OBLIKOVANJE BAZA PODATAKA

Predavanje 04

Blic

- https://bit.ly/3MBHZoG

Uvod u procedure

ALGEBRA BERNAYS SVEUČILIŠTE

Uvod

- Do sada smo upoznali tablice, poglede i okidače:
  - Tablice čuvaju podatke
  - Pogledi daju razinu apstrakcije nad tablicama
  - Okidače pokreće RDBMS nakon definiranog događaja
- Pogledi se primarno koriste za dohvaćanje podataka
- Što s implementacijom poslovnih pravila koja uključuju izmjene podataka?
  - Primjerice, kako možemo osigurati da korisnik umetanjem računa umetne i barem jednu stavku?
- Potrebna nam je nova vrsta objekata

ALGEBRA BERNAYS SVEUČILIŠTE

Procedure

- **Procedure** (engl. *stored procedures*) su objekti u bazi podataka koji sadržavaju SQL naredbe
  - Svaka procedura se može sastojati od proizvoljnog broja i tipa SQL naredbi

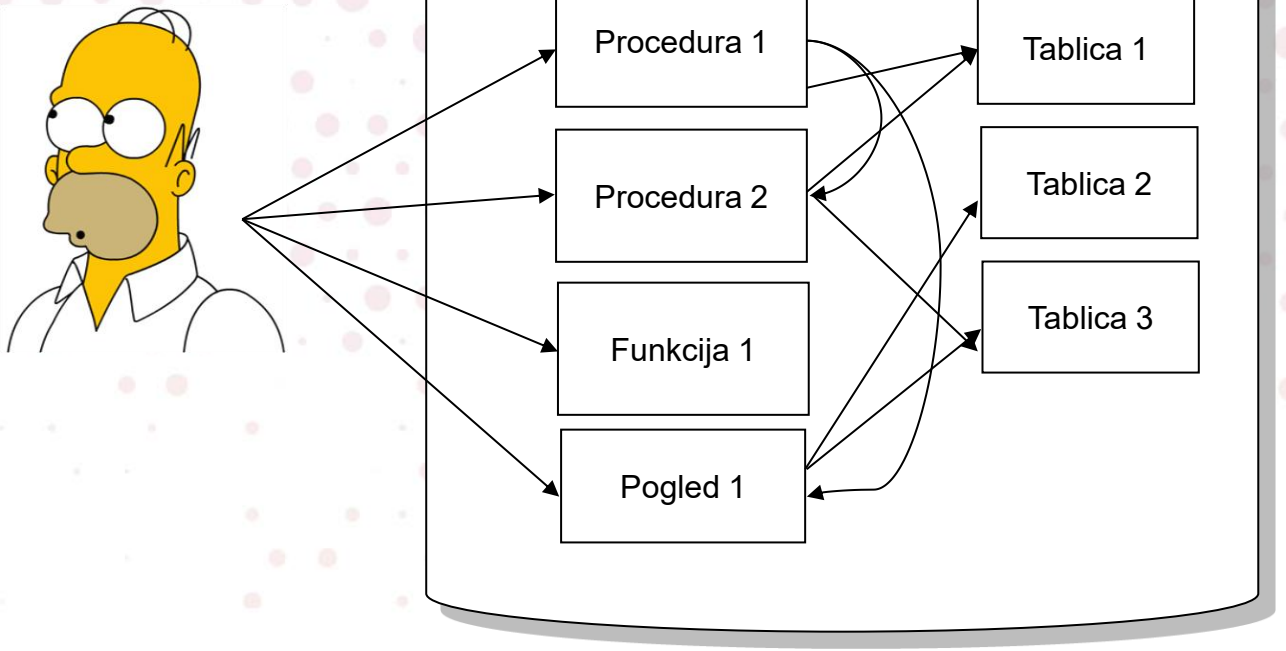
- Služe organiziranju operacija nad podacima u upravljive cjeline
  - Korisnik koristi proceduru po principu crne kutije (svrha, ulaz, izlaz)
  - Ekvivalentne funkcijama (metodama) u programskim jezicima

- Kao i svim objektima u bazi podataka, i procedurama se mogu postavljati prava pristupa

ALGEBRA BERNAYS SVEUČILIŠTE

Uloga procedura u korištenju baza podataka

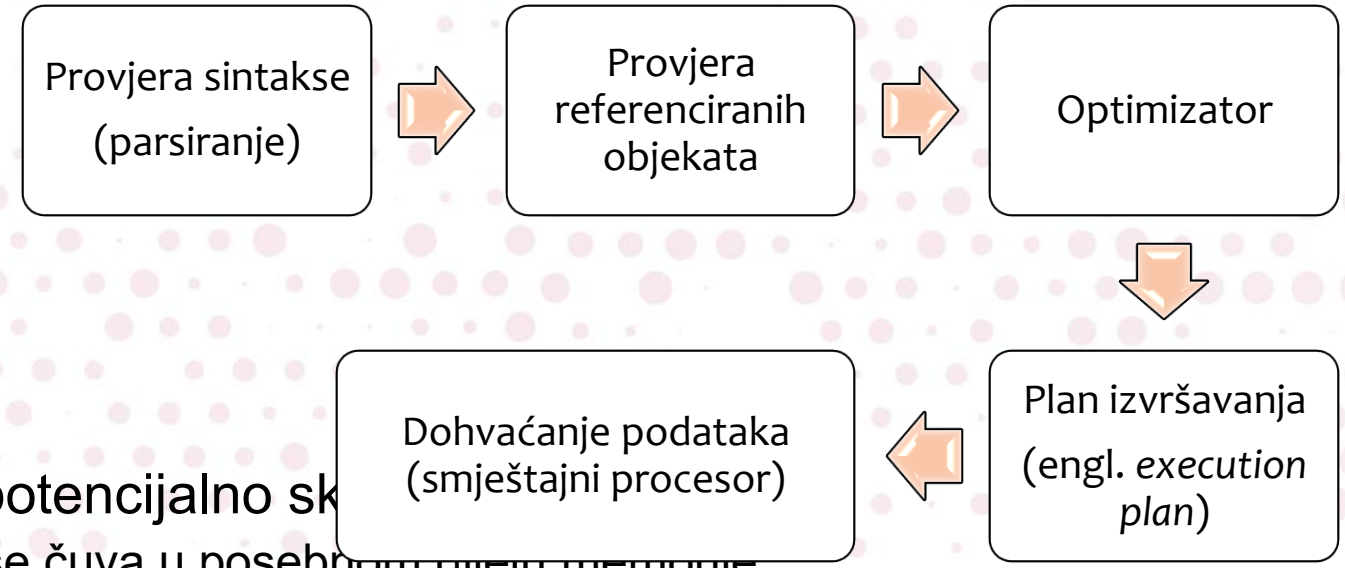
- Mogući način korištenja baze podataka:
  - Korisnik smije koristiti samo procedure, funkcije i poglede
  - Tablicama nema pristupa



ALGEBRA BERNAYS SVEUČILIŠTE

Plan izvršavanja

- Procesiranje svakog DML upita



- Izrada plana izvršavanja je potencijalno sk
  - Napravljeni plan izvršavanja se čuva u posebnom dijelu memorije

ALGEBRA BERNAYS SVEUČILIŠTE

Procedure i planovi izvršavanja

- Napravljeni plan izvršavanja se čuva u memoriji, ali kako RDBMS zna za koji upit ga može opet iskoristiti?
  - Ad-hoc upiti
    - Napravljeni plan se ponovno koristi samo za **identične upite**
    - Promjena vrijednosti parametra ili dodavanje jednog razmaka ili promjena veliko u malo slovo uzrokuje izradu novog plana
  - Procedure
    - Plan se ponovno koristi prema nazivu procedure
    - Inicijalnim pokretanjem procedure napravi se plan i smjesti u memoriju
    - Svakim sljedećim pokretanjem iz memorije se uzima plan prema njenom nazivu i iskorištava se

ALGEBRA BERNAYS SVEUČILIŠTE

DEMO

- Pogledajmo planove izvršavanja:

```
select
   qs.creation_time,
   qs.sql_handle,
   qs.plan_handle,
   qs.execution_count,
   st.text,
   ph.query_plan
from sys.dm_exec_query_stats qs
cross apply sys.dm_exec_sql_text(qs.sql_handle) st
cross apply sys.dm_exec_query_plan(qs.plan_handle) ph
order by qs.last_execution_time desc
```

ALGEBRA BERNAYS SVEUČILIŠTE

Izrada i korištenje procedura

ALGEBRA BERNAYS SVEUČILIŠTE

Sadržaj procedure

- Procedura može sadržavati proizvoljan broj SQL naredbi
  - Može sadržavati bilo koju kombinaciju `SELECT`, `INSERT`, `UPDATE` i `DELETE` naredbi (i još puno više)
    - Primjer procedure: dbo.GetUpisani
  - Pri korištenju SELECT naredbi u proceduri vrijedi:
    - Svaka `SELECT` naredba vraća jedan skup podataka (engl. *resultset*)
      - Dakle, procedura s više SELECT naredbi će vratiti **više skupova podataka**
    - Skup podataka koje vraća procedura ne može biti izvor podataka za vanjske `SELECT` naredbe (pa čak ni kad vraća samo jedan skup)
    - SELECT naredba koja dodjeljuje vrijednosti varijablama neće vratiti skup podataka

ALGEBRA BERNAYS SVEUČILIŠTE

Ulazi i izlazi iz procedura

- Procedura može imati **nula ili više** ulaznih parametara

- Izlazne vrijednosti mogu biti:
  - **Proizvoljan broj** skupova podataka kao rezultat `SELECT` naredbi
  - **Proizvoljan broj** izlaznih parametara
  - **Jedan ili nijedan** RETURN parametar

- Moguće su bilo koje kombinacije ulaznih i izlaznih opcija

ALGEBRA BERNAYS SVEUČILIŠTE

Procedure bez parametara

- T-SQL za kreiranje procedura bez parametara:

```
CREATE PROC[EDURE] shema.naziv
AS
sql_naredbe
```

- Za izmjenu procedure se koristi ALTER:

```
ALTER PROC[EDURE] shema.naziv
AS
sql_naredbe
```

- Za uklanjanje procedure koristi se DROP:

```
DROP PROC[EDURE] shema.naziv
```

- Proceduru koristimo (pozivamo):

```
EXEC[UTE] shema.naziv
```

ALGEBRA BERNAYS SVEUČILIŠTE

Primjeri

1. Napišite proceduru koja dohvaća sve retke iz tablice Kupac. Pozovite proceduru. Promijenite proceduru tako da vraća rezultate poredane po imenu pa po prezimenu. Uklonite proceduru.

2. Napišite proceduru koja dohvaća prvih 10 redaka iz tablice Proizvod, prvih 5 redaka iz tablice KreditnaKartica i zadnja 3 retka iz tablice Racun. Pozovite proceduru. Uklonite proceduru.

ALGEBRA BERNAYS SVEUČILIŠTE

Procedure s ulaznim parametrima

- T-SQL za kreiranje procedure s ulaznim parametrima:

```
CREATE PROC[EDURE] shema.naziv
 @p1 tip, ..., @pn tip
AS
sql_naredbe
```

Nema zagrada
(ali mogu biti)

- Dva načina korištenja procedure s parametrima:
  - Parametre prosljeđujemo **po redoslijedu**:

```
EXEC shema.naziv v1, ..., vn
```

  - Parametre prosljeđujemo **po nazivu**:

```
EXEC shema.naziv @p1 = v1, ..., @pn = vn
```

  - Kod prosljeđivanja po nazivu redoslijed nije bitan
  - U oba načina ne smijemo koristiti zagrade

Nema zagrada
(i ne smiju biti)

ALGEBRA BERNAYS SVEUČILIŠTE

Primjeri

3. Napišite proceduru koja prima @ID proizvoda i vraća samo taj proizvod iz tablice Proizvod. Pozovite proceduru na oba načina. Uklonite proceduru.

4. Napišite proceduru koja prima dvije cijene i vraća nazive i cijene svih proizvoda čija cijena je u zadanom rasponu. Pozovite proceduru na oba načina. Uklonite proceduru.

5. Napišite proceduru koja prima četiri parametra potrebna za unos nove kreditne kartice. Neka procedura napravi novi zapis u KreditnaKartica. Neka procedura prije i nakon umetanja dohvati broj zapisa u tablici. Pozovite proceduru na oba načina. Uklonite proceduru.

6. Napišite proceduru koja prima tri boje i za svaku boju vraća proizvode u njoj. Pozovite proceduru i nakon toga je uklonite.

ALGEBRA BERNAYS SVEUČILIŠTE

Procedure s izlaznim parametrima

- T-SQL za kreiranje procedure s izlaznim parametrima:

```
CREATE PROC[EDURE] shema.naziv
        @p1 tip OUTPUT, ..., @pn tip OUTPUT
AS
    sql_naredbe
```

- Parametri koje prosljeđujemo moraju biti definirani kao varijable kako bi procedura mogla u njih pisati

- Prosljeđivanje parametara po redoslijedu

```
EXEC shema.naziv @v1 OUTPUT, ..., @vn OUTPUT
```

- Prosljeđivanje parametara po nazivu

```
EXEC shema.naziv
    @p1 = @v1 OUTPUT, ..., @pn = @vn OUTPUT
```

ALGEBRA BERNAYS SVEUČILIŠTE

Primjeri

7. Napišite proceduru koja prima parametre @IDProizvod i @Boja. Parametar @Boja neka bude izlazni parametar. Neka procedura za zadani proizvod vrati njegovu boju pomoću izlaznog parametra. Pozovite proceduru na oba načina i ispišite vraćenu vrijednost. Uklonite proceduru.

8. Napišite proceduru koja prima kriterij po kojemu ćete filtrirati prezimena iz tablice Kupac. Neka procedura pomoću izlaznog parametra vrati broj zapisa koji zadovoljavaju zadani kriterij. Neka procedura vrati i sve zapise koji zadovoljavaju kriterij. Pozovite proceduru i ispišite vraćenu vrijednost. Uklonite proceduru.

9. Napišite proceduru koja za zadanog komercijalistu pomoću izlaznih parametara vraća njegovo ime i prezime te ukupnu količinu izdanih računa.

Procedure s RETURN parametrom

- Pomoću RETURN parametra možemo vratiti samo **cjelobrojnu (int)** vrijednost iz procedure
  - Ne možemo vratiti druge tipove podataka

- Najčešće se koristi za vraćanje **informacije o uspjehu ili pogreški** u proceduri
  - Običaj je da `0` znači uspjeh, a sve ostalo neuspjeh

- Vrijednost vraćamo naredbom RETURN koja ujedno označava i kraj izvršavanja procedure

- Sintaksa za dohvaćanje RETURN vrijednosti pri pozivu:

```
EXEC @vratila = shema.naziv parametri
```

ALGEBRA BERNAYS SVEUČILIŠTE

Zaštita procedura

- Procedure možemo zaštititi na isti način kao i poglede
  - Opcija `WITH ENCRYPTION`
  - Piše se ispred ključne riječi `AS`
  - Ponašanje isto kao i kod pogleda

ALGEBRA BERNAYS SVEUČILIŠTE

Primjeri

10. Napišite proceduru koja prima ime i prezime kupca i vraća 0 kao RETURN parametar ako kupac postoji u tablici, odnosno
    200 ako kupac ne postoji. Pozovite proceduru i ispišite RETURN vrijednost.

11. Promijenite proceduru iz prethodnog zadatka tako da bude zaštićena.

12. Uklonite proceduru.

ALGEBRA BERNAYS SVEUČILIŠTE

Odgođena provjera referenci

- Prilikom izvršavanja `CREATE` ili `ALTER` naredbe za proceduru, dešava se sljedeće:
  - Za svaku tablicu koju procedura koristi, RDBMS radi provjeru:
    - Ako tablica postoji, radi se provjera postojanja (i odgovarajuće definicije) stupaca iz tablice koje procedura koristi
      - Ako je sve u redu, ide se na provjeru sljedeće tablice
      - Ako stupac ne postoji ili nema odgovarajuću definiciju, RDBMS javlja grešku
    - **Ako tablica ne postoji, RDBMS smatra da je sve OK i prelazi na provjeru sljedeće tablice!**
      - Provjera postojanja ove tablice i njenih stupaca se odgađa za trenutak pokretanja procedure – odgođena provjera referenci (engl. *deferred name resolution*)

ALGEBRA BERNAYS SVEUČILIŠTE

Primjeri

13. Napravite tablicu Student koja se sastoji od stupaca IDStudent, Ime, Prezime i JMBAG i umetnite neke podatke. Napišite proceduru koja vraća ime i prezime iz te tablice i pozovite je.

14. Promijenite proceduru tako da uz ime i prezime vraća i datum rođenja. Što se desilo?

15. Promijenite proceduru tako da vraća sve zapise iz tablice IzmisljenaTablica. Što se desilo? Pokrenite proceduru. Što se desilo?

16. Napravite tablicu IzmisljenaTablica i pokrenite proceduru. Što se desilo?