**Jenkins Pipeline Considerations:**

**ALTER TABLE :-**

To alter the existing table using ALTER statement to perform the actions like adding columns, changing data type, increasing the length of data type – Modify the existing table DDL by changing CREATE TABLE <Table\_Name> to CREATE TABLE IF NOT EXISTS <Table\_Name> and add ALTER statements at the end of the table DDL.

**Link for the sample DDL:**

<https://github.caesars.com/Data-as-a-Service/DAAS_REPO/blob/UAT/sql/core/daas_core/ddl/table/digital_transactions_fact.sql>

* **Don’t** create separate Adhoc procs for running Alter statements.
* Currently dropping an existing column using ALTER statement is **not** supported via Jenkins.

**CREATE OR REPLACE :-**

CREATE OR REPLACE is **not supported** for Tables, Streams & Sequences when the code is deployed thru Jenkins.

**DROP :- Don’t DROP** any objects

Dropping objects are **not** allowed in Jenkins Pipeline. If the file contains DROP keyword, pipeline will not proceed the deployment and it comes under suspicious activity.

***Tip* :-** Always make sure the file that is checked-into Git is having UNIX conversion so that when comparing the code with lower environment, there won’t be any file size issues.

*In order to do, Go to Notepad++ -> Edit -> EOL Conversion -> UNIX (LF)*

**Snowflake Guidelines:**

**ADHOC Procedure :-**

* Always create the Adhoc procs with OWNER rights. (EXECUTE AS OWNER).
* Adhoc procedures are specifically created to perform any DML operations.

**CREATE\_OR\_REPLACE\_STREAM :-**

* “CREATE\_OR\_REPLACE\_STREAM” procedure is used to recreate REGULAR and APPEND\_ONLY streams.
* No need to place create or replace stream statement in Adhoc procedure.

**Syntax to create “Regular” stream:**

CALL DAAS\_COMMON.CREATE\_OR\_REPLACE\_STREAM('<Source\_Schema>', '<Source\_Table>', '<Target\_Schema>', '<Stream\_Name>', NULL);

**Ex:** CALL DAAS\_COMMON.CREATE\_OR\_REPLACE\_STREAM('DAAS\_CORE', 'TRIP\_DETAIL', 'DAAS\_CORE', 'TRIP\_DETAIL\_SUMMARY\_STREAM', NULL);

**Syntax to create “APPEND\_ONLY” stream:**

CALL DAAS\_COMMON.CREATE\_OR\_REPLACE\_STREAM('<Source\_Schema>', '<Source\_Table>', '<Target\_Schema>', '<Stream\_Name>', ‘APPEND\_ONLY’);

**Ex:** CALL DAAS\_COMMON.CREATE\_OR\_REPLACE\_STREAM('DAAS\_CORE', 'TRIP\_DETAIL', 'DAAS\_CORE', 'TRIP\_DETAIL\_SUMMARY\_STREAM',’APPEND\_ONLY’);

**CLONE\_TABLE\_PROC :-**

“CLONE\_TABLE\_PROC” procedure is used for :

* Taking table backups.
* Clone tables from PROD to UAT
* Clone tables from UAT to PROD

Backup tables are by default created in DAAS\_TEMP schema if the last 2 arguments are NULL.

**Table backup :-**

**Syntax:** CALL DAAS\_SECURITY.CLONE\_TABLE\_PROC('<Env\_Name>', '<Env\_Name>', '<Source\_Schema>', '<Source\_Table>', NULL, NULL);

**Ex:** CALL DAAS\_SECURITY.CLONE\_TABLE\_PROC('DAAS\_UAT', 'DAAS\_UAT', 'DAAS\_CORE', 'RATINGS\_FACT', NULL, NULL);

**Clone tables from PROD to UAT (***Run statement in UAT***)**

**Syntax:** CALL DAAS\_SECURITY.CLONE\_TABLE\_PROC('<Source\_DB>', '<Target\_DB>', 'Schema\_Name', '<Table\_Name>', NULL, 'CLONE FROM PROD TO UAT');

**Ex:** CALL DAAS\_SECURITY.CLONE\_TABLE\_PROC('DAAS\_PROD', 'DAAS\_UAT', 'DAAS\_RAW\_HETCMS', 'CMPRA\_RAW', NULL, 'CLONE FROM PROD TO UAT');

**Clone tables from UAT to PROD (***Run statement in PROD***)**

**Syntax:** CALL DAAS\_SECURITY.CLONE\_TABLE\_PROC('<Source\_DB>', '<Target\_DB>', 'Schema\_Name', '<Table\_Name>', NULL, 'CLONE\_FROM\_UAT\_TO\_PROD\_REQUIRED');

**Ex:** CALL DAAS\_SECURITY.CLONE\_TABLE\_PROC('DAAS\_UAT', 'DAAS\_PROD', 'DAAS\_RAW\_PDB', 'GST\_ID\_XREF\_RAW', NULL,'CLONE\_FROM\_UAT\_TO\_PROD\_REQUIRED');

**DATA\_REPLAY\_PROC :-**

“DATA\_REPLAY\_PROC” procedure is used to update REPLAY\_COUNTER from the tables.

**Syntax:** CALL DAAS\_COMMON.DATA\_REPLAY\_PROC('<Schema\_Name>', '<Table\_Name>', '<Filter\_Condition>', '<Activity>', <Threshold\_Value>);

**Ex:** CALL DAAS\_COMMON.DATA\_REPLAY\_PROC('DAAS\_RAW\_HETCMS', 'CMPRW\_RAW',

'WHERE DATE(OPERATION\_DATE) >= ''2023-08-22''',

'PROD\_RESTORE', 1000000000);

**ALTER\_TASK\_TIMEOUT\_PROC :-**

“ALTER\_TASK\_TIMEOUT\_PROC” procedure is used to update “USER\_TASK\_TIMEOUT\_MS” value within a task.

**Syntax:** CALL DAAS\_COMMON.ALTER\_TASK\_TIMEOUT\_PROC('<Schema\_Name>', '<Task\_Name>', <USER\_TASK\_TIMEOUT\_MS>);

**Ex:** CALL DAAS\_COMMON.ALTER\_TASK\_TIMEOUT\_PROC('DAAS\_COMMON', 'TRIP\_ROOT\_TASK', 72000000);

**Suspicious Activities - All Environments :**

If the file contains DROP keyword, deployment fails and triggers an email with the list of configured users.