```
-----------------------------------------------------------
-- Author: Logan Guerry (DAS42)
-- Date: 8/30/2021
-- Description: Example case for setting up dynamic
--      role-level security using Row Access Policies.
-- Prereq: SysAd and SecurityAd roles. Sample dataset.
--      Sales DBA, executive, and manager roles created.
-----------------------------------------------------------

-- set compute context
use warehouse das42dev;

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

-- use Security Admin to setup security objects and create/assign new
roles
use role securityadmin;


-- create needed roles:
-- sales executive with access to all records
create role sales_executive_role;
-- region-specific sales managers with access only to their regions
create role us_sales_manager;
create role japan_sales_manager;


-- kindly grant yourself all of the previously created roles
grant role sales_executive_role to user LOGAN;
grant role us_sales_manager to user LOGAN;
grant role japan_sales_manager to user LOGAN;


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

-- use System Admin to setup demo objects
use role sysadmin;


-- create sample DB and demo data tables
create database sf_user_mu_demo;
use database sf_user_mu_demo;
create or replace schema ecom clone sf_training.tpch_sf10; -- clone or
create a sample schema of your choice


-- grant usage and select on DEMO.SCHEMA for all analyst/manager roles
-- usage on DB
grant usage on database sf_user_mu_demo to role sales_executive_role;
grant usage on database sf_user_mu_demo to role us_sales_manager;
```

```
grant usage on database sf_user_mu_demo to role japan_sales_manager;

-- usage on SCHEMA
grant usage on schema sf_user_mu_demo.ecom to role
sales_executive_role;
grant usage on schema sf_user_mu_demo.ecom to role us_sales_manager;
grant usage on schema sf_user_mu_demo.ecom to role
japan_sales_manager;

-- select on all tables in SCHEMA
grant select on all tables in schema sf_user_mu_demo.ecom to role
sales_executive_role;
grant select on all tables in schema sf_user_mu_demo.ecom to role
us_sales_manager;
grant select on all tables in schema sf_user_mu_demo.ecom to role
japan_sales_manager;


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


-- create and use security schema within new sample DB
create schema sf_user_mu_demo.security;


-- create mapping table
create or replace table security.sales_manager_nation (
  sales_manager varchar,
  nation_key number
)
;


-- populate mapping table
insert into security.sales_manager_nation
    values
        ('us_sales_manager', 24),
        ('japan_sales_manager', 12)
;


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

-- use Security Admin to setup security schema and mapping table
use role securityadmin;

-- create mapping role
create role mapping_role;

-- kindly grant yourself the mapping role so that you can query the
```

```
mapping table further down the road
grant role mapping_role to user LOGAN;

-- grant ability to query mapping table to mapping role
grant select on table sf_user_mu_demo.security.sales_manager_nation to
role mapping_role;

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

-- use schema owner role to setup Row Access Policy
use role sysadmin;

-- create RLS policy
create row access policy
sf_user_mu_demo.security.sales_nation_policy_demo as (sales_nation_key
number) returns boolean ->
  'SALES_EXECUTIVE_ROLE' = current_role()
  or exists (
        select 1 from sales_manager_nation
          where upper(sales_manager) = current_role() -- be mindful of
case sensitivity
            and nation_key = sales_nation_key
      )
;

-- assign row access policy to ecom table(s)
alter table sf_user_mu_demo.ecom.customer add row access policy
sf_user_mu_demo.security.sales_nation_policy_demo on (c_nationkey);
alter table sf_user_mu_demo.ecom.supplier add row access policy
sf_user_mu_demo.security.sales_nation_policy_demo on (s_nationkey);

-- transfer ownership of Row Access Policy to mapping_role
-- WARNING: once you transfer ownership, the policy must be updated
via the mepping_role which will also require usage grants on the DB in
question
grant ownership on row access policy
sf_user_mu_demo.security.sales_nation_policy_demo to mapping_role;


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

-- TEST YOUR WORK
use role japan_sales_manager;
use role us_sales_manager;
use role sales_executive_role;

select
    *
from
    sf_user_mu_demo.ecom.customer
```

```
limit 10;
```