**Zadaci**

**Zadatak 1. Deadlock**

Izazvati deadlock pri konkurentnom izvršavanju dve transakcije.

**Zadatak 2.**

Date su tabele:

**klijenti(id,ime,iznos)
prenosi(id,ko,kome,iznos)**

Tabela **klijenti** čuva podatke o klijentima banke i o iznosima na njihovim računima.

Popuniti tabelu **klijenti** random podacima.

Tabela **prenosi** čuva podatke o prenosima novca sa računa jednog klijenta na račun drugog.

Kada god klijenti zatraže prenos novca, upisuje se nova vrsta u tabelu **prenosi**, ali se sam prenos ne izvršava odmah. Kreirati stornu proceduru **sp\_unosPrenosa** koja ubacuje novi prenos u okviru transakcije. Ova procedura se u proseku vrši na svakih 1s.

Kada se nakupi izvestan broj novih prenosa novca u tabeli **prenosi**, oni se obrade i ažuriraju se iznosi na računima u tabeli **klijenti**. Kreirati stornu proceduru **sp\_obradaPrenosa** koja u okviru transakcije obrađuje nove prenose tako što ih učitava u kursor, za svaki prenos ažurira iznos na računu pošiljaoca i na računu primaoca i zatim briše obrađeni prenos iz tabele **prenosi**. Ova procedura se u proseku izvršava na svakih 1min.

Dodatno – U proceduri **sp\_obradaPrenosa**, obrađivati 5 po 5 prenosa, i posle svakih 5 napraviti savepoint. U slučaju greške u izvršavanu transakcije treba izvršiti rollback na najbliži savepoint.

Postaviti odgovarajuće nivoe izolacije.