**View**

**1.**

create or replace view Studenti\_diplomuar as

select s.sid,s.emri.es.titulli\_temes

from student s,edukimi\_studentit es

where s.sid=es.sid

and es.i\_diplomuar='Po'

order by s.emri asc;

**2.**

create or replace view aplikimi as

select s.sid,s.emri.es.titulli\_temes

from student s,pozita p,studenti\_aplikon sa

where s.sid=sa.sid

and sa.pozId=p.pozId

and sa.data\_aplikimit>'2019.01.01';

**Trigger**

**1.**

delimiter //

CREATE TRIGGER upd\_check BEFORE UPDATE ON paga\_sipas\_pozites

FOR EACH ROW

BEGIN

IF NEW.paga < =500 THEN

SET NEW.paga = 500;

ELSEIF NEW.paga > 500 THEN

SET NEW.paga = 700;

END IF;

END;//

delimiter ;

**2.**

create trigger kontrolloPagen

before insert on mentori

for each row

begin

if new.paga<1500

then signal sqlstate '45000'

set message\_text='Paga duhet te jete me e madhe se 1500';

end if;

end //

**Procedure**

**1.**

delimiter $$

create procedure studenti\_shpendane()

begin

select sid from shperndarja;

end $$

delimiter ;

**2.**

delimiter $$

create procedure editMentor(para\_mid int,

para\_emri varchar (30),

para\_mbiemri varchar (30),

para\_data\_lindjes date,

para\_paga real)

begin

Update mentori

set emri = para\_emri,

mbiemri= para\_mbiemri,

data\_lindjes = para\_data\_lindjes,

paga = para\_paga,

where mid = para\_mid;

end//

delimiter ;

**Function**

**1.**

delimiter $$

create function kohezgjatja\_studimeve(viti\_fillimit integer,viti\_mbarimit integer)

returns integer

begin

return viti\_mbarimit-viti\_fillimit;

end $$