OBLIKOVANJE BAZA PODATAKA

Predavanje 05

Blic

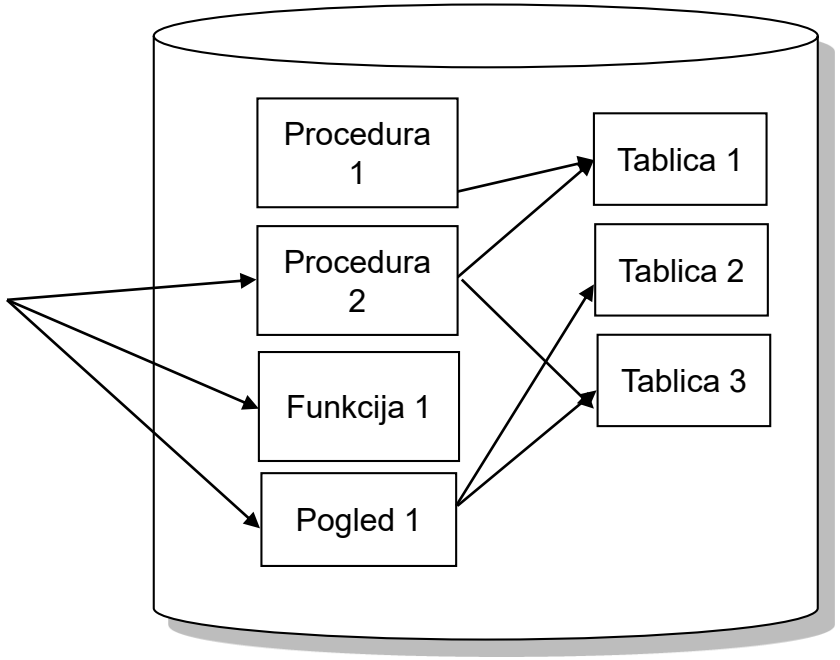
- https://forms.gle/K8gd8deKGXWR8gsf7

Korisničke funkcije

Korisničke funkcije

- **Korisničke funkcije** (engl. UDF, *user-defined functions*) su objekti u bazi podataka koji **služe za dohvaćanje podataka**

- Po strukturi i ulozi su slični procedurama:
  - Implementiraju pristupni sloj
  - Objekti su u bazi
  - Imaju naziv
  - Primaju parametre
  - Vraćaju rezultate



Procedure protiv funkcija

- Funkcije koristimo kad:
  1. Želimo vratiti jednu vrijednost i iskoristiti je na nekom mjestu u upitu (umjesto podupita)
  2. Želimo vratiti skup redaka i iskoristiti ga kao izvor podataka za FROM

- Procedure koristimo u svim ostalim slučajevima:
  - Kada je potrebno dodavati, mijenjati ili brisati podatke
  - Kada je potrebno vratiti više skupova redaka kao rezultate
  - Kad su nam potrebni izlazni parametri
  - Kad želimo poslati e-mail iz baze podataka
  - Kad želimo koristiti strukturirano hvatanje grešaka
  - …

Vrste funkcija

- Postoje tri vrste funkcija:
  - Funkcije koje vraćaju jednu **skalarnu vrijednost** (engl. *scalar-valued functions*)
  - **Jednostavne funkcije** koje vraćaju tablicu (engl. *inline table-valued functions*)
  - **Složene funkcije** koje vraćaju tablicu (engl. *multi-statement table-valued function*)

Skalarne funkcije

Izrada, izmjena i uklanjanje skalarnih funkcija

- Primaju nula ili više parametara i vraćaju **jednu vrijednost**
- Sintaksa kreiranja/izmjene funkcije:

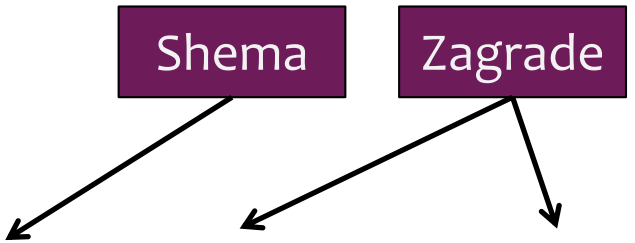

```
CREATE/ALTER FUNCTION shema.naziv
(
    @p1 tip, @p2 tip, ...
)
RETURNS povratni_tip
AS
BEGIN
    niz_naredbi
    RETURN vrijednost
END
```

- Sintaksa uklanjanja funkcije (**jednako za sve vrste funkcija**):

```
DROP FUNCTION shema.naziv
```

Korištenje skalarnih funkcija

- Kod korištenja **obavezno navesti shemu (iznimka!)**
- Korištenje funkcije
  - Samostalno




```
DECLARE @var tip
SET @var = shema.naziv(v1, v2, …)
```

  - Unutar SELECT naredbe (na istim mjestima kao skalarni podupit)

```
SELECT shema.naziv(v1, v2, …)
FROM tablica
```

    - Moguće ih koristiti i u drugim dijelovima SELECT naredbe
    - Funkcija se poziva za svaki redak – **paziti na performanse!**

Karakteristike skalarnih funkcija

- Bitno uočiti kod skalarnih funkcija:
  - Parametri kod definiranja i korištenja funkcija **moraju biti navedeni unutar zagrada**
  - Tip koji funkcija vraća određujemo naredbom **`RETURNS`**, a konkretnu vrijednost vraćamo naredbom **`RETURN`**
    - Zadnja naredba u funkciji mora biti **`RETURN`**
  - Tijelo funkcije mora biti ograničeno s **`BEGIN`** i **`END`**
  - Kod korištenja je **obavezno navesti shemu**
  - Mogu primiti samo ulazne parametre
  - **Plan izvršavanja se izrađuje odvojeno od vanjskog upita i čuva za ponovnu upotrebu**

Primjeri skalarnih funkcija

1. Napišite funkciju koja prima ID proizvoda i dohvaća broj prodanih primjeraka. Pozovite funkciju samostalno. Dohvatite sve nazive i boje i uz svaki proizvod ispišite koliko primjeraka je prodano. Promijenite funkciju tako da vrati 0 za one proizvode koji nisu prodani niti u jednom primjerku. Ako treba, optimizirajte. Uklonite funkciju.

2. Napišite funkciju koja prima string. Ako je broj znakova u stringu <= 10, neka funkcija vrati ulazni string. Ako ne, neka vrati prvih 7 znakova i iza toga tri točke. Ispišite nazive svih proizvoda koristeći napravljenu funkciju.

3. Napišite funkciju koja za zadanog kupca vraća datum najnovije kupovine. Ispišite sve kupce i kraj svakog ispišite datum najnovije kupovine. Ako treba, optimizirajte.

Tablične funkcije

Jednostavne funkcije koje vraćaju tablicu

- Vraćaju tablicu koja je rezultat jednog `SELECT` upita (slično kao i pogledi)
  - Zbog sličnosti s pogledima ovaj tip funkcija se često naziva i **parametrizirani pogledi**

- Glavna prednost – mogu se koristiti kao **izvori podataka**

- Sintaksa za kreiranje i izmjenu je nešto drukčija:

```
CREATE/ALTER FUNCTION shema.naziv
(
        @p1 tip, @p2 tip,   ...
)
RETURNS TABLE
AS
        RETURN jedan_select_upit
```

Specifičnosti jednostavnih tabličnih funkcija

- Korištenje:

  `SELECT * FROM` *`shema.naziv(v1, v2, …)`*
  - Nije obavezno navesti shemu

- Specifičnosti:
  - Za povratni tip podatka se postavlja **`TABLE`**
  - Tijelo funkcije je ograničeno na **samo jedan `SELECT`**
  - Tijelo funkcije ne smije biti unutar `BEGIN/END` bloka
  - Iza `RETURN` ide jedan `SELECT` upit koji čini tijelo funkcije
  - Plan izvršavanja se ne izrađuje za samu funkciju
    - Plan izvršavanja se izrađuje za cijeli upit u kojem sudjeluje funkcija (kao kod pogleda)

Primjeri jednostavnih tabličnih funkcija

4. Napišite jednostavnu tabličnu funkciju koja vraća IDKupac, ime i prezime svih osoba čije prezime započinje sa zadanim stringom. Iskoristite funkciju za dohvat svih osoba čije prezime započinje sa 'Ac'.
   Uz svaku osobu dohvatite i njegove račune.

5. Napišite jednostavnu tabličnu funkciju koja prima dva datuma. Neka funkcija vrati sve račune koji su izrađeni između tih datuma. Iskoristite funkciju za dohvat računa između 1.1.2004. i 10.1.2004.
   Promijenite funkciju tako da datum izdavanja vrati u hrvatskom formatu.

Složene funkcije koje vraćaju tablicu

- Slične jednostavnima, ali se tijelo može sastojati od proizvoljnog broja naredbi koje ne mijenjaju stanje baze

```
CREATE/ALTER FUNCTION shema.naziv
(
    @p1 tip, @p2 tip,    ...
)
RETURNS @temp_tablica TABLE
(
    naziv_stupca1 tip_stupca1,
    naziv_stupca2 tip_stupca2, ...
)
AS
BEGIN
    niz_naredbi   -- Uglavnom rade INSERT u temp_tablicu
    RETURN        -- vraća @temp_tablica
END
```

Specifičnosti složenih tabličnih funkcija

- Specifičnosti:
  - Tablica koju će vratiti složena funkcija mora imati naziv i definiranu strukturu stupaca
  - Tijelo funkcije mora biti unutar `BEGIN/END` bloka
  - Tijelo funkcije se može sastojati od proizvoljnog broja naredbi, ali **ni jedna naredba ne smije direktno vratiti retke pozivatelju**
    - Naredbe smiju umetati retke u privremenu tablicu (naredba `INSERT INTO … SELECT … )`
  - Funkcija završava naredbom `RETURN` bez parametara
    - RDBMS će u tom trenutku automatski vratiti tablicu definiranu kod kreiranja funkcije
  - **Plan izvršavanja se izrađuje odvojeno od vanjskog upita i čuva za ponovnu upotrebu**

Primjeri složenih tabličnih funkcija

6. Napišite funkciju jednaku onoj iz zadatka 11, ali neka ova bude složena. Iskoristite funkciju.

7. Napišite složenu tabličnu funkciju koja prima cijenu. Ako je cijena NULL, vratite nazive i cijene svih proizvoda iz tablice Proizvod. Ako nije, vratite nazive i cijene samo onih proizvoda čija cijena je veća od zadane cijene. Iskoristite funkciju s NULL i s nekom cijenom.

Karakteristike svih tipova funkcija

- Karakteristike funkcija smo već upoznali na drugim tipovima objekata:
  - Koriste odgođenu provjeru referenci (kao i procedure)
  - Moguće koristiti dodatne opcije
    - Uspostavljanje ovisnosti funkcija i tablica koje koriste
      - `WITH SCHEMABINDING` (kao i pogledi)
    - Zaštita sadržaja
      - `WITH ENCRYPTION` (kao pogledi i procedure)
    - Dodatne opcije se stavljaju prije ključne riječi `AS`