-- SCD /CDC

show stages;

u taskadmin;

-- Set the active role to SECURITYADMIN to show that this role can grant a role to another role

use role securityadmin;

grant role taskadmin to role sysadmin;

use role sysadmin;

create warehouse if not exists task\_warehouse with warehouse\_size = 'XSMALL' auto\_suspend = 120;

-- Create a task to schedule the MERGE statement

create or replace task populate\_nation\_history warehouse = task\_warehouse schedule = '2 minute' when system$stream\_has\_data('nation\_table\_changes')

as

merge into nation\_history nh

using nation\_change\_data m

on nh.n\_nationkey = m.n\_nationkey

and nh.start\_time = m.start\_time

when matched and m.dml\_type = 'U' then update

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when matched and m.dml\_type = 'D' then update

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when not matched and m.dml\_type = 'I' then insert

(n\_nationkey, n\_name, n\_regionkey, n\_comment,

country\_code, start\_time, end\_time, current\_flag)

values (m.n\_nationkey, m.n\_name, m.n\_regionkey, m.n\_comment,

m.country\_code, m.start\_tise role sysadmin;

create database streams\_and\_tasks;

use database streams\_and\_tasks;

create schema scd;

use schema scd;

create or replace table nation (

n\_nationkey number,

n\_name varchar(25),

n\_regionkey number,

n\_comment varchar(152),

country\_code varchar(2),

update\_timestamp timestamp\_ntz);

create or replace table nation\_history (

n\_nationkey number,

n\_name varchar(25),

n\_regionkey number,

n\_comment varchar(152),

country\_code varchar(2),

start\_time timestamp\_ntz,

end\_time timestamp\_ntz,

current\_flag int);

create or replace stream nation\_table\_changes on table nation;

show streams;

select \* from nation\_table\_changes;

create or replace view nation\_change\_data as

-- This subquery figures out what to do when data is inserted into the NATION table

-- An insert to the NATION table results in an INSERT to the NATION\_HISTORY table

select n\_nationkey, n\_name, n\_regionkey, n\_comment,

country\_code, start\_time, end\_time, current\_flag, 'I' as dml\_type

from (select n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code,

update\_timestamp as start\_time,

lag(update\_timestamp) over (partition by n\_nationkey order by update\_timestamp desc) as end\_time\_raw,

case when end\_time\_raw is null then '9999-12-31'::timestamp\_ntz else end\_time\_raw end as end\_time,

case when end\_time\_raw is null then 1 else 0 end as current\_flag

from (select n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code, update\_timestamp

from nation\_table\_changes

where metadata$action = 'INSERT'

and metadata$isupdate = 'FALSE'))

union

-- This subquery figures out what to do when data is updated in the NATION table

-- An update to the NATION table results in an update AND an insert to the NATION\_HISTORY table

-- The subquery below generates two records, each with a different dml\_type

select n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code, start\_time, end\_time, current\_flag, dml\_type

from (select n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code,

update\_timestamp as start\_time,

lag(update\_timestamp) over (partition by n\_nationkey order by update\_timestamp desc) as end\_time\_raw,

case when end\_time\_raw is null then '9999-12-31'::timestamp\_ntz else end\_time\_raw end as end\_time,

case when end\_time\_raw is null then 1 else 0 end as current\_flag,

dml\_type

from (-- Identify data to insert into nation\_history table

select n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code, update\_timestamp, 'I' as dml\_type

from nation\_table\_changes

where metadata$action = 'INSERT'

and metadata$isupdate = 'TRUE'

union

-- Identify data in NATION\_HISTORY table that needs to be updated

select n\_nationkey, null, null, null, null, start\_time, 'U' as dml\_type

from nation\_history

where n\_nationkey in (select distinct n\_nationkey

from nation\_table\_changes

where metadata$action = 'INSERT'

and metadata$isupdate = 'TRUE')

and current\_flag = 1))

union

-- This subquery figures out what to do when data is deleted from the NATION table

-- A deletion from the NATION table results in an update to the NATION\_HISTORY table

select nms.n\_nationkey, null, null, null, null, nh.start\_time, current\_timestamp()::timestamp\_ntz, null, 'D'

from nation\_history nh

inner join nation\_table\_changes nms

on nh.n\_nationkey = nms.n\_nationkey

where nms.metadata$action = 'DELETE'

and nms.metadata$isupdate = 'FALSE'

and nh.current\_flag = 1;

--merge to maintain SCD

merge into nation\_history nh -- Target table to merge changes from NATION into

using nation\_change\_data m -- nation\_change\_data is a view that holds the logic that determines what to insert/update into the NATION\_HISTORY table.

on nh.n\_nationkey = m.n\_nationkey -- n\_nationkey and start\_time determine whether there is a unique record in the NATION\_HISTORY table

and nh.start\_time = m.start\_time

when matched and m.dml\_type = 'U' then update -- Indicates the record has been updated and is no longer current and the end\_time needs to be stamped

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when matched and m.dml\_type = 'D' then update -- Deletes are essentially logical deletes. The record is stamped and no newer version is inserted

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when not matched and m.dml\_type = 'I' then insert -- Inserting a new n\_nationkey and updating an existing one both result in an insert

(n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code, start\_time, end\_time, current\_flag)

values (m.n\_nationkey, m.n\_name, m.n\_regionkey, m.n\_comment, m.country\_code, m.start\_time, m.end\_time, m.current\_flag);

-- insert new records

set update\_timestamp = current\_timestamp()::timestamp\_ntz;

begin;

insert into nation values(0,'ALGERIA',0,' haggle. carefully final deposits detect slyly agai','DZ',$update\_timestamp);

insert into nation values(1,'ARGENTINA',1,'al foxes promise slyly according to the regular accounts. bold requests alon','AR',$update\_timestamp);

insert into nation values(2,'BRAZIL',1,'y alongside of the pending deposits. carefully special packages are about the ironic forges. slyly special ','BR',$update\_timestamp);

insert into nation values(3,'CANADA',1,'eas hang ironic silent packages. slyly regular packages are furiously over the tithes. fluffily bold','CA',$update\_timestamp);

insert into nation values(4,'EGYPT',4,'y above the carefully unusual theodolites. final dugouts are quickly across the furiously regular d','EG',$update\_timestamp);

insert into nation values(5,'ETHIOPIA',0,'ven packages wake quickly. regu','ET',$update\_timestamp);

insert into nation values(6,'FRANCE',3,'refully final requests. regular ironi','FR',$update\_timestamp);

insert into nation values(7,'GERMANY',3,'l platelets. regular accounts x-ray: unusual regular acco','DE',$update\_timestamp);

insert into nation values(8,'INDIA',2,'ss excuses cajole slyly across the packages. deposits print aroun','IN',$update\_timestamp);

insert into nation values(9,'INDONESIA',2,' slyly express asymptotes. regular deposits haggle slyly. carefully ironic hockey players sleep blithely. carefull','ID',$update\_timestamp);

insert into nation values(10,'IRAN',4,'efully alongside of the slyly final dependencies. ','IR',$update\_timestamp);

insert into nation values(11,'IRAQ',4,'nic deposits boost atop the quickly final requests? quickly regula','IQ',$update\_timestamp);

insert into nation values(12,'JAPAN',2,'ously. final express gifts cajole a','JP',$update\_timestamp);

insert into nation values(13,'JORDAN',4,'ic deposits are blithely about the carefully regular pa','JO',$update\_timestamp);

insert into nation values(14,'KENYA',0,' pending excuses haggle furiously deposits. pending express pinto beans wake fluffily past t','KE',$update\_timestamp);

insert into nation values(15,'MOROCCO',0,'rns. blithely bold courts among the closely regular packages use furiously bold platelets?','MA',$update\_timestamp);

insert into nation values(16,'MOZAMBIQUE',0,'s. ironic unusual asymptotes wake blithely r','MZ',$update\_timestamp);

insert into nation values(17,'PERU',1,'platelets. blithely pending dependencies use fluffily across the even pinto beans. carefully silent accoun','PE',$update\_timestamp);

insert into nation values(18,'CHINA',2,'c dependencies. furiously express notornis sleep slyly regular accounts. ideas sleep. depos','CN',$update\_timestamp);

insert into nation values(19,'ROMANIA',3,'ular asymptotes are about the furious multipliers. express dependencies nag above the ironically ironic account','RO',$update\_timestamp);

insert into nation values(20,'SAUDI ARABIA',4,'ts. silent requests haggle. closely express packages sleep across the blithely','SA',$update\_timestamp);

insert into nation values(21,'VIETNAM',2,'hely enticingly express accounts. even final ','VN',$update\_timestamp);

insert into nation values(22,'RUSSIA',3,' requests against the platelets use never according to the quickly regular pint','RU',$update\_timestamp);

insert into nation values(23,'UNITED KINGDOM',3,'eans boost carefully special requests. accounts are. carefull','GB',$update\_timestamp);

insert into nation values(24,'UNITED STATES',1,'y final packages. slow foxes cajole quickly. quickly silent platelets breach ironic accounts. unusual pinto be','US',$update\_timestamp);

commit;

-- check the tables after merge

select \* from nation;

select \* from nation\_table\_changes;

-- MERGE statement that uses the CHANGE\_DATA view to load data into the NATION\_HISTORY table

merge into nation\_history nh -- Target table to merge changes from NATION into

using nation\_change\_data m -- NATION\_CHANGE\_DATA is a view that holds the logic that determines what to insert/update into the NATION\_HISTORY table.

on nh.n\_nationkey = m.n\_nationkey -- n\_nationkey and start\_time determine whether there is a unique record in the NATION\_HISTORY table

and nh.start\_time = m.start\_time

when matched and m.dml\_type = 'U' then update -- Indicates the record has been updated and is no longer current and the end\_time needs to be stamped

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when matched and m.dml\_type = 'D' then update -- Deletes are essentially logical deletes. The record is stamped and no newer version is inserted

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when not matched and m.dml\_type = 'I' then insert -- Inserting a new n\_nationkey and updating an existing one both result in an insert

(n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code, start\_time, end\_time, current\_flag)

values (m.n\_nationkey, m.n\_name, m.n\_regionkey, m.n\_comment,m.country\_code, m.start\_time, m.end\_time, m.current\_flag);

--query

select \* from nation\_history;

select \* from nation\_table\_changes;

begin;

update nation

set n\_comment = 'New comment for Arg', update\_timestamp = current\_timestamp()::timestamp\_ntz

where n\_nationkey = 1;

update nation

set n\_comment = 'New comment for Canada', update\_timestamp = current\_timestamp()::timestamp\_ntz

where n\_nationkey = 3;

commit;

select \* from nation where n\_nationkey in (1, 2,3);

select \* from nation\_table\_changes;

-- MERGE statement that uses the CHANGE\_DATA view to load data into the NATION\_HISTORY table

merge into nation\_history nh -- Target table to merge changes from NATION into

using nation\_change\_data m -- CHANGE\_DATA is a view that holds the logic that determines what to insert/update into the NATION\_HISTORY table.

on nh.n\_nationkey = m.n\_nationkey -- n\_nationkey and start\_time determine whether there is a unique record in the NATION\_HISTORY table

and nh.start\_time = m.start\_time

when matched and m.dml\_type = 'U' then update -- Indicates the record has been updated and is no longer current and the end\_time needs to be stamped

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when matched and m.dml\_type = 'D' then update -- Deletes are essentially logical deletes. The record is stamped and no newer version is inserted

set nh.end\_time = m.end\_time,

nh.current\_flag = 0

when not matched and m.dml\_type = 'I' then insert -- Inserting a new n\_nationkey and updating an existing one both result in an Insert

(n\_nationkey, n\_name, n\_regionkey, n\_comment, country\_code, start\_time, end\_time, current\_flag)

values (m.n\_nationkey, m.n\_name, m.n\_regionkey, m.n\_comment,

m.country\_code, m.start\_time, m.end\_time, m.current\_flag);

select \* from nation\_history;

select \* from nation\_history where n\_nationkey in (2,3) order by n\_nationkey, start\_time;

select \* from nation\_table\_changes;

--Set up TASKADMIN role

use role securityadmin;

create role taskadmin;

-- Set the active role to ACCOUNTADMIN before granting the EXECUTE TASK privilege to TASKADMIN

use role accountadmin;

grant execute task on account to roleme, m.end\_time, m.current\_flag);

show tasks;

alter task populate\_nation\_history resume;

show tasks;

select timestampdiff(second, current\_timestamp, scheduled\_time) as next\_run, scheduled\_time, current\_timestamp, name, state

from table(information\_schema.task\_history()) where state = 'SCHEDULED' order by completed\_time desc;

select timestampdiff(second, current\_timestamp, scheduled\_time) as next\_run, scheduled\_time, current\_timestamp, name, state

from table(information\_schema.task\_history()) order by completed\_time desc;

select \* from table(information\_schema.task\_history()) order by completed\_time desc;

-- Delete data

delete from nation where n\_nationkey in (3,7);

select \* from nation\_table\_changes;

select timestampdiff(second, current\_timestamp, scheduled\_time) as next\_run, scheduled\_time, current\_timestamp, name, state

from table(information\_schema.task\_history()) order by completed\_time desc;

select \* from nation\_history where n\_nationkey in (3,7) order by n\_nationkey, start\_time;

-- Insert, update, delete in one pass

begin;

insert into nation values(26, 'COLOMBIA', 1, 'New country', 'CO', current\_timestamp()::timestamp\_ntz);

update nation

set n\_comment = 'New comment for Indonesia', update\_timestamp =

current\_timestamp()::timestamp\_ntz

where n\_nationkey = 9;

delete from nation

where n\_nationkey in (20);

commit;

select \* from nation\_table\_changes;

select \* from nation where n\_nationkey in (26,9,20);

select \* from nation\_history;

select \* from nation\_history where n\_nationkey in (26,9,20);

alter task populate\_nation\_history suspend;

show tasks;