**Software Maintenance, ESOF 326, Spring 2020**
**Proof of Concept Prototype for Web Service Middleware Handling Permissions**
**Jan. 22**

Permission situations:
User Super: Permission to do all. A super user.
User Admin: Permission to create courses and outcomes. An administrative users.
User SE: Permission to associate courses and outcomes with the SE program.
User EE: permission to associate courses and outcomes with the EE program.
User None: No permissions

Determine the common functionality that can be placed into our web service to handle permissions. Feel free to assume that each user has logged into AbOut, so has been validated by CAS, and the web services has given the user a security token. Feel free to ignore semester intervals.

Tasks to consider:
1. User Super successfully utilizes the web service to create a course.
2. User Super successfully utilizes the web service to create an outcome.
3. User Admin successfully utilizes the web service to create a course.
4. User Admin successfully utilizes the web service to create an outcome.
5. User None unsuccessfully attempts to utilize the web service to create a course.
6. User SE successfully utilizes the web service to associate an existing course to the SE program.
7. User SE successfully utilizes the web service to associate an existing outcome to the SE program.
8. User EE unsuccessfully attempts to utilize the web service to associate an existing course to the SE program.
9. User None unsuccessfully attempts to utilize the web service to associate an existing course to the SE program.
10. User EE unsuccessfully attempts to utilize the web service to associate an existing outcome to the SE program.
11. User SE successfully utilizes the web service to associate an existing courses associated with SE, and an existing outcomes associated with SE, so that the course will measure the outcome.
12. User EE unsuccessfully attempts to utilize the web service to associate an existing courses associated with SE, and an existing outcomes associated with SE, so that the course will measure the outcome.
13. User None unsuccessfully attempts to utilize the web service to associate an existing courses associated with SE, and an existing outcomes associated with SE, so that the course will measure the outcome.
14. User Super successfully utilizes the web service to give User EE administrative permissions in the SE program.

1. What information will the security token contain? When information comes from a table, give the exact field name.

   Some identifier of the user that won't change. Maybe use ssh

2. Give pseudo code for checking permissions in the above cases (the goal is to have a few functions which can be used over and over to handle permissions).

   Can't be done. We need a new way of handling permissions.

   On the next page is a new way to handle permissions:

## Roles

super users :
Can view and change all
Only users with CRUD for programs and who can generate semesters

program managers:
CRUD for users, prefixes, outcomes, subjects, courses, students, program_outcomes, program_courses, measures, program_students
CRUD for course offerings, adding and removing students

offering owners  (faculty):
CRUD for assessment, scores
(students and program_students?)

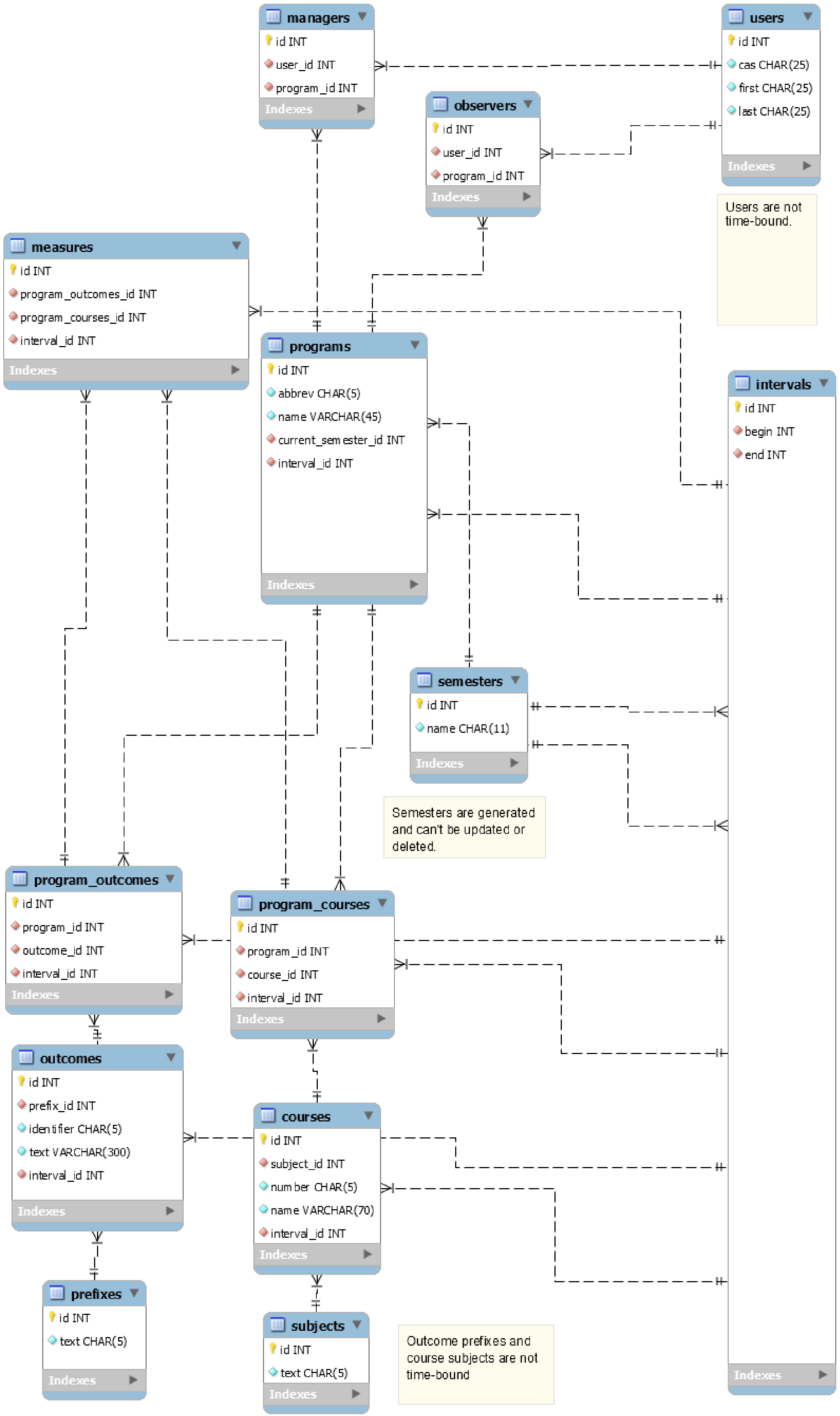
program observers:
Can observe all, but with students names redacted

public:
Can see items and lists of programs, semesters, prefixes, outcomes, program_outcomes, Subjects, courses, program_courses, measures, all reports

## Database

Permissions for programs are stored in two tables: “managers” and “observers”. Each of these will be a relation between users and programs.

Permissions for offerings are stored in an “owners” table which is a relation between users and offerings (not in our current model).


managers
id INT
user_id INT
program_id INT
Indexes
users
id INT
cas CHAR(25)
first CHAR(25)
last CHAR(25)
Indexes
observers
id INT
user_id INT
program_id INT
Indexes
Users are not time-bound.
measures
id INT
program_outcomes_id INT
program_courses_id INT
interval_id INT
Indexes
programs
id INT
abbrev CHAR(5)
name VARCHAR(45)
current_semester_id INT
interval_id INT
Indexes
intervals
id INT
begin INT
end INT
Indexes
semesters
id INT
name CHAR(11)
Indexes
Semesters are generated and can't be updated or deleted.
program_outcomes
id INT
program_id INT
outcome_id INT
interval_id INT
Indexes
program_courses
id INT
program_id INT
course_id INT
interval_id INT
Indexes
outcomes
id INT
prefix_id INT
identifier CHAR(5)
text VARCHAR(300)
interval_id INT
Indexes
courses
id INT
subject_id INT
number CHAR(5)
name VARCHAR(70)
interval_id INT
Indexes
prefixes
id INT
text CHAR(5)
Indexes
subjects
id INT
text CHAR(5)
Indexes
Outcome prefixes and course subjects are not time-bound