Bryce Bowles
INFO-610-901-2019 Fall
09/26/2019

Automated System's Security Access DBMS

Project Definition:

The purpose of *Automated System's Security Access DBMS* is to update the Bureau of Insurance's (BOI) security tracking methods to increase data protection of internal system and network resources.

Mission Objectives:

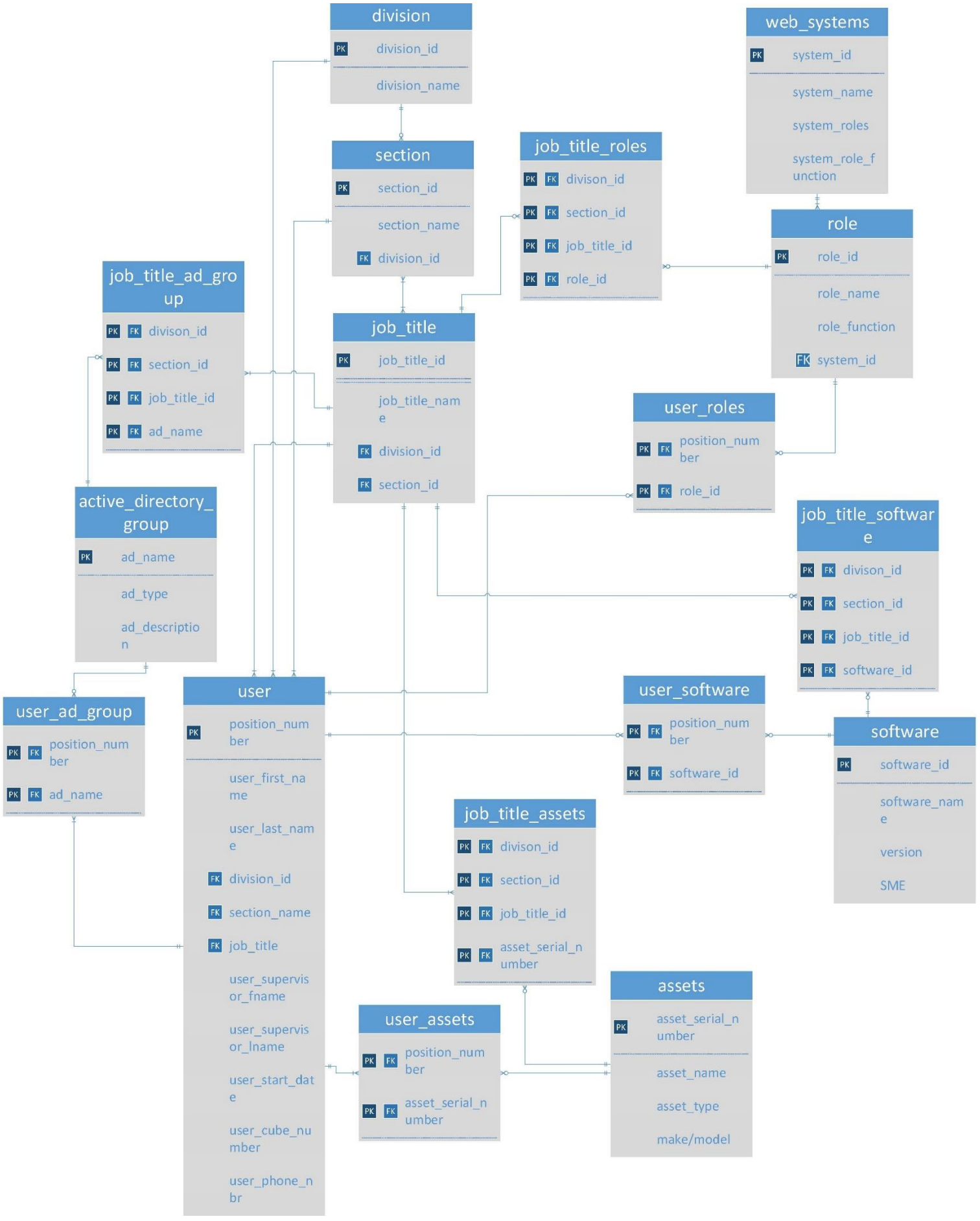
- Maintain record of BOI systems
- Maintain access roles per system
- Consolidate/unify system access per department and position
- Assign correct roles to new employees and employees changing positions
- Audit levels of system access
- Increase efficiency and effectiveness of access requests during employee position changes
- Providing a tool for access reference

The State Corporation Commission's (SCC) internal systems contain personal identifiable information (PII) and sensitive business information on insurance agents and companies therefore, maintaining the confidentiality, integrity & availability of systems and business data is a high priority. One of Automated System's goals is to protect the confidentiality of employees', businesses', and citizens' information by securing system and network resources.

Automated System's process currently consists of using a variety of methods to track system security access for the BOI including excel spreadsheets and service desk tickets. The team is responsible for activating, updating and deactivating user's system access as requested by division management. The manager of Automated systems is responsible of conducting yearly system access audits. Without a centralized database, Automated Systems struggles to keep track of employee's access throughout new hires, promotions/appraisals, and employees leaving. As of now employees who are promoted or change sections keep their security rights, even if they are not applicable to their new position at the Commission.

This database will help produce the appropriate roles for a position creating consistency throughout departments and job titles (with the exception of optional roles for additional access) and reduce the number of access roles that are kept when changing positions. The DBMS will unify and consolidate system access to improve data security.

It is required that the database produce reports (or queries that return data) containing access rights of all BOI systems for a particular job title/position. This will enable Automated Systems to reference access levels for onboarding/offboarding/promotions as well as increase effectiveness of security audits. With the DBMS, Automated Systems will easily evaluate and update system security access roles.


division
PK division_id
division_name
web_systems
PK system_id
system_name
system_roles
system_role_function
section
PK section_id
section_name
FK division_id
job_title_roles
PK FK divison_id
PK FK section_id
PK FK job_title_id
PK FK role_id
role
PK role_id
role_name
role_function
FK system_id
job_title_ad_group
PK FK divison_id
PK FK section_id
PK FK job_title_id
PK FK ad_name
job_title
PK job_title_id
job_title_name
FK division_id
FK section_id
user_roles
PK FK position_number
PK FK role_id
active_directory_group
PK ad_name
ad_type
ad_description
job_title_software
PK FK divison_id
PK FK section_id
PK FK job_title_id
PK FK software_id
user_ad_group
PK FK position_number
PK FK ad_name
user
PK position_number
user_first_name
user_last_name
FK division_id
FK section_name
FK job_title
user_supervisor_fname
user_supervisor_lname
user_start_date
user_cube_number
user_phone_nbr
user_software
PK FK position_number
PK FK software_id
software
PK software_id
software_name
version
SME
job_title_assets
PK FK divison_id
PK FK section_id
PK FK job_title_id
PK FK asset_serial_number
user_assets
PK FK position_number
PK FK asset_serial_number
assets
PK asset_serial_number
asset_name
asset_type
make/model

SQL for Database:

```
DROP TABLE IF EXISTS division;
CREATE TABLE division
   (division_id int primary key,
   division_name varchar (255));

INSERT INTO division
   (division_id, division_name)
   VALUES (1, 'Commissioners Office'),
   (2, 'Policy, Compliance, & Administration'),
   (3, 'Agent Regulation'),
   (4, 'Property & Casualty'),
   (5, 'Financial Regulation'),
   (6, 'Life & Health');

DROP TABLE IF EXISTS section
CREATE TABLE section
   (section_id int primary key,
   section_name varchar (255),
   division_id int NOT null REFERENCES division (division_id));

INSERT INTO section
   (section_id, section_name, division_id)
   VALUES (1, 'Commissioners Office'),

INSERT INTO section
   (section_id, section_name, division_id)
```

```
   VALUES (1, 'Deputy Commissioner', 3),

DROP TABLE IF EXISTS job_title
CREATE TABLE job_title
   (job_title_id int primary key,
   job_title_name varchar (255),
   section_id int NOT null REFERENCES section (section_id));

DROP TABLE IF EXISTS web_systems
CREATE TABLE web_systems
   (system_id int primary key,
   system_name varchar (255),
   system_roles varchar (255),
   system_role_function varchar (255));

DROP TABLE IF EXISTS role
CREATE TABLE role
   (role_id int primary key,
   role_name varchar (255),
   system_id int NOT null REFERENCES web_systems (system_id));

DROP TABLE IF EXISTS job_title_roles
CREATE TABLE job_title_roles
   (role_id int NOT NULL REFERENCES role (role_id),
   division_id int NOT NULL REFERENCES job_title (division_id),
   section_id int NOT NULL REFERENCES job_title (section_id),
   job_title_id int NOT NULL REFERENCES job_title (job_title_id),
   PRIMARY KEY (division_id, section_id, job_title_id, role_id));
```

```
DROP TABLE IF EXISTS user
CREATE TABLE user
   (position_number int primary key,
   user_first_name varchar (255),
   user_last_name varchar (255),
   division_id int NOT NULL REFERENCES division (division_id),
   section_id int NOT NULL REFERENCES section (section_id),
   job_title_id int NOT NULL REFERENCES job_title (job_title_id),
   user_supervisor_fname varchar (255),
   user_supervisor_lname varchar (255),
   user_start_date date,
   user_cube_number varchar (10),
   user_phone_number varchar (20));

DROP TABLE IF EXISTS user_roles
CREATE TABLE user_roles
   (position_number int NOT NULL REFERENCES user (position_number),
   role_id int NOT NULL REFERENCES role (role_id),
   PRIMARY KEY (position_number, role_id));

DROP TABLE IF EXISTS software
CREATE TABLE software
   (software_id int primary key,
   software_name varchar (255),
   version varchar (255),
   SME varchar (255));

DROP TABLE IF EXISTS job_title_software
CREATE TABLE job_title_software
```

```
   (software_id int NOT NULL REFERENCES software (software_id),
   division_id int NOT NULL REFERENCES job_title (division_id),
   section_id int NOT NULL REFERENCES job_title (section_id),
   job_title_id int NOT NULL REFERENCES job_title (job_title_id),
   PRIMARY KEY (division_id, section_id, job_title_id, software_id));

DROP TABLE IF EXISTS user_software
CREATE TABLE user_software
   (software_id int NOT NULL REFERENCES software (software_id),
   position_number_id int NOT NULL REFERENCES user (position_number),
   PRIMARY KEY (software_id, position_number));

DROP TABLE IF EXISTS assets
CREATE TABLE assets
   (asset_serial_number int primary key,
   asset_name varchar (255),
   asset_type varchar (255),
   make/model varchar (255));

DROP TABLE IF EXISTS job_title_assets
CREATE TABLE job_title_assets
   (asset_serial_number int primary key REFERENCES job_title_assets (asset_serial_number),
   division_id int NOT NULL REFERENCES job_title (division_id),
   section_id int NOT NULL REFERENCES job_title (section_id),
   job_title_id int NOT NULL REFERENCES job_title (job_title_id),
   PRIMARY KEY (division_id, section_id, job_title_id, asset_serial_number));

DROP TABLE IF EXISTS user_assets
CREATE TABLE user_assets
```

```
   (position_number int primary key REFERENCES user (position_number),
   (asset_serial_number int primary key REFERENCES assets (asset_serial_number),
   PRIMARY KEY (position_number, asset_serial_number));

DROP TABLE IF EXISTS active_directory_group
CREATE TABLE active_directory_group
   (ad_name varchar (255) PRIMARY KEY,
   ad_type varchar (255),
   ad_description varchar (255));

DROP TABLE IF EXISTS user_ad_group
CREATE TABLE user_ad_group
   (position_number int primary key REFERENCES user (position_number),
   (ad_name varchar (255) primary key REFERENCES active_directory_group (ad_name),
   PRIMARY KEY (position_number, asset_serial_number));

DROP TABLE IF EXISTS job_title_ad_group
CREATE TABLE job_title_ad_group
   (ad_name varchar (255) primary key REFERENCES active_directory_group (ad_name),
   division_id int NOT NULL REFERENCES job_title (division_id),
   section_id int NOT NULL REFERENCES job_title (section_id),
   job_title_id int NOT NULL REFERENCES job_title (job_title_id),
   PRIMARY KEY (division_id, section_id, job_title_id, ad_group));
```