MySQL baze

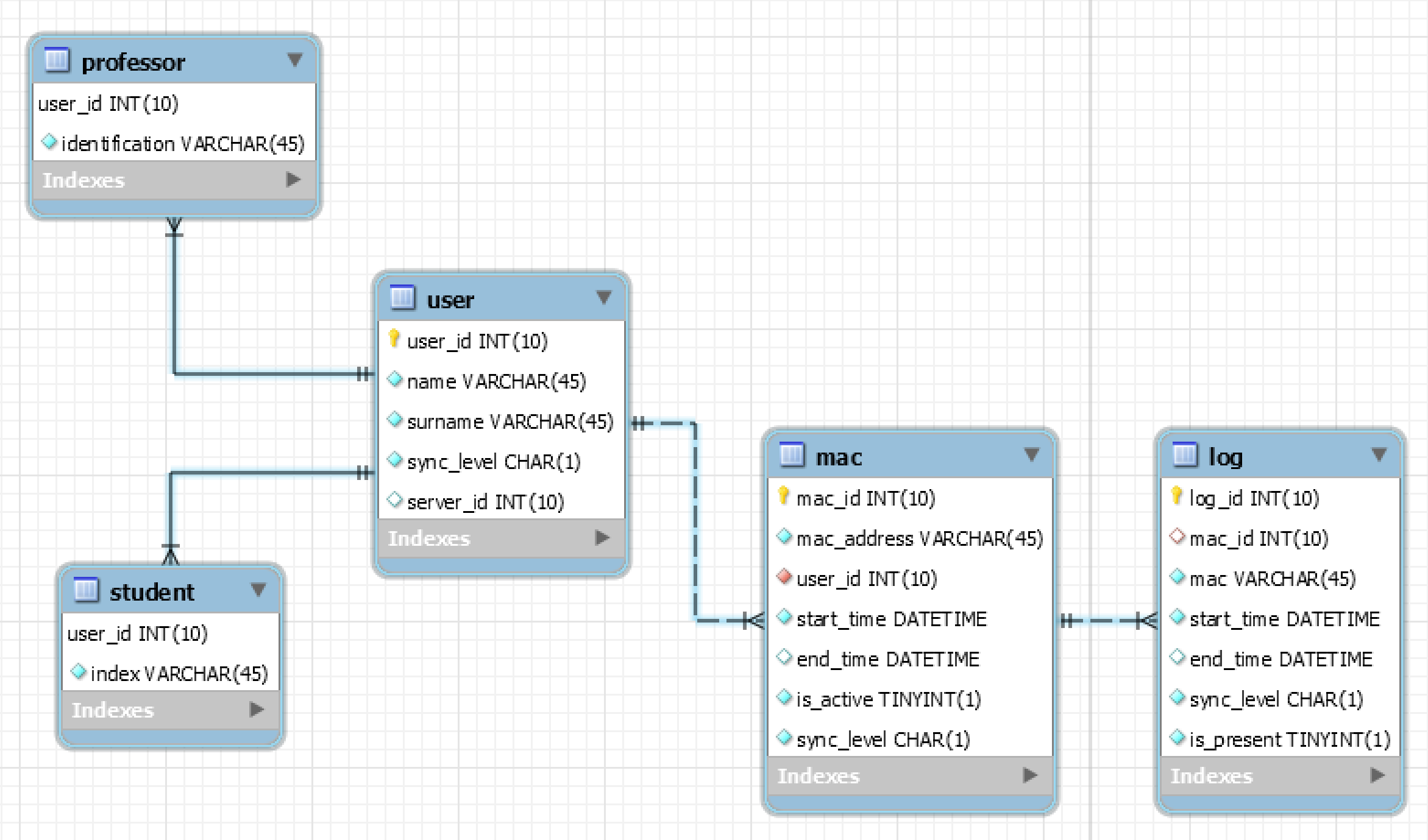
Postoje dve strukture MySQL baza u projektu, jedna koja je predviđena za server, i druga koja je predviđena za uređaje koji beleže prisustvo (logeri). Obe baze uz sebe sadrže i SQL procedure pomoću kojih je obrada u bazi svedena na SQL, kako bi se smanjila potrebna komunikacija između baze i ostalih programa, i kako bi se programi rasteretili od logike baze.

Baze su rađene u MySQL Workbench-u. U njega se mogu i importovati SQL fajlovi u kojima se nalaze baze i procedure.

# Loger baza

## Struktura

Baza na logeru ima strukturu prikazanu na slici 1.



Slika 1: dijagram baze logera

* ***user***

Entitet koji predstavlja korisnika. Polje *sync\_level* je nivo sinhronizacije date instance sa serverom (objašnjeno u kasnijem delu). Polje *server\_id* je jednako primarnom ključu na serveru (loger ga dobija od servera prilikom sinhronizacije), i služi da se korisnik jednoznačno adresira serveru u slučaju da je došlo do menjanja imena/prezimena/indexa/itd.

* ***student***

Specijalizacija entiteta *user*. Sadrži *index* studenta koji je jedinstven za svakog studenta. Nivo sinhronizacije sa serverom je označen poljem u *user*-u.

* ***professor***

Specijalizacija entiteta *user*. Sadrži identifikacioni broj (*identification*) koji je jedinstven za svakog profesora.

* ***mac***

Povezana je sa *user*-om i sadrži *MAC* adresu koji korisnik koristi, ili koju je koristio u određenom periodu. Prilikom novog registrovanja korisnika, podaci o korisniku se pamte u instanci *user*-a i u instanci *student*-a ili *professor*-a. Ako se korisnik registruje sa uređaja na kojem ranije nije bio registrovan, nova *MAC* adresa se upisuje u instancu *mac*, a njegova prethodna *MAC* adresa postaje neaktivna. Takođe, ako za nekog drugog korisnika postoji ta nova *MAC* adresa upisana instanci *mac*, i ona postaje neaktivna (npr. ako je korisnik A dao korisniku B uređaj X, pri registraciji korisnika B preko uređaja X, za korisnika A uređaj X postaje neaktivan).

* ***log***

Predstavlja belešku o prisustvu. Pamti se vreme prisustva i *MAC* adresa. U slučaju da za odgovarajuću *MAC* adresu ne postoji instanca *mac*, polje *mac\_id* ostaje *NULL*, dok se čuva samo *MAC* adresa u *log*-u. Ako se neki korisnik registruje preko neke *MAC* adrese, svi *log*-ovi koji su zabeležili tu adresu, a nisu povezani sa nekom instancom *mac*, postaju povezani sa novom instancom *mac*.

*Log* je povezan sa instancom *mac*-a, a samim tim i sa *user*-ima sa kojima su *mac*-ovi povezani.

## Logovanje

Na određen interval, proverava se prisustvo uređaja. Gleda se entitet *log*. Sve instance koje su nezavršene (*end\_time IS NOT NULL*) se beleže kao neprisutne (­*is\_present = 0*). Zatim se uzima podatak o tome koje su sve *MAC* adrese povezane.

1. One koje su povezane i za koje **ne postoji** nezavršena instanca *log* su došle posle prethodnog beleženja i one su od skoro prisutne – upisuje im se nova instanca *log*.
2. One koje su povezane i za koje **postoji** nezavršena instanca *log* su prisutne od ranije, i menja im se prisustvo u prisutne (*is\_present = 1*).
3. Sve instance *log* kojima ostane zabeleženo da su neprisutne, su stvarno neprisutne. Beleži im se vreme završetka (trenutno vreme).

## Sinhronizacija

Svi podaci u bazi imaju zabeležen nivo sinhronizacije sa serverom:

1. ***user***

*sync\_level – {s, n, x, k}*

* 1. *s*

Instanca je sinhronizovana sa serverom, i ne treba da se šalje pri sinhronizaciji.

* 1. *n*

Instanca postoji na serveru, ali je lokalno promenjena, i treba da se pošalje serveru pri sinhronizaciji kako bi i server kod sebe promenio instancu.

* 1. *x*

Instanca ne postoji na serveru, i treba da se pošalje serveru pri sinhronizaciji.

* 1. *k*

Instanca postoji na serveru, ali u lokalnoj bazi još nije dobijen primarni ključ sa servera, tako da instanca ne sme lokalno da se menja (inače loger ne bi mogao serveru da adresira *user*-a pri sinhronizaciji).

1. ***mac, log***

*sync\_level – {s, x}*

* 1. *s*

Instanca je sinhronizovana sa serverom, i ne treba da se šalje pri sinhronizaciji

* 1. *x*

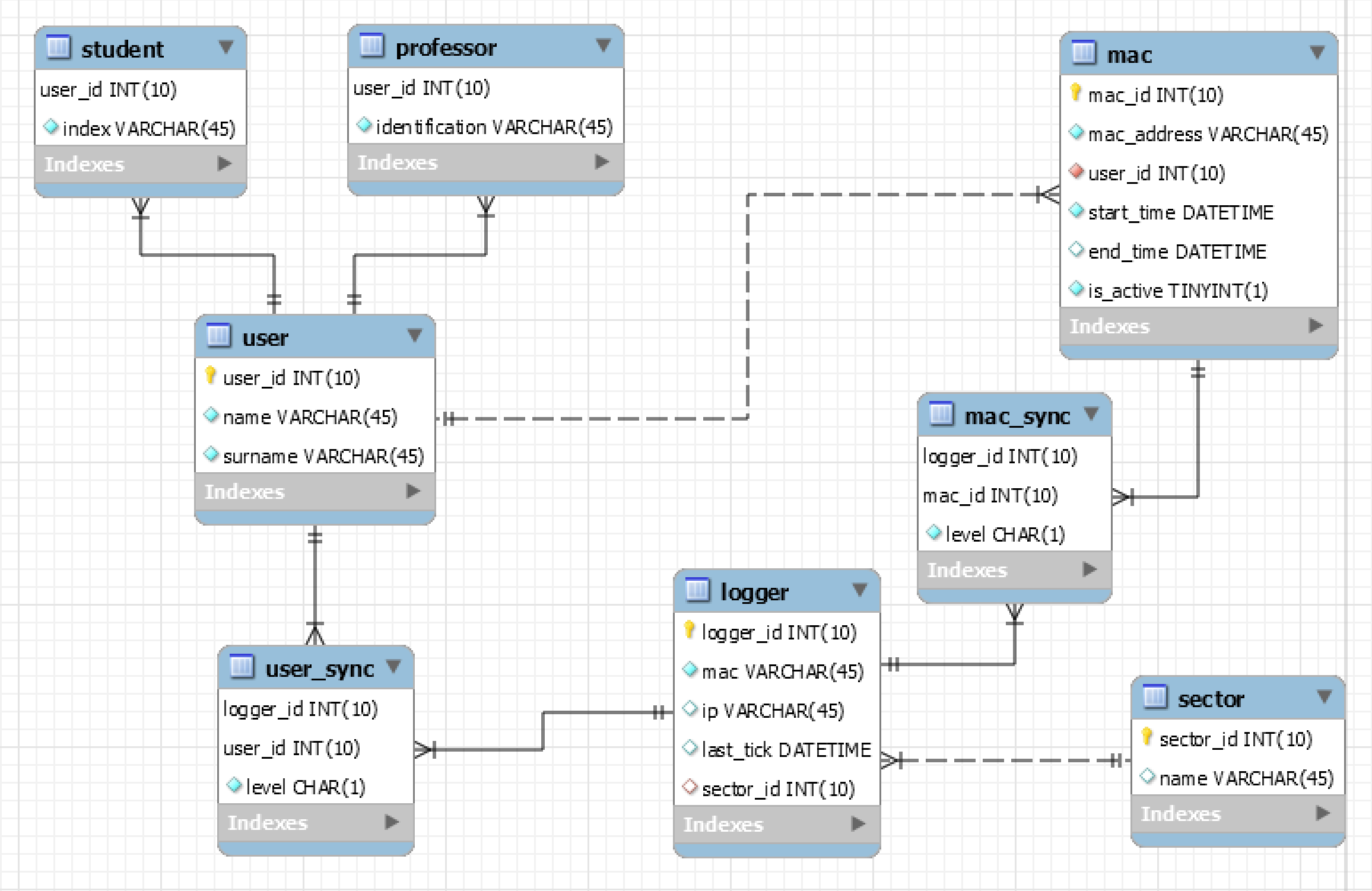
Instanca ne postoji na serveru, i treba da se pošalje serveru pri sinhronizaciji.

Jedini način na koji instanca *mac* može da se promeni jeste da postane neaktivna. Međutim, postaje neaktivna ubacivanjem nove instance *mac*, tako da je dovoljno serveru poslati podatak o novoj instanci.

# Server baza

## Struktura

Struktura baze na serveru je prikazana na slici 2.



Slika 2: dijagram baze server (bez entiteta log, zbog greške pri iscrtavanju dijagrama)

Struktura je slična bazi u logeru. Razlika je u entitetima:

1. ***sector***

Prostorija/prostor u kome se loger nalazi. Entitet *log* je povezan sa entitetom *sector*, čime se za svaki *log* pamti iz koje je prostorije.

1. ***logger***

Predstavlja uređaj koji loguje. Server pamti koji su sve logeri umreženi, kao i sinhronizaciju serverske baze sa svakim logerom.

1. ***user\_sync, mac\_sync***

Kao i kod logera, server beleži nivo sinhronizacije *user*-a i *mac*-ova, samo što se na serveru to beleži za svaki loger (*log*-ovi se ne beleže jer njih samo šalje loger na server, server ih ne sinhronizuje sa ostalim logerima).

## Sinhronizacija

Svi podaci u bazi imaju zabeležen nivo sinhronizacije sa serverom:

1. ***user\_sync***

*level – {s, n, x, k}*

* 1. *s*

Instanca je sinhronizovana sa logerom, i ne treba da se šalje pri sinhronizaciji.

* 1. *n*

Instanca postoji kod logera, ali je promenjena na serveru (od strane nekog drugog logera), i treba da se pošalje logeru pri sinhronizaciji kako bi i on kod sebe promenio instancu.

* 1. *x*

Instanca ne postoji kod logera, i treba da se pošalje logeru pri sinhronizaciji.

* 1. *k*

Instanca postoji kod logera, ali još nije dobio primarni ključ sa servera, tako da je treba poslati pri sledećoj sinhronizaciji.

1. ***mac\_sync***

*level – {s, x}*

* 1. *s*

Instanca je sinhronizovana sa logerom, i ne treba da se šalje pri sinhronizaciji

* 1. *x*

Instanca ne postoji na logeru, i treba da se pošalje serveru pri sinhronizaciji.

# Proces sinhronizacije između servera i logera

Server sadrži API koji čeka zahteve za sinhronizaciju. Loger na određeni period odrađuje sinhronizaciju sa serverom. Proces sinhronizacije:

1. Razmena korisnika
   1. Loger eksportuje korisnike iz svoje baze i šalje serveru zahtev za primanje tih korisnika.
   2. Server upisuje korisnike kod sebe u bazu, i kao odgovor logeru šalje eksportovane korisnike iz svoje baze.
   3. Loger upisuje nove korisnike u svoju bazu.
2. Razmena MAC-ova
   1. Loger eksportuje MAC-ove (instance) iz svoje baze I šalje serveru zahtev za primanje.
   2. Server upisuje te MAC-ove kod sebe, i kao odgovor logeru šalje eksportovane MAC-ove iz svoje baze.
   3. Loger upisuje nove MAC-ove u svoju bazu.
3. Razmena logova
   1. Loger eksportuje logove iz svoje baze i šalje serveru zahtev za primanje.
   2. Server upisuje logove kod sebe u bazu, i odgovara da je sve ok.
4. Loger završava proces sinhronizacije.