

Certification Questions Practice Set (Data Movement)

1. Which function should a data engineer use to recursively resume all tasks in a chain of tasks rather than resuming each task individually (using ALTER TASK ... RESUME) ?
- A) SYSTEM\$TASK_DEPENDENTS
 - B) SYSTEM\$TASK_DEPENDENTS_ENABLE
 - C) SYSTEM\$TASK_DEPENDENTS_RESUME
 - D) SYSTEM\$TASK_RECURSIVE_ENABLE

Answer : B

Explanation : <https://docs.snowflake.com/en/sql-reference/sql/alter-task>

RESUME | SUSPEND

Specifies the action to perform on the task:

- **RESUME** brings a suspended task to the 'Started' state. Note that accounts are currently limited to a maximum of 30000 started tasks.
Before resuming the root task of your Task Graph, resume all child tasks. To recursively resume the root task's child tasks, use SYSTEM\$TASK_DEPENDENTS_ENABLE.
- **SUSPEND** puts the task into a 'Suspended' state.

-
2. Mark the correct privileges required to RESUME or SUSPEND a task in Snowflake

- A) OWNERSHIP
- B) EXECUTE
- C) OPERATE
- D) MODIFY

Answers : A, C

Explanation : <https://docs.snowflake.com/en/sql-reference/sql/alter-task>

Usage notes

- Resuming or suspending a task (using ALTER TASK ... RESUME or ALTER TASK ... SUSPEND, respectively) requires either the OWNERSHIP or OPERATE privilege on the task.

When a task is resumed, Snowflake verifies that the role with the OWNERSHIP privilege on the task also has the USAGE privilege on the warehouse assigned to the task, as well as the global EXECUTE TASK privilege; if not, an error is produced.

3. Mark all the incorrect statements related to TASKS

- A) The TASK_HISTORY table function returns all task executions done in the past 7 days
- B) The TASK_HISTORY table function returns all task executions done in the past 365 days
- C) The TASK_HISTORY view returns all task executions done in the past 365 days
- D) The TASK_HISTORY view has a limit of maximum 10,000 records being returned

Answers :

B, D

Explanation : https://docs.snowflake.com/en/sql-reference/functions/task_history

Usage notes

- This function returns results only for the ACCOUNTADMIN role, the task owner, or a role with the global MONITOR EXECUTION privilege. Note that unless a role with the MONITOR EXECUTION privilege also has the USAGE privilege on the database and schema that store the task, the DATABASE_NAME and SCHEMA_NAME values in the output are NULL.
- This function returns a maximum of 10,000 rows, set in the RESULT_LIMIT argument value. The default value is 100. To avoid this limitation, use the TASK_HISTORY view (Account Usage).
- When calling an information schema table function, the session must have an INFORMATION_SCHEMA schema in use **or** the function name must be fully qualified. For more information, see [Snowflake Information Schema](#).
- This function can return all executions run in the past 7 days or the next scheduled execution within the next 8 days.

4. Identify which of the statements are NOT TRUE with respect to transaction management of Snowpipe & COPY INTO constructs.

- A) Bulk data loads are always performed on a single transaction whereas Snowpipe data loads are combined into single OR multiple based on the number & size of files in each data load.
- B) Bulk data load requires a user specified warehouse whereas Snowpipe uses Snowflake provided compute configurations.
- C) Bulk data load is billed for the amount of time the warehouse is active whereas in Snowpipe the billing happens based on compute resources that Snowflake manages during the pipe execution.
- D) All transaction load history for BULK data load is maintained for 365 days whereas for Snowpipe it is there for 64 days.

Answer : D

Explanation : https://docs.snowflake.com/en/sql-reference/info-schema/load_history

LOAD_HISTORY view

This Information Schema view enables you to retrieve the history of data loaded into tables using the COPY INTO <table> command within the last 14 days. The view displays one row for each file loaded.

Note

This view does not return the history of data loaded using Snowpipe. For this historical information, query the COPY_HISTORY table function instead.

5. The COPY command supports several options for loading data files from a stage i.e.

I. By path

II. Specifying a list of specific files to load.

- III. Using pattern matching to identify specific files by pattern.
- IV. Organize files into logical paths that reflect a scheduling pattern.

Of the aforementioned options for identifying/specifying data files to load from a stage, which option in general is the fastest and best considerate?

- A. I
- B. II
- C. III
- D. IV

Answer : B

Explanation : In option 2, the search is fastest because the file names are explicitly being mentioned. Others mention a path where an additional overhead is required to complete the search

6. As a Data Engineer, you have a requirement to load a set of new product files containing product-relevant information into the Snowflake internal tables. Later, you analyzed that some of the source files are already loaded in one of the historical batches, and you have pre checked the metadata column LAST_MODIFIED date for a staged data file and found out that LAST_MODIFIED date is older than 64 days for a few files, and the initial set of data was loaded into the table more than 64 days earlier. Which one is the best approach to load source data files with expired load metadata along with a set of files whose metadata might be available to avoid data duplication?

- A. Since the initial set of data for the table (i.e., the first batch after the table was created) was loaded, we can simply use the COPY INTO command to load all the product files with the known load status irrespective of their column LAST_MODIFIED date values.
- B. The COPY command cannot definitively determine whether a file has been loaded already if the LAST_MODIFIED date is older than 64 days and the initial set of data was loaded into the table more than 64 days earlier (and if the file was loaded into the table, that also occurred more than 64 days earlier). In this case, to prevent accidental reload, the command skips the product files by default.
- C. Set the FORCE option to load all files, ignoring load metadata if it exists.
- D. To load files whose metadata has expired, set the LOAD_UNCERTAIN_FILES copy option to true.

Answer : D

Explanation : <https://docs.snowflake.com/en/user-guide/data-load-considerations-load>

Workarounds

To load files whose metadata has expired, set the `LOAD_UNCERTAIN_FILES` copy option to true. The copy option references load metadata, if available, to avoid data duplication, but also attempts to load files with expired load metadata.

Alternatively, set the `FORCE` option to load all files, ignoring load metadata if it exists. Note that this option reloads files, potentially duplicating data in a table.

7. Snowpipe API provides a REST endpoint for defining the list of files to ingest that informs Snowflake about the files to be ingested into a table. A successful response from this endpoint means that Snowflake has recorded the list of files to add to the table. It does not necessarily mean the files have been ingested.

What is the name of this endpoint?

- A. REST endpoints --> insertReport
- B. REST endpoints --> loadHistoryScan
- C. REST endpoints --> ingestFiles
- D. REST endpoints --> insertFiles

Answer : D

Explanation : <https://docs.snowflake.com/en/user-guide/data-load-snowpipe-rest-apis>

8. Streams cannot be created to query change data on which of the following objects? (Select all that apply)

- A. Standard tables, including shared tables.
- B. Views, including secure views.
- C. Directory tables

- D. Query log tables
- E. External tables

Answer - D

Explanation : <https://docs.snowflake.com/en/user-guide/streams-intro>

Streams can be created to query change data on the following objects:

- Standard tables, including shared tables.
- Views, including secure views
- [Directory tables](#)
- [Dynamic tables](#)
- [Iceberg tables](#) with [Limitations](#).
- [Event tables](#)
- [External tables](#)

9. Which of the following systems keeps the following characteristics?

- a. It will keep in all the raw data.
- b. Generally, the users of it are data scientists and data developers.
- c. Flat architecture
- d. Highly agile

- A. Data Warehouse
- B. Data Mart
- C. Data Lake
- D. Data Hub

Answer : A

10. Dominic, a Data Engineer, wants to resume the pipe named stalepipe3, which got stale after 14 days. To do the same, he called the SYSTEM\$PIPE_FORCE_RESUME function:

```
select  
system$pipe_force_resume('snowmydb.mysnowschemastalepipe3','staleness_check_override');
```

Let's say the pipe is resumed 16 days after it was paused. What will happen to the event notifications that were received on the first and second days after the pipe was paused?

- A. Snowpipe generally skips any event notifications that were received on the first and second days after the pipe was paused.
- B. Pipe maintains metadata history of files for 64 days, so in this scenario Snowpipe processes all the event notifications that were received for 16 days or so.
- C. Once the pipe got stale, all the events got purged automatically and the pipe needs to be recreated with modified properties.
- D. All the events get processed from day 1 if the PURGE properties in the PIPE object definition are set to be FALSE initially

Answer : A

Explanation : <https://docs.snowflake.com/en/user-guide/data-load-snowpipe-manage>

As an event notification received while a pipe is paused reaches the end of the limited retention period, Snowflake schedules it to be dropped from the internal metadata. If the pipe is later resumed, Snowpipe processes these older notifications on a best effort basis. Snowflake cannot guarantee that they are processed.

For example, if a pipe is resumed 15 days after it was paused, Snowpipe generally skips any event notifications that were received on the first day the pipe was paused (i.e. that are now more than 14 days old). If the pipe is resumed 16 days after it was paused, Snowpipe generally skips any event notifications that were received on the first and second days after the pipe was paused. And so on.

11. As part of table designing, a Data Engineer added a timestamp column that inserts the current timestamp as the default value as records are loaded into a table. The intent is to capture the time when each record was loaded into the table; however, the timestamps are earlier than the LOAD_TIME column values returned by COPY_HISTORY view (Account Usage).

What could be the reason for this issue?

- A. LOAD_TIME column values returned by COPY_HISTORY view (Account Usage) give the same time as returned by CURRENT_TIMESTAMP.
- B. CURRENT_TIMESTAMP values might be different due to queries being executed in a warehouse located in a different region.
- C. It might be possible that Cloud Provider hosted on Snowflake belongs to a region having server time zone lagging cluster time zone of the warehouse where queries get processed and committed.
- D. The reason is, CURRENT_TIMESTAMP is evaluated when the load operation is compiled in cloud services rather than when the record is inserted into the table (i.e., when the transaction for the load operation is committed).

Answer : D

Explanation :

12. John, a Data Engineer, has technical requirements to refresh the external tables' metadata periodically or in auto mode. Which approach should John take to meet this technical specification?

- A. John can use AUTO_REFRESH parameter if the underlying external cloud host supports this for external tables.
- B. He can create a task that executes an ALTER EXTERNAL TABLE ... REFRESH statement every 5 minutes.
- C. External tables cannot be scheduled via Snowflake Tasks; third-party tools/scripts provided by the external cloud storage provider need to be used.
- D. Snowflake implicitly takes care of this infrastructure need, as the underlying warehouse layer internally manages the refresh. No action is needed from John.

Answer : A

Explanation : <https://docs.snowflake.com/en/sql-reference/sql/alter-external-table>

`AUTO_REFRESH = < TRUE / FALSE >`

Specifies whether Snowflake should enable triggering automatic refreshes of the external table metadata when **new or updated** data files are available in the named external stage specified in the `WITH] LOCATION =` setting.

Note

- You **must** configure an event notification for your storage location to notify Snowflake when new or updated data is available to read into the external table metadata. For more information, see the instructions for your cloud storage service:
 - **Amazon S3** [Refreshing external tables automatically for Amazon S3](#)
 - **Google Cloud Storage** [Refreshing external tables automatically for Google Cloud Storage](#)
 - **Microsoft Azure** [Refreshing external tables automatically for Azure Blob Storage](#)
- This parameter is **not** supported by partitioned external tables when partitions are added manually by the object owner (i.e. when `PARTITION_TYPE = USER_SPECIFIED`).
- Setting this parameter to TRUE is **not** supported for external tables that reference data files stored on an [S3-compatible external stage](#).

TRUE

Snowflake enables triggering automatic refreshes of the external table metadata.

13. Mark the incorrect statement in case a Data Engineer is using the COPY INTO `<table>` command to load data from files into Snowflake tables?

- A. For data loading of files with semi-structured file formats (JSON, Avro, etc.), the only supported character set is UTF-16.
- B. For loading data from all semi-structured supported file formats (JSON, Avro, etc.), as well as unloading data, UTF-8 is the only supported character set.
- C. For local environments, files are first copied ("staged") to an internal (Snowflake) stage, then loaded into a table.
- D. UTF-32 & UTF-16 both encoding character sets are supported for loading data from delimited files (CSV, TSV, etc.).

Answer : A

Explanation : <https://docs.snowflake.com/en/user-guide/intro-summary-loading>

File encoding	File format-specific	For delimited files (CSV, TSV, etc.), the default character set is UTF-8. To use any other characters sets, you must explicitly specify the encoding to use for loading. For the list of supported character sets, see Supported Character Sets for Delimited Files (in this topic).
		For semi-structured file formats (JSON, Avro, etc.), the only supported character set is UTF-8.
		Snowflake doesn't support loading data from tar (tape archive) files.

14. Tasks may optionally use table streams to provide a convenient way to continuously process new or changed data. A task can transform new or changed rows that a stream surfaces. Each time a task is scheduled to run, it can verify whether a stream contains change data for a table and either consume the change data or skip the current run if no change data exists.

Which system function can be used by a Data Engineer to verify whether a stream contains changed data for a table?

- A. SYSTEM\$STREAM_HAS_CHANGE_DATA
- B. SYSTEM\$STREAM_CDC_DATA
- C. SYSTEM\$STREAM_HAS_DATA
- D. SYSTEM\$STREAM_DELTA_DATA

Answer : C

15. Emma is tasked with loading data from various sources into Snowflake and needs to ensure that duplicate records are not loaded. What is the best approach to handle this scenario?

- A. Use the COPY INTO command with the ON_ERROR option set to SKIP_FILE.
- B. Perform a merge operation using the MERGE INTO command to update existing records and insert new ones.
- C. Use the INSERT INTO command with a UNIQUE constraint on the table.
- D. Load data into a staging table first and then use the DELETE command to remove duplicates.

Answer : B

16. Alex is implementing a data pipeline that needs to handle late-arriving data without reprocessing the entire dataset. Which Snowflake feature should he use?

- A. Streams
- B. Materialized Views
- C. Secure Views
- D. Data Cloning

Answer : Streams

Explanation : Materialized and secure views and data cloning are not there for data handling

17. As a part of data ingestion workflow, you've received a request to ingest Excel files being dumped in your external stage.

Excel, being not supported as a native file format, you've decided to write a procedure in Snowpark which takes file location as input and loads the data in the Excel sheet into a Snowflake table. The Excel file access should be secure and should not work after some time. Which of the following SQL functions you'd use to generate the URL?

- A) BUILD_STAGE_FILE_URL
- B) BUILD_SCOPED_FILE_URL.
- C) GET_STAGE_LOCATION
- D) GET_PREIGNED_URL

Answer : B

Explanation :

BUILD_SCOPED_FILE_URL

Generates a scoped Snowflake file URL to a staged file using the stage name and relative file path as inputs.

A scoped URL is encoded and permits access to a specified file for a limited period of time. The scoped URL in the output is valid for the caller until the persisted query result period ends (until the results cache expires). That period is currently 24 hours.

Call this SQL function in a query or view. You can also use this SQL function to pass a scoped URL to a user-defined function (UDF) or stored procedure.

18. A Data Engineer needs to load JSON output from some software into Snowflake using Snowpipe.

Which recommendations apply to this scenario? (Choose three.)


- A. Load large files (1 GB or larger).
- B. Ensure that data files are 100-250 MB (or larger) in size, compressed.
- C. Load a single huge array containing multiple records into a single table row.
- D. Verify each value of each unique element stores a single native data type (string or number).
- E. Extract semi-structured data elements containing null values into relational columns before loading.
- F. Create data files that are less than 100 MB and stage them in cloud storage at a sequence greater than once each minute.


Answers : B, D, E

Explanation :

For better pruning and less storage consumption, we recommend flattening your OBJECT and key data into separate relational columns if your semi-structured data includes:

- Dates and timestamps, especially non-ISO 8601 dates and timestamps, as string values
- Numbers within strings
- Arrays

Snowpipe recommends that data files be at least 10 MB in size, but the best cost-to-performance ratio is achieved with files between 100–250 MB. 

Here are some other tips for optimizing Snowpipe file sizes: 

Files need to be chunked to sizes between 100 - 250 MB. So, B is correct. A, C and F are surely wrong. We are left with B, D and E

19. A Data Engineer is working on a Snowflake deployment in AWS eu-west-1 (Ireland). The Engineer is planning to load data from staged files into target tables using the COPY INTO command.

Which sources are valid? (Choose three.)

- A. Internal stage on GCP us-central1 (Iowa)
- B. Internal stage on AWS eu-central-1 (Frankfurt)
- C. External stage on GCP us-central1 (Iowa)
- D. External stage in an Amazon S3 bucket on AWS eu-west-1 (Ireland)
- E. External stage in an Amazon S3 bucket on AWS eu-central-1 (Frankfurt)
- F. SSD attached to an Amazon EC2 instance on AWS eu-west-1 (Ireland)

Answers : B, D, E

Explanation :

A and C are definitely wrong because an AWS Snowflake deployment can't taken place on GCP

Also, F is wrong because the SSD attached to the EC2 instance can't be used as a stage. For AWS S3 bucket is required to set up stage

20) Which query will show a list of the 20 most recent executions of a specified task, MYTASK, that have been scheduled within the last hour that have ended or are still running?

- a.

```
select * from
table(information_schema.task_history(scheduled_time_range_start
=>dateadd('hour',-1,current_timestamp()), result_limit => 20,
task_name=>'MYTASK'));
```
- b.

```
select * from
table(information_schema.task_history(scheduled_time_range_start
=>dateadd('hour',-1,current_timestamp()), result_limit => 20,
task_name=>'MYTASK')) where query_id IS NOT NULL;
```
- c.

```
select * from
table(information_schema.task_history(scheduled_time_range_start
=>dateadd('hour',-1,current_timestamp()), result_limit => 20,
task_name=>'MYTASK')) where STATE IN ('EXECUTING', 'SUCCEEDED',
'FAILED');
```
- d.

```
select * from
table(information_schema.task_history(scheduled_time_range_end
```

```
=>dateadd('hour',-1,current_timestamp()), result_limit => 10,  
task_name=>'MYTASK')) where STATE IN ('EXECUTING', 'SUCCEEDED');
```

Answer : C

21. What is the purpose of the BUILD_STAGE_FILE_URL function in Snowflake?

- A. It generates an encrypted URL for accessing a file in a stage.
- B. It generates a staged URL for accessing a file in a stage.
- C. It generates a permanent URL for accessing files in a stage.
- D. It generates a temporary URL for accessing a file in a stage.

Answer : C

Explanation :

BUILD_STAGE_FILE_URL

Generates a Snowflake *file URL* to a staged file using the stage name and relative file path as inputs. A file URL permits prolonged access to a specified file. That is, the file URL does not expire.

22. Which statement will query the row level metadata columns for each record, stored in CSV files, on a stage?

1. `select * from @mystage (file_format => mycsvformat) t;`
2. `select $filename, $file_row_number, t.$1, t.$2, t.$3 from @mystage (file_format => mycsvformat) t;`
3. `select metadata$filename, metadata$file_row_number, t.$1, t.$2, t.$3 from @mystage (file_format => mycsvformat) t;`
4. `select t.$0, t.$1, t.$2, t.$3 from @mystage(file_format => mycsvformat) t;`

Answer : 3

Explanation :

A table has the following definition and sample data:

```
CREATE TABLE SAMPLE
```

```
(COL1 VARCHAR(1),
```

```
COL2 VARCHAR(10));
```

DATA TO BE LOADED - SAMPLE.CSV

ABC|XYZ

XYZ|1

1|"HELLO WORLD"

What statement will load the data without an error?

1. COPY INTO SAMPLE

```
FROM @~/SAMPLE.CSV
```

```
FILE_FORMAT = (TYPE=CSV DELIMITER='|')
```

2. COPY INTO SAMPLE

```
FROM (SELECT $1::VARCHAR, $2::VARCHAR FROM @~/SAMPLE.CSV)
```

```
FILE_FORMAT = (TYPE=CSV DELIMITER='|')
```

3. COPY INTO SAMPLE

```
FROM @~/SAMPLE.CSV
```

```
FILE_FORMAT = (TYPE=CSV DELIMITER='|'
```

```
FIELD_OPTIONALLY_ENCLOSED_BY='\"'
```

4. COPY INTO SAMPLE

```
FROM @~/SAMPLE.CSV
```

```
FILE_FORMAT = (TYPE=CSV DELIMITER='|' ENFORCE_LENGTH = FALSE)
```

Answer : 4

Explanation :

A Data Engineer has set up a continuous data pipeline using Snowpipe to load data into a table **MYTABLE**.

To see all errors that have occurred in the last hour, which of the following queries would need to be executed?

A. `select * from information_schema.copy_history`

`where table_name='MYTABLE' and start_time > DATEADD(hours, -1, current_timestamp());`

B. `select * from information_schema.copy_history`

`where table_name='MYTABLE' and start_time > DATEADD(hours, -1, current_timestamp());`

C. `select * from table(information_schema.copy_history(table_name=>'MYTABLE', start_time=> DATEADD(hours, -1, current_timestamp())));`

D. `select * from table(information_schema.load_history(table_name=>'MYTABLE', start_time=> DATEADD(hours, -1, current_timestamp())));`

Answer : C

Explanation :

Given the named stage:

`create or replace stage my_stage file_format = my_file_format;`

Which **COPY** command can be used to load files into the table **mytable**?

1. copy into mytable from @my_stage

```
file_format = my_file_format;
```

2. copy into mytable from @my_stage

```
file_format = (type = my_file_format);
```

3. copy into mytable from my_stage

```
file_format = (type = csv);
```

4. copy into mytable from @my_stage

```
file_format = (format_name = my_file_format);
```

Which metadata columns are added automatically to a stream? (Select THREE)

1. METADATA\$FILENAME
2. **METADATA\$ISUPDATE**
3. METADATA\$FILE_ROW_NUMBER
4. **METADATA\$ACTION**
5. **METADATA\$ROW_ID**
6. METADATA\$FILE_LAST_MODIFIED

When querying a stream to perform Change Data Capture (CDC), how would a Data Engineer identify the different types of changes in the data? (Select TWO).

1. `SELECT * FROM <Stream Name> WHERE SYSTEM$STREAM_HAS_DATA = 'True';`
2. `SELECT * FROM <Stream Name>`

`WHERE METADATA$ACTION = 'INSERT' AND METADATA$ISUPDATE = 'True';`

3. `SELECT * FROM <Stream Name> WHERE METADATA$ACTION = 'DELETE' AND METADATA$ISUPDATE = 'True';`
4. `SELECT * FROM <Stream Name> WHERE METADATA$ACTION = 'DELETE' AND METADATA$ISUPDATE = 'False';`
5. `SELECT * FROM <Stream Name>`

`WHERE METADATA$ACTION = 'INSERT' AND METADATA$ISUPDATE = 'False';`

Answer :

Explanation :

13. What is the BEST way to optimize a task that runs every 5 minutes in a warehouse designed to auto-suspend in 5 minutes?

1. Decrease the auto-suspend mode of the warehouse to 3 minutes.
2. Use the serverless task mode
3. It doesn't need any change; it's optimal.

Answer : 2

Explanation :

A task that runs every 5 minutes will frequently start and stop the warehouse, leading to inefficient use of resources and longer query execution times when the warehouse has to resume each time. Serverless tasks in Snowflake are not tied to a specific warehouse and can run without incurring the overhead of starting or suspending a warehouse, making them more efficient for frequent, short-running tasks.