

Performance Tuning Guidelines for Full Pushdown Optimization for Azure SQL Data Warehouse

Abstract

You can use pushdown optimization to push transformation logic to source databases or target databases. Use pushdown optimization to improve task performance by using the database resources. When you run a task configured for pushdown optimization, the task converts the transformation logic to an SQL query. The task sends the query to the database, and the database executes the query.

Supported Versions

- Cloud Data Integration Microsoft Azure SQL Data Warehouse

Table of Contents

Overview.	2
Best Practices.	2
Case Studies.	3
Configuring Azure SQL Data Warehouse ODBC Connection.	4
Configuring Azure SQL Data Warehouse ODBC Connection on Windows.	4
Configuring Azure SQL Data Warehouse ODBC Connection on Linux.	9
Creating an ODBC Connection.	10
Create a Mapping.	11
Create a Mapping Configuration Task.	11
Supported Pushdown Optimization Functions and Operators.	11

Overview

Pushdown Optimization (PDO) is the process of pushing transformation logic to the source or target database by Secure Agent. When you run a session configured for PDO, the Secure Agent analyzes the mapping and transformations to determine the transformation logic it can push to database.

In case of Full PDO, the source and target databases must be in the same relational database management system. The Secure Agent analyzes the mapping from the source to the target or until it reaches a downstream transformation it cannot push to the target database.

This document describes how to configure Azure SQL Data Warehouse ODBC Connection on Windows and Linux operating systems. It also explains how you can create an ODBC connection and case studies and best practices while using Full PDO.

Best Practices

Consider the following best practices while using Full Pushdown Optimization (PDO) over Azure SQL Data Warehouse:

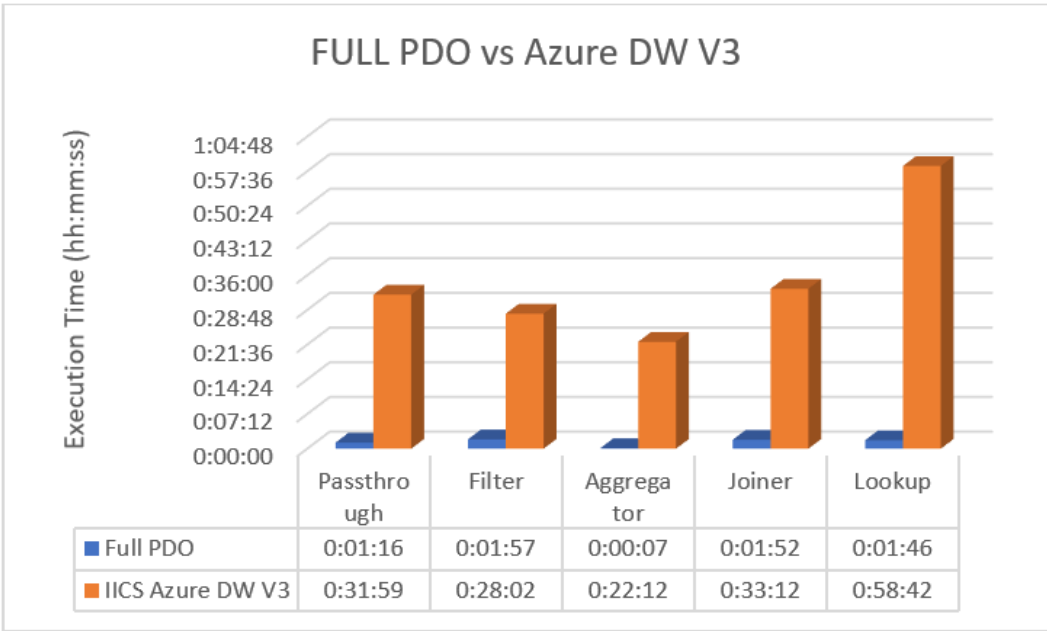
- Full PDO provides better performance for all supported transformations and their combinations over Azure SQL Data Warehouse V3 connector.
- Scaling Data Warehouse Units in Azure SQL Data Warehouse provides better performance.
- Operations such as insert, update, and delete in a full pushdown mapping perform on par when compared to executing SQL queries on tables in Azure SQL Data Warehouse from Azure SQL Data Warehouse Client or any third-party client.

Note: Azure SQL Data Warehouse source and target are in the same region as Secure Agent.

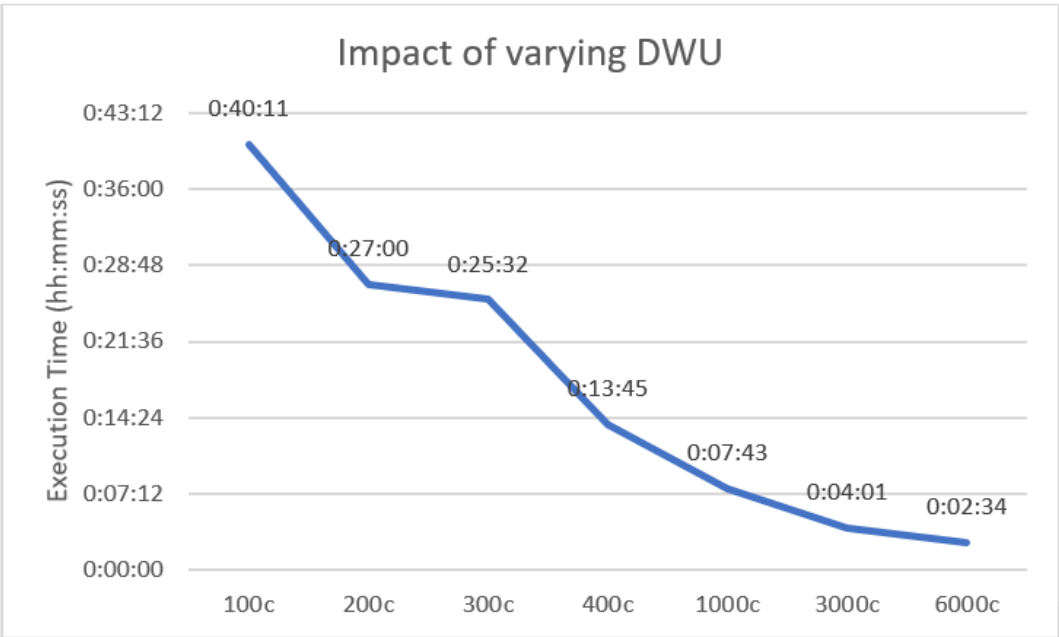
Case Studies

The following case studies show difference in execution time of Full Pushdown Optimization (PDO) and Azure SQL Data Warehouse V3 connector:

- Keeping Azure SQL Data Warehouse, Azure Blob Storage (for staging) and Secure Agent in the same region:



- Impact of varying Azure SQL Data Warehouse Units on performance:



Note: The time difference may vary with user's dataset.

Configuring Azure SQL Data Warehouse ODBC Connection

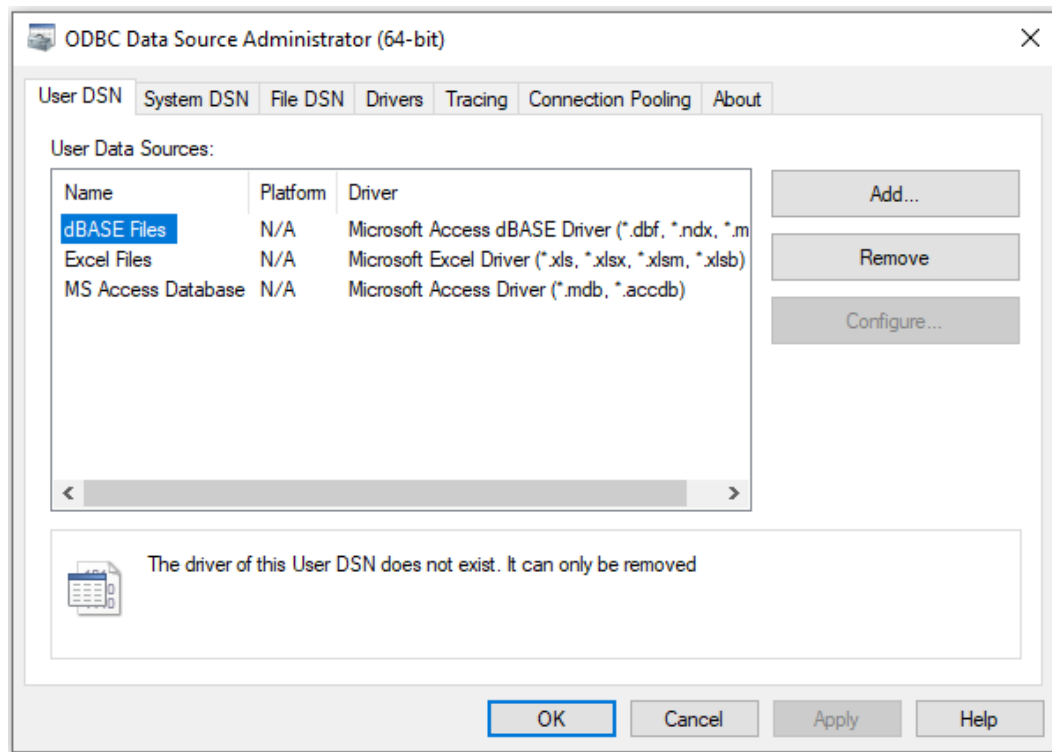
To use an ODBC connection to connect to Azure SQL Data Warehouse, you must create a data source name in ODBC datasource administrator.

Azure SQL Data Warehouse supports Microsoft ODBC drivers on Windows and Linux operating systems.

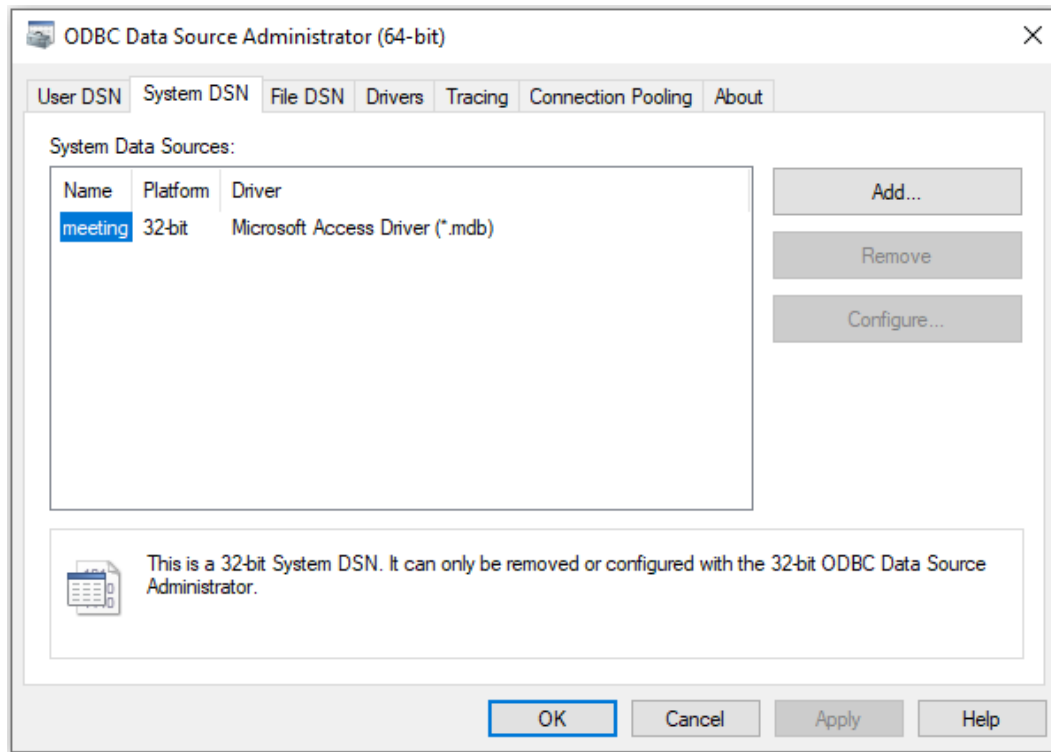
Configuring Azure SQL Data Warehouse ODBC Connection on Windows

Perform the following steps to configure an ODBC connection on Windows:

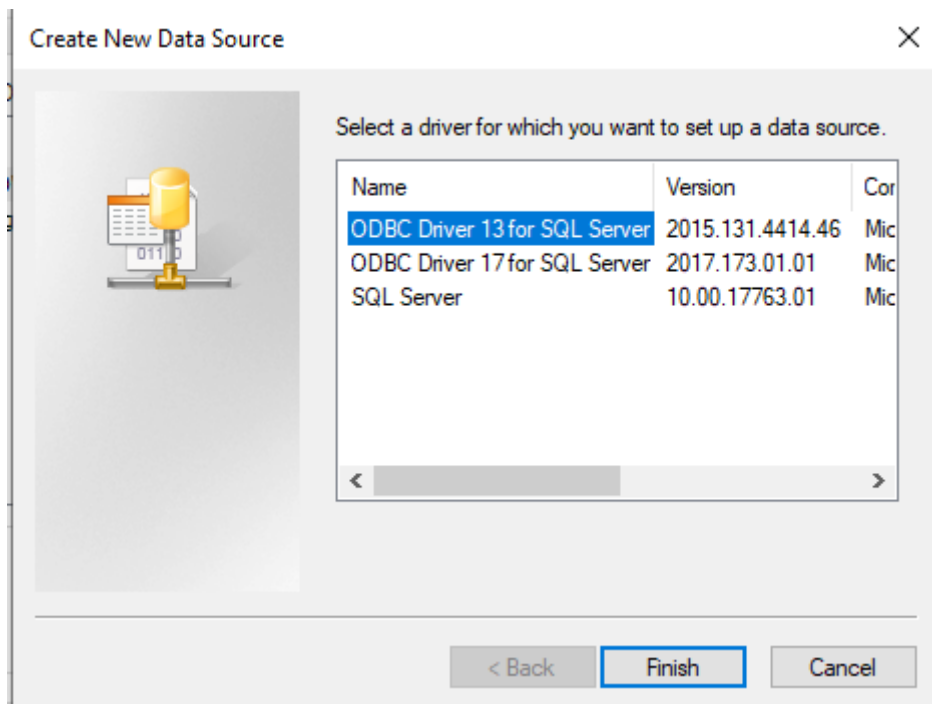
1. Download the Microsoft ODBC driver from the Microsoft website.
You must download the Microsoft ODBC 64-bit driver.
2. Install the Microsoft ODBC driver on the machine where the Secure Agent is installed.
3. Open the folder in which ODBC data source file is installed.
4. Run the `odbcad32.exe` file.
The **ODBC Data Source Administrator** dialog box appears:



5. Select **System DSN**.
The **System DSN** tab appears.



- Click **Add** to select a driver for which you want to set up a data source.
The following dialog box appears:

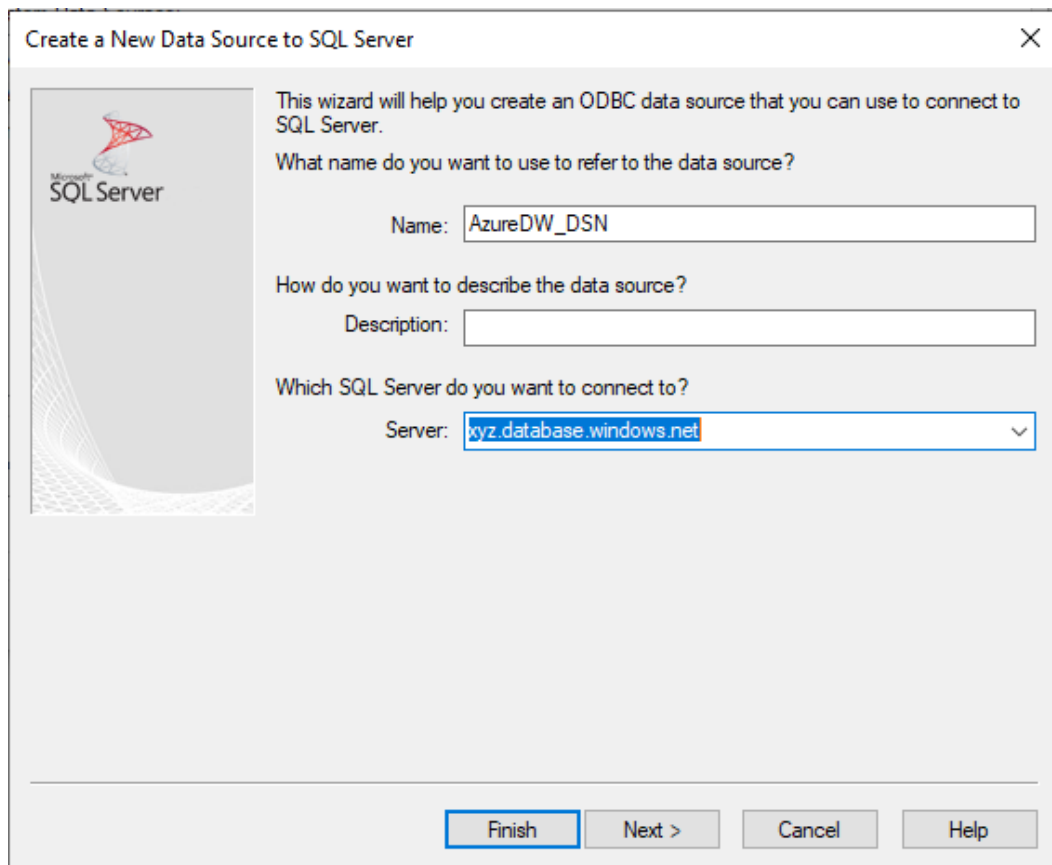


- Select **ODBC Driver 13 for SQL Server** and click **Finish**.

8. Specify the following connection properties for creating an ODBC data source to connect to SQL Server:

Property	Description
Name	Name of the data source.
Description	Description about the data source.
Server	Location of the Microsoft Azure SQL Data Warehouse Server.

The following image shows details of the ODBC data source:



Microsoft SQL Server

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: AzureDW_DSN

How do you want to describe the data source?

Description:

Which SQL Server do you want to connect to?

Server: xyz.database.windows.net

Finish Next > Cancel Help

9. Click Next.
10. Select **With SQL Server authentication using a login ID and password entered by user** and enter the login credentials.

The following image shows the verification page:

Create a New Data Source to SQL Server

How should SQL Server verify the authenticity of the login ID?

☐ With Integrated Windows authentication.
SPN (Optional):

☐ With Active Directory Integrated authentication.

☒ With SQL Server authentication using a login ID and password entered by the user.

☐ With Active Directory Password authentication using a login ID and password entered by the user.

☐ With Active Directory Interactive authentication using a login ID entered by the user.

Login ID: SampleAdminUser

Password: ●●●●●●●●

< Back Next > Cancel Help

11. Click **Next**.
12. Select **Change the default database to** and enter the value.
The following image shows the create new data source page:

Create a New Data Source to SQL Server

☒ Change the default database to:
SampleDatabase

Mirror server:

SPN for mirror server (Optional):

☐ Attach database filename:

☒ Use ANSI quoted identifiers.
☒ Use ANSI nulls, paddings and warnings.


Application intent:
READWRITE

☐ Multi-subnet failover.
☒ Transparent Network IP Resolution.
☐ Column Encryption.
☐ Use FMTONLY metadata discovery.

< Back Next > Cancel Help

13. Click **Next**.
14. Verify that **Use strong encryption for data** and **Trust server certificate** are selected. Click **Finish**.
The following image shows the new data source page:

Create a New Data Source to SQL Server



☐ Change the language of SQL Server system messages to:
(Default)
☒ Use strong encryption for data.
☒ Trust server certificate.
☒ Perform translation for character data.
☐ Use regional settings when outputting currency, numbers, dates and times.
☐ Save long running queries to the log file:
C:\Users\ababaj\AppData\Local\Temp\QUERY.LOG Browse...
Long query time (milliseconds): 30000
☐ Log ODBC driver statistics to the log file:
C:\Users\ababaj\AppData\Local\Temp\STATS.LOG Browse...
Connect retry count: 1
Connect retry interval (seconds): 10

< Back
Finish
Cancel
Help

15. Click **Test Data Source** to test the connection in **ODBC Microsoft SQL Server Setup** box.
16. Click **OK**.

Configuring Azure SQL Data Warehouse ODBC Connection on Linux

Perform the following steps to configure an ODBC connection on Linux:

1. Download the Microsoft ODBC 13 driver from the Microsoft Azure website.
You must download the Microsoft ODBC 64-bit driver.
2. Install the Microsoft ODBC drivers on the machine where the Secure Agent is installed.
3. Specify the following properties in the `odbc.ini` file:

Property	Description
Driver	Name of the driver returned by SQLDrivers.
Description	Description of the driver used.
Server	The name of a SQL Server instance.
Database	Name of the default SQL Server database for the connection.

Property	Description
LoginID	Login ID to access SQL Server database.
Password	Password for SQL Server database.
QuotedID	Select Yes to activate the QUOTED_IDENTIFIERS for the connection.
AnsiNPW	Select Yes to allow driver to use ANSI-defined behaviors for handling NULL comparisons, character data padding, warnings, and NULL concatenation.
EncryptionMethod	Select 1 to encrypt data before sending it over the network. Default value is 0.
ValidServerCertificate	Select 0 to validate the server certificate. Default value is 0.

- Run the following commands to export the `odbc.ini` file:
 - Export `ODBCHOME=/<ODBCHOME file path>/<release no.>`
 - Export `LD_LIBRARY_PATH=/<LD_LIBRARY_PATH file path>/`
 - Export `PATH=/<PATH file path>/`
 - Export `ODBCINI=/<odbc.ini file path>/odbc.ini`
- Restart the Secure Agent.

Creating an ODBC Connection

Create ODBC Connections to access Azure SQL Data Warehouse source and target objects.

Perform the following steps to create an ODBC connection on the Connections page:

- On the Connections page, click **New Connection**. The New connection page appears.
- Configure the following connection details in the Connection details section:

Property	Description
Connection Name	Name of the ODBC connection.
Description	Description of the connection.
Type	Type of the connection. Select the type of connection as ODBC.

- Configure the following connection details in the Connection Properties section:

Property	Description
Runtime Environment	The name of the runtime environment where you want to run the tasks.
User Name	User name of the Microsoft Azure SQL Data Warehouse account.
Password	Password for the Microsoft Azure SQL Data Warehouse account.

Property	Description
Data Source Name	Enter the name of the ODBC data source name that you created for the Microsoft Azure SQL Data Warehouse database.
Schema	Microsoft Azure SQL Data Warehouse schema name.
Code Page	Select the code page that the Secure Agent must use to read or write data.
ODBC Subtype	Enter the value of the ODBC Subtype field as Azure DW.

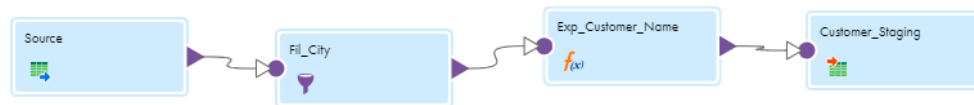
4. Click OK.

The Microsoft Azure SQL Data Warehouse ODBC connection is created successfully.

Create a Mapping

Consider a use case where you want to read data from the `Customers` table and fetch records of the people who lives in either Santa Cruz or Santa Barbara and concatenate their first and last names. You need to write the full name and city of the customers in Azure Data Warehouse target file.

The following image shows a sample mapping, `m_customer_staging`:



Select the `AzureDW_ODBC` connection you created to connect to Azure Data Warehouse database and read data from the `Customers` table. Use the Fetching and Concatenation transformations to get the desired output.

Create a Mapping Configuration Task

When you create a Mapping Configuration task, select the `m_customer_staging` mapping. Navigate to the **Schedule** tab and select **Pushdown Optimization** and set the value as **Full** in the advanced session properties.

Save and run the Mapping Configuration task.

To verify that the pushdown optimization has taken place, you can check the session log for the job. Click **Monitor>Activity Log** to view the session log for jobs.

Supported Pushdown Optimization Functions and Operators

Microsoft Azure SQL Data Warehouse V3 supports Full Pushdown Optimization (PDO).

The following table summarizes the availability of pushdown functions in a Microsoft Azure SQL Data Warehouse. Columns marked with an X indicate that the function can be pushed to the Microsoft Azure SQL Data Warehouse by

using full pushdown optimization. Columns marked with a dash (-) symbol indicate that the function cannot be pushed to the database.

Function	Pushdown	Function	Pushdown	Function	Pushdown
ABORT()	-	INITCAP()	-	REG_MATCH()	-
ABS()	X	INSTR()	X	REG_REPLACE	-
ADD_TO_DATE()	-	IS_DATE()	-	REPLACECHR()	X
AES_DECRYPT()	-	IS_NUMBER()	-	REPLACESTR()	X
AES_ENCRYPT()	-	IS_SPACES()	-	REVERSE()	-
ASCII()	X	ISNULL()	X	ROUND(DATE)	-
AVG()	X	LAST()	X	ROUND(NUMBER)	X
CEIL()	X	LAST_DAY()	-	RPAD()	-
CHOOSE()	-	LEAST()	-	RTRIM()	X
CHR()	X	LENGTH()	X	SET_DATE_PART()	-
CHRCODE()	-	LN()	-	SIGN()	X
COMPRESS()	-	LOG()	-	SIN()	X
CONCAT()	-	LOOKUP	-	SINH()	-
COS()	X	LOWER()	X	SOUNDEX()	X
COSH()	-	LPAD()	-	SQRT()	X
COUNT()	X	LTRIM()	X	STDDEV()	X
CRC32()	-	MAKE_DATE_TIME()	-	SUBSTR()	-
CUME()	-	MAX()	X	SUM()	X
DATE_COMPARE()	-	MD5()	X	SYSDATE()	-
DATE_DIFF()	X	MEDIAN()	-	SYSTIMESTAMP()	X
DECODE()	X	METAPHONE()	-	TAN()	X
DECODE_BASE64()	-	MIN()	X	TANH()	-
DECOMPRESS()	-	MOD()	X	TO_BIGINT	X
ENCODE_BASE64()	-	MOVINGAVG()	-	TO_CHAR(DATE)	X
EXP()	X	MOVINGSUM()	-	TO_CHAR(NUMBER)	X
FIRST()	X	NPER()	-	TO_DATE()	X

Function	Pushdown	Function	Pushdown	Function	Pushdown
FLOOR()	X	PERCENTILE()	-	TO_DECIMAL()	X
FV()	-	PMT()	-	TO_FLOAT()	X
GET_DATE_PART()	X	POWER()	X	TO_INTEGER()	-
GREATEST()	-	PV()	-	TRUNC(DATE)	-
IIF()	X	RAND()	-	TRUNC(NUMBER)	X
IN()	-	RATE()	-	UPPER()	X
INDEXOF()	-	REG_EXTRACT()	-	VARIANCE()	X

Microsoft Azure SQL Data Warehouse connector supports the following date formats with the `TO_DATE()` function:

- YYYY-MM-DD HH24:MI:SS.NS
- YYYY-MM-DD HH12:MI:SS.NSAM
- MON DD YYYY HH12:MI:SS.NSAM
- MON DD YYYY HH24:MI:SS.NS
- DD MON YYYY HH12:MI:SS.NSAM
- DD MON YYYY HH24:MI:SS.NS
- MM/DD/YY HH12:MI:SS.NSAM
- MM/DD/YY HH24:MI:SS.NS
- MM/DD/YYYY HH12:MI:SS.NSAM
- MM/DD/YYYY HH24:MI:SS.NS
- HH24:MI:SS.NS
- HH12:MI:SS.NSAM

Note: Use NS in the date format to support microseconds. Microsoft Azure SQL Data Warehouse supports fractional seconds upto 7 digits, which is 100 nano seconds.

The following table lists the pushdown operators that can be used in a Microsoft Azure SQL Data Warehouse. Columns marked with an X indicate that the operator can be pushed to the Microsoft Azure SQL Data Warehouse by using full pushdown optimization. Columns marked with a dash (-) symbol indicate that the operator cannot be pushed to the database.

Operator	Pushdown
+ - * /	X
%	X
	-
= > < >= <= <>	X

Operator	Pushdown
!=	X
^=	X
NOT AND OR	X

The following table summarizes the availability of variables in Microsoft Azure SQL Data Warehouse. Columns marked with an X indicate that the variable can be pushed to the Microsoft Azure SQL Data Warehouse by using full pushdown optimization. Columns marked with a dash (-) symbol indicate that the variable cannot be pushed to the Microsoft Azure SQL Data Warehouse.

Variable	Pushdown
SESSSTARTTIME	-
SYSDATE	X
WORKFLOWSTARTTIME	-

Microsoft Azure SQL Data Warehouse V3 supports the following transformations for full pushdown:

- Aggregator transformation
- Expression transformation
- Filter transformation
- Joiner transformation
- Lookup transformation
- Router transformation
- Union transformation

Authors

Krishna Prabhakar Devarakonda

Manav Khurana

Anushka Bajaj