FOOGLE

<http://scripts.bcs.org/sfiaplus/sfia.htm> za unos skillova (unosite podkategorije podkategorija). Kategorije su već unesene i skillovi za prvu kategoriju.

create table title (

id serial primary key,

title varchar(50) not null);

create table category(

id serial primary key,

name varchar(50) not null);

create table foogle\_user (

id serial primary key,

email varchar(50) unique not null,

firstname varchar(50),

lastname varchar(50),

title integer references title(id) on delete restrict on update cascade,

role char(1) check (role in ('P', 'S', 'A')) default 'S',

activity char(1) check (activity in ('A', 'P')) default 'P',

confirmed boolean,

linkedin varchar(255),

linkedin\_id varchar(255)

);

create table user\_category(

category integer references category on delete restrict on update cascade,

foogle\_user integer references foogle\_user on delete restrict on update cascade,

id serial,

primary key(category, foogle\_user));

create table skill (

id serial primary key,

name varchar(50) not null,

category integer references category on delete restrict on update cascade not null);

create table recommendation (

skill integer references skill,

teacher integer references foogle\_user on delete restrict on update cascade,

student integer references foogle\_user on delete restrict on update cascade,

id serial,

primary key(skill, teacher, student));

create table student (

id integer references foogle\_user on delete restrict on update cascade primary key,

jmbag varchar(5) unique not null,

gpa float not null,

experience float);

create table experience (

id serial primary key,

student integer references student(id) on delete restrict on update cascade,

company varchar(50) not null,

months\_of\_service integer not null);

create function years\_of\_service() returns trigger as $$

declare months integer;

years float;

begin

select cast((select sum(months\_of\_service) from experience where student=new.student) as float)/12 into years ;

update student set experience=years where id=new.student;

return null;

end;

$$language plpgsql;

create trigger years\_of\_service\_insert after insert on experience for each row execute procedure years\_of\_service();

create trigger years\_of\_service\_update after update on experience for each row execute procedure years\_of\_service();