

---

# SQL Connector

## Introduction

## Description

The SQL connector allows an easy way to configure and manage relational databases.

## Managed System

There are a lot of relational databases, currently, these are the supported databases.

- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Informix
- IBM DB2/400
- Sybase
- ODBC

For more information: [List of relational databases](#)

If your system is not in the previous list, it's possible to include it easily!

For more information to check if your system may be synchronized with this connector you do not hesitate to contact us through our [Contact form](#)

## Prerequisites

It is needed a user with access and permissions to the schemes and tables required in the scope of the integration.

To configure DB2/400 or Sybase it is mandatory to install the drivers in the lib directory of the Sync Server.

The Java-ODBC bridge is deprecated in Java, and the support will be removed shortly.

# Download and Install

The SQL is part of the default connectors, you do not need to install it, but you can upgrade it from the download management section.

You can visit the [Connector Getting started page](#) for more information about the installation process.

## Agent Configuration

### Basic

#### Generic parameters

After the installation of the addon, you may create and configure agent instances.

To configure this SQL connector you must select "Customizable SQL agent" in the attribute "Type" of the generic parameters section in the agent's page configuration.

For more information about how you may configure the generic parameters of the agent, see the following link: [Agents configuration](#)

Task engine mode: Automatic (each change is automatically sent to target systems)

Name: SQL\_Connector

Description: SQL\_Connector

Type: SQL Agent Class:com.soffid.iam.sync.agent.SQLAgent2

Server: Each main synchronization server

Shared Thread: ☒ Yes ☐ No Dedicated threads: 1

Task timeout (ms): Long task timeout (ms):

Trust passwords: ☒ Yes ☐ No

Authoritative identity source: ☒ Yes ☐ No -

Read only: ☒ Yes ☐ No

Manual account creation: ☒ Yes ☐ No

User domain: Default user domain \*

Passwords domain: Default password domain \*

# Custom parameters

Below there are the specific parameters for this agent implementation:

Parameter	Description
User name	Database user name to authenticate
Password	The password of the database user
Driver	Identifies the driver of the relational database to use. Currently, these are the supported databases: MySQL (& MariaDB), PostgreSQL, Oracle, MS SQL Server, Informix, DB2/400, DB2 Universal, Sybase, ODBC

Parameter	Description
DB URL	<p>URL that identifies the connection properties. Please refer to the specific database vendor documentation to build this URL.</p> <pre>jdbc:mariadb://&lt;HOST&gt;/&lt;DATA_BASE&gt;</pre> <pre>jdbc:mysql://&lt;HOST&gt;/&lt;DATA_BASE&gt;</pre> <pre>jdbc:postgresql://&lt;HOST&gt;/&lt;DATA_BASE&gt;</pre> <pre>jdbc:oracle:&lt;drivertype&gt;:@&lt;database&gt;</pre> <pre>jdbc:sqlserver://&lt;HOST&gt;;databaseName=&lt;DATA_BASE&gt;</pre> <p>(*) <i>More documentation about the DB URL</i></p>
SQL Sentence to execute at startup	Each time the connection to the agent is established, this SQL statement will be executed.
Password hash algorithm	The algorithm is used to encrypt the password. For instance SHA1, SHA256, MD5, etc
Password hash prefix	<p>Prefix to add it to the password.</p> <pre>{SHA1}BzE/DjIPIsV6Nc/CIFCOs/9FfH4=</pre> <pre>{SHA256}AIEM+LINb8ucXeSE077EGHYgs+KHblmquQ2FL+Dxj7Y=</pre>
Enable debug	<p>Two options: <b>Yes</b>, and <b>No</b>.</p> <p>It enables or not more log traces in the Synchronization Server log</p>
Synchronization method	<ul style="list-style-type: none"> <li>• <b>Full synchronization:</b> persists the changes made in Soffid, regardless of the possible changes made in the final system.</li> <li>• <b>Incremental synchronization:</b> this type of synchronization is used to avoid losing changes that have been made to the target system. First, Soffid's changes will be propagated to the target system, and then the changes on the target system will be made in the Soffid system. If the changes are in the same attribute, the Soffid value is the one that will persist.</li> </ul> <p>(**)</p>

User name	<input type="text" value="root"/>
Password	<input type="password" value="....."/>
Driver	<input type="text" value="MySQL"/>
DB URL	<input type="text" value="jdbc:mariadb://sql-server-test/RRHH"/>
SQL Sentence to execute at startup	<input type="text"/>
Password hash algorithm	<input type="text"/>
	e.g. SHA 
Password hash prefix	<input type="text"/> e.g. {SHA}
Enable debug	<input type="text" value="No"/>
Synchronization method	<input type="text" value="Full synchronization"/>

# Attribute mapping

This connector can manage users, accounts, roles, groups, and grants.

## Properties

Some agents require to configure some custom attributes, you will use the properties section to do that.

Any SQL sentence gets its parameters in three step process:

1. The synchronization engine creates the Soffid object.
2. The Soffid object is translated into a managed system object, using the attribute translation rules.
3. Soffid parser looks for any identifier preceded by a colon (:) symbol. For any symbol found, the symbol is replaced by a parameter whose value is the managed system attribute with the replaced identifier.

Once the SQL sentence has been executed, in the case of SELECT clauses, the column names are used to generate a virtual managed system object. The last step is to apply the attribute translation to generate the Soffid object to be populated.

These are the properties required to map every object of the mapping:

Property	Value
----------	-------

selectAll	<p>SQL sentence that needs to be executed to retrieve all the objects that currently exist on the database.</p> <ul style="list-style-type: none"> <li>• Applies to authoritative identity sources.</li> <li>• On non-authoritative identity sources, only the columns needed to calculate the <b>name</b> soffid attribute are needed.</li> </ul> <p>You can use this property with the following objects: <b>user</b>, <b>account</b>, <b>role</b>, and <b>authoritative change</b>.</p> <div data-bbox="373 423 1485 490">SELECT * FROM USERS</div> <div data-bbox="373 524 1485 591">SELECT * FROM ROLES</div>
check	<p>SQL sentence that will return when a single object already exists on the database. You can use this property with <b>all the Soffid objects</b>.</p> <div data-bbox="373 763 1485 831">SELECT ID FROM USERS WHERE USER=:USER</div> <div data-bbox="373 864 1485 931">SELECT ID FROM ROLES WHERE ROLE=:ROLE</div>
insert	<p>SQL sentence to create a new object. You can use this property with <b>all the Soffid objects</b>.</p> <div data-bbox="373 1104 1485 1171">INSERT INTO USERS VALUES (:USER, :FIRST_NAME, :LAST_NAME, :MAIL, :GROUP)</div> <div data-bbox="373 1205 1485 1272">INSERT INTO USER_ROLES (USER_NAME, ROLE_NAME) VALUES (:USER_NAME, :ROLE_NAME)</div>
update	<p>SQL sentence to update an existing object. You can use this property with <b>all the Soffid objects</b>.</p> <div data-bbox="373 1442 1485 1559">UPDATE USERS SET FIRST_NAME=:FIRST_NAME, LAST_NAME=:LAST_NAME, MAIL=:MAIL, GROUP=:GROUP WHERE ID=:ID</div> <div data-bbox="373 1592 1485 1659">UPDATE ROLES SET DESCRIPTION=:DESCRIPTION WHERE ROLE=:ROLE</div>
delete	<p>SQL sentence to remove (or disable) an existing object. You can use this property with <b>all the Soffid objects</b>.</p> <div data-bbox="373 1832 1485 1899">DELETE FROM USERS WHERE ID=:ID</div> <div data-bbox="373 1933 1485 2000">DELETE FROM USER_ROLES WHERE ID=:ID</div>

selectByAccount	<p>SQL sentence to retrieve all the role grants made to an account (for single account information). You can use this property with the following objects: <b>grant</b>.</p> <pre>SELECT * FROM USER_ROLES WHERE USERNAME=:USER</pre>
selectByName	<p>SQL sentence to fetch role information based on its name (for single role information). You can use this property with the following objects: <b>role</b>.</p> <pre>SELECT * FROM ROLES WHERE ROLE=:ROLE</pre>
updatePassword	<p>SQL sentence to update the user password. You can use this property with the following objects: <b>user</b> and <b>account</b>.</p> <pre>UPDATE USERS SET PASS=:PASS WHERE USER=:USER</pre>
validatePassword	<p>SQL sentence to check the user password. You can use this property with the following objects: <b>user</b> and <b>account</b>.</p> <pre>SELET 1 FROM USERS WHERE PASS=:PASS AND USER=:USER</pre>

## Attributes

You can customize attribute mappings, you only need to select system objects and the Soffid objects related, manage their attributes, and make either inbound or outbound attribute mappings.

You may map the attributes of the target system with the Soffid available attributes.

- For the target system attributes are required to be accessible to its specification
- For the Soffid attributes, you may follow the next link

For more information about how you may configure attribute mapping, see the following link: [Soffid Attribute Mapping Reference](#)

Example for roles:

Property	Value	+
delete	DELETE FROM USER_ROLES WHERE ID=:ID	—
insert	INSERT INTO USER_ROLES (USERTNAME, ROLNAME) VALUES (:USERNAME, :ROLNAME)	—
selectByAccount	SELECT * FROM USER_ROLES WHERE USERNAME=:USER	—
selectByRole	SELECT * FROM USER_ROLES WHERE USERNAME=:USER	—

System attribute	Direction	Soffid attribute	+
MAIL	← v	shortName==null ? attributes{"MAIL"} : shortName + "@" + mailDomain	—
LAST_NAME	⇄ v	lastName	—
PASS	← v	password	—
GROUP	⇄ v	primaryGroup	—
FIRST_NAME	⇄ v	firstName	—
USER	⇄ v	accountName	—

Example for accounts:

Property	Value	+
check	SELECT ID FROM USERS WHERE USER=:USER	—
delete	DELETE FROM USERS WHERE ID=:ID	—
insert	INSERT INTO USERS VALUES (:USER, :FIRST_NAME, :LAST_NAME, :MAIL, :GROUP)	—
selectAll	SELECT * FROM USERS	—
selectByAccountName	SELECT * FROM USERS WHERE USER=:USER	—
update	UPDATE USERS SET FIRST_NAME=:FIRST_NAME, LAST_NAME=:LAST_NAME, MAIL=:MAIL, GROUP=:GROUP WHERE ID=:ID	—
updatePassword	UPDATE USERS SET PASS=:PASS WHERE USER=:USER	—
validatePassword	SELET 1 FROM USERS WHERE PASS=:PASS AND USER=:USER	—

System attribute	Direction	Soffid attribute	+
USER	⇄ v	accountName	—
PASS	← v	password	—
LAST_NAME	⇄ v	lastName	—
FIRST_NAME	⇄ v	firstName	—
GROUP	⇄ v	primaryGroup	—
MAIL	← v	shortName==null ? attributes{"MAIL"} : shortName + "@" + mailDomain	—

## Triggers

You can define BeanShell scripts that will be triggered when data is loaded into the target system (outgoing triggers). The trigger result will be a boolean value, true to continue or false to stop.



Triggers can be used to validate or perform a specific action just before performing an operation or just after performing an operation on target objects.

To view some examples, visit the [Outgoing triggers examples page](#).

# Integration flows

## Update User

Visit the [Integration flows Update user page](#) for more information

## Update Account

Visit the [Integration flows Update account page](#) for more information

---

(\*)

<https://mariadb.com/kb/en/about-mariadb-connector-j/>

<https://docs.microsoft.com/es-es/sql/connect/jdbc/building-the-connection-url?view=sql-server-ver16>

---

(\*\*) *Soffid provides two synchronization types:*

- *Full synchronization*
- *Incremental synchronization*

*The first type, the **full synchronization** method, persists the changes made in Soffid, regardless of the possible changes made in the target system.*

*For the second type, the **incremental synchronization** method, Soffid has developed a synchronization system, using custom internal attributes, to check what changes have been made to the different attributes of an object. Thus, it tries to avoid losing the changes that have been made in the target system. First, Soffid's changes will be propagated to the target system, and then the changes on the target system will be made in the Soffid system. If the changes are in the same attribute, the Soffid value is the one that will persist.*

---

Revision #41

Created 19 April 2021 15:29:15 by pgarcia@soffid.com

Updated 1 March 2024 08:16:21 by pgarcia@soffid.com