

SNOWPARK MIGRATION ACCELERATOR

Version V8.00.019  
Snowpark Library for Python V1.27.000

Snowpark Library for Pandas V1.27.000

Snowpark Library for Scala and Java V1.14.000

### The Snowpark Migration Accelerator (SMA)

The SMA is a tool produced by Snowflake to assess and convert elements of a codebase running with Spark to Snowpark. This report contains a detailed summary of the assessment information available, but note that there is more information available in the output of the tool. Visit [the SMA documentation](https://docs.snowconvert.com/sma/user-guide/assessment/output-reports) to get a complete list of assessment information available.

The Snowpark Migration Accelerator (SMA) is not an official product of Snowflake Inc. and is not part of the Snowflake Service. Snowpark Migration Accelerator (SMA) is provided under its own terms and is “AS IS”. Snowflake’s support team does not provide support and is excluded from the support and service level obligations otherwise applicable to the Snowflake Service.

To learn more, visit [the SMA documentation site](https://docs.snowconvert.com/sma) or reach out to [sma-info@snowflake.com](mailto:sma-info@snowflake.com).

### 

### Execution Summary

|  |  |
| --- | --- |
| **Timestamp** 2025/07/15 11.20.17 | **Customer**  muhammad.ardiyan@snowflake.com  Snowflake |
| **Execution ID** b1fb77a9-958a-495d-82c1-be5dfafc0af9 | **Code Analyzed** 100% of your workload was successfully analyzed. [Learn more](https://docs.snowconvert.com/sma/user-guide/assessment/output-reports/curated-reports) |

## Readiness Scores Summary

The Snowpark Migration Accelerator (SMA) outputs a series of scores designed to help understand how ready a workload is for migration. The scores are given below along with more information on how to interpret each score.

|  |  |  |
| --- | --- | --- |
| [Spark API](#bookmark=id.gjdgxs) | 97.33% | ✅ |

|  |  |  |
| --- | --- | --- |
| [Third-Party Libraries](#bookmark=id.2et92p0) | 86.24% | ⚠️ |

|  |  |  |
| --- | --- | --- |
| [SQL](#bookmark=id.3dy6vkm) | - | ✅ |

*How to read through the score(s)*   
The SMA Readiness Scores are constantly evolving and more can be found out [in the SMA documentation](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores). Each score reports a red, yellow, or green status. Think of these scores as a stoplight. For each color:

* **Red** - there is an issue that has a significant impact on the migration or the tool’s ability to report on this codebase. Stop at this score, read the action steps, and act immediately to resolve.
* **Yellow** - there is an issue that might have a significant impact on the migration. Read the action steps, ensure you fully understand the impact of the yellow result, and move on to the next score when ready.
* **Green** - the tool did not detect a significant blocker for the migration of this codebase. This does not mean that the code is immediately ready to migrate. Read the action steps and proceed.

Each score will give you a number, a red/yellow/green status (described above), and a suggestion on what to do next. It is strongly advised to:

* **Go through the scores in order** - once you hit a red score, attempt to understand that problem immediately
* **Read all of the suggested next steps for each score** - regardless of the result (even the green results), that’s where you’ll find action items.

|  |  |  |
| --- | --- | --- |
| Spark API | 97.33% | ✅ |
| [Learn more](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores#spark-api-readiness-score) |  | |
| Identified Usages: | 1685 | |
| Usages Ready for Conversion: | 1640 |  |

##### What to do next?

|  |
| --- |
| **Green** - The majority of references to the Spark API are supported. This indicates a workload that is a good candidate for a migration. If the other indicators are green, it might be time to run a quick Proof of Concept. |

##### 

##### Understanding the Spark API Readiness Score

The Spark API Readiness Score measures how many of the references to the Spark API in the workload can be converted to the Snowpark API. The percentage shown is the percentage that can be migrated to Snowpark. This is the original “Readiness Score” that was produced by the SMA. More information can be found [in the SMA Documentation](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores).

|  |  |  |
| --- | --- | --- |
| Third-Party Libraries | 86.24% | ⚠️ |
| [Learn more](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores#third-party-api-readiness-score) |  | |
| Identified Library Calls: | 109 |  |
| Library Calls Supported in Snowpark: | 94 |  |

##### What to do next?

|  |
| --- |
| **Yellow** - There is at least 1 unrecognized or unsupported third party library being used in this database. In the output reports folder, check the import calls marked as unsupported. These will help you understand what to do next. |

##### 

##### Understanding the Third-Party Libraries Readiness Score

The Third-Party Library Indicator reports the percentage of imported libraries in this workload are categorized as supported in Snowflake. This is based on ALL counted library import statements, not just the unique ones. More information can be found [in the SMA Documentation](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores).

|  |  |  |
| --- | --- | --- |
| SQL | - | ✅ |
| [Learn more](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores#sql-readiness-score-levels) |  | |
| Identified Usages: | - | |
| Usages Ready for Conversion: | - |  |

##### What to do next?

|  |
| --- |
| **Green** - The majority of SQL present in this codebase is supported in either natively supported in Snowflake or can be automated by the SMA. No conversion is 100%, so there will still be SQL to convert, but this workload is nearly ready to Snowflake. |

##### Understanding the SQL Readiness Score

The SQL Readiness Score reports the percentage of SQL elements supported in Snowflake. This is only calculated on known SQL elements. There are still unknown elements that are not accounted for in this metric. More information can be found [in the SMA Documentation](https://docs.snowconvert.com/sma/user-guide/assessment/readiness-scores).

## File Summary

The Snowpark Migration Accelerator (SMA) scans all files that are present in your source codebase. This includes files that may not be code files or notebooks. In this section are a series of tables designed to show the type of files that are present in the scanned codebase.

### File Type Summary

Breakdown of each file type (based on extension) that was found in the scanned codebase.

| **File Type** | **Lines of Code** | **File Count** | **Percentage of Total Files** |
| --- | --- | --- | --- |
| Python | - | - | - |
| Scala | - | - | - |
| SQL | - | - | - |
| HiveSql | - | - | - |
| Notebook | 833 | 3 | 75.00% |
| Other\* | N/A | 1 | 25.00% |
| **Total** | **833** | **4** | **100.00%** |

*\* Other refers to files not recognized as code or notebook*

### File Extension Summary

Breakdown of each filetype (based on extension) that was found in the scanned codebase.

|  |  |  |  |
| --- | --- | --- | --- |
| **File Extension** | **Lines of Code** | **File Count** | **Percentage of Total Files** |
| ipynb | 833 | 3 | 75.00% |
| Other\* | N/A | 1 | 25.00% |
| **Total** | **833** | **4** | **100.00%** |

*\* Other refers to extensions not recognized as code*

### Code File Sizing

The Code File Sizing table provides an overview of the potential complexity of migrating code files. Larger files may pose greater challenges during conversion. This table applies sizing only to Python and Scala files, excluding SQL, HQL, and notebooks are handled in a separate section.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sizing Category** | **Abbreviation** | **File Count** | **Percentage of all Files** |
| Extra Small (less than 50 LOC) | XS | - | - |
| Small (less than 200 LOC) | S | - | - |
| Medium (less than 500 LOC) | M | - | - |
| Large (greater than 500 LOC) | L | - | - |
| Extra Large (greater than 1000 LOC) | XL | - | - |
| **Total** | - | - | - |

### Notebook Stats by Language

| **Source** | **Cells** | **LOC** |
| --- | --- | --- |
| Python | 66 | 833 |
| Scala | 0 | 0 |
| Sql | 0 | 0 |
| **Total** | **66** | **833** |

### Notebook Sizing by Language

Like code files, notebooks can be grouped into buckets based on size. While notebooks may have more than one language in them, the categories below show the predominant language for each one.

| **Sizing Category** | **Python Notebooks** | **Scala Notebooks** | **SQL Notebooks** |
| --- | --- | --- | --- |
| Extra Small | 0 | 0 | 0 |
| Small | 0 | 0 | 0 |
| Medium | 3 | 0 | 0 |
| Large | 0 | 0 | 0 |
| Extra Large | 0 | 0 | 0 |
| **Total** | **3** | **0** | **0** |

### Notebooks with DBX Elements

The Notebooks with DBX Elements table provides an overview of the usages of dbutils and magic commands. Also, the percentage of each element across all notebooks.

| **Element** | **Usages** | **Percent of all Notebooks with this dbx elements usages** |
| --- | --- | --- |
| **Total** | **-** | **-** |

## Spark API Summary

The SMA does a comprehensive analysis on references to the Spark API present in a codebase. This includes the location of each reference, how compatible each one is with the Snowpark API, and considerably more. A summary of the analysis on the Spark API can be found in the following tables.

### Files with Spark Usages

If code files have been found in this execution, this does not mean that all code files have references to the Spark API. Some or even all of the files found may have zero references to the Spark API. This does not mean that this workload is or is not compatible with Snowpark. It simply means that there is no code referencing the Spark API. This table breaks this down by technology

| **File Type** | **Files with Spark** | **All Files** | **Percentage of Files with Spark** |
| --- | --- | --- | --- |
| Python | - | - | - |
| Scala | - | - | - |
| Notebook | 3 | 3 | 100.00% |
| **Total** | **3** | **3** | **100.00%** |

NOTE: There may be additional files in the file summary that are not included in the above table. This could be sql or other files that would not possibly have references to the Spark API. This table is limited only to files that could have references to the Spark API.

### Files with Spark Usages by Support Status

Each reference to the Spark API is categorized as supported or not supported. Supported usages are references to the Spark API that have a known conversion to the Snowpark API. Unsupported usages do not yet have an equivalency in the Snowpark API.

| **File Type** | **All Files** | **Files w/ Supported** | **Files w/ Unsupported** | **Percentage of Files w/ Supported** | **Percentage of Files w/ Unsupported** |
| --- | --- | --- | --- | --- | --- |
| Python | - | - | - | - | - |
| Scala | - | - | - | - | - |
| Notebook | 3 | 3 | 3 | 100.00% | 100.00% |
| **Total** | **3** | **3** | **3** | **100.00%** | **100.00%** |

### Spark API Usage Summaries

Spark usages are references in the source codebase to the Spark API. There are several ways that the SMA categorizes these usages. This table summarizes by the category type present in the Spark API. Each category is given the count of each type by supported or not supported. The sum of supported usages divided by all usages is the Spark API Readiness Score. The final Spark API Readiness Score is calculated at the bottom of this table, and is the one reported by the SMA.

### 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Category** | **Python Supported Usages** | **Python Unsupported Usages** | **Scala Supported Usages** | **Scala Unsupported Usages** | **Readiness Score** |
| column | 182 | - | - | - | 100.00% |
| dataframe | 276 | - | - | - | 100.00% |
| functions | 963 | 4 | - | - | 99.59% |
| sparksession | 34 | - | - | - | 100.00% |
| types | 32 | - | - | - | 100.00% |
| group | 10 | - | - | - | 100.00% |
| readwriter | 27 | - | - | - | 100.00% |
| window | 101 | - | - | - | 100.00% |
| Others | 15 | 41 | - | - | 26.79% |
| **Total** | **1640** | **45** | **-** | **-** | **97.33%** |

|  |
| --- |
| Spark API Readiness Score: 97.33% (1640/1685) |

\*This total represents the Spark API Readiness Score, an estimate of the total quantity of references to the Spark API divided by the total number of supported references from Spark to the Snowpark API.

### Spark API Usages by Support Category

The SMA divides supported Spark elements into several categories based on the kind of mapping that is present from Spark to Snowpark. You can find more information about each of these categories in the appendixes at the end of this document.

| **Status Category\*** | **Count** | **Percentage of Total References** |
| --- | --- | --- |
| Direct | 1565 | 92.88% |
| Partial | - | - |
| Rename | 33 | 1.96% |
| Helper | - | - |
| Transformation | 42 | 2.49% |
| WorkAround | - | - |
| NotSupported | 42 | 2.49% |
| Pending | - | - |
| NotDefined | 3 | 0.18% |
| **Total** | **1685** | **-** |

\* Check **appendix A** for more information.

## Pandas API Summary

For executions with Python, using the Pandas API for constructing, manipulating, and analyzing data frames is incredibly common. The SMA does a comprehensive analysis on references to the Pandas API present in a codebase. This includes the location of each reference, how compatible each one is with the Snowpark implementation of Pandas, and considerably more. A summary of the analysis on the Pandas API can be found in the following tables.

### 

### Files with Pandas Usages

If code files have been found in this execution, this does not mean that all code files have references to the Spark API. Some or even all of the files found may have zero references to the Spark API. This does not mean that this workload is or is not compatible with Snowpark. It simply means that there is no code referencing the Spark API. This table breaks this down by technology

|  |  |  |  |
| --- | --- | --- | --- |
| **File Type** | **Files With Pandas** | **All Files** | **Percentage of Files with Pandas** |
| Python | - | - | - |
| Notebook | 3 | 3 | 100% |
| **Total** | **3** | **3** | **100%** |

NOTE: There may be additional files in the file summary that are not included in the above table. This could be sql or other files that would not possibly have references to the Spark API. This table is limited only to files that could have references to the Spark API.

### Pandas API Usage Summary

Pandas usages are references in the source codebase to the Pandas API. These are divided into several categories based on the Pandas library. Supported usages are references to the Pandas API that have a known conversion to the Snowpark API. Additional usages will be grouped into a row called ‘Others’. Unsupported usages do not yet have an equivalency in the Snowpark API.

|  |  |  |
| --- | --- | --- |
| **Pandas API Category** | **Pandas API Usages** | **Percentage of all Python Files** |
| pandas.core | 7 | 66.67% |
| pandas.io | 6 | 66.67% |
| Others | 35 | 100% |
| **Total** | **48** | **-** |

## 

## Import Reference Summary

The Snowpark Migration Accelerator (SMA) catalogs import calls to all other libraries present in the scanned codebase. It first categorizes each of these as built-in, supported in Snowpark, or other (meaning it’s usually a user created package created elsewhere in the codebase). Each of these is shown in detail in the output inventory files generated by the SMA.

### Third Party Import Call Summary

The presence of a lot of third party libraries can have an impact on the migration. The most common references to an external library in this codebase can be found below. Note that this list will not include built-ins, calls to the Spark API, or references to another user created package. These are only references to known third party imports. (This table will only show the top 20 packages. Additional packages will be grouped into a row with a package called ‘Other’. Specific information about these other packages can be found in the inventory files generated by the SMA.)

|  |
| --- |
| Total Count of Import Calls: **109** |

### 

| **Packages** | **Supported in Snowpark** | **Count of Files with Import** | **Percent of all Code Files with this Import Call** |
| --- | --- | --- | --- |
| h3 | No | 6 | 5.50% |
| h3\_pyspark | No | 3 | 2.75% |
| sedona | No | 3 | 2.75% |
| sklearn | No | 3 | 2.75% |
| pandas | Yes | 9 | 8.26% |
| numpy | Yes | 6 | 5.50% |
| geopandas | Yes | 3 | 2.75% |
| geopy | Yes | 3 | 2.75% |
| pyproj | Yes | 3 | 2.75% |
| shapely | Yes | 3 | 2.75% |
| shapely.wkt | Yes | 3 | 2.75% |
| Others |  | 64 | 58.72% |
| **Total** |  | **109** | **-** |

## SQL Reference Summary

The Snowpark Migration Accelerator (SMA) does more than inventory APIs. It does the same for some supported dialects of SQL.

### SQL Usages by File Type

SQL can be found in Spark workloads in a variety of ways. You could have separate SQL files or notebooks with SQL cells. This table summarizes where SQL was successfully scanned based on the identified technology.

| **File Type** | **All Files** | **Files w/ SQL** | **Percentage of Files w/ SQL** |
| --- | --- | --- | --- |
| Python\* | - | - | - |
| Scala\* | - | - | - |
| SQL | - | - | - |
| HQL | - | - | - |
| Notebook\*\* | 3 | - | - |
| **Total** | **3** | **-** | **-** |

*\*Code files with embedded SQL are not recognized in this version by the SMA tool, so they will*

*be always zero.  
\*\*Notebooks with SQL cells.*

### SQL Usages by Support Status

The SMA determines whether a given SQL element is supported in Snowflake in a vacuum. This means that the table below lists “elements” based on whether or not they have an equivalent in Snowflake, not whether all of these elements work together in Snowflake just as they did in the source platform.

| **Category** | **Count of all Elements** | **Percentage of all Elements** |
| --- | --- | --- |
| Supported | 0 | - |
| Partially Supported\* | 0 | - |
| Not Supported | 0 | - |
| Not Categorized\*\* | 0 | - |
| **Total** | **0** | **-** |

*\*Partially supported elements are supported, but may have parameters used in the source that are not supported in Snowflake.   
\*\*SQL elements that are identified, but have not yet been categorized as supported or not supported.*

## Issue Summary

As a part of the conversion process, the Snowpark Migration Accelerator (SMA) generates issues to highlight any errors or warnings that may be helpful when completing the migration process. Each of these errors and warnings have an error code associated with them. Each error code has a link to the page in our documentation that describes the error and gives recommendations on what you can do to resolve this error.

### Issue Categorization

Issues in the accelerator are broken down into three categories:

| **Warnings** | **Conversion Issues** | **Parsing Issues** |
| --- | --- | --- |
| Code that is recognized and converted, but needs review. | Code that is recognized, but not converted. | Code that is not recognized and not converted. |

### Issue Summary

Below is a count of the issues present. You can find more information on each unique issue by clicking the link to each error code listed below.

|  | **Warnings** | **Conversion Issues** | **Parsing Issues** |
| --- | --- | --- | --- |
| # of issues | 37 | 70 | - |
| # of unique issues | 7 | 3 | - |

### Issue List

This is the full list of unique issue codes generated by the SMA for this execution. You can find the exact location of each issue in the issues.csv file output by the SMA.

### 

| **Code** | **Description** | **Instances** | **Severity** |
| --- | --- | --- | --- |
| SPRKPY1002 | Spark element is not supported | 42 | ConversionError |
| PNDSPY1003 | Package is not yet recognized | 19 | ConversionError |
| SPRKPY1068 | toPandas function that is not supported when it has columns of type ArrayType | 12 | Warning |
| SPRKPY1032 | Spark element is not defined | 9 | ConversionError |
| PNDSPY1002 | There are scenarios not supported | 6 | Warning |
| SPRKPY1080 | The value of SparkContext is replaced with 'session' variable. | 6 | Warning |
| SPRKPY1077 | There is a SQL embedded that can not be processed. | 4 | Warning |
| SPRKPY1060 | Spark element requires assistance | 3 | Warning |
| SPRKPY1073 | Udf function without parameters or return type is not supported | 3 | Warning |
| SPRKPY1091 | Column name is in upper case and wrapped in double quotes. | 3 | Warning |

Note: There might be a mismatch between the issues listed in the table above and those that appear in the output source code. There are two main reasons for this, a certain amount of issues will not appear in the output source code and issues that may be shown several times in the table, but in the source code they only appear once, since they belong to the same sentence.

## APPENDIXES

### APPENDIX A: SPARK REFERENCE CATEGORIES

The Snowpark Migration Accelerator (SMA) divides supported Spark elements into several categories based on the kind of mapping that is present from Spark to Snowpark. Below is a summary of each of the categories that the accelerator outputs to describe the translation of each Spark reference, along with a description, example, whether the tool can automatically convert the reference (Tool Supported), and whether it’s currently possible to replicate the functionality (manually or automatically) in Snowpark. Note that these categorizations are included in the Spark API Usages by Support Category section of this report.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Direct | **Description** | Direct translation. The same function exists in PySpark and Snowpark with no change needed. | | | |
| **Snowpark Supported** | TRUE | | | |
| **Tool Supported** | TRUE | | | |
| **Spark Example** | col("col1") |  |  |  |
| **Snowpark Example** | col("col1") | | | |
| Rename | **Description** | The function from PySpark exists in Snowpark, but there is a rename that is needed. | | | |
| **Snowpark Supported** | TRUE | | | |
| **Tool Supported** | TRUE | | | |
| **Spark Example** | orderBy("date") | | | |
| **Snowpark Example** | sort("date") | | | |
| Helper | **Description** | This function has a small difference in Snowpark that can be addressed by creating a functionally equivalent function to resolve the difference. | | | |
| **Snowpark Supported** | TRUE | | | |
| **Tool Supported** | TRUE | | | |
| **Spark Example** | instr(str, substr) | | | |
| **Snowpark Example** | Create a helper function named instr with a identical signature as the PySpark function, like:  def instr(source: str, substr: str) => str :   return charindex(substr, str) | | | |
| Transformation | **Description** | The function is completely recreated to a functionally equivalent function in Snowpark, but doesn't resemble the original function. This can include calling several functions or adding multiple lines of code. | | | |
| **Snowpark Supported** | TRUE | | | |
| **Tool Supported** | TRUE | | | |
| **Spark Example** | col1 = col("col1") col2 = col("col2") col1.contains(col2) | | | |
| **Snowpark Example** | col1 = col("col1") col2 = col("col2") from snowflake.snowpark.functions as f f.contains(col, col2) | | | |
| WorkAround | **Description** | This category is employed when the tool cannot convert the PySpark element but there’s a known manual workaround to fix the conversion (the workaround is published in the tool’s documentation). | | | |
| **Snowpark Supported** | TRUE | | | |
| **Tool Supported** | FALSE | | | |
| **Spark Example** | instr(str, substr) | | | |
| **Snowpark Example** | charindex(substr, str) | | | |
| NotSupported | **Description** | NotSupported refers to any function that the tool cannot currently convert from PySpark because there's no applicable equivalent in Snowflake. And error message will be added to the output code. | | | |
| **Snowpark Supported** | FALSE | | | |
| **Tool Supported** | FALSE | | | |
| **Spark Example** | df:DataFrame = spark.createDataFrame(rowData, columns) df.alias("d") | | | |
| **Snowpark Example** | df:DataFrame = spark.createDataFrame(rowData, columns) # EWI: SPRKPY11XX => DataFrame.alias is not supported # df.alias("d") | | | |
| NotDefined | **Description** | Any detected usage of a Pyspark element that is not yet in the tool's conversion database. These elements will be marked for inclusion in a future version of the tool. | | | |
| **Snowpark Supported** | FALSE | | | |
| **Tool Supported** | FALSE | | | |
| **Spark Example** | N/A | | | |
| **Snowpark Example** | N/A | | | |