

A grayscale photograph of a man with dark hair, wearing large over-ear headphones. He is sitting at a desk, looking intently at a laptop screen. His hands are clasped together under his chin, suggesting deep concentration or contemplation. The background is blurred, showing what appears to be a dimly lit room with some furniture. Overlaid on the center of the image is white text in a serif font.

# TIME TRAVEL, FAILSAFE & ZERO COPY CLONES



# Time Travel

# Time Travel

Snowflake Time Travel enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period.

- Restoring data-related objects (tables, schemas, and databases) that may have been accidentally or intentionally deleted.
- Duplicating and backing up data from key points in the past.
- Analyzing data usage/manipulation over specified periods of time.





To support Time Travel, the following SQL extensions have been implemented:

**AT | BEFORE** clause which can be specified in SELECT statements and CREATE ... CLONE commands

**TIMESTAMP**

**OFFSET** (time difference in seconds from the present time)

**STATEMENT** (identifier for statement, e.g. query ID)

**UNDROP** command for tables, schemas, and databases.

**Hands On Demo : Time Travel**



## Snowflake Data Lifecycle with Time Travel

### Organizing Data

CREATE DATABASE ...  
ALTER DATABASE ...

CREATE SCHEMA ...  
ALTER SCHEMA ...

CREATE TABLE ...  
ALTER TABLE ...

### Storing Data

INSERT INTO <table> ...  
COPY INTO <table> ...

### Querying Data

SELECT FROM <table> ... AT | BEFORE

### Working with Data

UPDATE <table> ...  
MERGE INTO <table> ...  
DELETE FROM <table> ...

CREATE TABLE ... CLONE ... AT | BEFORE

CREATE SCHEMA ... CLONE ... AT | BEFORE

CREATE DATABASE ... CLONE ... AT | BEFORE

### Removing Data

TRUNCATE TABLE ...  
DROP TABLE ...  
UNDROP TABLE ...

DROP SCHEMA ...  
UNDROP SCHEMA ...

DROP DATABASE ...  
UNDROP DATABASE ...

DDL COMMANDS

DML COMMANDS

TIME TRAVEL EXTENSIONS

# Changing the Data Retention Period for an Object

The `DATA_RETENTION_TIME_IN_DAYS` object parameter can be used by users with the `ACCOUNTADMIN` role to set the default retention period for your account.

Increasing Retention  
Decreasing Retention

Changing the retention period for your account or individual objects changes the value for all lower-level objects that do not have a retention period explicitly set



Fail Safe

# Fail-safe

Fail-safe provides a (non-configurable) 7-day period during which historical data is recoverable by Snowflake. This period starts immediately after the Time Travel retention period ends.







# Zero Copy Clones

# Zero Copy Clone

Creates a copy of an existing object in the system

For databases, schemas, and non-temporary tables, CLONE supports an additional AT | BEFORE clause for cloning using Time Travel.

A person is working on a laptop, with a plant visible in the background. The image is faded and serves as a background for the text.

For tables, Snowflake only supports cloning permanent and transient tables; temporary tables cannot be cloned.

For databases and schemas, cloning is recursive

The following object types are not cloned:

- External tables

- Internal (Snowflake) stages

A cloned table does not include the load history of the source table. Data files that were loaded into a source table can be loaded again into its clones.

Cloning is not instantaneous, particularly for large objects (databases, schemas, tables), and does not lock the object being cloned.





QUIZ





If you change the retention period at the account level, then

- a. all databases, schemas, and tables that do not have an explicit retention period automatically inherit the new retention period.
- b. all databases, schemas, and tables automatically inherit the new retention period.
- c. databases, schemas, and tables do not automatically inherit the new retention period. It needs to be redefined for each DB, Schema and table.

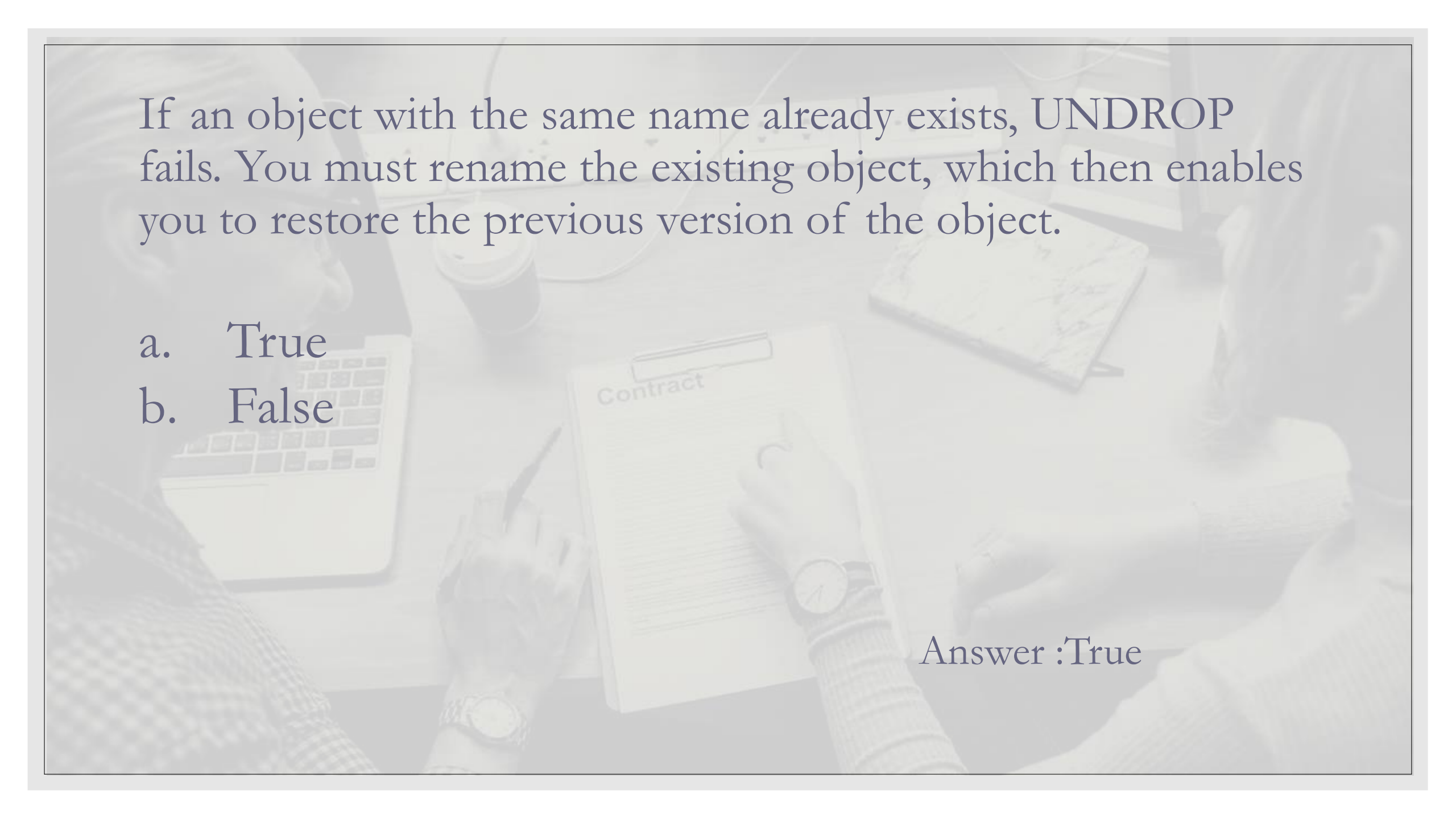
Answer : a



If in create clone table using `TIMESTAMP`, `OFFSET`, or `STATEMENT` specified in the `AT | BEFORE` clause falls outside the data retention period for the table, then

- a. The query fails and returns an error
- b. Query executes but table is not created
- c. Query executes and table is created

Answer : a

The background image shows a top-down view of a desk. On the left, a person's arm and hand are visible, wearing a watch and holding a pen over a document. The document has the word "Contract" at the top. In the center, there is a coffee cup. To the right, another person's hand is visible, also holding a pen. A laptop is partially visible on the left side of the desk. The overall scene suggests a business meeting or a signing process.

If an object with the same name already exists, UNDROP fails. You must rename the existing object, which then enables you to restore the previous version of the object.

- a. True
- b. False

Answer :True

Table A was Dropped and then immediately un  
dropped.

When you run show tables history for this table :

- a. There is only 1 entry for this table and  
Dropped\_on\_date column is NULL
- b. There are 2 versions for this table. 1 version  
has Dropped\_on\_date column as NULL and  
other as datetime when it was dropped

Answer :a





The following object types cannot be cloned:

- a. DB
- b. Table
- c. Internal Stage
- d. schema

Answer :c

A background image showing two people sitting at a desk. On the desk is a laptop, a coffee cup, a notebook, and a document titled 'Contract'. One person is pointing at the contract with a pen. The image is faded and serves as a background for the text.

Data files that were loaded into a source table can be loaded again into its clones.

- a. True
- b. False

Answer :True

# Recap



## Time Travel

Snowflake Time Travel enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period.

**AT | BEFORE** clause which can be specified in **SELECT** statements and **CREATE ... CLONE** commands

**TIMESTAMP**

**OFFSET** (time difference in seconds from the present time)

**STATEMENT** (identifier for statement, e.g. query ID)

**UNDROP** command for tables, schemas, and databases

## Fail-safe

Fail-safe provides a (non-configurable) 7-day period during which historical data is recoverable by Snowflake.

This period starts immediately after the Time Travel retention period ends.



## Zero Copy Clones

Creates a copy of an existing object in the system

For databases, schemas, and non-temporary tables, CLONE supports an additional AT | BEFORE clause for cloning using Time Travel.

For tables, Snowflake only supports cloning permanent and transient tables; temporary tables cannot be cloned.

For databases and schemas, cloning is recursive

The following object types are not cloned:

- External tables

- Internal (Snowflake) stages

A cloned table does not include the load history of the source table. Data files that were loaded into a source table can be loaded again into its clones.

Cloning is not instantaneous, particularly for large objects (databases, schemas, tables), and does not lock the object being cloned.