Informatica

Informatica® Cloud Data Integration

Microsoft Azure Synapse SQL Connector

Informatica Cloud Data Integration Microsoft Azure Synapse SQL Connector
July 2024

© Copyright Informatica LLC 2018, 2024

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, Informatica Cloud, and PowerCenter are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

See patents at https://www.informatica.com/legal/patents.html.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2024-12-11

Table of Contents

Preface 6

Informatica Resources 6

- Informatica Documentation 6
- Informatica Intelligent Cloud Services web site 6
- Informatica Intelligent Cloud Services Communities 6
- Informatica Intelligent Cloud Services Marketplace 7
- Data Integration connector documentation 7
- Informatica Knowledge Base 7
- Informatica Intelligent Cloud Services Trust Center 7
- Informatica Global Customer Support 7

Chapter 1: Introduction to Microsoft Azure Synapse SQL Connector 8

Microsoft Azure Synapse SQL Connector assets 8

Chapter 2: Connections for Microsoft Azure Synapse SQL 9

Prerequisites 9

- Azure Active Directory authentication 9
- Service Principal authentication 11
- Managed Identity authentication 11
- Serverless SQL pool 12

Connect to Microsoft Azure Synapse SQL 12

- Before you begin 12
- Connection details 13
- Azure storage types 15
- Advanced settings 18

Verify permissions 18

Chapter 3: Mappings for Microsoft Azure Synapse SQL 20

Microsoft Azure Synapse SQL sources in mappings 20

- Adding multiple source objects 23
- Key range partitioning 26

Microsoft Azure Synapse SQL targets in mappings 28

- Copy command 31
- Rejected rows 32
- Mapping tasks with CDC sources 32
- Changed data capture limitations 34
- Bulk processing for write operations 34
- Rules and guidelines for configuring update override 34

Microsoft Azure Synapse SQL lookups in mappings 35

- Enabling lookup caching 38

Optimize the staging performance for a mapping. . . . 39
Process SQL queries and stored procedures using an SQL transformation. . . . 40
Configuring an SQL transformation. . . . 40
Rules and guidelines for SQL transformations. . . . 41
Dynamic schema handling 42
Rules and guidelines for dynamic schema handling. . . . 42
IDENTITY column. . . . 43
Parameterization. . . . 43
Rules and guidelines for mappings and mapping tasks. . . . 44
Mappings in advanced mode example. . . . 48
Rules and guidelines for mappings in advanced mode. . . . 49

Chapter 4: Upgrading to Microsoft Azure Synapse SQL Connector. . . . 51

Connection switching example. . . . 52
Properties retained after the upgrade. . . . 54
Rules and guidelines. . . . 56

Chapter 5: Migrating a mapping. . . . 57

Use the same object path for the migrated mapping. . . . 57
Use a different object path for the migrated mapping. . . . 57
Migration options. . . . 58
General rules and guidelines. . . . 59

Chapter 6: SQL ELT optimization. . . . 61

SQL ELT optimization types. . . . 61
Data Integration behavior with source and full SQL ELT optimization. . . . 61
SQL ELT optimization scenarios. . . . 62
Previewing SQL ELT optimization 62
Configuring SQL ELT optimization. . . . 63
SQL ELT optimization using a Microsoft Azure Synapse SQL connection. . . . 63
Functions with Microsoft Azure Synapse SQL 64
Operators with Microsoft Azure Synapse SQL. . . . 69
Transformations with Microsoft Azure Synapse SQL. . . . 70
Supported features for Microsoft Azure Synapse SQL mappings. . . . 72
Supported features for Microsoft Azure Data Lake Storage Gen2 source. . . . 74
Configuring a custom query for the Microsoft Azure Synapse SQL source object. . . . 75
Rules and guidelines for SQL ELT optimization. . . . 75

Chapter 7: Data type reference. . . . 80

Microsoft Azure Synapse SQL and transformation data types. . . . 80

Chapter 8: Troubleshooting. . . . 83

Troubleshooting a Microsoft Azure Synapse SQL connection. . . . 83

Troubleshooting a mapping or mapping task. 83
Troubleshooting a mapping in advanced mode. 85

Index. 86

Preface

Use *Microsoft Azure Synapse SQL Connector* to learn how to read from or write to Microsoft Azure Synapse SQL. Learn to create a connection, develop and run mappings, mapping tasks, dynamic mapping tasks, and data transfer tasks in Cloud Data Integration. Learn how to push down the transformation logic for processing to the Microsoft Azure Synapse SQL database.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to Microsoft Azure Synapse SQL Connector

You can use Microsoft Azure Synapse SQL Connector to read data from or write data to Microsoft Azure Synapse SQL.

When you use Microsoft Azure Synapse SQL Connector, you can create a Microsoft Azure Synapse SQL connection and use the connection in Data Integration mappings and tasks. You can use Microsoft SQL Server or Azure Active Directory authentication to connect to Microsoft Azure Synapse SQL.

You can use external tables as Microsoft Azure Synapse SQL sources in mappings. You can read data from or write data to Microsoft Azure Synapse SQL case-sensitive databases as well. You can create mappings and mapping tasks to read data from or write data to a Microsoft Azure Synapse SQL endpoint that resides in a virtual network (VNet).

You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality. A mapping in advanced mode can run on an advanced cluster hosted on Microsoft Azure, local cluster, or a self-service cluster.

Microsoft Azure Synapse SQL Connector is optimized for large data sets and can perform better than traditional data integration methods, such as ODBC or JDBC. When you write data to Microsoft Azure Synapse SQL, the connector stages data files to Microsoft Azure Storage and uses T-SQL commands with Microsoft Polybase to extract or load relational and non-relational data in parallel.

You can connect to a serverless SQL pool when you read data from Microsoft Azure Synapse SQL.

Microsoft Azure Synapse SQL Connector assets

Create assets in Data Integration to integrate data using Microsoft Azure Synapse SQL Connector.

When you use Microsoft Azure Synapse SQL Connector, you can include the following Data Integration assets:

- Data transfer task
- Dynamic mapping task
- Mapping
- Mapping task

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

CHAPTER 2

Connections for Microsoft Azure Synapse SQL

Create a Microsoft Azure Synapse SQL connection to securely read data from or write data to Microsoft Azure Synapse SQL.

You can use a Microsoft Azure Synapse SQL connection to specify sources, targets, and lookups in mappings and mapping tasks. You can also use the Microsoft Azure Synapse SQL connection in an SQL transformation.

Prerequisites

You can configure Microsoft SQL Server, Azure Active Directory, Managed Identity, and Service Principal authentication types to access Microsoft Azure Synapse SQL.

You can also connect to a serverless SQL pool when you read data from Microsoft Azure Synapse SQL. When you connect to a serverless SQL pool, you can configure Microsoft SQL Server, Azure Active Directory, and Managed Identity authentication types to access Microsoft Azure Synapse SQL. For more information on the authentication details, see the Informatica How-To Library article, Prerequisites to connect to a serverless SQL pool.

Before you configure the connection properties, you need to keep the authentication details handy based on the authentication type that you want to use.

Azure Active Directory authentication

To connect to Microsoft Azure Synapse SQL using Azure Active Directory (AAD) authentication, you need to create an Azure Active Directory administrator and an Azure Active Directory user.

Import a server certificate

If a trust store file isn't configured for your organization and you want to use AAD authentication with Active Directory Federation Services in Azure, you need to import the server certificate. For more information, contact your organization administrator.

Import the server certificate to the following location:

```
<Secure Agent installation directory>\jdk\jre\lib\security\cacerts
```

Use the following command to import the certificate:

```
keytool -import -trustcacerts -alias <alias name of the certificate> -file <certificate file path> -keystore
```

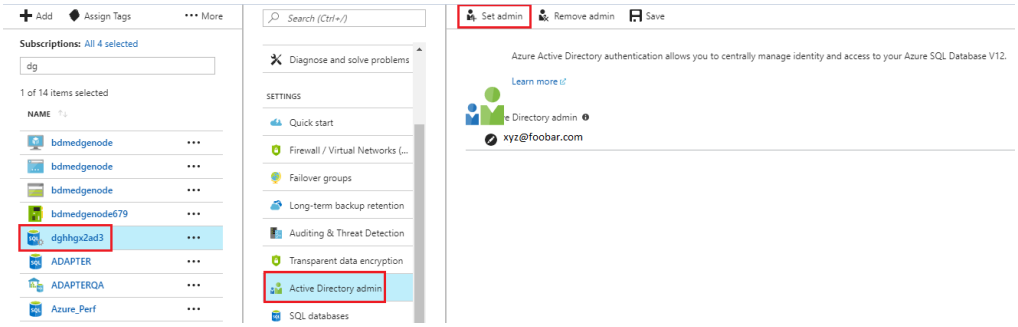
```
<Secure Agent installation directory>\jdk\jre\lib\security\cacerts

-storepass <password for the truststore>
```

Create an Azure Active Directory administrator

To add new users to your Azure Active Directory, you must have an administrator role.

To set up an Azure Active Directory administrator for AAD and Microsoft SQL Server that hosts your Microsoft Azure Synapse SQL, perform the following steps:

1. Log on to the Microsoft Azure portal using your credentials.
   The Dashboard page appears.
2. From the All Resources page, select the Microsoft SQL Server that hosts Microsoft Azure Synapse SQL.
3. Under Settings displayed for Microsoft SQL Server, select the **Active Directory admin** option.
   The image shows the Active Directory admin settings:



4. Click **Set admin**.
   The Add admin page appears.
5. Enter the email ID that you want to use as admin, and then click **Select**.
6. Click **Save**.

Create an Azure Active Directory user

Create an AAD user and use the AAD user credentials when you configure a Microsoft Azure Synapse SQL connection with AAD authentication.
Perform the following steps to create an AAD user:

1. Connect to Microsoft Azure Synapse SQL using the Azure Active Directory administrator created in the previous steps.
   You can use Microsoft SQL Server Management Studio to connect to the Microsoft Azure Synapse SQL.
2. In a new query window in Microsoft SQL Server Management Studio, run the following command to create an AAD user:
   ```
   create user [user@foobar.com] from external provider;
   ```

3. Assign the following privileges to the user:

```
CREATE USER [username] FROM EXTERNAL PROVIDER;
ALTER ROLE db_datareader ADD MEMBER [username]
ALTER ROLE db_datawriter ADD MEMBER [username]
GRANT EXECUTE TO [username]
grant ALTER ANY EXTERNAL DATA SOURCE to [username];
grant create table to [username];
grant create schema to [username];
grant select to [username];
grant update to [username];
grant insert to [username];
grant delete to [username];
grant create view to [username];
grant select on schema :: sys to [username];
grant control to [username];
EXEC sp_addrolemember 'db_owner','[username]';
ALTER ROLE db_owner ADD MEMBER [username]
```

Service Principal authentication

Service Principal authentication involves the use of a service principal identity to authenticate and authorize access to Azure resources. Before you use service principal authentication to connect to Microsoft Azure Synapse SQL, be sure to complete certain prerequisites.

1. Register a service principal application.
2. Configure a service principal user for a serverless SQL pool.
3. Configure a service principal user for a dedicated SQL pool.

For more information, see Prerequisites to use service principal authentication Informatica How-To Library article.

Managed Identity authentication

Managed Identity authentication uses managed identities in Azure Active Directory to authenticate and authorize access to Azure resources securely.

When you use managed identity authentication to connect to Microsoft Azure Synapse SQL, the user for the system assigned identity is the virtual machine for which you enable the identity. The user for the user assigned identity is the user identity that you create in the Azure portal.

Before you use managed identity authentication to connect to Microsoft Azure Synapse SQL or Microsoft Azure Data Lake Storage Gen2, be sure to complete certain prerequisites.

1. Create an Azure virtual machine.
2. Install the Secure Agent on the Azure virtual machine.
3. Enable system assigned identity or user assigned identity for the Azure virtual machine.
   If you enable both and do not specify the client ID, the system assigned identity is used for authentication.

4. After you add or remove a managed identity, restart the Azure virtual machine.

Serverless SQL pool

You can configure a Microsoft Azure Synapse SQL connection to connect to a serverless SQL pool. A serverless SQL pool does not store data or require any preconfigured infrastructure. You can connect to a serverless SQL pool when you want to query external tables that reference data stored in Microsoft Azure Data Lake Storage Gen2 or when you want to use queries with the OPENROWSET function.

To connect to a serverless SQL pool, specify the Azure DW JDBC URL connection string for a serverless SQL pool in the Microsoft Azure Synapse SQL connection.

Before you connect to a serverless SQL pool to read from Microsoft Azure Synapse SQL, be sure to complete the following prerequisites:

1. Configure an Azure Analytics serverless pool workspace.
2. Create an SQL database in the serverless pool workspace.
3. Get the JDBC URL for the following authentication types:
   - Microsoft SQL Server authentication
   - Azure Active Directory (AAD) authentication
   - Managed Identity authentication
4. To use Service Principal authentication to connect to Microsoft Azure Data Lake Storage Gen2 to stage the files, get credentials for Service Principal.
5. Configure steps to read data from a file using the OPENROWSET query or by creating an external table.

For more information, see Prerequisites to connect to a serverless SQL pool Informatica How-To Library article.

Connect to Microsoft Azure Synapse SQL

Let's configure the Microsoft Azure Synapse SQL connection properties to connect to Microsoft Azure Synapse SQL.

Before you begin

Before you get started, you'll need to get information from your Microsoft Azure Synapse SQL account based on the authentication type that you want to configure.

Check out "Azure Active Directory authentication" on page 9, "Managed Identity authentication" on page 11, and "Service Principal authentication" on page 11 for the information you need from Microsoft Azure Synapse SQL.

To get the JDBC URL from your Azure account, see Obtaining the JDBC URL How-To Library article.

Check out "Serverless SQL pool" on page 12 for the information you need to connect to a serverless SQL pool.

Connection details

The following table describes the basic connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.
Type	Microsoft Azure Synapse SQL
Runtime Environment	The name of the runtime environment where you want to run tasks. Select a Secure Agent, Hosted Agent, or serverless runtime environment. Hosted Agent doesn't apply to mappings in advanced mode.

Property	Description

Azure DW JDBC URL	The Microsoft Azure Synapse SQL JDBC connection string.
	Use the following string to connect to Microsoft Azure Synapse SQL:
	`jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database>`
	You can include an authentication parameter in the connection string to specify the authentication type. You can configure the following authentication types to connect to Microsoft Azure Synapse SQL:
	- Microsoft SQL Server
	- Azure Active Directory
	- Managed Identity
	- Service Principal
	If you don't include an authentication parameter in the connection string, the Secure Agent uses Microsoft SQL Server authentication as the authentication type.
	Use the following string to connect to a serverless SQL pool in Microsoft Azure Synapse SQL:
	`jdbc:sqlserver://<Serverless SQL endpoint>:1433;database=<Database>;Authentication=ActiveDirectoryMsi;`
	Connection string format for Microsoft SQL Server authentication
	`jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database>`
	Connection string format for Azure Active Directory (AAD) authentication
	`jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database>;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;Authentication=ActiveDirectoryPassword;`
	Connection string format for Service Principal authentication
	`jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database>;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30; Authentication=ActiveDirectoryServicePrincipal;`
	Connection string format for Managed Identity authentication
	`jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database>;Authentication=ActiveDirectoryMsi;`

Property	Description
Azure DW JDBC Username	User name to connect to the Microsoft Azure Synapse SQL account. - For AAD authentication, provide your AAD user name. - For Microsoft SQL server authentication, provide your SQL auth user name. - For service principal authentication, provide the application ID or client ID for your application registered in Azure Active Directory. This property doesn't apply to Managed Identity authentication.
Azure DW JDBC Password	Password to connect to the Microsoft Azure Synapse SQL account. - For AAD authentication, provide the password of the AAD user. - For Microsoft SQL server authentication, provide the password of SQL auth user. - For service principal authentication, provide the client secret for your application registered in the Azure Active Directory. This property doesn't apply to Managed Identity authentication.
Azure DW Client ID	Required if you want to use the user-assigned managed identity for Managed Identity Authentication to connect to Microsoft Azure Synapse SQL. The client ID of the user-assigned managed identity. If you use system-assigned managed identity, leave the field empty.
Azure DW Schema Name	Name of the schema in Microsoft Azure Synapse SQL.

Azure storage types

You can select Microsoft Azure Blob Storage or Microsoft Azure Data Lake Storage Gen2 as the Azure storage type to stage the data files. Default is Azure Blob.

Select your preferred storage type and then configure the storage-specific parameters.

To get credentials for shared key authentication when you connect to Microsoft Azure Blob Storage or Microsoft Azure Data Lake Storage Gen2 to stage files, see Get credentials for shared key authentication How-To Library article.

To get credentials for service principal authentication when you connect to Microsoft Azure Data Lake Storage Gen2 to stage files, see Get credentials for service principal authentication How-To Library article.

Azure Blob storage

When you select Microsoft Azure Blob as the storage type, you can configure Shared Key Authentication as the authentication type to stage the files.

Note: If you connect to a serverless SQL pool, you must configure Microsoft Azure Data Lake Storage Gen2 as the storage type.

The following table describes the authentication type that you can configure for Microsoft Azure Blob storage:

Property	Description
Authentication Type	Authentication type to connect to Microsoft Azure Blob storage to stage the files. You can configure Shared Key Authentication as the authentication type to stage the files.

Shared Key Authentication

Uses the storage account name and account key to connect to Microsoft Azure Blob storage.

The following table describes the basic connection properties for shared key authentication:

Property	Description
Azure Blob Account Name	Name of the Microsoft Azure Blob Storage account to stage the files.
Azure Blob Account Key	The Microsoft Azure Blob Storage access key to stage the files.
Container Name	The name of the container in the Azure Blob Storage account.

ADLS Gen2 storage

When you select Microsoft Azure Data Lake Storage Gen2 as the storage type, you can configure various authentication types to stage the files.

The following table describes authentication types that you can configure for Microsoft Azure Data Lake Storage Gen2 storage:

Property	Description
Authentication Type	Authentication type to connect to Azure storage to stage the files. Select one of the following options: - Shared Key Authentication - Service Principal Authentication - Managed Identity Authentication For more information on how to configure the authentication types, see Setting up authentication to connect to Microsoft Azure Synapse SQL.

Shared Key Authentication

Uses the storage account name and account key to connect to Microsoft Azure Data Lake Storage Gen2.

Note: You cannot select shared key authentication type when you connect to a serverless SQL pool.

The following table describes the basic connection properties for shared key authentication:

Property	Description
ADLS Gen2 Storage Account Name	Name of the Microsoft Azure Data Lake Storage Gen2 account to stage the files.
ADLS Gen2 Account Key	The Microsoft Azure Data Lake Storage Gen2 access key to stage the files.
File System Name	The name of the file system in the Microsoft Azure Data Lake Storage Gen2 account.

Service Principal Authentication

Uses the account name, client ID, client secret, and tenant ID to connect to Microsoft Azure Data Lake Storage Gen2.
The following table describes the basic connection properties for service principal authentication:

Property	Description
ADLS Gen2 Storage Account Name	Name of the Microsoft Azure Data Lake Storage Gen2 account to stage the files.
Client ID	The client ID of your application. Enter the application ID or client ID for your application registered in the Azure Active Directory.
Client Secret	The client secret for your application.
Tenant ID	The directory ID or tenant ID for your application.
File System Name	The name of the file system in the Microsoft Azure Data Lake Storage Gen2 account.

Managed Identity Authentication

Select this authentication type to authenticate using system-assigned or user-assigned identities that are assigned to applications in Azure to access Azure resources in Microsoft Azure Data Lake Storage Gen2.

The following table describes the basic connection properties for managed identity authentication:

Property	Description
ADLS Gen2 Storage Account Name	Name of the Microsoft Azure Data Lake Storage Gen2 account to stage the files.
Client ID	The client ID of your application. Enter the client ID for the user-assigned managed identity. If the managed identity is system-assigned, leave the field empty.
File System Name	The name of the file system in the Microsoft Azure Data Lake Storage Gen2 account.

Advanced settings

The following table describes the advanced connection properties:

Property	Description
External Data Source	The data source to create the external table. Ensure that the external data source exists in Microsoft Azure Synapse SQL and you have the permission to access the external data source. When you use the copy command method to load data from the staging location to Microsoft Azure Synapse SQL, you don't need to specify an external data source.
Staging Schema Name	The name of the schema that the Secure Agent uses to create external tables for staging data files. If you do not specify the staging schema name, the Secure Agent considers the configured Azure DW Schema Name.
Blob End-point	Type of Microsoft Azure endpoint. Select one of the following endpoints: - core.windows.net. Connects to Azure endpoints. Use this endpoint when you connect to a serverless SQL pool. - core.usgovcloudapi.net. Connects to US Government Microsoft Azure Synapse SQL endpoints. - core.chinacloudapi.cn. Connects to Microsoft Azure Synapse SQL endpoints in the China region. Default is core.windows.net.
VNet Rule	Enable to connect to a Microsoft Azure Synapse SQL endpoint residing in a virtual network (VNet). This property doesn't apply to a serverless runtime environment.

Verify permissions

Permissions define the level of access for the operations that you can perform in Microsoft Azure Synapse SQL.

You must verify the following permissions:

- Ensure that a default schema is present at the account level or user or group level in Microsoft Azure Synapse SQL.
- Verify that either the `db_owner` privilege or the following more granular privileges are granted to the user to connect to Microsoft Azure Synapse SQL and perform operations successfully:
  - `EXEC sp_addrolemember 'db_datareader', '<user>';` // Alternately assign permission to the individual table.
  - `EXEC sp_addrolemember 'db_datawriter', '<user>';` // Alternately assign permission to the individual table.
  - `GRANT ALTER ANY EXTERNAL DATA SOURCE TO <user>;`
  - `GRANT ALTER ANY EXTERNAL FILE FORMAT TO <user>;`
  - `GRANT CONTROL TO <user>;` // To grant all permissions on the database.
    or
    `GRANT ALTER ANY SCHEMA TO <user>;`// To grant permissions only on the schema.
  - `GRANT CREATE TABLE TO <user>;`

- Assign required privileges for tasks performed through Pre-SQL and Post-SQL commands.

- If you configure the staging schema name in the connection properties, ensure the following additional privileges are granted to the user:

  - `ALTER ROLE db_datareader ADD MEMBER <user>;`
  - `GRANT ALTER ANY EXTERNAL DATA SOURCE TO <user>;`
  - `GRANT ALTER ANY EXTERNAL FILE FORMAT TO <user>;`
  - `GRANT CREATE TABLE TO <user>;`
  - `GRANT ALTER ON SCHEMA::<staging_schema_name> TO <user>;`// Ensures this user does not have access to create or drop an external table in another schema.
  - `GRANT REFERENCES ON DATABASE SCOPED CREDENTIAL::<db_credential_name> TO <user>;`// Ensures this user does not have access to drop the credentials.
    For example, `GRANT REFERENCES ON DATABASE SCOPED CREDENTIAL::db_creds1 TO srvls;`

- If you have the ALTER ANY SCHEMA permissions, you must create the Master Key, Database Scoped Credential, and External Data Source in Microsoft Azure Synapse SQL that require the CONTROL permission on the database and specify the external data source when you create a connection.
  Also, Microsoft Azure Synapse SQL Connector does not delete the Database Scoped Credential and External Data Source. You must manually delete the Database Scoped Credential and External Data Source.

- When you use managed identity authentication to connect to Microsoft Azure Synapse SQL, grant permissions to the virtual machine and user identity. For example, `GRANT CONTROL TO <virtual machine name>;` and `GRANT CONTROL TO <user identity name>;`

CHAPTER 3

Mappings for Microsoft Azure Synapse SQL

When you configure a mapping, you describe the flow of data from the source to the target.

A mapping defines reusable data flow logic that you can use in mapping tasks.

When you create a mapping, you define the Source, Target, and Lookup transformations to represent a Microsoft Azure Synapse SQL object. Use the Mapping Designer in Data Integration to add the Source, Target, or Lookup transformations in the mapping canvas and configure the Microsoft Azure Synapse SQL source, target, and lookup properties. In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

You can use Monitor to monitor the jobs.

Microsoft Azure Synapse SQL sources in mappings

In a mapping, you can configure a Source transformation to represent a single Microsoft Azure Synapse SQL object. When you read data from Microsoft Azure Synapse SQL, you can configure key range partitioning to optimize the mapping performance at run time.

The following table describes the Microsoft Azure Synapse SQL properties that you can configure in a Source transformation:

Property	Description
Connection	Name of the source connection. You can select an existing connection, create a new connection, or define parameter values for the source connection property. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. You can switch between a non-parameterized and a parameterized Microsoft Azure Synapse SQL connection. When you switch between the connections, the advanced property values are retained.
Source Type	Type of the source object. Select Single Object, Multiple Objects, Query, or Parameter. When you specify a custom SQL query as source type, the Secure Agent evaluates the properties in the following order to run a mapping: 1. SQL Override 2. Table Name Override 3. Custom SQL You cannot parameterize a source query object at runtime in a mapping. You cannot specify Custom SQL as a parameter for mappings in advanced mode.
Object	The source object for the task. You can use external tables as Microsoft Azure Synapse SQL sources in mappings.
Parameter	Select an existing parameter for the source object or click **New Parameter** to define a new parameter for the source object. The **Parameter** property appears only if you select Parameter as the source type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. Ensure that the parameter file is in the correct format.
Filter	Filters records based on the filter condition. You can specify a simple filter or an advanced filter.
Sort	Not applicable.

The following table describes the Microsoft Azure Synapse SQL source advanced properties that you can configure in a Source transformation:

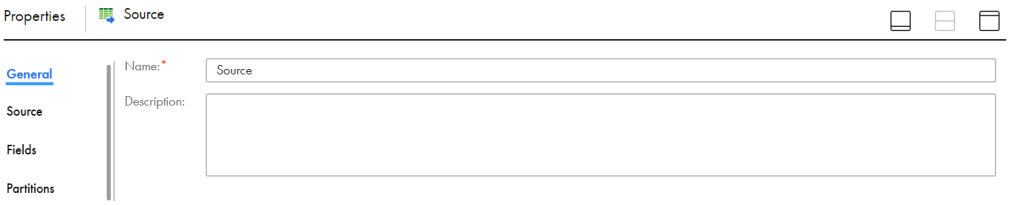
Property	Description
Azure Blob Container Name	Required if you select Azure Blob storage in the connection properties. The name of the container in Microsoft Azure Blob Storage. The container name must not contain special characters.
ADLS FileSystem Name	Required if you select ADLS Gen2 storage in the connection properties. The name of the file system in Microsoft Azure Data Lake Storage Gen2. The file system name must not contain special characters. You can also specify the path of the directory under the file system. Use only a forward slash to specify the directory path.
Schema Name Override	Overrides the schema specified in the connection.
Table Name Override	Overrides the table name of the imported Microsoft Azure Synapse SQL source table.
Staging File Format	Type of file format to use when you stage the files in Azure storage. Select one of the following formats: - Delimited Text - Parquet
Field Delimiter	Character used to separate fields in the file. Default is `0x1e`. You can specify 'TAB' or 0-256 single-char printable and non-printable ASCII characters. Non-printable characters must be specified in hexadecimal. **Note:** Multi-char ASCII characters except TAB are not applicable. You cannot use the following non-printable characters: `00x0`, `0x0`, `0x0A` , `0x1B`, `0x0D`, and `0x1F`
Number of Concurrent Connections to Blob Store[1]	Number of concurrent connections to extract data from the Microsoft Azure Blob Storage. When reading a large-size blob, you can spawn multiple threads to process data. Configure **Blob Part Size** to partition a large-size blob into smaller parts. Default is 4. Maximum is 10.
Blob Part Size[1]	Partitions a blob into smaller parts each of specified part size. When reading a large-size blob, consider partitioning the blob into smaller parts and configure concurrent connections to spawn required number of threads to process data in parallel. Default is 8 MB.
Pre-SQL	Pre-SQL command that must be run before reading data from the source.
Post-SQL	Post-SQL command that must be run after reading data from the source.
SQL Override	When you read data from a Microsoft Azure Synapse SQL object, you can configure SQL overrides and define constraints.

Property	Description
On Pre-Post SQL Error	Determines the behavior when a task that includes pre-SQL or post-SQL commands encounters errors. Select one of the following options: - Continue. The task continues regardless of errors. - Stop. The task stops when errors occur while executing pre-SQL or post-SQL commands.
Quote Character	Specifies the quote character to skip when you read data from Microsoft Azure Synapse SQL. The quote character that you specify must not exist in the source table. If it exists, enter a different quote character value. Default is `0x1f`.
Interim Directory[1]	Optional. Path to the staging directory in the Secure Agent machine. Specify the staging directory where you want to stage the files when you read data from Microsoft Azure Synapse SQL. Ensure that the directory has sufficient space and you have read permissions to the directory. Default staging directory is `/tmp`. You cannot specify an interim directory when you use the Hosted Agent.
Tracing Level[1]	Sets the amount of detail that appears in the log file. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.
[1]Doesn't apply to mappings in advanced mode.	

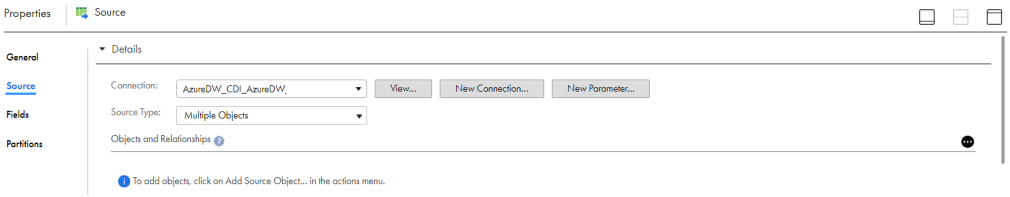
Adding multiple source objects

Perform the following steps to add multiple source objects in a mapping:

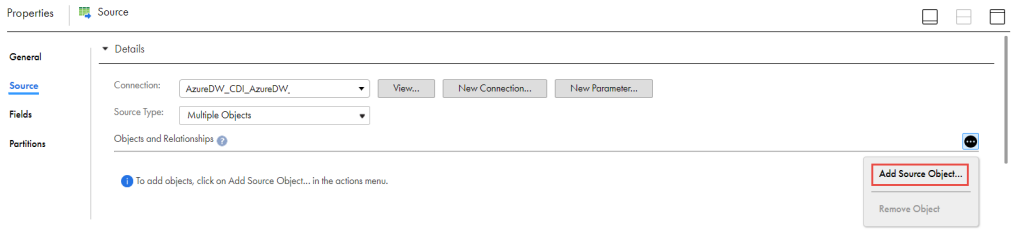
1. Click **New** > **Mappings**.
2. Select **Mapping** and click **Create**.
3. In the Source Properties page, specify the name and provide a description in the **General** tab.



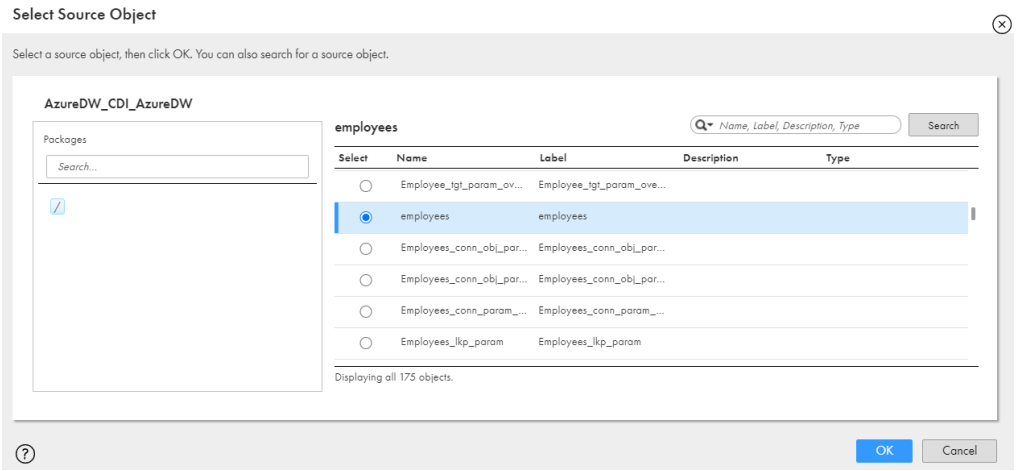
4. Click the **Source** tab.
5. Select the source connection and source type as **Multiple Objects** to be used for the task.



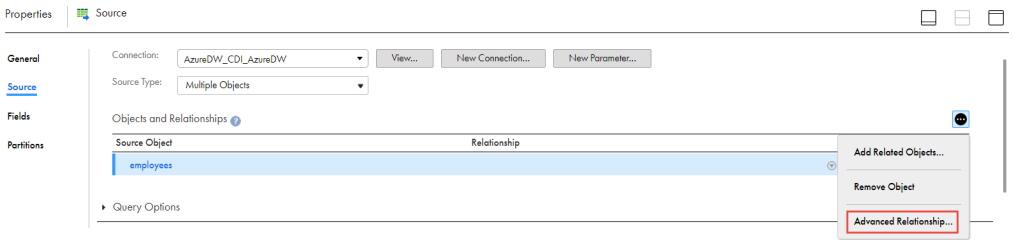
6. In the **Objects and Relationships** section, click the arrow to open the **Action** menu and then select **Add Source Object**.



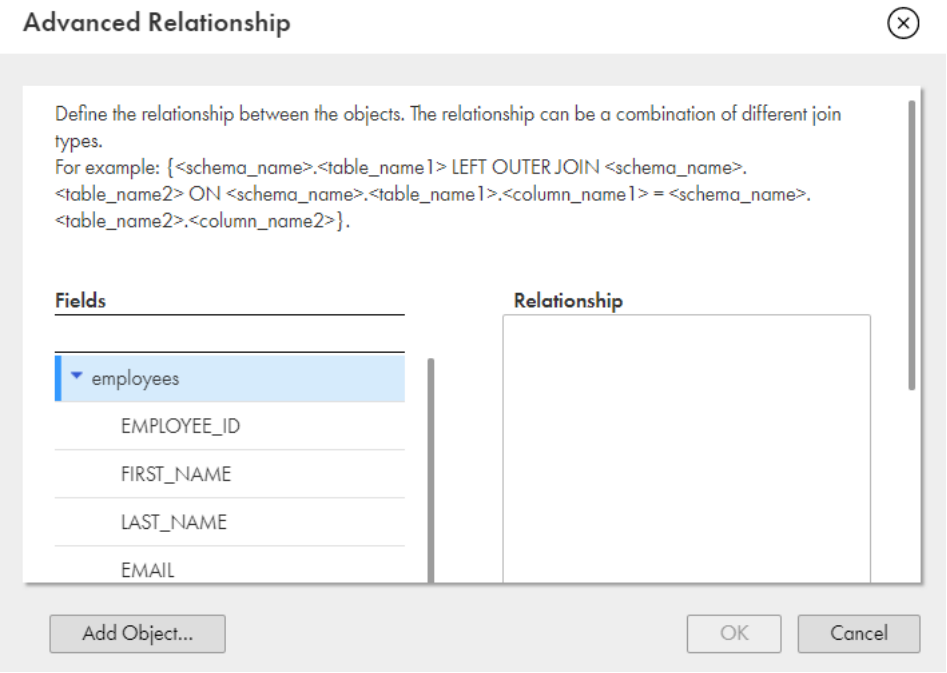
7. Select a source object from the list and click **OK**.



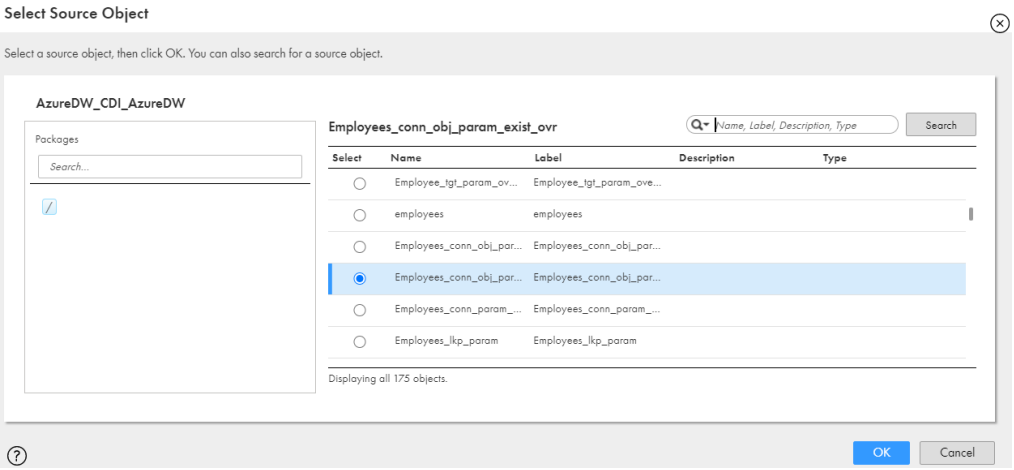
8. In the **Objects and Relationships** section, click the arrow next to the source object and then click **Advanced Relationship** to add related objects.



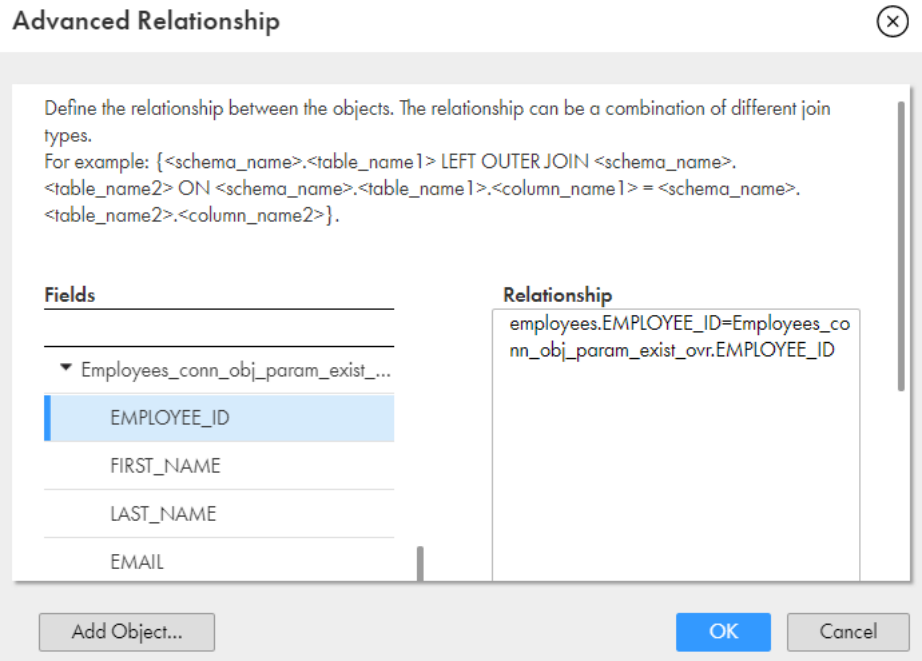
9. In the Advanced Relationship page, click **Add Object**.



10. In the Select Source Object page, select an object from the list and click **OK**.



11. In the Advanced Relationship page, select the required fields for the objects and create a join condition to define the relationship for all objects.



Note: You can join only two source tables. You must create an inner join to define the relationship. You must define only a join condition, not the complete join query. If the table name or column name is a reserved keyword or the table name contains special characters, enclose the name in square brackets. For example, `[AUTHORIZATION].[BROWSE]=[int].[BROWSE]`.

12. Click **OK**.

Key range partitioning

You can configure key range partitioning when you use a mapping task to read data from Microsoft Azure Synapse SQL sources. With key range partitioning, the Secure Agent distributes rows of source data based on the fields that you define as partition keys. The Secure Agent compares the field value to the range values for each partition and sends rows to the appropriate partitions.

Use key range partitioning for columns that have an even distribution of data values. Otherwise, the partitions might have unequal size. For example, a column might have 10 rows between key values 1 and 1000 and the column might have 999 rows between key values 1001 and 2000. If the mapping includes multiple sources, use the same number of key ranges for each source.

When you define key range partitioning for a column, the Secure Agent reads the rows that are within the specified partition range. For example, if you configure two partitions for a column with the ranges as 10 through 20 and 30 through 40, the Secure Agent does not read the rows 20 through 30 because these rows are not within the specified partition range.

You can configure a partition key for fields of the following data types:

- Any type of number data type. However, you cannot use decimals in key range values.

You cannot use key range partitions when you create a mapping in advanced mode and when a mapping includes any of the following transformations:

- Web Services
- XML to Relational

Note: When you use multiple partitions, the start range for the first partition and the end range for the last partition are not included in the WHERE clause of the polybase command. Custom queries and SQL override are not applicable when you use key range partitioning.

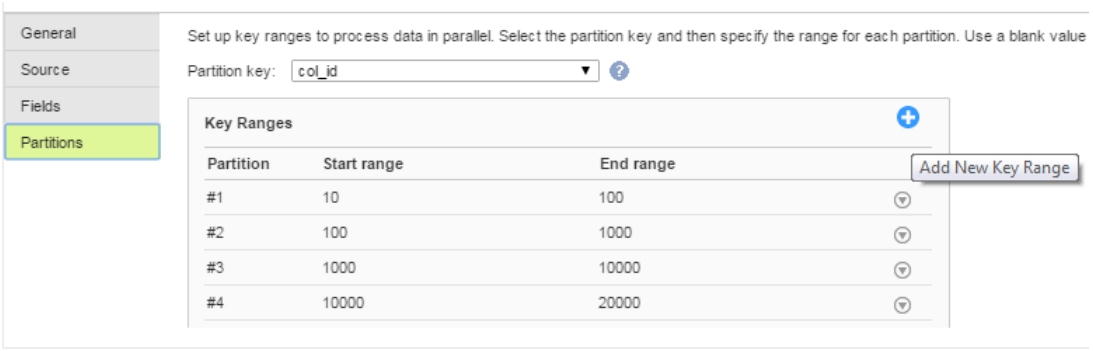
Configuring key range partitioning

Perform the following steps to configure key range partitioning for Microsoft Azure Synapse SQL sources:

1. In the Source Properties, click the **Partitions** tab.
2. Select the required partition key from the list.
3. Click **Add New Key Range** to define the number of partitions and the key ranges based on which the Secure Agent must partition data.

   Use a blank value for the start range to indicate the minimum value. Use a blank value for the end range to indicate the maximum value.

   The following image displays the **Partitions** tab:



Microsoft Azure Synapse SQL targets in mappings

In a mapping, you can configure a Target transformation to represent a single Microsoft Azure Synapse SQL. When the source is partitioned and you write data to Microsoft Azure Synapse SQL, the Secure Agent uses the pass-through partitioning to optimize the mapping performance at run time.

The following table describes the Microsoft Azure Synapse SQL properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the target connection. Select a target connection or click **New Parameter** to define a new parameter for the target connection. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. You can switch between a non-parameterized and a parameterized Microsoft Azure Synapse SQL connection. When you switch between the connections, the advanced property values are retained.
Target Type	Target type. Select one of the following types: - Single Object - Parameter. Select **Parameter** to define the target type when you configure the task.
Object	Name of the target object.
Parameter	Select an existing parameter for the target object or click **New Parameter** to define a new parameter for the target object. The **Parameter** property appears only if you select Parameter as the target type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties.
Create New at Runtime	Creates a Microsoft Azure Synapse SQL target at runtime. Enter a name for the target object and select the source fields that you want to use. By default, all source fields are used. Select **Use Exact Source Field Names in Target** to write special characters from the source column names to the target. You cannot parameterize the target at runtime.
Operation	The target operation. Select one of the following options: - Insert - Update - Upsert - Delete - Data Driven[1]. Select to honor flagged rows for an insert, update, delete, or reject operation from the Update Strategy transformation or a CDC source. You can also use the **Treat Source Rows As** advance property to perform insert, delete, update, or upsert operations. However, if you want to define a key for the delete, update, or upsert operation, you must use the Operation property. For more information about the operations, see "Rules and guidelines for mappings and mapping tasks" on page 44.

Property	Description
Data Driven Condition[1]	Enables you to define expressions that flag rows for an insert, update, delete, or reject operation. **Note:** Appears only when you select Data Driven as the operation type. However, you may leave the field empty as the rows in the Update Strategy transformation and CDC source tables are already marked with the operation types.
Update Column	The key columns to upsert or update data to or delete data from Microsoft Azure Synapse SQL. This property is honored only if you select delete, update, or upsert operation in the **Treat Source Rows As** advance property. **Note:** This property appears only if you select delete, update, upsert, or data driven operation in the **Operation** property.
[1]Doesn't apply to mappings in advanced mode.	

The following table describes the Microsoft Azure Synapse SQL properties that you can configure in a Target transformation:

Advanced Property	Description
Azure Blob Container Name	Required if you select Azure Blob storage in the connection properties. The name of the container in Microsoft Azure Blob Storage. The container name cannot contain special characters.
ADLS FileSystem Name	Required if you select ADLS Gen2 storage in the connection properties. The name of the file system in Microsoft Azure Data Lake Storage Gen2. The file system name cannot contain special characters. You can also specify the path of the directory under the file system. Use only a forward slash to specify the directory path.
Copy Method[1]	The method to load data from the staging location to Microsoft Azure Synapse SQL. Select one of the following options: - Polybase - Copy Command Default is Polybase.
Copy Command Options[1]	Options for the copy command in key=value format. Specify each option on a new line.
Schema Name Override	Overrides the schema specified in the connection.
Table Name Override	Overrides the table name of the imported Microsoft Azure Synapse SQL Data Warehouse target table.
Staging File Format	Type of file format to use when you stage the files in Azure storage. Select one of the following formats: - Delimited Text - Parquet

Advanced Property	Description
Field Delimiter	Character used to separate fields in the file. Default is `0x1e`. You can specify 'TAB' or 0-256 single-char printable and non-printable ASCII characters. Non-printable characters must be specified in hexadecimal. **Note:** Multi-char ASCII characters except TAB are not applicable. You cannot use the following non-printable characters: `00x0`, `0x0`, `0x0A` , `0x1B`, `0x0D`, and `0x1F`
Number of Concurrent Connections to Blob Storage[1]	Number of concurrent connections to extract data from the Microsoft Azure Blob Storage. When reading a large-size blob, you can spawn multiple threads to process data. Default is 4. Maximum is 10.
Truncate Table	Truncates the target before inserting data to the target.
Pre-SQL	Pre-SQL command that must be run before reading data from the source.
Post-SQL	Post-SQL command that must be run after writing data to the target.
On Pre-Post SQL Error	Determines the behavior when a task that includes pre-SQL or post-SQL commands encounters errors. You can select any of the following options: - Continue. The task continues regardless of errors. - Stop. The task stops when errors occur while executing pre-SQL or post-SQL commands.
Treat Source Rows As	Select one of the following options: - NONE - INSERT - DELETE - UPDATE - UPSERT - DATA DRIVEN[1]. Select to honor the flagged rows from the update strategy or any other custom transformation, or a CDC source. Default is None.
Batch Size[1]	Minimum number of rows in a batch. Enter a number greater than 0. Default is 2000000.
Reject Threshold	Number of errors within a batch that causes a batch to fail. Enter a positive integer. If the number of errors is equal to or greater than the property value, the Secure Agent rejects the entire batch to the error file and marks the session failed. **Note:** When you do not set the reject threshold, the mapping fails when an error is encountered.
Quote Character	Specifies the quote character to skip when you write data to Microsoft Azure Synapse SQL. The quote character that you specify must not exist in the source table. If it exists, enter a different quote character value.
Compression Format[1]	Compresses the staging files in the `.Gzip` format. Default is None.
Update Override	Overrides the default update SQL statement that the Secure Agent generates.

Advanced Property	Description
Interim Directory[1]	Optional. Path to the staging directory in the Secure Agent machine. Specify the staging directory where you want to stage the files when you write data to Microsoft Azure Synapse SQL. Ensure that the directory has sufficient space and you have write permissions to the directory. Default staging directory is `/tmp`. You cannot specify an interim directory when you use the Hosted Agent.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. By default, the mapping task forwards rejected rows to the next transformation.
[1]Doesn't apply to mappings in advanced mode.	

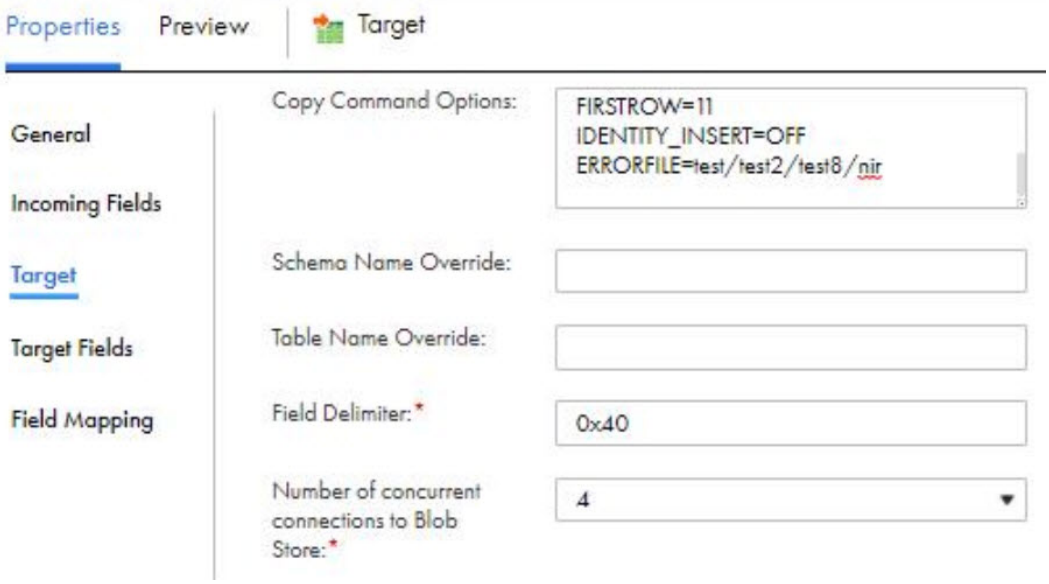
Copy command

You can use the copy command to load data from Microsoft Azure Blob Storage or Microsoft Azure Data Lake Storage Gen2 to Microsoft Azure Synapse SQL.

When you stage files in Microsoft Azure Data Lake Storage Gen2, you can use the copy command only with Service Principal Authentication.

You can specify the options for the copy command in key=value format in the Copy Command Options field. Specify each option in a new line.

The following image shows an example of the copy command options:



You can configure only the following copy command options in the advanced target properties. The default value is considered for other copy command options supported by Microsoft Azure.

ERRORFILE

Specifies the directory where you want to write rejected rows and the corresponding error file. The ERRORFILE option is equivalent to the Reject Directory advanced target property.

Use the following format: `ERRORFILE=<Directory Location>`.

You can specify the absolute path or relative path for the directory. The error file path must not contain special characters.

If you specify an absolute path, for example, `Dir1/Dir2`, the agent creates the reject directory in the following path:

```
Dir1/Dir2
```

If you specify a relative path, for example, `/Dir1/Dir2`, the agent creates the reject directory in the following path:

```
<staging path>/Dir1/Dir2
```

If you do not specify the directory, the agent creates the reject directory in the following path:

```
<staging path>/Reject_<UUID Randomly Generated>
```

MAXERRORS

Specifies the maximum number of reject rows allowed in the load before the copy command is canceled. Each row that cannot be imported by the copy command is ignored and counted as one error. The MAXERRORS option is equivalent to the Reject Threshold advanced target property.

Use the following format: `MAXERRORS=max_errors`. For example, `MAXERRORS=20`.

COMPRESSION

Specifies the data compression method for the data. You can use only Gzip compression for CSV files.

The COMPRESSION option is equivalent to the Compression Format advanced target property.

Use the following format: `COMPRESSION='method'`. For example, `COMPRESSION='GZIP'`.

FIELDQUOTE

Specifies a single character that is used as the quote character in the CSV file.

The FIELDQUOTE option is equivalent to the Quote Character advanced target property.

Use the following format: `FIELDQUOTE='field_quote'`.

FIELDTERMINATOR

Specifies the field terminator that is used in the CSV file.

The FIELDTERMINATOR option is equivalent to the Field Delimiter advanced target property.

Use the following format: `FIELDTERMINATOR='field_terminator'`. For example, `FIELDTERMINATOR='0x1f'`.

For more details about the copy command options, see the Microsoft Azure documentation.

Rejected rows

The Secure Agent generates two error files in the Microsoft Azure Blob container specified in the target properties.

The error files are generated with the keyword *Reject* as a prefix in the error file names. One error file contains an entry for each rejected row and the other error file lists the cause for the rejected rows. To generate the error files, specify **Reject Threshold** in advanced target properties.

Mapping tasks with CDC sources

Your organization needs to replicate real-time changed data from a mission-critical Oracle production system to minimize intrusive, non-critical work, such as offline reporting or analytical operations system. You can use Microsoft Azure Synapse SQL Connector to capture changed data from the Oracle CDC source and write the

changed data to a Microsoft Azure Synapse SQL. Add the Oracle CDC sources in mappings, and then run the associated mapping tasks to write the changed data to the target.

1. In Data Integration, click **New** > **Mapping** > **Create**.

   The **New Mapping** dialog box appears.
2. Enter a name and description for the mapping.
3. On the Source transformation, specify a name and description in the general properties.
4. On the **Source** tab, select the configured Oracle CDC connection and specify the required source properties.
5. On the Target transformation, specify a name and description in the general properties.
6. On the **Target** tab, perform the following steps to configure the target properties:
   a. In the **Connection** field, select the Microsoft Azure Synapse SQL connection.
   b. In the **Target Type** field, select the type of the target object.
   c. In the **Object** field, select the required target object.
   d. In the **Operation** field, select **Data Driven** to properly handle insert, update, and delete records from the source.
   e. In the **Data Driven Condition** field, leave the field empty.
   f. In the **Update Column** field, select the key columns to upsert or update data to or delete data from Microsoft Azure Synapse SQL. This property is honored only if you select delete, update, or upsert operation in the **Treat Source Rows As** advance property.
   g. In the **Advanced Properties** section, you must select Data Driven in the **Treat Source Rows As** property.
7. On the **Field Mapping** tab, map the incoming fields to the target fields. You can manually map an incoming field to a target field or automatically map fields based on the field names.
8. In the **Actions** menu, click **New Mapping Task**.

   The **New Mapping Task** page appears.
9. In the **Definition** tab, enter the task name and select the configured mapping.
10. In the **CDC Runtime** tab, specify the required properties.

    For more information about the **CDC Runtime** properties, see the *Informatica Cloud Data Integration Oracle CDC Connector User Guide*.
11. In the **Runtime Options** tab, specify the following properties in the **Advanced Session Properties** section:
    a. In the **Commit on End of File** field, select the value of the property as **No**.
    b. In the **Commit Type** field, select the value of the property as **Source**.
    c. In the **Recovery Strategy** field, select the value of the property as **Resume from last checkpoint**.
12. Click **Save** > **Run** the mapping.

    Alternatively, you can create a schedule that runs the mapping task on a recurring basis without manual intervention. You can define the schedule to minimize the time between mapping task runs.

In **Monitor**, you can monitor the status of the logs after you run the task.

Changed data capture limitations

Consider the following limitations when working with a Microsoft Azure Synapse SQL change data capture (CDC) target:

- The staging temporary tables are not dropped if a mapping task is terminated. You must clean up the temporary tables manually.
- The following keywords are not allowed for the CDC target:

  - Keywords not allowed in the table name: `"(","  SET ","  WHERE ","."`

    Keywords not allowed in column name: `") VALUES",",","  SET ","  WHERE ","  = ?","  AND "`

    Note: Consider the leading and trailing spaces with certain keywords.
- You cannot capture changed data for `datatimeoffset` and `time` data type.
- Date conversion from other data types to smalldatetime is not applicable.

Bulk processing for write operations

You can enable bulk processing to write large amounts of data to Microsoft Azure Synapse SQL. Bulk processing utilizes minimal number of API calls and the performance of the write operation is optimized.

To enable bulk processing, specify the property *-DENABLE_WRITER_BULK_PROCESSING=true* in the Secure Agent properties:

Perform the following steps to configure bulk processing before you run a mapping:

1. In Administrator, select the Secure Agent listed on the **Runtime Environments** tab.
2. Click **Edit**.
3. In the **System Configuration Details** section, select **Data Integration Server** as the service and **DTM** as the type.
4. Edit the JVM option, and enter *-DENABLE_WRITER_BULK_PROCESSING=true*.
5. Click **Save**.

You cannot use bulk processing in write operations for mappings in advanced mode.

You cannot use bulk processing in mapping tasks configured with SQL ELT optimization using the Microsoft Azure Synapse SQL ODBC connection or the Microsoft Azure Synapse SQL connection.

Rules and guidelines for configuring update override

Consider the following rules and guidelines when you use the update override property for a Microsoft Azure Synapse SQL target:

- Select **Update** for the **Treat source rows as** property in the advanced target properties.
- Specify the update query in the following format:

```
UPDATE <Target table name> SET <Target table name>.<Column1> = :TU.<Column1>, <Target
table name>.<Column2> = :TU.<Column2>, … <Target table name>.<ColumnN>
= :TU.<ColumnN> FROM :TU WHERE <Target table name>.<Update Column1> = :TU.<Update
Column1> AND <Target table name>.<Update Column2> = :TU.<Update Column2> AND …
<Target table name>.<Update ColumnN> = :TU.<Update ColumnN>
```

- Column names for :TU must match the target table column names.
- You must specify the update query with a valid SQL syntax because Microsoft Azure Synapse SQL Connector replaces :TU with a temporary table name and does not validate the update query.
- You cannot change the order of the column mappings using the update override.

- You cannot use unconnected fields to the target in the update query.

Microsoft Azure Synapse SQL lookups in mappings

Use a Lookup transformation to retrieve data based on a specified lookup condition.

Use a Microsoft Azure Synapse SQL Lookup transformation to look up data in a Microsoft Azure Synapse SQL object. For example, the source table includes the customer code, but you want to include the customer name in the target table to make summary data easy to read.

You can use the Microsoft Azure Synapse SQL Lookup transformation to look up the customer name in another Microsoft Azure Synapse SQL object.

You can also set the default column value for the return field in a cached Lookup transformation in a Microsoft Azure Synapse SQL mapping. However, in advanced mode, you can't set the default column value for the return field.

You can create the following lookups when you configure field mappings in a mapping task:

- Connected with cached or uncached
- Unconnected lookup
- Connected lookup with dynamic lookup cache

You cannot configure a connected or unconnected uncached lookup for mappings in advanced mode.

The following table describes the Microsoft Azure Synapse SQL lookup object properties that you can configure in a Lookup transformation:

Property	Description
Connection	Name of the lookup connection. You can select an existing connection, create a new connection, or define parameter values for the lookup connection property. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties. You can switch between a non-parameterized and a parameterized Microsoft Azure Synapse SQL connection. When you switch between the connections, the advanced property values are retained.
Source Type	Type of the source object. Select Single Object, Query, or Parameter. You cannot use custom query as the source type in cached and dynamic cache lookups.
Lookup Object	Name of the lookup object for the mapping.
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the lookup object or click **New Parameter** to define a new parameter for the lookup object. The **Parameter** property appears only if you select parameter as the source type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option when you create a parameter. When the task runs, the agent uses the parameters from the file that you specify in the task advanced session properties.

Property	Description
Define Query	If the source type is a query, displays the **Edit Custom Query** dialog box. Enter a valid custom query and click **OK**.
Multiple Matches	The behavior when the lookup condition returns multiple matches. Select one of the following options: - Return first row[1] - Return last row[1] - Return any row - Return all rows - Report error **Note:** If the lookup table has an IDENTITY column, the data for the first row and last row is returned based on the sorting of the IDENTITY column. If the matched rows have duplicates then the values returned are random.
[1]Doesn't apply to mappings in advanced mode.	

You cannot use Binary, Varbinary, and Datetimeoffset datatypes columns as lookup conditions.

When you use a cached lookup with Microsoft Azure Synapse SQL connection and if the lookup condition contains a NULL value, the lookup condition is ignored.

The following table describes the Microsoft Azure Synapse SQL lookup advanced properties that you can configure in a Lookup transformation:

Property	Description
Azure Blob Container Name	Required if you select Azure Blob storage in the connection properties. The name of the container in Microsoft Azure Blob Storage. The container name must not contain special characters.
ADLS FileSystem Name	Required if you select ADLS Gen2 storage in the connection properties. The name of the file system in Microsoft Azure Data Lake Storage Gen2. The file system name must not contain special characters. You can also specify the path of the directory under the file system. Use only a forward slash to specify the directory path.
Schema Name Override	Overrides the schema specified in the connection.
Table Name Override	Overrides the table name of the imported Microsoft Azure Synapse SQL source table.
Field Delimiter	Character used to separate fields in the file. Default is `0x1e`. You can specify 'TAB' or 0-256 single-char printable and non-printable ASCII characters. Non-printable characters must be specified in hexadecimal. **Note:** Multi-char ASCII characters except TAB are not applicable. You cannot use the following non-printable characters: `00x0`, `0x0`, `0x0A` , `0x1B`, `0x0D`, and `0x1F`

Property	Description
Number of Concurrent Connections to Blob Store[1]	Number of concurrent connections to extract data from the Microsoft Azure Blob Storage. When reading a large-size blob, you can spawn multiple threads to process data. Configure **Blob Part Size** to partition a large-size blob into smaller parts. Default is 4. Maximum is 10.
Blob Part Size[1]	Partitions a blob into smaller parts each of specified part size. When reading a large-size blob, consider partitioning the blob into smaller parts and configure concurrent connections to spawn required number of threads to process data in parallel. Default is 8 MB.
Pre-SQL	Pre-SQL command that must be run before reading data from the source.
Post-SQL	Post-SQL command that must be run after reading data from the source.
SQL Override	When you read data from a Microsoft Azure Synapse SQL object, you can configure SQL overrides and define constraints.
On Pre-Post SQL Error	Determines the behavior when a task that includes pre-SQL or post-SQL commands encounters errors. You can select any of the following options: - Continue. The task continues regardless of errors. - Stop. The task stops when errors occur while executing pre-SQL or post-SQL commands.
Quote Character	The Secure Agent skips the specified character when you read data from Microsoft Azure Synapse SQL. Default is `0x1f`.
Interim Directory[1]	Optional. Path to the staging directory in the Secure Agent machine. Specify the staging directory where you want to stage the files when you read data from Microsoft Azure Synapse SQL. Ensure that the directory has sufficient space and you have write permissions to the directory. Default staging directory is `/tmp`. You cannot specify an interim directory when you use the Hosted Agent.
Tracing Level[1]	Sets the amount of detail that appears in the log file. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.

Property	Description
Lookup Data Filter[1]	Limits the number of lookups that the mapping performs on the cache of the lookup source table based on the value you specify in the filter condition. This property is applicable when you select object as the source type and enable lookup cache on the **Advanced** tab for the Lookup transformation. Maximum length is 32768 characters. For more information about this property, see *Transformations* in the Data Integration documentation.
[1]Doesn't apply to mappings in advanced mode.	

Enabling lookup caching

When you configure a Lookup transformation in a mapping, you can cache the lookup data during the runtime session.

When you select **Lookup Caching Enabled**, Data Integration queries the lookup source once and caches the values for use during the session, which can improve performance. You can specify the directory to store the cached lookup.

Lookup Cache Persistent

Use lookup cache persistent to save the lookup cache file to reuse it the next time Data Integration processes a Lookup transformation configured to use the cache.

You can specify the file name prefix to use with persistent lookup cache files in the **Cache File Name Prefix** field.

If the lookup table changes occasionally, you can enable the **Re-cache from Lookup Source** property to rebuild the lookup cache.

Dynamic Lookup Cache

Use a dynamic lookup cache to keep the lookup cache synchronized with the target. By default, the dynamic lookup cache is disabled and represents static cache.

If the cache is static, the data in the lookup cache does not change as the mapping task runs.

If the task uses the cache multiple times, the task uses the same data. If the cache is dynamic, the task updates the cache based on the actions in the task, so if the task uses the lookup multiple times, downstream transformations can use the updated data.

Pre-build Lookup Cache

Use a pre-build lookup cache to build the lookup cache before the Lookup transformation receives data. Multiple lookup cache files can be built at the same time to improve performance.

For information about lookup caching, see *Transformations* in the Data Integration documentation.

Optimize the staging performance for a mapping

Data Integration, by default, creates a flat file locally in a temporary folder to stage the data after reading from or before writing to Microsoft Azure Synapse SQL. You can set Data Integration to optimize the staging performance.

If you do not set the staging property, Data Integration performs staging without the optimized settings, which might impact the performance of the task.

Consider the following rules when you enable the staging property:

- When you read multiple objects, you cannot optimize the staging performance.
- If you run a mapping enabled for SQL ELT optimization, the mapping runs without SQL ELT optimization.
- If the data contains timestamp data types with time zone, the job runs without staging the data in the local flat file.
- If you configure a mapping in advanced mode, the mapping runs without staging the data in the local flat file.
- When you use a custom query or SQL override with the Order By clause and the `AzureSynapseDisableStagingForSort` property is set to `true`, the mapping runs without staging the data in the local flat file.
- When you read source data with decimal values and write to a Microsoft Azure Synapse SQL target and enable the staging property to optimize the staging performance, the decimal values in the target get truncated and rounded off to the nearest integer.
  To avoid this issue, in the system configuration details section, select the Type as DTM for the Data Integration service, and set the value for `DisableInfaDoubleHandlingForStaging` as yes.
- When you enable the staging optimization property for lookups configured to use persistent cache and define the cache file name prefix, the mapping fails at runtime for consecutive runs.
- When you use Parquet as the staging file format, you cannot use the staging property to optimize the staging performance and the mapping fails with validation errors.

Enabling the Microsoft Azure Synapse SQL Connector to optimize the staging performance

Perform the following tasks to enable the Microsoft Azure Synapse SQL Connector to optimize the staging performance:

1. In Administrator, click **Runtime Environments**.
2. Edit the Secure Agent for which you want to set the property.
3. In the **System Configuration Details** section, select the **Service** as **Data Integration Server** and the type as **Tomcat**.
4. Set the value of the Tomcat property to the plugin ID of the Microsoft Azure Synapse SQL connector. You can find the plugin ID in the manifest file located in the following directory: `<Secure Agent installation directory>/downloads/<Microsoft Azure Synapse SQL package>/CCIManifest`

   The following image shows the property set for the Secure Agent for the read, write, and lookup operations:

   To set the property for multiple connectors, specify the plugin ID for each connector separated by a comma.

When you run the mapping, the flat file is created in the following directory in your machine: `C:\Windows\Temp\AzureSynapse\stage\<AzureSynapse_Target.txt>`

You can check the session logs. If the flat file is created successfully, Data Integration logs the following message in the session log:

```
The INFA_DTM_STAGING is successfully enabled to use the flat file to create local staging files.
```

Process SQL queries and stored procedures using an SQL transformation

You can configure an SQL transformation to process SQL queries and stored procedures midstream in a Microsoft Azure Synapse SQL mapping. When you add an SQL transformation, you define the database connection and the type of SQL that the transformation processes.

You cannot configure an SQL transformation for mappings in advanced mode.

The SQL transformation can process the following types of SQL statements:

Stored procedure

You can configure an SQL transformation to call a stored procedure in Microsoft Azure Synapse SQL. The stored procedure must exist in the Microsoft Azure Synapse SQL database before you create the SQL transformation. When the SQL transformation processes a stored procedure, it passes input parameters to the stored procedure. The stored procedure returns the values to the output fields of the transformation.

SQL Query

You can configure an SQL transformation to process an entered query that you define in the SQL editor. The SQL transformation processes the query and returns the rows. The SQL transformation also returns any errors that occur from the underlying database or if there is an error in the user syntax.

Note: Saved query type is not applicable.

For more information about SQL queries and stored procedures, see *Transformations* in the Data Integration documentation.

Configuring an SQL transformation

Your mapping includes user IDs in the data flow. You want to include user names in addition to user IDs. You have a stored procedure that matches user IDs with user names in the database. You add an SQL transformation to your mapping, select the stored procedure, and map the userId incoming field with the userId input field in the stored procedure. Add an SQL transformation in a Microsoft Azure Synapse SQL mapping.

You check the Output Fields tab for the SQL transformation to confirm that it includes the username field. When you run the mapping, the username value is returned with the user ID.

This example lists the tasks required to configure an SQL transformation that calls a stored procedure in Microsoft Azure Synapse SQL:

1. Enter a name and description for the SQL transformation.
2. In the **Incoming Fields** tab, define field rules that determine the data to include in the transformation.
3. In the Properties panel of the SQL transformation, click the **SQL** tab.
4. In the **SQL** tab, perform the following tasks:
    a. Select the connection to the database. You can select the connection or use a parameter.
    b. Set the SQL type to **Stored Procedure**.
    c. Click **Select** to select the stored procedure from the database, or enter the exact name of the stored procedure to call.
    The stored procedure name is case-sensitive.

    Note: If you add a new stored procedure to the database while you have the mapping open, the new stored procedure does not appear in the list of available stored procedures. To refresh the list, close and reopen the mapping.
5. In the Field Mapping tab, specify how to map incoming fields to the input fields of the selected stored procedure.
6. Define advanced properties for the transformation according to your requirement.

To configure an SQL transformation using the SQL type as SQL entered query, see *Transformations* in the Data Integration help.

Rules and guidelines for SQL transformations

Consider the following rules and guidelines when you configure an SQL transformation in mappings:

- The RETURN_VALUE parameter is not applicable for Microsoft Azure Synapse SQL Connector.
- A stored procedure returns output fields that contain only integer values. For example, fields of int, bigint ,tinyint, smallint, and bit data type are returned as output fields.
- You cannot have input fields of datetimeoffset, binary, or varbinary data type in a stored procedure.
- When you configure a mapping to call a stored procedure in Microsoft Azure Synapse SQL, do not configure an override for the schema. The Secure Agent fails to fetch the metadata from the stored procedure when you specify an override to the schema name.
- You cannot read from a stored procedure whose name contains Unicode characters. The mapping fails with an SQL compilation error.
- Do not specify more than one SQL query in an SQL transformation. A mapping with multiple SQL queries fails.
- You cannot call a stored procedure from an entered query.
- If you want to parameterize the connection in an SQL transformation, perform the following steps:
    1. Select a valid non-parameterized connection for the SQL transformation.
    2. Configure the stored procedure or SQL query.
    3. Change the connection to a parameterized connection.
- When you use a parameterized connection for a SQL query, you cannot add new output fields.
- When the connection and the SQL query point to different schemas, the agent considers the schema specified in the SQL query.

Dynamic schema handling

You can choose how Data Integration handles changes that you make to the data object schemas. To refresh the schema every time the mapping task runs, you can enable dynamic schema handling in the task.

A schema change includes one or more of the following changes to the data object:

- Fields added, deleted, or renamed.
- Fields updated for data type, precision, or scale.

Configure schema change handling on the **Runtime Options** page when you configure the task.

The following table describes the schema change handling options:

Option	Description
Asynchronous	Default. Data Integration refreshes the schema when you edit the mapping or mapping task, and when Informatica Intelligent Cloud Services is upgraded.

Dynamic	Data Integration refreshes the schema every time the task runs.
	You can choose from the following options to refresh the schema:
	- **Alter and apply changes.** Data Integration applies the following changes from the source schema to the target schema:
	  - New fields: Alters the target schema and adds the new fields from the source.
	  - Renamed fields: Adds renamed fields as new columns in the target.
	  - Data type and precision updates. Applies these changes to the target.
	  - Deleted fields: Ignores deleted fields.
	- **Don't apply DDL changes.** Data Integration does not apply the schema changes to the target.
	- **Drop current and recreate.** Drops the existing target table and then recreates the target table at runtime using all the incoming metadata fields from the source.

For more information, see the "Schema change handling" topic in *Tasks* in the Data Integration help.

Rules and guidelines for dynamic schema handling

Consider the following rules and guidelines when you enable dynamic schema change handling:

Mappings

- When you enable dynamic schema change handling, you cannot make changes to the columns of Uniqueidentifier data type.
- When you read data from or write data to a Microsoft Azure Synapse SQL and override the schema name or table name, do not use the **Alter and apply changes** or **Drop current and recreate** option.
- When you alter the datetime data type to another, Data Integration always maps the altered field to the `datetime` field in the Microsoft Azure Synapse SQL target.
  For example, if you alter the source `datetime` field to `datetime2`, Data Integration maps the `datetime2` field to `datetime` in the Microsoft Azure Synapse SQL target.
- When you read data from Microsoft Azure Synapse SQL, any table or column constraints in the source table are not retained in the target.

Mappings in advanced mode

- You cannot use schema change handling when you use a custom query as the source type.
- When you use the **Drop current and recreate** option, special characters in the column name are written as underscore in the target.

IDENTITY column

You can use the IDENTITY column in mappings and mapping tasks.

Consider the following instructions when you use the IDENTITY column:

- You can use the IDENTITY column only for insert operation.
- Ensure that the metadata of the target object including the IDENTITY column matches the metadata fetched in an override.
- You cannot enable dynamic schema change handling for the IDENTITY column.
- You cannot define an IDENTITY column when you create a new target at runtime .
- The IDENTITY column is not highlighted in the imported objects.

Parameterization

You can parameterize the connection, objects, and the advanced runtime properties in mappings.

Consider the following rules and guidelines when you use parameterization:

General guidelines

- When you create a mapping with a parameterized target that you want to create at runtime, set the target field mapping to automatic.
- You cannot parameterize the field mapping.

Mappings

When you use input parameters, specify the parameter name in the following format:

- When you use input parameters, specify the parameter name in the following format:
  - Format in a mapping task: $name$
  - Format in a parameter file: $name
- When you use in-out parameters, specify the parameter name in the following format in a mapping task or a parameter file: $$name.

Mappings in advanced mode

- When you use input parameters, specify the parameter name in the following format in a mapping task: $name$.
- When you create a target at runtime, you cannot parameterize the target object.
- You cannot use input parameters in a parameter file to parameterize the source filter, custom query, multiple objects, and advanced properties at runtime.
- Ensure that you create the parameter before you override it using a parameter file.
- You cannot parameterize only the connection. You can either parameterize both the connection and the object or only the object.
- When you parameterize the object, ensure that the metadata of the imported object matches the metadata fetched in the override.
- When you use the parameter file to parameterize the source or target objects, you must specify the absolute path of the objects.

For example, `$$SrcObj=<table name>;`

- When you parameterize the connection or object, ensure that the connection or object that you want to override exists in Microsoft Azure Synapse SQL.

Rules and guidelines for mappings and mapping tasks

Consider the following rules and guidelines for mappings and mapping tasks:

General rules and guidelines

- When you change the storage type in a Microsoft Azure Synapse SQL connection in an existing mapping, you must either refresh that mapping or create a new mapping to display the source and target properties for the updated storage type.
- When you use Microsoft Azure Data Lake Storage Gen2 or Microsoft Azure Blob Storage to stage files in Microsoft Azure Synapse SQL mapping, consider the following guidelines:
  - You can specify the Microsoft Azure Data Lake Storage Gen2 file system name and Microsoft Azure Blob Storage container name at the connection level for all mappings that use the Microsoft Azure Synapse SQL connection. You can override file system name and container name specified in the connection from the advanced source or target properties.
  - You must specify the file system name and container name either in the connection or in the advanced source or target properties. If you specify at both the places, the Secure Agent considers values that you specify in the advanced properties.
- When you use Microsoft Azure Data Lake Storage Gen2 to stage files in Microsoft Azure Synapse SQL mapping, consider the following guidelines:
  - You can use managed identity authentication for Microsoft Azure Data Lake Storage Gen2 to stage files only when you run a mapping to read data from a Microsoft Azure Synapse SQL source or Microsoft Azure Data Lake Storage Gen2 source.
  - If the object ID for the system assigned identity is reset in the Azure portal, you must delete and recreate the user.
- You can use role-based access control or access control lists to authenticate access to the storage account for Microsoft Azure Data Lake Storage Gen2. Consider the following rules and guidelines when you use managed identity authentication with role-based access control or access control lists:
  - You cannot use both role-based access control and access control lists for the same storage account.
  - When you use access control lists, assign the read, write, and execute permissions to Microsoft Azure Synapse SQL to access the file system.
  - Assign read, write, and execute permissions to Microsoft SQL Server to access the file system.
  - If you enable system assigned identity, assign the required role or permissions to the Azure virtual machine to run the mappings and tasks.
    If you enable user assigned identity, assign the required role or permissions to the Azure virtual machine and the user assigned identity to run the mappings and tasks.
- When you connect to a Microsoft Azure Synapse SQL case-sensitive database and if a mapping uses multiple sources or targets, ensure that the sources or targets do not have the same name.

- When you specify the external data source, the container or file system used by the external data source is used for staging data. The container or file system specified in the connection properties or advanced properties is not considered for staging data.
- The session load summary in the session log is not captured for each commit interval.
- When you use an ODBC connection to connect to Microsoft Azure Synapse SQL, do not use the MERGE statement in the pre-SQL or post-SQL query. Else, the mapping task fails.
- When the source fields are empty, the Secure Agent writes the empty values as NULL in the target.
- When you use the $$$SESSSTARTTIME variable in an SQL query, the variable returns the session start time as a string value.
  Use the following syntax to convert the string values to datetime:

  SELECT CAST(CONVERT(VARCHAR,SUBSTRING('$$$SESSSTARTTIME',0,20),113) as datetime)

  Use the following syntax to convert the string values to date:

  SELECT CAST(CONVERT(VARCHAR,SUBSTRING('$$$SESSSTARTTIME',0,20),113) as date)
- When you use a custom query, SQL override, or update override for Uniqueidentifier fields, you must convert the Uniqueidentifier data type to string data type. Else, the mapping fails.
  For example, when you use `Select * from uid.NEWID_TEST2;`, the mapping fails.

  Use the following SQL query:

  ```
  Select CONVERT(CHAR(36),ID) AS ID, TESTCOLUMN from uid.NEWID_TEST2;
  ```
- When you use a custom query for Time field and select **Parquet** as the staging file format, you must convert the Time data type to nchar data type. Else, the mapping fails.
  For example, `Select CONVERT(nchar(16),[TIME]) AS [TIME] FROM [AzureDW_DIatScale].[TimeNUUID];`
- When you run a mapping to write float or real data types, the source and target data don't match in some cases. The values written to the target are approximate and not the same as in the source. To write the exact values, map real and float data types with numeric or decimal data types.
- To use the IS NULL and IS NOT NULL operators in a source filter or an uncached lookup, set the following property as a JVM option under the DTM type in the Secure Agent properties:
  `-DENABLE_NULL_FLAG_FOR_UNCACHED_LOOKUP=true`
- When you define a Not Parameterized filter condition with IS NULL or IS NOT NULL operators, the mapping fails with the following error:

  ```
  - Operation failed: Invalid expression string for filter condition
  ```

  You must define an advanced filter condition to use IS NULL or IS NOT NULL operators to filter rows that contain null values.
- When you use pre-SQL and post-SQL queries in a Microsoft Azure Synapse SQL mapping and a runtime error is encountered, the mapping fails if you have configured the following property in the JVM options of the Secure Agent:

  ```
  - -DAzureSynapseFailOnPrePostSQLError=true
  ```

  If you don't configure this property, the mapping does not display an error and runs successfully.
- Use only non-UTF8 collation when you create a database in Microsoft Azure Synapse SQL.

Source transformation

- You cannot preview data for binary or varbinary columns for Microsoft Azure Synapse SQL objects in a mapping.

- When read a JSON file from a Microsoft Azure Data Lake Storage Gen2 source and write to a Microsoft Azure Synapse SQL target created at runtime, ensure that the source data does not have null values.
- When you read data from multiple Microsoft Azure Synapse SQL sources, ensure that the table names for the source object and the related object are not the same, regardless of the name case.
- When you configure a join to read from multiple objects where one or more tables have the same column name, the mapping fails.
- When you read external tables, ensure that two or more tables are not in the same location in Microsoft Azure Blob Storage or Microsoft Azure Data Lake Storage Gen2.
- When you read data from a source that contains varbinary columns, the Secure Agent reads the empty values from varbinary columns as NULL.
- Custom query as source
  - When you use the query source type in a mapping to read from multiple tables, and you configure a join for one or more tables that have the same column names, the mapping fails.
    For example, see the following SQL query that involves a full outer join between two tables EMPLOYEE and DEPARTMENT that are part of the SALES.PUBLIC schema, where two columns have the same name, CITY:

    SELECT EMP_ID, NAME, CITY, DEPT_ID, DEPT_NAME, CITY

    FROM SALES.PUBLIC.EMPLOYEE

    FULL OUTER JOIN SALES.PUBLIC.DEPARTMENT ON EMP_ID = DEPT_ID

    To distinguish the conflicting column names, add aliases that the database can refer to while joining the tables:

    SELECT e.EMP_ID, e.NAME, e.CITY as ecity, d.DEPT_ID, d.DEPT_NAME, d.CITY as dcity

    FROM SALES.PUBLIC.EMPLOYEE e

    FULL OUTER JOIN SALES.PUBLIC.DEPARTMENT d ON e.EMP_ID = d.DEPT_ID
  - If the query starts with the WITH clause, ensure that there is no semicolon (;) before WITH.
    For example, `;With TEMP_OBJ AS (Select 'abc' AS FIELD1,123 AS Field2) Select Field1,Field2 from Temp_OBJ` is not a valid query.
  - Do not use a DECLARE statement in a source custom query.
  - The parameters in the query must have a default value that can be overridden at runtime.
  - When you use a custom query to read from multiple tables that have similar column names, you must use aliases for such column names while joining the tables.
  - When you read data from Microsoft Azure Synapse SQL, you must configure the following property in the JVM options of the Secure Agent:
    `-DAzureSynapseDisableStagingForSort=true`
  - When you write data to Microsoft Azure Synapse SQL, you cannot use the ORDER BY clause.
  - If you use a custom query in a Lookup transformation, you must perform the following tasks:
    - Configure the `-DAzureSynapseDisableStagingForSort=true` property in the JVM options of the Secure Agent.
    - Enable the **Sorted Input** option in the advanced lookup properties.
    - The query must first contain the ORDER BY clause for the condition fields are included in the lookup condition in the ascending order and then the other fields in any desired order.
    - When you configure the Lookup transformation to return multiple rows, the **Return All Rows** option does not return the values in the expected order.

Target transformation

- Behavior of target operations

Operation property	Treat Source Rows As property	Behavior
Insert, delete, or update	None or data driven	Value of the Operation property is given precedence.
Upsert	None or data driven	Update operation is performed. In this case, you must set both the Operation property and the Treat Source Rows As property to Upsert to perform an upsert operation.
Insert, delete, update, or upsert	Insert, delete, update, or upsert	Value of Treat Source Rows As property is given precedence.

- When you use the copy command, adhere to the following rules:
  - When you stage files in Microsoft Azure Data Lake Gen2, you can use the copy command only with Service Principal Authentication.
  - You cannot completely parameterize a multi-line copy command using a parameter file.
- When you create a Microsoft Azure Synapse SQL target at run time, ensure that the field names in the source do not contain special characters.
- When you create a target at runtime, the fields of the Uniqueidentifier data type in the source are converted to Nvarchar data type in the target.
  To retain the Uniqueidentifier data type, edit the metadata for the target fields and change the native type of the fields to Uniqueidentifier.
- When you write data to a Microsoft Azure Synapse SQL target and if the source table has columns of the Datetime or Datetime2 data type, the seconds value is rounded to the nearest value.
- When you write data to Microsoft Azure Synapse SQL and override the default update SQL statement, ensure that the parameter query does not exceed 600 characters.
- When you upsert or update data to a Microsoft Azure Synapse SQL target and multiple rows in the source table contain the same value for the column on which the primary key is defined, the first row with the value is correctly updated but the subsequent rows with the same value are incorrectly updated in the target.
- When you use an ODBC connection to connect to Microsoft Azure Synapse SQL and the target table has an IDENTITY column, you cannot update data in the target.
- When you run a Microsoft Azure Synapse SQL mapping that reads data with multiple non-null columns and all the non-null columns in the source are not mapped in the target, the mapping fails with the following error:

  ```
  [FATAL] Exception:
  com.informatica.adapter.azuredw.metadata.adapter.InfaSynapseException: Non-null fields
  are not projected
  ```

  Ensure that all the non-null columns in the source are mapped in the target.
- When you use the Snappy compression format to write data to Microsoft Azure Synapse SQL, the mapping retains a **snappy-1.1.8****-libsnappyjava.so** file in the temp directory on the agent machine after it runs successfully.
- When the data driven condition contains only the DD_REJECT operation, the mapping runs without generating any query.

- When you create a target at runtime and the real data type in the source object is mapped to float data type in the target, certain junk characters are appended to the values in the target.
  For example, `9.999999E19` is written as `9.999999320799471E19` in the target.

Lookup transformation

- When you use an uncached lookup, you cannot look up data with unicode characters.
- When you specify a SQL override to override the default lookup query in a Microsoft Azure Synapse SQL source, ensure that you select **Lookup Caching Enabled** in the advanced properties of the lookup object.
- You can't use a lookup object as a query for uncached lookups.
- Guidelines for dynamic lookup cache
  - The Lookup transformation must be a connected transformation.
  - You can specify only an equal operator in a lookup condition.
  - You cannot use dynamic lookup cache in SQL ELT optimization.

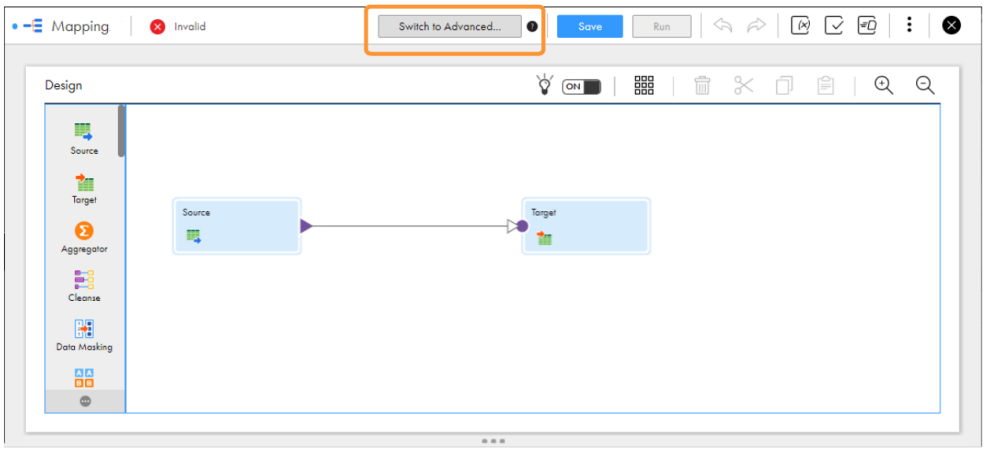
Mappings in advanced mode example

You work for an organization that stores large amount of purchase order details, such as customer ID, item codes, and item quantity for all regions in Microsoft Azure Data Lake Storage Gen2. You need to port the data from Microsoft Azure Data Lake Storage Gen2 to Microsoft Azure Synapse SQL to quickly analyze the purchase order details and to increase future revenues.

Create a mapping in advanced mode to achieve faster performance when you read all the purchase records from Microsoft Azure Data Lake Storage Gen2 source and write the records to a Microsoft Azure Synapse SQL target.

1. In Data Integration, click **New** > **Mappings** > **Mapping**.
2. In the Mapping Designer, click **Switch to Advanced**.

   The following image shows the **Switch to Advanced** button in the Mapping Designer:



3. In the **Switch to Advanced** dialog box, click **Switch to Advanced**.

   The Mapping Designer updates the mapping canvas to display the transformations and functions that are available in advanced mode.

4. Enter a name, location, and description for the mapping.
5. On the Source transformation, specify a name and description in the general properties.
6. On the **Source** tab, perform the following steps to provide the source details to read data from the Microsoft Azure Data Lake Storage Gen2 source:
    a. In the **Connection** field, select the Microsoft Azure Data Lake Storage Gen2 source connection.
    b. In the **Source Type** field, select the type of the source.
    c. In the **Object** field, select the required object.
    d. In the **Advanced Properties** section, provide the appropriate values.
7. On the **Fields** tab, map the source fields to the target fields.
8. On the Target transformation, specify a name and description in the general properties.
9. On the **Target** tab, perform the following steps to provide the target details to write data to theMicrosoft Azure Synapse SQL target:
    a. In the **Connection** field, select the Microsoft Azure Synapse SQL target connection.
    b. In the **Target Type** field, select the type of the target.
    c. In the **Object** field, select the required object.
    d. In the **Operation** field, select the required operation.
    e. In the **Advanced Properties** section, provide appropriate values for the advanced target properties.
10. Map the flat file source and the Microsoft Azure Synapse SQL target.
11. Click **Save** > **Run** to validate the mapping.

    In Monitor, you can monitor the status of the logs after you run the task.

Rules and guidelines for mappings in advanced mode

Consider the following rules and guidelines when you create mappings in advanced mode:

- When you configure a Sequence transformation in a mapping in advanced mode, the mapping fails.
- When you create a connection and specify an incorrect value for an Azure storage-specific connection property, the test connection is successful. However, the mapping in advanced mode fails with the following exception: `Caused by: org.apache.hadoop.fs.azure.AzureException: com.microsoft.azure.storage.StorageException: One of the request inputs is not valid` Ensure that you specify valid storage parameters when you configure the connection.
- When you select Microsoft Azure Data Lake Storage Gen2 as the storage type for both source and target connections, ensure that you use the same access key for both the source and target connections. For example, if you use Key 1 as the access key for the source connection and Key 2 for the target connection, the mapping fails.
- You cannot use system assigned identity for mappings in advanced mode.
- You cannot add multiple pipelines in a mapping.
- When you specify the external data source in the connection properties, you must specify the container or file system used to create the external data source either in the connection properties or in the advanced properties.
- When you run a mapping in advanced mode and map the source fields of double or float data type to the target fields of string data type, the format of the values changes in the target.

The following table describes the change in the format of the values in the target:

Value in the source	Value in the target
1.7976931348623157e+308	1.79769313486232e+308
-9999999999999.99	-10000000000000
4.94065645841247e-324	4.9e-324
7956318123.99392483	7956318123.99392

- When you run a mapping in advanced mode to write float or real data types, the source data and target data don't match in some cases. The values written to the target are approximate and not the same as in the source. To write the exact values, map real and float data types with numeric or decimal data types.

CHAPTER 4

Upgrading to Microsoft Azure Synapse SQL Connector

If you are accessing Microsoft Azure Synapse SQL using the Microsoft Azure Synapse SQL ODBC connection, Microsoft Azure SQL Data Warehouse V1 connection, or Microsoft Azure SQL Data Warehouse V2 connection, you can upgrade to the newer Microsoft Azure Synapse SQL connection.

If you switch the Microsoft Azure Synapse SQL ODBC connection to Microsoft Azure Synapse SQL connection, the object is not retained. You must re-import the Microsoft Azure Synapse SQL object.

The configured advanced source, target, and lookup properties in the fields that are common between the two connectors are retained in the new connector. You also have the option to retain the configured field mapping from the old mapping.

You can run the mapping successfully using the configured values from the old connector. You can additionally configure features that the enhanced Microsoft Azure Synapse SQL Connector offers.

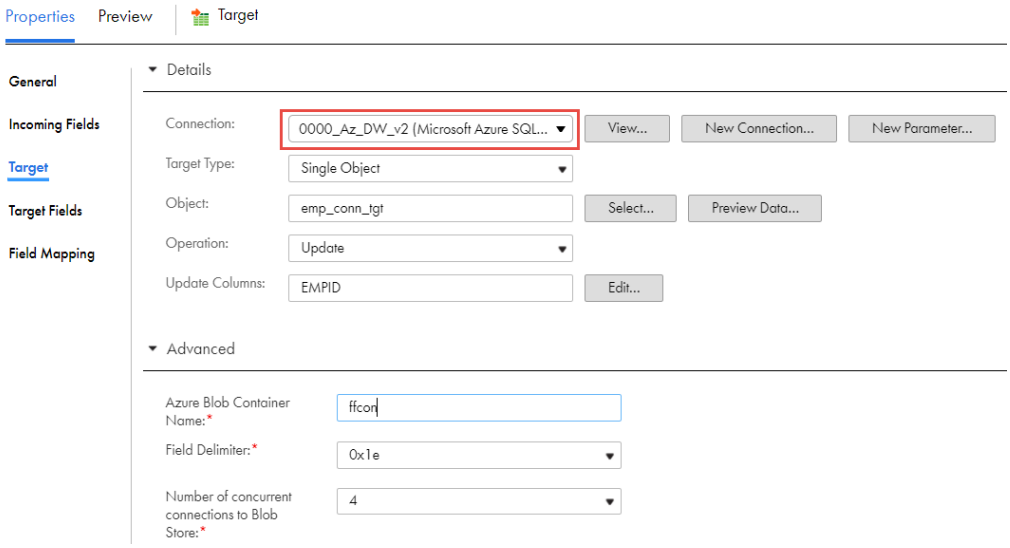
Note: If you are using the Microsoft Azure Synapse SQL ODBC connection, Microsoft Azure SQL Data Warehouse V1 connection, or Microsoft Azure SQL Data Warehouse V2 connection in mappings to read from or write data to Microsoft Azure Synapse SQL, Informatica recommends you to use the Microsoft Azure Synapse SQL connection to make use of the features that the enhanced connector offers.

Connection switching example

You want to upgrade your existing mapping that uses the Microsoft Azure SQL Data Warehouse V2 connection to use the Microsoft Azure Synapse SQL connection.

Before you switch the connection, perform the following steps:

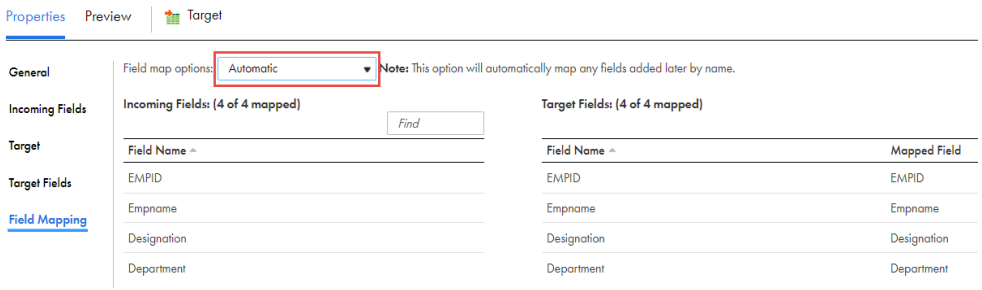
1. Open the existing Microsoft Azure SQL Data Warehouse V2 mapping that you want to upgrade to Microsoft Azure Synapse SQL.
   The following image shows an existing mapping that uses the Microsoft Azure SQL Data Warehouse V2 connection and contains the configured advanced properties in the Target transformation:



   The configured target object in this example is: `emp_conn_tgt`.

2. To retain the mapped fields from the field mapping when you switch the connection, on the **Field Mapping** tab, choose from the following **Field map options** menu in the Microsoft Azure SQL Data Warehouse V2 mapping:

   - To retain the fields automatically mapped after the switch, select **Automatic**.



   - To manually map the retained fields after the switch, select **Manual**.

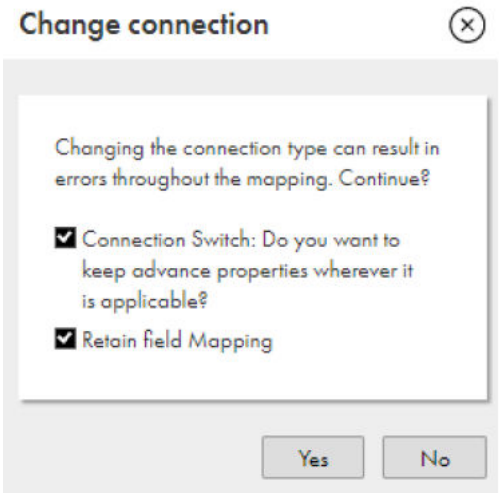
     Note: When you select manual, after switching the connection, you have the option to automap using the previous mapping.

3. To switch the connection, in the **Connection** field, change the connection from Microsoft Azure SQL Data Warehouse V2 to Microsoft Azure Synapse SQL.
4. In the **Change Connection** dialog box, select the following properties, and click **Yes**:

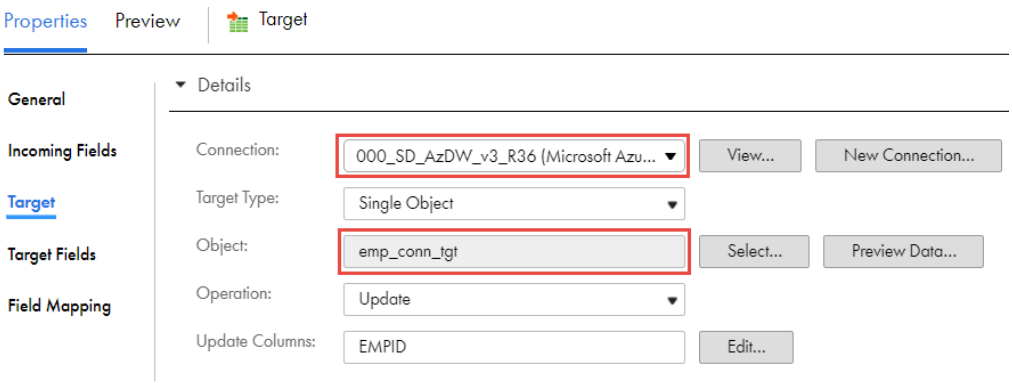
   - **Connection switch**. Switches to the connection that you select.

- **Retain field mapping**. Retains the configured field mappings from Microsoft Azure SQL Data Warehouse V2.

The following image shows the options that you must select:



5. The object from Microsoft Azure SQL Data Warehouse V2 is retained.
The following image shows the switched connection with the same object:



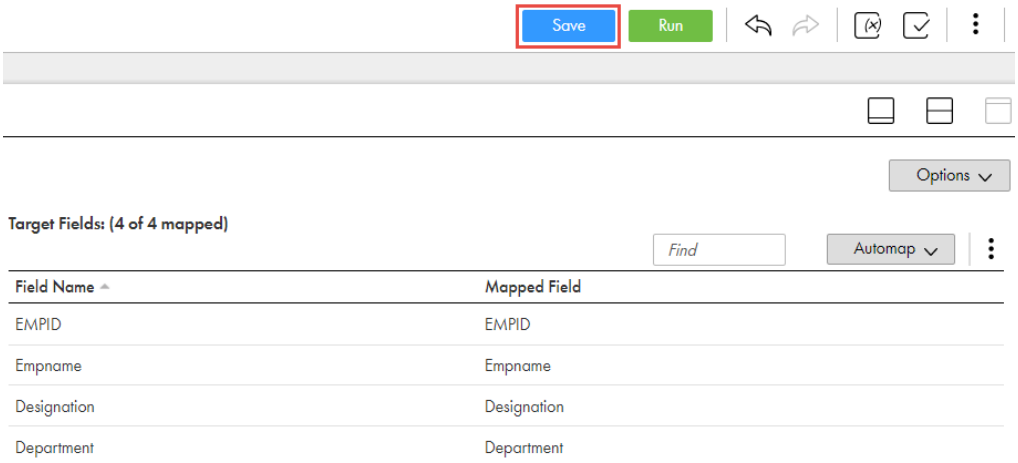
The configured target advanced properties from the Microsoft Azure SQL Data Warehouse V2 mapping reflect in the Target transformation.

6. If you had selected **Manual** on the **Field Mapping** tab in the Microsoft Azure SQL Data Warehouse V2 mapping and you want to reflect the field mappings in Microsoft Azure Synapse SQL, on the **Field Mapping** tab, select **Automap**, and then select **Previous Mapping**.

   Note: If you had selected **Automatic** in Microsoft Azure SQL Data Warehouse V2, you do not have to perform this task.

7. Click **Save**.

In the following image, the configured mapped fields from the Microsoft Azure SQL Data Warehouse V2 mapping now reflects in the Microsoft Azure Synapse SQL mapping:



8. Click **Run**.

Properties retained after the upgrade

When you replace the source or target connection type in existing mappings or mapping tasks that use the Microsoft Azure Synapse SQL ODBC connection, Microsoft Azure SQL Data Warehouse V1 connection, or Microsoft Azure SQL Data Warehouse V2 connection with the Microsoft Azure Synapse SQL connection, you have the option to retain the configured advanced properties and the field mappings.

The following table lists the configured advanced source, lookup, and target properties from the Microsoft Azure Synapse SQL ODBC connection that are retained in Microsoft Azure Synapse SQL mappings:

Properties	Microsoft Azure Synapse SQL ODBC properties retained in Microsoft Azure Synapse SQL
Source	- Pre SQL - Post SQL
lookup	- Pre SQL - Post SQL
Target	- Pre SQL - Post SQL

The following table lists the configured advanced source, lookup, and target properties from Microsoft Azure SQL Data Warehouse V1 retained in Microsoft Azure Synapse SQL mappings:

Properties	Microsoft Azure SQL Data Warehouse V1 properties retained in Microsoft Azure Synapse SQL
Source	- Azure Blob Container Name - Field Delimiter - Number of concurrent connection to Blob Store - Quote Character
Lookup	- Azure Blob Container Name - Field Delimiter - Number of concurrent connection to Blob Store - Quote Character
Target	- Azure Blob Container Name - Field Delimiter - Number of concurrent connection to Blob Store - Quote Character

The following table lists the configured advanced source, lookup, and target properties from Microsoft Azure SQL Data Warehouse V2 retained in Microsoft Azure Synapse SQL mappings:

Properties	Microsoft Azure SQL Data Warehouse V2 properties retained in Microsoft Azure Synapse SQL
Source	- Azure Blob Container Name - Field Delimiter - Number of concurrent connection to Blob Store - Pre SQL - Post SQL - SQL Override - Quote Character
Lookup	- Azure Blob Container Name - Field Delimiter - Number of concurrent connection to Blob Store - Pre SQL - Post SQL - SQL Override - Quote Character
Target	- Azure Blob Container Name - Field Delimiter - Number of concurrent connection to Blob Store - Pre SQL - Post SQL - SQL Override - Quote Character

Rules and guidelines

Consider the following rules when you switch the Microsoft Azure Synapse SQL ODBC connection, Microsoft Azure SQL Data Warehouse V1 connection, or Microsoft Azure SQL Data Warehouse V2 connection in mappings with Microsoft Azure Synapse SQL connection:

- When you switch the Microsoft Azure Synapse SQL ODBC connection with Microsoft Azure Synapse SQL connection, the field mapping is retained only if the **Field map options** was set to **Manual** in the previous mapping.
- If the source, lookup, or target connection is parameterized in the mapping, ensure that the **Connection Type** field in the input parameters is blank.
- If the source object is parameterized, the object is not retained when you switch the connection. You must re-import the object.
- If the existing mapping contains multiple objects, the relationship between these objects is not preserved when you switch the connection.
- If you do not enable the Multiple Matches option for the Lookup transformation in the existing mapping, after the switch the Multiple Matches option is set to **Return all rows**.
- The filter and sort values are not retained. You must manually add the sort and filter values after you upgrade to the new connection.

CHAPTER 5

Migrating a mapping

You can configure a connection and mapping in one environment and then migrate and run the mapping in another environment.

You can also migrate mappings configured in advanced mode. After the migration, you can change the connection properties from the Administrator service, but you do not need to modify the mapping. Data Integration uses the configured runtime attributes from the earlier environment to run the mapping successfully in the new environment.

Consider a scenario where you develop a mapping in the development organization (Org 1) and you then migrate and run the mapping in the production organization (Org 2). After you migrate, you might want to use the same or a different connection endpoint or object path in Org 2. Based on your requirement, follow the guidelines in this section before you plan the migration.

Use the same object path for the migrated mapping

If you want the migrated mapping in Org 2 to use the same object path as in Org 1, you must maintain the same schema and table in the Microsoft Azure Synapse SQL account for Org 2.

For example, if you have two different accounts, Account1 used for Org 1 and Account2 used for Org 2, the object path for the schema and table must be the same in both the accounts:

Account1: SCHEMA1/TABLE1

Account2: SCHEMA1/TABLE1

In this scenario, you do not need to override the schema and table in the advanced properties.

Use a different object path for the migrated mapping

After you migrate the mapping, you can use a different object path to run the mapping from the new environment. In this scenario, before you migrate the mapping, you can change the object metadata, runtime attributes, or the connection attributes to reflect the object path in the migrated environment. You do not have to edit or update the mapping in the new environment.

As a rule, when you specify the schema or table in the advanced properties, connection, or object properties, Data Integration honors the attributes in the following order of precedence:

1. **Runtime advanced attribute**s. The advanced properties such as schema and table in the Source, Target, or Lookup transformation in a mapping.
2. **Connection attributes**. Attributes such as database and schema set in the connection properties.
3. **Object metadata**. The object selected in the Source, Target, or Lookup transformation in a mapping.

When you migrate a mapping, ensure that the metadata of the imported object matches the metadata fetched in an override.

Migration options

When you migrate, you can choose from one of the following options to update the object path:

Option 1. Update the connection properties to reference the new object

When you import the mapping into Org 2, in the **Review Connections** section, you can change the existing connection to map to the connection that has access to the schema and table in Org 2.

Option 2. Override the properties from the advanced properties

Before the migration, specify the required schema and table name for the object from Org 2 in the advanced properties of the Org 1 mapping.

After the migration, when you run the mapping, the Secure Agent uses the configured advanced parameters to override the object specified in the mapping imported from Org 1.

Option 3. Parameterize the properties in the mapping

You can choose to parameterize the advanced attributes, such as the schema and table name before the migration. You can configure input parameters, in-out parameters, and parameter files in the mapping. After you migrate the mapping, do not edit or update the mapping. If you have used in-out parameters for the advanced attributes such as for the schema and table name, you can update these from the parameter file.

Parameterizing only the advanced properties, but not the object or connection in the mapping

You can choose to parameterize only the advanced properties before the migration, and use them at runtime. After the migration, when you run the mapping, the Secure Agent uses the value of the parameters specified in the advanced properties.

Parameterizing both the advanced properties and the object or connection

When you parameterize both the advanced properties and the object or connection, you can choose to select or deselect the Allow parameter to be overridden at runtime option.

Allow parameter to be overridden at runtime option is selected

When you parameterize both the advanced properties and the object or connection and select the **Allow parameter to be overridden at runtime** option before the migration, you must select a placeholder object in the object properties in the mapping and then specify an override to this placeholder object from the advanced properties. Ensure that the placeholder object contains the same metadata as the corresponding table that you specify as an override. When you run the mapping, the value specified in the advanced property overrides the placeholder object.

Allow parameter to be overridden at runtime option is not selected

When you parameterize both the advanced properties and the object or connection and do not select the **Allow parameter to be overridden at runtime** option before the migration, you do not need a placeholder object and the values specified in the advanced properties take precedence.

Parameterization rules

Consider the following rules to parameterize the object and advanced properties:

- Parameterization is not applicable when you create a new target at runtime and select the Allow parameter to be overridden at runtime option in a mapping.
- Parameterization is not applicable for mappings in advanced mode in the migration use case.
- If there are multiple pipelines configured in a mapping, do not parameterize the Microsoft Azure Synapse SQL object. You must select a placeholder object while creating the mapping before you migrate.

General rules and guidelines

Applicable object types, transformations, and fields

The following table lists the transformation, object type, and the fields in the advanced properties of a mapping that you can retain when you migrate to the new environment:

Transformations	Object type
Source	Single object, multiple objects, query
Lookup	Single object, query Applicable for unconnected, and connected cached and uncached.
Target	Single object
Note: Advanced fields such as the schema and table name are supported for the Source, Lookup, and Target transformations.	

Multiple sources

Consider the following guidelines:

- If the mapping contains multiple sources, you can override the schema in the advanced properties only if the schema selected in the source objects is the same, as in the following example:
  object1: `schema1.object1`

  object2: `schema1.object2`

  If the schema is different, for example, if the schema for object1 is `schema1.object1` and object2 is `schema2.object2`, migration does not work.
- Do not override the table from the advanced properties if the mapping contains multiple objects.

Advanced filter and table name override combination

If the mapping contains an advanced filter for the object, for example, <Table2>, in the query options section and an override to the table name in the advanced properties, consider the following rules to override the table:

In mappings, consider one of the following options:

- The advanced filter in the mapping must include the `$$Filtercondition` value. In the mapping task, you can override the value from the parameter file in the task properties. The parameter condition in the parameter file must include the following condition:
  `$$Filtercondition =<Table2>.id >= 1`
- Use the table name that you specify as an override in the advanced properties directly in the advanced filter condition. For example, `<Table2>.id >= 1`

When you create mappings in advanced mode, use the table name specified as an override in the advanced properties directly in the advanced filter condition. For example, `<Table2>.id >= 1`

SQL override

In the mapping, do not specify an SQL override when the object type is a single object. A mapping with this combination fails.

Table name override

In the mapping, do not specify a table name override when the object type is a custom query. A mapping with this combination fails.

Schema change handling

You cannot dynamically refresh the data object schema at runtime. You must maintain the same metadata for the table selected in the source, target, or lookup transformations and the corresponding advanced field overrides, as schema change handling is not applicable.

Update override

When you write data to Microsoft Azure Synapse SQL and specify an update override for the target object, after you migrate the mapping, ensure that the target object specified in the override exists in the mapping.

CHAPTER 6

SQL ELT optimization

You can use SQL ELT optimization to push the transformation logic to Microsoft Azure Synapse SQL.

Use SQL ELT optimization to improve task performance by using the database resources. When you run a task configured for SQL ELT optimization, the task converts the transformation logic to a PolyBase query or SQL query. The task sends the query to the database, and the database executes the query.

Configure SQL ELT optimization for a mapping in the tasks properties. You can configure SQL ELT optimizationin task that references a mapping or a mapping in advanced mode.

If your mapping contains multiple pipelines, you can define the flow run order to load the targets from the pipelines in a particular order.

SQL ELT optimization types

When you apply SQL ELT optimization, the task pushes transformation logic to the source or target database based on the optimization type you specify in the task properties.

You can configure the following types of SQL ELT optimization:

None

The task does not push down the transformation logic to the Microsoft Azure Synapse SQL database.

Full

The task pushes as much of the transformation logic as possible to process in the target database.

Source

The task pushes down as much as the transformation logic as possible to process in the source database. When you select source SQL ELT optimization, the task pushes the transformation logic for all the supported transformations downstream in the mapping.

Note: Doesn't apply to mappings in advanced mode.

Data Integration behavior with source and full SQL ELT optimization

When you select full or source SQL ELT optimization for a mapping that reads from or writes to Microsoft Azure Synapse SQL, Data Integration analyzes the mapping from the source to the target or until it reaches a downstream transformation to determine whether to push down only partial or the entire mapping logic for processing to the database.

If all the transformations in the mapping are supported for SQL ELT optimization, the task pushes down the entire mapping logic to the database.

When a transformation is not supported in the mapping, the task partially pushes down the mapping logic to the point where the transformation is supported for SQL ELT optimization. Data Integration generates and executes a SELECT statement for the transformation logic that needs to be pushed down. Then, it reads the results of this SQL query and processes the remaining transformations in the mapping.

SQL ELT optimization scenarios

You can configure SQL ELT optimization using the Microsoft Azure Synapse SQL Connector in mappings and mappings in advanced mode.

You can use configure SQL ELT optimization for the following scenarios:

Source and target endpoints	Supported SQL ELT scenarios in mappings
Microsoft Azure Synapse SQL source Microsoft Azure Synapse SQL target	Reads from and writes to Microsoft Azure Synapse SQL using the Microsoft Azure Synapse SQL connection. SQL ELT optimization type is Source or Full. Source SQL ELT optimization is not applicable to mappings in advanced mode.
Microsoft Azure Data Lake Storage Gen2 source Microsoft Azure Synapse SQL target	Reads from Microsoft Azure Data Lake Storage Gen2 using a Microsoft Azure Data Lake Storage Gen2 connection and writes to Microsoft Azure Synapse SQL using a Microsoft Azure Synapse SQL connection. SQL ELT optimization type is Full. Doesn't apply to mappings in advanced mode.
Note: You can use the Secure Agent or the Hosted Agent to run mappings enabled with SQL ELT optimization.	

Previewing SQL ELT optimization

Before you can run a mapping task configured for SQL ELT optimization, you can preview if SQL ELT optimization is possible when you create the mapping. You can preview SQL ELT optimization from the SQL ELT optimization panel in the Mapping Designer.

After you select the required SQL ELT optimization options and run the preview, Data Integration creates and runs a temporary SQL ELT preview mapping task. When the job completes, Data Integration displays the SQL queries to be executed and any warnings in the SQL ELT Optimization panel. The warning messages help you understand which transformations in the configured mapping are not applicable for SQL ELT optimization. If SQL ELT optimization fails, Data Integration lists any queries generated up to the point of failure. You can edit the mapping and fix the required transformations before you run the mapping for SQL ELT optimization.

You can also view the temporary job created under **My Jobs** and download the session log to view the queries generated.

For more information about how to preview SQL ELT optimization, see the topic "SQL ELT optimization preview" in *Mappings* in the Data Integration help.

Note: Doesn't apply to mappings in advanced mode.

Configuring SQL ELT optimization

To optimize a mapping, add the mapping to a task, and then configure SQL ELT optimization in the mapping task.

1. Create a mapping task.
2. In the SQL ELT Optimization section on the Runtime Options tab, set the SQL ELT optimization value to **Full** or **To Source**.

   You can use Source SQL ELT optimization only when you read from a Microsoft Azure Synapse SQL source.

   Note: The Optimization context type option is not applicable for a Microsoft Azure Synapse SQL mapping task.
3. If full SQL ELT optimization is not available, select how Data Integration handles SQL ELT optimization in the **SQL ELT Optimization Fallback Option menu**:
   - Partial SQL ELT optimization. Default. Data Integration pushes as much transformation logic as possible to the source and target database. The task processes any transformation logic that it can't push to a database. You can use Partial SQL ELT optimization only when you read from and write to Microsoft Azure Synapse SQL.
   - Non SQL ELT optimization. The task runs without SQL ELT optimization.
   - Fail Task. Data Integration fails the task.

   Note: The fallback options are not applicable to mappings in advanced mode.

When you run the mapping task, the transformation logic is pushed to the Microsoft Azure Synapse SQL. To verify that the SQL ELT optimization has taken place, you can check the session log for the job.
You can monitor the jobs that you initiated on the **My Jobs** page. You choose to stop or clean stop a job. When you use clean stop, Data Integration terminates all the issued statements and processes spawned by the job. This is not applicable to mappings in advanced mode.

SQL ELT optimization using a Microsoft Azure Synapse SQL connection

You can configure SQL ELT optimization for a mapping that contains a Microsoft Azure Synapse SQL connection. SQL ELT optimization enhances the mapping performance.

Use the **SQL ELT optimization** advanced session property to configure SQL ELT optimization in a mapping task.

When you run a task that reads from Microsoft Azure Data Lake Storage Gen2 and writes to Microsoft Azure Synapse SQL with SQL ELT optimization, the task converts the transformation logic to a PolyBase query.

When you can configure SQL ELT optimization in a mapping to read from and write to Microsoft Azure Synapse SQL using a Microsoft Azure Synapse SQL connection, you can read data from tables, external tables, custom queries, and views.

Functions with Microsoft Azure Synapse SQL

You can push functions in a transformation to Microsoft Azure Synapse SQL using SQL ELT optimization.

When you use SQL ELT optimization, the Secure Agent converts the expression, filter, or aggregator in the transformation by determining equivalent functions in the database.

The following table summarizes the availability of SQL ELT functions when you configure SQL ELT optimization for a mapping that contains a Microsoft Azure Synapse SQL connection.

Function	Function	Function
ABS()	LAST_DAY()	SIGN()
ADD_TO_DATE()	LENGTH()	SIN()
ASCII()	LN()	SINH()
AVG()	LOG()	SOUNDEX()
CEIL()	LOWER()	SQRT()
CHR()	LPAD()	STDDEV()
CONCAT()	LTRIM()	SUBSTR()
COS()	MAKE_DATE_TIME()	SUM()
COUNT()	MAX()	SYSDATE()
DATE_DIFF()	MD5()	SYSTIMESTAMP()
DECODE()	MIN()	TAN()
EXP()	MOD()	TANH()
FIRST()[1]	POWER()	TO_BIGINT()
FLOOR()	REG_MATCH()	TO_CHAR(DATE)
GET_DATE_PART()	REPLACECHR()	TO_CHAR(NUMBER)
IIF()	REPLACESTR()	TO_DATE()
IN()	ROUND(DATES)	TO_DECIMAL()
INSTR()	ROUND(NUMBER)	TO_FLOAT()
IS_DATE()	RPAD()	TO_INTEGER()
IS_NUMBER()	RTRIM()	TRUNC(DATE)
IS_SPACES()	SESSSTARTTIME()[1]	TRUNC(NUMBER)
ISNULL()	SET_DATE_PART()	UPPER()

Function	Function	Function
LAST()[1]	SHA-256 ()[1]	VARIANCE()
[1]Doesn't apply to mappings in advanced mode.		

The following table describes the syntax for the SQL ELT functions:

Function	Syntax	Description
ADD_TO_DATE	`ADD_TO_DATE (date, format, amount)`	Supported formats: - yyyy - mm - dd - hh - mi - ss - ms - us - ns
DATE_DIFF	`DATE_DIFF (date1, date2, format)`	Supported formats: - yyyy - mm - dd - hh - mi - ss - ms - us - ns
DECODE	`value , first_search , first_result [, second_search , second_result ]...[, default ]`	Value can contain any datatype except Binary. You can enter any valid expression. You cannot use true or false as values. You cannot add conditions to the search argument. For example, `DECODE ( CONST_NAME, 'Five', 5, 'Pythagoras', 1.414213562, 'Archimedes', 3.141592654, 'Pi', 3.141592654 )`
FIRST	`FIRST (value)`	

Function	Syntax	Description
GET_DATE_PART	`GET_DATE_PART (date, format)`	Supported formats: - `yyyy` - `mm` - `dd` - `hh` - `mi` - `ss` - `ms` - `us` - `ns`
IIF	`IIF (condition, value1 [,value2])`	

IN	- Aggregator and Expression transformation
	`IIF(IN(search_value, value1, [value2, ..., valueN,]), 'TRUE', 'FALSE')`
	- Filter transformation
	`IN(search_value, value1, [value2, ..., valueN,])`	

INSTR	`INSTR (string, search_value)` `INSTR (string, search_value [,start [,occurrence]])`	When you use the `INSTR (string, search_value)` syntax, the search value must be enclosed in %. When you use the `INSTR (string, search_value [,start [,occurrence]])` syntax, the search value must be a character expression that contains the sequence that you want to search. When you use the default values in start and occurrence parameters, the search value must be enclosed in %. Default value for start and occurrence parameters is 1.

Function	Syntax	Description

IS_DATE	IS_DATE(value [,format])
	- Aggregator and Expression transformation
	  - IS_DATE(column name)
	  - IIF(IS_DATE(colname), value1 [,value2])
	  - IIF(IS_DATE(colname)=0 or 1, value1 [,value2])
	- Filter and Router transformation
	  - IS_DATE(column name)=0 or 1
	  - IIF (IS_DATE(colname), 0 or 1, 0 or 1)
	  - IIF(IS_DATE(column name)=0 or 1, 0 or 1, 0 or 1)
		Supported formats:
		- dd mon yyyy
		- hh:mi:ss
		- mm/dd/yyyy
		- mm/dd/yyyy hh24:mi:ss
		- yyyy
		- yyyy-mm-dd
		- yyyy-mm-dd hh:mi:ss
		- yyyy.mm.dd
		- yyyy/mm/dd
		- yyyymmdd
		- mon dd yyyy hh:miAM
		- Mon dd, yyyy
		- mon dd yyyy hh:mi:ss:mmmAM
		- mm-dd-yyyy
		- dd mon yyyy hh:mi:ss:mmm
		- yyyy-mm-dd hh:mi:ss.mmm
		- mm/dd/yy hh:mi:ss AM
		- yyyy-mm-ddThh:mi:ss.mmm
		- yyyy-MM-ddThh:mm:ss.fffZ
		- dd mon yyyy hh:mi:ss:mmmAM
		Date range: 1753-01-01 to 9999-12-31.

ISNULL	- Aggregator and Expression transformation
	  - IIF(ISNULL(colname), value1 [,value2])
	  - IIF(ISNULL(colname)=0 or 1, value1 [,value2])
	- Filter and Router transformation
	  - ISNULL(column_name)
	  - IIF(ISNULL(column name), 0 or 1, 0 or 1)	

IS_NUMBER	- Aggregator and Expression transformation
	  - IS_NUMBER(column name)
	  - IIF(IS_NUMBER(colname), value1 [,value2])
	  - IIF(IS_NUMBER(colname)=0 or 1, value1 [,value2])
	- Filter and Router transformation
	  - IS_NUMBER(column name)=0 or 1
	  - IIF(IS_NUMBER(column name), 0 or 1, 0 or 1)
	  - IIF(IS_NUMBER(column name)=0 or 1, 0 or 1, 0 or 1)	

Function	Syntax	Description

IS_SPACES	- Aggregator and Expression transformation
	  - `IS_SPACES(column name)`
	  - `IIF(IS_SPACES(colname), value1 [,value2])`
	  - `IIF(IS_SPACES(colname)=0 or 1, value1 [,value2]`
	- Filter and Router transformation
	  - `IS_SPACES(colname) = 0 or 1`
	  - `IIF (IS_SPACES(colname), 0 or 1, 0 or 1)`
	  - `IIF(IS_SPACES(column name)=0 or 1, 0 or 1, 0 or 1)`	

LAST	`LAST (value)`	
LAST_DAY	`LAST_DAY (date)`	
LTRIM and RTRIM	`LTRIM (string)` `RTRIM (string)`	You can pass only a single argument in the LTRIM or RTRIM function.
REG_MATCH	`REG_MATCH(subject, pattern)`	Supported regular expression syntax used in pattern: - `.` - `[^a-z]` - `[a-zA-Z0-9]` - `\d` - `\s` - `\w`
ROUND(DATES)	`ROUND( date [, format ] )`	Supported formats: - `yy` - `mm` - `dd` - `hh` - `mi` - `ss`
TANH	Use the `EXP()` function in the following formula to calculate `TANH`: `TANH(x) = (EXP(2 * x) - 1) / (EXP(2 * x) + 1)`	Maximum value of x: 354 Minimum value of x: -354 If the value of x is greater than 354 or less than -354, use the follwing expression: `CASE WHEN X > 354 THEN 1 WHEN X < -354 THEN -1 ELSE TANH(X) END`
TO_BIGINT	`TO_BIGINT (numeric expression [, flag ])` TO_BIGINT truncates the decimal portion when the flag is TRUE or a number other than 0. TO_BIGINT rounds the value to the nearest integer if the flag is FALSE or 0 or if you omit this argument.	You can pass only a single argument in the function.

Function	Syntax	Description
TO_CHAR (Date)	`TO_CHAR (date [,format])` Specify a supported format. The `format` defines the format of the return value, not the format for the values in the date argument.	Supported formats: - `mm.yyyy` - `dd/mm/yyyy` - `dd.mm.yyyy` - `dd-mm-yyyy` - `dd mon yyyy` - `mm/dd/yyyy` - `hh:mi:ss` - `yyyy.mm.dd` - `yyyy/mm/dd` - `yyyymmdd` - `yyyy-mm-dd hh:mi:ss` - `mm/dd/yyyy hh24:mi:ss`
TO_DATE	`TO_DATE (string [,format])` The `format` must match the parts of the `string` argument.	Supported formats: - `dd/mm/yyyy` - `dd.mm.yyyy` - `dd-mm-yyyy` - `dd mon yyyy` - `mm/dd/yyyy` - `hh:mi:ss` - `yyyy` - `yyyymm` - `yyyymmdd` - `yyyy-mm-dd` - `yyyy.mm.dd` - `yyyy/mm/dd` - `yyyy-mm-dd hh:mi:ss`
SUBSTR	`SUBSTR (string, start [,length ])`	The `start` argument must be a positive number. The length must be an integer greater than 0.

Operators with Microsoft Azure Synapse SQL

When you use SQL ELT optimization, the Secure Agent converts the expression in the transformation by determining equivalent operators in the database. If there is no equivalent operator, the Secure Agent processes the transformation logic.

The following table summarizes the availability of SQL ELT operators when you configure SQL ELT optimization for a mapping that contains a Microsoft Azure Synapse SQL connection.

Operator
+ - * /
%
= > < >= <= <>
!=
NOT AND OR

Transformations with Microsoft Azure Synapse SQL

When you configure SQL ELT optimization, Data Integration tries to push the configured transformation to Microsoft Azure Synapse SQL.

You can push the following transformations to Microsoft Azure Synapse SQL:

- Aggregator
- Expression
- Filter
- Joiner
- Lookup*
- Normalizer*
- Rank*
- Router
- SQL*
- Union

*Doesn't apply to mappings in advanced mode.

Aggregator transformation

You can configure full SQL ELT optimization to push an Aggregator transformation to process in Microsoft Azure Synapse SQL .

Aggregate calculations

You can perform the following aggregate calculations:

- AVG
- COUNT
- MAX
- MEDIAN
- MIN
- SUM
- STDDEV
- VARIANCE

Incoming ports

When you configure an Aggregator transformation and the incoming port is not used in an aggregate function or in a group by field in a mapping in advanced mode, a dummy MAX() function is generated in the query and the mapping runs with full SQL ELT optimization.

Expression transformation

You can configure full SQL ELT optimization to push an Expression transformation to process in Microsoft Azure Synapse SQL.

You can add an Expression transformation to each of the sources in the mapping, followed by a join downstream in the mapping. Additionally, you can add multiple Expression transformations that branch out from a transformation and then branch in into a transformation downstream in the mapping.

When you configure an Expression transformation, consider the following rules to include variables in the expression:

- You cannot use variables where you are using the value assigned while processing a previous row for calculations in the current row. if you do, the mapping runs without SQL ELT optimization.
- The variables can be nested, but you cannot refer to a variable before it is defined in the expression. If the variables are not defined in that order, the mapping runs without SQL ELT optimization.

  For example,

  ```
   var: AGEPLUS2 = AGEPLUS1 + 1
  var: AGEPLUS1 = AGE + 1
  out: NEXTAGE = AGEPLUS2 + 1
  ```

  Here, AGE +1 is defined later. AGEPLUS2 in the first variable refers to AGEPLUS1 and remains unresolved. To resolve this, specify the variables in the following order:

  ```
   var: AGEPLUS1 = AGE + 1
  var: AGEPLUS2 = AGEPLUS1 + 1
  out: NEXTAGE = AGEPLUS2 + 1
  ```

- The variables cannot have an expression that is cyclic or refers to itself:
  For example,

  ```
   var: AGEPLUS1 = AGEPLUS2 + 1
  var: AGEPLUS2 = AGEPLUS1 + 1
  out: NEXTAGE= AGEPLUS2
  ```

  Here, AGEPLUS1 refers to AGEPLUS2 and remains unresolved.

Lookup transformation

You can configure full SQL ELT optimization to push a Lookup transformation to process in Microsoft Azure Synapse SQL. You can push both a connected and an unconnected lookup.

Consider the following rules when you configure lookups:

- When you look up data and the lookup condition finds multiple matches, the lookup returns all rows. In a mapping with Microsoft Azure Synapse SQL as target, set the **Multiple Matches** option for the lookup object to **Return all rows**. If you enabled **Multiple Matches** to any option other than **Return all rows**, the agent ignores it.
- When you lookup data with null values in a Microsoft Azure Synapse SQL source and write to the target, there is a difference in the number of records in the target for a mapping that runs with and without SQL ELT optimization.
- When you read from and write to Microsoft Azure Synapse SQL, you cannot use filters in a Lookup transformation.

SQL transformation

You can configure an SQL transformation to process SQL queries in a Microsoft Azure Synapse SQL mapping enabled for SQL ELT optimization.

Use functions to run queries

You can include functions in an entered query in an SQL transformation and run queries with the Microsoft Synapse SQL target endpoint. You must use only the SELECT clause SQL statement to push a function. Specify the column name in the select query or function.

Do not use FROM or WHERE clause in the SQL statement. For example, do not push functions using statements like "SELECT * FROM TABLE".

You can use the following functions in an entered query:

- CURRENT_USER
- HAS_DBACCESS
- RAND
- SESSION_USER
- SIGN
- SUSER_NAME
- SUSER_SNAME
- SYSTEM_USER
- USER
- USER_NAME

For more information about the supported functions, see the Microsoft Azure Synapse SQL documentation.

User defined functions

You can configure a custom query in an SQL transformation to read from SQL user-defined functions (UDF) in Microsoft Azure Synapse SQL. You cannot read UDFs that have newline characters in the UDF name.

Rules and guidelines

When you configure an SQL transformation, consider the following rules and guidelines:

- You cannot use an unconnected SQL transformation.
- You can use queries that are static. Dynamic queries are not applicable.
- You cannot use functions that return multiple rows.
- You cannot use the **Track the number of rows affected by this query** option for the output fields.
- If the mapping fails, the SQL error is not logged.
- You cannot use the **Max output row count** and **Continue on SQL Error within row** options in the advanced properties.

Supported features for Microsoft Azure Synapse SQL mappings

When you configure SQL ELT optimization, the mappings support the following Microsoft Azure Synapse SQL properties in the Source, Lookup, and Target transformations:

If you configure properties that are not supported, the mapping runs with SQL ELT optimization but the properties are ignored. If the SQL ELT query generation fails, the mapping runs without SQL ELT optimization.

Source

When you configure SQL ELT optimization, the mappings support the following properties for a Microsoft Azure Synapse SQL source:

- Source type - Single object, multiple objects, query, and parameter.
- Parameter - Allow source object to be overridden at run time
- Query options - Filter. Supports both simple and advanced filter conditions.
- Schema Name Override
- Table Name Override
- Pre-SQL

- Post-SQL
- SQL Override
- On Pre-Post SQL Error
- Edit Metadata

Target

When you configure SQL ELT optimization, the mappings support the following properties for a Microsoft Azure Synapse SQL target:

- Target type - Single object and parameter.
- Target object - Existing target, Create new at runtime.
- Operation - Insert, Update, Upsert, Delete, Data driven
- Azure Government endpoints
- Schema Name Override
- Table Name Override
- Truncate Table
- Pre-SQL
- Post-SQL
- On Pre-Post SQL Error
- Treat Source Rows As
- Reject Threshold. Applicable only when you read from a Microsoft Azure Data Lake Storage Gen2 source.
- Edit metadata when you create target at the runtime
- Update override

Lookup

When you configure SQL ELT optimization, the mappings support the following properties for Microsoft Azure Synapse SQL connected and unconnected lookup:

- Source type
  - Single Object - table, views, external table
  - Custom Query
  - Parameter
- Multiple matches
  - Connected lookup - Return all rows
  - Unconnected lookup - Report error
- Parameter - Allow source object to be overridden at run time
- Schema Name Override
- Table Name Override
- Pre-SQL
- Post-SQL
- SQL Override
- On Pre-Post SQL Error

- Edit Metadata
- Dynamic Lookup Cache for connected lookup
- Output Old Value On Update for connected lookup
- Insert Else Update for connected lookup
- Pre-build Lookup Cache - Auto, Value

Note: When a mapping runs without SQL ELT optimization , the Azure Blob Container Name or ADLS FileSystem Name are required fields. When you run a mapping with Source SQL ELT optimization, Azure Blob Container Name is a required field.

Supported features for Microsoft Azure Data Lake Storage Gen2 source

When you configure SQL ELT optimization, the mappings support the following properties for a Microsoft Azure Data Lake Storage Gen2 source:

Source

- Azure Government endpoints
- Source Type
  - Single object
  - Parameter
- Format Type
  - Delimited
  - ORC
  - Parquet
- Formatting options
  - Delimiter
  - Qualifier
  - First Data Row
  - Import from schema file
- Filesystem Name Override
- Directory Override
- File Name Override
- Compression Format
- Edit metadata

Note: You can also read data from a Microsoft Azure Data Lake Storage Gen2 case-sensitive database.

Lookup

When you configure SQL ELT optimization, the mappings support the following properties for Microsoft Azure Data Lake Storage Gen2 connected and unconnected lookup:

- Azure Government endpoints
- Source type
  - Single Object

- Parameter

- Multiple matches
  - Connected lookup - Return all rows
  - Unconnected lookup - Report error
- Format Type
  - Delimited
  - ORC
  - Parquet
- Formatting options
  - Delimiter
  - Qualifier
  - First Data Row
  - Import from schema file
- Filesystem Name Override
- Directory Override
- File Name Override
- Compression Format
- Edit metadata

Configuring a custom query for the Microsoft Azure Synapse SQL source object

You can push down a custom query to Microsoft Azure Synapse SQL.

Before you run a task that contains a custom query as the source object, you can set the **Create Temporary View** property in the mapping task properties.

Perform the following task to set the property:

1. In the mapping task, navigate to the SQL ELT Optimization section on the **Runtime Options** tab.
2. Set the **Create Temporary View** option based on your requirement.
   - When the **Create Temporary View** option is disabled, the task incorporates the custom query as a sub-query in the SQL ELT query.
   - When the **Create Temporary View** option is enabled, the task creates a temporary view based on the provided custom query and then generates the SQL ELT query with the temporary view.
3. Click **Finish**.

Rules and guidelines for SQL ELT optimization

Certain rules and guidelines apply for SQL ELT optimization to a Microsoft Azure Synapse SQL database.

General rules and guidelines

Consider the following rules and guidelines when you enable a mapping for SQL ELT optimization:

- When you use Rank transformation, the data is not stored as per the Rankindex order in the database.

- When you read varchar or nvarchar fields and use the Rank transfomation, the target data is sorted as per collation of the database.
- When you use Rank transformation, the session sort order that you specify in the advanced session property is not considered in a Microsoft Azure Synapse SQL mapping.
- When you select multiple objects as the source type, you cannot read more that two objects.
- When you use a custom query as the source type, the first three letters of the table name alias is always `INF`.
- To write date/time fields to a Microsoft Azure Synapse SQL target created at the runtime, edit the metadata for the target fields and change the native data type of all the date/time fields to datetime2. Else, the mapping fails.
- When you use a custom query as the source type, ensure that the table name or column name that you specify in the query does not contain unicode characters. Else, the mapping fails.
- When you use a Filter transformation, ensure that the filter value does not contain special characters. Else, the mapping does not write any rows to the target.
- You can use the update and delete operations only for an existing target.
- When you enable the **Create New at Runtime** option in a Microsoft Azure Synapse SQL target to write data from file such as Delimited, ORC, or Parquet, delete the **FileName** field.
- The reject threshold for a Microsoft Azure Synapse SQL target is considered for the whole data set, not for a batch of data.
- Do not use the Aggregator transformation with update operation.
- Do not configure a nested aggregate function in an Aggregator transformation.
- Do not configure parallel transformations in a mapping.
- Ensure that the Expression transformation does not contain a variable field.
- Use the ISNULL function within the IIF clause in Aggregator and Expression transformations.
- Use the IN function in a Filter transformation or within the IIF clause in Aggregator and Expression transformations.
- When you use STDDEV, VARIANCE, and TRUNC functions in an Aggregator transformation, data corruption is possible if the table has a defined decimal column with precision more than 28, but the table contains data with precision less than 28.
- When you use the `lastruntime` variable in an Expression transformation in full SQL ELT optimization, you must use the `yyyy-mm-dd hh:mi:ss` date format.
  For example, `to_char($lastruntime,'yyyy-mm-dd hh:mi:ss')`
- You cannot use the `ISNULL(column name1)<>ISNULL(column name2)` syntax.
  Use the following syntax for Filter and Router transformations:

  ```
  (ISNULL(column name1) AND ISNULL(column name2)) OR ( NOT ISNULL(column name1) AND NOT ISNULL(column name2))
  ```

  Use the following syntax for Expression and Aggregator transformations:

  ```
  IIF((NOT ISNULL(column name1) AND ISNULL(column name 2)) OR (ISNULL(column name1) AND NOT ISNULL(column name 2)), 1, 0)
  ```

- When you use a Filter or Router transformation and specify a string value in the filter condition, the mapping appends the character N before the condition.
  For example,

  ```
  Select*from [Azure].[employee] where (10125 <= "INF1"."EMPLOYEE_ID_NEW") AND ("INF1"."JOB_ID_NEW" = N'FI_ACCOUNT');
  ```

- When you pass columns with Null values in a Normalizer transformation, Null values are not written to the target.
- When you push the TO_DATE() function to process in Microsoft Azure Synapse SQL, ensure that the value matches the supported format. Prefix 0 if the value contains a single digit.
- When you push the TO_DATE() function to the Microsoft Azure Synapse SQL database and the argument contains null data, the mapping runs without SQL ELT optimization.
- When you push the MD5() function to process in Microsoft Azure Synapse SQL, you must specify a value of the String data type. Else, the mapping runs without SQL ELT optimization.
- When you push the MD5() function to process in Microsoft Azure Synapse SQL, the function returns different values for non-English characters in a mapping and in a mapping enabled with SQL ELT optimization.
- You cannot push the MD5() function to Microsoft Azure Synapse SQL if the table or column names contain unicode or special characters.
- When you push the REG_MATCH() function to process in Microsoft Azure Synapse SQL, it is recommended that you specify all the characters instead of a range of characters in the regular expression in a case-sensitive database. For example, use `[ABCDEF]` instead of `[A-F]`.
- When you push the TO_BIGINT() function to convert Varchar data to Bigint in Microsoft Azure Synapse SQL, use the format TO_BIGINT (numeric expression [, 1]) for the function to return the correct values. To round values stored in the Varchar column, convert values to any numeric format, and then convert to Bigint.
- When you run a mapping enabled with SQL ELT optimization and push functions to process in Microsoft Azure Synapse SQL, the function considers the collation rules that are set in the database. This causes the functions to return different values in a mapping compared to a mapping enabled with SQL ELT optimization.
- When you read from a Microsoft Azure Synapse SQL source and write to multiple targets from a mapping enabled for SQL ELT optimization in a single pipeline, and the Source transformation contains pre-SQL and post-SQL queries, the mapping runs these queries for each of the targets. This might cause data discrepancy. To avoid this, define the pre-SQL and post-SQL queries only in the Target transformation.

Mapping with Microsoft Azure Synapse SQL source and target

Use the following rules and guidelines when you configure SQL ELT optimization in a mapping that reads from and writes to Microsoft Azure Synapse SQL:

- The source and target table must be in the same database.
- When you configure filters in a Source transformation, consider the following guidelines:
  - If a mapping contains a Filter transformation and also a filter in the Source transformation, the mapping consolidates the filter conditions from both these transformations to filter the records. However, it is recommended that you use only one of these filters at a time in a mapping.
  - The column names in the filter condition must not contain unicode characters.
  - You cannot use system variables in filters.
  - You cannot use filters when the source object type is query.
- Ensure that each key column for the update operation is mapped, else the query generation fails and the mapping runs without SQL ELT optimization.
- When you use a custom query as the source type, ensure that there is no semicolon at the beginning of or within the query.

- When you use a custom query as the source type, ensure that the table name or column name that you specify in the query does not contain special characters.
- You cannot completely parameterize a multi-line custom query using a parameter file.
- If the custom query starts with the WITH clause, you must enable the **Create Temporary View** option to run the mapping with SQL ELT optimization.
- You cannot use the ORDER BY clause in a source custom query unless you also specify a TOP, OFFSET, or FOR XML clause in the query.
- You cannot use the OPTION clause in a custom query or SQL override.
- When you run a mapping to write data to new target created at runtime and you stop the job using clean stop, the target table created is not dropped.
- Source SQL ELT optimization
  - To run a mapping with SQL ELT optimization, the mapping must have at least one transformation between the source and target.
  - When you select source SQL ELT optimization, the task pushes the transformation logic for all the supported transformations downstream in the mapping.
  - When a function within a transformation is not supported, the query generation fails and the mapping runs without SQL ELT optimization.
  - You can connect the Source transformation only to a single downstream transformation.

Mapping with Microsoft Azure Data Lake Storage Gen2 source and write to a Microsoft Azure Synapse SQL target

Use the following rules and guidelines when you configure SQL ELT optimization in a mapping that reads from a Microsoft Azure Data Lake Storage Gen2 source and write to a Microsoft Azure Synapse SQL target:

- The Microsoft Azure Data Lake Storage Gen2 account and Microsoft Azure Synapse SQL account must be associated with the same Azure Active Directory tenant.
- The Azure Government endpoints for the source and target must be associated with the same Azure Active Directory tenant.
- You cannot configure a private endpoint or a virtual network to connect to Microsoft Azure Data Lake Storage Gen2. You can use managed identity authentication instead to securely connect to Microsoft Azure Data Lake Storage Gen2.
- You cannot read objects stored in subdirectories in Microsoft Azure Data Lake Storage Gen2.
- Ensure that the Microsoft Azure Data Lake Storage Gen2 source object is not parameterized.
- When you read data from Microsoft Azure Data Lake Storage Gen2, do not use escape characters. Else, it might lead to incorrect results.
- When you push down a mapping that reads from a Microsoft Azure Data Lake Storage Gen2 source and writes to a Microsoft Azure Synapse SQL target, the mapping task fails to generate the push down query if the task has all the following configurations:
  - The Microsoft Azure Synapse target connection is parameterized.
  - The target operation and Treat Source Rows As property in the mapping is set to Insert.
  - The Treat Source Rows As property is changed to Update in the mapping task.

Mappings in advanced mode

Use the following rules and guidelines when you configure SQL ELT optimization in a mapping in advanced mode:

- When you configure SQL ELT optimization to push the transformation logic to Microsoft Azure Synapse SQL, the queries generated for a mapping in advanced mode differs from the queries generated for a mapping.
- When you push functions to process in Microsoft Azure Synapse SQL, ensure that you specify valid parameters and return types for the functions. Otherwise, the mapping fails.
- When you configure SQL ELT optimization, ensure that the transformation does not contain a variable port.
- When you create a new target at runtime to write data to Microsoft Azure Synapse SQL, map all the incoming fields to the target fields.
- When a mapping enabled with SQL ELT optimization cannot push certain transformations to Microsoft Azure Synapse SQL, the mapping runs without SQL ELT optimization and the session log does not log an error.

CHAPTER 7

Data type reference

Data Integration uses the following data types in Microsoft Azure Synapse SQL mappings and mapping tasks:

- Microsoft Azure Synapse SQL native data types appear in the source and target transformations when you choose to edit metadata for the fields.
- Transformation data types. Set of data types that appear in the transformations. These are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. They appear in all transformations in a mapping.

When the Secure Agent reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the Secure Agent writes to a target, it converts the transformation data types to the comparable native data types.

Microsoft Azure Synapse SQL and transformation data types

The following table compares the Microsoft Azure Synapse SQL native data type to the transformation data type:

Microsoft Azure Synapse SQL Data Type	Transformation Data Type	Range and description
bigint	bigint	-9,223,372,036,854,770,000 to 9,223,372,036,854,770,000
binary	binary	1 to 8000 bytes
bit	integer	0,1,NULL
char	string	1 to 8000 characters
date	date/time	January 1, 1753 00:00:00 to 12/31/9999
datetime	date/time	January 1, 1753 00:00:00 to December 31, 9999 23:59:59.997

Microsoft Azure Synapse SQL Data Type	Transformation Data Type	Range and description
datetime2	date/time	1 to 8000 characters When you create a mapping in advanced mode, the fractional seconds in the datetime2 fields are written only up to 6 digits in the target.
datetimeoffset	string	
decimal	decimal	Number with 28-bit precision and scale 0 to 5 for mappings. **Note:** When the table has a defined decimal column with precision more than 28, but the table contains data less than 28, Data corruption might occur. Number with 38-bit precision and scale 0 to 5 for mappings in advanced mode.
float	decimal	Precision 7, scale 0 to 7
int	integer	-2,147,483,648 to 2,147,483,647
money	decimal	-922,337,203,685,477.0000 to 922,337,203,685,477.0000
nchar	string	1 to 8000 characters
nvarchar	string	1 to 4000 characters
real	double	Precision 7, scale 0 to 7
smalldatetime	date/time	1/1/1900 0:00 to 6/6/2079 23:59
smallint	integer	-32,768 to 32,767
smallmoney	decimal	-214,748.3648 to 214,748.3647
time	string	00:00:00.0000000 to 23:59:59.9999999
tinyint	integer	0 to 255
Uniqueidentifier	string	0 to 36 characters
varbinary	binary	1 to 8000 bytes
varchar	string	1 to 8000 characters

Rules and guidelines for data types

- Binary and Varbinary data types are not applicable for mappings in advanced mode.
- The `datetimeoffset` data type is supported only for passthrough mappings that read data from and write data to Microsoft Azure Synapse SQL objects.
- When you read or write varchar or nvarchar data types, do not use `\n` or `\r` character in the string data.

- When you create a new target, the fields of Uniqueidentifier data type in the source are converted to Nvarchar data type in the target.
  To retain the Uniqueidentifier data type, edit the metadata for the target fields and change the native type of the fields to Uniqueidentifier.

CHAPTER 8

Troubleshooting

Use the following section to troubleshoot errors in Microsoft Azure Synapse SQL Connector.

Troubleshooting a Microsoft Azure Synapse SQL connection

Unable to connect to Microsoft Azure Blob Storage or Microsoft Azure Synapse SQL VNet endpoints

When the Secure Agent machine is outside the virtual network (VNet) and you connect to Microsoft Azure Blob Storage or Microsoft Azure Synapse SQL VNet endpoints, the **Test Connection** fails.

You must configure the Secure Agent IP in both the **Virtual Network** section and **Firewall** section in your account and test the connection again.

Troubleshooting a mapping or mapping task

Time zone for the Date and Timestamp data type fields in Parquet file format defaults to the Secure Agent host machine time zone.

When you run a mapping to read from or write to Microsoft Azure Synapse SQL and select Parquet as the staging format in the advanced source properties, the time zone defaults to the Secure Agent host machine time zone.

To change the Date and Timestamp to the UTC time zone, configure the JVMOptions in the Secure Agent.

Perform the following steps to configure the JVM options in the Secure Agent:

1. Select **Administrator** > **Runtime Environments**.
2. On the **Runtime Environments** page, select the Secure Agent for which you want to configure the JVMOptions.
3. In the upper-right corner, click **Edit**.
4. In the **System Configuration Details** section, select **Data Integration Server** as the service and **DTM** as the type.
5. Edit the **JVMOption** field and set the value to `-Duser.timezone=UTC`.
6. Click **Save**.

When the staging property is enabled, the mapping fails even though the rows are successfully written to the target.

When you enable the staging property in a mapping that reads from Microsoft Azure Synapse SQL, the mapping might fail with one of the following errors even though the rows are successfully written to the target:

- `[FATAL] Error : HdfsBridge::recordReaderFillBuffer - Unexpected error encountered filling record reader buffer: HadoopSqlException: Error converting data type VARCHAR to DATETIME.`
- `[FATAL] Error : HdfsBridge::recordReaderFillBuffer - Unexpected error encountered filling record reader buffer: HadoopExecutionException: Too long string in column [-1]: Actual len = [1015]. MaxLEN=[1000]`

To resolve this issue, increase the reject threshold in the advanced target properties and run the mapping again.

[FATAL] Exception: com.microsoft.sqlserver.jdbc.SQLServerException: 100001;Failed to generate query plan.

When you use an ORDER BY clause in a custom query or SQL override, the mapping fails with the following error:

```
[FATAL] Exception: com.microsoft.sqlserver.jdbc.SQLServerException: 100001;Failed to generate query plan.
```

To use the ORDER BY clause in a query, you must configure the following property in the JVM options of the Secure Agent:

```
-DAzureSynapseDisableStagingForSort=true
```

If you use custom query in a Lookup transformation, you must perform the following additional tasks:

- Enable the **Sorted Input** option in the advanced lookup properties.
- The query must first contain the ORDER BY clause for the condition fields are included in the lookup condition in the ascending order and then the other fields in any desired order.
- When you configure the Lookup transformation to return multiple rows, the **Return All Rows** option does not return the values in the expected order.

Mapping failed with a Java heap space error

When you run a mapping that reads a large volume of data from or writes a large volume of data to Microsoft Azure Synapse SQL, the mapping fails with a Java heap space error.

Set the JVM options for type Tomcat JRE to increase the -Xms and -Xmx values in the system configuration details of the Secure Agent. You must then restart the Secure Agent.

[ERROR] com.informatica.cci.cloud.client.impl.CCIClientExceptionImpl: Invalid expression string for filter condition.

If you configure an uncached Lookup transformation to look up data from a Microsoft Azure Synapse SQL source and the source contains a column of DateTime or Timestamp data type, the mapping fails for certain values with the following error:

```
[ERROR] com.informatica.cci.cloud.client.impl.CCIClientExceptionImpl: Invalid expression string for filter condition.
```

Troubleshooting a mapping in advanced mode

Mapping configured to read or write Date and Int96 data types for Parquet files fails

A mapping configured to read from or write to Microsoft Azure Synapse SQL where you select Parquet as the staging format in the advanced source properties, fails in the following cases:

- Data is of the Date data type and the date is less than 1582-10-15.
- Data is of the Int96 data type and the timestamp is less than 1900-01-01T00:00:00Z.

To resolve this issue, specify the following spark session properties in the mapping task or in the custom properties file for the Secure Agent:

- `spark.sql.parquet.int96RebaseModeInWrite=LEGACY`
- `spark.sql.parquet.datetimeRebaseModeInWrite=LEGACY`
- `spark.sql.parquet.int96RebaseModeInRead=LEGACY`
- `spark.sql.parquet.datetimeRebaseModeInRead=LEGACY`

Time zone for the Date and Timestamp data type fields in Parquet file format defaults to the Secure Agent host machine time zone.

When you run a mapping to read from or write to Microsoft Azure Synapse SQL and select Parquet as the staging format in the advanced source or target properties, the time zone defaults to the Secure Agent host machine time zone.

To change the Date and Timestamp to the UTC time zone, you can either set the Spark properties globally in the Secure Agent directory for all the tasks in the organization that use this Secure Agent, or you can set the Spark session properties for a specific task from the task properties:

To set the properties globally, perform the following tasks:

1. Add the following properties to the `<Secure Agent installation directory>/apps/At_Scale_Server/41.0.2.1/spark/custom.properties` directory:
   - infacco.job.spark.driver.extraJavaOptions=-Duser.timezone=UTC
   - infacco.job.spark.executor.extraJavaOptions=-Duser.timezone=UTC
2. Restart the Secure Agent.

To set the properties for a specific task, navigate to the Spark session properties in the task properties, and perform the following steps:

- Select the session property name as `spark.driver.extraJavaOptions` and set the value to `-Duser.timezone=UTC` .
- Select `spark.executor.extraJavaOptions` and set the value to `-Duser.timezone=UTC`.

INDEX

A

Azure Blob Container Name 20, 28

C

cache
  enable lookup cache 38
CDC source
  Microsoft Azure Synapse SQL mapping task 33
Cloud Application Integration community
  URL 6
Cloud Developer community
  URL 6
connections
  Microsoft Azure Synapse SQL 12

D

Data Integration community
  URL 6
data types 80

E

error file directory 28

F

field delimiter 20, 28
flat file
  staging data 39
forward rejected rows 28

I

IDENTITY column 43
Informatica Global Customer Support
  contact information 7
Informatica Intelligent Cloud Services
  web site 6

L

lookup caches
  dynamic 38
Lookup transformation
  lookup caching 38

M

maintenance outages 7
mappings
  overview 20
  source properties 20
  target properties 28
mappings in advanced mode
  example 48
Microsoft Azure Synapse SQL
  connection properties 12
  introduction 8
  SQL ELT optimization 63

N

native data type 80
number of concurrent sessions to Blob Storage 20, 28

P

properties
  in mappings 20, 28

S

SQL ELT optimization
  preview 62
SQL ELT optimization preview 62
staging data
  flat file 39
status
  Informatica Intelligent Cloud Services 7
success file directory 28
system status 7

T

tracing level 20
transformation data type 80
trust site
  description 7

U

upgrade notifications 7

W

web site 6