RLS & Masking for EC,CMP,PM

EC:

dim_param_sf_roles – static table maintained by team for rls

In this table FLG_EMP_MAN and SEQ columns are not used anywhere in our current logic.

dim_param_roles_tags – static table maintained by team for masking

This field DPRT_TAG_VALUE_DPRT denotes the tag names such as CMP, KEYPLAYERS …

dim_sf_roles_group – source is api which contains roles and group

The greater and target group field is to mention for example the role id (601) is linked to greater group 2312 and this greater group persons are having access to the group 3589 persons**.**

dim_group_user – source is api which contains group and user

We will come to know how the employees are mapped to group id.

rel_ec_employee_manager – union with above data with manager and employee details

User id SK – The surrogate key belongs to the employee.

Rel User id SK – The surrogate key belongs to the Manager.

dim_employee_profil**e** – to get the employee UPN

DDEP_EMPLOYEE_PROFILE_KEY_DDEP – The surrogate key belongs to the employee.

DDEP_EMPLOYEE_UPN_DDEP - a User Principal Name (UPN) is a username and domain in an email address format. In our project we are getting mail address and for some of them employee id. I hope in future they are changing the employee id to mail address, so we can match the user with snowflake. account.usage table.

DDEP_EMPLOYEE_SRC_UPN_DDEP - This field is having the values with UPN and EMPLOYEE ID. In our logic we are deriving everything with EMPLOYEE_UPN not with SRC_UPN.

snowflake.account_usage.users – to check the user is part of snowflake

Table locations:

select * from HRDP_CORE_QA_DB.CMN_CORE_SCH.DIM_PARAM_SF_ROLES;

select * from HRDP_CORE_QA_DB.CMN_CORE_SCH.DIM_PARAM_ROLES_TAGS;

select * from HRDP_CORE_QA_DB.CMN_CORE_SCH.DIM_SF_ROLES_GROUP;

select * from HRDP_CORE_QA_DB.CMN_CORE_SCH.DIM_GROUP_USER;

select * from HRDP_CORE_QA_DB.CMN_CORE_SCH.REL_EC_EMPLOYEE_MANAGER;

select * from HRDP_CORE_QA_DB.CMN_CORE_SCH.DIM_EMPLOYEE_PROFILE;

select * from snowflake.account_usage.users where LOGIN_NAME in

('CHETAN.JADHAV@LOREAL.COM','EMILY.PAGAN@LOREAL.COM','NAN.MOHABIR@LOREAL.COM');

RLS FLOW

The flow will work in such a way that to know the employee has access to how many employees.

With the first three DIM tables are joined to know about the roles and the corresponding groups and users from the api.

The relation employee manager table is to get the employees and different types of managers relation details which is not available in the role api's.

Masking FLOW

The flow will work in such a way that to know the employee has access to how many employees.

With the first three DIM tables are joined to know about the roles and the corresponding groups and users from the api.

EC_RLS_POL :

```
exists (select 1 from cmn_core_sch.dim_param_snowflake_roles
            where dpsr_snowflake_role_dpsr=current_role() and dpsr_exclude_security_dpsr='Y')
          or exists (select 1 from cmn_core_sch.rel_employee_user
            where reeu_access_employee_upn_ddep = current_user() and reeu_subject_domain_reeu = 'EC'
              and (reeu_employee_id_ddep = 'ALL' or reeu_employee_id_ddep = p_user_id))
```

COL_KEY_PLAYER_TEXT_MASKING :

```
case when SYSTEM$GET_TAG_ON_CURRENT_COLUMN('SK') = 'KEY_PLAYER' then
      case when exists (select 1 from CMN_CORE_SCH.DIM_PARAM_SNOWFLAKE_ROLES
                  where DPSR_SNOWFLAKE_ROLE_DPSR=current_role() and DPSR_EXCLUDE_SECURITY_DPSR='Y')
          then val
        when exists (select 1 from CMN_CORE_SCH.REL_EMPLOYEE_USER_TAG
                  where CURRENT_USER()=REUR_ACCESS_EMPLOYEE_UPN_DDEP
                    and REUR_TAG_VALUE_DPTC= 'KEY_PLAYER'
                    and REUR_EMPLOYEE_ID_DDEP='ALL')
          then val
        when exists (select 1 from CMN_CORE_SCH.REL_EMPLOYEE_USER_TAG
                  where CURRENT_USER()=REUR_ACCESS_EMPLOYEE_UPN_DDEP
                    and REUR_TAG_VALUE_DPTC= 'KEY_PLAYER'
                    and REUR_EMPLOYEE_ID_DDEP=emp_id)
          then val
        else cast('-201' as TEXT)
      end
   else NULL
end
```

Automation Process:

In snowflake, the view HRDP_CORE_NP_DB.PUBLIC.val_ec_hc) is created under HRDP_CORE_NP_DB in the schema public. For the testing users, Just take the user id's and set it like below. Through this you can check the given user status for current month and current year .

```
set vUserID='["00547206","00308209","00840451","00842943","00530051","00421498","00532291","00842943","00266621","00483785"]';
```

Then you will come to know the user statuses in the below columns

SNOWFLAKE_STATUS – You can check whether the given user is belong to snowflake or not in the snowflake.account_usage.user table

TOT_CNT – Total row count of the table which the user has access.

RLS_COUNT – After applying RLS, how many records he can access.

CMP – Same as above for masking on CMP

KEY_PLAYER - Same as above for masking on KEY PLAYER

PMGM - Same as above for masking on PMGM

RLS_BY – The user's RLS roles.

For EC Masking, only KEY PLAYERS tag is applied on the column FAHC_SK_PLAY in FACT_HEADCOUNT and FACT HEADCOUNT_SNAPSHOT table. The above result is based on my access. Please check the columns RLS_BY, CMP_MASKING_BY, KEYPLAYER_MASKING_BY, PMGM_MASKING_BY, PERSONAL_MASKING_BY. If you are have access to any of the roles then you can able to see the data.

CMP – VP:

dim_param_sf_roles – static table maintained by team for rls

dim_param_roles_tags – static table maintained by team for masking

dim_sf_roles_group – source is api which contains roles and group

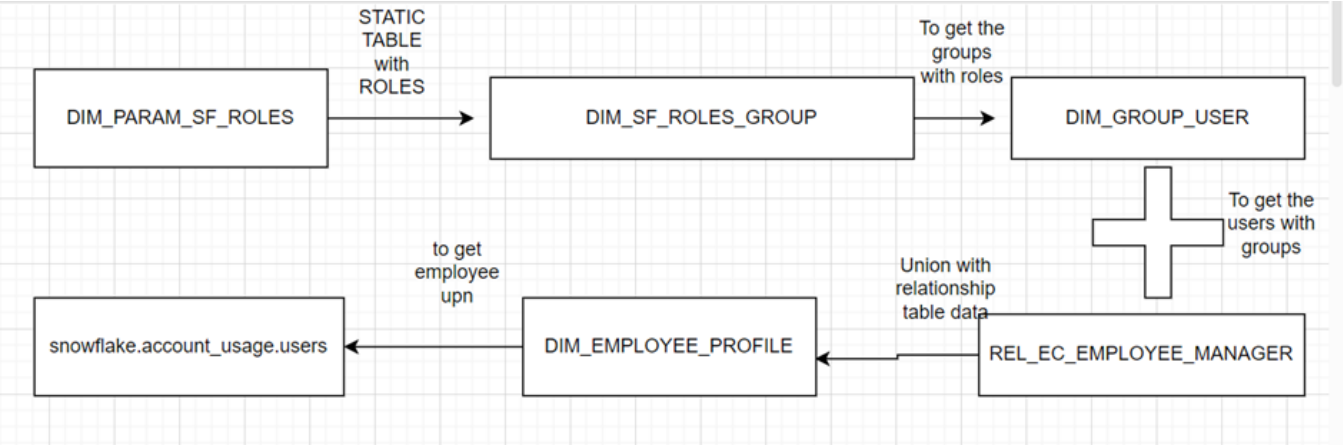
dim_group_user – source is api which contains group and user

rel_ec_employee_manager – union with above data with manager and employee details

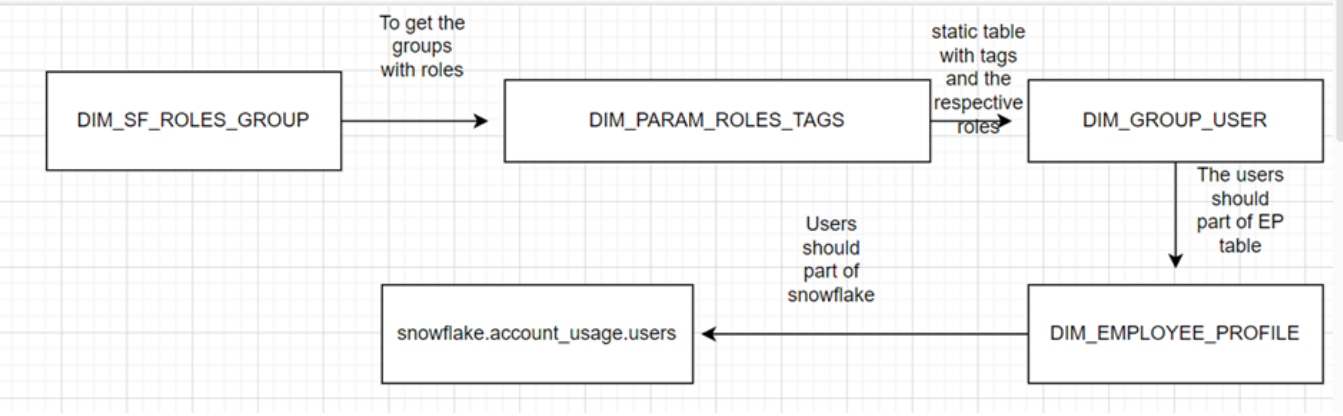
dim employee profile – to get the employee UPN

snowflake.account_usage.users – to check the user is part of snowflake

RLS FLOW



Masking FLOW



RLS Policy

```
exists (select 1 from cmn_core_sch.dim_param_snowflake_roles

                where dpsr_snowflake_role_dpsr=current_role() and dpsr_exclude_security_dpsr='Y')

              or exists (select 1 from cmn_core_sch.rel_employee_user

                where reeu_access_employee_upn_ddep = current_user() and reeu_subject_domain_reeu = 'CMP'

                    and (reeu_employee_id_ddep = 'ALL' or reeu_employee_id_ddep = p_user_id))
```

Automation Process:

In snowflake, the view (val_cmp_vp) is created under HRDP_CORE_NP_DB in the schema public. For the testing users, Just take the user id's and set it like below. Through this you can check the given user status for current month and current year .

```
set
vUserID='[''00547206'',''00308209'',''00840451'',''00842943'',''00530051'',''00421498'',''00532291'',''00842943'
','',''00266621'',''00483785'']';
```

Then you will come to know the user statuses in the below columns

SNOWFLAKE_STATUS – You can check whether the given user is belong to snowflake or not in the snowflake.account_usage.user table

TOT_CNT – Total row count of the table which the user has access.

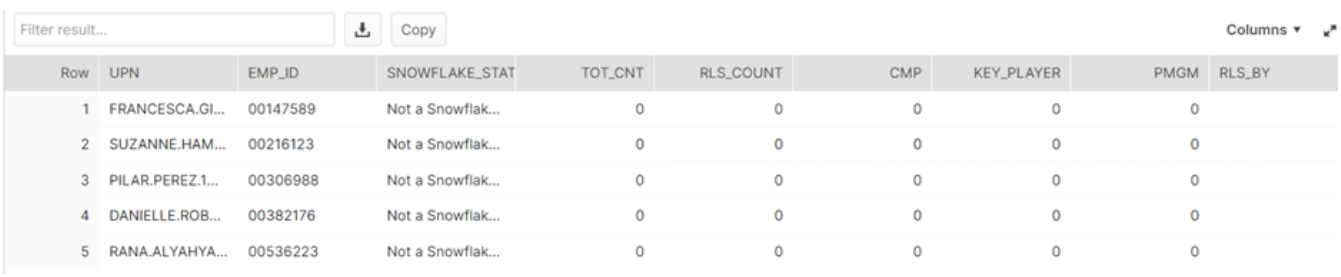
RLS_COUNT – After applying RLS, how many records he can access.

CMP – Same as above for masking on CMP

KEY_PLAYER - Same as above for masking on KEY PLAYER

PMGM - Same as above for masking on PMGM

RLS_BY – The user's RLS roles.



Row	UPN	EMP_ID	SNOWFLAKE_STAT	TOT_CNT	RLS_COUNT	CMP	KEY_PLAYER	PMGM	RLS_BY
1	FRANCESCA.GI...	00147589	Not a Snowflak...	0	0	0	0	0	
2	SUZANNE.HAM...	00216123	Not a Snowflak...	0	0	0	0	0	
3	PILAR.PEREZ.1...	00306988	Not a Snowflak...	0	0	0	0	0	
4	DANIELLE.ROB...	00382176	Not a Snowflak...	0	0	0	0	0	
5	RANA.ALYAHYA...	00536223	Not a Snowflak...	0	0	0	0	0	

For Compensation variable pay Masking, two tags are applied

CMP - FTVP_BUSINESS_RATING_GUIDELINE_DVPG, FTVP_PEOPLE_RATING_GUIDELINE_DVPG, FTVP_BONUSABLE_SALARY_FTVP, FTVP_TARGET_AMOUNT_FTVP, FTVP_BU_PAYOUT_FTVP, FTVP_BG_BUSINESS_TARGET_FTVP, FTVP_BG_BUSINESS_RATING_FTVP, FTVP_BG_TEAM_RESULT_PAYOUT_AMOUNT_FTVP, FTVP_PG_PEOPLE_TARGET_FTVP, FTVP_PG_PEOPLE_RATING_FTVP

PMGM - FTVP_BG_RATING_FTVP, FTVP_PG_RATING_FTVP

Always key players will remain zero.

CMP – YER:

dim_param_sf_roles – static table maintained by team for rls

dim_param_roles_tags – static table maintained by team for masking

dim_sf_roles_group – source is api which contains roles and group

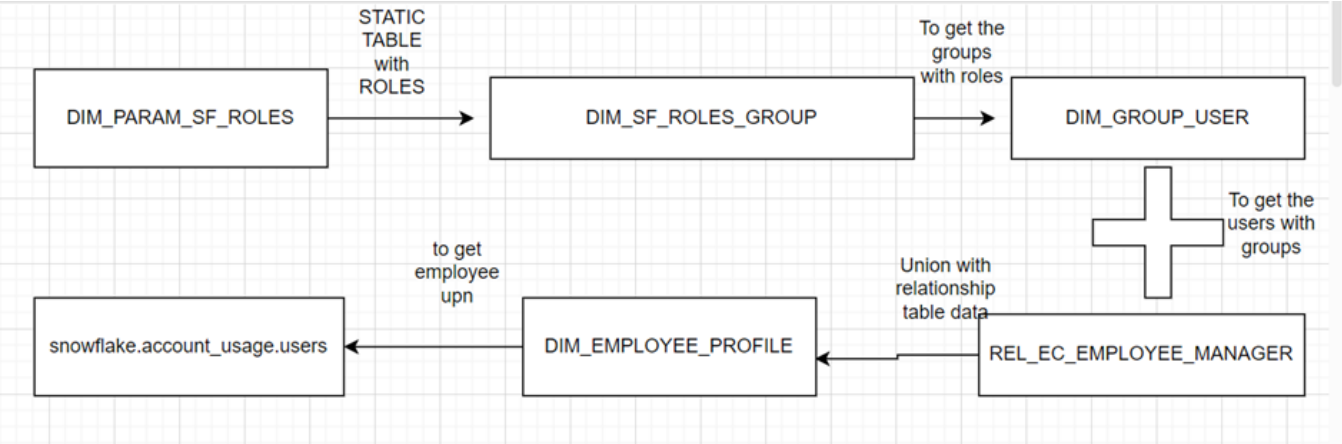
dim_group_user – source is api which contains group and user

rel_ec_employee_manager – union with above data with manager and employee details

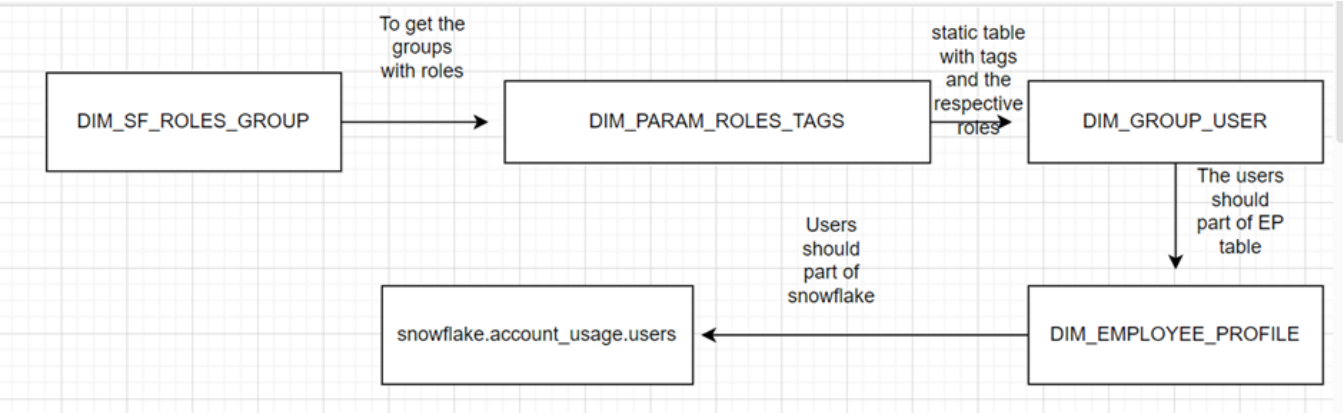
dim employee profile – to get the employee UPN

snowflake.account_usage.users – to check the user is part of snowflake

RLS FLOW



Masking FLOW



RLS Policy

```
exists (select 1 from cmn_core_sch.dim_param_snowflake_roles

                where dpsr_snowflake_role_dpsr=current_role() and dpsr_exclude_security_dpsr='Y')

              or exists (select 1 from cmn_core_sch.rel_employee_user

                where reeu_access_employee_upn_ddep = current_user() and reeu_subject_domain_reeu = 'CMP'

                    and (reeu_employee_id_ddep = 'ALL' or reeu_employee_id_ddep = p_user_id))
```

Automation Process:

In snowflake, the view (val_cmp_yer) is created under HRDP_CORE_NP_DB in the schema public. For the testing users, Just take the user id's and set it like below. Through this you can check the given user status for current month and current year .

```
set
vUserID='[''00547206'',''00308209'',''00840451'',''00842943'',''00530051'',''00421498'',''00532291'',''00842943'
','',''00266621'',''00483785'']';
```

Then you will come to know the user statuses in the below columns

SNOWFLAKE_STATUS – You can check whether the given user is belong to snowflake or not in the snowflake.account_usage.user table

TOT_CNT – Total row count of the table which the user has access.

RLS_COUNT – After applying RLS, how many records he can access.

CMP – Same as above for masking on CMP

KEY_PLAYER - Same as above for masking on KEY PLAYER

PMGM - Same as above for masking on PMGM

RLS_BY – The user's RLS roles.

Row	UPN	EMP_ID	SNOWFLAKE_STAT	TOT_CNT	RLS_COUNT	CMP	KEY_PLAYER	PMGM	RLS_BY
1	FRANCESCA.GI...	00147589	Not a Snowflak...	2327	1	0	0	1	MGR
2	SUZANNE.HAM...	00216123	Not a Snowflak...	2327	0	0	0	0	
3	PILAR.PEREZ.1...	00306988	Not a Snowflak...	2327	0	0	0	0	
4	DANIELLE.ROB...	00382176	Not a Snowflak...	2327	0	0	0	0	
5	RANA.ALYAHYA...	00536223	Not a Snowflak...	2327	0	0	0	0	

For Compensation year end Masking, all three tags are applied for the below columns

CMP –

FCYT_ANNUALIZED_SALARY_INCREASE_MERIT_FCYT
FCYT_BUDGET_AMOUNT_SALARY_INCREASE_FYCT
FCYT_BUDGET_PERCENT_SALARY_INCREASE_FYCT
FCYT_BUDGET_TOTAL_PROPOSED_SPEND_FYCT
FCYT_FK_FG_PROMOTION_DFPR
FCYT_PERCENT_INCREASE_DELTA_FCYT
FCYT_PK_NEW_POSITION_IN_RANGE_TOTAL_CASH_FCYT
FCYT_PK_POSITION_IN_RANGE_AC_DPIR
FCYT_PK_POSITION_IN_RANGE_BC_DPIR
FCYT_PK_POSITION_IN_RANGE_TOTAL_CASH_FCYT
FCYT_UC_CURRENT_BASE_SALARY_FCYT
FCYT_UC_INCREASE_DELTA_FCYT
FCYT_UC_NEW_BASE_SALARY_PROPOSAL_FCYT
FCYT_FK_ALL_PLAYERS_KEY_DALP

PMGM - FYCT_FK_RATINGS_DYER

KEY PLAYERS - FCYT_FK_ALL_PLAYERS_KEY_DALP

> Always key players will remain zero.

EC- PM:

dim_param_sf_roles – static table maintained by team for rls

dim_param_roles_tags – static table maintained by team for masking

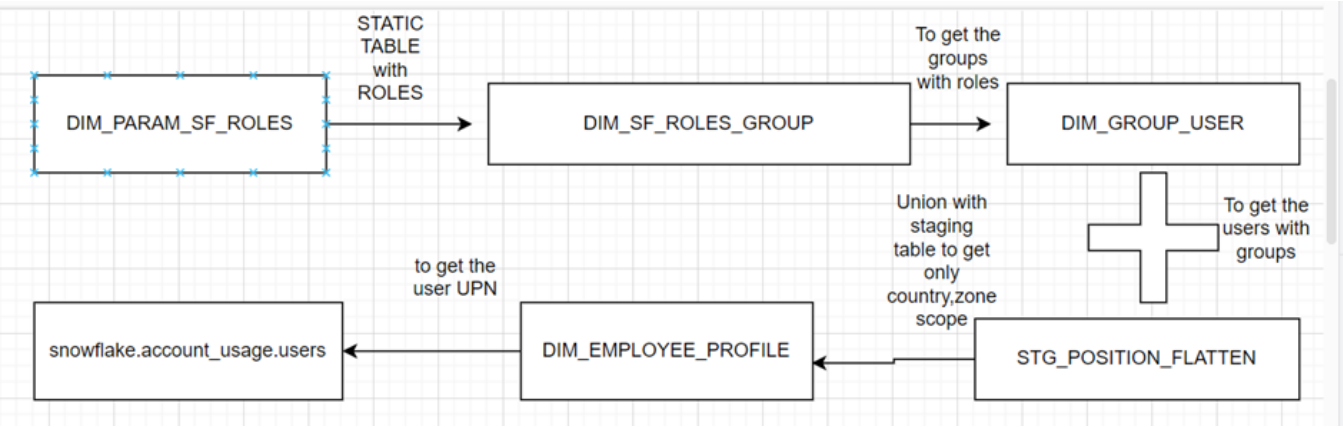
dim_sf_roles_group – source is api which contains roles and group

dim_group_user – source is api which contains group and user

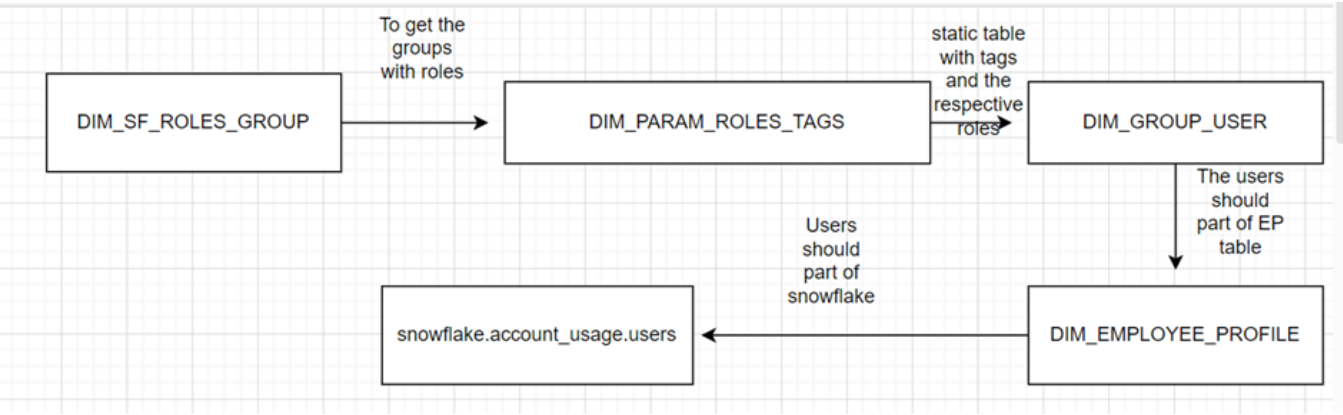
dim employee profile – to get the employee UPN

snowflake.account_usage.users – to check the user is part of snowflake

RLS FLOW



Masking FLOW



RLS Policy

```
(
    exists (select 1 from cmn_core_sch.dim_param_snowflake_roles
                where dpsr_snowflake_role_dpsr=current_role() and dpsr_exclude_security_dpsr='Y')
    or
    exists (select 1 from cmn_core_sch.dim_position_access
                where access_employee_upn = current_user()
                    and scope_type='ST000004')
    or exists (select 1 from cmn_core_sch.dim_position_access
                where access_employee_upn = current_user()
                    and scope_type='ST000002' and scope_access=v_zone)
    or exists (select 1 from cmn_core_sch.dim_position_access
                where access_employee_upn = current_user()
                    and scope_type='ST000016' and scope_access=v_country)
    )
```

Automation Process:

In snowflake, the view (val_ec_pm) is created under HRDP_CORE_NP_DB in the schema public. For the testing users, Just take the user id's and set it like below. Through this you can check the given user status for current month and current year .

```
set
vUserID='[''00547206'',''00308209'',''00840451'',''00842943'',''00530051'',''00421498'',''00532291'',''00842943'
','',''00266621'',''00483785'']';
```

Then you will come to know the user statuses in the below columns

SNOWFLAKE_STATUS – You can check whether the given user is belong to snowflake or not in the snowflake.account_usage.user table

TOT_CNT – Total row count of the table which the user has access.

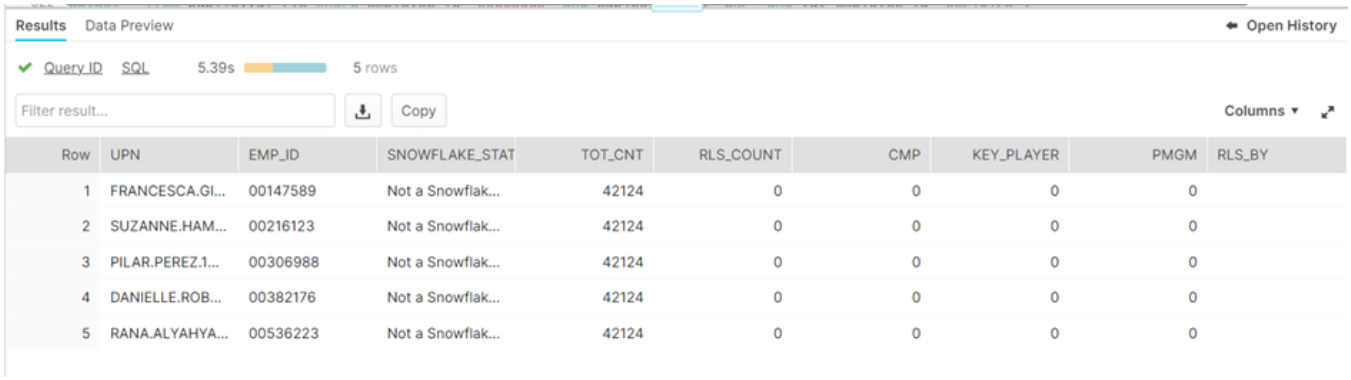
RLS_COUNT – After applying RLS, how many records he can access.

CMP – Same as above for masking on CMP

KEY_PLAYER - Same as above for masking on KEY PLAYER

PMGM - Same as above for masking on PMGM

RLS_BY – The user's RLS roles.



Results Data Preview — Open History

Query ID SQL 5.39s 5 rows

Filter result... Copy Columns

Row	UPN	EMP_ID	SNOWFLAKE_STAT	TOT_CNT	RLS_COUNT	CMP	KEY_PLAYER	PMGM	RLS_BY
1	FRANCESCA.GI...	00147589	Not a Snowflak...	42124	0	0	0	0	
2	SUZANNE.HAM...	00216123	Not a Snowflak...	42124	0	0	0	0	
3	PILAR.PEREZ.1...	00306988	Not a Snowflak...	42124	0	0	0	0	
4	DANIELLE.ROB...	00382176	Not a Snowflak...	42124	0	0	0	0	
5	RANA.ALYAHYA...	00536223	Not a Snowflak...	42124	0	0	0	0	

For position management Masking, two tags are applied for the below columns

CMP –

FAPM_PAY_RANGE_POSN
FAPM_SALARY_MIN_POSN
FAPM_SALARY_MID_POSN
FAPM_SALARY_MAX_POSN

KEY PLAYERS - FAPM_KEY_POSITION_POSN, FAPM_KEY_POSITION_LEVEL_POSN