Add SQL Server data store

Updated July 17, 2020

Once you have established communications with a SecureAuth Connector in the SecureAuth® Identity Platform, you can add data stores to assert or manage user identity information.

> To migrate from SecureAuth IdP (on-prem) version 9.3 with an existing SQL Server data store configured using the New Experience UI to the Identity Platform (cloud), you will need to reenter the data store credentials after downloading and installing the SecureAuth Connector.

Prerequisites

- Identity Platform version 19.07 or later
- SecureAuth Connector installed and connected for Identity Platform cloud deployments
- SQL Server data store

> Make sure you have the correct SQL tables and stored procedures configured, specific to the Identity Platform version.
>
> - For Identity Platform versions 9.1 to 19.07.01-10 (hotfix 10), see SQL tables and stored procedures configuration (9.1 to 19.07.10-10)
> - For Identity Platform versions 19.07.01-11 (hotfix 11) and later, see SQL tables and stored procedures configuration (19.07.01-11+)

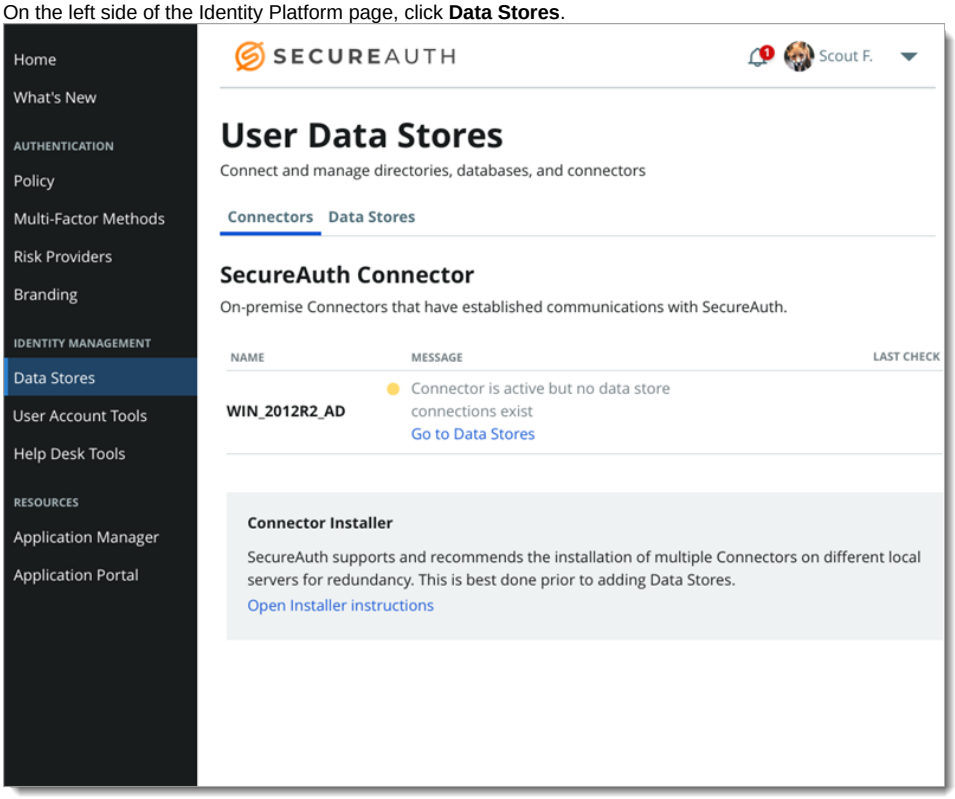
Process

There are two parts to adding a data store in the Identity Platform — (1) adding the data store and (2) mapping the data store properties.

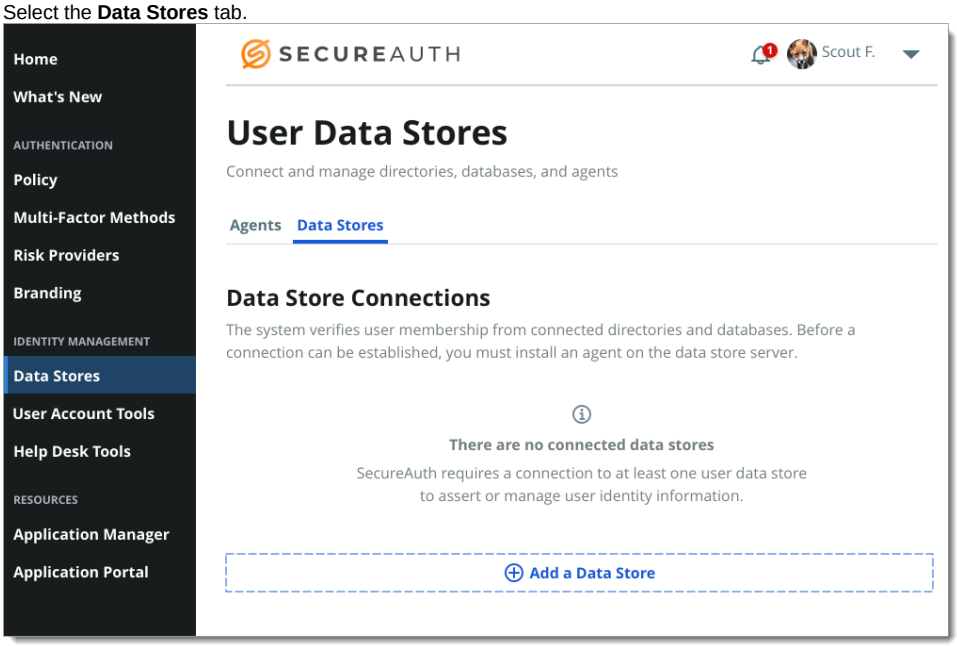
Step 1 of 2: Add a SQL Server data store

The first part of adding a SQL Server data store is configuring the data store name, connections, credentials, and search attributes.

1. On the left side of the Identity Platform page, click **Data Stores**.

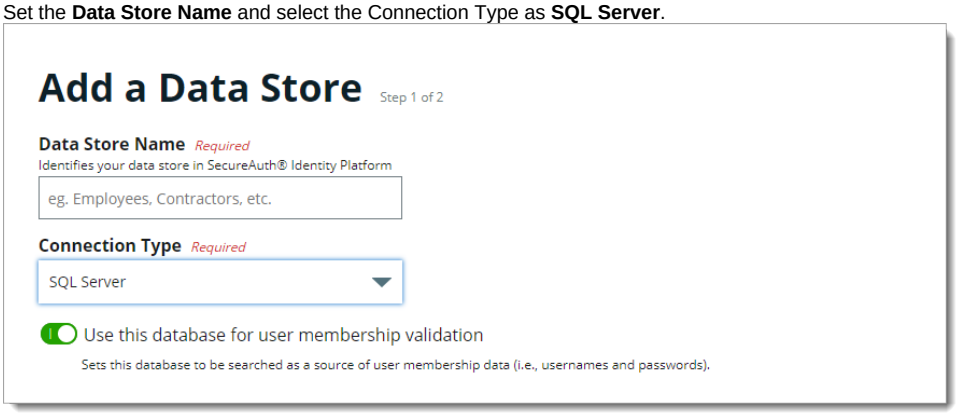


2. Select the **Data Stores** tab.



3. Click **Add a Data Store**.

4. Set the **Data Store Name** and select the Connection Type as **SQL Server**.



5. For the **Use this database for user membership validation** slider, use one of the following options:

On	Enable membership validation; use the database to search for the user's membership in a user group. This means the database is a Membership Store, containing the password to validate with the username.
Off	Disable membership validation; use the database to search only for the user profile information. This means the database is only used to find the username and profile information (such as phone number, email address, device recognition profiles, OATH tokens, and so on).

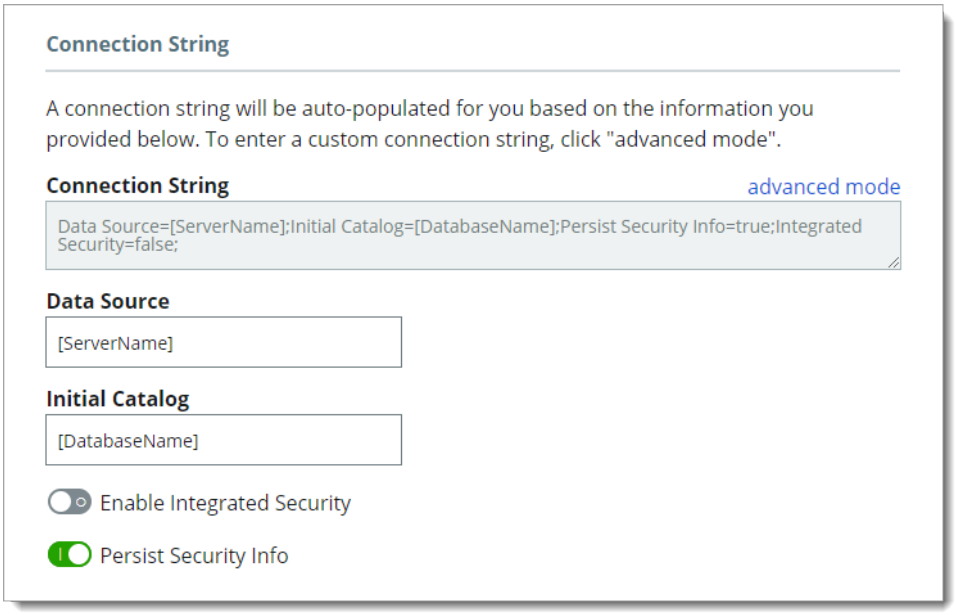
After the data store is saved, this field is the **Membership Store** label shown on the View Summary.

A common use case for a Membership Store would be to have a database with username and password information (and maybe some profile information), and then have a second database used to store and access data that the Identity Platform writes to the database (such as device recognition, device enrollment, push notification tokens, and so on).

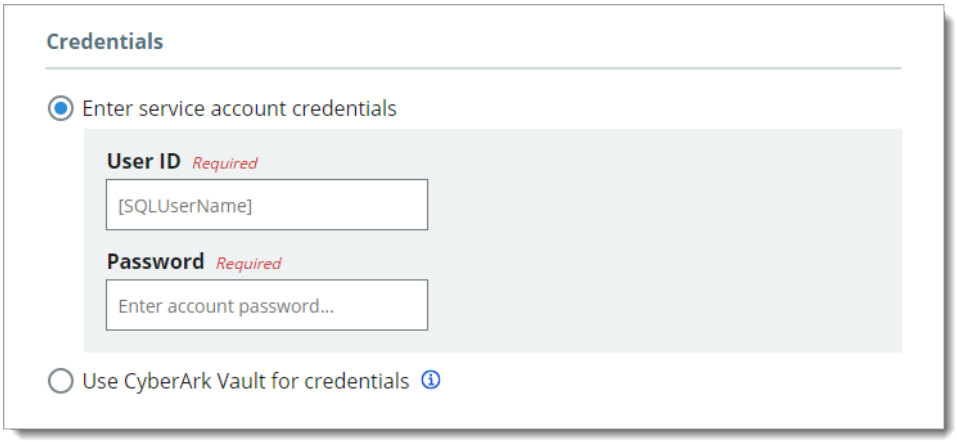
6. In the **Connection String** section, set the connection string to the SQL Server.

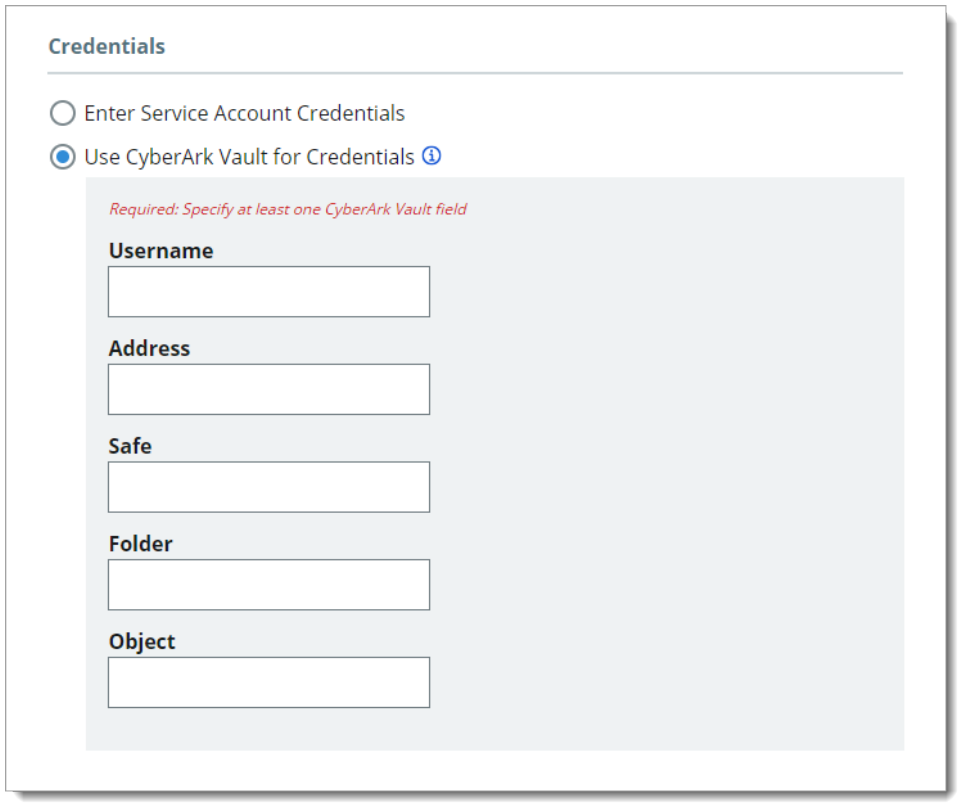
> For information about content in the corporate connection string, see the external article All SQL Server SqlConnection Properties.

Connection String	The value in this field is the connection string auto-populated by the Data Source and Initial Catalog fields (unless a custom connection string is manually entered).
advanced mode	To manually enter the connection string, click the **advanced mode** link.
Data Source	Name or network address of the SQL Server instance to which to connect. For example, 111.22.33.444\sqlserver
Initial Catalog	The initial catalog name (or database name). For example, secureauthconsult
Enable Integrated Security	Move the slider to indicate whether to enable integrated security for a secure connection.
Persist Security Info	Move the slider to indicate whether to persist security information such as the password in the connection string.



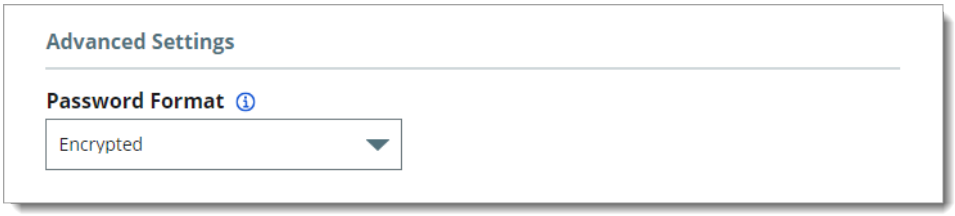
7. In the **Credentials** section, provide the log in credentials to access the SQL data store.

Enter Service Account Credentials	With this option, enter the following fields: • **User ID** – SQL user ID email address for the service account login • **Password** – Password for the service account login 
Use CyberArk Vault for Credentials	With this option, enter at least one field for the service account login: • **Username** – User name of machine to be scanned by CyberArk Application Identity Manger (AIM). This information appears on the Account Details page of the CyberArk Password Vault Web Access (PVWA) Admin Console. • **Address** – Address of machine to be scanned by AIM. • **Safe** – Name of Access Control Safe where credentials are stored. • **Folder** – Name of folder where account resides (by default, it its the root folder). • **Object** – Unique identifier Object name for the account.



8. In the **Advanced Settings** section, define how the service account password is to be stored in the directory.

Password Format	Choose one of the following formats: • **Clear** – Password is stored as plain text. This improves performance of storage and retrieval but is less secure. • **Encrypted** – Password is stored as encrypted and can be decrypted for password comparison or retrieval. This is more secure, but requires additional processing or storage. • **Hashed** – Password is hashed using a one-way hash algorithm and random salt-value. When password is validated, it is hashed with the salt value of the dates for verification. Hashed passwords cannot be retrieved.

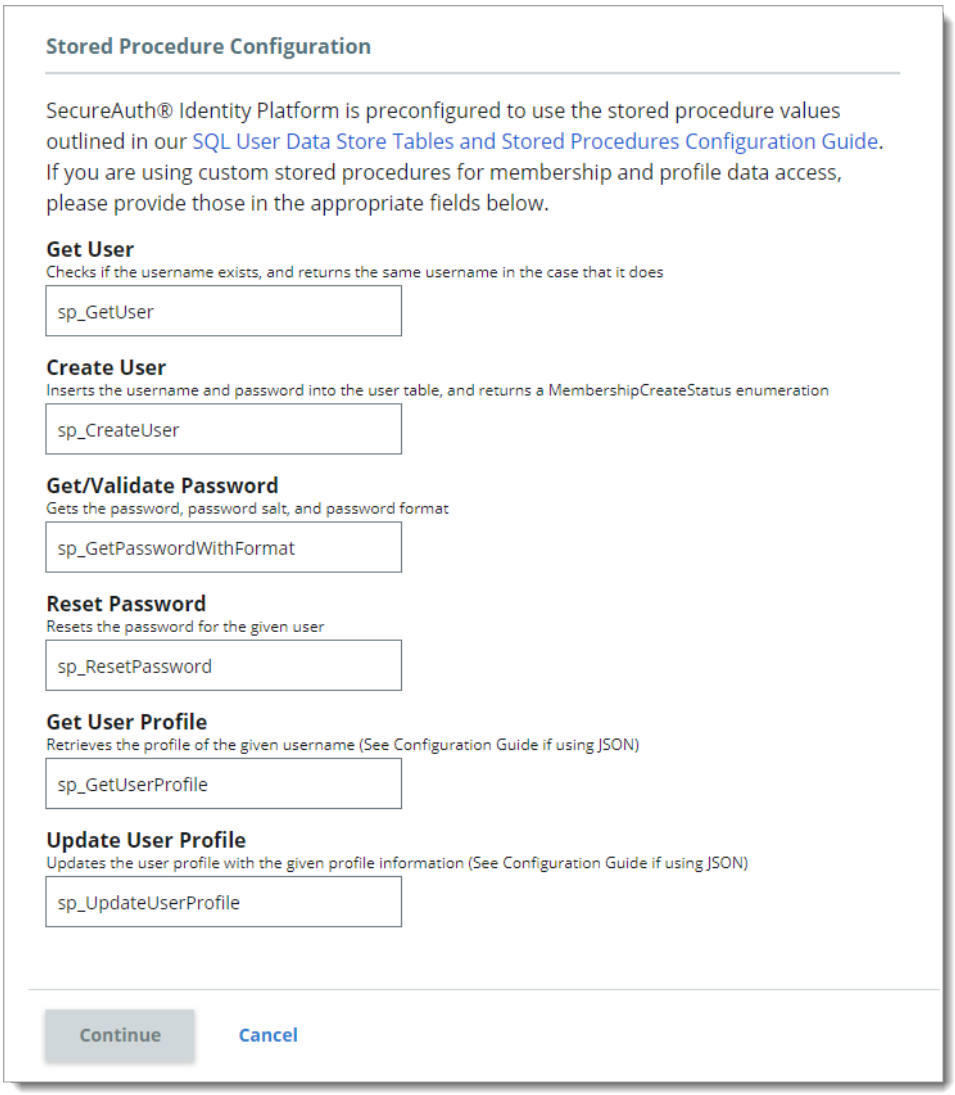


9. In the **Stored Procedure Configuration** section, use the default values unless custom stored procedures are used for membership and profile data access.
The Identity Platform is preconfigured to use the stored procedure values you have configured, specific to your version of the Identity Platform. See the appropriate topic:

   - For Identity Platform versions 9.1 to 19.07.01-10 (hotfix 10), see SQL tables and stored procedures configuration (9.1 to 19.07.10-10)
   - For Identity Platform versions 19.07.01-11 (hotfix 11) and later, see SQL tables and stored procedures configuration (19.07.01-11+)

Get User	Checks if a username exists, and returns the same username in the case that it does.
Create User	Inserts the username and password into the user table, and returns a MembershipCreateStatus enumeration.
Get/Validate Password	Gets the password, password salt, and password format.
Reset Password	Resets the password for the given user.
Get User Profile	Retrieves the profile of the given username (See Configuration Guide if using JSON).

Update User Profile	Updates user profile with the given profile information (See Configuration Guide if using JSON).



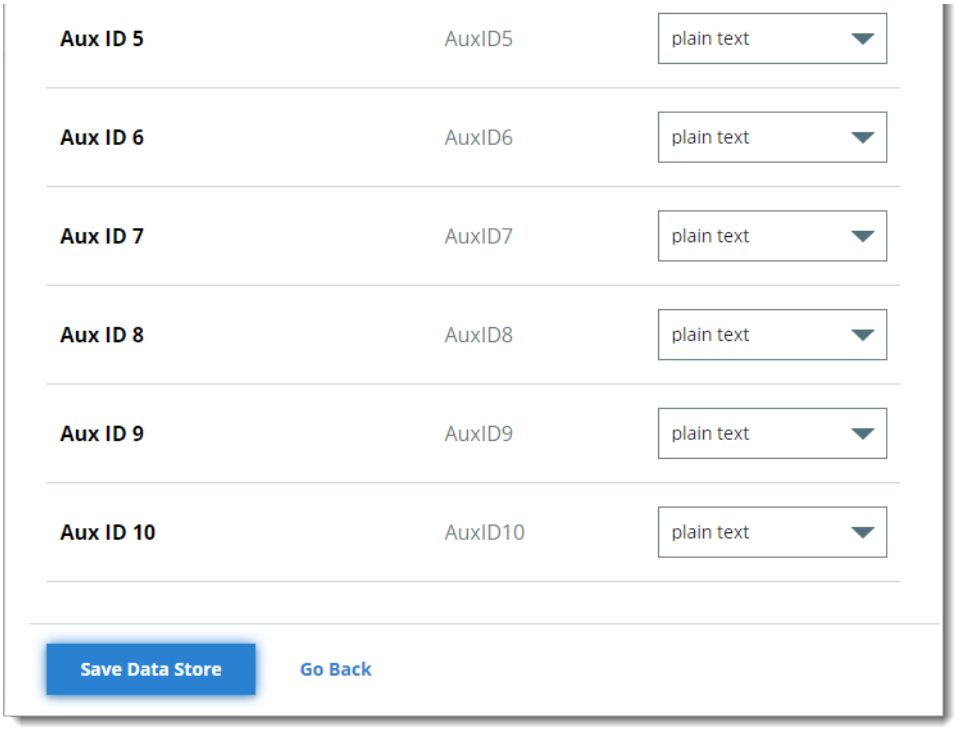
10. Click **Continue**.
    The Map Data Store Properties page opens.

Map Data Store Properties

Step 2 of 2

Properties are required user data that need to be read from, or stored in, your directories or databases. Below is the list of stored procedures associated with each property. Click the edit icon to change the data format. Please refer to our SQL User Data Store Tables and Stored Procedures Configuration Guide for details on how to properly configure your SQL database.

NAME	SQL FIELD	DATA FORMAT
First Name	FirstName	plain text
Last Name	LastName	plain text
Groups	GroupList	plain text
Phone 1 (Work)	Phone1	plain text
Phone 2 (Mobile)	Phone2	plain text
Phone 3 (Alternate)	Phone3	plain text
Phone 4 (Alternate)	Phone4	plain text
Email 1 (Work)	Email1	plain text
Email 2 (Personal)	Email2	plain text
Email 3 (Alternate)	Email3	plain text
Email 4 (Alternate)	Email4	plain text
Aux ID 1	AuxID1	plain text
Aux ID 2	AuxID2	plain text
Aux ID 3	AuxID3	plain text
Aux ID 4	AuxID4	plain text



Step 2 of 2: Map the SQL Server data store properties

The second part of adding a SQL Server data store is mapping the data store properties.

Each user is uniquely identified by profile data that is read from or stored in your directories and databases.

The Identity Platform does not store user profiles, so your SQL Server attributes must be mapped to Identity Platform profile properties to be read and updated in the directory by the Identity Platform. The directory attribute mapped to the property is retrieved only when required for authentication or assertion purposes.

For more information about how data store profile properties are stored for on-premises, hybrid, or cloud Identity Platform deployments, see List of stored profile field properties.

1. On the Map Data Store Properties page, for the mapped Aux ID 1 through Aux ID 10 fields, specify the **Data Format** to define how data is encrypted and stored in the directory.
   For cloud deployments, any profile properties mapped to Aux ID 1 through Aux ID 10 are stored in the cloud. These properties are generated and used by SecureAuth, such as device recognition profiles, OATH tokens, push tokens, knowledge-based questions and answers (KBQ/KBA), PIN, and access histories.
   The selection options are:
   - **plain text** – store data as regular, readable text (default)
   - **standard encryption** – store and encrypt data using RSA encryption
   - **advanced encryption** – store and encrypt data using AES encryption
   - **standard hash** – store and encrypt data using SHA-256 hash
2. Click **Save Data Store**.
   The SQL Server data store you just added appears in the User Data Stores list.

Related information

SQL tables and stored procedures configuration (9.1 to 19.07.10-10)

SQL tables and stored procedures configuration (19.07.01-11+)

View and edit data store integration