

## Baze podataka 2 - II popravni kolokvijum

14.01.2016.

Data je baza koja prikazuje podatke o poslovanju jedne knjižare. U bazi se nalaze sledeće tabele:

**knjige** (sifra, naslov, autor, zanr, cena, stanje)

**citaoci**(brClanskeKarte, ime, adresa, kontakt, popust)

**prodaja**(sifra,sifraKnjige,brCKCitaoca,vremeProdaje,obradjeno)

U tabeli **knjige** se nalaze podaci o knjigama koje su trenutno u prodaji, kao i trenutna količina tih knjiga u knjižari (kolona **stanje**). Knjižara ima otvoren klub čitalaca. Preko kartice kluba čitalaca, svaki čitalac može da ostvari izvesni popust na kupovinu knjiga. Podaci o članovima kluba čitalaca se čuvaju u tabeli **citaoci**. Tabela **prodaja** čuva podatke o svakoj ostvarenoj prodaji u knjižari. U tabelu se ubacuje nova vrsta prilikom svakog očitavanja na kasi. Ako kupovinu ostvaruje član kluba, kolona **brCKCitaoca** sadrži broj članske karte tog čitaoca. U suprotnom, kolona **brCKCitaoca** ima vrednost NULL. Kolona **obradjeno** ima vrednost "da" ako je izvršen presek od trenutka prodaje knjige, u suprotnom ima vrednost "ne".

Date su sledeće storne procedure:

- **sp\_novaProdaja (sifraKnjige, brCKCitaoca)** – Ova procedura se poziva prilikom prodaje knjige, odnosno prilikom očitavanja na kasi. Unosi se nova vrsta u tabelu **prodaja**, i skida se jedan primerak knjige sa stanja. Vrednost kolone **obradjeno** se postavlja na „ne“.
- **sp\_presek** – Ova procedura se poziva kada se vrši popis u knjižari. Računa se prihod od svih knjiga prodatih od poslednjeg preseka. Za sve takve knjige se vrednost kolone **obradjeno** postavlja na „da“. Vodi se računa da je članovima kluba cena smanjena za njihov popust.
- **sp\_brojKupljenihKnjiga(granica)** – Za svakog člana kluba koji je kupio bar onoliko knjiga koliko je prosleđeno parametrom **granica**, ispisuje se broj članske karte, ime čitaoca i broj kupljenih knjiga.
- **sp\_knjigeNajveciPopust** – Ova procedura izlistava sve knjige koje je kupio čitalac sa trenutno najvećim popustom.

Ubrzati izvršavanje zadatih upita kreiranjem odgovarajućih indeksa tako da se postigne što manje zbirno vreme izvršavanja. Imati u vidu da se procedura **sp\_novaProdaja** izvršava 100 puta češće od drugih procedura. Ostale procedure se izvršavaju podjednako često. Voditi računa da u svakoj tabeli količina memorije upotrebljena za indekse bude što manja, a da se to ne odrazi na postignuto ubrzanje. Efikasnost upotrebe memorije donosi dodatne bodove. Neracionalna upotreba memorije ne donosi bodove.

Rešenje zadatka treba da sadrži:

- a) SQL skriptu sa komandama za kreiranje potrebnih indeksa i ključeva nad tabelama;
- b) planove izvršavanja svih upita pre i posle upotrebe indeksa (za svaku proceduru/upit ponaosob);
- c) komentare (2-3 rečenice) koji treba da pojasne dobijeno ubrzanje za svaki plan izvršavanja.

**Napomena: Vreme rada je 60 minuta.**

