```
//ADDITION 1 - roll up new roles to SYSADMIN
use role useradmin;
grant role itc_admin to role sysadmin;
grant role marketing to role sysadmin;
grant role it to role sysadmin;
grant role infosec to role sysadmin;
grant role executive to role sysadmin;

//ADDITION 2 - users don't need to change the
password
use role useradmin;
create user "roy@itcrowd"
    default_warehouse=demo_wh default_role=it
password='PickleRick!!' must_change_password = false;
create user "moss@itcrowd"
    default_warehouse=demo_wh default_role=infosec
password='PickleRick!!' must_change_password = false;
create user "jen@itcrowd"
    default_warehouse=demo_wh default_role=it
password='PickleRick!!' must_change_password = false;
create user "denholm@itcrowd"
    default_warehouse=demo_wh default_role=executive
password='PickleRick!!' must_change_password = false;
create user "douglas@itcrowd"
    default_warehouse=demo_wh default_role=marketing
password='PickleRick!!' must_change_password = false;
create user "richmond@itcrowd"
    default_warehouse=demo_wh default_role=itc_admin
password='PickleRick!!' must_change_password = false;

//ADDITION 3 - disabled forced MFA
use role sysadmin;
create or replace database security_db;
create or replace schema
security_db.authentication_policies;
create or replace authentication policy
security_db.authentication_policies.mfa_optional
MFA_ENROLLMENT = optional;
grant usage on database security_db to role
securityadmin;
grant usage on schema
security_db.authentication_policies to role
securityadmin;
```

```
grant apply on authentication policy
security_db.authentication_policies.mfa_optional to
role securityadmin;

use role securityadmin;
alter user "roy@itcrowd" set authentication policy
security_db.authentication_policies.mfa_optional;
alter user "moss@itcrowd" set authentication policy
security_db.authentication_policies.mfa_optional;
alter user "jen@itcrowd" set authentication policy
security_db.authentication_policies.mfa_optional;
alter user "denholm@itcrowd" set authentication
policy
security_db.authentication_policies.mfa_optional;
alter user "douglas@itcrowd" set authentication
policy
security_db.authentication_policies.mfa_optional;
alter user "richmond@itcrowd" set authentication
policy
security_db.authentication_policies.mfa_optional;

/*
alter user "roy@itcrowd" unset authentication policy;
alter user "moss@itcrowd" unset authentication
policy;
alter user "jen@itcrowd" unset authentication policy;
alter user "denholm@itcrowd" unset authentication
policy;
alter user "douglas@itcrowd" unset authentication
policy;
alter user "richmond@itcrowd" unset authentication
policy;
*/

//ADDITION/FIX 4 - grant access to snowflake sample
data for itcadmin role
use role securityadmin;
grant imported privileges on database
snowflake_sample_data to role itcadmin;

//FIX 5 -- Make sure to comment this so the line
below it runs on Page 5
```

```
-- >>>>>>>>> grant ownership on table
REYNHOLM_IND_DATA.BASEMENT.ROW_ACCESS_MAPPING to role
infosec;

//FIX 6 -- extract semantic categories has changed
output
select
    f.key,

f.value:"recommendation":"privacy_category"::varchar
as privacy_category,

f.value:"recommendation":"semantic_category"::varchar
as semantic_category,
    f.value:"recommendation":"confidence"::varchar as
probability
from
    TABLE(

FLATTEN(EXTRACT_SEMANTIC_CATEGORIES('REYNHOLM_IND_DAT
A.BASEMENT.CUSTOMERS')::VARIANT)
    ) AS f
where f.key='CD_GENDER';
```