ARA-C01 Q&As

SnowPro Advanced: Architect Certification Exam

Pass Snowflake ARA-C01 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/ara-c01.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Snowflake Official Exam Center

QUESTION 1

What are some of the characteristics of result set caches? (Choose three.)

A. Time Travel queries can be executed against the result set cache.

B. Snowflake persists the data results for 24 hours.

C. Each time persisted results for a query are used, a 24-hour retention period is reset.

D. The data stored in the result cache will contribute to storage costs.

E. The retention period can be reset for a maximum of 31 days.

F. The result set cache is not shared between warehouses.

Correct Answer: BCE

Explanation: Comprehensive and Detailed Explanation: According to the SnowPro Advanced: Architect documents and learning resources, some of the characteristics of result set caches are: Snowflake persists the data results for 24 hours. This means that the result set cache holds the results of every query executed in the past 24 hours, and can be reused if the same query is submitted again and the underlying data has not changed1. Each time persisted results for a query are used, a 24-hour retention period is reset. This means that the result set cache extends the lifetime of the results every time they are reused, up to a maximum of 31 days from the date and time that the query was first executed1. The retention period can be reset for a maximum of 31 days. This means that the result set cache will purge the results after 31 days, regardless of whether they are reused or not. After 31 days, the next time the query is submitted, a new result is generated and persisted1. The other options are incorrect because they are not characteristics of result set caches. Option A is incorrect because Time Travel queries cannot be executed against the result set cache. Time Travel queries use the AS OF clause to access historical data that is stored in the storage layer, not the result set cache2. Option D is incorrect because the data stored in the result set cache does not contribute to storage costs. The result set cache is maintained by the service layer, and does not incur any additional charges1. Option F is incorrect because the result set cache is shared between warehouses. The result set cache is available across virtual warehouses, so query results returned to one user are available to any other user on the system who executes the same query, provided the underlying data has not changed1. References: Using Persisted Query Results | Snowflake Documentation, Time Travel | Snowflake Documentation

QUESTION 2

How can an Architect enable optimal clustering to enhance performance for different access paths on a given table?

A. Create multiple clustering keys for a table.

B. Create multiple materialized views with different cluster keys.

C. Create super projections that will automatically create clustering.

D. Create a clustering key that contains all columns used in the access paths.

Correct Answer: B

Explanation: According to the SnowPro Advanced: Architect documents and learning resources, the best way to enable optimal clustering to enhance performance for different access paths on a given table is to create multiple materialized views with different cluster keys. A materialized view is a pre-computed result set that is derived from a query on one or

more base tables. A materialized view can be clustered by specifying a clustering key, which is a subset of columns or expressions that determines how the data in the materialized view is co-located in micro-partitions. By creating multiple materialized views with different cluster keys, an Architect can optimize the performance of queries that use different access paths on the same base table. For example, if a base table has columns A, B, C, and D, and there are queries that filter on A and B, or on C and D, or on A and C, the Architect can create three materialized views, each with a different cluster key: (A, B), (C, D), and (A, C). This way, each query can leverage the optimal clustering of the corresponding materialized view and achieve faster scan efficiency and better compression. References: Snowflake Documentation: Materialized Views Snowflake Learning: Materialized Views https://www.snowflake.com/blog/using-materialized-views-to-solve-multi-clustering- performance-problems/

QUESTION 3

A large manufacturing company runs a dozen individual Snowflake accounts across its business divisions. The company wants to increase the level of data sharing to support supply chain optimizations and increase its purchasing leverage with multiple vendors.

The company\\'s Snowflake Architects need to design a solution that would allow the business divisions to decide what to share, while minimizing the level of effort spent on configuration and management. Most of the company divisions use Snowflake accounts in the same cloud deployments with a few exceptions for European-based divisions.

According to Snowflake recommended best practice, how should these requirements be met?

A. Migrate the European accounts in the global region and manage shares in a connected graph architecture. Deploy a Data Exchange.

B. Deploy a Private Data Exchange in combination with data shares for the European accounts.

C. Deploy to the Snowflake Marketplace making sure that invoker_share() is used in all secure views.

D. Deploy a Private Data Exchange and use replication to allow European data shares in the Exchange.

Correct Answer: B

Explanation: According to Snowflake recommended best practice, the requirements of the large manufacturing company should be met by deploying a Private Data Exchange in combination with data shares for the European accounts. A Private Data Exchange is a feature of the Snowflake Data Cloud platform that enables secure and governed sharing of data between organizations. It allows Snowflake customers to create their own data hub and invite other parts of their organization or external partners to access and contribute data sets. A Private Data Exchange provides centralized management, granular access control, and data usage metrics for the data shared in the exchange1. A data share is a secure and direct way of sharing data between Snowflake accounts without having to copy or move the data. A data share allows the data provider to grant privileges on selected objects in their account to one or more data consumers in other accounts2. By using a Private Data Exchange in combination with data shares, the company can achieve the following benefits: The business divisions can decide what data to share and publish it to the Private Data Exchange, where it can be discovered and accessed by other members of the exchange. This reduces the effort and complexity of managing multiple data sharing relationships and configurations. The company can leverage the existing Snowflake accounts in the same cloud deployments to create the Private Data Exchange and invite the members to join. This minimizes the migration and setup costs and leverages the existing Snowflake features and security. The company can use data shares to share data with the European accounts that are in different regions or cloud platforms. This allows the company to comply with the regional and regulatory requirements for data sovereignty and privacy, while still enabling data collaboration across the organization. The company can use the Snowflake Data Cloud platform to perform data analysis and transformation on the shared data, as well as integrate with other data sources and applications. This enables the company to optimize its supply chain and increase its purchasing leverage with multiple vendors. The other options are incorrect because they do not meet the requirements or follow the best practices. Option A is incorrect because migrating the European accounts to the global region may violate the data sovereignty and privacy regulations, and deploying a Data Exchange may not provide the level of control and management that the

company needs. Option C is incorrect because deploying to the Snowflake Marketplace may expose the company\\'s data to unwanted consumers, and using invoker_share() in secure views may not provide the desired level of security and governance. Option D is incorrect because using replication to allow European data shares in the Exchange may incur additional costs and complexity, and may not be necessary if data shares can be used instead. References: Private Data Exchange | Snowflake Documentation, Introduction to Secure Data Sharing | Snowflake Documentation

QUESTION 4

An Architect has been asked to clone schema STAGING as it looked one week ago, Tuesday June 1st at 8:00 AM, to recover some objects.

The STAGING schema has 50 days of retention.

The Architect runs the following statement:

CREATE SCHEMA STAGING_CLONE CLONE STAGING at (timestamp => \\'2021-06-01 08:00:00\\');

The Architect receives the following error: Time travel data is not available for schema STAGING. The requested time is either beyond the allowed time travel period or before the object creation time.

The Architect then checks the schema history and sees the following:

CREATED_ON|NAME|DROPPED_ON

2021-06-02 23:00:00 | STAGING | NULL

2021-05-01 10:00:00 | STAGING | 2021-06-02 23:00:00

How can cloning the STAGING schema be achieved?

A. Undrop the STAGING schema and then rerun the CLONE statement.

B. Modify the statement: CREATE SCHEMA STAGING_CLONE CLONE STAGING at (timestamp => \\'2021-05-01 10:00:00\\');

C. Rename the STAGING schema and perform an UNDROP to retrieve the previous STAGING schema version, then run the CLONE statement.

D. Cloning cannot be accomplished because the STAGING schema version was not active during the proposed Time Travel time period.

Correct Answer: C

The error message indicates that the schema STAGING does not have time travel data available for the requested timestamp, because the current version of the schema was created on2021-06-02 23:00:00, which is after the timestamp of 2021-06-01 08:00:00. Therefore, the CLONE statement cannot access the historical data of the schema at that point in time. Option A is incorrect, because undropping the STAGING schema will not restore the previous version of the schema that was active on 2021-06-01 08:00:00. Instead, it will create a new version of the schema with the same name and no data or objects. Option B is incorrect, because modifying the timestamp to 2021-05-01 10:00:00 will not clone the schema as it looked one week ago, but as it looked when it was first created. This may not reflect the desired state of the schema and its objects. Option C is correct, because renaming the STAGING schema and performing an UNDROP to retrieve the previous STAGING schema version will restore the schema that was dropped on 2021-06-02 23:00:00. This schema has time travel data available for the requested timestamp of 2021-06-01 08:00:00, and can be cloned using the CLONE statement. Option D is incorrect, because cloning can be accomplished by using the UNDROP command to access the previous version of the schema that was active during the proposed time travel

period. References: : Cloning Considerations : Understanding and Using Time Travel : CREATE ... CLONE

QUESTION 5

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?

A. 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.

B. 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.

C. 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.

D. 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.

Correct Answer: C

Explanation: 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 cleanup12. References: Snowflake Documentation: Stored Procedures Snowflake Documentation: Understanding Caller\\'s Rights and Owner\\'s Rights Stored Procedures