

# **Domaći zadatak**

## **Softverski alati baze podataka**

### **Školska 2021/2022**

#### **Opis sistema**

Potrebno je napraviti sistem za kurirsku službu. U sistemu se vodi evidencija o gradovima (naziv, poštanski broj), adresama (ulica, broj, grad, x koordinata (u km), y koordinata (u km)) i lokacijama magacina (adresa). U sistemu postoje korisnici, za koje se pamti ime, prezime, korisničko ime, šifra i adresa. Korisnici sistema mogu biti kupci (obični korisnici), kuriri (imaju i sve privilegije kao i kupci) i administratori. Korisnici sistema istovremeno mogu biti i administratori i kuriri. Korisnici mogu postati kuriri tako što šalju zahtev administratoru uz informacije o broju vozačke dozvole. Administrator može da prihvati ili odbije zahtev (zahtevi se ne čuvaju trajno u bazi). Za vozila se pamte registracioni broj, tip goriva (0 – „plin“, 1 – „dizel“, 2 – „benzin“), potrošnja (litri po kilometru) i nosivost. Nakon odobrenja zahteva, korisnik postaje kurir. Prilikom planiranja vožnje, kurir preuzima vozilo koje će služiti za tu vožnju. U svakom trenutku vozilo može da vozi samo jedan kurir i kurir ne može da vozi više vozila (ovo ograničenje potrebno je implementirati preko Transact-SQL). Za svakog kurira se u sistemu pamti vozilo koje on trenutno vozi, informacije o svim prethodnim vozilima koja je koristio, broj isporučenih paketa, ostvaren profit i status (0 – „ne vozi“, 1 – „vozi“).

Isporuka paketa se obavlja tako što korisnik kreira zahtev biranjem adrese sa koje se preuzima paket, adrese na u koju se dostavlja paket, tip paketa (0 – „mali paket“, 1 – „standardni paket“, 2 – „paket nestandardnih dimenzija“, 3 – „lomljiv paket“) i težinu paketa (decimalni broj). Ponuda se kreira automatski po pristizanju zahteva (okidač ili okidači sa prefiksom **TR\_TransportOffer**) i sadrži cenu isporuke. Ponuda može biti prihvaćena ili odbijanja. Pre nego što se ponuda prihvati ili odbije, moguća je izmena informacija o paketu. Izmjena informacija o paketu dovodi i do izmene cene paketa. Po prihvatanju ponude, kurir može raditi isporuku paketa. Za svaki paket se čuvaju informacije o statusu isporuke (0 – „zahtev kreiran“, 1 – „prihvaćena ponuda“, 2 – „paket preuzet“, 3 – „isporučen“, 4 – „ponuda odbijena“), ceni, vremenu kreiranja zahteva, vremenu prihvatanja ponude i lokaciji na kojoj se paket trenutno nalazi ukoliko je on u statusu 2.

Vožnja se započinje preuzimanjem slobodnog vozila koje se nalazi u gradu u kome se stanuje vozač. Slobodna vozila su parkirana u magacinima. Smatrati da vozači žive isključivo u mestima gde postoje magacini. Pre početka vožnje, kurir pravi plan preuzimanja i isporuke paketa. Kurir preuzima sve pakete iz grada u kome se nalazi ili to radi u redosledu u kome su zahtevi bili prihvaćeni. Ukoliko nosivost vozila to ne dozvoljava, on će preuzeti maksimalno onoliko koliko može. Nakon toga odlazi do magacina i preuzima pakete iz magacina do maksimalne nosivosti vozila. Prioritet imaju ranije prihvaćene porudžbine. Nakon preuzimanja paketa iz grada u kome se nalazi, počinje faza isporuke paketa. Isporuka se vrši tako što se svaki put bira najbliža lokacija paketa koji je potrebno isporučiti u odnosu na lokaciju gde se vozač trenutno nalazi. Tokom ovog postupka, za svaki grad u koji se ide se proverava da li postoje paketi za preuzimanje iz tog grada. Ukoliko takvi paketi postoje, oni se preuzimaju po istom principu kao i paketi iz polaznog grada. Ovi paketi neće biti isporučeni tokom ove vožnje, već će na kraju vožnje biti transportovani do magacina iz koga je vožnja i krenula.

Vožnja se završava povratkom u magazin i tada se računa ostvaren profit na osnovu te vožnje. Profit se računa kao razlika sume cena svih isporučenih paketa i potrošnje goriva prilikom te vožnje (cena goriva: „plin“ - 15/litar, „benzin“ - 36/litar, „dizel“ - 32/litar). Vožnja traje od preuzimanja vozila iz magacina do povratka vozila u magazin. Sve distance se računaju kao euklidske distance.

$$CenaIsporuke = (OSNOVNA\_CENA[i] + weight * CENA\_PO\_KG[i]) * euklidska\_distanca$$

	Mali paket	Standardni paket	Paket nestandardnih dimenzija	Lomljiv pakte
Početna cena	115	175	250	350
Cena po KG	/	100	100	500

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:

$$OsojeniPoeni = (PoeniSaJavnogTesta + PoeniSaTajnogTesta) * faktorModifikacije$$

Domaći zadatak vredi 20 poena (javni testovi – 10 poena, tajni testovi – 10 poena, faktor modifikacije - vrednost između 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\\_1920](https://rti.etf.bg.ac.rs/domaci/index.php?servis=SAB_domaci_1920) unutar koje se nalaze:

1. klase sa prefiksom **piggbbbb\_** koje implementiraju interface (treba da se nalaze unutar paketa *rs.etf.sab.student*),
2. modifikovana klasa **studentMain.java**
3. ostale java klase i paketi ukoliko postoje (treba da se nalaze unutar paketa *rs.etf.sab.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, stone procedure i funkcije iz **piggbbbb-tsql.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