

Baze podataka

Drugi domaći zadatak

Računarski fakultet

22. maj 2022.



Važne napomene

1. Zakasneli i/ili prepisani projekti će biti ocenjeni sa 0 poena.
2. Neodbrani projekti će biti ocenjeni sa 0 poena.
3. Projekti koji nisu postavljeni na studentski repozitorijum biće ocenjeni sa 0 poena.
4. Projekti koji ne mogu da se učitaju/pokrenu biće ocenjeni sa 0 poena.
5. Projekte ne slati na mejl, već samo komitovati finalni kod do roka.
6. Obavezno napraviti Readme.md u kome će biti ime i prezime studenta.
7. Domaći vredi maksimalno 20 poena.
8. Svi komitovi moraju biti opisani. Svi studenti u timu moraju imati komitove. Koristiti PR i grananje na Git-u.
9. Projekat koji ne poštuje navedene šablone vredi maksimalno 15 poena.

Rok za predaju projekta:

6. jun 2022. u 12:00h.

Odbrana projekta:

Poslednja nedelja vežbi. Biće dodatnih termina. Odbrana je isključivo uživo i može da uključi dodatno programiranje.

Zadatak

Napraviti napredni, GUI-orijentisan alat koji izvršava SQL upit koji korisnik unese i rezultujući skup podataka prikazuje u tabeli. Neophodno je podržati EXEC, INSERT, UPDATE, DELETE; SELECT iskaze (sa spajanjima tabela i podupitima), kao i pravljenje uskladištenih procedura i funkcija. Alat ima nekoliko naprednih funkcija:


1. (2) **Bulk import** - korisnik može da odabere CSV fajl i tabelu u koju će importovati podatke iz njega. U ovom slučaju, SQL se generiše u pozadini.
2. (1) **Result Set export** - rezultujući skup podataka se eksportuje u CSV formatu na željenu destinaciju na računaru.
3. (3) **sql.pretty** - formatiranje SQL upita tako da su ključne reči obojene i velikim slovima i da je svaki deo iskaza u zasebnom predu. Primer:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

4. (14 = 1 + 10 + 3) **Query checker** - provera sintakse i validnost upita:

- da li kolone i tabele koje su navedene postoje u bazi
- da li je kolona odabrana za spajanje zapravo strani ključ
 - Select last_name, first_name, department_name from hr.employees e join hr.departments d on (e.department_name = d.department_name)
- da li postoje obavezni delovi upita
- da li je redosled iskaza ispravan
- da li aliasi imaju navodnike ako su više reči
- da li funkcije agregacije imaju GROUP BY ako im je potreban)
- da u WHERE iskazu nije funkcija agregacije
- da su istovetni tipovi podataka za promenljive i oni koji se selektuju u pravljenju procedure/funkcije
- da se sve definisane promenljive koriste u kodu procedure/funkcije
- da li je CSV fajl koji se učitava za Bulk Import u skladu sa strukturom izabrane tabele

Definisati konfiguracioni fajl u JSON formatu, koji sadrži ime pravila, opis greške i predlog za ispravku. Konfiguracioni fajl se učitava prilikom pokretanja aplikacije i koristi za formiranje *stack_trace*. Opis greške ne treba da bude uopšten već konkretan, npr.: "Kolona DEPARTMENT_NAME nije strani ključ za spajanje tabela EMPLOYEES i DEPARTMENTS". Isto važi i za sugestiju, npr. "Pokušajte sa DEPARTMENT_ID". Svaki upit se validira pre nego što se pošalje bazi na izvršavanje (šalje se samo ukoliko nema grešaka). Ako greške postoje, ispisuju se sve greške (*stack*) sa predlozima za korekcije. Jedini izuzetak može da bude poziv da se izvrši procedura/funkcija.



Sve 4 stavke definisati kao komponente (imaju interfejs i implementaciju, deklarisanje izvoditi isključivo preko interfejsa). Koristiti Java SWING biblioteku. Bridge šablon je obavezujući, kao i MVC+O arhitektura.

****Koristiti HR bazu podataka koju možete napraviti na svojoj timskoj bazi brisanjem starih tabela i pokretanjem skripte sa sledećeg linka:***

<https://github.com/nomemory/hr-schema-mysql/blob/master/hr-schema-mysql.sql>

*****Zajednički server:***

<http://164.92.145.191/phpmyadmin>

Kredencijale je svaki tim dobio za prethodni domaći u Readme.md.

******Hint: Uključiti Maven biblioteku za rukovanje JSON fajlom kako bi se najlakše moguće isparsirala pravila.***