

Domaći zadatak

Softverski alati baze podataka

Školska 2023/2024.

Opis sistema

Potrebno je napraviti sistem za transport paketa. U sistemu se vodi evidencija o gradovima (naziv, poštanski broj), opštinama (naziv, grad, x koordinata (u km), y koordinata (u km)). U sistemu postoje korisnici, za koje se pamti ime, prezime, korisničko ime, šifra, broj poslanih paketa. Neki korisnici su administratori. Korisnici mogu postati kuriri tako što šalju zahtev administratoru koji pored informacija o korisniku sadrži i informacije o vozilu koji bi želeo da vozi. Administrator može da prihvati ili odbije zahtev (zahtevi se ne čuvaju trajno u bazi). Za vozilo se pamte registracioni broj, tip goriva (0 - "plin", 1 - "dizel", 2 - "benzin") i potrošnja (litri po kilometru). Nakon odobrenja zahteva, korisnik postaje kurir (u svakom trenutku kola može da vozi samo jedan kurir i kurir ne može da vozi više automobila). Prethodnu funkcionalnost potrebno je uraditi kao storu proceduru. Za svakog kurira se u sistemu pamti njegovo vozilo, broj isporučenih paketa, ostvaren profit i status (0 - "ne vozi", 1 - "vozi").

Prevoz se obavlja tako što korisnik postavlja zahtev za prevozom tako što bira opštinu u kojoj se preuzima paket, opštinu u koju se dostavlja paket, tip paketa (0 - "pismo", 1 - "standardno", 2 - "lomljivo") i težinu paketa (decimalni broj) . Zatim kuriri koji su u statusu "ne vozi" daju svoje ponude za postavljen zahtev. Ponude sadrže procenat cene isporuke (na primer: 10 označava 10% cene isporuke koja se računa po formuli *CenaJedneIsporuke*), informacije o kuriru i informacije o paketu. Korisnik koji je postavio zahtev za prevozom bira jednu od ponuda. Nakon izbora ponude, sve ponude za taj zahtev se brišu (okidač ili okidači sa prefiksom *TR_TransportOffer_*). Za svaki paket se čuvaju informacije o kuriru, statusu isporuke (0 - "kreiran", 1 - "zahtev prihvaćen", 2 - "pokupljen", 3 - "isporučen"), cena i vreme prihvatanja zahteva. Vožnja se sastoji od svi odobrenih ponuda u trenutku početka vožnje. Pre početka vožnje, kurir određuje koje pakete isporučuje i zatim ih prevozi u FCFS redosledu. Po završetku vožnje, kurir računa ostvaren profit. Profit za jednu vožnju računa se kao razlika sume cena svih isporuka paketa i potrošnje goriva prilikom te vožnje (cena goriva: "plin" - 15/litar, "benzin" - 36/litar, "dizel" - 32/litar). Vožnja traje od preuzimanja prvog paketa do isporuke poslednjeg paketa.

$$CenaJedneIsporuke = (OSNOVNA_CENA[ij] + (TEŽINSKI_FAKTOR[ij] * weight) * CENA_PO_KG[ij]) * euklidska_distanca$$

	Pismo	Standardno	Lomljivo
Početna cena	10	25	75
Težinski faktor	0	1	2
Cena po KG	/	100	300

Poželjno je da onde gde je to moguće referencijalni integriteti budu: ON UPDATE CASCADE, ON DELETE NO ACTION. Iz tog razloga, metoda koja briše određeni red iz tabele ne treba da briše i ostale redove drugih tabela koje referenciraju red za brisanje. Ostale redove treba prvo eksplicitno obrisati drugim metodama (ukoliko ta metoda ne postoji koristiti referencijalni integritet ON DELETE CASCADE).

Svaka kolona Id koja je primarni ključ i nije strani ključ treba da bude IDENTITY kolona.

Koristiti tip DECIMAL(10,3) za brojeve sa zarezom. Podrazumevana maksimalna dužina svih tekstualnih kolona u tabelama je 100 karaktera, osim ukoliko za neku kolonu nije drugačije navedeno.

Zahtevi i testiranje

Potrebno je napraviti bazu podataka koja je definisana unutar sekcije “Opis sistema”.

Potrebno je napisati klase sa prefiksom **piggbbbb_** koje implementiraju sve interfejsa i unutar njih implementirati sve metode interfejsa. Klasu možete nadograditi novim metodama, poljima i drugim elementima, ali je neophodno implementirati metode interfejsa jer će one biti pozivane u testovima.

Postoje par javnih testova i nekoliko tajnih testova. Broj ostvarenih poena na domaćem zadatku formira se po sledećoj formuli:

$$\text{OsojeniPoeni} = (\text{PoeniSaJavnogTesta} + \text{PoeniSaTajnogTesta}) * \text{faktorModifikacije}$$

Domaći zadatak vredi 20 poena (javni testovi – 10 poena, tajni testovi – 10 poena, faktor modifikacije - vrednost izmedju 0 i 1).

Struktura rešenja

Potrebno je poslati zip arhivu pod nazivom **piggbbbb.zip** na stranici https://rti.etf.bg.ac.rs/domaci/index.php?servis=SAB_domaci_2324 unutar koje se nalaze:

1. klase sa prefiksom **piggbbbb_** koje implementiraju interface (treba da se nalaze unutar paketa *student*),
2. modikovanu klasa **studentMain.java**
3. ostale java klase i paketi ukoliko postoje (treba da se nalaze unutar paketa *student*),
4. **piggbbbb.erwin** i **piggbbbb.png** – model baze napravljen u alatu ERwin Data Modeler r9.7,
5. **piggbbbb.sql** – SQL fajl dobijen alatom ERwin Data Modeler + sav dodatni SQL kod (okidači, procedure, funkcije i ostali SQL kod) uključujući i okidače iz **piggbbbb-triggers.sql** i stone procedure iz **piggbbbb-stored_procedure.sql**,
6. **piggbbbb-tsql.sql** – SQL fajl sa okidačima, stonim procedurama i funkcijama,
7. **piggbbbb.bak** – backup baze podataka (proveriti kompatibilnost ovog fajla sa fakultetskom verzijom alata ili ispravnost sql fajla iz tačke 5).

Napomene

1. Gore pomenuti format za nazivanje fajlova ima sledeću definiciju: p – prvo slovo prezimena, i – prvo slovo imena, gg – poslednje dve cifre iz godine upisa, bbbb – broj indeksa
2. Za komunikaciju sa bazom možete koristiti JPA ili JDBC.
3. Obavezno je korišćenje MS SQL baze