Paper SAS394-2014

Exploring Data Access Control Strategies for Securing and Strengthening Your Data Assets Using SAS® Federation Server

Mark Craver, Mike Frost, SAS Institute Inc., Cary, NC

ABSTRACT

"Potential of One, Power of All." That has a really nice ring to it, especially as it pertains to accessing all of your corporate data through one single data access point. It means the potential of having a single source for all of your data connections from throughout the enterprise. It also means that the complexities of connecting to these data assets from the various source systems throughout the enterprise are hidden from the end user. With this, however, comes the possibility of placing personally identifiable information in the hands of a user who should not have access to it. The bottom line is that there is risk and uncertainty with allowing users to have access to data that is disallowed by your existing data governance strategy. Blocking these data elements from specific users or groups of users is a challenge that many corporations face today, whether it is secure financial information, confidential personnel records, or personal medical information protected by strict regulations. How do you surface "All" necessary data to "All" necessary users, while at the same time maintaining the security of the data? SAS® Federation Server Manager is an easy-to-use interface that allows the data administrator to manage your data assets in such a way that it alleviates this risk by controlling access to critical data elements and maintaining the proper level of data disclosure control. This session focuses on how to employ various data access control strategies from within SAS Federation Server Manager.

INTRODUCTION

The purpose of this paper is to discuss the features available in SAS Federation Server Manager that enable you to secure your corporate data assets, providing you with the peace of mind that is often absent in enterprise-level data management projects. Through lecture and live demonstration, you will see how SAS® Federation Server technology can be applied to your data management project in order to fulfill your data access control requirements. This session will cover:

- A discussion of the business issue
- A review of the typical users/roles
- An explanation of how SAS Federated Query Language (FedSQL) components can be used to perform data masking tasks
- An overview of the SAS Federation Server technology components
- A demonstration of how the SAS Federation Server Manager interface can be used to accomplish various data access control tasks.

THE BUSINESS ISSUE

Today, more than ever before, data is seen as a strategic corporate asset and is at the heart of key business decisions. New technologies for gathering and sharing data, as well as cheaper storage have made processing huge volumes of data a reality for companies worldwide. With more and more data being collected and stored, companies seek to place more data in the hands of more users for the purposes of analysis, forecasting, querying and reporting. This allows corporations to stay on the cutting edge competitively, but poses a significant challenge for them in getting as much data as possible to as many users as possible, while at the same time maintaining privacy and security on sensitive data elements.

Often, data records contain information that needs to exist for the purposes of analysis and reporting, but must be surfaced in such a way that the end user cannot determine specifics about the record itself. This type of information might contain specific information about a person's name or address, account information, financial identification numbers and health-related information. This information is referred to as "personally-identifiable information" (PII).

Table 1 shows an example of data that contains PII that is typical in a financial transactions data table.

Table 1

ID	ACCOUNT_NUM	TRANSACTION_TYPE	BRANCH	TRANSACTION_ AMOUNT	TELLER_TYPE
1111	1234567890	DEPOSIT	555	500.00	ATM
2222	2345678901	DEPOSIT	888	100.00	MOBILE
3333	3456789012	INQUIRY	555	1433.45	ATM
4444	1234567890	WITHDRAWAL	444	-250.00	BANK
5555	9991112222	PAYMENT	888	1200.50	ONLINE

Table 1. Financial data table containing personally-identifiable information.

As you can see in Table 1, there are data elements in the data that certain users potentially should not have access to. For example, a reporting user in a business unit might be able to see only data elements pertaining to the ID, Transaction_Type, Branch and Teller_Type fields for the purposes of querying and reporting. Information pertaining to Account_Num and Transaction_Amount represent data elements that should be hidden from certain users (or groups of users).

This is also a common scenario with Human Resource (or Personnel) data where certain data elements are not accessible to all users. Some analysts in the Human Resources department can see all employees' records for the sake of reporting, but some can see demographic information only. Salary information and job performance data might be considered confidential and not available to all analysts. Compensation analysts, on the other hand, might be able to see salary and job title information, but not be able to see employee IDs and names associated with the salary information. Department managers might be able to see all data for all employees that report up to them, but not other employees in the company. All of these are examples of where a good security system must be in place before making the data available to users.

THE PERSONAS

The availability of more data from more sources throughout the enterprise brings a new challenge of more users requiring access to the data for their unique business needs. These users serve in various roles throughout the enterprise, and so the one-data-fits-all data warehousing practices of the past are no longer practical. You must employ a strategy that meets the needs of users throughout the enterprise without compromising the security of the data itself.

Consider the following types of reporting users who need access to data for their day-to-day reporting tasks:

- **Federation Server Administrator** – The Federation Server administrator is responsible for the installation, configuration, maintenance, tuning, and operation of SAS Federation Server. Duties include server startup and shutdown, and quiescence of the server process. This administrator is typically responsible for the Federation Server instances, but typically does not handle the data connections, or the user security settings, that it manages.

- **Data Administrators** – The Data Administrator's job involves overseeing data management projects in preparing data for report users. These users are skilled at fulfilling the requirements of the report users, and can optimize data stores for querying and/or reporting. These users can typically see all data and have full permissions for doing whatever they need to do to manage the data for reporting and querying purposes. These administrators are also responsible for defining and managing the users, groups and permission settings for the data elements managed by the server.

- **Data Architect** – Someone who is responsible for defining the structure and rules around how the data will be made available and presented. This role defines views, filter conditions, and caching parameters of the data. Data Architects are in the unique position of understanding the intricate details of the data, in addition to the business uses of the data. They are in a particularly good position to be the liaison between IT and the business users. These users are capable of writing some SQL code when business needs dictate. These users can typically see most data elements, and have permission to perform most operations on the data.

- **Report Users** – These users have access to certain data elements for the purpose of performing ad hoc querying and reporting, but they are limited in the types of processes they can perform on the data. In many cases, the data needs to be filtered to just the data elements they can see, including limiting rows of data as well as columns of data. These users are often not technical enough to write the necessary SQL code to access the data that they need. In addition, these users do not understand the complexities of accessing the data from various databases and source systems throughout the enterprise. Therefore, these users rely heavily on the planning and programming by IT staff and data administrators to get them the data they need in a timely fashion.

THE BUSINESS ISSUE

For any of the users listed above, other than the data administrator, there are almost certain to be restrictions on the data they are allowed to see and use for decision support activities. There needs to be some way to provide all of the necessary data to all of the users throughout the enterprise, without compromising the security, privacy or identity of the data that they need to perform their job.

The challenge, then, lies in how to surface data from throughout the enterprise, to users throughout the enterprise, without compromising confidential and personally-identifiable information.

Table 2 is an example of how the data would appear to a report user who does not have access to all data elements.

Table 2

ID	ACCOUNT_NUM	TRANSACTION_TYPE	BRANCH	TELLER_TYPE
1111	xxxxxx7890	DEPOSIT	555	ATM
2222	xxxxxx8901	DEPOSIT	888	MOBILE
3333	xxxxxx9012	INQUIRY	555	ATM
4444	xxxxxx7890	WITHDRAWAL	444	BANK

Table 2. Financial data table after filtering confidential and personally-identifiable information.

You can see in Table 2 that the account number column has been masked in such a way that it would be impossible to for a user to determine what the full account number is for the particular record. Also notice that the transaction amount column has been removed from the view of the data.

If we are unable to provide the subset of data to the appropriate user, then one of the following typically happens:

- Data administrators have to pre-aggregate the data, and the end user potentially lacks the detail necessary to perform the desired tasks. In addition, the report user has little or no knowledge of how the data was aggregated or manipulated. The user is limited in their ability to query data at the detail level, or maybe even perform joins that rely on detail data. Often, the data administrator does not perform the task to the expectation of the end user.
- IT has to build the data marts and queries for the report users. This is not an ideal situation because IT becomes backlogged and over-extended, they do not understand the business requirements as well as the end user, and by the time the data is delivered to the end user, it is no longer current. This situation is simply not flexible enough for the end user. In addition, frequent requests from different users for very similar data structures is not an efficient user of IT resources.

- Users are disallowed from accessing the data at all, because there is no mechanism in place to ensure that confidential and PII does not fall into the hands of the wrong user. This hinders any efforts on the report users' part to build the desired query or report, which in effect, negatively impacts enterprise decision support efforts. Executives are left to rely on "gut-level" decision making or worst yet, guessing at what the appropriate answer might be.

None of the above options are ideal. The ideal solution is to have a process in place that:

- Provides users with a single source for accessing the disparate data throughout the enterprise
- Authenticates the user at the time they request access to the data
- Returns only the portion of the data the user is allowed to see.

A solution built around federated data connections meets the criteria above, and provides the ideal enterprise solution for comprehensive and secure access to data assets throughout the enterprise.

HOW DOES DATA FEDERATION HELP?

Data federation is the process of making data, which may not have been previously available to certain users, available in such a way that it:

- **Provides a single source of data to users** – users do not have to set up ODBC connections, libnames, or even know the physical path to where the source data resides.
- **Maintains security on the data** – a security layer provides protection from data elements that certain users are not allowed to see (for example, data connections, tables/views, columns can all be hidden from users).
- **Masks certain data elements from certain users** – some users are permitted to have access to data, provided that it is masked in such a way that they cannot determine any PII in the data.
- **Hides the complexities of accessing the data from the end user** – the end user does not have to know physical paths, filenames, or database connection information in order to utilize the data in a query or a report.
- **Alleviates the need for the user to have to understand the intricate details of how the data fits together** – some reporting and querying tasks require that data be joined together from multiple data sources from multiple systems. A pre-built and cached federated SQL view keeps the report user from needing to know the details of how all of the data fits together in order to use it for reporting.
- **Does not require the user to be a programmer** – federated SQL views (or data stores) are cached and available to users just like any other data source for querying and reporting. They do not have to write code or programs in order to access the data they need in order to do their job.

TERMINOLOGY

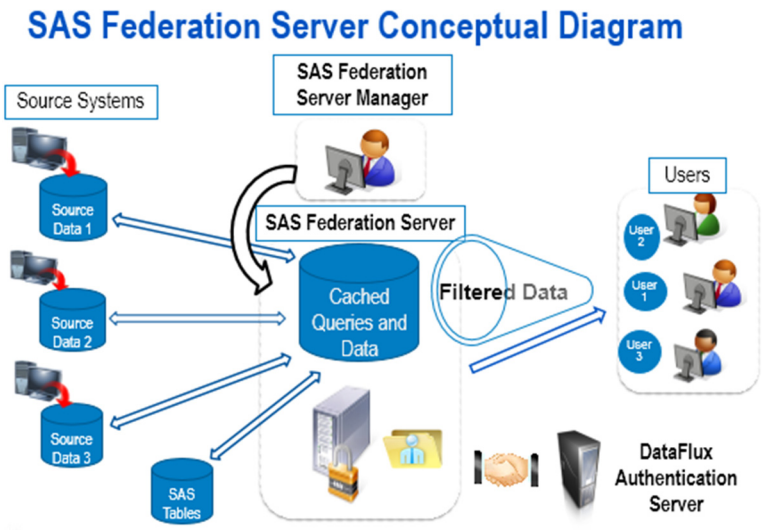
Terminology typically associated with data federation, includes:

- **Data Federation** - the ability to use data across multiple heterogeneous source systems without physically having to move the data
- **Data Virtualization** – the process of accessing and manipulating data from disparate systems, through a common data access approach, that hides the complexity of data access from the end user. This includes how the data is formatted, where it is located, database security, and so on.
- **Data Disclosure Control** – modifying data such that no sensitive information remains. The challenge of Data Disclosure Control lies in the ability to share information with users, while at the same time, protecting PII (that is, account numbers, addresses, phone numbers, taxpayer IDs, and so on) from the user.

HOW DOES SAS FEDERATION SERVER HELP?

SAS Federation Server is the technology that provides you with the ability to manage data connections in such a way that security on the data is not compromised. SAS Federation Server Manager is a web-based interface that provides a single, easy-to-use interface through which you can create federated data connections that alleviate the issues associated with the varying levels of users attempting to access data from throughout the enterprise. Display 1 shows a typical architecture with federated data connections to multiple data sources throughout the enterprise.

Display 1 shows a conceptual example of an architecture that includes SAS Federation Server, SAS Federation Server Manager, and DataFlux® Authentication Server. Using available technology resources from SAS, you can surface disparate data from throughout the enterprise to a myriad of users, each with their own distinct view of the data.



Display 1. SAS Federation Server Conceptual Architecture Diagram

On the left side of the diagram in Display 1, you see that there are numerous source systems, each having its own physical data storage, and each potentially having a different storage (for example, relational databases, SAS Data Tables, spreadsheets, text files, and so on). This creates a level of complexity that is not always intuitive to the end users when attempting to access the data for their respective reporting tasks.

The right side of the diagram shows the many users throughout the enterprise, each with their own distinctive roles and responsibilities, and each with a need to access the same data from the same source systems. The challenge is making as much data as possible available to as many users as possible without compromising the security of the data. SAS Federation Server provides these users with a single access point for all of their data requests, and has the ability to attach a security layer when accessing the data to ensure no confidential data is made available to a user who is not supposed to see it.

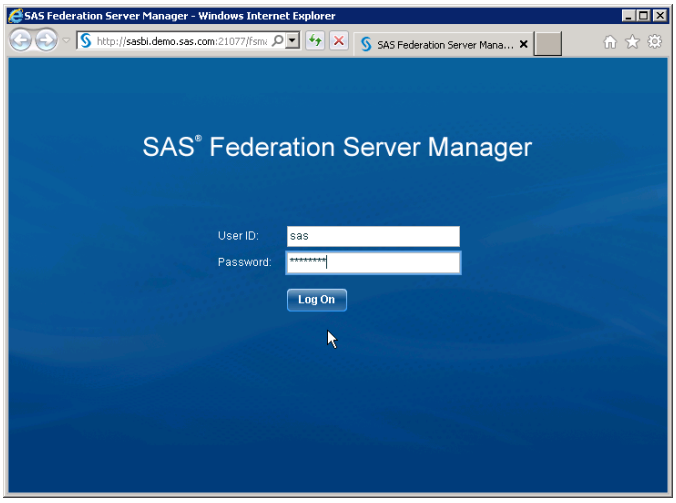
FEDERATED QUERY LANGUAGE (FEDSQL)

The SAS Federation Server uses Federated Query Language (FedSQL) to access data from a variety of relational databases. FedSQL is SAS' proprietary implementation of ANSI SQL:1999 core standard. FedSQL provides a scalable, threaded, high-performance way to access, manage, and share relational data in multiple data sources. FedSQL provides a common SQL dialect that can access data from a variety of data sources (that conform to the standard), without having to submit database-specific SQL. When possible, FedSQL queries are optimized with multi-threaded algorithms in order to resolve large-scale operations. Also, FedSQL executes on the database, which alleviates the risk of moving the data to the SAS Federation Server at all.

SAS FEDERATION SERVER MANAGER

SAS Federation Server Manager is the interface which empowers the data administrator user to create and manage federated SQL views all within a single user interface. SAS Federation Server Manager generates FedSQL code to access data contained in relational databases. The application is configured to interact with the SAS Federation Server for processing data, and the DataFlux Authentication Server for user authentication within the available interfaces.

Display 2 shows the main login screen for SAS Federation Server Manager.

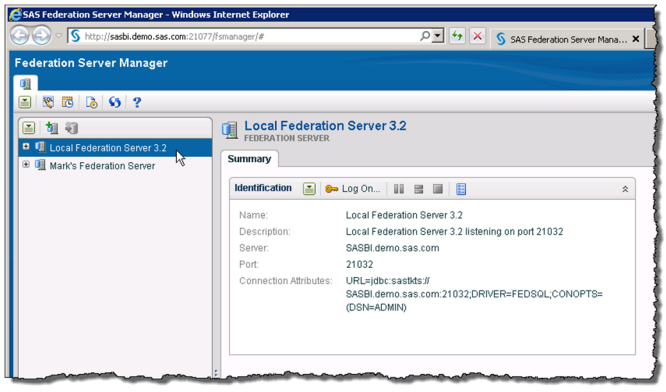


Display 2. SAS Federation Server Manager login screen.

Once you are authenticated to the SAS Federation Server Manager, you are taken to the main interface where you can perform a variety of data federation tasks. Specifically, you can:

- Register and manage one or more SAS Federation Servers
- Connect to a SAS Federation Server to display its contents
- Create and manage virtual connections to a variety of data sources
- Create and manage federated SQL views
- Manage user authorizations (LDAP or ACL)
- Assign user permissions to federated SQL views at the data connection, table/view, row and column levels
- Monitor activity on one or more SAS Federation Servers

Display 3 shows the initial screen and navigational panels for SAS Federation Server Manager once you have authenticated yourself to the application.



Display 3. SAS Federation Server Manager main screen.

Using SAS Federation Server Manager Functionality for Data Security and Data Access Control

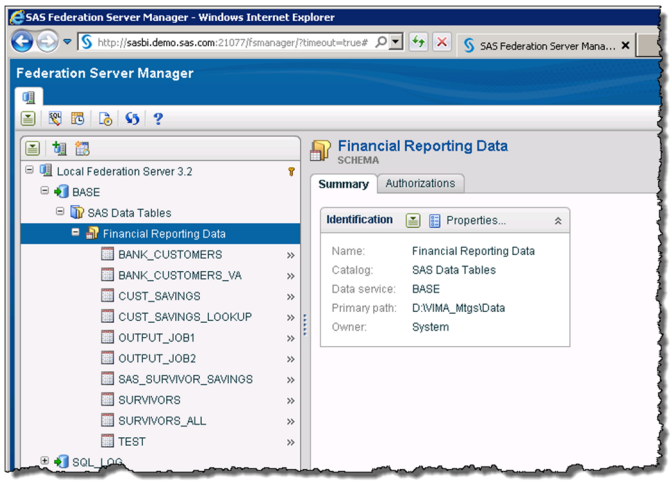
SAS Federation Server Manager provides you with a number of options for controlling the security of your corporate data assets, including:

- Setting permissions on data connections
- Setting permissions on tables/views
- Using the FedSQL SELECT statement to effectively “hide” columns from all users
- Using permissions on columns of data
- Using row-level filters to remove rows of data for certain users

In this section, you will see the interfaces available for setting up the various options listed above.

Once you have logged into SAS Federation Server Manager, you are taken to the main interface that contains a list of available federation servers in the left navigational panel. When you select the desired federation server connection, you are prompted to log in with your user credentials. The selected federation server entry in the navigational panel to the left will refresh to show the contents of the selected server, subject to the objects the user has been granted permission to see.

Display 4 shows the results of connecting to the *Local Federation Server 3.2* entry.



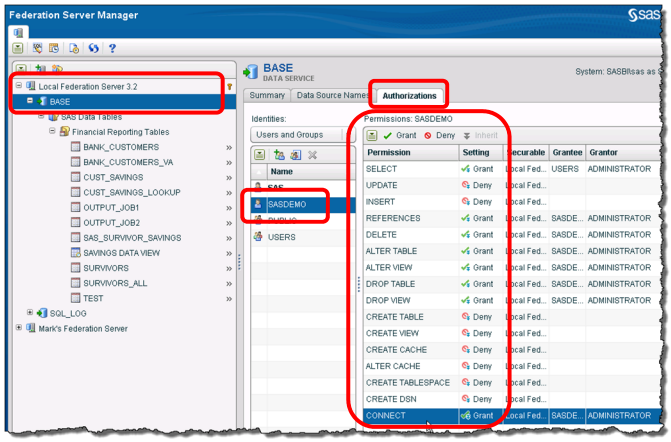
Display 4. SAS Federation Server Manager main screen.

You can see in Display 4 that the *Local Federation Server 3.2* has containers for BASE and SQL logs. You can add data connections to the registered federation server by using the provided menus and icons. In the example, the user has added a data connection to a directory containing SAS data tables. In order to make this connection available to all users throughout the enterprise, it is critical to have the option of blocking users from certain tables in the directory.

Controlling Access to Objects on the SAS Federation Server

The Authorizations tab in the right panel of SAS Federation Server Manager can be used to control which users have access to the different data assets in the various data connections on the server. By simply selecting an item in the left navigational panel, and then selecting the *Authorizations* tab in the right panel, you can **Grant** and **Deny** access for specific users (or groups of users) using the provided icons and menus.

Display 5 shows how user authorizations can be used to specifically control permissions for specific users to access data elements on the selected server.



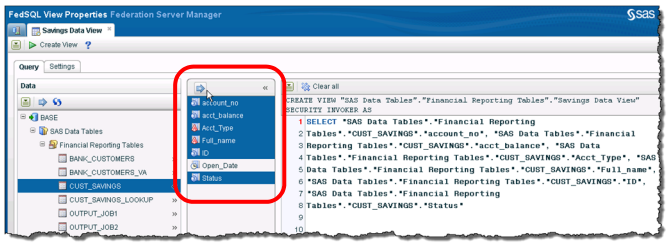
Display 5. Setting User Permissions on Data Objects in SAS Federation Server Manager Authorizations tab.

As you can see in Display 5, SAS Federation Server Manager Authorizations tab is being used to grant **CONNECT** permissions to the *SASDEMO* user for the *BASE* data source on the *Local Federation Server 3.2* server. You will also notice that the *SASDEMO* user has been specifically denied from being able to **UPDATE** and **INSERT** into federated data, as well as **CREATE TABLE**, **CREATE VIEW**, **CREATE CACHE**, and others.

Controlling Access to Columns of Data Using the SELECT FedSQL Statement

One way of controlling access to data columns in a federated SQL view is with the SELECT statement. From within the navigational panels in SAS Federation Server Manager, you have the ability to build a SELECT federated SQL view using available columns of data from one or more source tables. By carefully selecting which columns of data you wish to have in the resulting federated SQL view, you can remove a column of data completely from the resulting view. This will effectively "hide" the column of data from all users who access the view, regardless of any authorizations that are set on the view.

Display 6 shows an example of using the FedSQL SELECT statement to effectively hide a column of data from the federated SQL view for all users.



Display 6. Hiding a Column Using the FedSQL SELECT Statement in a Federated SQL View.

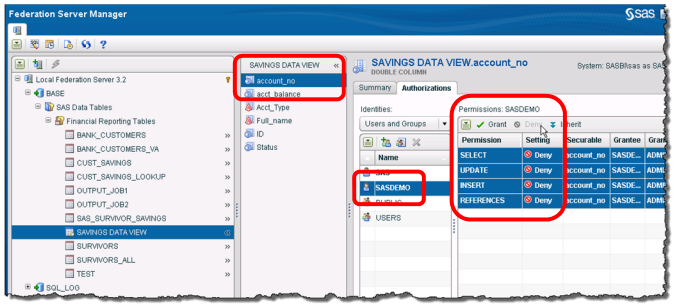
In Display 6, you can see that in building the SELECT statement for the federated SQL view, all columns except the *Open_Date* column have been selected for the new view being created.

Using Security Features to Specifically Deny Access to Certain Columns for Certain Users

Another way of controlling access to data columns in a federated SQL view is by specifically denying permission to the column(s) of data that you do not want certain users to see. This is easily accomplished in SAS Federation Server Manager's authorization tab. Once the federated SQL view is built, not only can you set authorizations to hide the entire view from certain users, but you can also set permissions at the column level within the federated view.

This feature is especially useful when you have columns of data that should not be made visible to a particular user (or group of users). For example, if you have a financial data source that contains data for all transactions that took place in one day. The transactional data store contains information about account numbers that not all users are allowed to see, but other users do need to access. This scenario is a perfect application for setting column-level permissions on a federated SQL view to deny access to those users who do not have permission to view the protected column.

Display 7 shows an example of using the DENY permission within the Authorizations tab to block a user from seeing a column of data that exists in the federated SQL view.



Display 7. DENY Access to a User for a Column In a Federated SQL View.

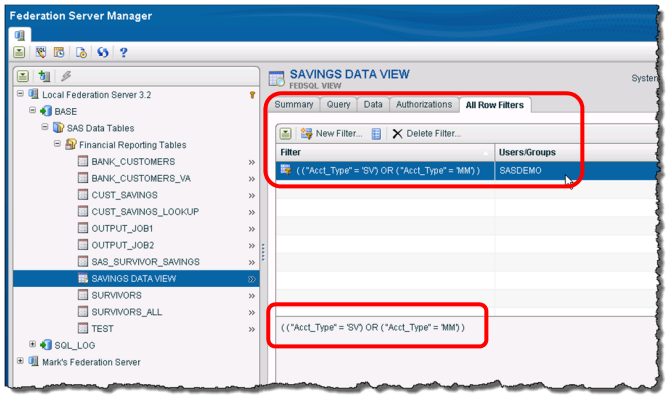
In Display 7, you can see that each column of data in a federated SQL view has authorizations associated with it. In order to keep the *SASDEMO* user from seeing the *Account_No* column, you can easily **DENY** all permissions for the column in the **Authorizations** tab.

Using Row-level Filters to Hide Rows of Data from Users

Another way of controlling access to data columns in a federated SQL view is by creating row-level filters to remove rows of data from the federated SQL views for certain users. Using a point-and-click interface, you can build SQL WHERE clauses that filter the rows of data from the federated SQL query, and then assign the filter to one or more users. When the user accesses the federated SQL view, the data is filtered in real time to remove the records the user is not permitted to see.

This feature is specifically useful when you have rows of data that simply are not needed for decision support, or if you have rows of data that you want to physically block a user (or group of users) from seeing. For example, consider a financial data source that contains data for all transactions that took place in one day. If there is a user who only needs to report on savings and money market accounts, then that user would not need to see transactions pertaining to checking accounts or loan payments. This scenario is a perfect application for a row-level filter in a federated SQL view.

Display 8 shows an example of applying a filter to a user of a federated SQL view in order to control the rows of data displayed to that user.



Display 8. DENY Access to a User for Rows of Data in a Federated SQL View.

In Display 8, you can see that a filter (WHERE clause) has been created that only keeps the records where the account type is a savings account or a money market account. This filter has specifically been applied to the SASDEMO user, so when a user logs in as the SASDEMO user, the data will automatically be subset to just the collection of rows that this user can see.

DATA MASKING

Data masking is a method of 'de-identifying' sensitive information stored within data sources. The purpose of data masking is to protect the original data by using a functional substitute in situations where the audience is not privileged to access the original data, effectively de-identifying the sensitive information from the end user. The new function, for performing data masking from within an FedSQL query, is the SYSCAT.DM.mask function. This function accepts defaults configured as package options in addition to the various arguments associated with each rule-type.

The three rule types that are currently available for use in the SYSCAT.DM.mask function include ENCRYPT, DECRYPT and HASH. These rule types are applied at the column level to any column in the FedSQL query. Let's take a closer look at each of these three types.

ENCRYPT Rule Type

The ENCRYPT rule type encrypts a single value using symmetric key encryption.

NOTE: Encrypted values cannot be decrypted if no KEY argument is specified and the ENCRYPT_KEY package configuration is not configured.

DECRYPT Rule Type

The DECRYPT rule type decrypts a previously encrypted value using symmetric key encryption.

HASH Rule Type

The HASH rule type hashes a single value into a fixed-length hash digest or HMAC string.

NOTE: The effect of using the HASH data type on a data value is not reversible.

Example Code

The code example below shows the syntax of the SYSCAT.DM.mask function.

```
SYSCAT.DM.mask( 'rule-type', value [, rule-argument1, rule-argument2, …])
```

Where:

rule-type is the name of the rule type

value is the item that requires masking (this could be a value or a column name)

rule-argument*n* is the list of arguments that apply to the type of rule selected, in the format:

```
[, 'rule-arg-name1', 'rule-arg-value1']
```

The code example below shows the use of the SYSCAT.DM.mask function using the "HASH" rule type on the 'Acct_No' column in the FIN.TRANSACTIONS table using an HMAC-MD5 hash algorithm, and the "5c39b18d77d5f297ff92e4942e5522b5" key value. The output of the hash is written to a new column named "HASH_ACCT".

```
select SYSCAT.DM.mask('HASH', "Acct_No", 'alg', 'MD5', 'key',
'5c39b18d77d5f297ff92e4942e5522b5' ) as "HASH_ACCT"

from FIN.TRANSACTIONS
```

Would yield a hash string similar to this:

```
B462BF54E510B0FE41441BE2BF1A232AFF0B8F7E05E24780574E8748EACEB20A
```

Incorporating this functionality into a FedSQL view would create data similar to what is shown in Table 3 below.

Table 3

ID	ACCOUNT_NUM	TRANSACTION_TYPE	BRANCH	TELLER_TYPE
1111	B462BF54E510B0FE41441BE2BF1A232A FF0B8F7E05E24780574E8748EACEB20A	DEPOSIT	555	ATM

Table 3. Financial data table using hashing algorithm to protect personally-identifiable information.

CONCLUSION

SAS Federation Server provides many options that allow you to strengthen the security of your data assets throughout the enterprise, while still making as much data as possible to as many users as possible. The SAS Federation Server Manager interface provides an easy-to-use administrative console that provides a variety of security features to support your data access control requirements, including:

- Column selection via the SELECT statement
- Column-level security (user-specific)
- Row-level filtering (user-specific).

Using these features in managing federated SQL views, you can make more data available to more users, while at the same time alleviating the risks of placing sensitive information in the hands of the wrong user.

The addition of the new SYSCAT.DM.mask function further enhances data disclosure control efforts by providing HASH and ENCRYPT functionality to mask the identity of data values for use in decision support activities.

RECOMMENDED READING

- *SAS Federation Server 3.2: Administrator's Guide*
- *SAS 9.4 FedSQL Language Reference*
- *SAS Drivers for Federation Server 3.2: User's Guide*

CONTACT INFORMATION

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

Name: Mark Craver
Enterprise: SAS
Address: 100 SAS Campus Drive H-1259
City, State ZIP: Cary, N.C. 27513
Work Phone: 919-531-6702
Fax: 919-677-8444
E-mail: Mark.Craver@SAS.com
Web: www.sas.com

Name: Mike Frost
Enterprise: SAS
Address: 940 Cary Parkway #273
City, State ZIP: Cary, N.C. 27513
Work Phone: 919-531-8285
Fax: 919-677-8444
E-mail: Mike.Frost@SAS.com
Web: www.sas.com