

# Transakcije

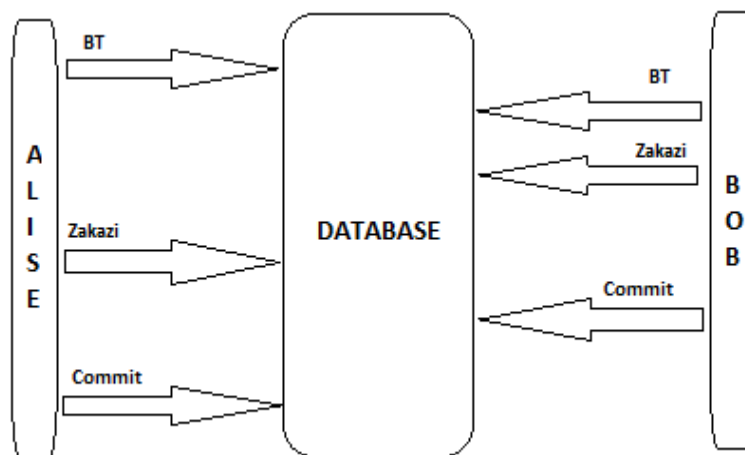
Konfliktne situacije koje su rjesavane:

- a. Vise pacijenata zeli da rezervise unaprijed definisani pregled.
- b. Vise pacijenata zeli da rezervise pregled (u istom terminu, kod istog doktora).
- c. Prilikom pretrage klinika, moguće je da administrator ubaci novu ili izbrise istu, sto rezultuje pogresnim podacima.
- d. Prilikom pretrage ljekara, moguće je da administrator ubaci novog ili izbrise istog, sto rezultuje pogresnim podacima.

Rjesenja:

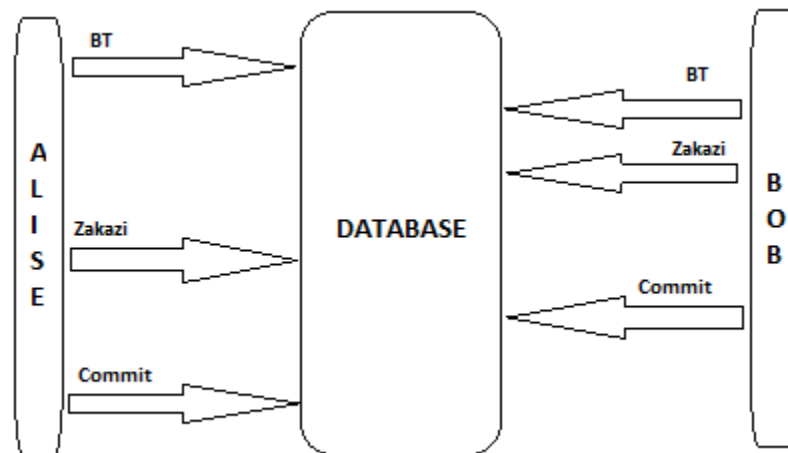
- a) Koristeci nivo izolacije SERIALIZABLE, eliminisan je "phantom read" problem. Ukoliko neki korisnik zakaze unaprijed definisani pregled, drugi korisnik je ucitao sve dostupne preglede, prije nego je rezervisan. Bice mu onemoguceno zakazivanje uz povratnu informaciju.

Crtez toka:



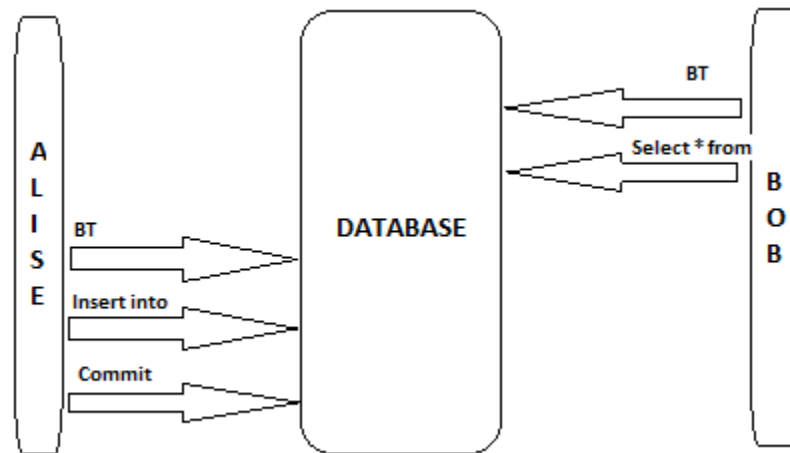
b) Slicna situacija kao kod problema pod a. Koristeci nivo izolacije SERIALIZABLE, eliminisan je "phantom read" problem. Ukoliko dva korisnika zakazu termin pregleda kod istih ljekara i u isto vrijeme, zakazivanje ce biti onemoguceno uz povratnu informaciju.

Crtez toka:



c) Kod pretrage klinika, moguca konfliktna situacija je da administrator klinike u toku pretrage, doda ili obrise kliniku. Time bi korisnik koji salje upit, dobio ispravan odgovor. Drugi korisnik doda novi red koji zadovoljava kriterijum pretrage. Prvi korisnik bi dobio pogresne podatke ukoliko bi izvrsio isti upit. Klasican "phantom read" problem, koji bi se mogao rijesiti optimistickim zakljucavanjem baze podataka. Ideja navedenog zakljucavanja nije u zakljucavanju resursa, vec svaka operacija prije izmjene, treba da provjeri da li je u medjuvremenu neko drugi mijenjao. Ako su podaci mijenjani, prijavi se greska korisniku .

Crtez toka:



d) Situacija identicna situaciji za pretragu klinika. Korisnik pri pretrazi ljekara dobije gresku, ukoliko je u medjuvremenu dodat ili obrisan ljekar. Rjesenje je optimisticko zakljucavanje, opisano u prethodnoj situaciji.

Crtez toka:

