**RBAC Automation**

**Application Workflow**

**By Ihor Karbovskyy**

Reference: [Design Doc](https://docs.google.com/document/d/1kLZHa4-PyeUzgd45u-2pTSiUc6F2uTEWrkzY4EAvVhg/edit#)

**VERSIONING**

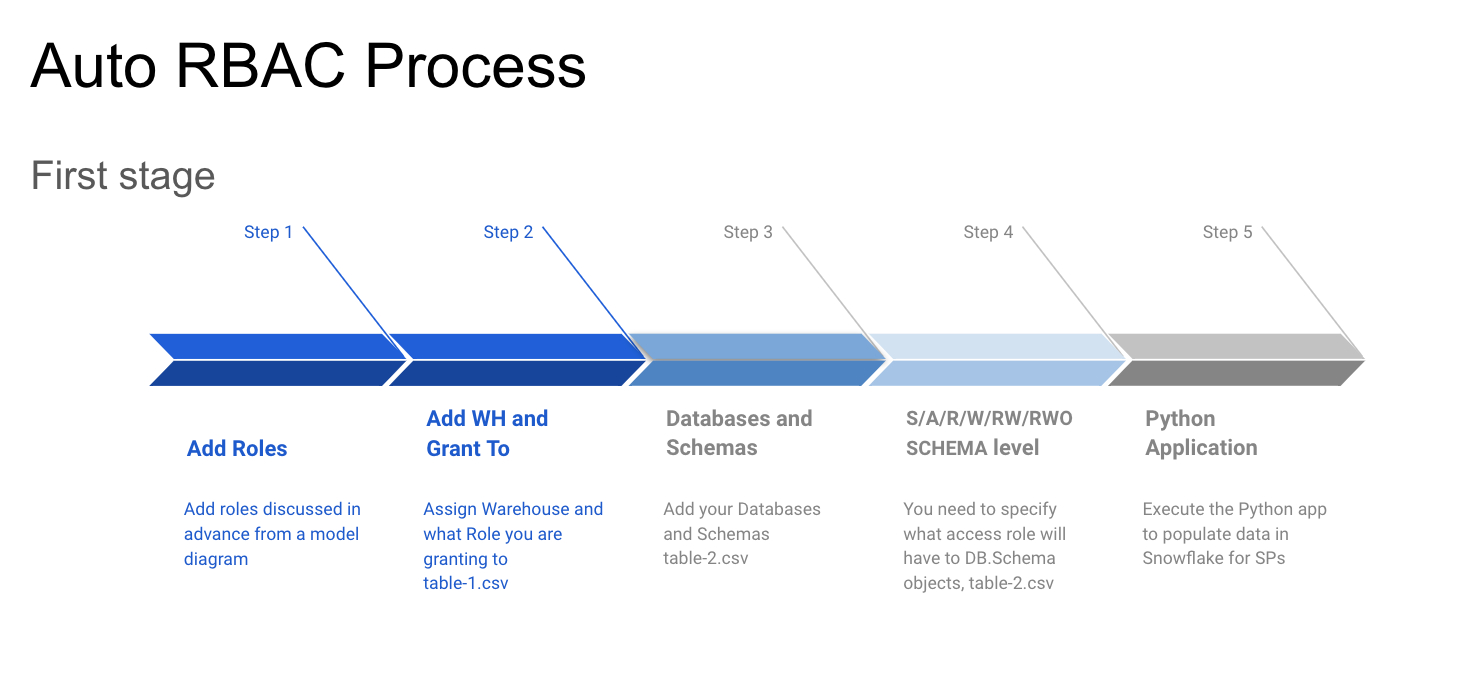
|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Notes** | **Version** |
| 5/20/2019 | Ihor Karbovskyy | Prototype | v0.1 |
| 7/2/2019 | Ihor Karbovskyy | Added schema | v0.2 |
| 7/25/2019 | Ihor Karbovskyy | Added SCH\_MNGR | v0.3 |
| 8/2/2019 | Ihor Karbovskyy | Test on Blue KC | v0.4 |
| 8/10/2019 | Ihor Karbovskyy | Pivot Modification | v0.5 |

**OVERVIEW**

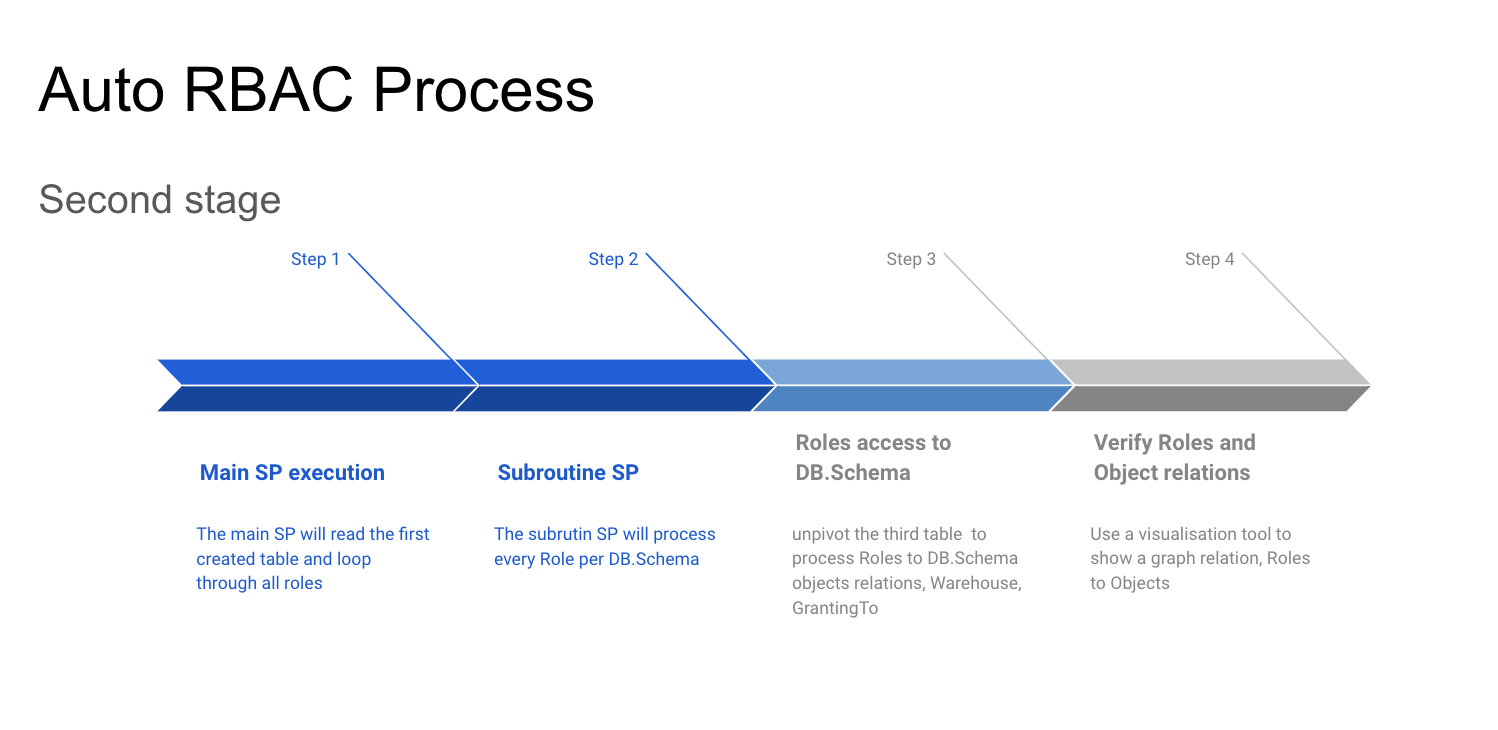
The application will create or execute an SQL script in a Snowflake that will create ROLES, GRANTS what will be referenced to Users structured in the Groups (Access, Functional roles) or on MS ADFS Groups. Roles may have hierarchy, access level to the data objects based on preset RBAC access matrix (RAM). RAM need to be discussed in details, designed and approved by customer.

**DETAILS**

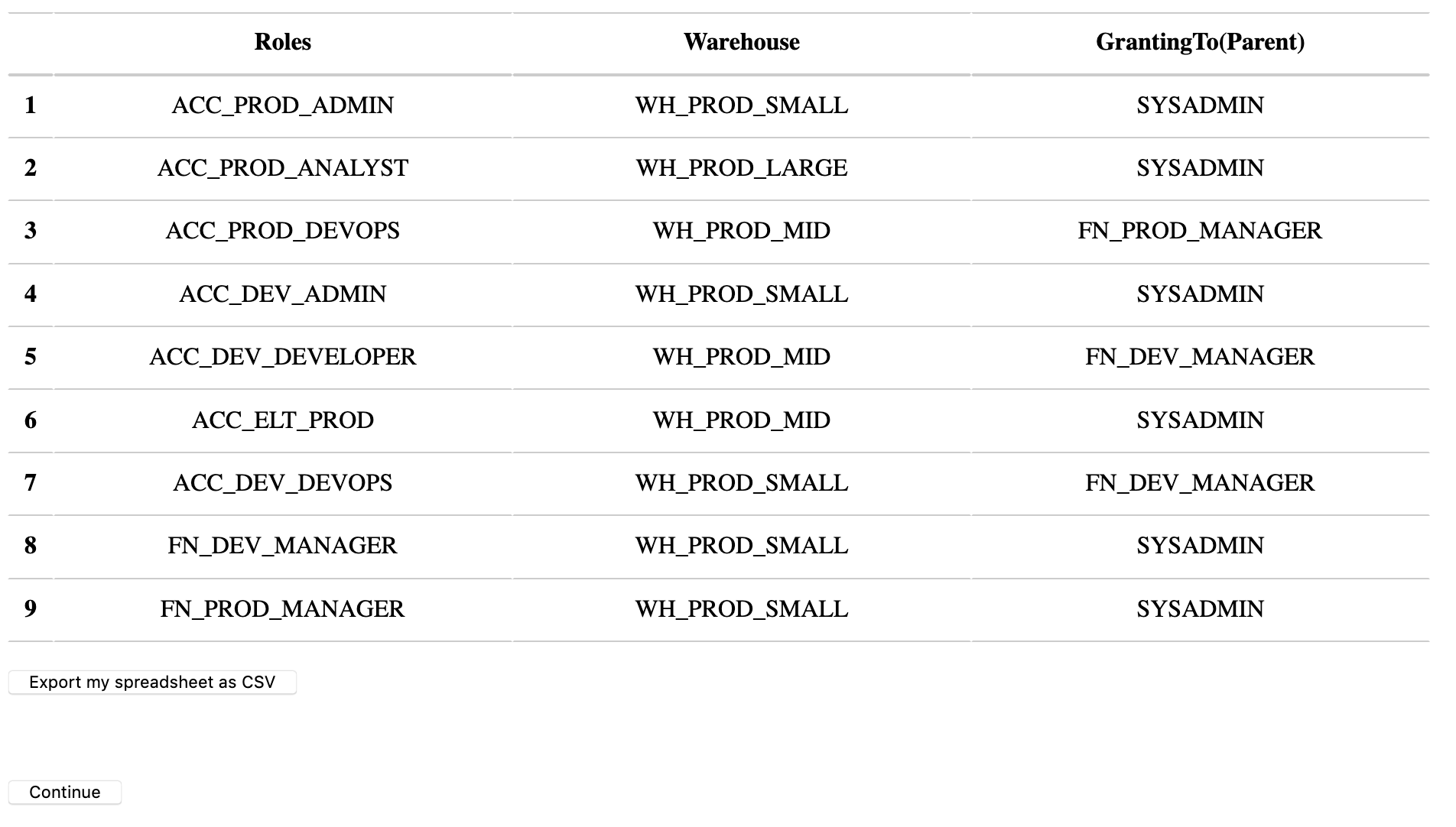
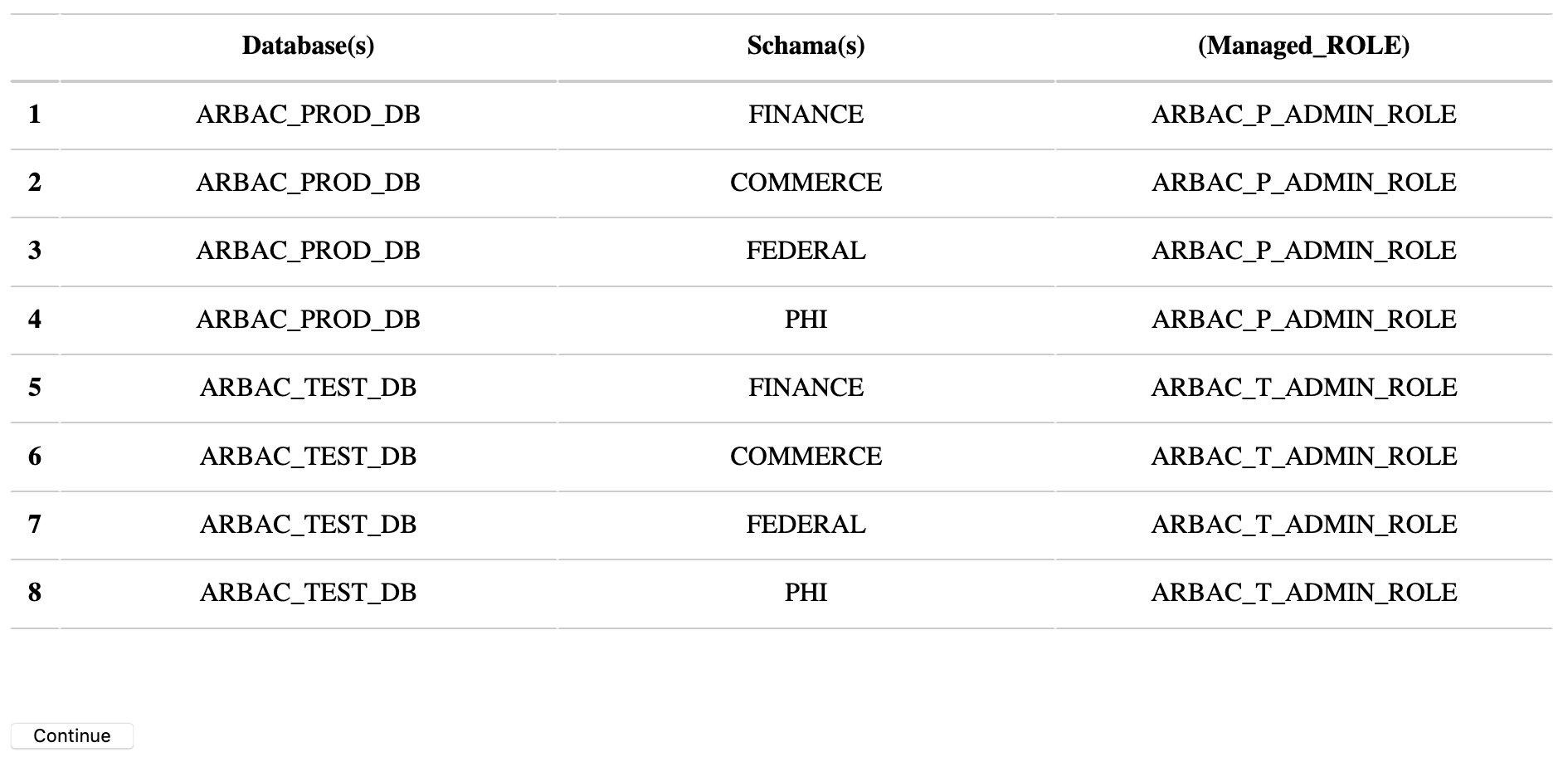
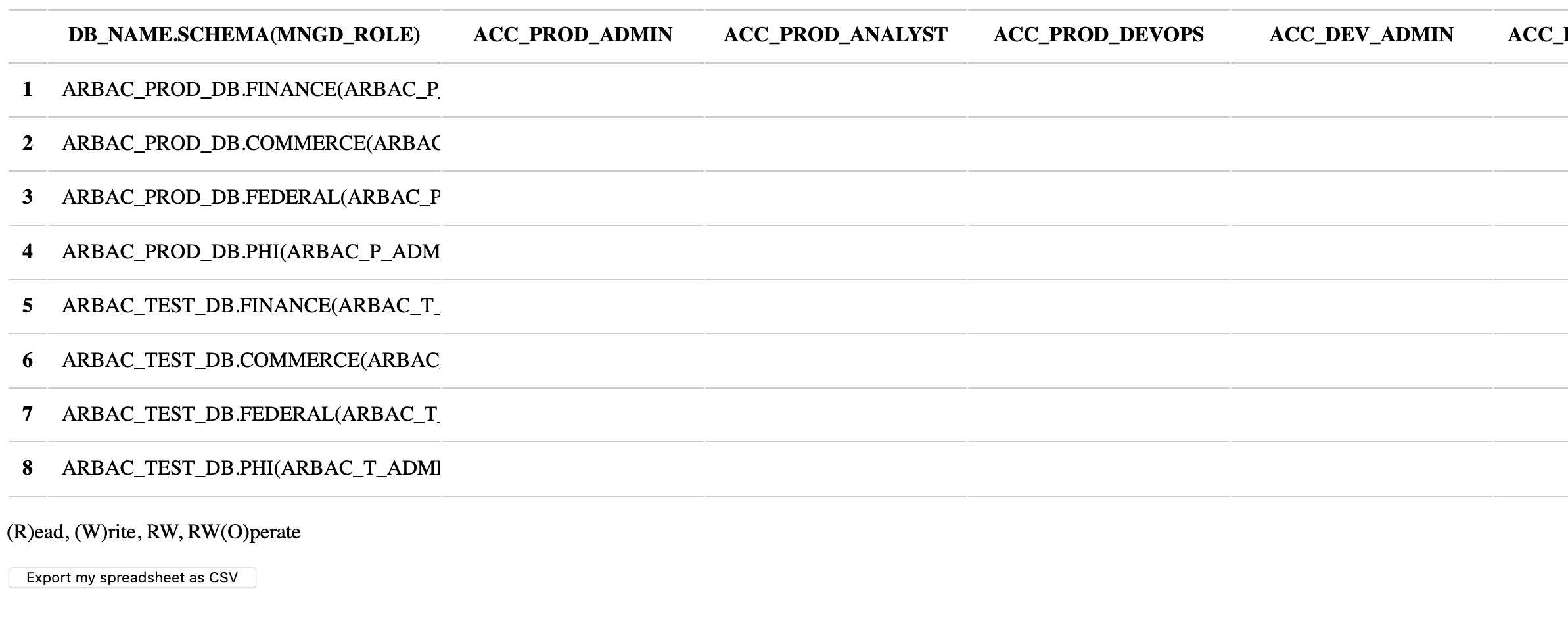
**First Stage**



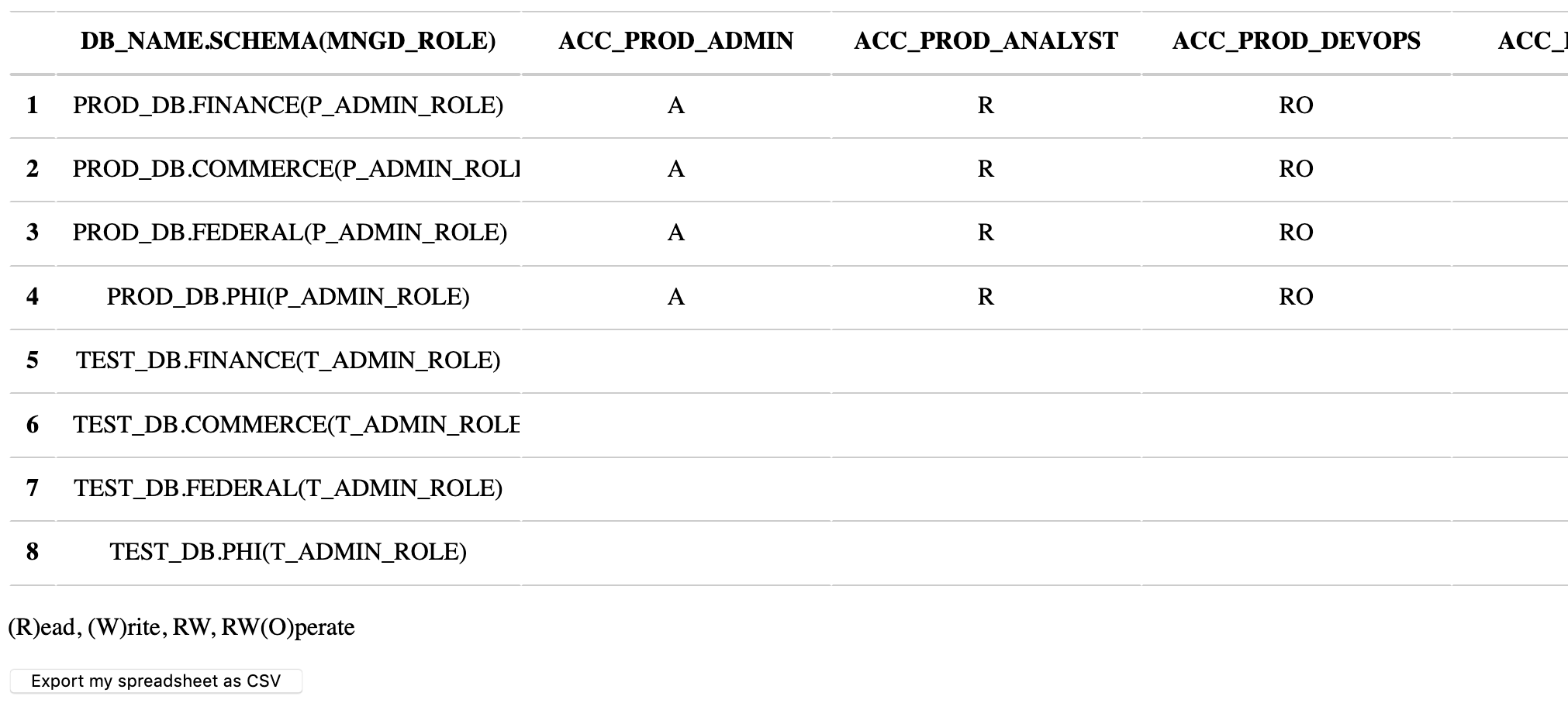
**Second Stage**



# 1. Insert initial data to create Roles

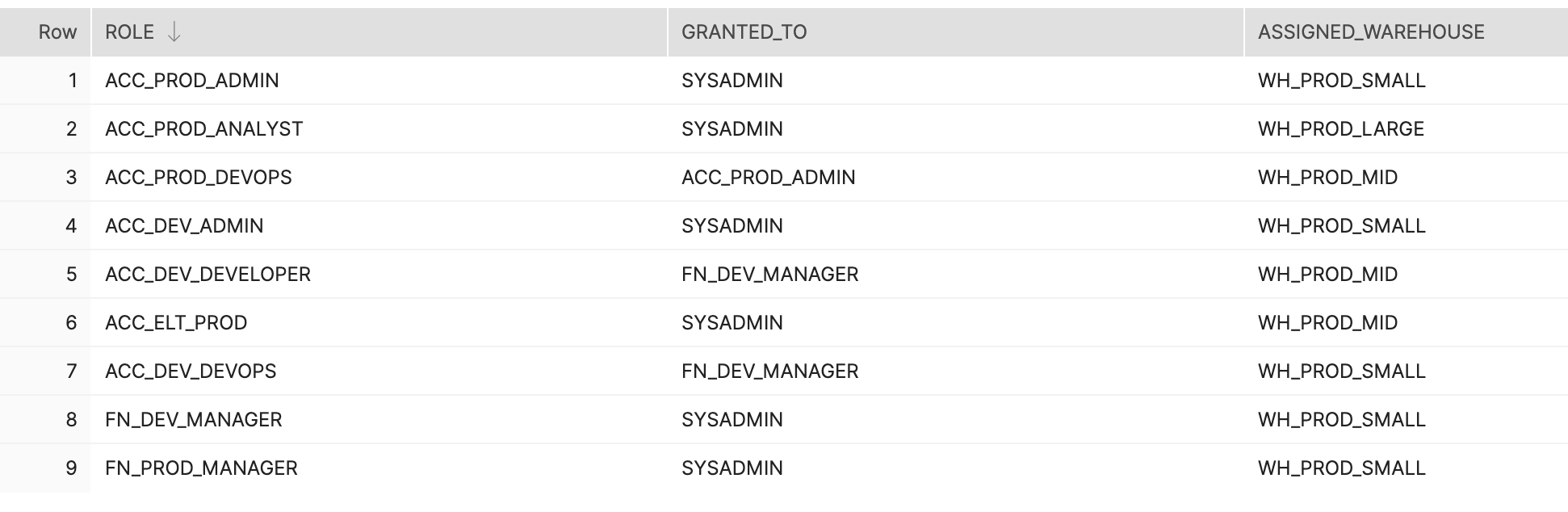
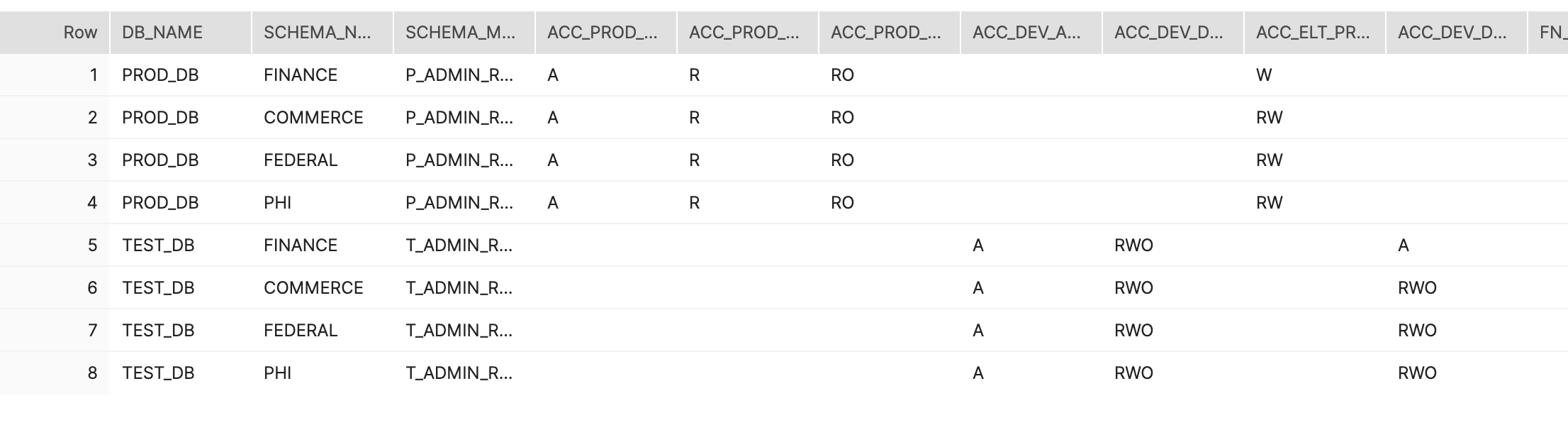
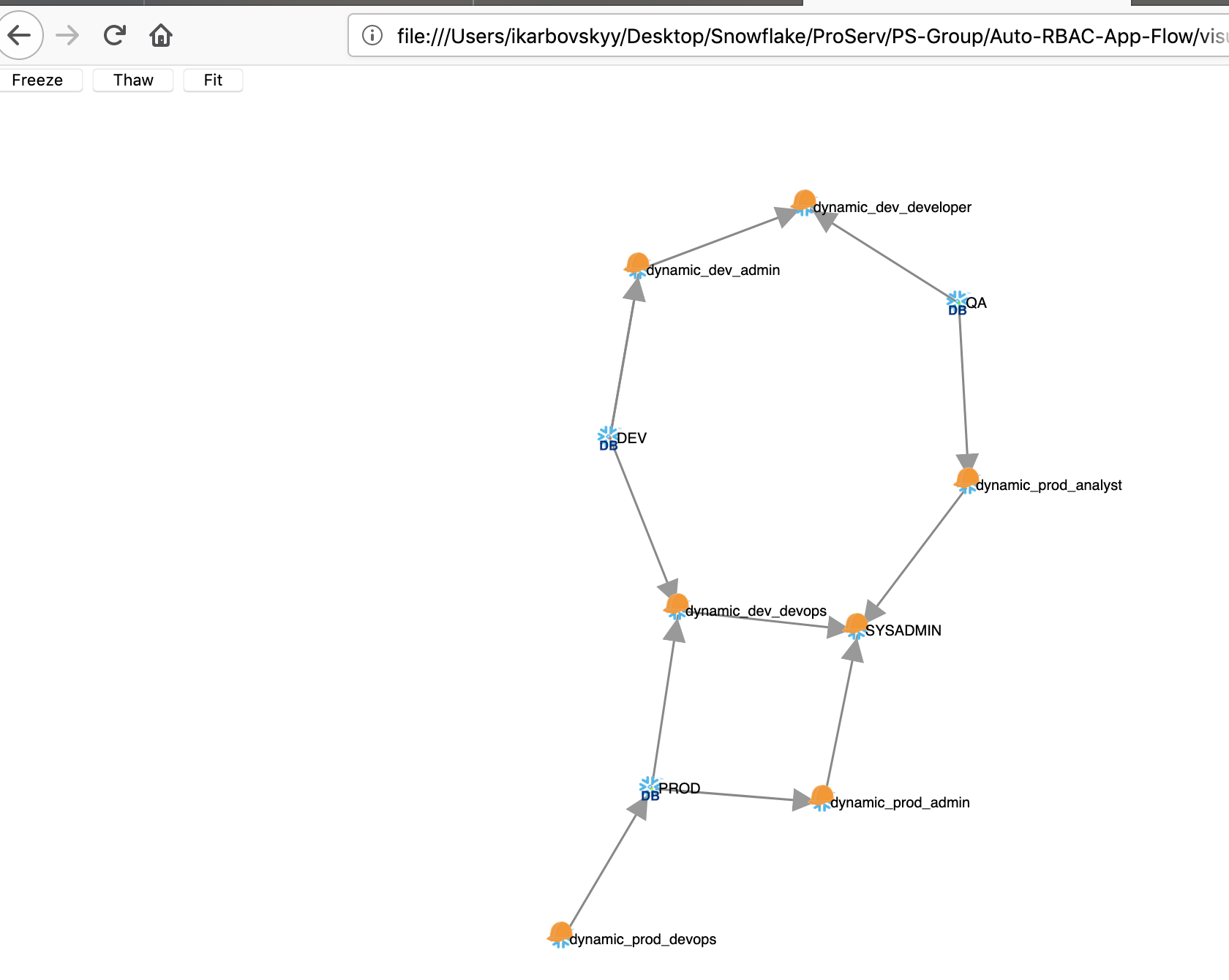
1. As a first step, add all Roles, Warehouse names and granted to Role names into a local table on HTML page. Roles and Warehouses are mandatory to populate. You can skip GrantingTo to create a Functional Role and you may Grant to a parent role manually later. You can create a hierarchy by assigning roles to ‘GrantingTo’ when the parent role will have a higher level having all privileges from the granter. Follow you designed Roles-Access matrics. Save output to a CSV table as ‘table-1.csv’ and store it in the local drive when your Python application located, and click ‘Continue’:  
   
2. By default PROD, QA, DEV working environments are set. Please add, fix database names, schemas, roles to manage schema access for your use case. Click ‘Continue’.  
   
3. On this page, Roles and DB with Schemas will be populated from the previous page.  
   
4. Populate Roles access permissions editing a table on the page.

A - DB and Schema full access - no read-write; R - Read; W - Write; O - Grants ALL on schema.  
Combinations - **S**, **A**, R, W, RW, RW(O).   
 **S**(ecurity) - should be used only as a separate setting, ‘grant CREATE USER, CREATE ROLE, MANAGE GRANTS to role ...’  
 **A**(dmin) - NO READ-WRITE on tables, will have ALL privileges on databases and schemas, plus ‘grant CREATE DATABASE, CREATE WAREHOUSE, CREATE SHARE, IMPORT SHARE ...’  
 **R** - ‘grant select on all tables in schema ’  
 **W** - ‘GRANT INSERT, UPDATE, DELETE, REFERENCES ON ALL TABLES IN SCHEMA ...’

**RW** - ‘GRANT ALL ON ALL TABLES IN SCHEMA ...’   
 RW**O** - in combination with ‘RW’ - ‘O’ has ‘grant USAGE on database…’ and ‘GRANT ALL ON SCHEMA ...’  


# II. Process initial data to create Roles, and Permissions

### Process initial data to create Roles

1. Python application will be executed and will use two tables from step I. The app will copy data from the CSVs files to tables ‘ROLE\_RBAC\_MAIN\_PY’ and ‘RBAC\_DB\_ACCESS\_MNGD’.
2. To create the output script please run the main stored procedure, ‘SP\_RBAC\_PROCESSOR()’
3. The main SP will read data from the first created table and will call subroutine process.   
   
4. The main SP will iterate through the roles and filter data by role from a second table, assigning to roles a corresponding permission, ‘SP\_ROLE\_ASSIGNER(...)’ 
   1. By default, the stored procedure creates an output script for review and execution. To run the stored procedure and execute commands update parameter to ‘true’ in this line - ‘var EXECUTE\_Script = false;’
5. Before assigning Roles to Users a Roles to DB Objects Visualization Graph will be generated.   
   

**CONCLUSION**

The application still in constant development by PS Security Tech group. The main goal is to automate and visualize a creation of the RBAC process and make it easier for SAs and customers. The application will able to be set for specific Role model option, as Simple model, Multi Layers model, with combination roles ETL, Reportes, Shares. The models will be pre built tables with provided Role access diagrams. These Models can be tuned for customer needs.

**REFERENCES**

[Design Doc](https://docs.google.com/document/d/1kLZHa4-PyeUzgd45u-2pTSiUc6F2uTEWrkzY4EAvVhg/edit#)