

Ace ARA-C01 Certification with 116 Actual Questions [Q56-Q73]



Ace ARA-C01 Certification with 116 Actual Questions PASS Snowflake ARA-C01 EXAM WITH UPDATED DUMPS Q56.
You have create a task as below

```
CREATE TASK mytask1
```

```
WAREHOUSE = mywh
```

```
SCHEDULE = '&#8216;5 minute&#8217;';
```

```
WHEN
```

```
SYSTEM$STREAM_HAS_DATA('&#8216;MYSTREAM&#8217;);
```

```
AS
```

```
INSERT INTO mytable1(id,name) SELECT id, name FROM mystream WHERE METADATA$ACTION =  
&#8216;INSERT&#8217;;
```

Which statement is true below?

- * If SYSTEM\$STREAM_HAS_DATA returns false, the task will be skipped
- * If SYSTEM\$STREAM_HAS_DATA returns false, the task will still run
- * If SYSTEM\$STREAM_HAS_DATA returns false, the task will go to suspended mode

Q57. Which security, governance, and data protection features require, at a MINIMUM, the Business Critical edition of Snowflake?
(Choose two.)

- * Extended Time Travel (up to 90 days)
- * Customer-managed encryption keys through Tri-Secret Secure
- * Periodic rekeying of encrypted data
- * AWS, Azure, or Google Cloud private connectivity to Snowflake
- * Federated authentication and SSO

Q58. At which object type level can the APPLY MASKING POLICY, APPLY ROW ACCESS POLICY and APPLY SESSION POLICY privileges be granted?

- * Global
- * Database
- * Schema
- * Table

Explanation

The object type level at which the APPLY MASKING POLICY, APPLY ROW ACCESS POLICY and APPLY SESSION POLICY privileges can be granted is global. These are account-level privileges that control who can apply or unset these policies on objects such as columns, tables, views, accounts, or users. These privileges are granted to the ACCOUNTADMIN role by default, and can be granted to other roles as needed.

The other options are incorrect because they are not the object type level at which these privileges can be granted. Database, schema, and table are lower-level object types that do not support these privileges. References: [Access Control Privileges | Snowflake Documentation](#), [Using Dynamic Data Masking | Snowflake Documentation](#), [Using Row Access Policies | Snowflake Documentation](#), [Using Session Policies | Snowflake Documentation](#)

Q59. A user needs access to create materialized view on a schema mydb.myschema.

What is the appropriate command to provide the access?

- * GRANT ROLE MYROLE TO USER USER1; GRANT CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO MYROLE;
- * GRANT CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO USER USER1;
- * GRANT CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO USER1;

Q60. COMPRESSION = AUTO can automatically detect below compression techniques when FORMAT TYPE is CSV

- * GZIP
- * BZ2
- * BROTLI
- * ZSTD
- * DEFLATE
- * RAW_DEFLATE

Q61. Which of the following are characteristics of how row access policies can be applied to external tables?

(Choose three.)

- * An external table can be created with a row access policy, and the policy can be applied to the VALUE column.
- * A row access policy can be applied to the VALUE column of an existing external table.
- * A row access policy cannot be directly added to a virtual column of an external table.
- * External tables are supported as mapping tables in a row access policy.
- * While cloning a database, both the row access policy and the external table will be cloned.
- * A row access policy cannot be applied to a view created on top of an external table.

Explanation

These three statements are true according to the Snowflake documentation and the web search results. A row access policy is a feature that allows filtering rows based on user-defined conditions. A row access policy can be applied to an external table, which is a table that reads data from external files in a stage. However, there are some limitations and considerations for using row access policies with external tables.

- * An external table can be created with a row access policy by using the WITH ROW ACCESS POLICY clause in the CREATE EXTERNAL TABLE statement. The policy can be applied to the VALUE column, which is the column that contains the raw data from the external files in a VARIANT data type¹.
- * A row access policy can also be applied to the VALUE column of an existing external table by using the ALTER TABLE statement with the SET ROW ACCESS POLICY clause².
- * A row access policy cannot be directly added to a virtual column of an external table. A virtual column is a column that is derived from the VALUE column using an expression. To apply a row access policy to a virtual column, the policy must be applied to the VALUE column and the expression must be repeated in the policy definition³.
- * External tables are not supported as mapping tables in a row access policy. A mapping table is a table that is used to determine the access rights of users or roles based on some criteria. Snowflake does not support using an external table as a mapping table because it may cause performance issues or errors⁴.
- * While cloning a database, Snowflake clones the row access policy, but not the external table. Therefore, the policy in the cloned database refers to a table that is not present in the cloned database. To avoid this issue, the external table must be manually cloned or recreated in the cloned database⁴.
- * A row access policy can be applied to a view created on top of an external table. The policy can be applied to the view itself or to the underlying external table. However, if the policy is applied to the view, the view must be a secure view, which is a view that hides the underlying data and the view definition from unauthorized users⁵.

References:

- * [CREATE EXTERNAL TABLE | Snowflake Documentation](#)
- * [ALTER EXTERNAL TABLE | Snowflake Documentation](#)
- * [Understanding Row Access Policies | Snowflake Documentation](#)
- * [Snowflake Data Governance: Row Access Policy Overview](#)
- * [Secure Views | Snowflake Documentation](#)

Q62. How is the change of local time due to daylight savings time handled in Snowflake tasks? (Choose two.)

- * A task scheduled in a UTC-based schedule will have no issues with the time changes.

- * Task schedules can be designed to follow specified or local time zones to accommodate the time changes.
- * A task will move to a suspended state during the daylight savings time change.
- * A frequent task execution schedule like minutes may not cause a problem, but will affect the task history.
- * A task schedule will follow only the specified time and will fail to handle lost or duplicated hours.

According to the Snowflake documentation¹ and the web search results², these two statements are true about how the change of local time due to daylight savings time is handled in Snowflake tasks. A task is a feature that allows scheduling and executing SQL statements or stored procedures in Snowflake. A task can be scheduled using a cron expression that specifies the frequency and time zone of the task execution.

A task scheduled in a UTC-based schedule will have no issues with the time changes. UTC is a universal time standard that does not observe daylight savings time. Therefore, a task that uses UTC as the time zone will run at the same time throughout the year, regardless of the local time changes¹.

Task schedules can be designed to follow specified or local time zones to accommodate the time changes. Snowflake supports using any valid IANA time zone identifier in the cron expression for a task. This allows the task to run according to the local time of the specified time zone, which may include daylight savings time adjustments. For example, a task that uses Europe/London as the time zone will run one hour earlier or later when the local time switches between GMT and BST¹².

Reference:

Snowflake Documentation: Scheduling Tasks

Snowflake Community: Do the timezones used in scheduling tasks in Snowflake adhere to daylight savings?

Q63. Which copy options are not supported by CREATE PIPE AS COPY FROM command?

- * FILES = (file_name1 [, file_name2 , ;])
- * FORCE = TRUE | FALSE
- * ON_ERROR = ABORT_STATEMENT
- * VALIDATION_MODE = RETURN_n_ROWS | RETURN_ERRORS | RETURN_ALL_ERRORS
- * MATCH_BY_COLUMN_NAME = CASE_SENSITIVE | CASE_INSENSITIVE | NONE

Q64. What conditions should be true for a table to consider search optimization

- * The table size is at least 100 GB
- * The table is not clustered OR The table is frequently queried on columns other than the primary cluster key
- * The table can be of any size

Q65. Why might a Snowflake Architect use a star schema model rather than a 3NF model when designing a data architecture to run in Snowflake? (Select TWO).

- * Snowflake cannot handle the joins implied in a 3NF data model.
- * The Architect wants to remove data duplication from the data stored in Snowflake.
- * The Architect is designing a landing zone to receive raw data into Snowflake.
- * The BI tool needs a data model that allows users to summarize facts across different dimensions, or to drill down from the summaries.
- * The Architect wants to present a simple flattened single view of the data to a particular group of end users.

Q66. An Architect is integrating an application that needs to read and write data to Snowflake without installing any additional software on the application server.

How can this requirement be met?

- * Use SnowSQL.

- * Use the Snowpipe REST API.
- * Use the Snowflake SQL REST API.
- * Use the Snowflake ODBC driver.

Q67. Refreshing a secondary database is not allowed in the following circumstances

- * Materialized views
- * Primary database contains transient tables
- * Databases created from shares
- * Primary database has external table

Q68. You have a table named customer_table. You want to create another table as customer_table_other which will be same as customer_table with respect to schema and data.

What is the best option?

- * CREATE TABLE customer_table_other CLONE customer_table
- * CREATE TABLE customer_table_other AS SELECT * FROM customer_table
- * ALTER TABLE customer_table_other SWAP WITH customer_table

Q69. You have create a table in snowflake as below

CREATE TABLE EMPLOYEE(EMPLOYEE_NAME STRING, SALARY NUMBER); When you do a DESCRIBE TABLE EMPLOYEE, what will you see as the data type of EMPLOYEE_NAME?

- * VARCHAR(10)
- * VARCHAR
- * VARCHAR(16777216)
- * STRING

Q70. Let's say that you have two JSONs as below

1. {“stuId”:2000, “stuName”:”Amy”}
2. {“stuId”:2000,”stuCourse”:”Snowflake”}

How will you write a query that will check if stuId in JSON in #1 is also there in JSON in#2

- * with stu_demography as (select parse_json(column1) as src, src:stuId as ID from values(‘{“stuId”:2000, “stuName”:”Amy”}’)),
- * stu_course as (select parse_json(column1) as src, src:stuId as ID from values(‘{“stuId”:2000,”stuCourse”:”Snowflake”}’)) select case when stdemo.ID in(select ID from stu_course) then ‘True’ else ‘False’ end as result from stu_demography stdemo;
- * with stu_demography as (select parse_json(column1) as src, src[‘stuId’] as ID from values(‘{“stuId”:2000, “stuName”:”Amy”}’)), stu_course as (select parse_json(column1) as src, src[‘stuId’] as ID from values(‘{“stuId”:2000,”stuCourse”:”Snowflake”}’)) select case when stdemo.ID in(select ID from stu_course) then ‘True’ else ‘False’ end as result from stu_demography stdemo;
- * SELECT CONTAINS(‘{“stuId”:2000, “stuName”:”Amy”}’,'{“stuId”:2000,”stuCourse”:”Snowflake”}’);
- * with stu_demography as (select parse_json(column1) as src, src[‘STUID’] as ID from

```
values(&#8216;{&#8220;stuId&#8221;:2000, &#8220;stuName&#8221;:&#8221;Amy&#8221;}&#8217;)), stu_course as (select
parse_json(column1) as src, src[&#8216;stuId&#8217;] as ID from
values(&#8216;{&#8220;stuId&#8221;:2000,&#8221;stuCourse&#8221;:&#8221;Snowflake&#8221;}&#8217;)) select case when
stdemo.ID in(select ID from stu_course) then &#8216;True&#8217; else &#8216;False&#8217; end as result from stu_demography
stdemo;
```

Q71. Multi-cluster warehouses are best utilized for

- * Scaling resources to improve concurrency for users/queries
- * Improving the performance of slow-running queries
- * Improving the performance of data loading

Q72. An Architect needs to grant a group of ORDER_ADMIN users the ability to clean old data in an ORDERS table (deleting all records older than 5 years), without granting any privileges on the table. The group's manager (ORDER_MANAGER) has full DELETE privileges on the table.

How can the ORDER_ADMIN role be enabled to perform this data cleanup, without needing the DELETE privilege held by the ORDER_MANAGER role?

- * Create a stored procedure that runs with caller's rights, including the appropriate > 5 years business logic, and grant USAGE on this procedure to ORDER_ADMIN. The ORDER_MANAGER role owns the procedure.
- * Create a stored procedure that can be run using both caller's and owner's rights (allowing the user to specify which rights are used during execution), and grant USAGE on this procedure to ORDER_ADMIN. The ORDER_MANAGER role owns the procedure.
- * Create a stored procedure that runs with owner's rights, including the appropriate > 5 years business logic, and grant USAGE on this procedure to ORDER_ADMIN. The ORDER_MANAGER role owns the procedure.
- * This scenario would actually not be possible in Snowflake; any user performing a DELETE on a table requires the DELETE privilege to be granted to the role they are using.

This is the correct answer because it allows the ORDER_ADMIN role to perform the data cleanup without needing the DELETE privilege on the ORDERS table. A stored procedure is a feature that allows scheduling and executing SQL statements or stored procedures in Snowflake. A stored procedure can run with either the caller's rights or the owner's rights. A caller's rights stored procedure runs with the privileges of the role that called the stored procedure, while an owner's rights stored procedure runs with the privileges of the role that created the stored procedure. By creating a stored procedure that runs with owner's rights, the ORDER_MANAGER role can delegate the specific task of deleting old data to the ORDER_ADMIN role, without granting the ORDER_ADMIN role more general privileges on the ORDERS table. The stored procedure must include the appropriate business logic to delete only the records older than 5 years, and the ORDER_MANAGER role must grant the USAGE privilege on the stored procedure to the ORDER_ADMIN role. The ORDER_ADMIN role can then execute the stored procedure to perform the data cleanup.

Reference:

Snowflake Documentation: Stored Procedures

Snowflake Documentation: Understanding Caller's Rights and Owner's Rights Stored Procedures

Q73. How is the change of local time due to daylight savings time handled in Snowflake tasks? (Choose two.)

- * A task scheduled in a UTC-based schedule will have no issues with the time changes.
- * Task schedules can be designed to follow specified or local time zones to accommodate the time changes.
- * A task will move to a suspended state during the daylight savings time change.
- * A frequent task execution schedule like minutes may not cause a problem, but will affect the task history.
- * A task schedule will follow only the specified time and will fail to handle lost or duplicated hours.

Explanation

According to the Snowflake documentation¹ and the web search results², these two statements are true about how the change of local time due to daylight savings time is handled in Snowflake tasks. A task is a feature that allows scheduling and executing SQL statements or stored procedures in Snowflake. A task can be scheduled using a cron expression that specifies the frequency and time zone of the task execution.

* A task scheduled in a UTC-based schedule will have no issues with the time changes. UTC is a universal time standard that does not observe daylight savings time. Therefore, a task that uses UTC as the time zone will run at the same time throughout the year, regardless of the local time changes¹.

* Task schedules can be designed to follow specified or local time zones to accommodate the time changes. Snowflake supports using any valid IANA time zone identifier in the cron expression for a task. This allows the task to run according to the local time of the specified time zone, which may include daylight savings time adjustments. For example, a task that uses Europe/London as the time zone will run one hour earlier or later when the local time switches between GMT and BST¹².

References:

* Snowflake Documentation: Scheduling Tasks

* Snowflake Community: Do the timezones used in scheduling tasks in Snowflake adhere to daylight savings?

ARA-C01 Questions PDF [2024 Use Valid New dump to Clear Exam:

<https://www.actualtests4sure.com/ARA-C01-test-questions.html>]