THE POWER TO KNOW®

Virtualizing Enterprise Data

Course Notes

Virtualizing Enterprise Data Course Notes was developed by David Ghan and Johnny Starling. Editing and production support was provided by the Curriculum Development and Support Department.

Virtualizing Enterprise Data Course Notes

Book code E71150, course code ATEVED, prepared date 26Jul2017. ATEVED_001

Table of Contents

To learn more…

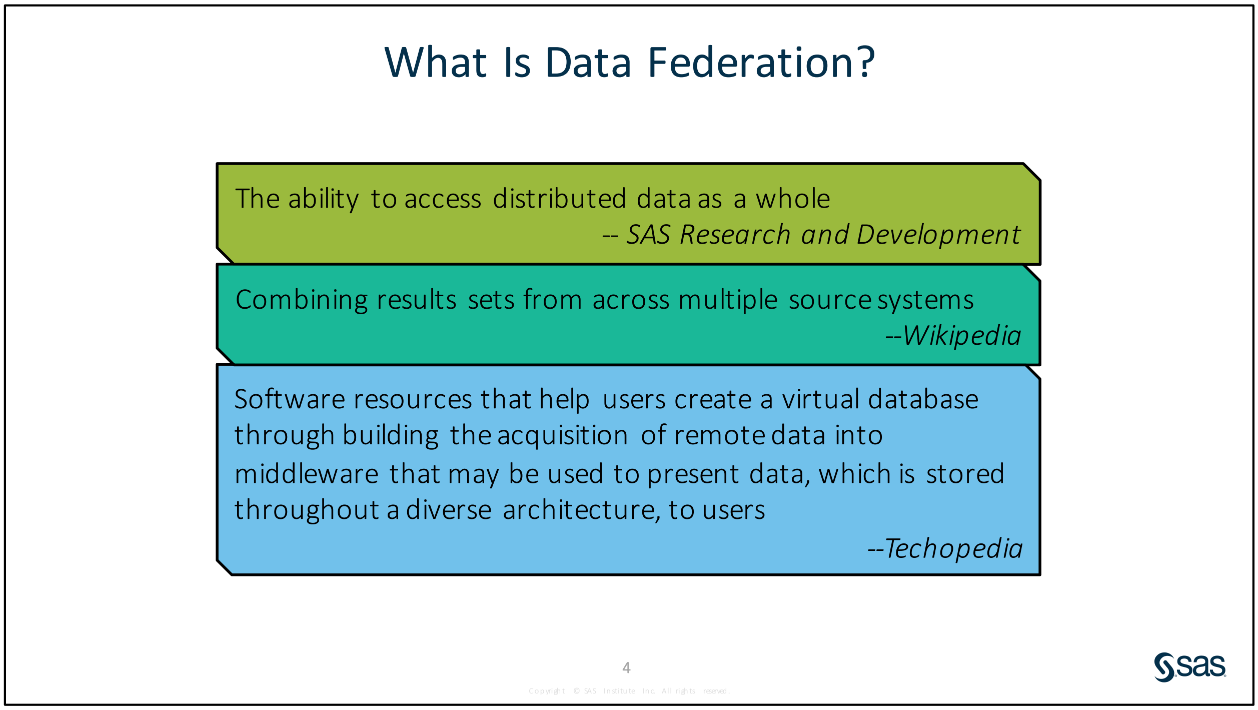
For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the web at http://support.sas.com/training/ as well as in the Training Course Catalog.

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 Virtualizing Enterprise Data

1.1 Introduction to Data Virtualization and the SAS Federation Server



What Is Data Federation?

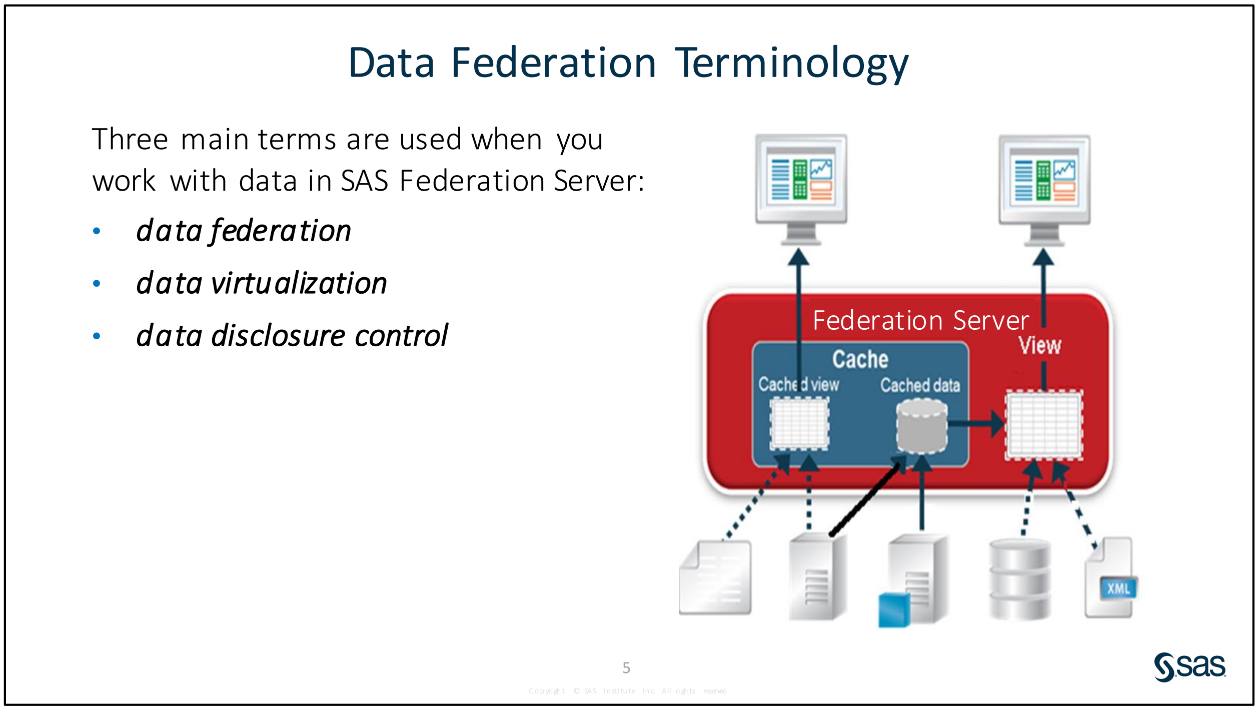
The ability to access distributed data as a whole
-- *SAS Research and Development*

Combining results sets from across multiple source systems
--*Wikipedia*

Software resources that help users create a virtual database through building the acquisition of remote data into middleware that may be used to present data, which is stored throughout a diverse architecture, to users
--*Techopedia*

4

In general, *federation* is the ability to access distributed data as a whole. A federated database (or virtual database) is the fully integrated, logical composite of all constituent databases within a federated database system. Through data abstraction, federated database systems can provide a uniform front-end user interface, enabling users and clients to store and retrieve data in multiple noncontiguous databases ***with a single query*** – even if the constituent databases are ***heterogeneous***.



Data Federation Terminology

Three main terms are used when you work with data in SAS Federation Server:

- *data federation*
- *data virtualization*
- *data disclosure control*

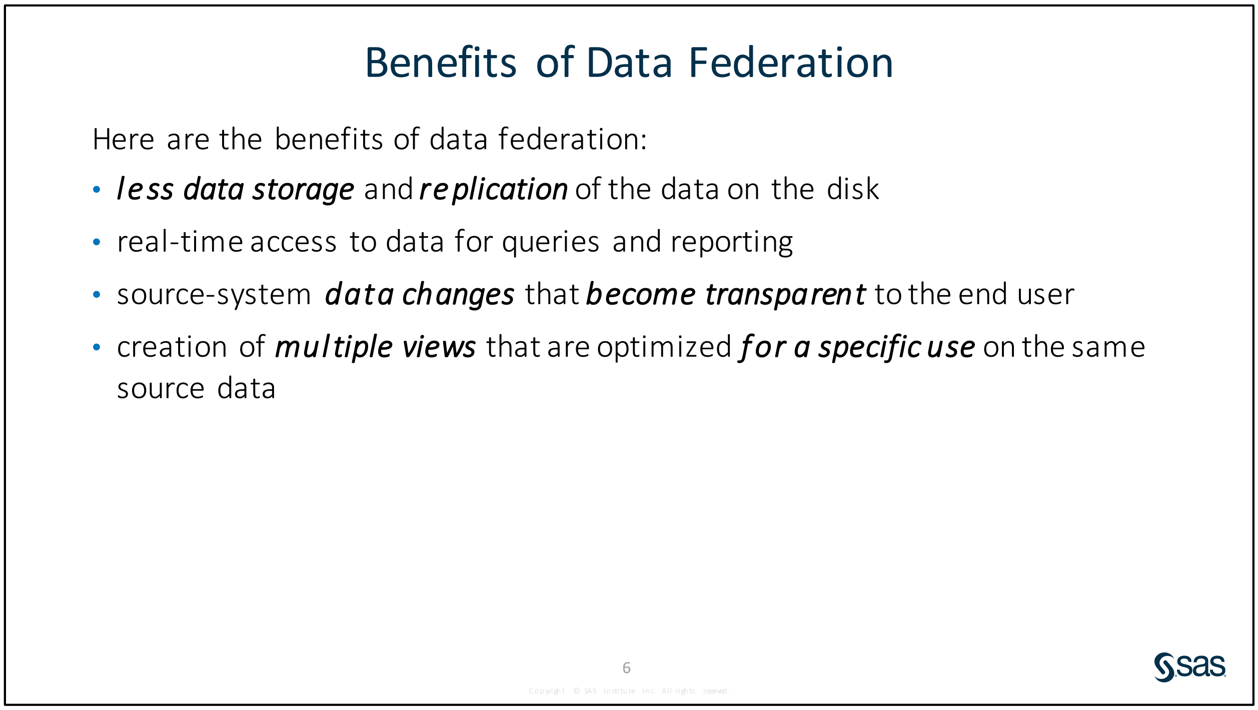
5

Here are the three main terms that you hear used in conjunction with SAS Federation Server:

data federation – the ability to use data across multiple heterogeneous source systems without physically having to move the data. The access to the data is provided via SQL views, and these views populate data only when the view is accessed.

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 database schemas/table names, as well as how data across multiple sources fits together.

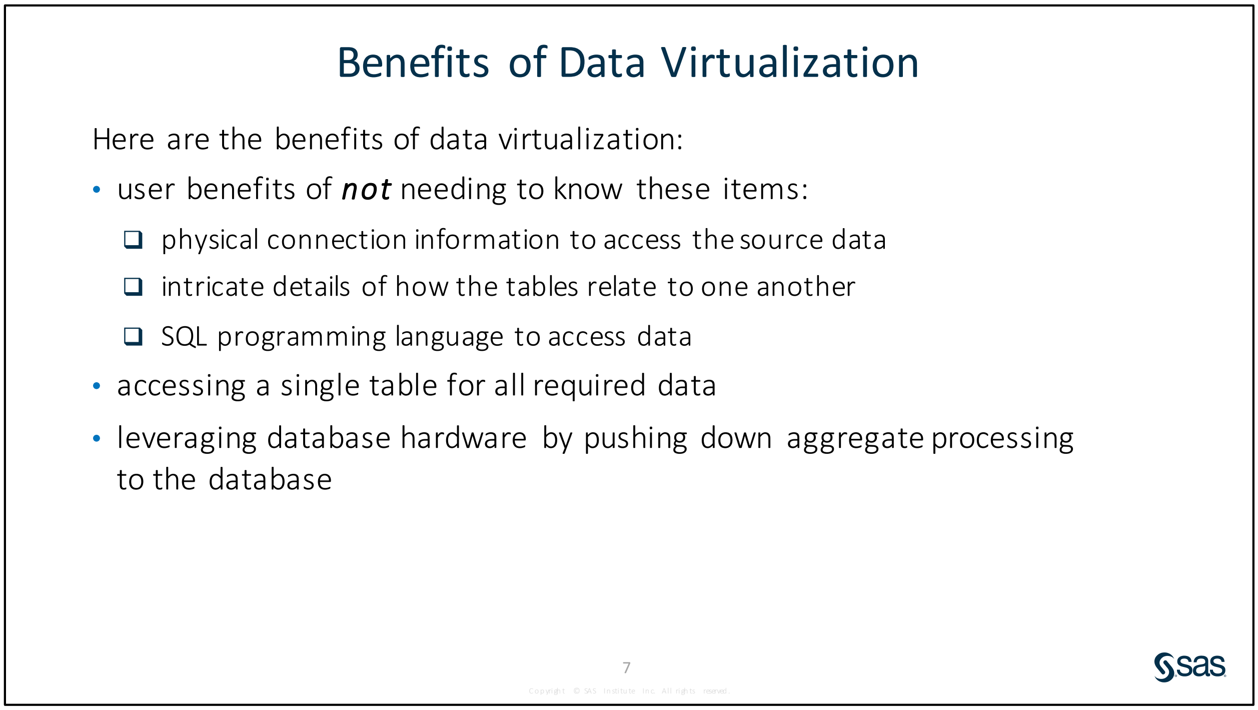
data disclosure control – modifying data such that no sensitive information remains. The challenge of data disclosure control lies in the ability to share information with users, while at the same time, protecting personally identifiable information (for example, account numbers, addresses, phone numbers, and taxpayer IDs) from the end user. It requires that those data elements be masked in some way from the end user.



Benefits of Data Federation

Here are the benefits of data federation:

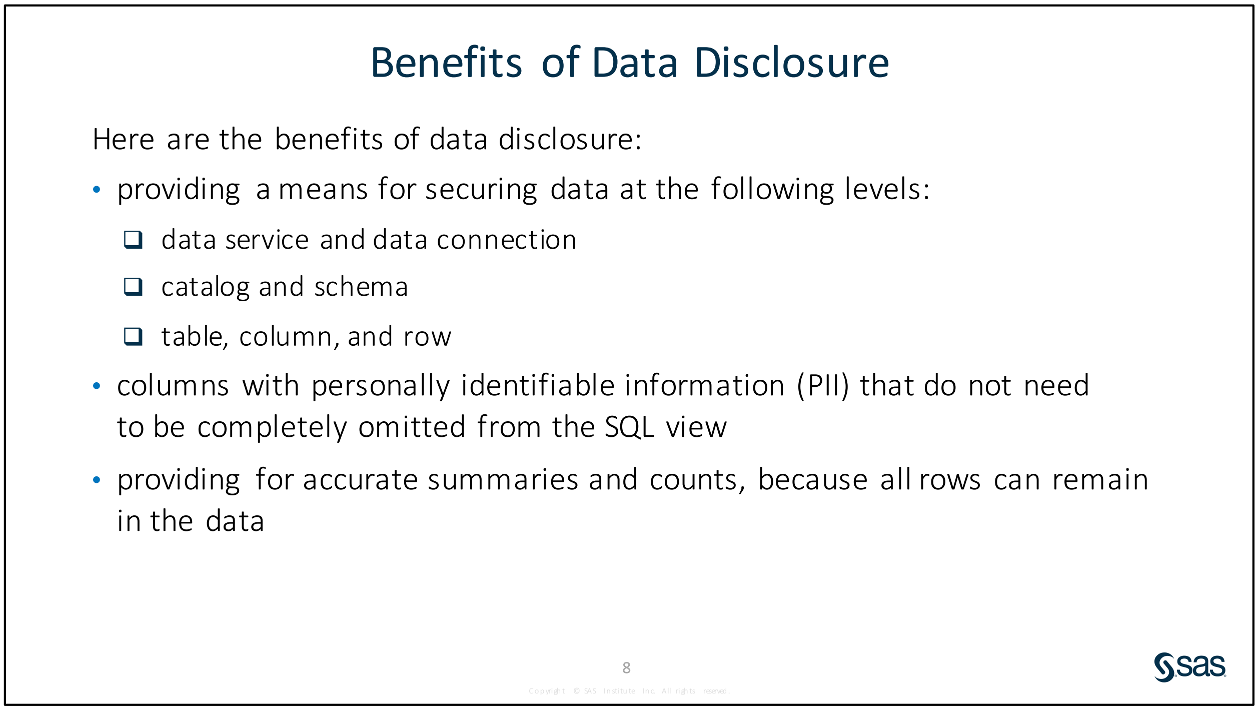
- ***less data storage*** and ***replication*** of the data on the disk
- real-time access to data for queries and reporting
- source-system ***data changes*** that ***become transparent*** to the end user
- creation of ***multiple views*** that are optimized ***for a specific use*** on the same source data



Benefits of Data Virtualization

Here are the benefits of data virtualization:

- user benefits of ***not*** needing to know these items:
  - ❑ physical connection information to access the source data
  - ❑ intricate details of how the tables relate to one another
  - ❑ SQL programming language to access data
- accessing a single table for all required data
- leveraging database hardware by pushing down aggregate processing to the database

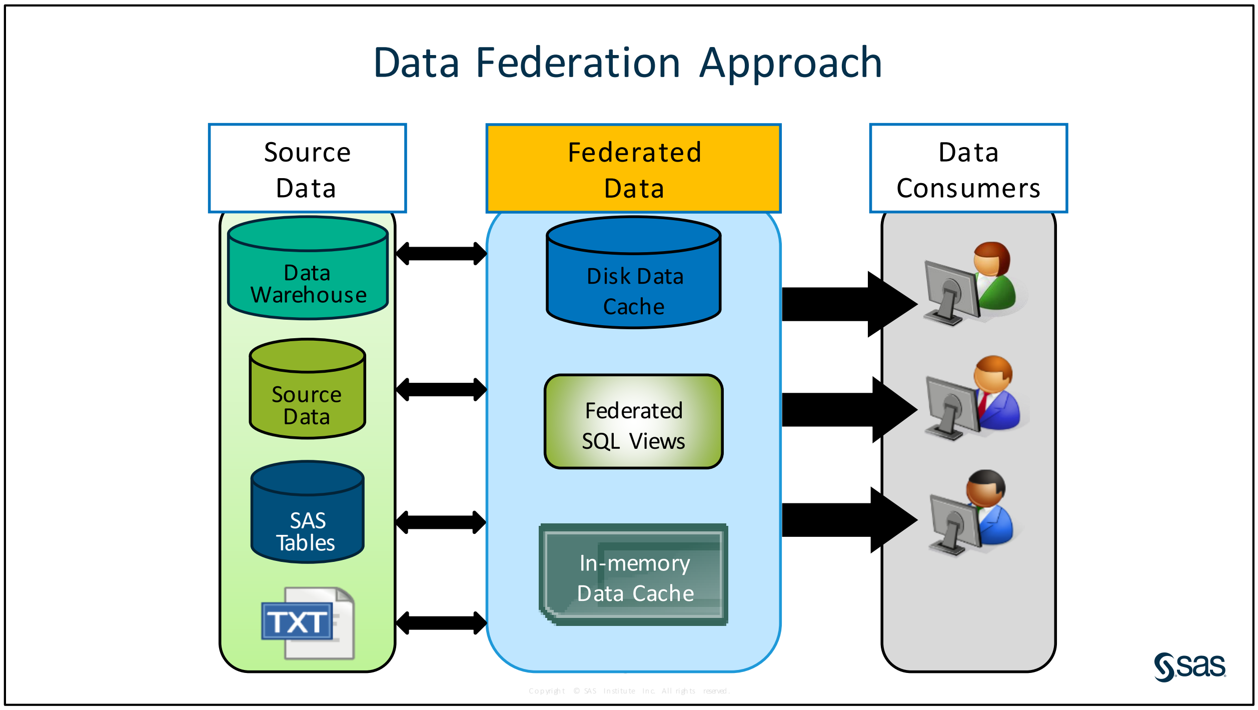


Benefits of Data Disclosure

Here are the benefits of data disclosure:

- providing a means for securing data at the following levels:
  - ❑ data service and data connection
  - ❑ catalog and schema
  - ❑ table, column, and row
- columns with personally identifiable information (PII) that do not need to be completely omitted from the SQL view
- providing for accurate summaries and counts, because all rows can remain in the data

8



Data Federation Approach

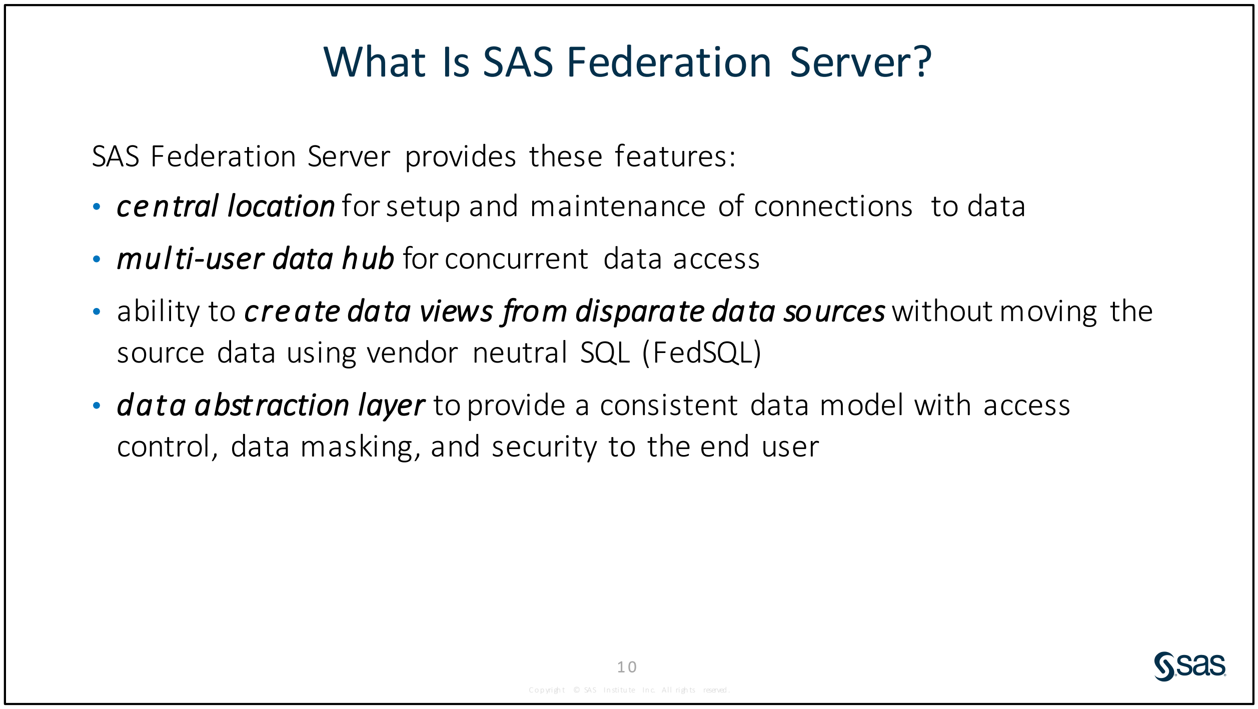
How does data federation help?

In concept, data federation accomplishes the same task as traditional data warehousing: it makes data from the disparate source systems throughout the enterprise available to the end users. These federated data sources could even be the same data marts and data warehouses that have traditionally been available to end users. However, it accomplishes this in a different way.

Rather than extracting the data from the source systems, transforming it, and then loading it into physical data tables, data federation leaves the physical source data in place. It provides access to the data through federated SQL views, which makes data available to users in real time. The views contain the technical information required for connecting to the data, the join information for assembling data from multiple tables, and any necessary transformations that need to take place on the data.

Data federation also has a granular security component that enables you to control access to the data at the data connection, table, row, and column level. This means that you can make more data available to more users, without compromising the security on the data.

The ability to cache data on the disk, or in memory, on the server, means that the data can be made available on a 24-7 basis.

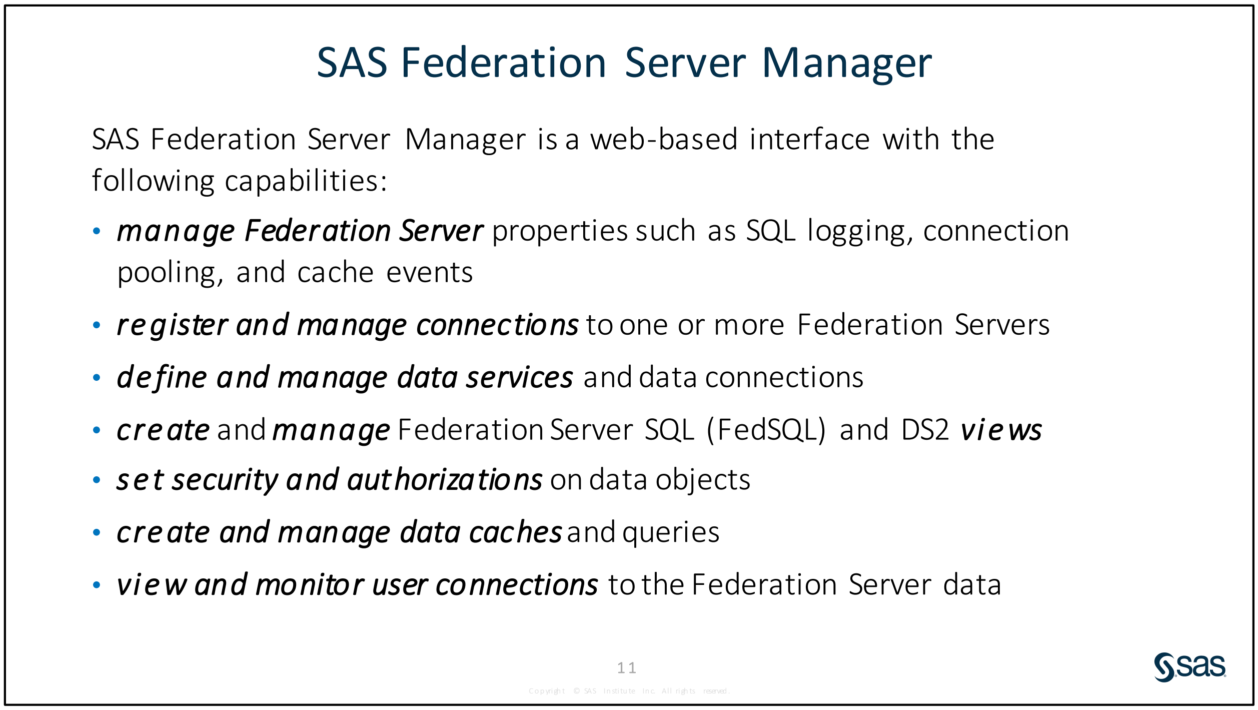


What Is SAS Federation Server?

SAS Federation Server provides these features:

- ***central location*** for setup and maintenance of connections to data
- ***multi-user data hub*** for concurrent data access
- ability to ***create data views from disparate data sources*** without moving the source data using vendor neutral SQL (FedSQL)
- ***data abstraction layer*** to provide a consistent data model with access control, data masking, and security to the end user

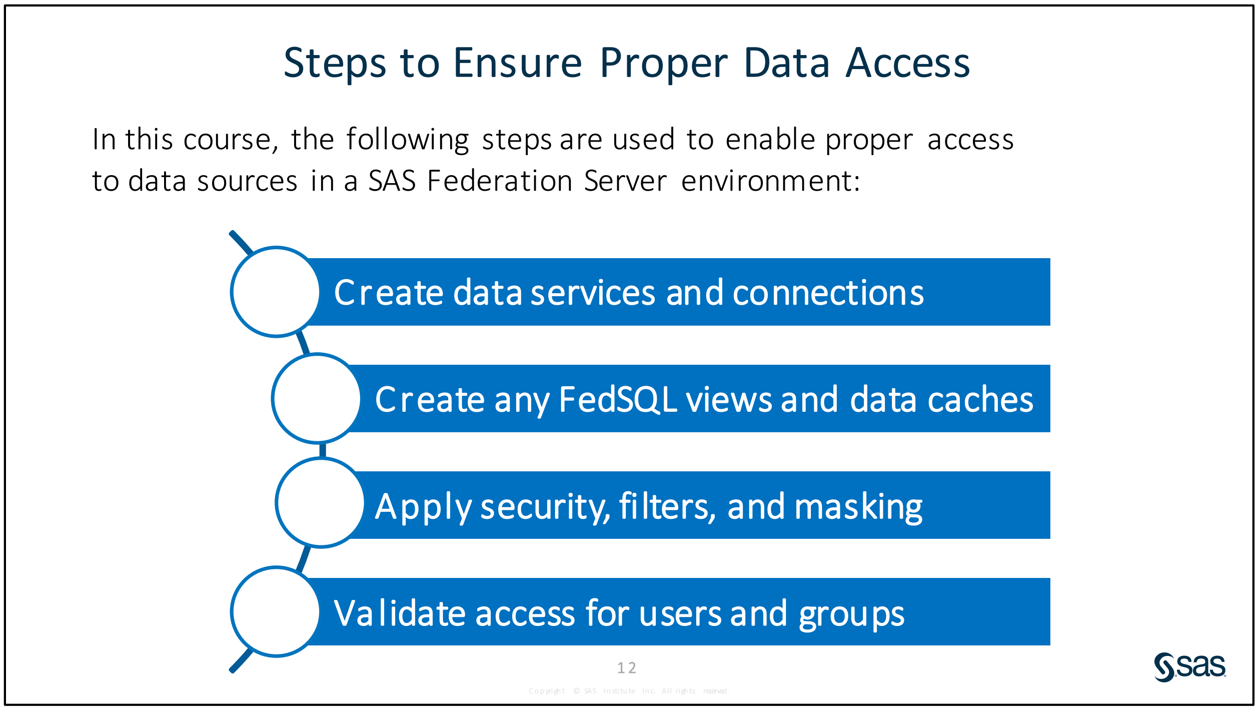
10



SAS Federation Server Manager

SAS Federation Server Manager is a web-based interface with the following capabilities:

- *manage Federation Server* properties such as SQL logging, connection pooling, and cache events
- *register and manage connections* to one or more Federation Servers
- *define and manage data services* and data connections
- *create* and *manage* Federation Server SQL (FedSQL) and DS2 *views*
- *set security and authorizations* on data objects
- *create and manage data caches* and queries
- *view and monitor user connections* to the Federation Server data



Steps to Ensure Proper Data Access

In this course, the following steps are used to enable proper access to data sources in a SAS Federation Server environment:

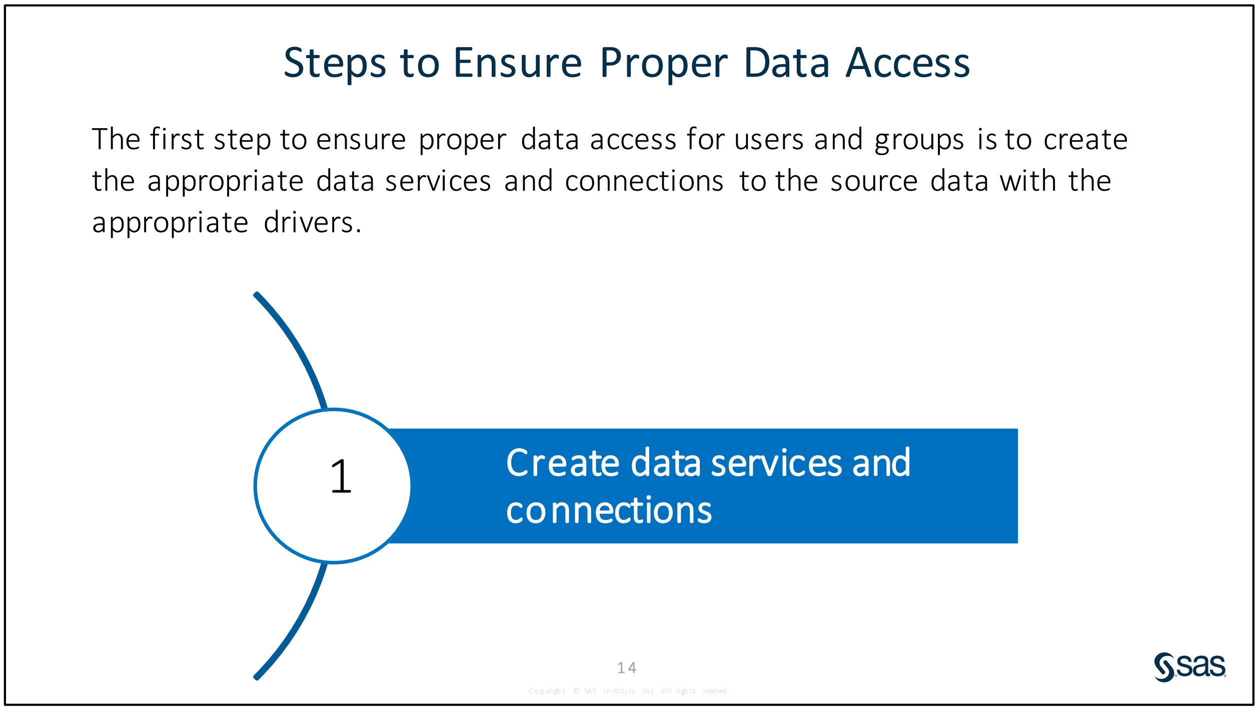
Create data services and connections

Create any FedSQL views and data caches

Apply security, filters, and masking

Validate access for users and groups

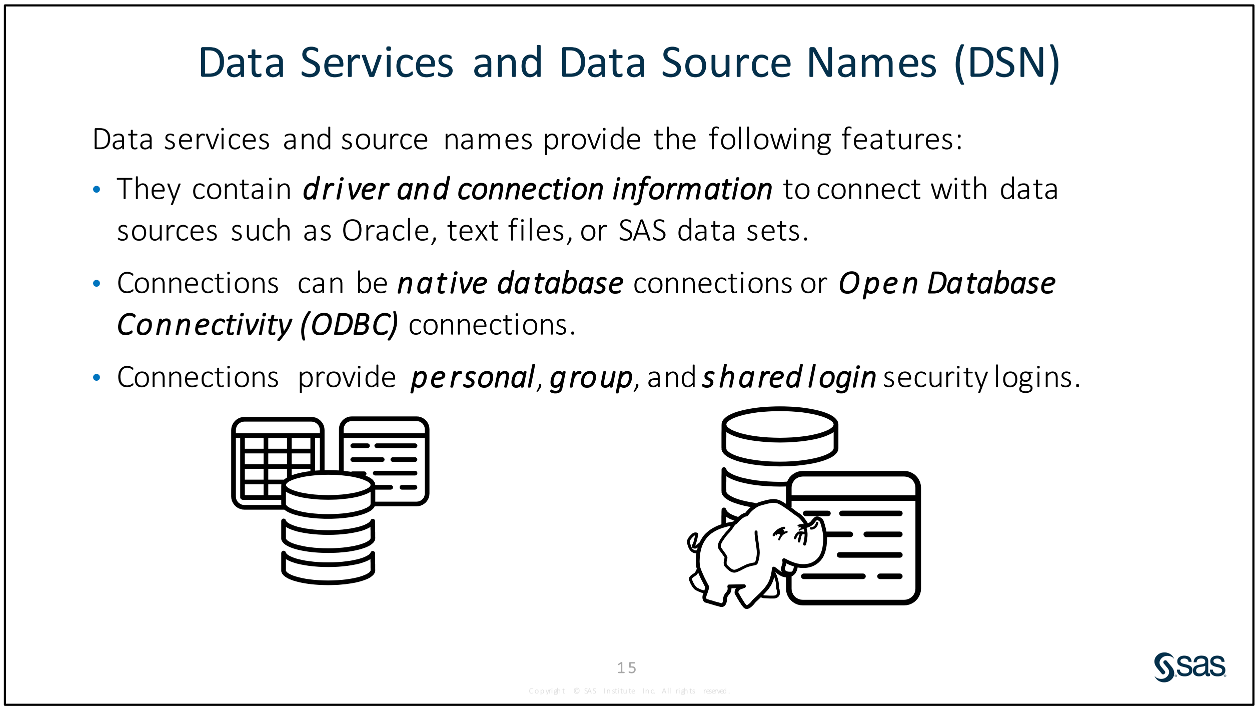
1.2 Creating Data Services and Connections



Steps to Ensure Proper Data Access

The first step to ensure proper data access for users and groups is to create the appropriate data services and connections to the source data with the appropriate drivers.

1
Create data services and connections



Data Services and Data Source Names (DSN)

Data services and source names provide the following features:

- They contain ***driver and connection information*** to connect with data sources such as Oracle, text files, or SAS data sets.
- Connections can be ***native database*** connections or ***Open Database Connectivity (ODBC)*** connections.
- Connections provide ***personal***, ***group***, and ***shared login*** security logins.

SAS Federation Server data services contain information that identifies the location of source data tables. Data services can be created for third-party relational databases, as well as SAS data directories.

Database services are created and managed through a set of properties that are available in SAS Federation Server Manager. There are two options available for creating data services on the SAS Federation Server:

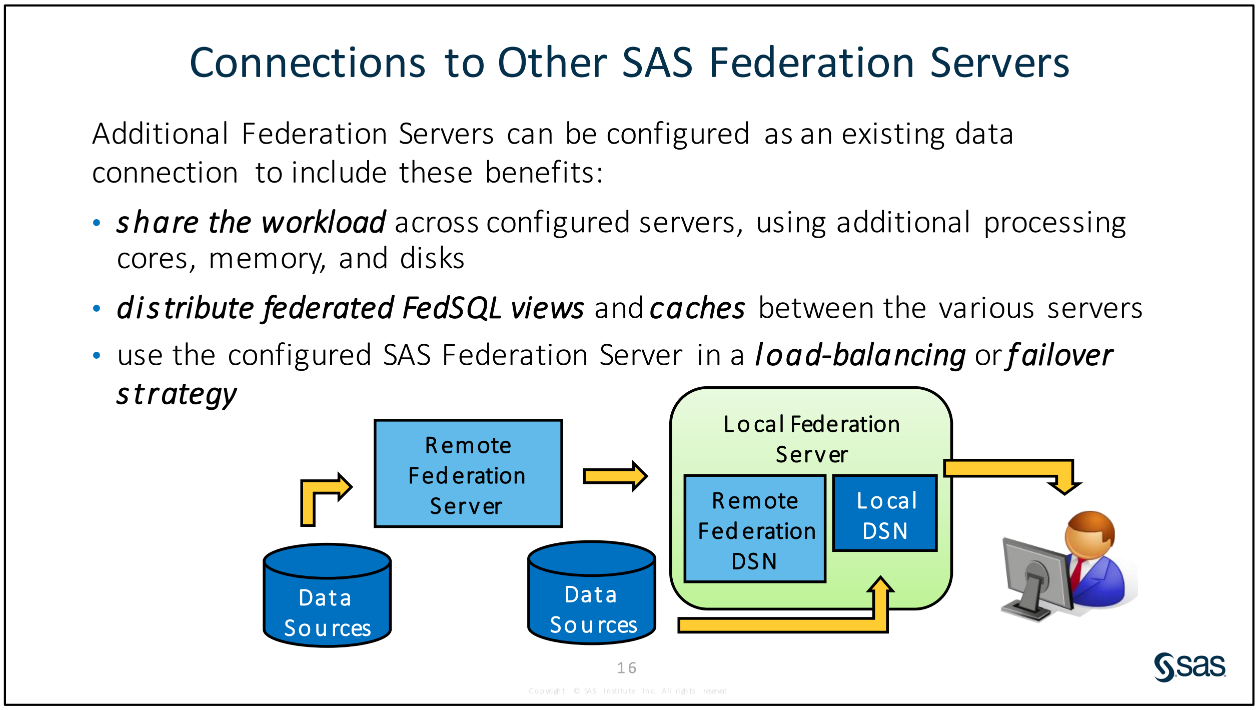
- programmatically, using administration DDL statements
- using the provided functionality in the SAS Federation Server Manager

Each time that a new data service is created on SAS Federation Server, a new DSN is created for that data service.

- This DSN name remains unchanged, even if you change the name of the data service itself.

There is a ***BASE*** data service that is created automatically the first time Federation Server is started. This data service appears automatically in the navigator panel of the SAS Federation Server Manager the first time you connect up to Federation Server.

There is also an ***SQL_LOG*** data service and DSN that are automatically created. This is the configuration file that is used to facilitate SQL logging, and it is located in the ***/etc*** directory of the installation path.



Connections to Other SAS Federation Servers

Additional Federation Servers can be configured as an existing data connection to include these benefits:

- ***share the workload*** across configured servers, using additional processing cores, memory, and disks
- ***distribute federated FedSQL views*** and ***caches*** between the various servers
- use the configured SAS Federation Server in a ***load-balancing*** or ***failover strategy***

16

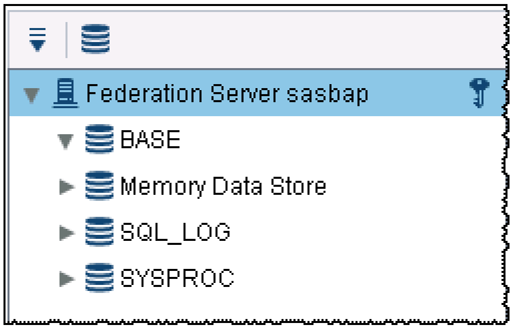
When working with FedSQL views, note that views cannot be created, modified, and subsequently cached if you are using a remote SAS Federation Server that is connecting with a SAS Federation Server data service. You should create and cache these views from a local SAS Federation Server only. FedSQL views should also be administered on a local SAS Federation Server (for example, changing a view from definer's rights to invoker's rights).

Creating a Federated Data Connection from Disparate Sources

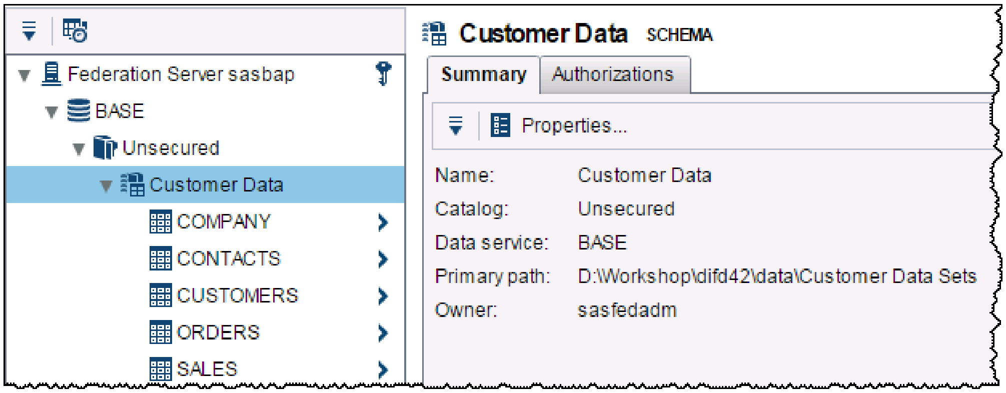
This demonstration illustrates creating a federated data connection from disparate data sources using a SAS data set and an Oracle relational database table.

Exploring the BASE SAS Data Service

1. Log on to the SAS Federation Server Manager as the user **sasfedadm@saspw** with the password **Student1**.
2. Log on to the Federation Server sasbap as **sasfedadm@saspw**.
3. Explore the SAS Federation Server Manager data connections tree.



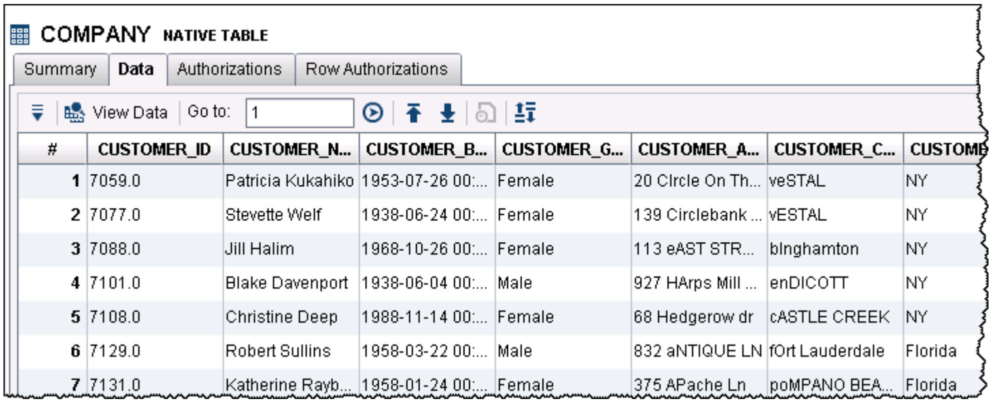
4. Navigate to **BASE** ⇨ **Unsecured** ⇨ **Customer Data**.
5. Double-click **Customer Data** to expand and view the tables.



Name: Customer Data
Catalog: Unsecured
Data service: BASE
Primary path: D:\Workshop\difd42\data\Customer Data Sets
Owner: sasfedadm

   a. Select **Company**.
   b. Click the **Data** tab.

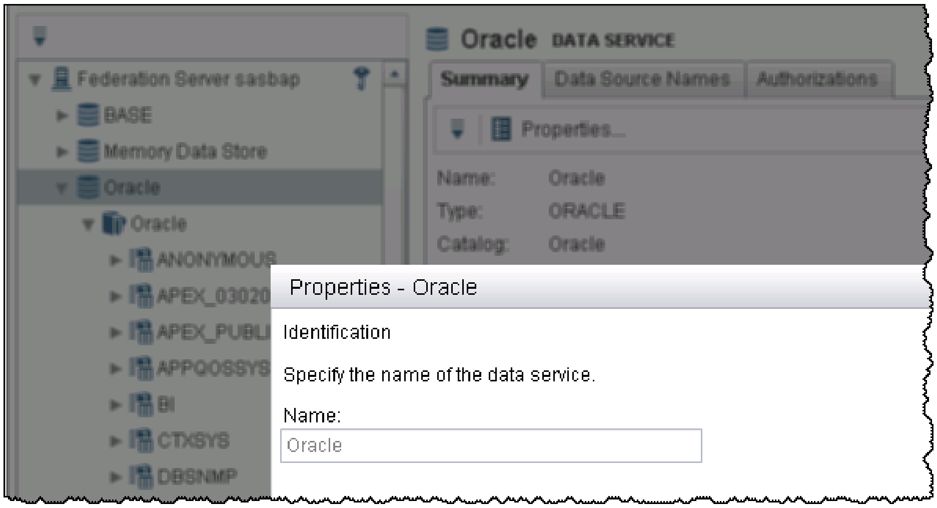
c. Click **View Data** to verify that the user, **student**, can view the data.



#	CUSTOMER_ID	CUSTOMER_N...	CUSTOMER_B...	CUSTOMER_G...	CUSTOMER_A...	CUSTOMER_C...	CUSTOME
1	7059.0	Patricia Kukahiko	1953-07-26 00:...	Female	20 CIrcle On Th...	veSTAL	NY
2	7077.0	Stevette Welf	1938-06-24 00:...	Female	139 Circlebank ...	vESTAL	NY
3	7088.0	Jill Halim	1968-10-26 00:...	Female	113 eAST STR...	bInghamton	NY
4	7101.0	Blake Davenport	1938-06-04 00:...	Male	927 HArps Mill ...	enDICOTT	NY
5	7108.0	Christine Deep	1988-11-14 00:...	Female	68 Hedgerow dr	cASTLE CREEK	NY
6	7129.0	Robert Sullins	1958-03-22 00:...	Male	832 aNTIQUE LN	fOrt Lauderdale	Florida
7	7131.0	Katherine Rayb...	1958-01-24 00:...	Female	375 APache Ln	poMPANO BEA...	Florida

Exploring the Oracle Data Service

1. Explore the Oracle data service properties.



Name: Oracle

Type: ORACLE

Catalog: Oracle

Properties - Oracle

Identification

Specify the name of the data service.

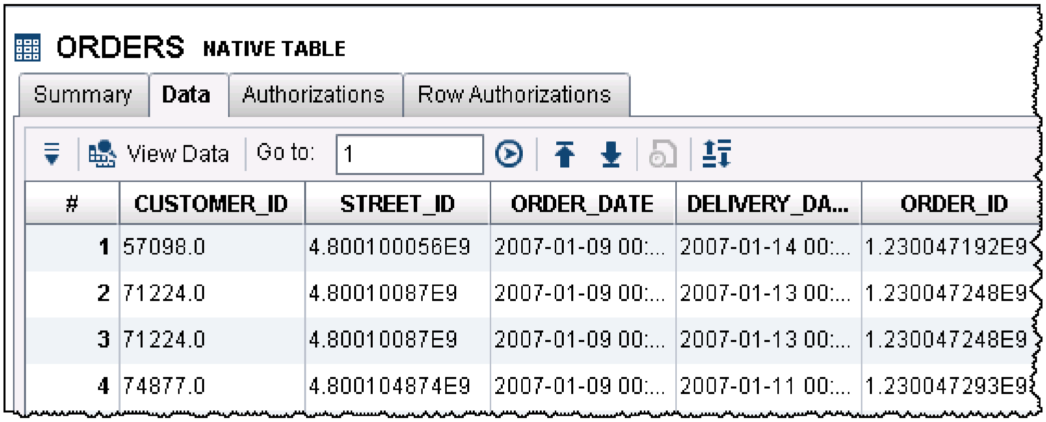
Name:

Oracle

2. Navigate to **Oracle** ⇨ **Oracle** ⇨ **STUDENT**.
3. Click the **ORDERS** table.

   a. Click the **Data** tab.

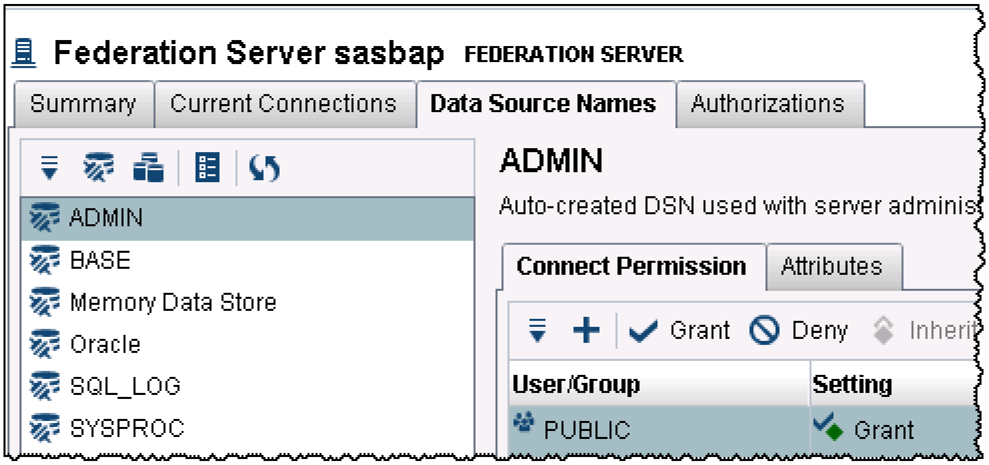
   b. Click **View Data** and verify the content.



#	CUSTOMER_ID	STREET_ID	ORDER_DATE	DELIVERY_DA...	ORDER_ID
1	57098.0	4.800100056E9	2007-01-09 00:...	2007-01-14 00:...	1.230047192E9
2	71224.0	4.80010087E9	2007-01-09 00:...	2007-01-13 00:...	1.230047248E9
3	71224.0	4.80010087E9	2007-01-09 00:...	2007-01-13 00:...	1.230047248E9
4	74877.0	4.800104874E9	2007-01-09 00:...	2007-01-11 00:...	1.230047293E9

Creating a Federated Data Connection Using SAS and Oracle Tables

1. Select **Federation Server sasbap**.
2. Click the **Data Source Names** tab.



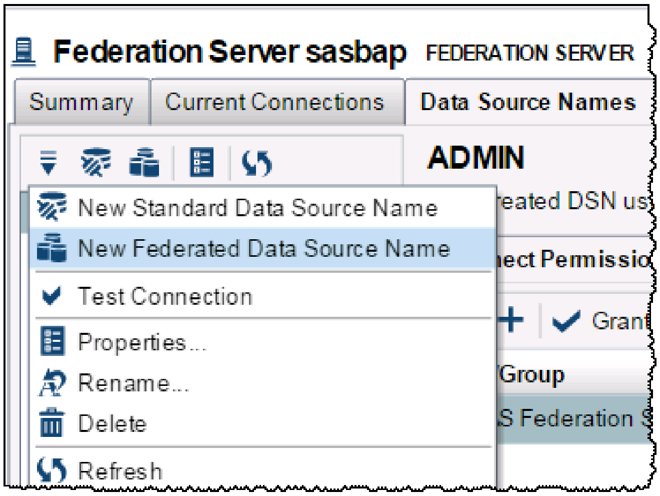
ADMIN

Auto-created DSN used with server adminis

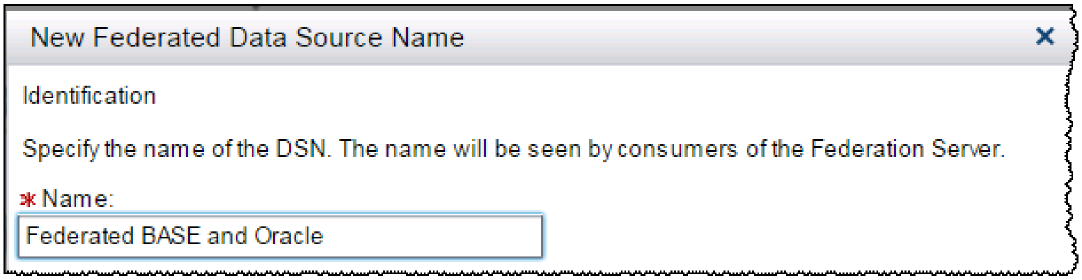
User/Group	Setting
PUBLIC	Grant

   a. Click ▾ (action menu down arrow).

   b. Select **New Federated Data Source Name**.



   c. Enter **Federated BASE and Oracle** in the **Name** field.



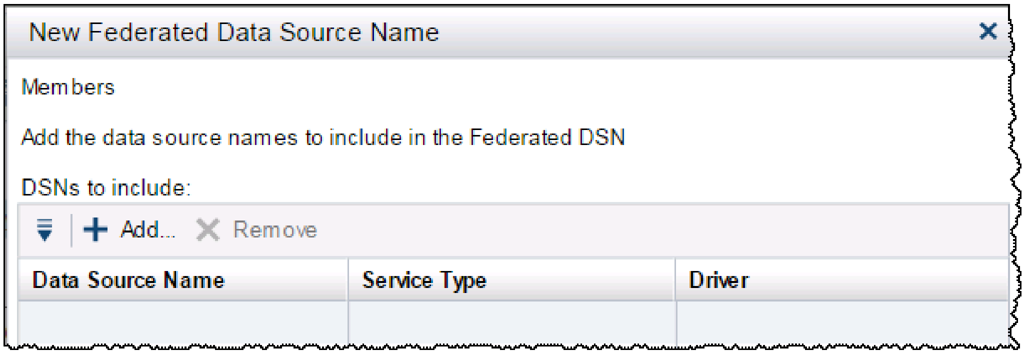
Identification

Specify the name of the DSN. The name will be seen by consumers of the Federation Server.

* Name: Federated BASE and Oracle

   d. Click **Next**.

   e. Click **Add**.



New Federated Data Source Name

Members

Add the data source names to include in the Federated DSN

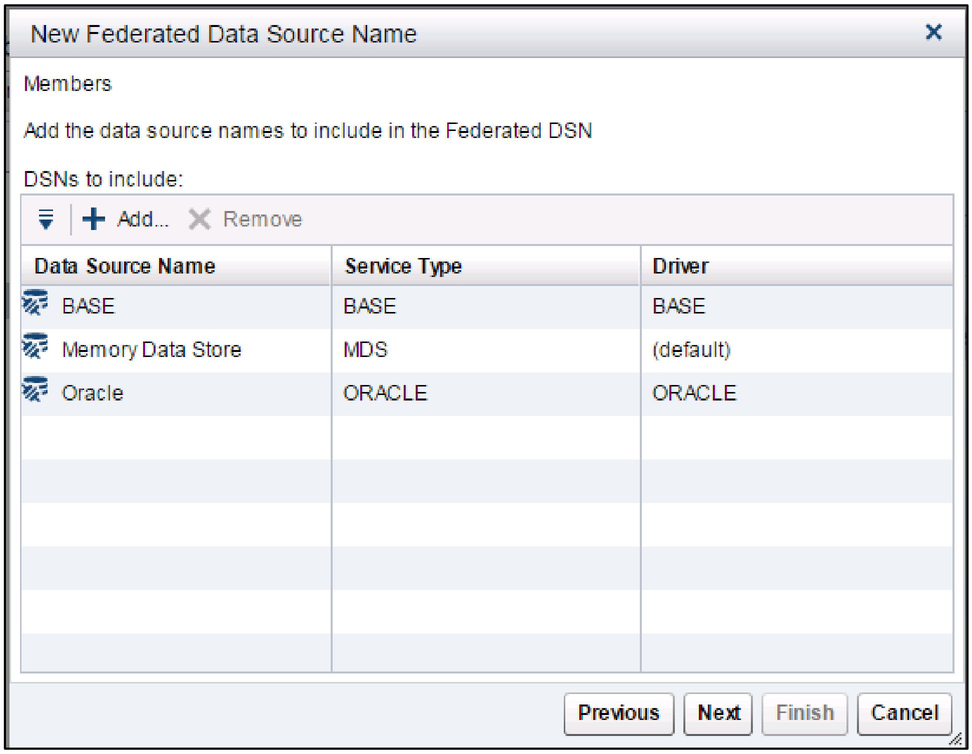
DSNs to include:

Data Source Name	Service Type	Driver

1) Click **BASE**.

2) Click **Memory Data Store**.

3) Click **Oracle**.



New Federated Data Source Name

Members

Add the data source names to include in the Federated DSN

DSNs to include:

Data Source Name	Service Type	Driver
BASE	BASE	BASE
Memory Data Store	MDS	(default)
Oracle	ORACLE	ORACLE

4) Click **OK**.

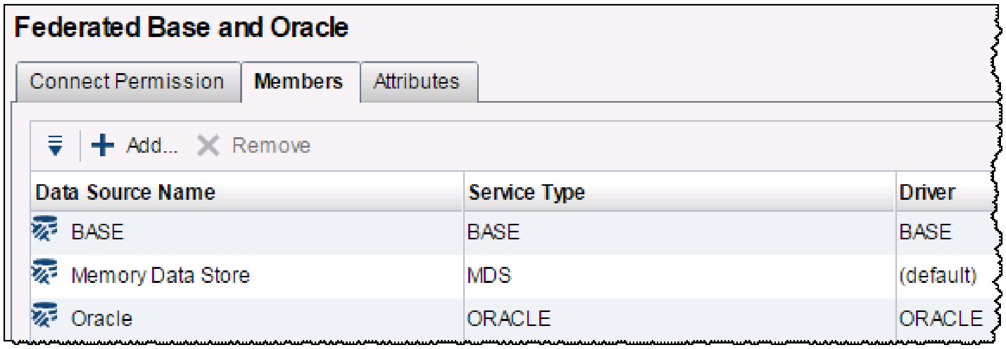
f. Click **Next**.

g. Click **Next** in the Security panel.

h. Click **Next** in the Syntax panel.

i. Click **Finish**.

3. Click the **Members** tab of the new DSN.



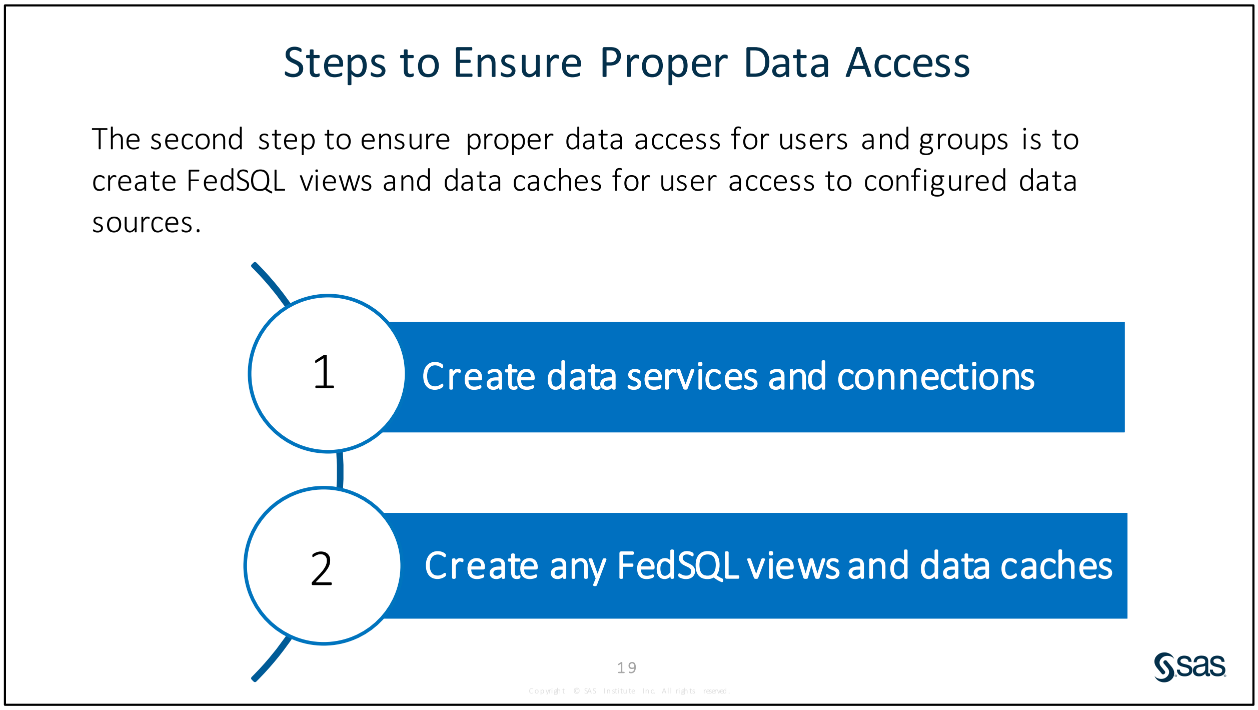
Federated Base and Oracle

Data Source Name	Service Type	Driver
BASE	BASE	BASE
Memory Data Store	MDS	(default)
Oracle	ORACLE	ORACLE

Note: The federated connection is used later in a FedSQL view.

End of Demonstration

1.3 Creating and Caching Federated Data Views

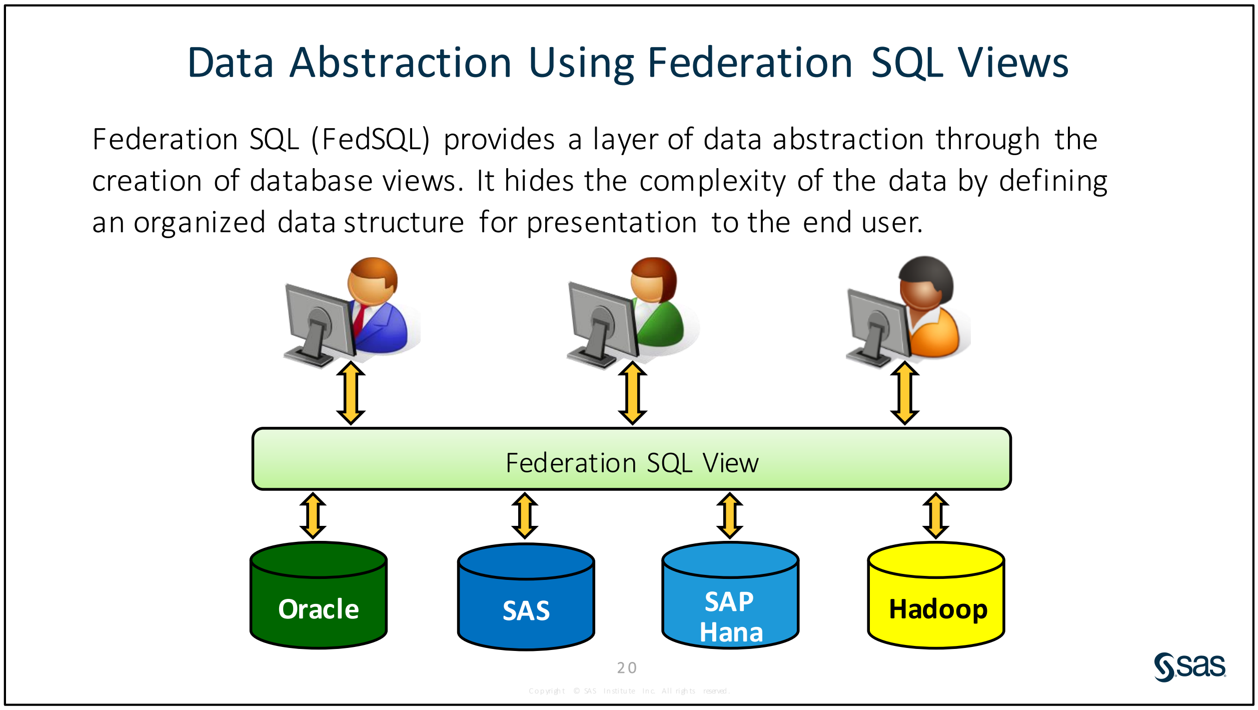


Steps to Ensure Proper Data Access

The second step to ensure proper data access for users and groups is to create FedSQL views and data caches for user access to configured data sources.

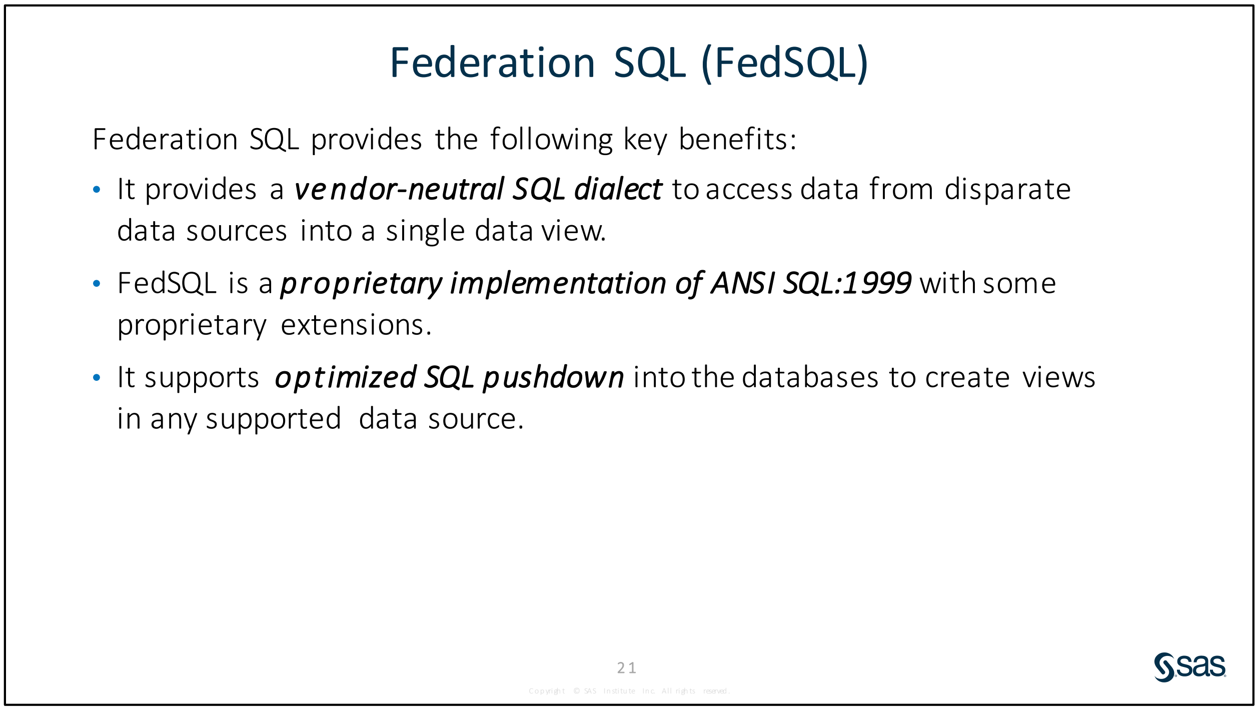
1 Create data services and connections

2 Create any FedSQL views and data caches



Data Abstraction Using Federation SQL Views

Federation SQL (FedSQL) provides a layer of data abstraction through the creation of database views. It hides the complexity of the data by defining an organized data structure for presentation to the end user.



Federation SQL (FedSQL)

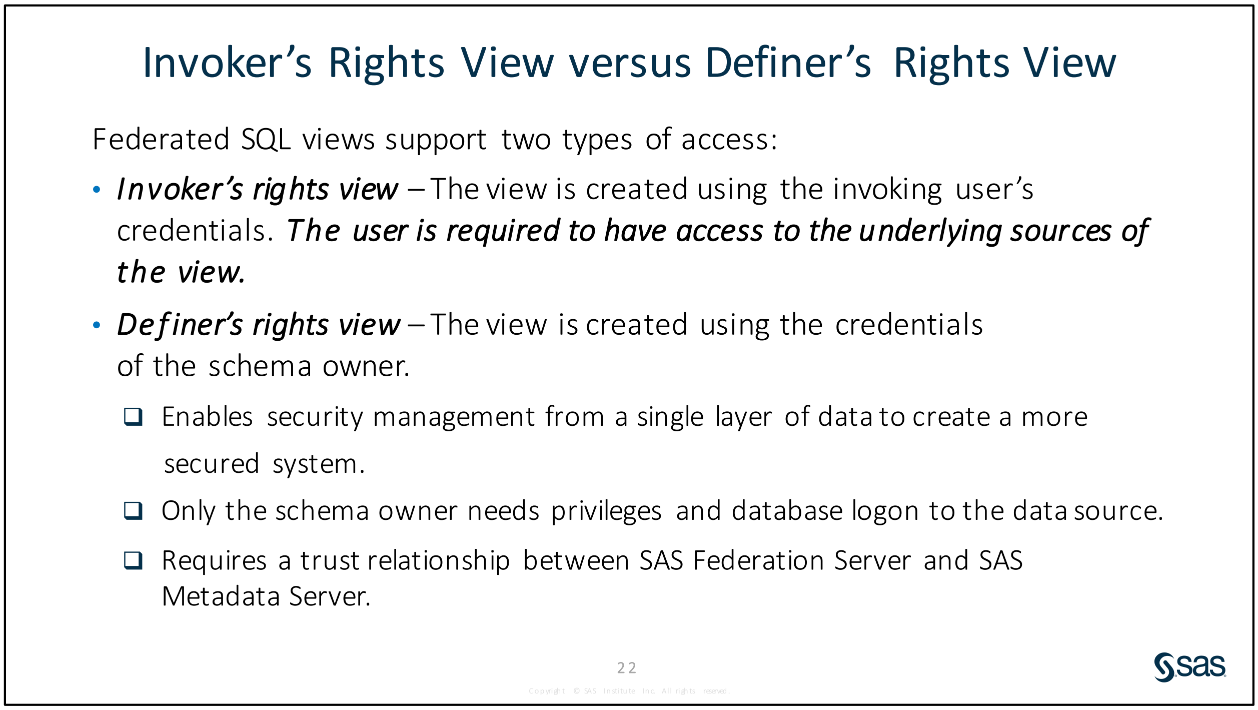
Federation SQL provides the following key benefits:

- It provides a ***vendor-neutral SQL dialect*** to access data from disparate data sources into a single data view.
- FedSQL is a ***proprietary implementation of ANSI SQL:1999*** with some proprietary extensions.
- It supports ***optimized SQL pushdown*** into the databases to create views in any supported data source.

21

FedSQL provides many benefits if you are working in an environment in which you need more features than are provided in the SQL procedure.

- FedSQL conforms to the ANSI SQL:1999 core standard. This conformance enables it to process queries in its own language and the native languages of other data sources that conform to the standard.
- FedSQL supports many more data types than previous SAS SQL implementations. Traditional data source access through SAS/ACCESS translates target data source data types to and from two legacy SAS data types: SAS numeric and SAS character. When FedSQL connects to a data source, the language matches or translates the target data source's definition to these data types, as appropriate, which allows greater precision.
- FedSQL handles federated queries. With the traditional DATA step or the SQL procedure, a SAS/ACCESS LIBNAME engine can access only the data of its intended data source.
- The FedSQL language can create data in any of the supported data sources, even if the target data source is not represented in a query. This enables you to store data in the data source that most closely meets the needs of your application.



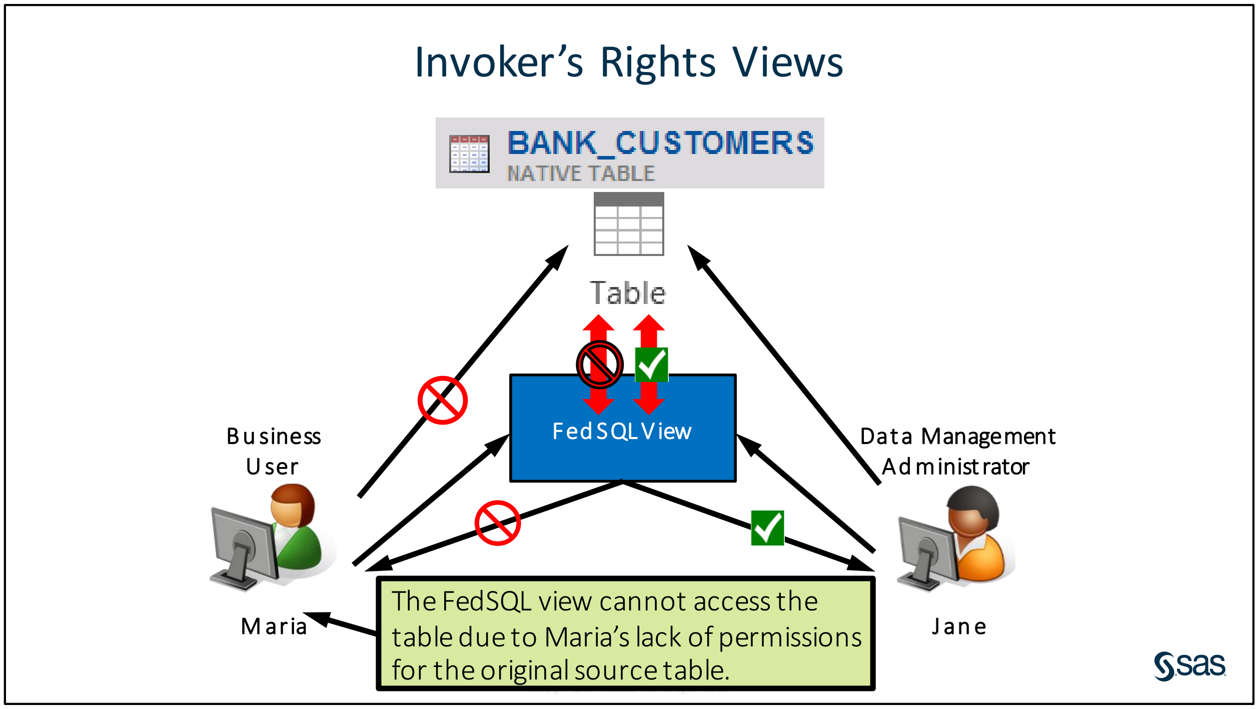
Invoker’s Rights View versus Definer’s Rights View

Federated SQL views support two types of access:

- *Invoker’s rights view* – The view is created using the invoking user’s credentials. ***The user is required to have access to the underlying sources of the view.***
- *Definer’s rights view* – The view is created using the credentials of the schema owner.
  - ❑ Enables security management from a single layer of data to create a more secured system.
  - ❑ Only the schema owner needs privileges and database logon to the data source.
  - ❑ Requires a trust relationship between SAS Federation Server and SAS Metadata Server.

22

Because definer's rights views effectively allow a user to impersonate the schema owner, the creation of definer's rights views requires special consideration. It can be desirable to grant CREATE VIEW privilege to a user, but only for the intention of creating invoker's rights views.



Invoker’s Rights Views

BANK_CUSTOMERS
NATIVE TABLE

Table

FedSQL View

Business User

Maria

Data Management Administrator

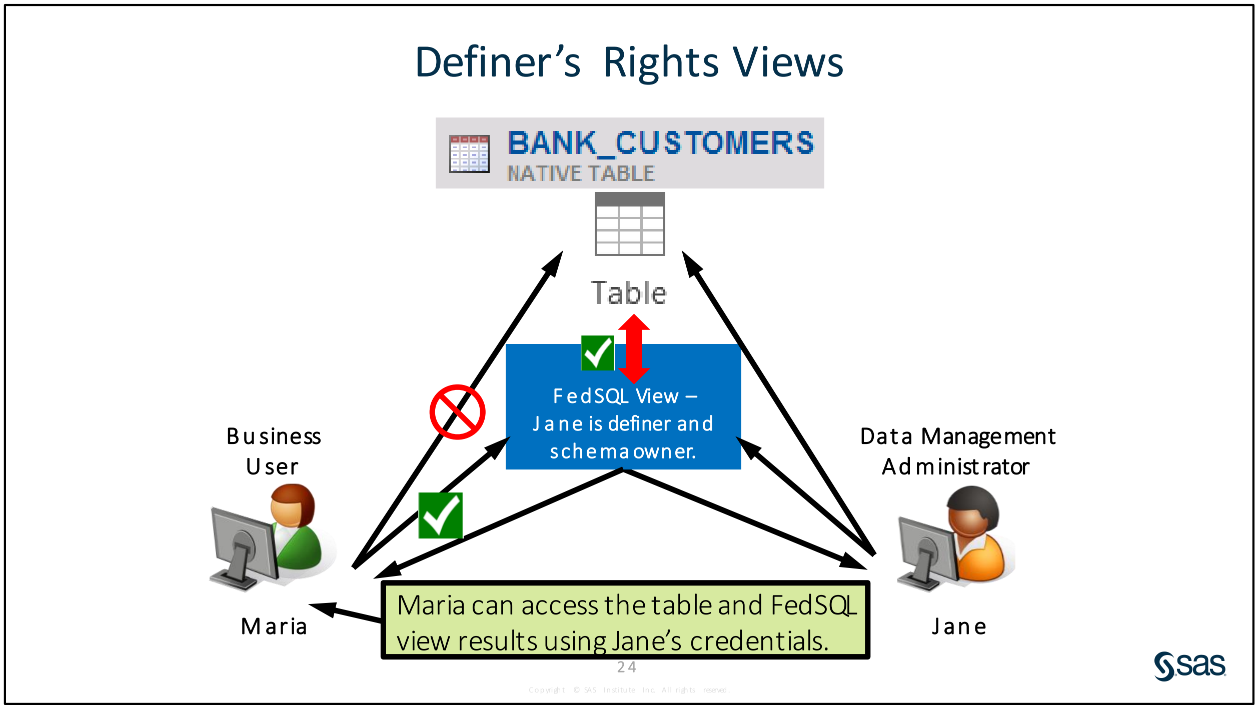
Jane

The FedSQL view cannot access the table due to Maria’s lack of permissions for the original source table.

In an attempt to make some of the data available to Maria, Jane creates a FedSQL query to return only selected columns of data to Maria. This will effectively keep Maria from seeing columns (and, potentially, rows) of information that she should not see. Maria still does not have access to the physical data table itself, but she does have access to the FedSQL view defined by Jane. When

Maria attempts to access the data in the FedSQL view, she will be denied access because she does not have access to the physical table that populates the view.

Note: As invoker, Maria cannot view the actual FedSQL query.

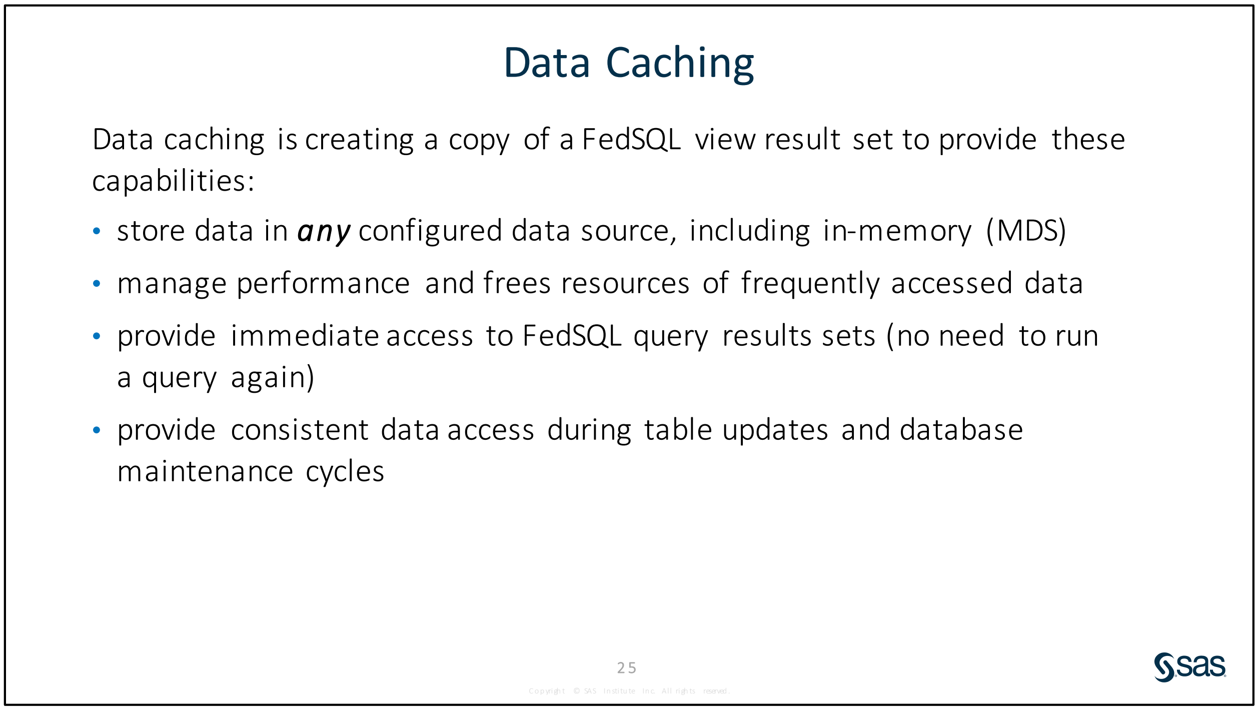


Definer’s Rights Views

BANK_CUSTOMERS
NATIVE TABLE
Table
FedSQL View – Jane is definer and schema owner.
Business User
Maria
Data Management Administrator
Jane
24

Maria can access the table and FedSQL view results using Jane’s credentials.

In the example above, Jane’s user credentials would be attached to the FedSQL query, because she used Definer’s Rights when generating the FedSQL view. Now Maria can see the data created from the view with applied column- and row-level authorizations. Ideally, Maria should be given permission to access the view but not see the actual FedSQL query used to generate the view. The FedSQL query could contain encryption key information.

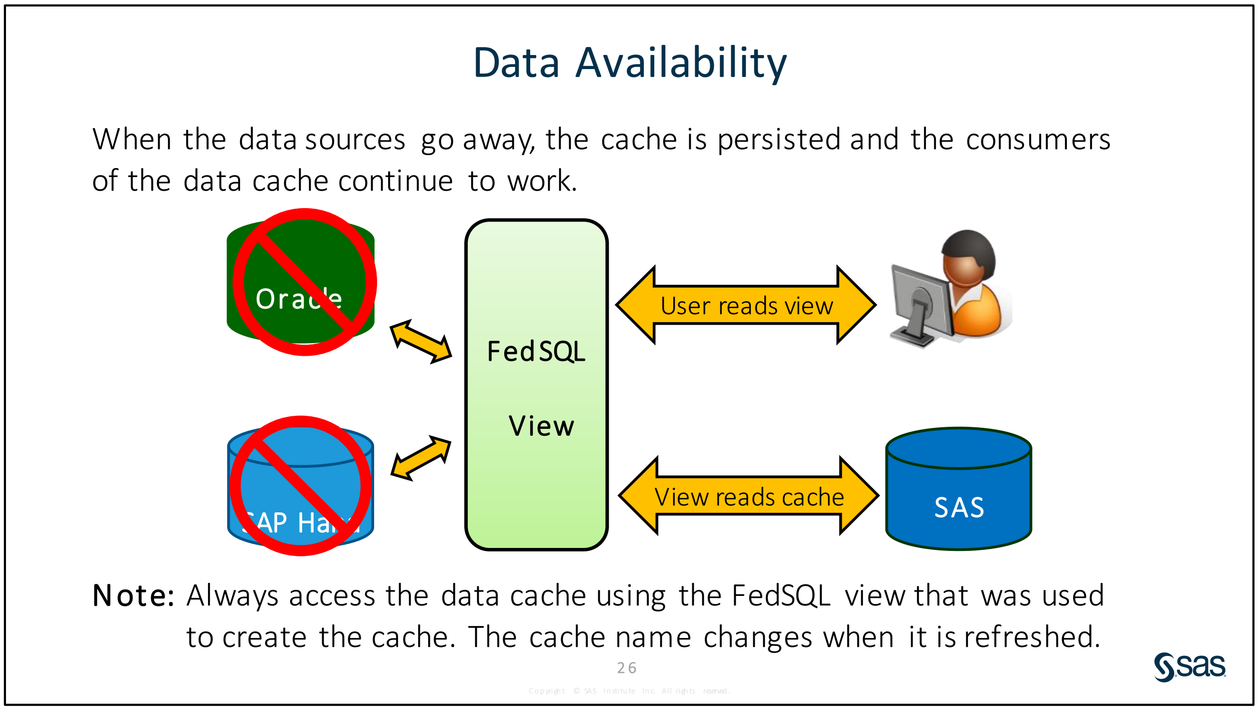


Data Caching

Data caching is creating a copy of a FedSQL view result set to provide these capabilities:

- store data in ***any*** configured data source, including in-memory (MDS)
- manage performance and frees resources of frequently accessed data
- provide immediate access to FedSQL query results sets (no need to run a query again)
- provide consistent data access during table updates and database maintenance cycles

25



Data Availability

When the data sources go away, the cache is persisted and the consumers of the data cache continue to work.

Note: Always access the data cache using the FedSQL view that was used to create the cache. The cache name changes when it is refreshed.

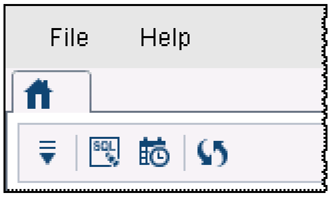
26

Caching data or results sets ensures 24/7 data access for business users and reports, especially during database maintenance cycles or during peak data access times to the tables on the relational databases. When the data needs refreshing, a manual or scheduled refresh updates the cache from the contributing sources.

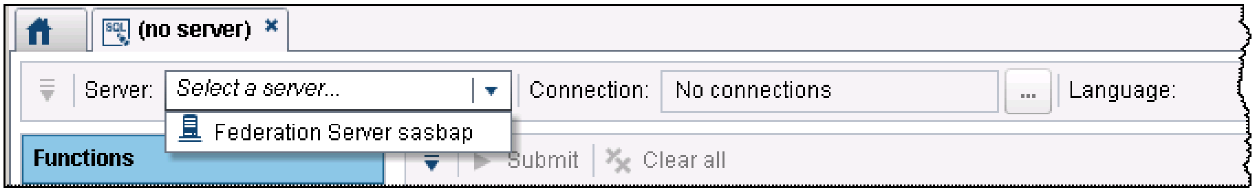
Creating and Caching a FedSQL View for Performance and Availability

This demonstration illustrates how to create a FedSQL view from a federated data source and cache the view in-memory for better performance and increased availability to users.

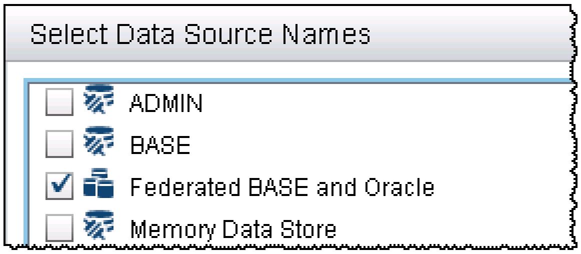
1. Click (**Console**) to open the SQL Console Editor.



2. Select **Federation Server sasbap** from the Server menu.



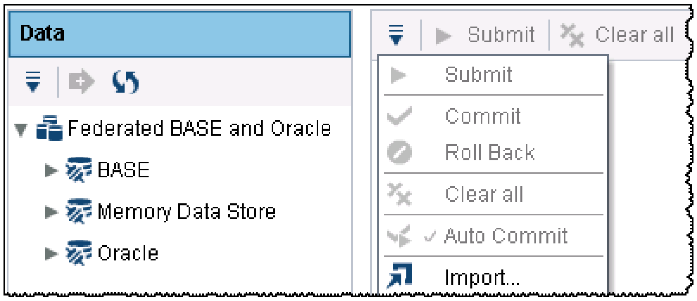
3. Click the **Connection** ellipsis button.
4. Clear the **ADMIN** selection.
5. Select **Federated BASE and Oracle**.



Select Data Source Names

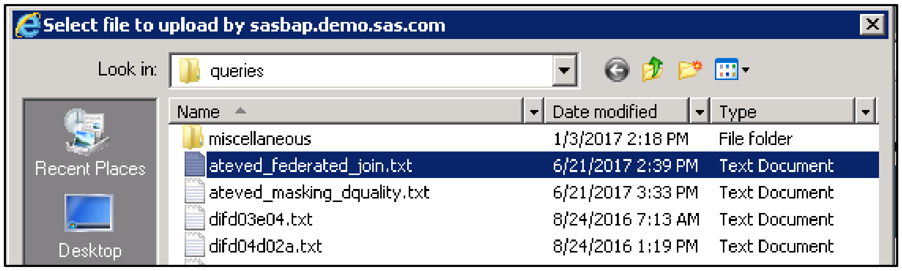
□ ADMIN
□ BASE
☑ Federated BASE and Oracle
□ Memory Data Store

6. Click **OK**.
7. Select **Import** on the SQL Editor menu.



   a. Navigate to **D:\Workshop\difd42\solutions\queries**.

b. Select **ateved_federated_join.txt.**



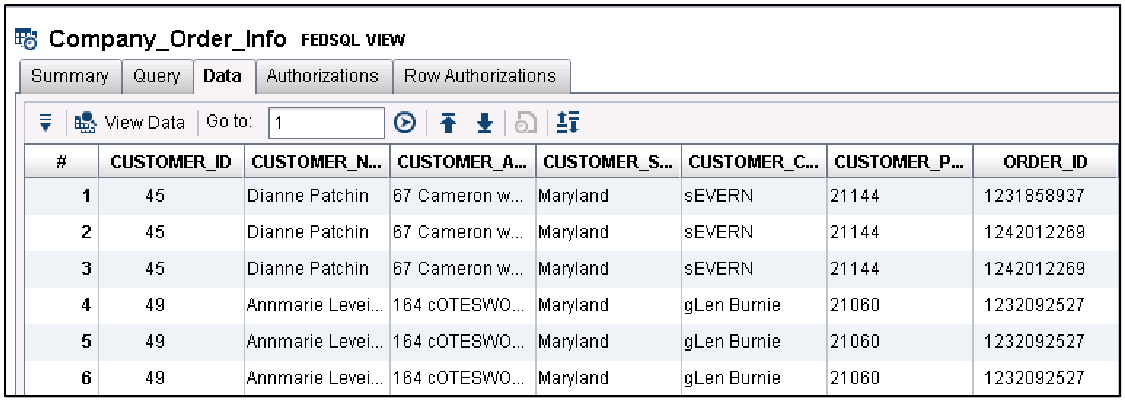
Name	Date modified	Type
miscellaneous	1/3/2017 2:18 PM	File folder
ateved_federated_join.txt	6/21/2017 2:39 PM	Text Document
ateved_masking_dquality.txt	6/21/2017 3:33 PM	Text Document
difd03e04.txt	8/24/2016 7:13 AM	Text Document
difd04d02a.txt	8/24/2016 1:19 PM	Text Document

c. Click **Open**.

8. Click **OK** again to import and review the code.
9. Click **Submit**
10. Click the **Home** tab.
11. Select **Memory Data Store** ⇨ **MDS** ⇨ **Views** ⇨ **Company_Order_Info**.

a. Click the **Data** tab.

b. Click **View Data**.



#	CUSTOMER_ID	CUSTOMER_N...	CUSTOMER_A...	CUSTOMER_S...	CUSTOMER_C...	CUSTOMER_P...	ORDER_ID
1	45	Dianne Patchin	67 Cameron w...	Maryland	sEVERN	21144	1231858937
2	45	Dianne Patchin	67 Cameron w...	Maryland	sEVERN	21144	1242012269
3	45	Dianne Patchin	67 Cameron w...	Maryland	sEVERN	21144	1242012269
4	49	Annmarie Levei...	164 cOTESWO...	Maryland	gLen Burnie	21060	1232092527
5	49	Annmarie Levei...	164 cOTESWO...	Maryland	gLen Burnie	21060	1232092527
6	49	Annmarie Levei...	164 cOTESWO...	Maryland	gLen Burnie	21060	1232092527

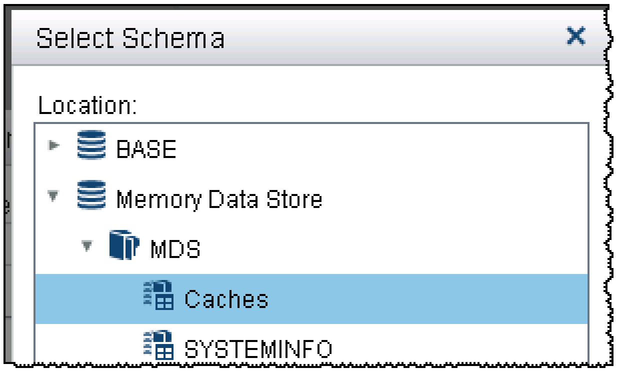
Note: There are data inconsistencies that require data cleansing. Those inconsistencies are addressed in the next section.

12. Click **Company_Order_Info**.
13. Click the **Summary** tab.

14. Click **Cache FedSQL View**.

   a. Click the ellipsis button.

      1) Navigate to **Memory Data Store** ⇨ **MDS** ⇨ **Caches**.



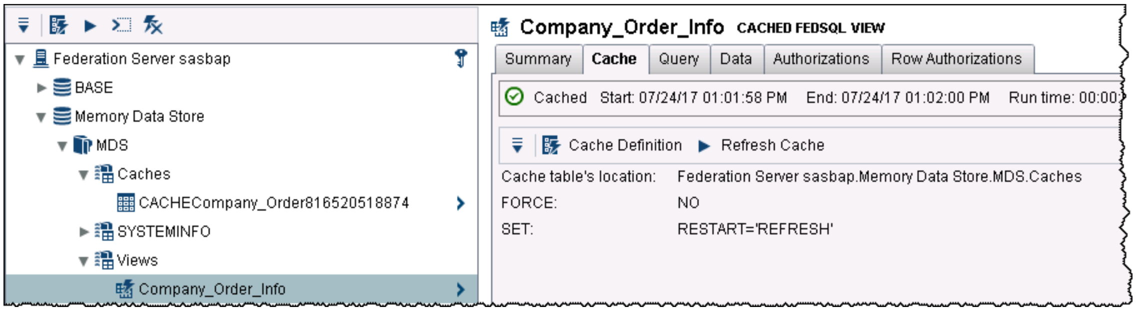
      2) Click **OK**.

   b. Click **Create Cache Table**.

   c. Click **New** on the scheduled refreshes at the bottom of the Cache tab.

   d. Click **OK** on the schedule refresh to take the defaults.

15. Expand **Memory Data Store** ⇨ **MDS** ⇨ **Caches** to see the newly created cache.



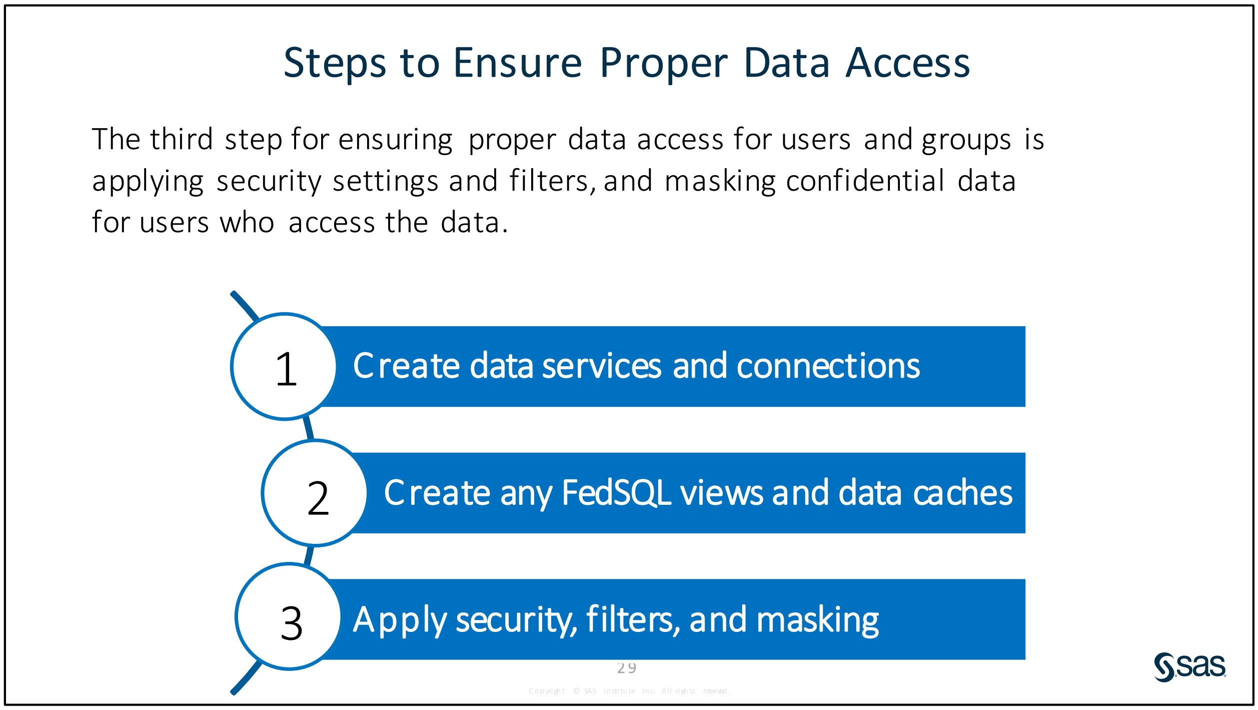
Cached Start: 07/24/17 01:01:58 PM End: 07/24/17 01:02:00 PM Run time: 00:00

Cache table's location: Federation Server sasbap.Memory Data Store.MDS.Caches
FORCE: NO
SET: RESTART='REFRESH'

Note: Caches should ***never*** be accessed directly. Always use the view to access the cached data because the cache is refreshed in a new cache, and the old cache is discarded.

End of Demonstration

1.4 Applying Security, Filters, and Masking



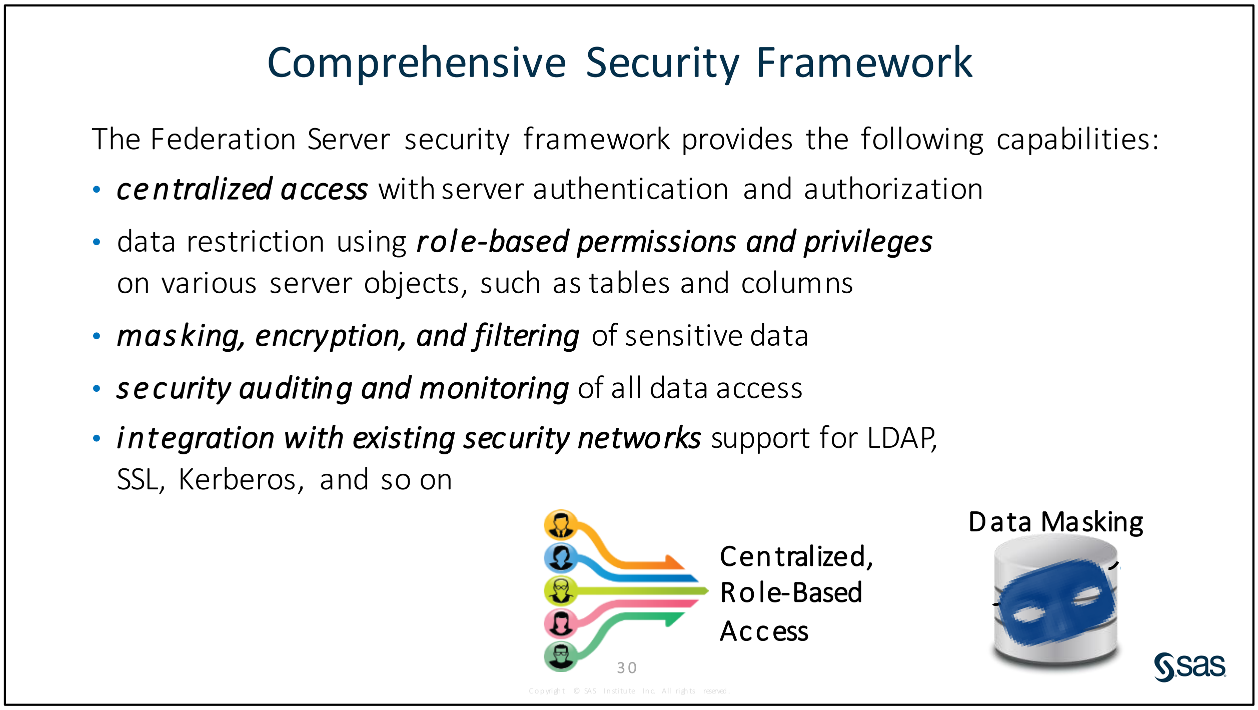
Steps to Ensure Proper Data Access

The third step for ensuring proper data access for users and groups is applying security settings and filters, and masking confidential data for users who access the data.

1 Create data services and connections

2 Create any FedSQL views and data caches

3 Apply security, filters, and masking



Comprehensive Security Framework

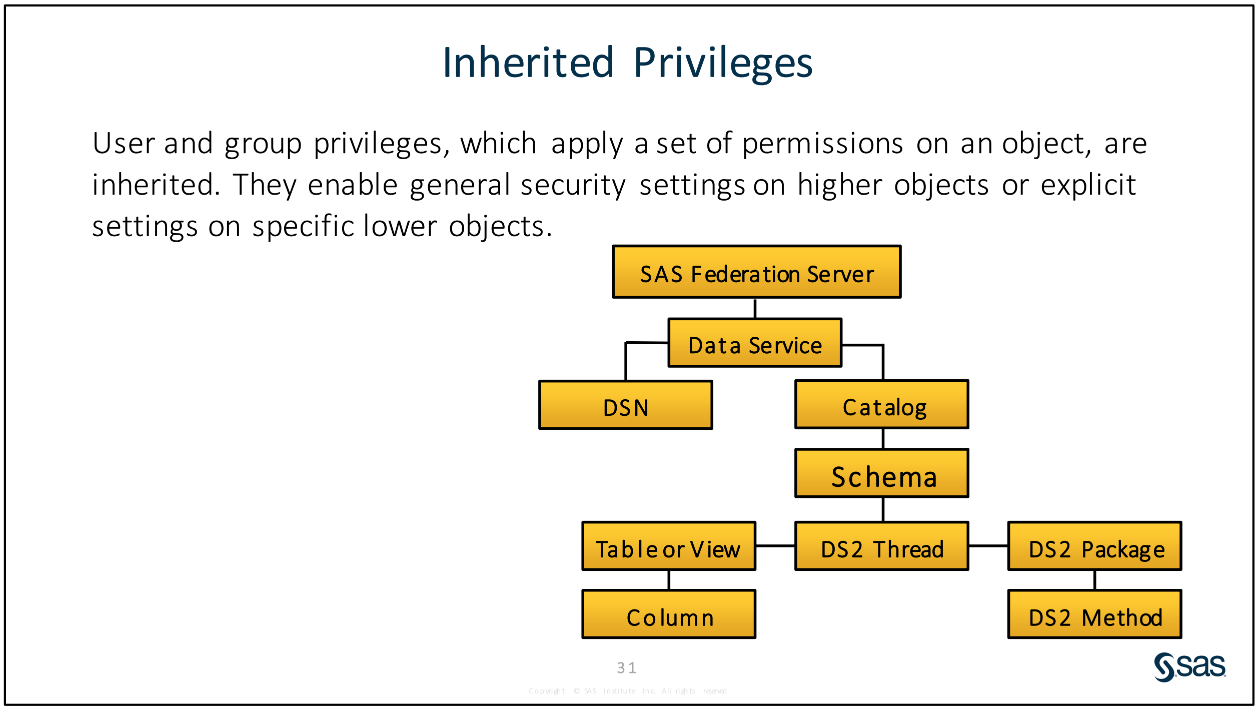
The Federation Server security framework provides the following capabilities:

- *centralized access* with server authentication and authorization
- data restriction using *role-based permissions and privileges* on various server objects, such as tables and columns
- *masking, encryption, and filtering* of sensitive data
- *security auditing and monitoring* of all data access
- *integration with existing security networks* support for LDAP, SSL, Kerberos, and so on

The *Lightweight Directory Access Protocol (LDAP)* is a directory service protocol that provides a mechanism used to connect to, search, and modify Internet directories.

SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. It ensures that all data passed between the web server and browsers remains private and integral.

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography.



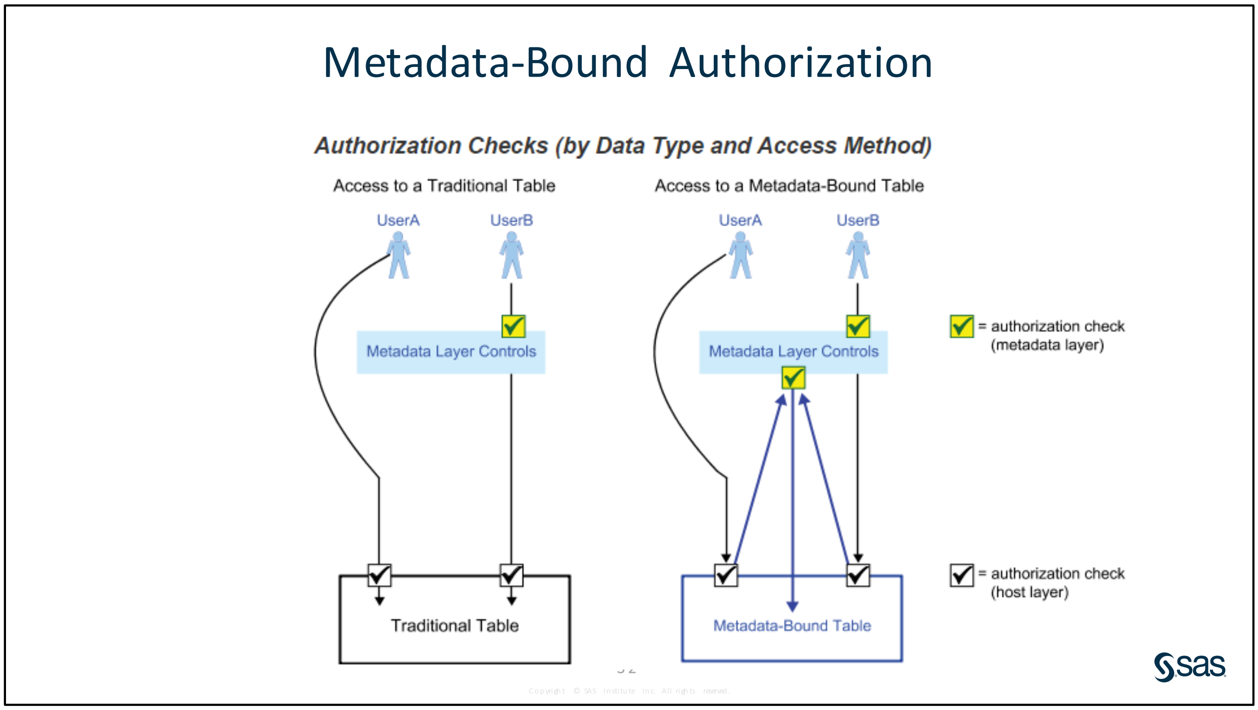
Inherited Privileges

User and group privileges, which apply a set of permissions on an object, are inherited. They enable general security settings on higher objects or explicit settings on specific lower objects.

SAS Federation Server
Data Service
DSN
Catalog
Schema
Table or View
DS2 Thread
DS2 Package
Column
DS2 Method

31

Whereas privileges on the server are inherited by the data service, privileges on the data service are inherited by the DSN and catalog. Privileges on the catalog are inherited by the schema, table (view), and column. This inheritance hierarchy enables an administrator to set general security rules on higher level objects and then set exceptions on only the more specific (subordinate) objects.



Metadata-Bound Authorization

Authorization Checks (by Data Type and Access Method)

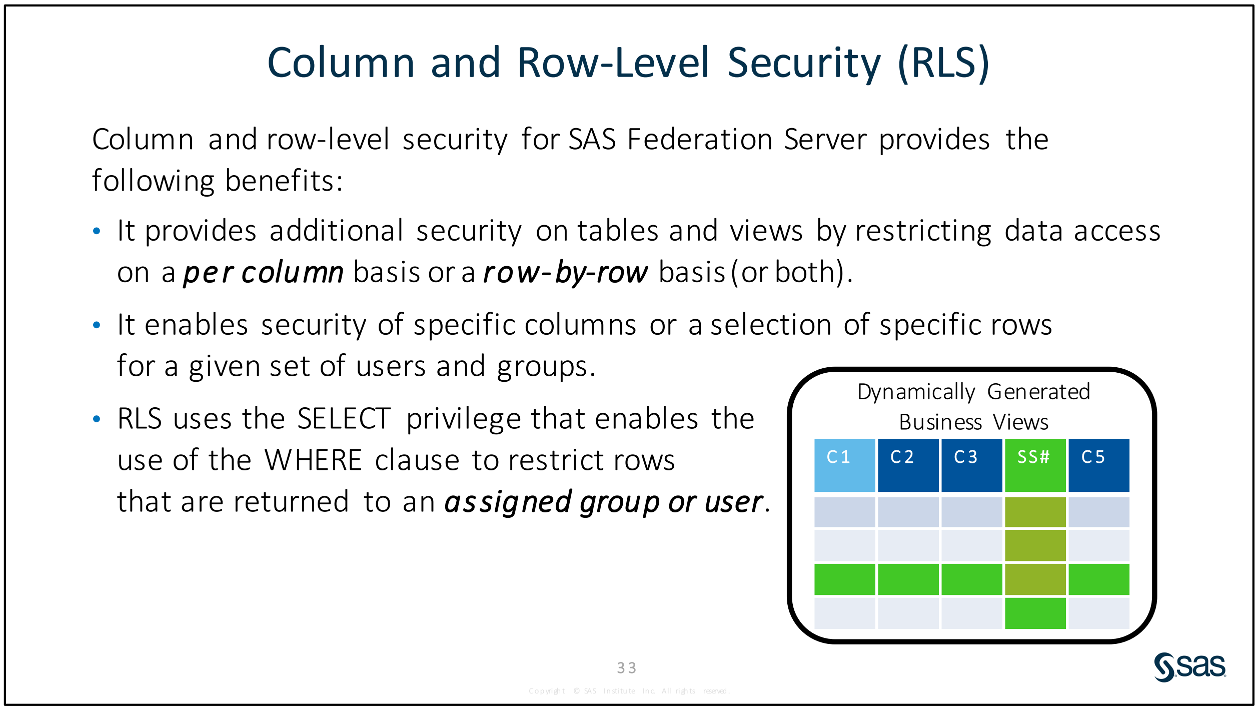
Access to a Traditional Table

Access to a Metadata-Bound Table

The picture above is a review of the metadata-bound library authorization process comparing traditional SAS data set access to metadata-bound library access.

Here are the key differences:

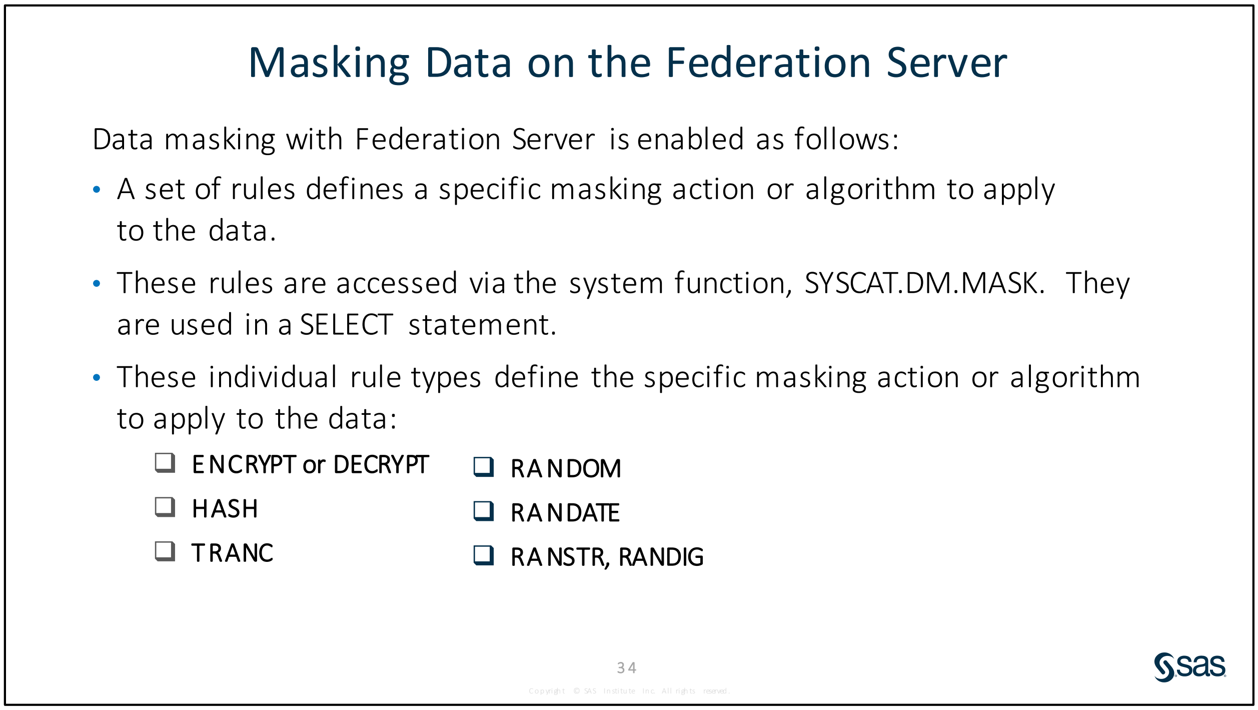
- When accessing a traditional table, a user can bypass metadata-layer controls by making a direct request.
- When accessing a metadata-bound table, a user cannot completely bypass metadata-layer controls. Even on a direct request, UserA is always subject to a metadata-layer permissions check before accessing SAS data from SAS.



Column and Row-Level Security (RLS)

Column and row-level security for SAS Federation Server provides the following benefits:

- It provides additional security on tables and views by restricting data access on a ***per column*** basis or a ***row-by-row*** basis (or both).
- It enables security of specific columns or a selection of specific rows for a given set of users and groups.
- RLS uses the SELECT privilege that enables the use of the WHERE clause to restrict rows that are returned to an ***assigned group or user***.



Masking Data on the Federation Server

Data masking with Federation Server is enabled as follows:

- A set of rules defines a specific masking action or algorithm to apply to the data.
- These rules are accessed via the system function, SYSCAT.DM.MASK. They are used in a SELECT statement.
- These individual rule types define the specific masking action or algorithm to apply to the data:

  - ❑ ENCRYPT or DECRYPT
  - ❑ HASH
  - ❑ TRANC
  - ❑ RANDOM
  - ❑ RANDATE
  - ❑ RANSTR, RANDIG

Data masking enables you to hide PII data while still retaining the critical characteristics and meaning of the data needed for analytics.

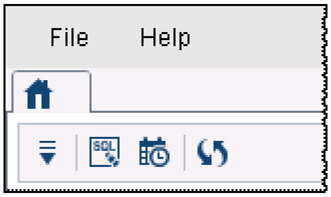
Encryption is the process of converting data into an alternate form that prevents it from easily being understood except by authorized users. Typically, encrypted data can be decrypted with the appropriate key used to encrypt the data.

Hash represents the process of using a hashing algorithm to compute a unique, fixed-length hash value based on the original data value. The original data value cannot be recovered when using hashing.

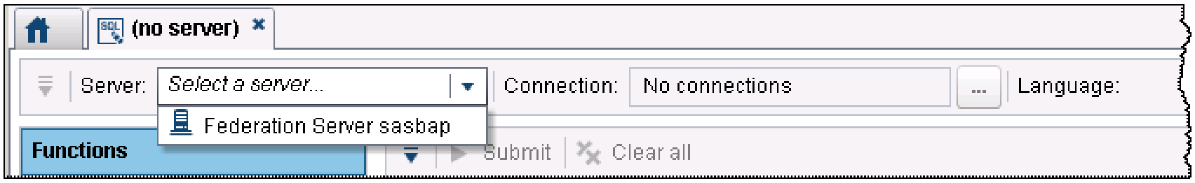
Defining Permissions and Privileges and Applying Masking and Filtering on Federated Views.

This demonstration illustrates how to limit user data access using table permissions and privileges. It also illustrates how to apply data concealment with masking and row-level filtering to conceal personally identifiable information (PII).

1. Click (**Console**) to open the SQL Console Editor.



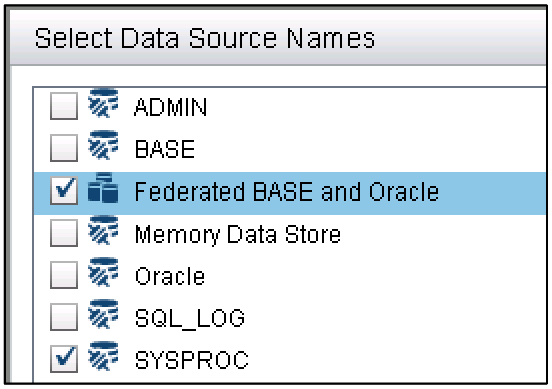
2. Select **Federation Server sasbap** from the Server menu.



3. Click the **Connection** ellipsis button.

   a. Clear the **ADMIN** selection.

   b. Select **Federated BASE and Oracle** and **SYSPROC**.



Select Data Source Names

- □ ADMIN
- □ BASE
- ☑ Federated BASE and Oracle
- □ Memory Data Store
- □ Oracle
- □ SQL_LOG
- ☑ SYSPROC

Note: **SYSPROC** contains some data quality packages and methods to cleanse data.

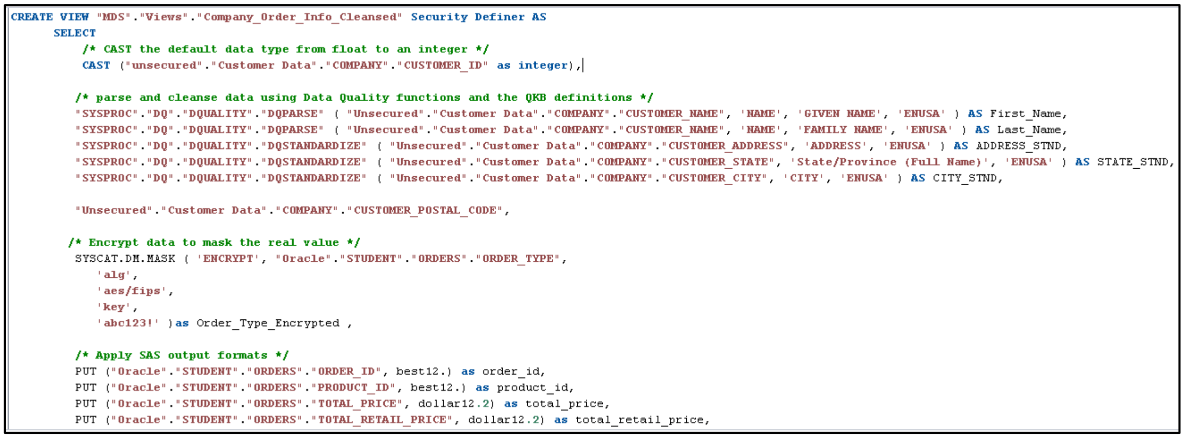
   c. Click **OK**.

4. Select **Import** on the SQL Editor menu.

5. Select **ateved_masking_quality.txt**.

6. Click **Open**.

7. Click **OK** and review the code.

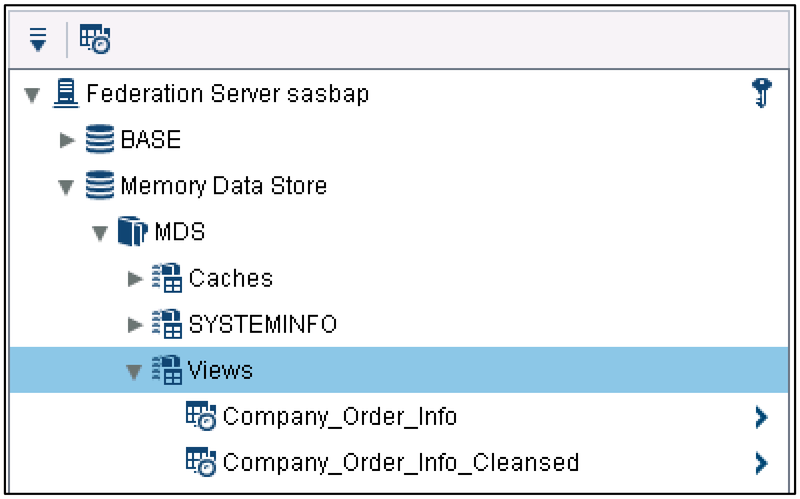


```
CREATE VIEW "MDS"."Views"."Company_Order_Info_Cleansed" Security Definer AS
      SELECT
          /* CAST the default data type from float to an integer */
          CAST ("unsecured"."Customer Data"."COMPANY"."CUSTOMER_ID" as integer),

         /* parse and cleanse data using Data Quality functions and the QKB definitions */
         "SYSPROC"."DQ"."DQUALITY"."DQPARSE" ( "Unsecured"."Customer Data"."COMPANY"."CUSTOMER_NAME", 'NAME', 'GIVEN NAME', 'ENUSA' ) AS First_Name,
         "SYSPROC"."DQ"."DQUALITY"."DQPARSE" ( "Unsecured"."Customer Data"."COMPANY"."CUSTOMER_NAME", 'NAME', 'FAMILY NAME', 'ENUSA' ) AS Last_Name,
         "SYSPROC"."DQ"."DQUALITY"."DQSTANDARDIZE" ( "Unsecured"."Customer Data"."COMPANY"."CUSTOMER_ADDRESS", 'ADDRESS', 'ENUSA' ) AS ADDRESS_STND,
         "SYSPROC"."DQ"."DQUALITY"."DQSTANDARDIZE" ( "Unsecured"."Customer Data"."COMPANY"."CUSTOMER_STATE", 'State/Province (Full Name)', 'ENUSA' ) AS STATE_STND,
         "SYSPROC"."DQ"."DQUALITY"."DQSTANDARDIZE" ( "Unsecured"."Customer Data"."COMPANY"."CUSTOMER_CITY", 'CITY', 'ENUSA' ) AS CITY_STND,

         "Unsecured"."Customer Data"."COMPANY"."CUSTOMER_POSTAL_CODE",

        /* Encrypt data to mask the real value */
         SYSCAT.DM.MASK ( 'ENCRYPT', "Oracle"."STUDENT"."ORDERS"."ORDER_TYPE",
            'alg',
            'aes/fips',
            'key',
            'abc123!' )as Order_Type_Encrypted ,

         /* Apply SAS output formats */
         PUT ("Oracle"."STUDENT"."ORDERS"."ORDER_ID", best12.) as order_id,
         PUT ("Oracle"."STUDENT"."ORDERS"."PRODUCT_ID", best12.) as product_id,
         PUT ("Oracle"."STUDENT"."ORDERS"."TOTAL_PRICE", dollar12.2) as total_price,
         PUT ("Oracle"."STUDENT"."ORDERS"."TOTAL_RETAIL_PRICE", dollar12.2) as total_retail_price,
```

8. Click **Submit**.

9. Click the **Home** tab.

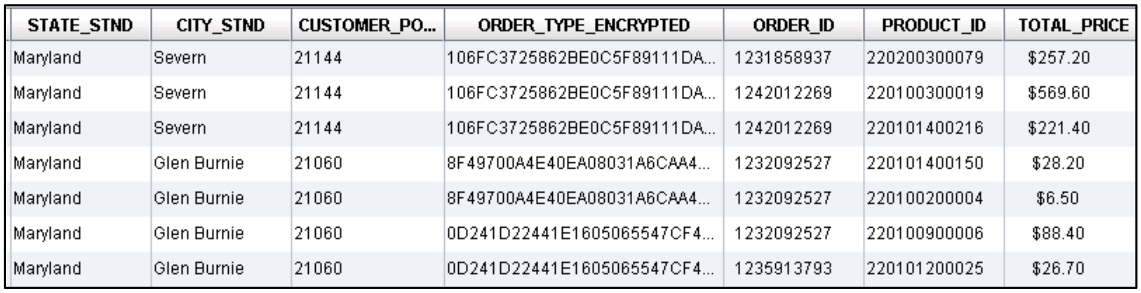
10. Navigate to **Memory Data Store** ⇨ **MDS** ⇨ **Views**.



11. Select **Company_Order_Info_Cleansed**.

12. Click the **Data** tab.

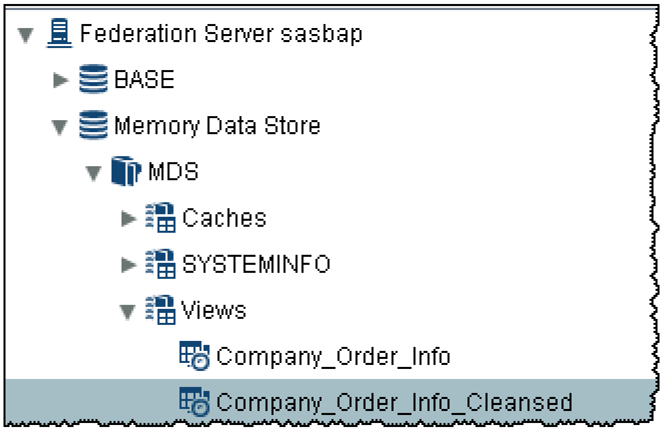
13. Click **View Data**.



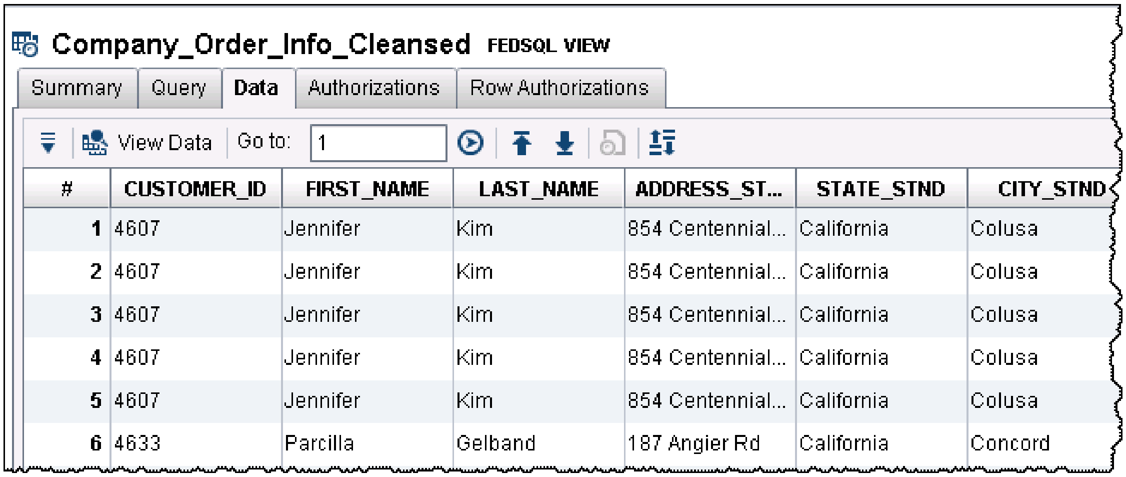
STATE_STND	CITY_STND	CUSTOMER_PO...	ORDER_TYPE_ENCRYPTED	ORDER_ID	PRODUCT_ID	TOTAL_PRICE
Maryland	Severn	21144	106FC3725862BE0C5F89111DA...	1231858937	220200300079	$257.20
Maryland	Severn	21144	106FC3725862BE0C5F89111DA...	1242012269	220100300019	$569.60
Maryland	Severn	21144	106FC3725862BE0C5F89111DA...	1242012269	220101400216	$221.40
Maryland	Glen Burnie	21060	8F49700A4E40EA08031A6CAA4...	1232092527	220101400150	$28.20
Maryland	Glen Burnie	21060	8F49700A4E40EA08031A6CAA4...	1232092527	220100200004	$6.50
Maryland	Glen Burnie	21060	0D241D22441E1605065547CF4...	1232092527	220100900006	$88.40
Maryland	Glen Burnie	21060	0D241D22441E1605065547CF4...	1235913793	220101200025	$26.70

Note: The data has been cleansed, encrypted, and output using SAS formats

14. Click the **Row Authorizations** tab.
    a. Click **New Filter**
    b. Click the **SQL Clause builder** icon.
        1) Select **STATE_STND** in the Column menu.
        2) Select **= Equals** in the Operator menu.
        3) Enter **California** in the **Value** field.
        4) Click **OK**.
    c. Click the **Users and Groups** tab.
        1) Click the plus sign.
        2) Select **Anita**.
        3) Click **OK**.
15. Click **OK** to save the filter.
16. Select **Federation Server sasbap**.
    a. Click the **Authorizations** tab.
    b. Add a plus sign.
    c. Select **Anita** to add her to the Authorization list.
    d. Click **Close**
    e. Click the **Summary** tab.
    f. Click **Sign out**.
    g. Click **Sign in**.
    h. Enter **Anita** in the **User ID** field.
    i. Enter **Student1** in the **Password** field.
17. Expand **Memory Data Store** ⇨ **MDS** ⇨ **Views**.
    a. Select **Company_Order_Info_Cleansed**.



    b. Click the **Query** tab. Note that Anita cannot see the query due to lack of permission.
    c. Click the **Data** tab.
    d. Click **View Data**.

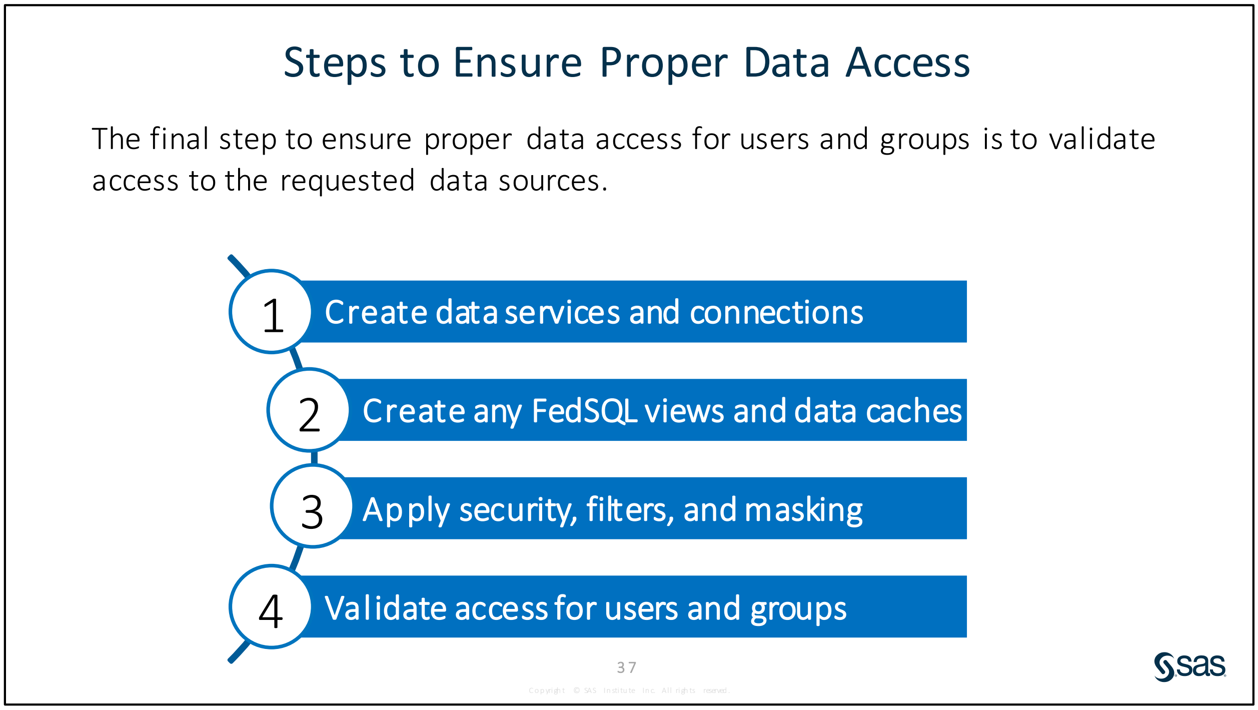


#	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_ST...	STATE_STND	CITY_STND
1	4607	Jennifer	Kim	854 Centennial...	California	Colusa
2	4607	Jennifer	Kim	854 Centennial...	California	Colusa
3	4607	Jennifer	Kim	854 Centennial...	California	Colusa
4	4607	Jennifer	Kim	854 Centennial...	California	Colusa
5	4607	Jennifer	Kim	854 Centennial...	California	Colusa
6	4633	Parcilla	Gelband	187 Angier Rd	California	Concord

Note: Only data where **STATE_STND** equals *California* is shown to Anita due to the row filter.

End of Demonstration

1.5 Accessing Virtualized Data



Steps to Ensure Proper Data Access

The final step to ensure proper data access for users and groups is to validate access to the requested data sources.

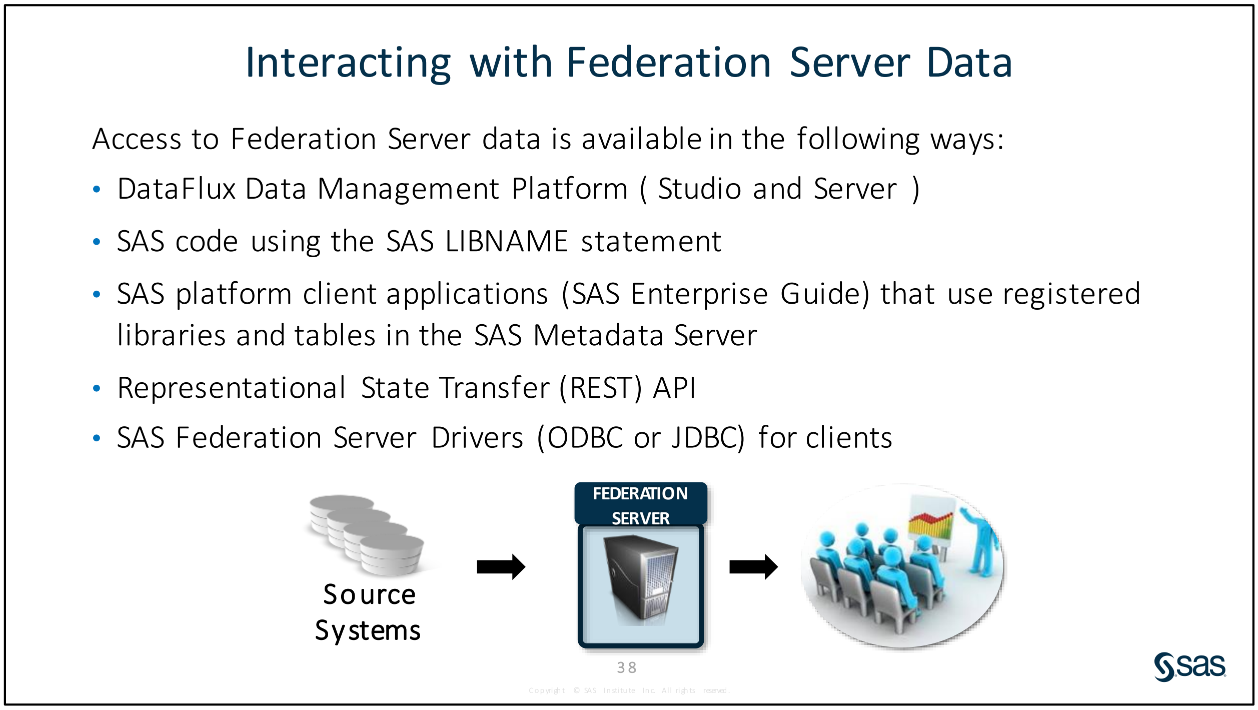
1 Create data services and connections

2 Create any FedSQL views and data caches

3 Apply security, filters, and masking

4 Validate access for users and groups

37



Interacting with Federation Server Data

Access to Federation Server data is available in the following ways:

- DataFlux Data Management Platform (Studio and Server)
- SAS code using the SAS LIBNAME statement
- SAS platform client applications (SAS Enterprise Guide) that use registered libraries and tables in the SAS Metadata Server
- Representational State Transfer (REST) API
- SAS Federation Server Drivers (ODBC or JDBC) for clients

Source Systems

FEDERATION SERVER

38

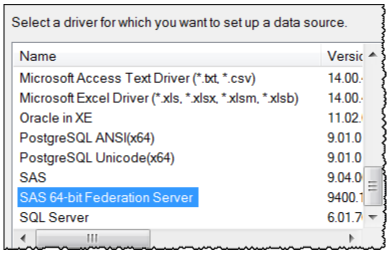
There are numerous options for accessing federated data on SAS Federation Server, providing you great flexibility in making the data available to business processes and end users.

These options include the following:

- connecting SAS Data Management Studio to the SAS Metadata Server where SAS Federation Server was configured during installation
- directly accessing federated data in SAS code using a LIBNAME statement
- registering Federation Server tables and views in SAS metadata for SAS platform applications such as SAS Visual Analytics and SAS Enterprise Guide
- accessing federated data using the REST interface in web services, where the responses can be requested in JSON or XML

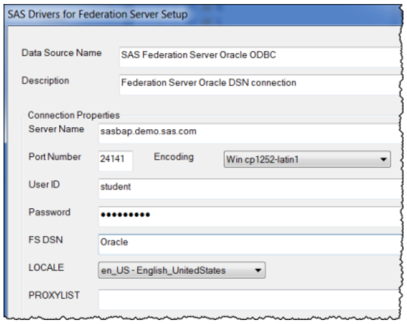
SAS Federation Server Client Drivers

The SAS Federation Server client drivers provide standard JDBC and ODBC interfaces to enable third-party applications to interact with SAS Federation Server.



Select a driver for which you want to set up a data source.

Name	Versic
Microsoft Access Text Driver (*.txt, *.csv)	14.00.
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	14.00.
Oracle in XE	11.02.
PostgreSQL ANSI(x64)	9.01.0
PostgreSQL Unicode(x64)	9.01.0
SAS	9.04.0
SAS 64-bit Federation Server	9400.1
SQL Server	6.01.7



SAS Drivers for Federation Server Setup

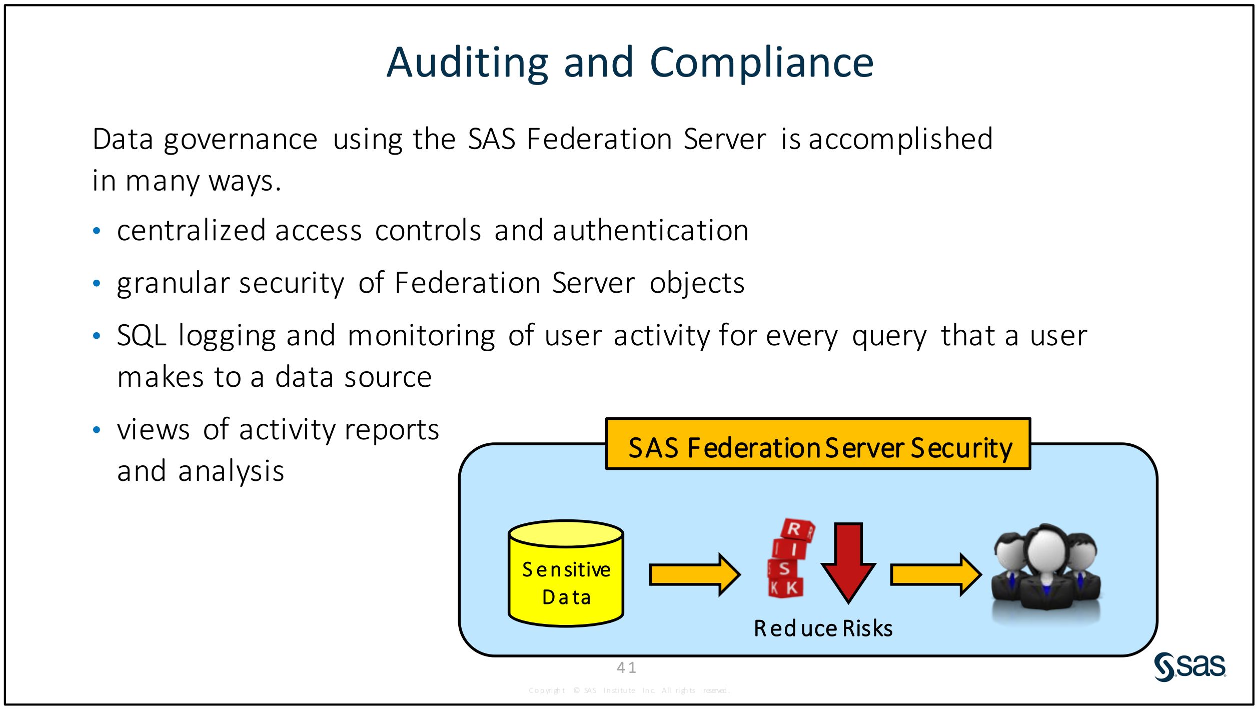
Data Source Name: SAS Federation Server Oracle ODBC
Description: Federation Server Oracle DSN connection

Connection Properties
Server Name: sasbap.demo.sas.com
Port Number: 24141 Encoding: Win cp1252-latin1
User ID: student
Password: ••••••••••
FS DSN: Oracle
LOCALE: en_US - English_UnitedStates
PROXYLIST:

```
jdbc:sastkts://sasbap.demo.sas.com:24141?constring=(DSN=Oracle)
              &userName=student
              &password=Metadata0
```

39

1.6 Auditing and Logging



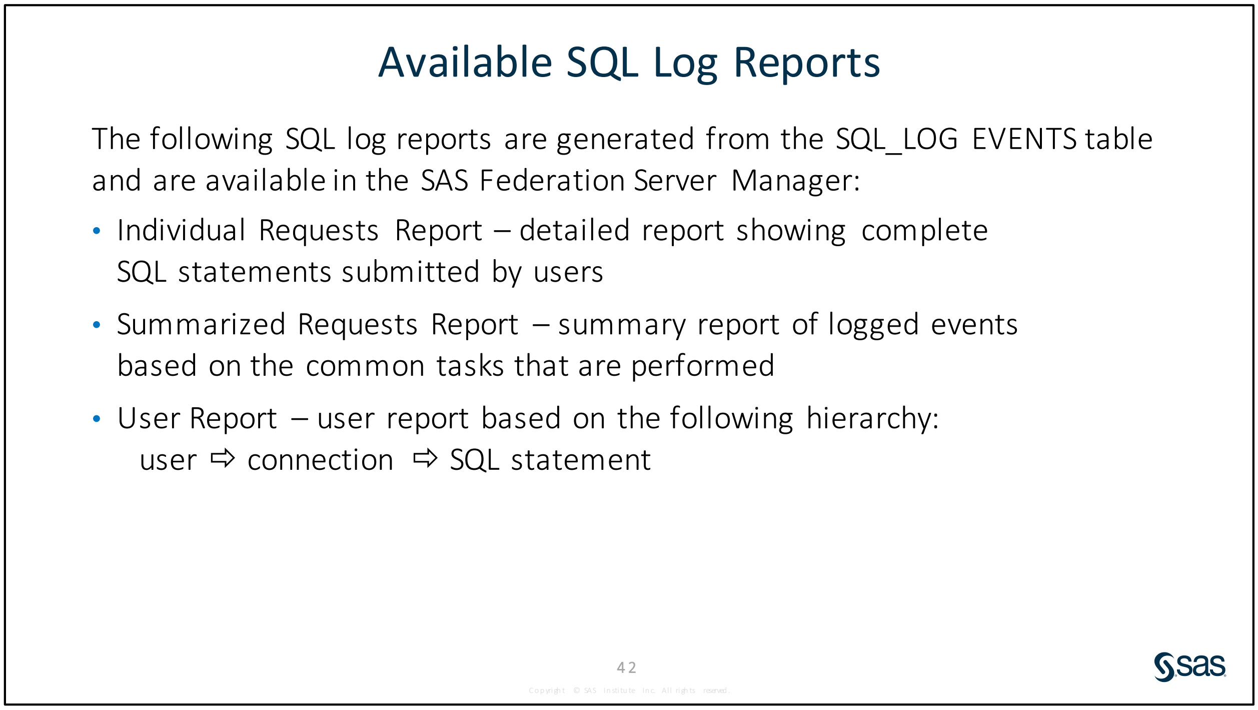
Auditing and Compliance

Data governance using the SAS Federation Server is accomplished in many ways.

- centralized access controls and authentication
- granular security of Federation Server objects
- SQL logging and monitoring of user activity for every query that a user makes to a data source
- views of activity reports and analysis

SAS Federation Server Security
Sensitive Data
RISK
Reduce Risks

41



Available SQL Log Reports

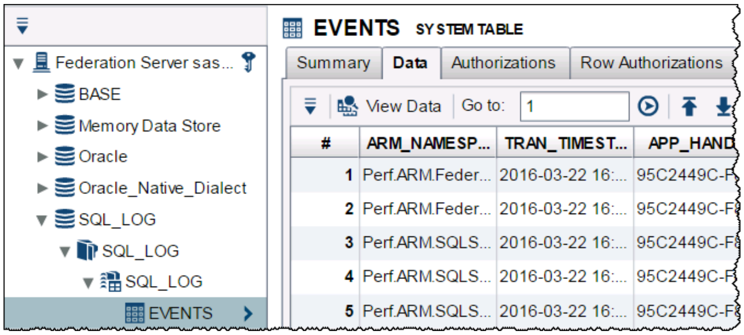
The following SQL log reports are generated from the SQL_LOG EVENTS table and are available in the SAS Federation Server Manager:

- Individual Requests Report – detailed report showing complete SQL statements submitted by users
- Summarized Requests Report – summary report of logged events based on the common tasks that are performed
- User Report – user report based on the following hierarchy:
  user ⇨ connection ⇨ SQL statement

42

Managing the SQL Log Database

All SQL logging is written to the EVENTS table in the SQL_LOG database.



- Federation Server sas...
  - BASE
  - Memory Data Store
  - Oracle
  - Oracle_Native_Dialect
  - SQL_LOG
    - SQL_LOG
      - SQL_LOG
        - EVENTS

EVENTS SYSTEM TABLE

Summary | Data | Authorizations | Row Authorizations

View Data Go to: 1

#	ARM_NAMESP...	TRAN_TIMEST...	APP_HAND
1	Perf.ARM.Feder...	2016-03-22 16:...	95C2449C-F
2	Perf.ARM.Feder...	2016-03-22 16:...	95C2449C-F
3	Perf.ARM.SQLS...	2016-03-22 16:...	95C2449C-F
4	Perf.ARM.SQLS...	2016-03-22 16:...	95C2449C-F
5	Perf.ARM.SQLS...	2016-03-22 16:...	95C2449C-F

Note: Never change the name of the EVENTS table.

As the EVENTS table grows in size, you can move data to another table for archive purposes. This is accomplished by creating a federated DSN to the SQL_LOG DSN and another data source to use for storing the archived data.

- Use the following statement to move data to a new table:
  CREATE TABLE .. AS SELECT * FROM SQL_LOG.SQL_LOG.EVENTS WHERE
- To determine what records are needed, use the WHERE condition with dates:
  WHERE TRAN_TIMESTAMP > TIMESTAMP '2012-01-25 01:00:00'
- To move data to an existing table, use the INSERT statement:
  INSERT INTO .. SELECT * FROM SQL_LOG.SQL_LOG.EVENTS WHERE
- Use the DELETE statement to remove outdated records that are no longer needed or have been archived.

  Here is an example:

  DELETE FROM SQL_LOG.SQL_LOG.EVENTS WHERE TRAN_TIMESTAMP < TIMESTAMP ‘2011-04-11 12:00:00’

Additional Resources

SAS Federation Server Product Info:
https://www.sas.com/en_us/software/federation-server.html

SAS Federation Server Training:
https://support.sas.com/edu/schedules.html?ctry=us&crs=DIFD

44