

SNOWPRO® SPECIALTY: SNOWPARK EXAM STUDY GUIDE

Last Updated: August 22, 2025



EXAM STUDY GUIDE TABLE OF CONTENTS:

SNOWPRO® SPECIALTY: SNOWPARK STUDY GUIDE OVERVIEW.....3
RECOMMENDATIONS FOR USING THE GUIDE..... 3
SNOWPRO® SPECIALTY: SNOWPARK EXAM CONTENT AND FORMAT..... 4
 CERTIFICATION OVERVIEW..... 4
SNOWPRO® SPECIALTY: SNOWPARK PREREQUISITE.....5
 STEPS TO SUCCESS.....5
EXAM DOMAINS AND WEIGHTINGS..... 5
 Domain 1.0: Snowpark Concepts..... 6
 Domain 2.0: Snowpark API for Python..... 7
 Domain 3.0: Snowpark for Data Transformations..... 9
 Domain 4.0: Snowpark Performance Optimization..... 11
SNOWPRO® SPECIALTY: SNOWPARK SAMPLE QUESTIONS..... 11
MAINTAINING YOUR CERTIFICATION..... 12

SNOWPRO® SPECIALTY: SNOWPARK STUDY GUIDE OVERVIEW

This is a self-learning guide that will outline the Snowflake domains, objectives, and topics that will be covered on the **SnowPro Specialty: Snowpark Certification** exam. Use of this study guide does not guarantee certification success.

Holding the SnowPro Core Certification in good standing is a prerequisite for taking the Specialty: Snowpark Certification Exam.

For an overview and more information on available SnowPro Certifications, please navigate [here](#).

RECOMMENDATIONS FOR USING THE GUIDE

This guide will show the Snowflake topics and subtopics covered on the exam. Following the topics will be additional resources consisting of documentation, blogs, and/or exercises to help you understand Snowpark.

Estimated length of study time to complete the guide: 10 – 13 hours

Some links may have more value for your exam preparation than others, depending on your experience. The same amount of time should not be spent on each link. Some links may appear in more than one domain.

SNOWPRO® SPECIALTY: SNOWPARK EXAM CONTENT AND FORMAT

CERTIFICATION OVERVIEW

The **SnowPro Specialty: Snowpark Certification Exam** tests specialized knowledge, skills, and best practices used to build Snowpark DataFrame data solutions in Snowflake including key concepts, features, and programming constructs. The exam will assess skills through scenario-based questions, interactive questions, and real-world examples.

This certification will test the candidate's ability to:

- Perform data transformations using Snowpark DataFrame functions.
- Query data sources as Snowpark DataFrame objects.
- Connect to Snowflake using a Snowpark Session object.
- Process results client-side or persist results in Snowflake through Snowpark DataFrame actions.
- Design a sequence of operations or conditional logic with Snowpark stored procedures.
- Create shareable and reusable code as Snowpark User-Defined Functions (UDFs).
- Conduct troubleshooting and telemetry observability while writing code.

Target Audience:

We recommend that individuals have 1+ years of Snowpark experience with Snowflake, in an enterprise environment. The candidate is expected to have knowledge of the Snowpark API for Python and Snowpark's client-side and server-side capabilities, as well as experience with data migration

In addition, successful candidates may have:

- Advanced proficiency writing code in Python.
- Advanced proficiency writing code in PySpark.

This exam is designed for:

- Data Engineers
- Data Scientists
- AI or ML Engineers
- Data Application Developers
- Data Analysts with programming experience

SNOWPRO® SPECIALTY: SNOWPARK PREREQUISITE

Eligible individuals must hold an active SnowPro Core Certification or SnowPro Associate: Platform Certification.

STEPS TO SUCCESS

1. Review this Snowpark Exam Study Guide
2. Attend Snowflake's Instructor-Led [Snowpark DataFrame Programming Training Course](#)
3. Get hands-on experience with our [Level Up Snowpark Essentials Track](#)
4. [Review and study applicable white papers and documentation](#)
5. Get hands-on practical experience with relevant business requirements using Snowflake
6. [Attend Snowflake Webinars](#)
7. [Attend Snowflake Virtual Hands-on Labs](#) to gain practical experience
8. Practice with sample questions [here](#)
9. [Schedule your exam](#)
10. Take your exam!

EXAM DOMAINS AND WEIGHTINGS

The following table contains the domains and weightings covered on the exam. It is not a comprehensive listing of all the content that will be presented on the exam.

Domain	Domain Weightings
1.0 Snowpark Concepts	15%
2.0 Snowpark API for Python	30%
3.0 Snowpark for Data Transformations	35%
4.0 Snowpark Performance Optimization	20%



Domain 1.0: Snowpark Concepts

1.1 Outline Snowpark architecture

- Lazy evaluation
- Use of key objects
 - Snowpark DataFrames
 - User-Defined Functions (UDFs)
 - User-Defined Table Functions (UDTFs)
 - Stored procedures
 - File operations
- Types of libraries (DataFrames, Machine Learning)
 - Anaconda repository (Python packages directly into Snowflake)
 - Other third-party libraries (not managed by Anaconda repository)
- Client-side and server-side capabilities

1.2 Set-up Snowpark

- Installation
 - Versioning
 - Python environment
- Development environments
 - Third-party tools
 - Snowflake Notebooks
 - Jupyter Notebooks
 - Microsoft Visual Studio Code (VS Code)

Domain 1.0 Study Resources

Snowflake Documentation

[Snowflake.snowpark.DataFrame](#)
[Creating User-Defined Functions \(UDFs\) for DataFrames in Python](#)
[Performing an Action to Evaluate a DataFrame](#)
[Snowflake.snowpark.DataFrame.show](#)
[Snowpark ML Modeling: ML Model Development](#)

[Setting Up Your Development Environment for Snowpark Python](#)
[Creating a Session for Snowpark Python](#)

Reading Asset

[Snowflake-snowpark-python 1.22.1](#)



➤ Domain 2.0: Snowpark API for Python

2.1 Create and manage user sessions

- Account identifiers
- Parameters for the `CONNECT` function
- Authentication methods
 - Construct a dictionary
 - Key pair authentication
 - Snowflake CLI or `.env` parameters
- Session creation
- `SessionBuilder`
- Session methods
- Session attributes
- `AsyncJob`

2.2 Use Snowpark with unstructured data

- Read files with `SnowflakeFile` object
- Use UDFs and UDTFs to process files
- Use stored procedures to process files

2.3 Create Snowpark DataFrames

- Multiple methods to create Snowpark DataFrames
 - From Snowflake tables/views
 - From Python objects (list, dictionary)
 - From SQL statements
 - From files (JSON, CSV, Parquet, XML)
 - From pandas DataFrames
- Schemas (apply to DataFrames)

- Data types (for example, `IntegerType`, `StringType`, `DateType`)

2.4 Operationalize UDFs and UDTFs in Snowpark

- Create UDFs from files (locally, on a stage)
- Use Python modules (packaged Python code) with UDFs
- Write Python function to create UDFs and UDTFs
- Register UDFs and UDTFs (for example, `session.utf(...)`, `functions.utf(...)`)
- Secure UDFs and UDTFs
 - Use SQL to alter UDFs and UDTFs created with Snowpark
 - Grant access to UDFs and UDTFs to share code
 - Understanding how to grant object permissions so other Snowflake users can see and use the UDFs and UDTFs
- Data types (type hints vs. registration API)
 - Provide the data types as parameters when creating a UDF or UDTF to return as Python hints/specify them as part of the registration
- Compare scalar and vectorized operations

2.5 Operationalize Snowpark stored procedures

- Create stored procedures from files (locally, on stage)
- Write Python functions to power stored procedures
- Use Python modules (packaged code, Anaconda) with stored procedures
- Register stored procedures
- Make dependencies available to code
- Secure stored procedures
 - Use SQL to alter stored procedures created with Snowpark
 - Caller versus owner rights
- Use Snowpark Python stored procedures to run workloads
- Data types (type hints vs. registration API)
 - Provide the data types as parameters when creating a stored procedure to return as Python hints/specify them as part of the registration
- Create Directed Acyclic Graphs (tasks) executing stored procedures
 - Python API
- Bring Python modules (packaged code) to be used with UDFs
 - Stored procedures to enable reuse of code

Domain 2.0 Study Resources

Snowflake Documentation

[Snowflake.snowpark.stored_procedure.StoredProcedureRegistration](#)

[Connecting to Snowflake with the Python Connector](#)

[Snowflake.snowpark.Session.createDataFrame](#)

[Snowflake.snowpark.DataFrameReader](#)

[Creating User-Defined Table Functions \(UDTFs\) for DataFrames in Python](#)

[Multi-factor authentication \(MFA\)](#)

[Snowflake.snowpark.DataFrame.sort](#)

[Python UDF handler examples](#)

[Creating User-Defined Functions \(UDFs\) for DataFrames in Python](#)

[Creating User-Defined Functions \(UDFs\) for DataFrames in Scala](#)

[Snowflake.snowpark.files.SnowflakeFile.open](#)

[Snowflake.snowpark.Session.add_import](#)

[Creating Stored Procedures for DataFrames in Python](#)

[Snowflake.snowpark.functions.sproc](#)

[Creating a Session for Snowpark Python](#)

[Managing Snowflake tasks and task graphs with Python](#)

[Calling Functions and Stored Procedures in Snowpark Python](#)

Reading Asset

[Python Built-in Functions](#)

[UDFs in Snowflake Snowpark \(Article\)](#)



➤ Domain 3.0: Snowpark for Data Transformations

3.1 Apply operations for filtering and transforming data

- Use scalar functions and operators
- Sort and limit results
- Input/output (parameters)
- Snowpark DataFrames
- Columns
- Data type casting
- Rows and data extraction from a Rows object

3.2 Clean and enrich data using Snowpark for Python

- Perform joins
- Handle missing values
- Sample data

3.3 Perform aggregate and set-based operations on DataFrames

- Functions
- Window
- Grouping
- Table functions
- UDFs

3.4 Transform semi-structured data in DataFrames

- Traverse semi-structured data
- Explicitly cast values in semi-structured data
- Flatten an array of objects into rows
- Load semi-structured data into DataFrames

3.5 Persist the results of Snowpark DataFrames

- Create views from DataFrames
- Save DataFrame results as Snowflake tables
- Save DataFrame results as files in a stage

3.6 Perform DML operations using Snowpark DataFrames

- Delete data
- Update data
- Insert data
- Merge data

Domain 3.0 Study Resources

Snowflake Documentation

[Working with DataFrames in Snowpark Python](#)

[Snowflake.snowpark.Row.asDict](#)

[Snowpark Functions](#)

[SnowPark Dataframe](#)

[Snowpark Column](#)

[Snowpark Window](#)

[Snowpark Input/Output](#)

[Snowpark Table](#)

[Quick reference: Snowpark Java APIs for SQL commands](#)

[Querying Semi-structured Data](#)

Reading Assets

[Handling missing values with Snowpark for Python \(Article\)](#)

[Window Functions in Snowflake Snowpark \(Article\)](#)



➤ Domain 4.0: Snowpark Performance Optimization

4.1 Configure Snowpark-optimized warehouses

- Use cases for Snowpark-optimized virtual warehouses
- Modify Snowpark-optimized virtual warehouse properties
- Billing for Snowpark-optimized virtual warehouses
- When to scale up/down virtual warehouses

4.2 Enhance performance in Snowpark applications

- Materialize results (caching)
 - Caching DataFrames (using `.cache_result()`) and understanding why this is useful
 - Create a temporary table

- Vectorization
 - Understanding the difference between vectorized and scalar UDFs
 - Vectorized UDFs for batching
 - Snowpark DataFrames versus pandas on Snowflake
- Synchronous versus asynchronous calls
 - `Block` parameter

4.3 Troubleshoot common errors in Snowpark

- Event tables
- Snowpark Python local testing framework
- Writing tests (pytest)
- Query history (SQL equivalency to help identify bottlenecks)

Domain 4.0 Study Resources

Snowflake Documentation

[Snowpark-optimized warehouses](#)

[CREATE WAREHOUSE](#)

[Snowflake.snowpark.DataFrameWriter.copy_into_location](#)

[Using Persisted Query Results](#)

[Event table overview](#)

[Troubleshooting logging and tracing](#)

Reading Assets

[Snowflake Shorts: Snowpark Optimized Warehouses \(Article\)](#)

SNOWPRO® SPECIALTY: SNOWPARK SAMPLE QUESTIONS

1. Which workload would benefit the MOST from using a Snowpark-optimized virtual warehouse?
 - A. Machine learning training
 - B. Machine learning inference
 - C. Registering a model into the Snowflake Model Registry
 - D. Creating a compute pool in Snowpark Container Services
2. A Snowpark Specialist wants to create a Python User-Defined Function (UDF) and operationalize it in Snowflake. The function will not use the `IMPORTS` clause.

What can the Specialist do with this Python UDF in Snowflake?

- A. Share the Python UDF directly.
 - B. Share a view that calls the Python UDF.
 - C. Access the session object within the Python UDF.
 - D. Grant the `USAGE` privilege on the Python UDF to a role.
3. How can a Snowpark Specialist summarize the sales quantity by product, given a `DataFrame` containing product sales quantities in columns named `product_id` and `quantity`?

- A. `df.sum("quantity").group_by("product_id")`
- B. `df.summarize("quantity").over("product_id")`
- C. `df.group_by("product_id").agg(sum("quantity"))`
- D. `df.agg("quantity", type="sum").group_by("product_id")`

4. A Snowpark Specialist needs to define a Python function to be used as a stored procedure.

What should they consider?

- A. The function must always return a `Snowpark DataFrame`.
- B. The first parameter for the function must be a `Session` class object.
- C. The `@sproc` decorator must always be used before the function definition.
- D. The `pandas DataFrame` or `pandas Series` object can be used as parameters for the function.

5. A Snowpark Specialist developed an application that uses Snowpark for Python to interact with Snowflake tables. Users are reporting constant Multi-Factor Authentication (MFA) alerts.

What is the MOST secure method of reducing the MFA requests?

- A. Create a `NETWORK POLICY` for the affected users.
- B. Set the account parameter `ALLOW_CLIENT_MFA_CACHING` to `TRUE`.
- C. Allow users to add a passcode as part of their Snowpark session creation.
- D. Disable MFA temporarily for affected users using the parameter `DISABLE_MFA`.

Correct responses for sample questions:

1: a, 2: d, 3: c, 4: b , 5: b

MAINTAINING YOUR CERTIFICATION

All Snowflake Certifications expire two (2) years after your certification issue date.

SnowPro Certifications can now be recertified through the Snowflake Continuing Education (CE) program which includes these options -

- Completion of eligible Snowflake [Instructor Led \(ILT\) Training Courses](#)
- Earning of an equivalent or higher-level SnowPro Certification

Note: You must have a valid Certification to participate in the Continuing Education (CE) program.

The information provided in this guide is provided for your internal purposes only and may not be provided to third parties.

IN ADDITION, THIS STUDY GUIDE IS PROVIDED “AS IS”. NEITHER SNOWFLAKE NOR ITS SUPPLIERS MAKES ANY OTHER WARRANTIES, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, TITLE, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.