

Snowflake ARA-C01 PDF Questions—Ideal Material for Quick Preparation

Snowflake ARA-C01 Practice Questions

SnowPro Advanced Architect Certification

Order our ARA-C01 Practice Questions Today and Get Ready to Pass with Flying Colors!



ARA-C01 Practice Exam Features | QuestionsTube

- Latest & Updated Exam Questions
- Subscribe to FREE Updates
- Both PDF & Exam Engine
- Download Directly Without Waiting

<https://www.questionstube.com/exam/ara-c01/>

At QuestionsTube, you can read ARA-C01 free demo questions in pdf file, so you can check the questions and answers before deciding to download the Snowflake ARA-C01 practice questions. These free demo questions are parts of the ARA-C01 exam questions. Download and read them carefully, you will find that the ARA-C01 test questions of QuestionsTube will be your great learning materials online. Share some ARA-C01 exam online questions below.

1. An Architect needs to allow a user to create a database from an inbound share.

P.S. Free 2025 Snowflake ARA-C01 dumps are available on Google Drive shared by Itcertkey: https://drive.google.com/open?id=11-_7bO_C-9dV9YnYpwxlg4i4UvAvqGe

It is generally acknowledged that candidates who earn the SnowPro Advanced Architect Certification (ARA-C01) certification ultimately get high-paying jobs in the tech market. Success in the SnowPro Advanced Architect Certification (ARA-C01) exam not only validates your skills but also helps you get promotions. To pass the SnowPro Advanced Architect Certification test in a short time, you must prepare with ARA-C01 Exam Questions that are real and updated. Without studying with ARA-C01 actual questions, candidates fail and waste their time and money.

Snowflake ARA-C01 Exam is divided into four different sections, each focusing on a specific area of Snowflake architecture. These sections include Snowflake Architecture and Design, Data Warehousing, Data Processing, and Data Integration. ARA-C01 exam is a computer-based test and is administered through a third-party exam provider.

Snowflake ARA-C01 (SnowPro Advanced Architect Certification) Certification Exam is a cloud-based certification exam that is designed to validate the advanced skills and knowledge of Snowflake architects. SnowPro Advanced Architect Certification certification exam is intended for those professionals who possess a deep understanding of Snowflake data warehouses and their architecture, and who can design and implement complex Snowflake solutions using best practices. The SnowPro Advanced Architect Certification Exam is a vendor-neutral certification, which means that it is not affiliated with any particular vendor or technology.

Perfect ARA-C01 Cert Guide Supply you Fantastic Test Prep for ARA-C01: SnowPro Advanced Architect Certification to Prepare easily

At least 2/3 top 500 global companies choose Snowflake electronic business software products as their key products or daily use. So if you get a Snowflake certification you will be outstanding over others. Candidates want to pass ARA-C01 exam, the fastest and convenient method is to use our ARA-C01 Study Guide, many candidates choose this method to pass exam. You also can make this as practice exam materials or use test engine file to test like the real test scene.

Snowflake SnowPro Advanced Architect Certification Sample Questions (Q28-Q33):

NEW QUESTION # 28

A table, EMP_TBL has three records as shown:

```
create or replace TABLE EMP_TBL (
  ID NUMBER(38,0),
  NAME VARCHAR(16777216)
);
```

ID	NAME
1	Name1
2	Name2
3	Name3

The following variables are set for the session:

```
set tbl_ref = EMP_TBL;
set col_ref = 'NAME';
set (var1, var2, var3) = (select 'Name1', 'Name2', 'Name3');
```

Which SELECT statements will retrieve all three records? (Select TWO).

- A. **SELECT * FROM EMP_TBL WHERE identifier(Scol_ref) IN ('Name1','Name2', 'Name3');**
- B. **Select * FROM \$tbl_ref WHERE \$col_ref IN ('Name1','Name2','Name3');**
- C. **SELECT * FROM identifier(\$tbl_ref) WHERE ID IN Cvar1,'var2','var3');**
- D. **SELECT * FROM identifier(\$tbl_ref) WHERE NAME IN (\$var1, \$var2, \$var3);**
- E. **SELECT * FROM \$tbl_ref WHERE \$col_ref IN (\$var1, \$var2, \$var3);**

Answer: A,E

Explanation:

- * The correct answer is B and E because they use the correct syntax and values for the identifier function and the session variables.
- * The identifier function allows you to use a variable or expression as an identifier (such as a table name or column name) in a SQL statement. It takes a single argument and returns it as an identifier. For example, identifier(\$tbl_ref) returns EMP_TBL as an identifier.
- * The session variables are set using the SET command and can be referenced using the \$ sign. For example, \$var1 returns Name1 as a value.
- * Option A is incorrect because it uses \$tbl_ref and \$col_ref, which are not valid session variables or identifiers. They should be \$tbl_ref and \$col_ref instead.
- * Option C is incorrect because it uses identifier(\$tbl_ref), which is not a valid syntax for the identifier function. It should be identifier(\$tbl_ref) instead.
- * Option D is incorrect because it uses Cvar1, var2, and var3, which are not valid session variables or values. They should be \$var1, \$var2, and \$var3 instead. References:
- * Snowflake Documentation: Identifier Function

- * Snowflake Documentation: Session Variables
- * Snowflake Learning: SnowPro Advanced: Architect Exam Study Guide

NEW QUESTION # 29

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

- A. Create a clustering key that contains all columns used in the access paths.
- B. Create multiple clustering keys for a table.
- C. Create multiple materialized views with different cluster keys.
- D. Create super projections that will automatically create clustering.

Answer: C

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/>

NEW QUESTION # 30

Company A would like to share data in Snowflake with Company B. Company B is not on the same cloud platform as Company A. What is required to allow data sharing between these two companies?

- A. Create a pipeline to write shared data to a cloud storage location in the target cloud provider.
- B. Setup data replication to the region and cloud platform where the consumer resides.
- C. Company A and Company B must agree to use a single cloud platform. Data sharing is only possible if the companies share the same cloud provider.
- D. Ensure that all views are persisted, as views cannot be shared across cloud platforms.

Answer: B

Explanation:

According to the SnowPro Advanced: Architect documents and learning resources, the requirement to allow data sharing between two companies that are not on the same cloud platform is to set up data replication to the region and cloud platform where the consumer resides. Data replication is a feature of Snowflake that enables copying databases across accounts in different regions and cloud platforms. Data replication allows data providers to securely share data with data consumers across different regions and cloud platforms by creating a replica database in the consumer's account. The replica database is read-only and automatically synchronized with the primary database in the provider's account. Data replication is useful for scenarios where data sharing is not possible or desirable due to latency, compliance, or security reasons¹. The other options are incorrect because they are not required or feasible to allow data sharing between two companies that are not on the same cloud platform. Option A is incorrect because creating a pipeline to write shared data to a cloud storage location in the target cloud provider is not a secure or efficient way of sharing data. It would require additional steps to load the data from the cloud storage to the consumer's account, and it would not leverage the benefits of Snowflake's data sharing features. Option B is incorrect because ensuring that all views are persisted is not relevant for data sharing across cloud platforms. Views can be shared across cloud platforms as long as they reference objects in the same database. Persisting views is an option to improve the performance of querying views, but it is not required for data sharing². Option D is incorrect because Company A and Company B do not need to agree to use a single cloud platform. Data sharing is possible across different cloud platforms using data replication or other methods, such as listings or auto-fulfillment³. Reference: Replicating Databases Across Multiple Accounts | Snowflake Documentation, Persisting Views | Snowflake Documentation, Sharing Data Across Regions and Cloud Platforms | Snowflake Documentation

NEW QUESTION # 31

A table, EMP_TBL has three records as shown:

```
create or replace TABLE EMP_TBL (  
  ID NUMBER(38,0),  
  NAME VARCHAR(16777216)  
);
```

ID	NAME
1	Name1
2	Name2
3	Name3

The following variables are set for the session:

```
set tbl_ref = 'EMP_TBL';  
set col_ref = 'NAME';  
set (var1, var2, var3) = (select 'Name1', 'Name2', 'Name3');
```

Which SELECT statements will retrieve all three records? (Select TWO).

- A. **SELECT * FROM EMP_TBL WHERE identifier(Scol_ref) IN ('Name1','Name2', 'Name3');**
- B. Select * FROM Stbl_ref WHERE Scol_ref IN ('Name1','Name2','Name3');
- C. SELECT * FROM identifier(\$tbl_ref) WHERE ID IN Cvar1,'var2','var3');
- D. SELECT * FROM identifier<Stbl_ref> WHERE NAME IN (\$var1, \$var2, \$var3);
- E. **SELECT * FROM \$tbl_ref WHERE \$col_ref IN (\$var1, \$var2, \$var3);**

Answer: A,E

Explanation:

- * The correct answer is B and E because they use the correct syntax and values for the identifier function and the session variables.
- * The identifier function allows you to use a variable or expression as an identifier (such as a table name or column name) in a SQL statement. It takes a single argument and returns it as an identifier. For example, identifier(\$tbl_ref) returns EMP_TBL as an identifier.
- * The session variables are set using the SET command and can be referenced using the \$ sign. For example, \$var1 returns Name1 as a value.
- * Option A is incorrect because it uses Stbl_ref and Scol_ref, which are not valid session variables or identifiers. They should be \$tbl_ref and \$col_ref instead.
- * Option C is incorrect because it uses identifier<Stbl_ref>, which is not a valid syntax for the identifier function. It should be identifier(\$tbl_ref) instead.
- * Option D is incorrect because it uses Cvar1, var2, and var3, which are not valid session variables or values. They should be \$var1, \$var2, and \$var3 instead. References:
- * Snowflake Documentation: Identifier Function
- * Snowflake Documentation: Session Variables
- * Snowflake Learning: SnowPro Advanced: Architect Exam Study Guide

NEW QUESTION # 32

An Architect has chosen to separate their Snowflake Production and QA environments using two separate Snowflake accounts. The QA account is intended to run and test changes on data and database objects before pushing those changes to the Production account. It is a requirement that all database objects and data in the QA account need to be an exact copy of the database objects, including privileges and data in the Production account on at least a nightly basis.

Which is the LEAST complex approach to use to populate the QA account with the Production account's data and database objects on a nightly basis?

- A. 1) Create a share in the Production account for each database
2) Share access to the QA account as a Consumer
3) The QA account creates a database directly from each share
4) Create clones of those databases on a nightly basis
5) Run tests directly on those cloned databases

- Answer: A**

• • • • •

Test ARA-C01 Prep: https://www.itcertkey.com/ARA-C01_braindumps.html

- P.S. Free 2025 Snowflake ARA-C01 dumps are available on Google Drive shared by Itcertkey: <https://drive.google.com/open?id=1l-7bO-C-9dV9YnYpwxlg4i4UvAvqGe>