Role Hierarchy Ownership - Sysadmin

When Azure AD creates roles in Snowflake, it will not tie those roles to other roles, which could create "silo" roles that could become not managable by the ADMINs.

Until Azure AD doesn't allow nested groups to be created, our only option is to create a script that will tie the roles created through Azure AD on a reoccurring basis.

The Stored Procedure uses show and grant commands, so it will have no to little impact on credit consumption.

The SP can be created by the ADMIN roles to manage, but will need to be ran with SECURITYADMIN as it will have manage grants to run show commands and grants to SYSADMIN.

Stored Procedure for Sysadmin grant

```
use role securityadmin;

create or replace procedure SYSADMIN_GRANTS()
returns string
language javascript
comment = 'SYSADMIN grants to Azure AD roles'
execute as caller
as
$$
   var get_role_list = `show roles;`;
   snowflake.execute({sqlText: get_role_list });
   var temp_table_roles =`select 'GRANT ROLE "' || "name" || '" TO ROLE SYSADMIN;' from table(result_scan
(LAST_QUERY_ID())) where "owner"='AAD_PROVISIONER';`;
   var result_set = snowflake.execute({sqlText: temp_table_roles });
   while (result_set.next())
   {
       var grant_stmnt = snowflake.createStatement({sqlText: result_set.getColumnValue(1)}).execute();
       snowflake.execute({sqlText: grant_stmnt });
   }
   return 'SUCCESS'
$$
;


call SYSADMIN_GRANTS();
```

This can then be executed by a task on a reoccurring basis.
For example:

Task For SP

```
CREATE TASK SYSADMIN_GRANTS_TASK
  WAREHOUSE = HRDP_ADM_WH
  SCHEDULE = 'USING CRON 0 0 1 * * UTC'
AS
SYSADMIN_GRANTS();
```