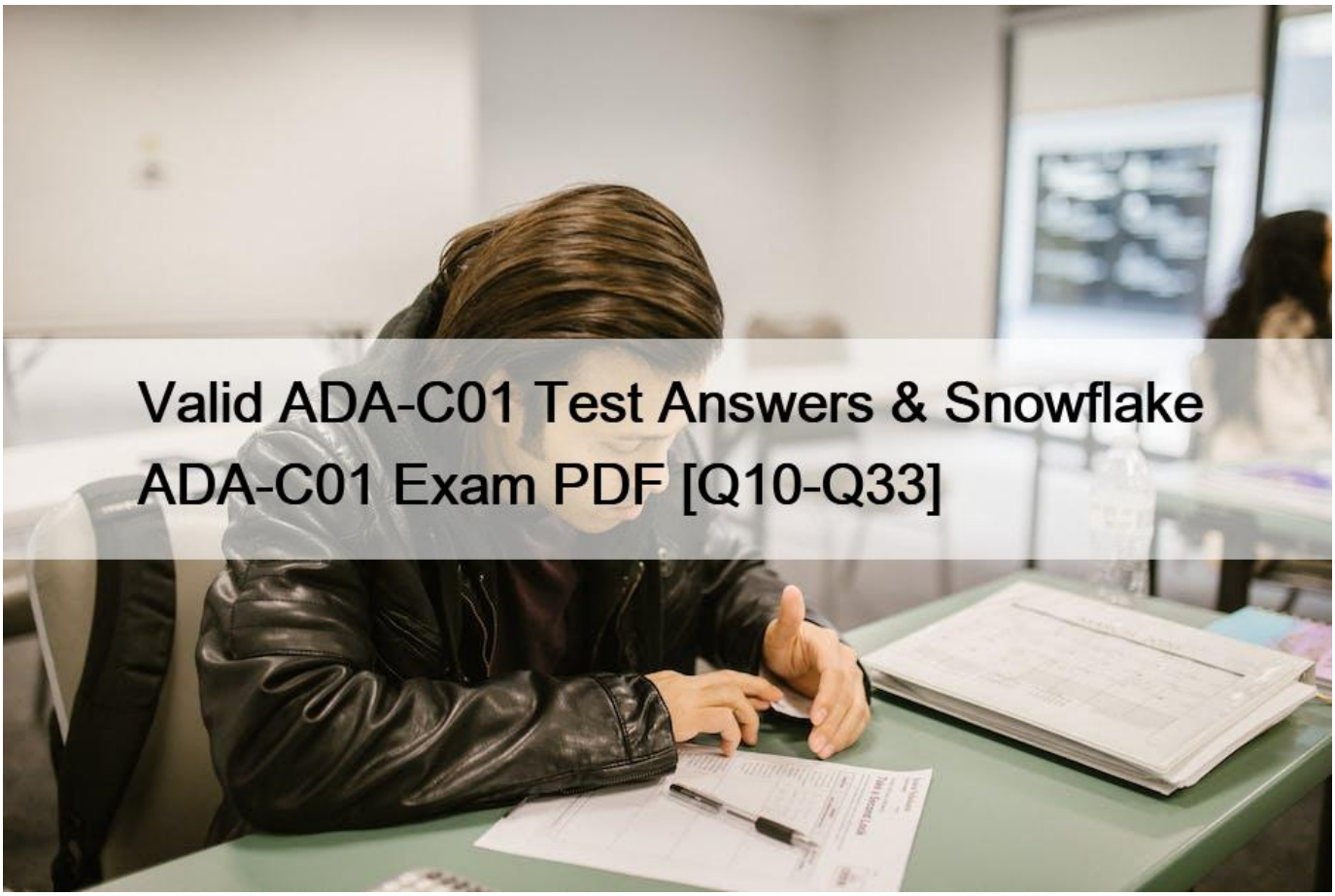
Valid ADA-C01 Test Answers & Snowflake ADA-C01 Exam PDF [Q10-Q33



Valid ADA-C01 Test Answers & Snowflake ADA-C01 Exam PDF
Snowflake ADA-C01 Certification Real 2024 Mock Exam

NEW QUESTION 10

Which Snowflake objects can be managed using SCIM integration? (Select TWO).
* Stages
* Users
* Warehouses
* Roles
* Shares

Explanation

A SCIM security integration allows the automated management of user identities and groups (i.e. roles) by creating an interface between Snowflake and a third-party Identity Provider (IdP)1. Snowflake supports SCIM integration with Okta, Azure, and custom SCIM clients2. SCIM integration does not support managing other Snowflake objects, such as stages, warehouses, or shares3. Therefore, the answer is B. Users and D. Roles.

NEW QUESTION 11

A Snowflake Administrator needs to persist all virtual warehouse configurations for auditing and backups. Given a table already exists with the following schema:

Table Name : VWH_META

Column 1 : SNAPSHOT_TIME TIMESTAMP_NTZ

Column 2 : CONFIG VARIANT

Which commands should be executed to persist the warehouse data at the time of execution in JSON format in the table VWH META?
* 1. SHOW WAREHOUSES;

2. INSERT INTO VWH META

SELECT CURRENT TIMESTAMP (),

FROM TABLE (RESULT_SCAN (LAST_QUERY_ID(1))) ;
* 1. SHOW WAREHOUSES;

2. INSERT INTO VWH META

SELECT CURRENT TIMESTAMP (), *

FROM TABLE (RESULT_SCAN (LAST_QUERY_ID ())) ;
* 1. SHOW WAREHOUSES;

2. INSERT INTO VWH_META

SELECT CURRENT_TIMESTAMP (),

OBJECT CONSTRUCT (*)

FROM TABLE (RESULT_SCAN (LAST_QUERY_ID ()));
* 1. SHOW WAREHOUSES;

2. INSERT INTO VWH META

SELECT CURRENT TIMESTAMP (), *

FROM TABLE (RESULT_SCAN (SELECT

LAST QUERY ID(-1)));
According to the Using Persisted Query Results documentation, the RESULT_SCAN function allows you to query the result set of a previous command as if it were a table. The LAST_QUERY_ID function returns the query ID of the most recent statement executed in the current session. Therefore, the combination of these two functions can be used to access the output of the SHOW WAREHOUSES command, which returns the configurations of all the virtual warehouses in the account. However, to persist the warehouse data in JSON format in the table VWH_META, the OBJECT_CONSTRUCT function is needed to convert the output of

the SHOW WAREHOUSES command into a VARIANT column. The OBJECT_CONSTRUCT function takes a list of key-value pairs and returns a single JSON object. Therefore, the correct commands to execute are:

1. SHOW WAREHOUSES;

2. INSERT INTO VWH_META SELECT CURRENT_TIMESTAMP (), OBJECT_CONSTRUCT (*) FROM TABLE (RESULT_SCAN (LAST_QUERY_ID ())); The other options are incorrect because:

* A) This option does not use the OBJECT_CONSTRUCT function, so it will not persist the warehouse data in JSON format. Also, it is missing the * symbol in the SELECT clause, so it will not select any columns from the result set of the SHOW WAREHOUSES command.

* B) This option does not use the OBJECT_CONSTRUCT function, so it will not persist the warehouse data in JSON format. It will also try to insert multiple columns into a single VARIANT column, which will cause a type mismatch error.

* D) This option does not use the OBJECT_CONSTRUCT function, so it will not persist the warehouse data in JSON format. It will also try to use the RESULT_SCAN function on a subquery, which is not supported. The RESULT_SCAN function can only be used on a query ID or a table name.

NEW QUESTION 12

.

Company A uses Snowflake to manage audio files of call recordings. Company A hired Company B, who also uses Snowflake, to transcribe the audio files for further analysis.

Company A’s Administrator created a share.

What object should be added to the share to allow Company B access to the files?
* A secure view with a column for file URLs.
* A secure view with a column for pre-signed URLs.
* A secure view with a column for METADATA$FILENAME.
* A secure view with a column for the stage name and a column for the file path.
Explanation

According to the Snowflake documentation1, pre-signed URLs are required to access external files in a share.

A secure view can be used to generate pre-signed URLs for the audio files stored in an external stage and expose them to the consumer account. Option A is incorrect because file URLs alone are not sufficient to access external files in a share. Option C is incorrect because METADATA$FILENAME only returns the file name, not the full path or URL. Option D is incorrect because the stage name and file path are not enough to generate pre-signed URLs.

NEW QUESTION 13

An Administrator receives data from a Snowflake partner. The partner is sharing a dataset that contains multiple secure views. The Administrator would like to configure the data so that only certain roles can see certain secure views.

How can this be accomplished?
* Apply RBAC directly onto the partner’s shared secure views.
* Individually grant imported privileges onto the schema in the share.

* Clone the data and insert it into a company-owned share and apply the desired RBAC on the new tables.
* Create views over the incoming shared database and apply the desired RBAC onto these views.

Explanation

According to the Snowflake documentation1, secure views are only exposed to authorized users who have been granted the role that owns the view. Therefore, applying RBAC directly onto the partner’s shared secure views (option A) is not possible, as the administrator does not own those views. Individually granting imported privileges onto the schema in the share (option B) is also not feasible, as the privileges granted on the schema do not apply to existing secure views, only tofuture ones2. Cloning the data and inserting it into a company-owned share (option C) is not recommended, as it would create unnecessary duplication of data and increase storage costs. The best option is to create views over the incoming shared database and apply the desired RBAC onto these views (option D). This way, the administrator can control the access to the data based on the roles in their account, without modifying the original data or views from the partner.

NEW QUESTION 14

A company’s Snowflake account has multiple roles. Each role should have access only to data that resides in the given role’s specific region.

When creating a row access policy, which code snippet below will provide privileges to the role ALL_ACCESS_ROLE to see all rows regardless of region, while the other roles can only see rows for their own regions?

* create or replace row access policy region policy as (region_value varchar) returns boolean ->

‘ALL ACCESS_ROLE’ = current_role ()

and exists (

select 1 from entitlement_table

where role = current_role ()

and region = region_value

)

* create or replace row access policy region policy as (region_value varchar) returns boolean -> exists (select 1 from entitlement_table where role = current_role () and region = region_value)
* create or replace row access policy region policy as (region_value varchar) returns boolean ->

‘ALL_ACCESS_ROLE’ = current_role ()

or exists (

select 1 from entitlement_table

where role = current_role ()

and region = region_value

)

* create or replace row access policy region policy as (region_value varchar) returns boolean ->

‘ALL ACCESS ROLE’ = current_role ()

)
Explanation

This code snippet will create a row access policy that returns true if the current role is ALL_ACCESS_ROLE or if the current role matches the region value in the entitlement_table. This means that the ALL_ACCESS_ROLE can see all rows regardless of region, while the other roles can only see rows for their own regions. According to the Snowflake documentation, the CURRENT_ROLE context function returns the name of the current role for the session. The EXISTS function returns true if the subquery returns any rows.

The OR operator returns true if either operand is true. Therefore, this code snippet satisfies the requirements of the question.

NEW QUESTION 15

Which actions are considered breaking changes to data that is shared with consumers in the Snowflake Marketplace? (Select TWO).
* Dropping a column from a table
* Deleting data from a table
* Unpublishing the data listing
* Renaming a table
* Adding region availability to the listing

According to the Snowflake documentation1, breaking changes are changes that affect the schema or structure of the shared data, such as dropping or renaming a column or a table. These changes may cause errors or unexpected results for the consumers who query the shared data. Deleting data from a table, unpublishing the data listing, or adding region availability to the listing are not breaking changes, as they do not alter the schema or structure of the shared data.

1: Managing Data Listings in Snowflake Data Marketplace | Snowflake Documentation

NEW QUESTION 16

A requirement has been identified to allow members of a corporate Data Product team to bring in data sets from the Snowflake Marketplace. The members of this team use the role DP_TEAM.

What grant statements must the ACCOUNTADMIN execute in order for the DP TEAM role to import and work with data from the Marketplace?
* grant marketplace in account to role dp_team;

grant create database from share to role dp_team;
* grant usage on snowflake_marketplace to role dp_team;

grant create database on account to role dp_team;
* grant imported privileges on account to role dp_team;

grant create database on account to role dp_team;
* grant import share on account to role dp_team;

grant create database on account to role dp_team;

Option D is the correct answer because it follows the steps described in the Snowflake documentation for importing data from the Snowflake Marketplace. The ACCOUNTADMIN role needs to grant the IMPORT SHARE privilege on the account to the DP_TEAM role, which allows the role to import data from any provider in the marketplace. The ACCOUNTADMIN role also needs

to grant the CREATE DATABASE privilege on the account to the DP_TEAM role, which allows the role to create a database from a share. Option A is incorrect because there is no MARKETPLACE privilege in Snowflake. Option B is incorrect because the USAGE privilege on SNOWFLAKE_MARKETPLACE is not sufficient to import data from the marketplace. Option C is incorrect because there is no IMPORTED PRIVILEGES privilege in Snowflake.

NEW QUESTION 17

A Snowflake Administrator wants to create a virtual warehouse that supports several dashboards, issuing various queries on the same database.

For this warehouse, why should the Administrator consider setting AUTO_SUSPEND to 0 or NULL?
* To save costs on warehouse shutdowns and startups for different queries
* To save costs by running the warehouse as little as possible
* To keep the data cache warm to support good performance of similar queries
* To keep the query result cache warm for good performance on repeated queries

According to the Snowflake documentation1, the AUTO_SUSPEND property specifies the number of seconds of inactivity after which a warehouse is automatically suspended. If the property is set to 0 or NULL, the warehouse never suspends automatically. For a warehouse that supports several dashboards, issuing various queries on the same database, setting AUTO_SUSPEND to 0 or NULL can help to keep the data cache warm, which means that the data used by the queries is already loaded into the warehouse memory and does not need to be fetched from the storage layer. This can improve the performance of similar queries that access the same data. Option A is incorrect because setting AUTO_SUSPEND to 0 or NULL does not save costs on warehouse shutdowns and startups, but rather increases the costs by keeping the warehouse running continuously. Option B is incorrect because setting AUTO_SUSPEND to 0 or NULL does not run the warehouse as little as possible, but rather runs the warehouse as much as possible. Option D is incorrect because setting AUTO_SUSPEND to 0 or NULL does not affect the query result cache, which is a separate cache that stores the results of previous queries for a period of time. The query result cache is not dependent on the warehouse state, but on the query criteria2.

NEW QUESTION 18

A company has set up a new Snowflake account. An Identity Provider (IdP) has been configured for both Single Sign-On (SSO) and SCIM provisioning.

What maintenance is required to ensure that the SCIM provisioning process continues to operate without errors?
* The IdP service account requires a new RSA key pair to be generated every six months.
* The Administrator must issue a POST RENEW call to the REST API at least once every six months.
* The OAuth Bearer Tokens have a lifespan of six months and must be regenerated prior to expiration.
* The IdP Administrator must issue a REFRESH transaction at least once every six months to synchronize all users and roles.

Explanation

According to the Snowflake documentation1, the authentication process for SCIM provisioning uses an OAuth Bearer token and this token is valid for six months. Customers must keep track of their authentication token and can generate a new token on demand. If the token expires, the SCIM provisioning process will fail.

Therefore, the token must be regenerated before it expires. The other options are not required for SCIM provisioning.

NEW QUESTION 19

A company has many users in the role ANALYST who routinely query Snowflake through a reporting tool. The Administrator has noticed that the ANALYST users keep two small clusters busy all of the time, and occasionally they need three or four clusters of that size.

Based on this scenario, how should the Administrator set up a virtual warehouse to MOST efficiently support this group of users?
* Create a multi-cluster warehouse with MIN_CLUSTERS set to 1. Give MANAGE privileges to the ANALYST role so this group can start and stop the warehouse, and increase the number of clusters as needed.
* Create a multi-cluster warehouse with MIN_CLUSTERS set to 2. Set the warehouse to auto-resume and auto-suspend, and give USAGE privileges to the ANALYST role. Allow the warehouse to auto-scale.
* Create a standard X-Large warehouse, which is equivalent to four small clusters. Set the warehouse to auto-resume and auto-suspend, and give USAGE privileges to the ANALYST role.
* Create four virtual warehouses (sized Small through XL) and set them to auto-suspend and auto-resume. Have users in the ANALYST role select the appropriate warehouse based on how many queries are being run.
According to the Snowflake documentation1, a multi-cluster warehouse is a virtual warehouse that consists of multiple clusters of compute resources that can scale up or down automatically to handle the concurrency and performance needs of the queries submitted to the warehouse. A multi-cluster warehouse has a minimum and maximum number of clusters that can be specified by the administrator. Option B is the most efficient way to support the group of users, as it allows the administrator to create a multi-cluster warehouse with MIN_CLUSTERS set to 2, which means that the warehouse will always have two clusters running to handle the standard workload. The warehouse can also auto-scale up to the maximum number of clusters (which can be set according to the peak workload) when there is a spike in demand, and then scale down when the demand decreases. The warehouse can also auto-resume and auto-suspend, which means that the warehouse will automatically start when a query is submitted and automatically stop after a period of inactivity. The administrator can also give USAGE privileges to the ANALYST role, which means that the users can use the warehouse to execute queries and load data, but not modify or operate the warehouse. Option A is not efficient, as it requires the users to manually start and stop the warehouse, and increase the number of clusters as needed, which can be time-consuming and error-prone. Option C is not efficient, as it creates a standard X-Large warehouse, which is equivalent to four small clusters, which may be more than needed for the standard workload, and may not be enough for the peak workload. Option D is not efficient, as it creates four virtual warehouses of different sizes, which can be confusing and cumbersome for the users to select the appropriate warehouse based on how many queries are being run, and may also result in wasted resources and costs.

NEW QUESTION 20

A user with the proper role issues the following commands when setting up and activating network policies:

CREATE OR REPLACE NETWORK POLICY foo_policy

ALLOWED_IP_LIST = (‘1.1.1.0/24’, ‘2.2.2.0/24’ , ‘3.3. 3. 0/24’)

BLOCKED IP LIST = (‘1.1.1.1’)

COMMENT = ‘Account level policy’;

ALTER ACCOUNT SET NETWORK_POLICY=FOO_POLICY;

CREATE OR REPLACE NETWORK POLICY bar_policy

ALLOWED_IP_LIST = (‘3.3.3.0/24’)

BLOCKED IP LIST = (‘3.3.3.10’)

COMMENT = ‘user level policy’;

ALTER USER userl SET NETWORK_POLICY=BAR_POLICY;

Afterwards, user1 attempts to log in to Snowflake from IP address 3.3.3.10.

Will the login be successful?
* Yes, because 3.3.3.10 is found in the ALLOWED_IP_LIST of bar_policy.
* No, because 3.3.3.10 is found in the BLOCKED_IP_LIST of bar_policy.
* Yes, because 3.3.3.10 is found in the ALLOWED_IP_LIST of foo_policy.
* No, because 3.3.3.10 is not found in the ALLOWED_IP_LIST of foo_policy.

Explanation

According to the Snowflake documentation1, network policies are a feature that allows restricting access to your account based on user IP address. A network policy can be applied to an account, a user, or a security integration, and can specify a list of allowed IP addresses and a list of blocked IP addresses. If there are network policies applied to more than one of these, the most specific network policy overrides more general network policies. In this case, the user1 has a network policy (bar_policy) applied to them, which overrides the account-level network policy (foo_policy). The bar_policy allows access only from the IP range 3.3.3.0/24, and blocks access from the IP address 3.3.3.10. Therefore, the user1 will not be able to log in to Snowflake from IP address 3.3.3.10, as it is found in the BLOCKED_IP_LIST of bar_policy. Option A is incorrect because the ALLOWED_IP_LIST of bar_policy does not override the BLOCKED_IP_LIST of bar_policy.

Option C is incorrect because the ALLOWED_IP_LIST of foo_policy does not apply to user1, as it is overridden by the user-level network policy. Option D is incorrect because the ALLOWED_IP_LIST of foo_policy does not matter, as it is overridden by the user-level network policy.

NEW QUESTION 21

Which command can temporarily disable Multi-factor Authentication (MFA) for the Snowflake username user1 for 24 hours?
* alter user userl set MINS_TO_BYPASS_MFA=1440;
* alter user userl set DISABLE_MFA=1440;
* alter user userl set TEMPORARY_MFA_BYPASS=1440;
* alter user userl set HOURS_TO_BYPASS_MFA=24;

Explanation

According to the Snowflake documentation1, the MINS_TO_BYPASS_MFA property specifies the number of minutes to temporarily disable MFA for a user so that they can log in without the temporary token generated by the Duo Mobile application. After the time passes, MFA is enforced and the user cannot log in without the token. Therefore, to disable MFA for 24 hours, the value of this property should be set to 1440 minutes (24 x

60). Option B is incorrect because the DISABLE_MFA property is a boolean value that permanently disables MFA for a user, not a numeric value that specifies the duration. Option C is incorrect because there is no such property as TEMPORARY_MFA_BYPASS in Snowflake. Option D is incorrect because there is no such property as HOURS_TO_BYPASS_MFA in Snowflake.

NEW QUESTION 22

Which masking policy will mask a column whenever it is queried through a view owned by a role named MASKED_VIEW_ROLE?
* create or replace masking policy maskstring as (val string) returns string -> case when is_role_in_session (‘MASKED_VIEW_ROLE’) then ‘ ** else val end;

*,
* create or replace masking policy maskString as (val string) returns string -> case when array_contains (‘MASKED_VIEW_ROLE’ :: variant, parse_json (current_available_roles ())) then ‘* else val end;

** ‘
* create or replace masking policy maskstring as (val string) returns string -> case when invoker_role() in (‘MASKED_VIEW_ROLE’) then else val end;

‘ **
* create or replace masking policy maskString as (val string) returns string -> case when current_role() in (‘MASKED_VIEW_ROLE’) then ‘ ********* ‘ else val end;
A masking policy is a SQL expression that transforms the data in a column based on the role that queries the column1. The is_role_in_session function returns true if the specified role is in the current session2. Therefore, the masking policy in option A will mask the column data with asterisks whenever it is queried through a view owned by the MASKED_VIEW_ROLE3. The other options use different functions that do not check the ownership of the view, but rather the current role, the invoker role, or the available roles in the session45. These functions may not return the desired result if the role that owns the view is different from the role that queries the view.

NEW QUESTION 23

What are benefits of creating and maintaining resource monitors in Snowflake? (Select THREE).
* The cost of running a resource monitor is only 10% of a credit, per day of operation.
* Multiple resource monitors can be applied to a single virtual warehouse.
* Resource monitors add no additional load to virtual warehouse compute.
* Multiple triggers can be configured across various virtual warehouse thresholds.
* Resource monitor governance is tightly controlled and monitors can only be created by the ACCOUNTADMIN role or users with the CREATE MONITOR privilege.
* Resource monitors can be applied to more than one virtual warehouse.
According to the Snowflake documentation1, resource monitors are a feature that helps you manage and control Snowflake costs by monitoring and setting limits on your compute resources. Resource monitors do not consume any credits or add any load to the virtual warehouses they monitor1. Resource monitors can also have multiple triggers that specify different actions (such as suspending or notifying) when certain percentages of the credit quota are reached2. Resource monitors can be applied to either the entire account or a specific set of individual warehouses1. The other options are not benefits of resource monitors. The cost of running a resource monitor is negligible, not 10% of a credit3. Multiple resource monitors cannot be applied to a single virtual warehouse; only one resource monitor can be assigned to a warehouse at a time2. Resource monitor governance is not tightly controlled; account administrators can enable users with other roles to view and modify resource monitors using SQL2.

NEW QUESTION 24

What session parameter can be used to test the integrity of secure views based on the account that is accessing that view?
* MIMIC_CONSUMER_ACCOUNT
* TEST_ACCOUNT_ID
* PRODUCER_TEST_ACCT
* SIMULATED_DATA_SHARING_CONSUMER
Explanation

The SIMULATED_DATA_SHARING_CONSUMER session parameter allows a data provider to test the integrity of secure views based on the account that is accessing that view2. By setting this parameter to the name of the consumer account, the data provider can query the secure view and see the results that a user in the consumer account will see2. This helps to ensure that sensitive data in a shared database is not exposed to unauthorized users1. The other options are not valid session parameters in Snowflake3

NEW QUESTION 25

Which type of listing in the Snowflake Marketplace can be added and queried immediately?
* Monetized listing
* Standard listing
* Regional listing
* Personalized listing

According to the Snowflake documentation1, a standard listing is a type of listing that provides free access to the full data product, with no payment required. A standard listing can be added and queried immediately by the consumer, as long as they accept the terms and conditions of the listing. A monetized listing is a type of listing that charges for access to the data product, using the pricing models offered by Snowflake. A monetized listing requires the consumer to provide payment information and agree to the billing terms before accessing the data product. A regional listing is not a type of listing, but a way to specify the regions where the listing is available. A personalized listing is a type of listing that provides limited trial access to the data product, with unlimited access to the full data product available upon request. A personalized listing requires the consumer to request access from the provider and wait for the provider to grant access before accessing the data product. Therefore, the only type of listing that can be added and queried immediately is the standard listing.

NEW QUESTION 26

Which Snowflake objects can be managed using SCIM integration? (Select TWO).
* Stages
* Users
* Warehouses
* Roles
* Shares

A SCIM security integration allows the automated management of user identities and groups (i.e. roles) by creating an interface between Snowflake and a third-party Identity Provider (IdP)1. Snowflake supports SCIM integration with Okta, Azure, and custom SCIM clients2. SCIM integration does not support managing other Snowflake objects, such as stages, warehouses, or shares3. Therefore, the answer is B. Users and D. Roles.

NEW QUESTION 27

What are benefits of using Snowflake organizations? (Select TWO).
* Administrators can change Snowflake account editions on-demand based on need.
* Administrators can monitor and understand usage across all accounts in the organization.
* Administrators can simplify data movement across all accounts within the organization.
* User administration is simplified across all accounts within the organization.
* Administrators have the ability to create accounts in any available cloud provider or region.

Explanation

According to the Snowflake documentation1, organizations are a feature that allows linking the accounts owned by a business entity, simplifying account management and billing, replication and failover, data sharing, and other account administration tasks. Some of the benefits of using organizations are:

*Administrators can monitor and understand usage across all accounts in the organization using the ORGANIZATION_USAGE schema, which provides historical usage data for all accounts in the organization via views in a shared database named SNOWFLAKE2. This can help to optimize costs and performance across the organization.

*Administrators have the ability to create accounts in any available cloud provider or region using the CREATE ACCOUNT command, which allows specifying the cloud platform and region for the new account3.

This can help to meet the business needs and compliance requirements of the organization.

Option A is incorrect because administrators cannot change Snowflake account editions on-demand based on need, but rather have to contact Snowflake Support to request an edition change4. Option C is incorrect because administrators cannot simplify data movement across all accounts within the organization, but rather have to enable account database replication for both the source and target accounts, and use the ALTER DATABASE … ENABLE REPLICATION TO ACCOUNTS command to promote a local database to serve as the primary database and enable replication to the target accounts5. Option D is incorrect because user administration is not simplified across all accounts within the organization, but rather requires creating and managing users, roles, and privileges for each account separately, unless using a federated authentication method such as SSO or SCIM.

NEW QUESTION 28

An Administrator has a warehouse which is intended to have a credit quota set for 3000 for each calendar year. The Administrator needs to create a resource monitor that will perform the following tasks:

1. At 80% usage notify the account Administrators.

2. At 100% usage suspend the warehouse and notify the account Administrators.

3. At 120% stop all running executions, suspend the warehouse, and notify the account Administrators.

Which SQL command will meet these requirements?
* create or replace resource monitor RM1 with credit_quota=3000

start_timestamp = ‘2022-01-01 00:00 CET’

triggers on 80 percent do notify

on 100 percent do suspend

on 120 percent do suspend_immediate;

alter warehouse WH1 set resource_monitor = RM1;
* create or replace resource monitor RM1 with credit_quota=3000

frequency = yearly

start_timestamp = ‘2022-01-01 00:00 CET’

triggers on 80 percent do notify

on 100 percent do suspend

on 120 percent do suspend_immediate;

alter warehouse WH1 set resource monitor = RM1;
* create or replace resource monitor RM1 with credit_quota=3000

start_timestamp = ‘2022-01-01 00:00 CET’

triggers on 80 percent do notify

on 100 percent do notify and suspend

on 120 percent do notify and suspend_immediate;

alter warehouse WH1 set resource monitor = RM1;
* create or replace resource monitor RM1 with credit_quota=3000

frequency = yearly

triggers on 80 percent do notify

on 100 percent do suspend

on 120 percent do suspend_immediate;

alter warehouse WH1 set resource_monitor = RM1;
Option B is the correct SQL command to create a resource monitor that meets the requirements. It sets the credit quota to 3000, the frequency to yearly, the start timestamp to January 1, 2022, and the triggers to notify and suspend the warehouse at the specified thresholds. Option A is incorrect because it does not specify the frequency. Option C is incorrect because it does not specify the frequency and it uses notify and suspend instead of suspend and suspend_immediate. Option D is incorrect because it does not specify the start timestamp. For more information about resource monitors, see Working with Resource Monitors and CREATE RESOURCE MONITOR.

NEW QUESTION 29

Which commands can be performed by a user with the ORGADMIN role but not the ACCOUNTADMIN role? (Select TWO).
* SHOW REGIONS;
* SHOW USERS;
* SHOW ORGANIZATION ACCOUNTS;
* GRANT ROLE ORGADMIN TO USER <username>;
* SELECT SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER (

‘ACCOUNT LOCATOR’,

‘ENABLE ACCOUNT DATABASE_REPLICATION’,

‘true’

);
According to the Snowflake documentation1, the ORGADMIN role is a special system role that is responsible for managing operations at the organization level, such as creating and viewing accounts, enabling database replication, and setting global account parameters. The ACCOUNTADMIN role is a system role that is responsible for managing operations at the account level, such as creating and managing users, roles, warehouses, databases, and shares. Therefore, the commands that can be performed by the ORGADMIN role but not the ACCOUNTADMIN role are:

* SHOW ORGANIZATION ACCOUNTS: This command lists all the accounts in the organization and their properties, such as region, edition, and status2. The ACCOUNTADMIN role can only show the current account and its properties using the SHOW ACCOUNTS command3.

* SELECT SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER: This function sets a global account parameter for an account in the organization, such as enabling account database replication4. The ACCOUNTADMIN role can only set local account parameters using the ALTER ACCOUNT command.

Option A is incorrect because the SHOW REGIONS command can be executed by any role, not just the ORGADMIN role. Option B is incorrect because the SHOW USERS command can be executed by the ACCOUNTADMIN role, as well as any role that has been granted the MONITOR privilege on the account. Option D is incorrect because the GRANT ROLE ORGADMIN TO USER <username> command can be executed by the ACCOUNTADMIN role, as well as any role that has been granted the ORGADMIN role1.

NEW QUESTION 30

If the query matches the definition, will Snowflake always dynamically rewrite the query to use a materialized view?
* No, because joins are not supported by materialized views.
* No, because the optimizer might decide against it.
* No, because the materialized view may not be up-to-date.
* Yes, because materialized views are always faster.

Snowflake’s query optimizer can automatically rewrite queries against the base table or regular views to use the materialized view instead, if the query matches the definition of the materialized view1. However, this is not always guaranteed, as the optimizer might decide against using the materialized view based on various factors, such as the freshness of the data, the size of the result set, the complexity of the query, and the availability of the materialized view2. Therefore, the answer is no, because the optimizer might decide against it.

NEW QUESTION 31

An Administrator has a user who needs to be able to suspend and resume a task based on the current virtual warehouse load, but this user should not be able to modify the task or start a new run.

What privileges should be granted to the user to meet these requirements? (Select TWO).
* EXECUTE TASK on the task
* OWNERSHIP on the task
* OPERATE on the task
* USAGE on the database and schema containing the task
* OWNERSHIP on the database and schema containing the task

The user needs the OPERATE privilege on the task to suspend and resume it, and the USAGE privilege on the database and schema containing the task to access it1. The EXECUTE TASK privilege is not required for suspending and resuming a task, only for triggering a new run1. The OWNERSHIP privilege on the task or the database and schema would allow the user to modify or drop the task, which is not desired.

NEW QUESTION 32

A Snowflake Administrator wants to create a virtual warehouse that supports several dashboards, issuing various queries on the same database.

For this warehouse, why should the Administrator consider setting AUTO_SUSPEND to 0 or NULL?
* To keep the query result cache warm for good performance on repeated queries
* To save costs by running the warehouse as little as possible
* To keep the data cache warm to support good performance of similar queries
* To save costs on warehouse shutdowns and startups for different queries

NEW QUESTION 33

The ACCOUNTADMIN of Account 123 works with Snowflake Support to set up a Data Exchange. After the exchange is populated with listings from other Snowflake accounts, what roles in Account 123 are allowed to request and get data?

* Only the ACCOUNTADMIN role, and no other roles
* Any role with USAGE privilege on the Data Exchange
* Any role with IMPORT SHARE and CREATE DATABASE privileges
* Any role that the listing provider has designated as authorized

To request and get data from a Data Exchange, the role in Account 123 must have the USAGE privilege on the Data Exchange object. This privilege allows the role to view the listings and request access to the data. According to the Snowflake documentation, “To view the listings in a data exchange, a role must have the USAGE privilege on the data exchange object. To request access to a listing, a role must have the USAGE privilege on the data exchange object and the IMPORT SHARE privilege on the account.” The other options are either incorrect or not sufficient to request and get data from a Data Exchange. Option A is incorrect, as the ACCOUNTADMIN role is not the only role that can request and get data, as long as other roles have the necessary privileges. Option C is incorrect, as the IMPORT SHARE and CREATE DATABASE privileges are not required to request and get data, but only to create a database from a share after the access is granted. Option D is incorrect, as the listing provider does not designate the authorized roles in Account 123, but only approves or denies the requests from Account 123.