

# SNOWFLAKE SNOWCONVERT FOR SQL SERVER

SnowConvert App Version 1.4.1.0

Engine Version 29.2.43

Executed on 02-05-2025 at 19:25:00

Conversion Time:  00:00:13

Conversion Speed: 243 lines/sec

The purpose of this document is to summarize the technical considerations and code analysis in migrating SQL to Snowflake from SQL Server that either have an impact on the automated code conversion or cannot be handled by automated code conversion, as well as provide a high-level inventory and automation capability of the code that will need to be addressed. A glossary of terms used is located at the end of the document.

##### KEY TERMS

Code Unit (CU) -SnowConvert breaks down code for reporting here into code units.  Please see the documentation “[here](https://docs.snowconvert.com/sc/general/getting-started/running-snowconvert/review-results/reports/top-level-code-units-report)” for an explanation of how code units are defined.

Code Unit Parent Category (CUPC) - For summary purposes in certain sections of this document you code units are grouped together to display conversion rates, counts and other metrics.  All detail for code units can still be found and analyzed in the top level code units document. For information on how code units are grouped, please refer to the documentation [here](https://docs.snowconvert.com/sc/general/getting-started/running-snowconvert/review-results/reports/top-level-code-units-report).

##### CODE COMPLETENESS SCORE

SnowConvert results are only as good as the completeness of the provided code.  A full lineage of information is needed in order to properly convert many objects.  The Code Completeness score is an indication of how complete the provided code base is.  Anything less than a score of 100 means SnowConvert identified missing object references in the code. It is advisable to convert dependent objects together to avoid getting missing object remarks. As an example, a Procedure definition converted individually without the dependent tables or functions would result in missing dependency remarks.

|  |
| --- |
| **93.6** |

[Learn more](https://docs.snowconvert.com/sc/general/getting-started/running-snowconvert/review-results/reports/assessment-report/code-completeness-score)

For details on objects reported as missing from the submitted code, see the section on Missing Dependent Objects.  For best results, revise the submitted code base to include a complete set of code.

**EXCLUDED SCOPE SUMMARY**

SnowConvert only supports [certain file types and code units](https://docs.snowconvert.com/sc/general/getting-started/running-snowconvert/review-results/snowconvert-scopes).  The following outlines what was identified in the submitted code and has been ***excluded*** from the scope of this assessment.  These excluded items are not converted by SnowConvert and do not affect the overall conversion rates reported in this assessment.  For additional information see the section on Excluded Scope Breakdown.

| **Excluded from Assessment:**  **Files: Out of Scope NaN%** |  |  | **Excluded from Conversion:  Code Units:  Out of Scope 0%** |  |
| --- | --- | --- | --- | --- |
| Unsupported extensions: | 0 |  | TRIGGER | 0 |
| Unexpected encoding: | 0 |  | GRANT | 0 |

##### ASSESSED CONVERSION SCOPE SUMMARY

| Files | 4 | Fully Converted Code Units | 76.6% |
| --- | --- | --- | --- |
| Code Units | 406 | Lines of Code Conversion Rate | 81.33% |
| Lines of Code | 3283 | Functional Difference Messages | 277 |
| Files Not Generated | 0 | Performance Reviews | 4 |
| Parsing EWIs | 0 | Missing Dependent Objects | 20 |
| Other EWIs | 294 |  |  |

## TABLE OF CONTENTS

[**SNOWFLAKE SNOWCONVERT FOR SQL SERVER 1**](#_heading=h.gjdgxs)

[TABLE OF CONTENTS 4](#_heading=h.30j0zll)

[**CONVERSION SETTINGS 5**](#_heading=h.1fob9te)

**EXCLUDED** [**SCOPE**](#_heading=h.tyjcwt) **BREAKDOWN** [**5**](#_heading=h.tyjcwt)

[File Details 5](#_heading=h.3dy6vkm)

[Code Unit Detail 5](#_heading=h.1t3h5sf)

[CODE UNITS SUMMARY 6](#_heading=)

[Top Level Code Units 6](#_heading=h.3rdcrjn)

[**CONVERSION REMARKS DETAIL 6**](#_heading=h.26in1rg)

[Functional Difference Messages (FDMs) 6](#_heading=h.lnxbz9)

[Errors, Warnings, & Issues (EWIs) 7](#_heading=h.35nkun2)

[Missing Dependent Objects (MDOs) 9](#_heading=)

[Performance Reviews (PRFs) 9](#_heading=)

[**CODE UNIT CONSIDERATIONS DETAIL 10**](#_heading=h.2jxsxqh)

[Tables 10](#_heading=h.3j2qqm3)

[Views 11](#_heading=h.1y810tw)

[Materialized Views 11](#_heading=h.4i7ojhp)

[Functions 11](#_heading=h.2xcytpi)

[Procedures 12](#_heading=h.1ci93xb)

[**COMPLEX CODE PATTERNS 12**](#_heading=h.3as4poj)

[Subqueries 12](#_heading=h.1pxezwc)

[Cursors 13](#_heading=h.2vhlpabwk2b0)

[Statements with usages of dynamic SQL 14](#_heading=h.hq7iqqmce63c)

[Transactions 14](#_heading=h.ut9cawm7as1j)

[**GLOSSARY 14**](#_heading=h.147n2zr)

## 

## 

## CONVERSION SETTINGS

SnowConvert settings which were used to run this assessment/conversion which affect certain behaviors and translations in the conversion engine.

| **General** | |
| --- | --- |
| Comment objects with missing dependencies | Off |
| Disable EWI comments generation (errors, warnings and issues) | Off |
| Encoding of the input files | Unicode (UTF-8) |

| **DB Object Names** | |
| --- | --- |
| Schema | None |
| Database | None |
| Default | On |
| Use Existing Name Qualification | Off |

| **Stored procedures target language** | |
| --- | --- |
| Snowflake Scripting | On |
| JavaScript | Off |

| **Materialized views** | |
| --- | --- |
| Target lag | 1 day |
| Warehouse | UPDATE\_DUMMY\_WAREHOUSE |

## 

## EXCLUDED SCOPE BREAKDOWN

This section contains details on the scope which was submitted but was not assessed or converted by SnowConvert. [See the SnowConvert Documentation for more detail](https://docs.snowconvert.com/sc/general/getting-started/running-snowconvert/review-results/snowconvert-scopes)[.](https://docs.snowconvert.com/snowconvert/general/getting-started/running-snowconvert/review-results/snowconvert-scopes)

### Excluded from Assessment

##### File Details

| **File Path/Name** | **File Size** | **Reason** |
| --- | --- | --- |
| N/A | N/A | N/A |

### 

### Excluded from Conversion

##### Code Unit Detail

| **Type** | **Number of Objects** | **Lines of Code** |
| --- | --- | --- |
| Grant | 0 | 0 |
| Trigger | 0 | 0 |

## 

## CODE UNITS SUMMARY

### Top Level Code Units

Code units are used to holistically count code across multiple types of files and scenarios and then grouped into categories here for summarization.  For information on how CUs are determined please see the [SnowConvert documentation here](https://docs.snowconvert.com/sc/general/getting-started/running-snowconvert/review-results/reports/top-level-code-units-report)[.](https://docs.snowconvert.com/snowconvert/general/getting-started/running-snowconvert/review-results/reports/top-level-code-units-report)

## 

| **Code Unit Parent Category** | **Code**  **Unit** | **Conversion Rate** | | **Total # of Code Units** | **Lines of Code** | **Parsing EWIs** | **Other EWIs** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fully Converted  Code Units** | **Lines of**  **Code** |
| DML | EXECUTE | 100 % | 100 % | 5 | 11 | 0 | 0 |
| FLOW CONTROL | IF | 0 % | 50 % | 2 | 8 | 0 | 4 |
| FUNCTION | CREATE FUNCTION | 85.71 % | 80.72 % | 14 | 332 | 0 | 2 |
| INDEX | CREATE INDEX | 100 % | 100 % | 12 | 14 | 0 | 0 |
| OTHER | PRINT | 0 % | 50 % | 12 | 24 | 0 | 12 |
| PROCEDURE | CREATE PROCEDURE | 21.05 % | 68.46 % | 19 | 612 | 0 | 47 |
| SESSION / BATCH CONTROL | DECLARE | 100 % | 100 % | 8 | 10 | 0 | 0 |
| SESSION / BATCH CONTROL | GO | 100 % | 100 % | 174 | 174 | 0 | 0 |
| SESSION / BATCH CONTROL | SET | 50 % | 50 % | 8 | 8 | 0 | 4 |
| TABLE | CREATE TABLE | 53.21 % | 88.73 % | 109 | 1207 | 0 | 133 |
| VIEW | CREATE VIEW | 79.07 % | 76.72 % | 43 | 859 | 0 | 92 |

## 

## 

## CONVERSION REMARKS DETAIL

### Functional Difference Messages (FDMs)

In the course of converting your legacy platform code to Snowflake, it is essential to acknowledge that SQL Server and Snowflake are distinct platforms, each with its unique set of features, functions, and capabilities.  In many cases, full functional equivalence cannot be achieved through brute force automation and involvement is required to bridge these functional differences.  SnowConvert calls out all known cases of these functional differences as Functional Difference

Messages where the code has been successfully converted to the extent possible and can only be further improved by reviewing details that lie outside of the code which may include business use cases, data ingestion processes or other architectural considerations.

##### Purpose

Customer Review: The purpose of this remark is to ensure transparency and provide visibility into areas of the conversion where human intervention may be required. It allows you, as the customer, to review and address these platform-specific differences according to your specific business requirements.

Customization Opportunities: By highlighting these differences, it enables you to assess whether any custom adaptations, workarounds, or alternative solutions may be required to achieve the desired functionality on the Snowflake platform.

Full Compatibility: It aims to ensure that your Snowflake environment, after migration, aligns with your expectations, while also complying with Snowflake's architecture and capabilities.

##### Action Required

Your review of these platform differences is essential. Depending on your specific use case and business needs, there could be substantial work to perform both outside and inside the code in order to achieve a successful migration.  You may need to collaborate with database administrators or developers to implement solutions or workarounds to achieve functionality equivalent to what was present in your legacy platform.  Each of the FDMs should be reviewed and any impacts fully understood prior to deploying any of the code and moving into testing.

Please reach out to our support team for further guidance in addressing these platform differences.

##### Summary

|  | **FDMs** |
| --- | --- |
| # of total remarks | 277 |
| # of unique remarks | 10 |

##### Detail

| **Code** | **Description** | **Instances** |
| --- | --- | --- |
| [SSC-FDM-0012](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/general/ssc-fdm-0012) | Constraint in Default Expression is not Supported | 151 |
| [SSC-FDM-TS0015](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/sql-server/ssc-fdm-ts0015) | Data type is not supported in Snowflake | 47 |
| [SSC-FDM-0007](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/general/ssc-fdm-0007) | Element with missing dependencies | 33 |
| [SSC-FDM-0019](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/general/ssc-fdm-0019) | Semantic information could not be loaded. | 13 |
| [SSC-FDM-0021](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/general/ssc-fdm-0021) | Create Index is not supported. | 12 |
| [SSC-FDM-TS0014](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/sql-server/ssc-fdm-ts0014) | Computed column transformed. | 11 |
| [SSC-FDM-0029](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/general/ssc-fdm-0029) | User Defined function was transformed to a Snowflake procedure | 5 |
| [SSC-FDM-0003](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/general/ssc-fdm-0003) | Conversion Rate Inconsistency | 2 |
| [SSC-FDM-TS0027](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/sql-server/ssc-fdm-ts0027) | SET ANSI\_NULLS ON statement may have a different behavior in Snowflake | 2 |
| [SSC-FDM-TS0023](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/functional-difference/sql-server/ssc-fdm-ts0023) | Error message could be different in snowflake | 1 |

### 

### Errors, Warnings, & Issues (EWIs)

EWIs in the output code are generated by SnowConvert in places where full automation is not implemented, they require review and manual remediation. They typically produce a functional or runtime difference. The SnowConvert team uses internal metrics to classify EWIs based on how much effort, on average, it takes to correct the code.

| **LOW** | **MEDIUM** | **HIGH** | **CRITICAL** |
| --- | --- | --- | --- |
| The user may have to invest a low amount of manual effort to complete the conversion. | The user may have to invest a medium amount of manual effort to complete the conversion. | The user may have to invest a high amount of manual effort to complete the conversion. | Errors that cause exceptions in SnowConvert.  The user may have to invest a substantial amount of manual effort to complete the conversion. |

*For more information about EWIs, such as their severity and examples of each type, please visit our* [*documentation page*](https://docs.snowconvert.com/snowconvert/general/technical-documentation/issues-and-troubleshooting)

##### Parsing

#### SUMMARY

Total Unparsed Lines of Code:

|  | **CRITICAL** |
| --- | --- |
| # of occurrences | 0 |
| # of unique issues | 0 |

#### DETAIL

| **Error Code** | **Description** | **Instances** | **Severity** |
| --- | --- | --- | --- |
| [N/A](https://docs.snowconvert.com/snowconvert/) | N/A | N/A | N/A |

##### Other

#### SUMMARY

|  | **LOW** | **MEDIUM** | **HIGH** | **CRITICAL** |
| --- | --- | --- | --- | --- |
| # of occurrences | 56 | 238 | 0 | 0 |
| # of unique issues | 3 | 3 | 0 | 0 |

#### DETAIL

| **Error Code** | **Description** | **Instances** | **Severity** |
| --- | --- | --- | --- |
| [SSC-EWI-0073](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/general/ssc-ewi-0073) | Pending Functional Equivalence Review | 214 | Medium |
| [SSC-EWI-TS0041](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/sql-server/ssc-ewi-ts0041) | XML data type methods are not supported in Snowflake. | 22 | Medium |
| [SSC-EWI-0002](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/general/ssc-ewi-0002) | Default Parameters May Need To Be Reordered | 2 | Medium |
| [SSC-EWI-0040](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/general/ssc-ewi-0040) | Statement Not Supported | 44 | Low |
| [SSC-EWI-0036](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/general/ssc-ewi-0036) | Data type converted to other data type | 7 | Low |
| [SSC-EWI-0035](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/general/ssc-ewi-0035) | CHECK statement not supported | 5 | Low |

### 

### Missing Dependent Objects (MDOs)

Completeness Score (0-100)      93.6

Total Missing Object References 91

Unique Missing Object References       20

### 

### Performance Reviews (PRFs)

##### Summary

|  | **Informational** |
| --- | --- |
| # of remarks | 4 |
| # of unique remarks | 1 |

##### Detail

| **Code** | **Description** | **Instances** |
| --- | --- | --- |
| [SSC-PRF-TS0001](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/performance-review/sql-server/ssc-prf-ts0001) | Recursion for CTE not checked. | 4 |

*If you are using the full version of SnowConvert, you can find out the exact file and location of each error in the Issues Report in the output reports folder created by SnowConvert.*

## CODE UNIT CONSIDERATIONS DETAIL

This section considers both top-level and embedded code units, i.e. tables defined inside procedures or script files will also be counted in the table section. That is the reason why the number of code units here might differ compared to the Top-level code unit breakdown.

The sections below summarize many different key considerations for some specific code unit categories.  Much of the information covered is identified by the different remarks in the code that SnowConvert inserts (FDMs, EWIs, PRFs, etc..).

### Tables

Code Unit Conversion Rate:  53.21 %

Lines of Code Conversion Rate:  88.73 %

Number of Tables:  109

Lines of Code:  1207

Total Parsing Errors:  0

**Table types** – Table properties may not perfectly line up between Snowflake and your source code platform. Here are some key type differences between Snowflake and your source language.

| **Table types** | **Instances** | **Tables Impacted** | **Percent Impacted** |
| --- | --- | --- | --- |
| Temporary Tables | 0 | 0/109 | 0% |
| Global Temporary Tables | 0 | 0/109 | 0% |

If a hyphen (‘-‘) is listed in the table above, it means no objects of that kind were found on the input folder.

**Unique Conversion Elements** – Other elements that may not explicitly fit into one of the previous categories that have significance in a migration.

| **Data Types** | **Instances** | **Tables Impacted** | **Percent Impacted** |
| --- | --- | --- | --- |
| Case insensitive collation | 0 | 0/109 | 0% |

If a hyphen (‘-’) is listed in the table above, it means no objects of that kind were found in the input folder.

**Notes:**

* Case insensitive collation – This collation setting can cause performance degradation in Snowflake when executing queries. Its use should be analyzed to decide if removing the collation is needed or not.

### Views

Code Unit Conversion Rate:  79.07 %

Lines of Code Conversion Rate: 76.72 %

Number of Views:  43

Number of Views created with only SELECT \* FROM: 0

Lines of Code:  859

Total Parsing Errors:  0

### Materialized Views

Materialized views are converted to dynamic tables in Snowflake.  How materialized views are refreshed is a consideration that must be made outside of the automated conversion process.

Code Unit Conversion Rate:  - %

Lines of Code Conversion Rate: - %

Number of Views:  -

Lines of Code:  -

Total Parsing Errors:  -

Number of Materialized Views referencing multiple objects: -

### Functions

Snowflake does not support the same features in functions as SQL Server. Not all possible logic and database access can be consistently replicated with Snowflake JavaScript or SQL functions. Some of them are transformed to Snowflake procedures to keep the functional equivalence, but these kinds of functions cannot be called from queries due to the procedure’s behavior. These functions are currently transformed to dummy user-defined functions.

Code Unit Conversion Rate: 90.91 %

Lines of Code Conversion Rate: 100 %

Number of Scalar Functions: 22

Number of Single Statement Scalar Functions: 3

Number of Single Statement Table-Valued Functions: 0

Number of Multi-Statement Table-Valued Functions: 1

Number of Multi-Statement Scalar Functions: 19

Scalar Function lines of code: 810

Table-Valued Function lines of code: 65

Lines of Code: 332

Total Parsing Errors: 0

Total Functions: 22

##### *Note*

Only functions written by SQL are evaluated by SnowConvert and appear in the code conversion rate, number of functions, lines of code, and total parsing errors listed above.

##### SQL Function Conversion Breakdown

Function calls inside DML: 0

Function calls inside DDL: 2

If you want to know more about this transformation you can follow [this link](https://docs.snowconvert.com/sc/for-transactsql/translation-reference)

### Procedures

Code Unit Conversion Rate:  21.05 %

Lines of Code Conversion Rate: 68.46 %

Number of Procedures:  19

Lines of Code:  612

Total Parsing Errors:  0

## COMPLEX CODE PATTERNS

### Subqueries

| **Subquery pattern** | **Instances** | **Views Impacted** | **Procedures Impacted** | **Functions Impacted** |
| --- | --- | --- | --- | --- |
| Subquery clause | 0 | 0/43 | 0/19 | 0/22 |
| Correlated subquery | 0 | 0/43 | 0/19 | 0/22 |

##### Unsupported subqueries

To determine if a subquery is supported, Snowflake considers two characteristics of the subquery: if it is correlated and if it is scalar.

A query is correlated if it contains a reference to any column from outside the query, while a query is scalar if it returns a single value (a single row containing a single column). For correlated subqueries, to determine if a query is scalar Snowflake evaluates the query statically, meaning the query should have only an aggregate function and no group by.

According to the [Snowflake documentation](https://docs.snowflake.com/en/user-guide/querying-subqueries.html#types-supported-by-snowflake), the supported subqueries are:

* Uncorrelated scalar subqueries in any place that a value expression can be used.
* Correlated scalar subqueries in WHERE clauses.
* EXISTS, ANY / ALL, and IN subqueries. These subqueries can be correlated or uncorrelated.

The above list is not extensive, meaning there would be cases that do not comply with any of the list conditions but are still valid.

The following table shows a set of subquery patterns found in the input code that normally invalidate subqueries in Snowflake.

| **Subquery patterns** | **Instances** | **Views Impacted** | **Procedures Impacted** | **Functions Impacted** |
| --- | --- | --- | --- | --- |
| More than one item in the query column list | 0 | 0/43 | 0/19 | 0/22 |
| Subquery outside a WHERE clause | 0 | 0/43 | 0/19 | 0/22 |

### Cursors

### Cursor complex patterns

The following breakdown gives information about cursor complex patterns.

| **Pattern type** | **Instances** | **Procedures Impacted** | **Functions Impacted** |
| --- | --- | --- | --- |
| Fetch inside loops  (SSC-PRF-0003) | 0 | 0 | 0 |
| Nested Cursors  (SSC-PRF-0005) | 0 | 0 | 0 |

##### Cursor relevant information

The following breakdown provides more relevant information that might be helpful to better understand the migration.

| **Cursor Definitions** | **Instances** | **Procedures Impacted** | **Functions Impacted** |
| --- | --- | --- | --- |
| Basic Cursor | 0 | 0 | 0 |
| Cursor FOR UPDATE | 0 | 0 | 0 |
| Cursor with variables | 0 | 0 | 0 |

| **Cursor Usages** | **Instances** | **Procedures Impacted** | **Functions Impacted** |
| --- | --- | --- | --- |
| Fetch | 0 | 0 | 0 |

### 

### Statements with usages of dynamic SQL

Tables created with dynamic SQL: 0

Views created with dynamic SQL: 0

| **Statements** | **Instances** | **Procedures Impacted** | **Functions Impacted** |
| --- | --- | --- | --- |
| EXEC and EXECUTE | 0 | 0 | 0 |
| INSERT INTO with EXECUTE | 0 | 0 | 0 |

Total statements with usages of dynamic SQL: 0

### 

### Transactions

Number of transactions with nested transactions: 0

Number of nested transactions in transactions: 0

In SQL Server, transactions can be nested and are considered a complex pattern. This means that it is possible to start a new transaction within an existing transaction, and it is not possible to have nested transactions in Snowflake. For more information, please refer to [SSC-EWI-TS0009.](https://docs.snowconvert.com/sc/general/technical-documentation/issues-and-troubleshooting/conversion-issues/sql-server/ssc-ewi-ts0009)

## GLOSSARY

Visit the [glossary](https://docs.snowconvert.com/sc/general/review-results/reports#glossary) to understand the terminology used in multiple report documents generated by SnowConvert.