

# **OBLIKOVANJE BAZA PODATAKA**

## **Vježbe 11**

# Transakcije - uvod

- **Transakcija** (engl. *transaction*) predstavlja količinu posla koju baza treba odraditi kao cjelinu
- Postoje dva tipa transakcija:
  - Transakcije na nivou jedne naredbe
    - Svaka SQL naredba je transakcija sama po sebi
      - Ako zadamo DELETE za 5 redaka, ne može se desiti da se 2 retka obrišu, a 3 ne
  - Eksplicitne transakcije koje obuhvaćaju proizvoljan broj naredbi
    - Korisnik zadaje početak i kraj transakcije

# Korisnost transakcija

- Transakcije rješavaju problem da je baza uvijek u **ispravnom (konzistentnom)** stanju
  - Svaka naredba se mora izvršiti do kraja ili uopće ne
    - Radimo UPDATE na 20 redaka
  - Ponekad treba osigurati da se niz SQL naredbi odradi kompletan ili da se ne odradi niti jedna naredba iz niza
    - Želimo nekome prebaciti 1.000 kuna na račun:
      - UPDATE koji skida 1.000 kuna s našeg računa
      - UPDATE koji stavlja 1.000 kuna na odredišni račun

# Transakcijska datoteka

- Transakcijska datoteka omogućava postojanje transakcija
- Svaka relacijska baza se obavezno sastoji od **barem jedne transakcijske datoteke**
  - Sadržava povijest akcija rađenih nad bazom
    - Svaka akcija nad bazom se zapisuje u transakcijsku datoteku zajedno s podacima na koje je utjecala
  - Kod SQL Servera se ta datoteka naziva i **transakcijski zapisnik** (engl. *transactional log*) i ima ekstenziju **.ldf**
- Zbog performansi je dobro transakcijsku datoteku držati na drugom fizičkom disku
  - Na taj način se mogu paralelno zapisivati podaci i transakcijske aktivnosti

# Važni trenuci u životu jedne transakcije

- **Početak** transakcije
- **Kraj** transakcije
  - **Potvrda** transakcije
    - Sve operacije koje čine transakciju su **uspješno** obavljene
  - **Odustajanje** od transakcije
    - Desila se jedna ili više grešaka i transakcija je **neuspješna**
    - Stanje baze vraća se na stanje prije početka transakcije
- Opcionalno postavljanje **kontrolne točke** (engl. *savepoint*)

# T-SQL sintaksa za upravljanje transakcijama

- Početak transakcije:  
**BEGIN TRAN**[SACTION]
- Potvrda transakcije:  
**COMMIT TRAN**[SACTION] (ili samo **COMMIT**)
- Odustajanje od transakcije:  
**ROLLBACK TRAN**[SACTION] (ili samo **ROLLBACK**)

# T-SQL sintaksa za upravljanje transakcijama

- Dodatno, za rad s kontrolnim točkama koristimo:

- Postavljanje kontrolne točke:

**SAVE TRAN**[SACTION] *naziv\_kontr\_točke*

- Vraćanje na kontrolnu točku:

**ROLLBACK TRAN**[SACTION] *naziv\_kontr\_točke*

# Primjer transakcije

- Svaka transakcija se odnosi na konekciju na kojoj je zadana
  - Otvorimo SSMS, napravimo novu konekciju na neku bazu i zadamo naredbe:



- Nakon kraja transakcije u bazi će ostati promjene napravljene naredbama 1, 2 i 4
- Učinak naredbe 3 je poništen vraćanjem na kontrolnu točku naziva spa



# Sistemska funkcija @@TRANCOUNT

- Sistemska funkciju **@@TRANCOUNT** koristimo za provjeru jesmo li unutar transakcije
- Način rada:
  - **BEGIN TRAN** povećava **@@TRANCOUNT** za 1
  - **COMMIT TRAN** smanjuje **@@TRANCOUNT** za 1
  - **ROLLBACK TRAN** smanjuje **@@TRANCOUNT** na 0
- Povratak na kontrolnu točku ne mijenja **@@TRANCOUNT**

# Transakcije (1)

Napravite tablicu Osoba s IDOsoba (PK i IDENTITY), Ime i Prezime.

1. U transakciji umetnite 3 zapisa u Osoba i na kraju odustanite od transakcije.
2. U transakciji umetnite 3 zapisa u Osoba i na kraju potvrdite transakciju.
3. U transakciji umetnite 1 zapis u Osoba i postavite kontrolnu točku. Umetnite još 1 zapis. Na kraju odustanite od transakcije.
4. U transakciji umetnite 1 zapis u Osoba i postavite kontrolnu točku. Umetnite još 1 zapis. Na kraju potvrdite transakciju.

# Transakcije (2)

5. U transakciji umetnite 1 zapis u Osoba i postavite kontrolnu točku. Umetnite još 1 zapis i postavite kontrolnu točku. Na kraju odustanite od transakcije.
6. U transakciji umetnite 1 zapis u Osoba i postavite kontrolnu točku. Umetnite još 1 zapis i postavite kontrolnu točku. Na kraju potvrdite transakciju.
7. U transakciji umetnite 1 zapis u Osoba i postavite kontrolnu točku. Umetnite još 1 zapis i vratite se na kontrolnu točku. Na kraju odustanite od transakcije.
8. U transakciji umetnite 1 zapis u Osoba i postavite kontrolnu točku. Umetnite još 1 zapis i vratite se na kontrolnu točku. Na kraju potvrdite transakciju.

# Transakcije (3)

9. Napišite proceduru za brisanje države. Neka procedura prima 1 parametar, IDDrzava. Transakciju vodite izvan procedure. Ispišite uspjeh ili neuspjeh.
  - Pozovite 3 puta proceduru s vrijednostima parametara jednakim 50, 51 i 52.
  - Pozovite 3 puta proceduru s vrijednostima parametara jednakim 50, 51 i 1.
10. Napišite proceduru s 3 parametra za brisanje triju država. Transakciju vodite unutar procedure. Ispišite uspjeh ili neuspjeh.
  - Pozovite proceduru s vrijednostima 50, 51 i 52.
  - Pozovite proceduru s vrijednostima 50, 51 i 1.
11. Unutar vanjske transakcije pozovite prethodnu proceduru s vrijednostima 50, 51 i 52. Nakon toga odustanite od vanjske transakcije. Ispišite uspjeh ili neuspjeh. Je li brisanje napravljeno?

# Problemi kod konkurentnosti (1/2)

- Postoje tri osnovna problema koji mogu nastati kad više korisnika pristupa istom podatku:
  - Prljavo čitanje (engl. *dirty read*)
    1. Prva transakcija promijeni vrijednost retka
    2. Druga transakcija pročita tu vrijednost – prljavo čitanje
    3. Prva transakcija odustane

# Problemi kod konkurentnosti (1/2)

## – Neponovljivo čitanje (engl. *non-repeatable read*)

1. Prva transakcija pročitava vrijednost stupca u nekom retku – neponovljivo čitanje
2. Druga transakcija promijeni tu vrijednost i potvrdi se
3. Prva transakcija istim SELECT-om sad čita drukčiju vrijednost

# Problemi kod konkurentnosti (2/2)

## – Fantom (engl. *phantom*)

1. Prva transakcija zada SELECT koji vrati  $n$  redaka
2. Druga transakcija umetne redak u raspon
3. Prva transakcija zada isti SELECT koji sad vrati  $n + 1$  redaka – fantom

# Izolacijski nivoi

- Podrazumijevano je aktivan izolacijski nivo READ COMMITTED
- Nivoi za konekcije postavljaju se na sljedeći način:

SET TRANSACTION ISOLATION LEVEL **READ COMMITTED**

SET TRANSACTION ISOLATION LEVEL **READ UNCOMMITTED**

SET TRANSACTION ISOLATION LEVEL **REPEATABLE READ**

SET TRANSACTION ISOLATION LEVEL **SERIALIZABLE**



# Izolacijski nivoi

Kod rješavanja zadatka s transakcijama potrebne su **dvije SQL skripte**, svaka u svojoj konekciji te svaka SQL naredba treba sadržavati **redni broj izvršavanja**

12. Demonstrirajte problem prljavog čitanja na tablici Drzava na retku s IDDrzava = 3. Napišite gdje se desilo prljavo čitanje.
13. Demonstrirajte rješenje problem prljavog čitanja na tablici Drzava na retku s IDDrzava = 3. Napišite gdje bi se desilo prljavo čitanje i kako ste to spriječili.

# Izolacijski nivoi

14. Demonstrirajte problem neponovljivog čitanja na tablici Drzava na retku s  $IDDrzava = 3$ . Napišite gdje se desilo neponovljivo čitanje.
15. Demonstrirajte rješenje problema neponovljivog čitanja na tablici Drzava na retku s  $IDDrzava = 3$ . Napišite gdje bi se desilo neponovljivo čitanje.
16. Demonstrirajte problem fantoma na tablici Drzava na recima s nazivima koji započinju s 'H'. Napišite gdje se pojavljuju fantomi.
17. Demonstrirajte rješenje problema fantoma na tablici Drzava na recima s nazivima koji započinju s 'H'. Napišite gdje bi se pojavili fantomi.