**Backup / Recovery / Clone/Clone from time travel**

Documenting the full detailed steps required to Create a clone or any Backup/ Recovery of Object from the Time Travel.

>>Cloning  of schema from Time Travel.

>>Cloning of Table from Time Travel.

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**>>**Initial validation of of retention Period -- **DATA\_RETENTION\_TIME\_IN\_DAYS**:

show parameters;

show parameters in database SALES;

show parameters like '%DATA%';

show parameters like 'DATA%' in account;

--CHANGES DATA RETENTION FOR THE DATABASE/SCHEMA if not set

show parameters in database SALES;

--CHANGES DATA RETENTION FOR THE DATABASE

show parameters in database SALES;

---alter DATABASE SALES set DATA\_RETENTION\_TIME\_IN\_DAYS = 7 ;

---alter SCHEMA SALES.SNOW set DATA\_RETENTION\_TIME\_IN\_DAYS = 7 ;

--CHANGES DATA RETENTION FOR THE WHOLE ACCOUNT

show parameters like 'DATA%' in account;

--alter ACCOUNT set DATA\_RETENTION\_TIME\_IN\_DAYS = 1 ;

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

select \* from PATIENTS;

INSERT INTO PATIENTS values (5,'TEST PATIENT','AMSTERDAM','Mayo Clinic') select \* from PATIENTS at(offset => -60\*5);

delete from PATIENTS where id=5;

drop table sales.snow.PATIENTS ;

--Over a day

select \* from PATIENTS at(offset => -60\*500000);

select \* from PATIENTS at(timestamp => 'Sat, 16 Mar 2019 16:20:00 -0700'::timestamp);

==============================================================================================================================================================================

**>>**CLONING SCHMEAS based on time,query\_id

SYNTAX:

CREATE [OR REPLACE] SCHEMA <database\_name>.<schema\_name> CLONE <database\_name>.<schema\_name> [ { **AT** | **BEFORE** } ( { **TIMESTAMP** => <timestamp> | **OFFSET** => <time\_difference> | **STATEMENT** => <id> } ) ] ]

Example:

--Schema as existed 1 hour before

CREATE SCHEMA restored\_schema CLONE snow at(offset => -3600);

**>>**--Before changes make by a specific query

select \* from patients before(statement => '025e545d-fc23-4e8d-9ac5-335943a1bec2');

CREATE SCHEMA restored\_schema CLONE patient\_schema before(statement => '025e545d-fc23-4e8d-9ac5-335943a1bec2');

CREATE SCHEMA restored\_schema CLONE patient\_schema at(timestamp => 'Mon, 09 May 2022 01:01:00 +0300'::timestamp);

**>>For Table**

create table restored\_table clone user\_access at(timestamp => 'Mon, 09 May 2019 01:01:00 +0300'::timestamp);

create table restored\_table clone user\_access at(offset => -60\*10);

select \* from restored\_table;

**>>**For Database

**CREATE** [ **OR** **REPLACE** ] **DATABASE** [ **IF** **NOT** **EXISTS** ] <database\_name> [ **CLONE** <database\_name> [ { **AT** | **BEFORE** } ( { **TIMESTAMP** => <timestamp> | **OFFSET** => <time\_difference> | **STATEMENT** => <id> } ) ] ]

create database restored\_db clone sales before(statement => '025e545d-fc23-4e8d-9ac5-335943a1bec2');

DROP database restored\_db;

show tables history in sales.snow;

show tables history in sales;---Restoring objects

**>>Cloning Table/ Schema Using Time Travel**

**--STEP 1 :**

Verify the data at a particular period you require the data

SELECT \* FROM PHCDW.PHCDW\_RPTG\_CAR.RPTG\_NGSF\_ADDR\_DIM at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

--CREATE OR REPLACE SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_CAR CLONE PHCDW.PHCDW\_RPTG\_CAR at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

--CREATE OR REPLACE SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_COMMON CLONE PHCDW.PHCDW\_RPTG\_COMMON at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

--CREATE OR REPLACE SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_HEM CLONE PHCDW.PHCDW\_RPTG\_HEM at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

--CREATE OR REPLACE SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_ONC CLONE PHCDW.PHCDW\_RPTG\_ONC at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

--CREATE OR REPLACE SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_WHC CLONE PHCDW.PHCDW\_RPTG\_WHC at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

--CREATE OR REPLACE TABLE CPH\_DB\_CLONE\_PROD.ANALYTICS.DIM\_ACCOUNT CLONE CPH\_DB\_PROD.ANALYTICS.DIM\_ACCOUNT at(timestamp=>'Wed, 05 Jul 2023 18:00:00 -0700'::timestamp);

//IF There is a requirement of Sharing a Cloned Database/schema which have been created using Time Travel or as it in current. Follow Below Process.

**--STEP 2 :**

This step is only for specific schemas for which RLS (Row Level Security) is enabled. Only drop policy if it comes under clone database/schema,

Drop Masking Policy from the Tables

Drop Tags from the Tables.

In case Of CYRUS and DAYLIGHT No policy is attached, but if any Tables/ Views dependent or joined then Policy needs to be dropped from respective table/view before sharing.

Follow Below Documentation for More Information on dropping policies and Tag.

[PROD to QA Data Refresh Documentation](https://cognizantonline-my.sharepoint.com/personal/856976_cognizant_com/Documents/Desktop/DBA%20TASKS%20and/PROD%20to%20QA/PROD%20to%20QA%20Data%20Refresh%20Documentation.docx?web=1)

Note: Do not drop policies from the Main Database which is used by business users ex: CPH\_DB\_PROD

Policy cannot be dropped at once from whole schema. It needs to be dropped one by one from table.

**--STEP 3 :**

GRANT SELECT ON TABLE CPH\_DB\_CLONE\_PROD.ANALYTICS.DIM\_ACCOUNT to share CDP\_CPH\_DB\_CLONE\_PROD\_SHARE;

GRANT SELECT ON ALL TABLES IN SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_CAR TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT SELECT ON ALL TABLES IN SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_COMMON TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT SELECT ON ALL TABLES IN SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_HEM TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT SELECT ON ALL TABLES IN SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_ONC TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT SELECT ON ALL TABLES IN SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_WHC TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT USAGE ON SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_CAR TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT USAGE ON SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_COMMON TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT USAGE ON SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_HEM TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT USAGE ON SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_ONC TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE;

GRANT USAGE ON SCHEMA PHCDW\_CLONE\_DB\_PROD\_03072023.PHCDW\_RPTG\_WHC TO SHARE CDP\_PHCDW\_DAYLIGHT\_DB\_03072023\_SHARE

================================================================================================================================================================================

**>>Time Travel ----undrop tables**

undrop table patients;

undrop schema snow;

undrop database sales;

**SHOW DATABASES** **HISTORY;**

SHOW TABLES HISTORY LIKE 'load%' IN mytestdb.myschema;

SHOW SCHEMAS HISTORY IN some\_db;

SHOW DATABASES HISTORY;

In order to recover the dropped table, you will need to perform the following steps:

1. Rename the recreated table:

 alter table <table\_name> rename to <new\_name>;

1. Execute UNDROP to reclaim the dropped table:

undrop table <original\_name>;

**NOTE:**

* Tables can be recovered within the Data Retention Period. The standard retention period is 1 day (24 hours) and is automatically enabled for all Snowflake accounts.
* Restoring tables is only supported in the current schema or current database, even if a fully-qualified table name is specified.

--1 Preparing table "TABLE\_COLUMN\_MAPPING\_check" for testing undrop command  
**select** **count**(1) **from** METADATA.TABLE\_COLUMN\_MAPPING ;--203,045  
**create** table METADATA.TABLE\_COLUMN\_MAPPING\_check clone METADATA.TABLE\_COLUMN\_MAPPING   
**delete** **from** METADATA.TABLE\_COLUMN\_MAPPING\_Check **where** datasource\_name='IQVIA'  
;  
  
--2 Count in the table before replace command  
**select** **count**(1) **from** METADATA.TABLE\_COLUMN\_MAPPING\_Check;--193,357  
;  
  
--3 By mistakes replace of the table happens  
**create** or replace table METADATA.TABLE\_COLUMN\_MAPPING\_Check clone METADATA.TABLE\_COLUMN\_MAPPING;   
--data is changes in the table after replace  
**select** **count**(1) **from** METADATA.TABLE\_COLUMN\_MAPPING\_Check;--203,045  
  
--4 Process to regain the table  
-- Rename the existing table  
**alter** table METADATA.TABLE\_COLUMN\_MAPPING\_Check rename to METADATA.TABLE\_COLUMN\_MAPPING\_Check\_renamed;  
  
-- undrop the table which got replace in point 3  
undrop table METADATA.TABLE\_COLUMN\_MAPPING\_Check;  
  
-- data got restored (count is same as before replace)  
**select** **count**(1) **from** METADATA.TABLE\_COLUMN\_MAPPING\_Check;--193,357

**>>Restoring Schema in Snowflake**

>>Create Database in Snowflake

create or replace database [database-name] ;

create or replace database dezyre\_test ;

### >>Select Database

Syntax of the statement:

Use database [database-name];

use database dezyre\_test;

### >>Create Schema

Syntax of the statement:

Create schema schema-name;

Example:

use dezyre\_test;

Create schema myschema;

show schemas;

### >>Drop the Schema

Syntax of the statement:

DROP SCHEMA [ IF EXISTS ] [ CASCADE | RESTRICT ]

Example

DROP SCHEMA IF EXISTS myschema;

### >>Restore the Schema

Syntax:

UNDROP SCHEMA

Example:

UNDROP SCHEMA IF EXISTS myschema;

=========================================================================================================================================================================