

# Konkurentni pristup resursima u bazi

Student: Bojana Todorović IN33/2017

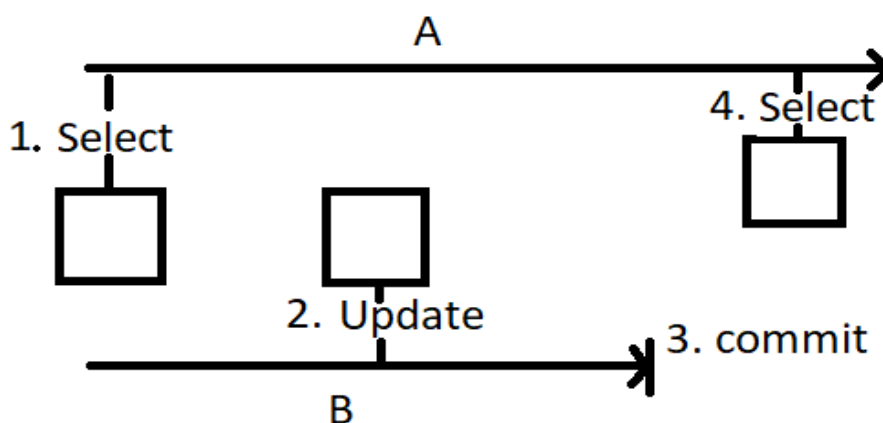
## Konfliktna situacija – na jednu žalbu može da odgovori samo jedan administrator sistema

Opis konfliktne situacije – Jedan administrator sistema je počeo da piše odgovor na žalbu, odnosno počeo je sa update-om torke u tabeli, dok u tom istom trenutku drugi administrator sistema pokušava da pristupi istoj torki.

Endpoint koji se gađa prilikom odgovaranja na žalbu -  
<http://localhost:8081/myPharmacy/complaints/sendResponse/{complaintId}>

Konfliktna situacija je rešena uz pomoć nivoa izolacije *Repeatable read* koja onemogućava sistem adminu da selektuje određenu torku u trenutku kada je drugi sistem admin update-uje.

```
48 @Override
49 @Transactional(isolation = Isolation.REPEATABLE_READ)
50 public Complaint updateComplaint(Complaint complaint) {
51     Complaint c = this.complaintRepository.findOne(complaint.getId())
```



Problem – Unrepeatable read

Nijedna druga transakcija ne može da menja podatke koji su pročitani od strane neke druge transakcije, dokle god se ta transakcija ne završi.

Konfliktna situacija – Prilikom izdavanja eRecepta izdaju se svi ili nijedan lek, a stanje u apoteci se ažurira

```
@Override
public List<Pharmacy> findByPatientDrugs(List<Long> drugs) {
    List<Pharmacy> pharmacies = this.pharmacyRepository.findAll();
    List<Pharmacy> toShow = new ArrayList<>();
    List<Pharmacy> toShowFinal = new ArrayList<>();
    for (Pharmacy pharmacy : pharmacies) {
        var a : Set<Long> = pharmacy.getDrugsQuantity().keySet();

        Set<Long> b = new HashSet<>();
        for (var x : drugs)
            b.add(x);
        var c : boolean = a.containsAll(b);

        if (c) {
            toShow.add(pharmacy);
        }
    }
}
```

Korisniku se nakon učitavanja QR koda prikazuje lista samo onih apoteka u kojima su svi rezervisani lekovi na stanju.

Endpoint koji se gađa prilikom kupovine lekova prepisanih eReceptom –

```
@PostMapping("/buyDrugs/{id}/{drugCodes}")
```

```
@Override
@Transactional(isolation = Isolation.REPEATABLE_READ)
public Pharmacy update(Pharmacy pharmacy) {
```

Konfliktna situacija – prilikom prikaza apoteka koje na stanju imaju neophodne lekove, može doći do situacije da je u tom trenutku neki drugi korisnik kupio neki od tih lekova i on više nije na stanju.

Konfliktna situacija je rešena uz pomoć nivoa izolacije *Repeatable read* koja onemogućava korisniku da selektuje određenu torku u trenutku kada je ona u procesu update-a.

Konfliktna situacija – Dva sistem admina pokušavaju u isto vreme da definišu loyalty program

```
@Override
@Transactional(isolation = Isolation.REPEATABLE_READ)
public LoyaltyProgram updateLP(LoyaltyProgram loyaltyProgram) throws Exception {
```

Konfliktna situacija je rešena uz pomoć nivoa izolacije *Repeatable read* koja onemogućava sistem adminu da menja loyalty program u trenutku kada je on u procesu update-a od strane drugog sistem admina.

Endpoint koji se gađa prilikom update-ovanja –  
<http://localhost:8081/myPharmacy/loyaltyProgram/updateLoyaltyProgram>