure like the following:

```
LIBNAME ORACLE ORACLE PATH="(DESCRIPTION = (ADDRESS_LIST = (ADDRESS =
(PROTOCOL = TCP) (HOST = localhost) (PORT = 1521))) (CONNECT_DATA = (SID =
xe)))"


proc server id=sea authenticate=optional;
run;
```

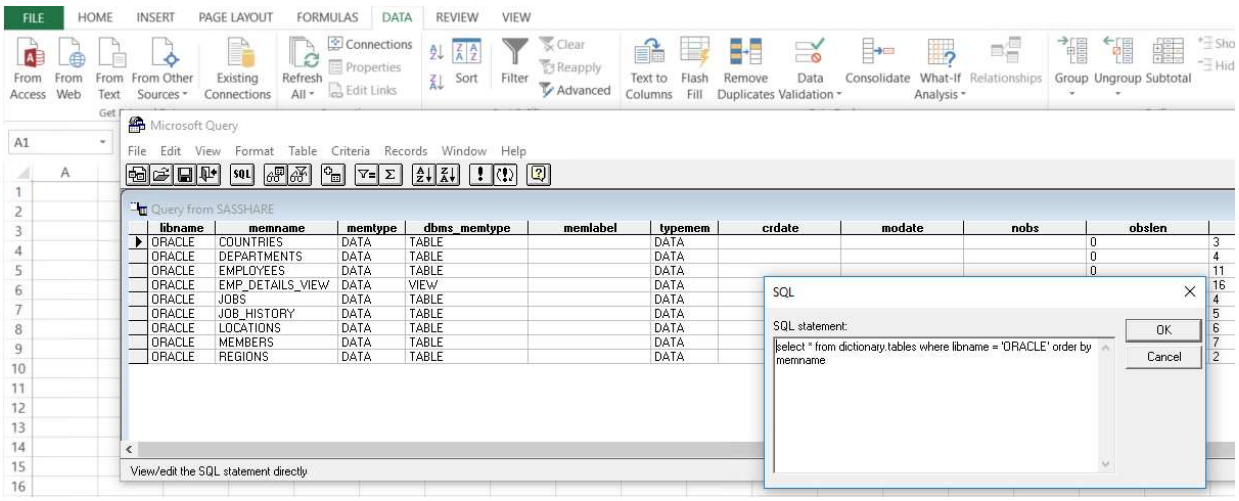
Additional libnames can be added in order to make them available through ODBC or JDBC.

Then an ODBC or JDBC connection must be set up (as shown in Display 11).



Display 11. SAS ODBC Driver to SAS configuration

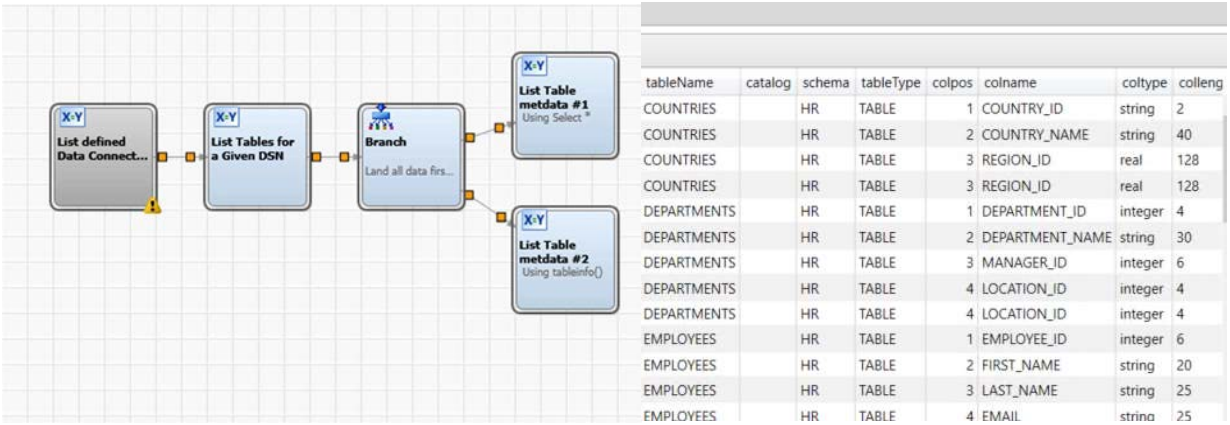
Finally, you can use your favorite SQL client or specific product (such as Microsoft Excel) for extracting metadata through the ODBC or JDBC connection that you created (as presented in Display 12).



Display 12. Microsoft Query over SAS ODBC Driver to SAS connection

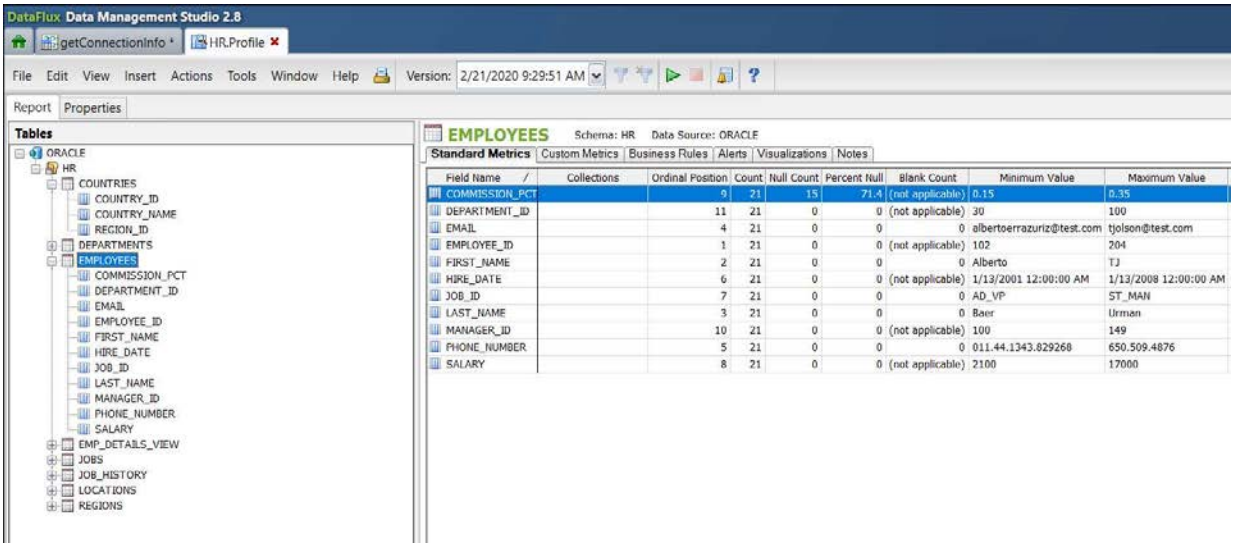
Crawl Metadata with SAS Data Quality

SAS Data Quality also offers metadata crawling capabilities using expression engine language that allows you to list all connections, the tables they contain, and the structure of each table. It won't make sense to show the code behind each node. However, this job is available in the GitHub project related to this paper. Display 13 shows an example of such a job and its output.



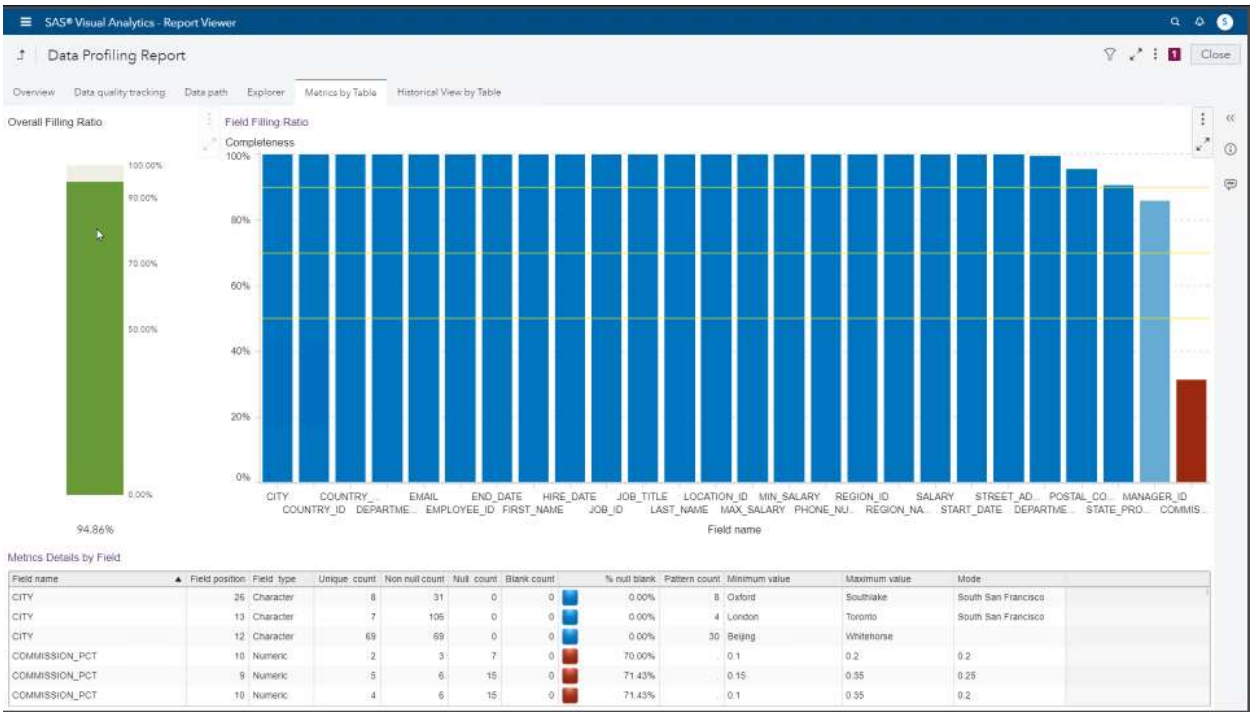
Display 13. SAS DataFlux Data Management Studio Job

SAS Data Quality allows you to profile data, which can be considered as a metadata crawling method too. In addition, SAS Data Quality calculates descriptive measures as well as additional metadata metrics such as completeness and frequency distributions. Display 14 presents a data profiling job that was created within SAS DataFlux® Data Management Studio (SAS Data Quality).



Display 14. SAS DataFlux Data Management Studio – Data Profiling Job

Data profiling metrics can be then extracted from the SAS Data Quality repository and made available for a data catalog or for reporting into SAS® Visual Analytics (or other data visualization tool) as shown in Display 15. This data quality process job is also available in the GitHub repository.



Display 15. Data Profiling Report in SAS Visual Analytics

BORN TO BE WILD: OPEN-SOURCE DATA CATALOG AND SAS

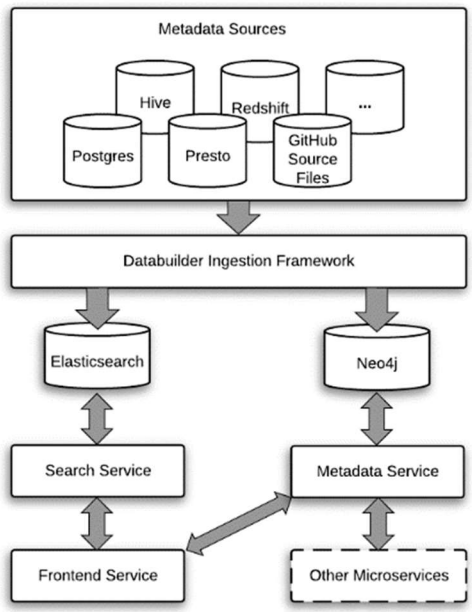
What if you want to deploy a data catalog over the SAS9 platform? Thanks to the former section you now have some options for getting metadata out, but which product to use? There are more than 50 data cataloging products available on the market, and the purpose of this paper is not to review them all. SAS Institute is also working on its data catalog that will be delivered on the SAS® Viya® platform in 2020. It will enable you to access SAS9 metadata.

However, what could be the alternatives for easily deploying a data catalog over SAS9 metadata? In terms of data cataloging, the open sources projects are popular and the internet unicorns such as Lyft (Amundsen), LinkedIn (DataHub), Netflix (Metacat), Uber (DataBook), and Airbnb (Data Portal) have been quite active over the last few years. In this last section, we focus on the Amundsen project that is one of the most energetic and promising projects.

HERE COMES THE SUN: LYFT AMUNDSEN

Amundsen is a project that was started in 2017 at Lyft that is now delivered under Apache 2.0 license and available on GitHub. It is a data discovery application built on top of a metadata engine. "Amundsen" refers to the Norwegian explorer, Roald Amundsen. Many other organizations also contribute to or use Amundsen like Bang & Olufsen, ING, iRobot, and Workday.

As shown in Display 16, it relies on Neo4j for the backend repository, Elasticsearch for the search engine, and a front-end service that is a Flask application with a React frontend. Note that you can substitute Neo4j for Apache Atlas as a persistent layer. It contains a data ingestion library for building a metadata graph and search index. Users can either load the data with a Python script with the library or with an Airflow DAG importing the library. Amundsen can connect to any database that provides a DBAPI or SQLAlchemy interface (which most DBs provide). The databases we see most frequently used in the community are Hive and anything that works with Hive metastore (Spark SQL, Presto, Athena, and so on); BigQuery; PostgreSQL and anything that uses the same interface (Redshift); Snowflake; and Amazon Glue and anything built over it.



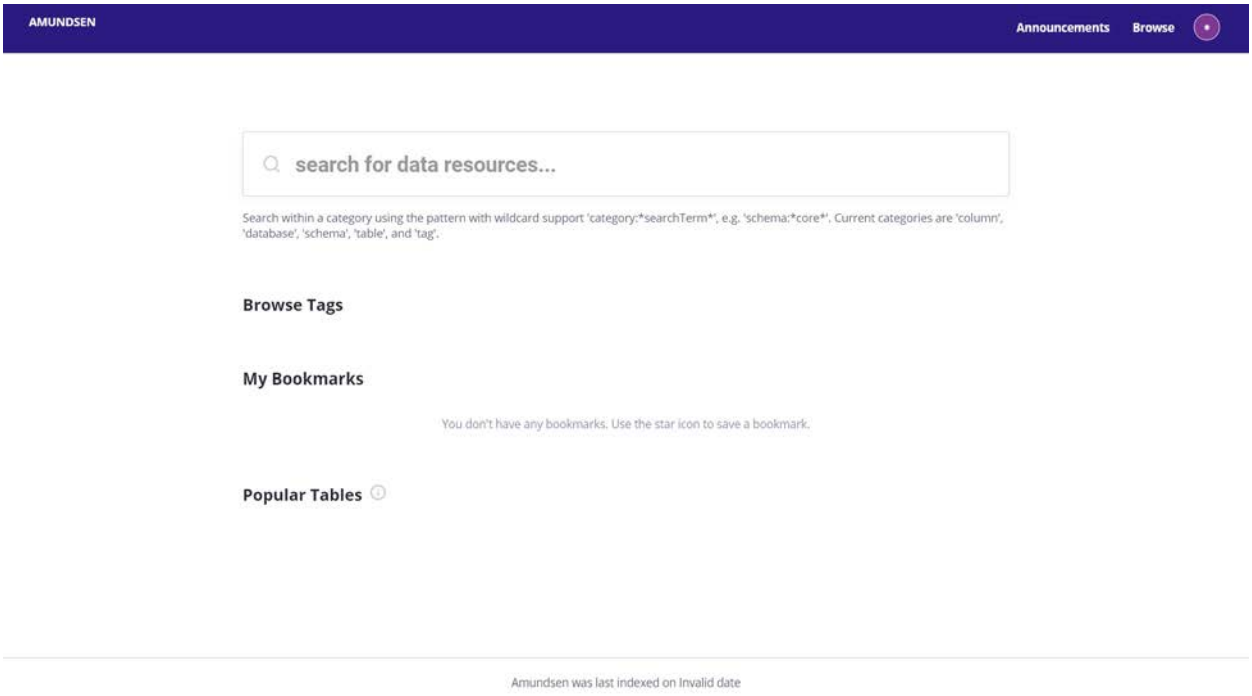
Display 16. Lyft Amundsen Architecture

WE WILL ROCK LYFT AMUNDSEN

Thanks to the use of containers, it won't take you more than 5 minutes to deploy Amundsen. Once Docker and Git are installed, you only need to run the following command:

```
git clone --recursive git@github.com:lyft/amundsen.git

docker-compose -f docker-amundsen.yml up
```

Once the different containers and services are deployed, the Amundsen web interface is up and running, but with no metadata (as shown in Display 17).



Display 17. Lyft Amundsen Search UI

Amundsen does support metadata ingestion through CSV files. As of today, 12 different files can be loaded for either adding metadata or configuring the product. The main files are the following:

- Sample_col.csv – List of all the columns (Table 5)

name	description	col_type	sort_order	database	cluster	schema_name	table_name
col1	col1 description	string	1	db1	cluster1	schema1	test_table1

Table 5. Amundsen Sample_col.csv File

- Sample_table.csv – List of all the tables (Table 6)

database	cluster	schema_name	name	description	tags	is_view	description_source
db1	gold	schema1	test_table1	1st test table	tag1	false	

Table 6. Amundsen Sample_table.csv File

- Sample_table_column_stats.csv – Statistics on columns (Table 7)

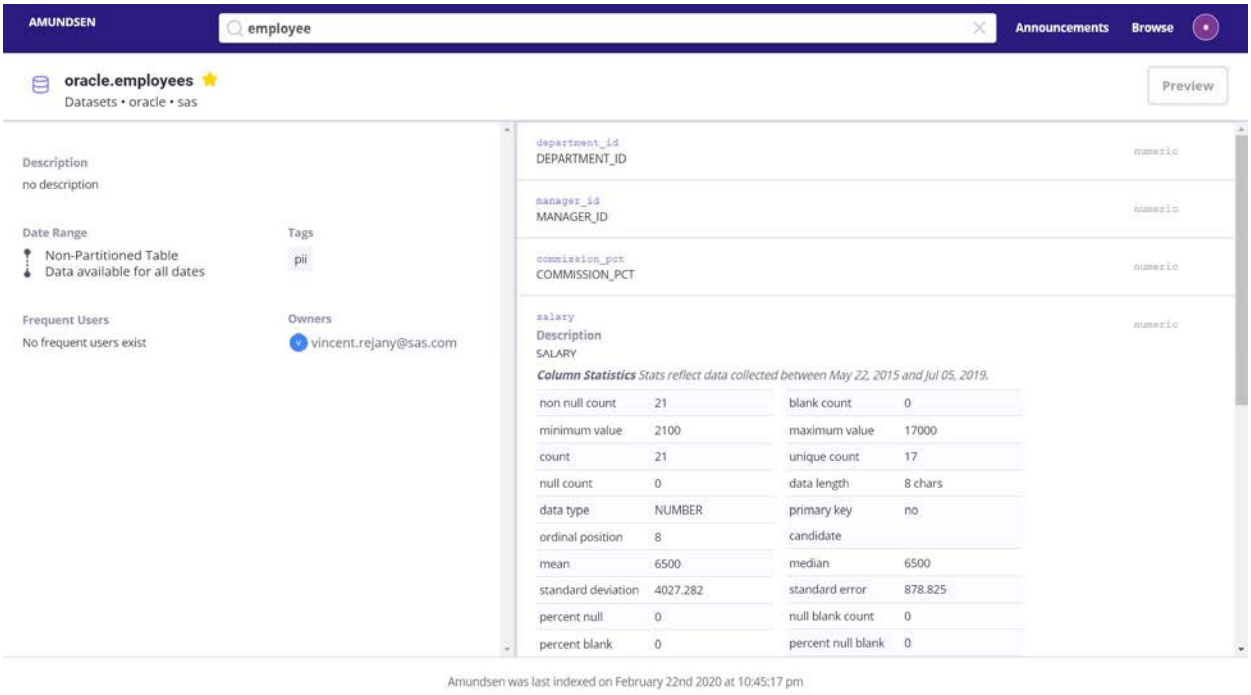
cluster	db	schema_name	table_name	col_name	stat_name	stat_val	start_epoch	end_epoch
cluster1	db1	schema1	test_table1	col1	distinct values	8	1432300762	1562300762

Table 7. Amundsen Sample_table_colum_stat.csv File

One these files are built and uploaded, the import can be launched by running the following commands:

```
$ python3 -m venv venv
 $ source venv/bin/activate
 $ pip3 install -r requirements.txt
 $ python3 setup.py install
 $ python3 example/scripts/sample_data_loader.py
```

One the import is completed; the metadata is accessible through Amundsen UI (as shown in Display 18):



Display 18. – Amundsen – Oracle Employee Metadata and Profiling Metrics

CONCLUSION

The world of metadata management is now waking up. It is empowered by the democratization of data access and consumption through self-service capabilities and the call for more ease of use, simplification for faster time to value. Business users need a centralized repository for data that has been categorized and classified. However, data catalogs need to be more ambitious and to extend to other metadata objects like reports, models, process flows, data pipelines, business rules, and any other objects that are a part of the data and analytical lifecycle. The support of data lineage capabilities is key for giving the visibility of where the data is coming from, where it is used, and by whom.

Data catalogs need to transition to "information catalogs" and embed built-in security and governance capabilities., This transition could be ideally driven by AI/ML features for strengthening collaboration, policies definition, error detection, and data privacy principles. Automation is the only way to address the big data challenge and to comply with data privacy regulations.

Moreover, the openness and extensibility of catalogs is becoming a critical topic. With an increasing number of solutions delivered by cloud vendors and data management software vendors, the ability to integrate catalogs with each other through API or standards like the ODPI Egeria project is a judicious bet on the future. Without such an approach, the ambition of bringing clarity in the metadata confusion will surely fail. Getting metadata or generating metadata out from a platform is the easiest part. Maintaining a catalog and sustaining the governance is a higher hill to climb.

REFERENCES

Aslett, M. 2018. "From Out of Nowhere: The Unstoppable Rise of the Data Catalog." Available https://www.alation.com/data-catalog-rise-451-research/.

Bourn, E., Peters, A., and Wolfe, B. 2015. "Hands-Off SAS® Administration—Using Batch Tools to Make Your Life Easier." *Proceedings of the SAS Global Forum 2015 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings15/SAS1857-2015.pdf.

Gartner, Inc. 2017. "Data Catalogs Are the New Black in Data Management and Analytics." Available https://www.gartner.com/en/documents/3837968/data-catalogs-are-the-new-black-in-data-management-and-a.

Julian, P. 2011. "Using Dictionary Tables to Explore SAS® Datasets." *Proceedings of the SouthEast SAS Users Group Conference*. Available https://analytics.ncsu.edu/sesug/2011/PO17.Julian.pdf.

McIntosh, L., Rausch, N., and Wolfe, B. 2014. "Understanding Change in the Enterprise." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings14/SAS396-2014.pdf.

McIntosh, L. 2016. "New for SAS® 9.4: Integrating Third-Party Metadata for Use with Lineage and Impact Analysis." *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings16/SAS5562-2016.pdf.

Mistry, J. 2013. "SAS® Metadata Reporting: Extracting Information from SAS Metadata." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings13/518-2013.pdf.

Nowlin, S. J. 2006. "Data Profiling Using Base SAS® Software: A Quick Approach to Understanding Your Data." *Proceedings of the Thirty-First Annual SAS Users Group*

International Conference. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings/proceedings/sugi31/161-31.pdf.

Poling, J. W. 2011. Finding Oracle® Table Metadata: When PROC CONTENTS Is Not Enough." *Proceedings of the South Central SAS Users Group Conference*. Available http://www.scsug.org/SCSUGProceedings/2011/poling2/FINDING%20ORACLE%20TABLE%20METADATA%20WITH%20SAS.pdf.

van Stein, E. J. 2010. "Getting Your Metadata Using PROC METADATA." *Proceedings of the SAS Global Forum 2010 Conference*. Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings10/071-2010.pdf.

Ventura, D. 2008. "Using Proc SQL to run SAS® Procedures on a Group of Datasets within a Single Directory." *Proceedings of the NorthEast SAS Users Group Conference*. Available https://www.lexjansen.com/nesug/nesug08/cc/cc08.pdf.

Website of the ODPi Egeria Guidance on Governance. Available https://opengovernance.odpi.org. Accessed February 1, 2020.

RESOURCES

Paper-related GitHub repository: https://github.com/sascommunities/sas-global-forum-2020/tree/master/papers/4356-2020-Rejany

Apache Atlas project: https://atlas.apache.org/#/

Lyft Amundsen project: https://github.com/lyft/amundsen

LinkedIn Datahub project: https://github.com/linkedin/datahub/

ACKNOWLEDGMENTS

I would like to thank the Data Management Product Management and R&D teams for their confidence and wise advice. I would like to especially thank Nancy Rausch, Liz McIntosh, Ron Agresta, Chris Replogle, and Jeff Bailey.

RECOMMENDED READING

- *SAS® 9.4 Language Interfaces to Metadata, Third Edition*
- *SAS4079-2020 - What's New in SAS Data Management by Nancy Rausch*
- *SAS4615-2020 - Getting from Governance Practice to Data Awareness by Chris Replogle*
- *SAS4223-2020 - Steer Your Hybrid SAS® Viya®/SAS® 9 Ship Towards the "Governed Data" Port by Bogdan Teleuca*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Vincent Rejany
SAS Institute Inc
+33 6 40 54 17 99
Vincent.rejany@sas.com