

*Baza podataka poslodavaca I radnika****.***

*IT350 – Baze podataka*

*Projektna dokumentacija*

***Student****:*

Dušan Stanković 3611

***Asistent****:*

Tamara Vukadinović

# *Apstrakt Projektnog Zadatka*

## Opis problema

Za potrebe Univerziteta potrebno je napraviti aplikaciju za vođenje evidencije o predatim predispitnim obavezama studenata I ocenjivanje od strane profesora.

Svaki predmet ima 15 domaćih, 5 testa I jedan projektni zadatak. Student (za svaku nedelju u semestru) može da upload-uje po jedan file za svaki domaći zadatak I svaki projektni zadatak na predmetima koji su mu dodeljeni.

Uslov za predaju domaćeg zadatka u određenoj nedelji je da su svi prethodni zadaci predate (upload-ovani). Tako, u npr. trećoj nedelji uslov je da je student poslao domaće za prethodne dve nedelje.

Postoji I vremensko ograničenje za predaju radova. Nakon toga umanjuje se broj poena za 5% za svaki dan kašnjenja

Uslov za predaju projekta je da student poslao (upload-ovao) apstrakt I da je on odobren od strane profesora ili mentora.

Testovi se generišu iz baze pitanje za dati predmet. Pitanja I ponuđene odgovore (beleži se I tačan odgovor) unosi predmetni profesor kroz aplikaciju. Test sadrži 5 pitanja koja se generišu random I pokrivaju određenu oblast predmeta. Nakon završetka testa, studentu se poeni automatski unose u bazu. Testove I pitanja/odgovore kreira profesor za svoje predmete. Profesor može da pristupi poenima studenata I njihovim radovima samo za predmet na kome je angažovan. Nakon pregledavanja radova, profesor unosi ocenu (broj poena) za određenu predispitnu obavezu.

Svaki predmet ima šifru I naziv. Svaki projektni zadatak ima ime, apstrakt I ključne reči. Svaka predispitna obaveza dodeljena student takođe ima lokaciju file-a koji je student upload-ovao i datum kada je obaveza predata. Predmete, student, profesore I obaveze na sistem dodaje sistem administrator koji ima korisničko ime I šifru.

## SQL upiti vezani za projekat

1. Naći sve predmete koji se drže na nekom studijskom programu (odaberi neki koji imaš u bazi) i nose 6 ili više espb. Prikazati uz naziv predmeta i profesora koji drži predmet.

2. Naći studenta koji ima najveću ocenu za predati projekat.

3. Naći broj predmeta za svakog profesora, ne prikazivati profesore koji drže manje od jednog predmeta.

4. Naći sve testove koji imaju maksimalan broj pitanja. (Npr. mogu biti 3 testa sa istim, najvećim brojem pitanja) Sortirati ih po date\_sent.

5. Za svaki studijski program čiji naziv sadrži reč 'info' prikazati profesore koji drže neke predmete na njemu. Rezultate sortirati po nazivu Studijskog programa po nazivu opadajuće, i po broju esbp rastuće.

# *Uvod*

## Cilj

Cilj ovog projekta je da se dizajnira sistem univerziteta koji omogućava profesoru pregled, ocenjivanje i upravljanje predispitnim obavezama, kao i studentu da pregleda i šalje iste.

Sistem takođe mora omogućiti lak i efikasan način izmene kao i dodavanje novih obaveza, ocenjivane poslatih obaveza.

Sama baza podataka mora biti maksimalno efikasna, sa što manje redudatnih informacija kao i bespotrebnih tabela i view-a.

## Realizacija

Baza podataka će biti kreirana u **MySQL-u** na **Windows** sistemu uz pomoć SQL upita i phpMyAdmin alata.

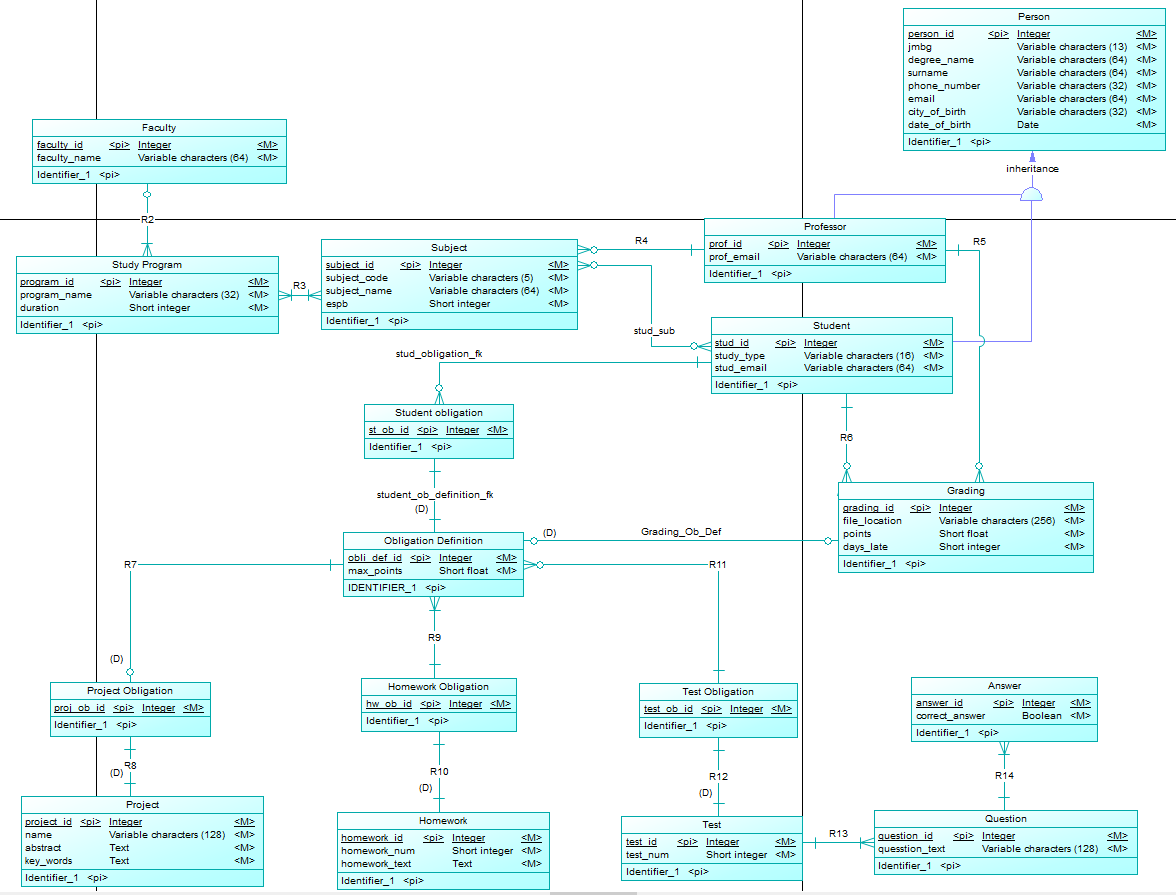
Dizajniranje konceptualnog modela se izvršava uz pomoć **Powerdesigner-a**.

Za izradu SQL upita koristićemo **XAMPP phpMyAdmin.**

# *Projektno rešenje*

## Konceptualni model baze podataka

Konceptualni model predstavlja prvi najniži nivo apstrakcije problema koji je dobijen iz detaljnog opisa projektnog zadatka.



*Slika 1 Konceptualni model baze podataka*

## Kreiranje baze

Sledeći korak je generisanje fizičkog modela od već postojaćeg klasnog modela. Odakle možemo da generišemo SQL kod koji ćemo da koristimo za kreiranje same baze podataka tako što ćemo je importovati sql fajl unutar phpMyAdmin.

Kako bi optimizovao funkcionisanje baze, naknadno sam obradio sve sql upite, takođe sam uklonio sve datume za slanje predispitnih obaveza i ocenjivanja, jer sam uključio audit koji sam postavlja date\_created polja na svakoj koloni svih tabela.

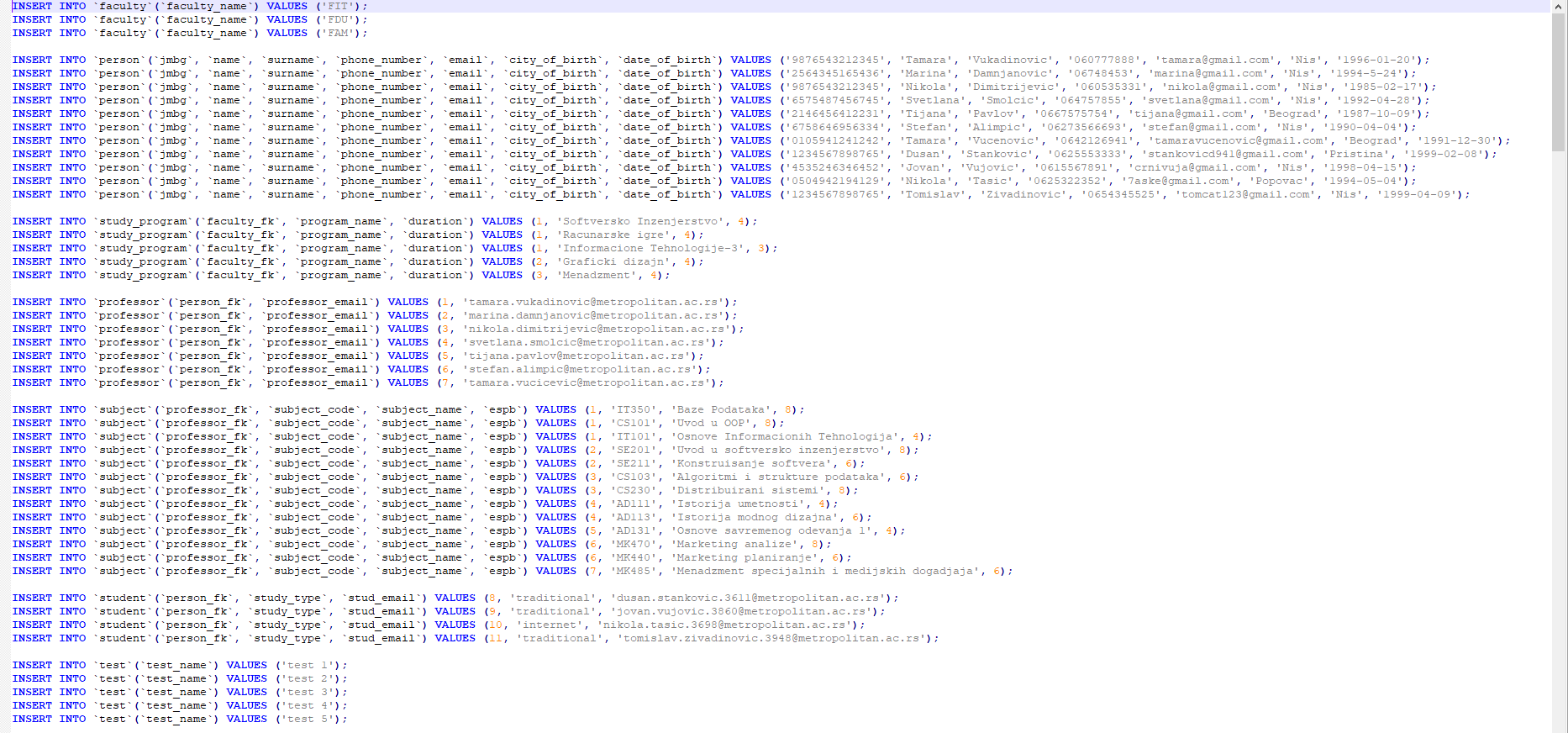


*Slika 2 Generisanje Baze podataka uz pomoć phpMyAdmin-a*

Nakon toga unosimo nekoliko redova u svakoj tabeli baze podataka kako bismo uspešno istestirali sve upite.

Unošenje informacija se može izvršiti preko insertovanje SQL fajla, korišćenjem grafičkog korisničkog interfejsa na phpMyAdmin-u ili direktno kucanjem SQL upita preko konzole na phpMyAdmin-u.

Ja sam u ovom slučaju koristio SQL upite kao na sledećoj slici:

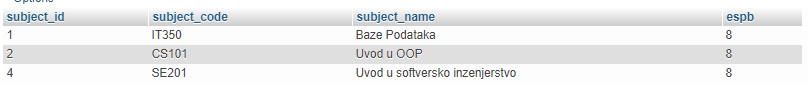


*Slika 3 SQL Insert upiti*

# *Rešenja SQL upita*

## Upit *1*

Naći sve predmete koji se drže na nekom studijskom programu (odaberi neki koji imaš u bazi) i nose 6 ili više espb. Prikazati uz naziv predmeta i profesora koji drži predmet.



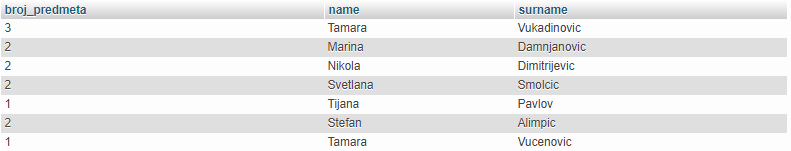
## Upit *2*

Naći studenta koji ima najveću ocenu za predati projekat



## Upit *3*

Naći broj predmeta za svakog profesora, ne prikazivati profesore koji drže manje od jednog predmeta.



## Upit 4

Naći sve testove koji imaju maksimalan broj pitanja. (Npr. mogu biti 3 testa sa istim, najvećim brojem pitanja) Sortirati ih po date\_sent



## Upit 5

Za svaki studijski program čiji naziv sadrži reč 'info' prikazati profesore koji drže neke predmete na njemu. Rezultate sortirati po nazivu Studijskog programa po nazivu opadajuće, i po broju esbp rastuće.)



# *Literatura*

LAMS literatura, lekcije DZ01-DZ13 : <http://lams.metropolitan.ac.rs:8080/lams/index.do>