

Baze podataka

Specifikacija projekta

Računarski fakultet

15. maj 2020.

Važne napomene

1. Projekat se radi u paru. Ukoliko neko ne može da pronađe para, uvek može da mi se obrati za pomoć. U specifičnim slučajevima, biće dopuštena samostalna izrada zadatka. Samostalno urađeni projekti će biti uvaženi, ali će kriterijum ocenjivanja biti **identičan** kao za timski urađene projekte.
2. Timovi se moraju prijaviti do subote, 15. maja 2020 u 14h. Studenti koji ne prijave tim (bez obzira da li će raditi individualno ili u paru) do predviđenog roka neće moći da rade projekat.
3. Obavezno je korišćenje dodeljene MSSQL baze koja će se nalaziti na udaljenom serveru. Nije dozvoljeno koristiti servere podešene lokalno, ili privremene baze za projekat.
4. Za bazu podataka može da se upotrebi bilo [baza](#) koju smo koristili na vežbama, bilo neka po slobodnom izboru.
5. Ako se tim odluči da pravi sopstvenu bazu podataka, ona treba da sadrži barem 7 tabela, i svaka tabela treba da ima popunjeno 10-15 redova za demonstraciju. Podaci mogu da budu obezličeni, ali ne i nasumično generisani (primer: u polju "Ime" ne mogu stajati podaci koji liče na "ABbsfAbdvvg"). Veze između tabela moraju da budu definisane ograničenjima, a baza treba da bude normalizovana u skladu sa onim što je rađeno u prvom delu semestra. Eksportovanje DDL skripte iz PowerDesigner alata, za studente koji žele da koriste svoju bazu, biće demonstrirano na času.

Rok za predaju projekta:

Grupa koja vežbe sluša ponedeljkom: 31. maj 2020. u 23:59:59h.

Grupa koja vežbe sluša četvrtkom: 3. jun 2020. u 23:59:59h.

Grupa koja vežbe sluša petkom: 4. jun 2020. u 23:59:59h.

Odbrane će biti u poslednjoj nedelji vežbi. Pošto je veliki broj timova, verovatno ćemo iskoristiti i redovne termine vežbi, i dodatne termine nevezano za nastavu. Raspored će biti napravljen unapred, tako da svi mogu da se organizuju i uklope.

Arhivu sa projektom nazvati prema šablonu "tim_5_bp2020" i poslati na mejl amarkovic@raf.rs, do predviđenog roka. Radi lakšeg grupisanja projekta i organizacije, neka i naslov mejla bude po šablonu: "Projekat tim_5_bp2020".

Napomena: Zakasneli i/ili prepisani projekti će biti ocenjeni sa 0 poena. Projekti koji nisu nazvani i poslati prema opisanom šablonu, neće biti ocenjeni.

Povezivanje sa bazom

Bodova: 4p.

Prilikom pokretanja projekta učitati kredencijale za komunikaciju sa bazom i obezbediti odgovarajući drajver. Neposredno nakon uspešnog uspostavljanja konekcije, u aplikaciji prikazati stablo koje reprezentuje strukturu baze na koju je konekcija uspostavljena. Struktura baze podataka se formira na osnovu podataka koji se dobijaju uz oslonac na DatabaseMetaData interfejs.

Aplikacija je zamišljena tako da može da komunicira sa proizvoljnom bazom podataka i da stablo kreira dinamički, zavisno od toga koje tabele se nalaze na serveru. Samim tim, rad aplikacije nije zavisn od domena problema koji baza podataka rešava, već je cilj da se napravi univerzalno rešenje. Baza sa kojom će aplikacija komunicirati je namenjena samo za demonstraciju.

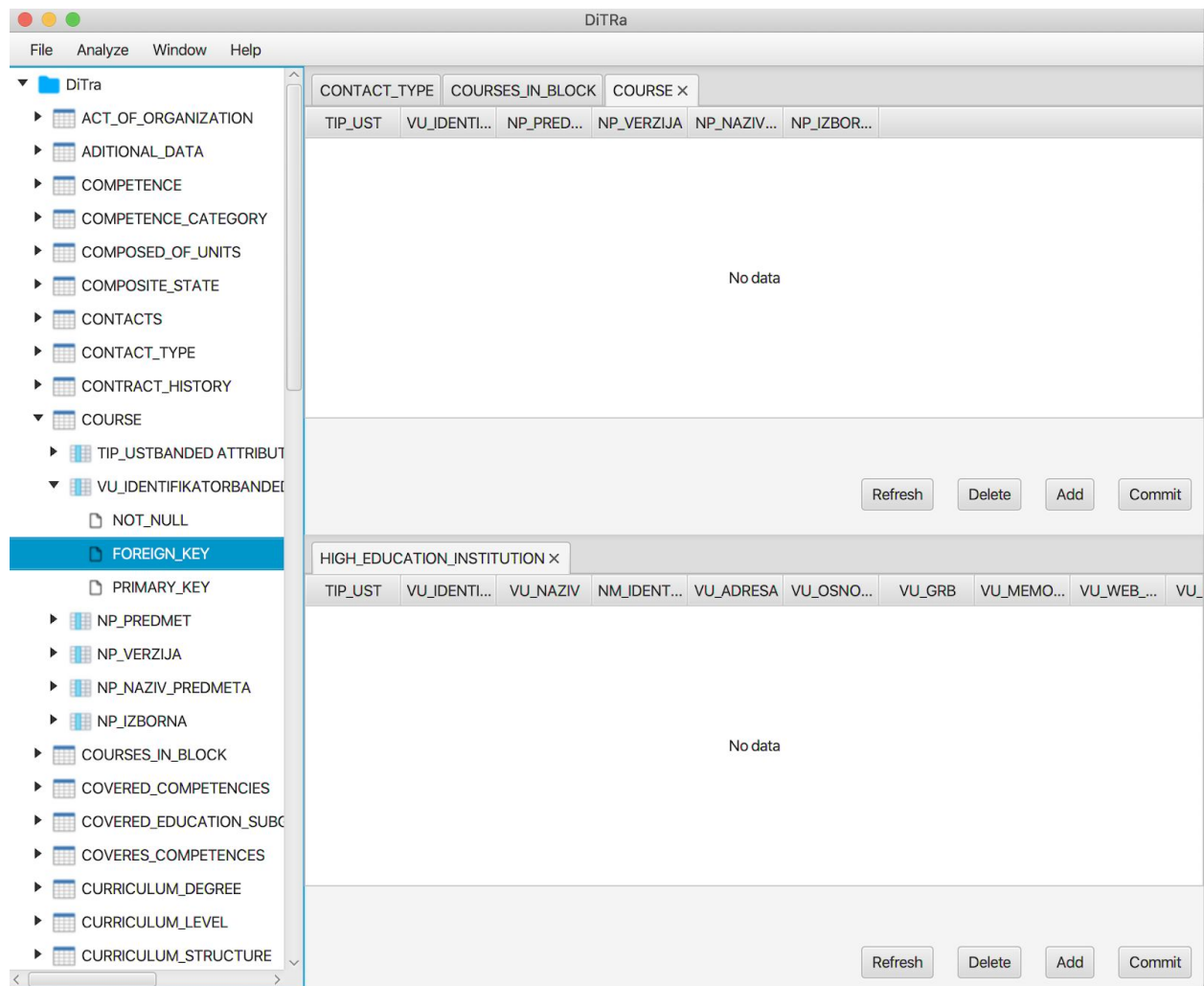
Hijerarhija stabla je sledeća:

Informacioni resurs - Entitet (Tabela) - Atribut - Ograničenje

Na slici na narednoj strani je prototip aplikacije koja se ponaša na gorepomenuti način.

Selekcija (dupli klik) čvora (Entitet/Tabela) u stablu otvara odgovarajući tab koji sadrži panel sa tabelom i toolbar-om (koji će objediniti dugmiće za pojedinačne akcije koje se mogu izvršiti nad slogovima ili celom tabelom). U panelu dole desno se otvaraju dodatni tabovi i prikazuju se tabele za sve one entitete koji su u relaciji sa selektovanim. Promena selekcije čvora u stablu ili otvorenog taba, ažurira i set otvorenih tabela koje se nalaze u panelu dole desno.

Selekcija entiteta i otvaranje njegove tabele i relacija, povlači i popunjavanje tabela podacima koji se nalazi u bazi.



CRUD

Bodova: 2p.

Obezbediti podršku za dodavanje, izmenu i brisanje slogova koji se nalaze u bazi. Na toolbar-u tabele se nalaze dugmići za **Add**, **Update** i **Delete**. Dodavanje i izmena mogu da se vršiti ili direktno u tabeli ili pomoću prozora za dijalog. Metode treba da budu realizovane korišćenjem objekta PreparedStatement. Dozvoljena je i izmena vrednosti primarnog ključa. Sve SQLException-e u slučaju neuspešnog menjanja tabele obraditi i prikazati odgovarajuće poruke kroz JOptionPane.

Filtriranje kolona i sortiranje

Bodova: 1p.

Omogućiti podršku za odabir kolona koje će biti prikazane u selektovanoj tabeli i sortiranje podataka prema zadatim kriterijumima. Na toolbar-u tabele se nalazi dugme za **Filter & Sort**. Korisnik može da odabere proizvoljan set kolona po kojima će sortirati podatke, i poredak se može razlikovati za svaku od njih.

Primer filtriranja kolona: select last_name, phone_number....

Primer sortiranja kolona: order by first_name asc, department_id desc, salary desc,...

Relacije

Bodova: 3p.

Prikaz podataka u tabelama dole desno - relacije. Nepходно je filtrirati podatke prema relacijama, i za selektovani red iz glavne tabele prikazati samo relevantne podatke iz povezanih tabela.

Primer: ako su povezane tabele Employees i Departments i glavna tabela je Employees - selekcijom jednog sloga u Employees (odabirom jednog zaposlenog) u povezanoj tabeli će se prikazati podaci samo o onom sektoru u kojem on radi.

Pretraga

Bodova: 6p.

Realizovati metodu za pretragu tabele po zadatim parametrima pretrage.

Pretraga po zadatom kriterijumu daje mogućnost korišćenja specijalnih karaktera % i _ za VARCHAR i CHAR polja, kao i operator =,>,< za NUMERIC i INTEGER polja. Tako kreirane konstrukte korisnik može da povezuje AND i OR operatorima, čime se dobija kompleksna pretraga baze.

Primer: Korisnik želi da pronađe sve zaposlene koji rade u sektoru sa šifrom 50 ili 90, broj telefona im počinje sa "555" i plata im je manja od 17000.

Metoda treba da bude realizovana korišćenjem objekta PreparedStatement. Kao rezultat rada metode u tabeli se prikazuju samo oni slogovi koji odgovaraju parametrima pretrage.

Izveštaji (COUNT i AVERAGE)

Bodova: 4p.

Realizovati metodu za generisanje izveštaja, koja dopušta korisniku da selektuje kolonu čije će vrednosti brojati ili tražiti prosek, i kolone po kojima će grupisati prikaz. Za AVG funkciju, neophodno je dopustiti izbor samo kolona koje prikazuju brojeve vrednosti za agregaciju. Grupisanje rezultata se može vršiti po proizvoljnom setu kolona koje pripadaju posmatranoj tabeli.

Primer: Želimo da vidimo prosečne ocene za predmete po godinama studija.

Implementacija

Bodova: -10p.

Implementacija uz oslonac na JDBC koncepte i Java SWING biblioteku. Obavezna upotreba MVC, Observer i Bridge šablona. Nije dozvoljeno koristiti dodatne biblioteke za rad sa bazom.

Projekat koji ne poštuje navedenu arhitekturu vredi maksimalno 10 poena.

Obratiti posebnu pažnju na upravljanje greškama i izuzecima (program ne može da puca ili izbacuje greške), promenu selektovanog taba, tipove podataka koji su očekivani u tabelama... Postojanje funkcionalnosti koje ispisuju generisane upite i/ili rezultujuće setove podataka u konzoli, bez ažuriranja grafičkog korisničkog interfejsa neće biti tretirane kao uspešno isprogramirana stavka projekta i samim tim se neće bodovati.