

OBLIKOVANJE BAZA PODATAKA

Vježba 4

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
```

Procedure bez parametara

1. Napišite proceduru koja dohvaća sve retke iz tablice Drzava. Pozovite proceduru. Promijenite proceduru tako da vraća rezultate poredane padajuće po nazivu države. Uklonite proceduru.
2. Napišite proceduru koja dohvaća prvih 5 redaka iz tablice Racun, prvih 5 redaka iz tablice Stavka i prvih 5 redaka iz tablice Proizvod. Pozovite proceduru. Uklonite proceduru.

Procedure s ulaznim parametrima

- T-SQL za kreiranje procedure s ulaznim parametrima:

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

- 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

Procedure s ulaznim parametrima

3. Napišite proceduru koja prima @ID proizvoda i vraća samo taj proizvod iz tablice Proizvod. Pozovite proceduru na oba načina. Uklonite je.
4. Napišite proceduru koja prima dvije cijene i vraća nazive i cijene svih proizvoda čija je cijena u zadanom rasponu. Pozovite proceduru na oba načina. Uklonite proceduru.
5. Napišite proceduru koja prima naziv države i naziv grada. Neka procedura umetne grad koji pripada zadanoj državi. Pazite na to postoji li već država upisana u tablicu Drzava ili ne postoji. Ako postoji, nemojte je umetati ponovno. Pozovite proceduru za državu "Japan" i grad "Osaka". Pozovite proceduru za državu "Japan" i grad "Tokyo" i potvrdite da radi ispravno. Uklonite proceduru.

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
```

Procedure s izlaznim parametrima

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

Dodatni zadaci

8. Napišite proceduru koja za zadanog komercijalistu pomoću izlaznih parametara vraća njegovo ime i prezime te ukupnu zarađenu količinu novaca.
9. Napišite proceduru koja umeće novu kategoriju i kroz izlazni parametar vraća generiranu IDENTITY vrijednost. Ako već postoji kategorija zadanog imena, ne treba je upisivati ponovno i u tom slučaju kroz izlazni parametar vratite vrijednost -1. Pozovite proceduru i ispišite vraćenu vrijednost. Uklonite proceduru.
10. Napišite proceduru koja kroz tri izlazna parametra vraća najmanju, najveću i prosječnu cijenu proizvoda iz tablice Proizvod. Neka procedura vrati i sve proizvode koji imaju cijenu veću od 0 i manju od prosječne. Pozovite proceduru, ispišite vraćene vrijednosti i uklonite proceduru.

Procedure s RETURN parametrima

- 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

Procedure s RETURN parametrima

11. Napišite proceduru koja prima ime i prezime osobe i vraća 0 kao RETURN parametar ako osoba postoji u tablici, odnosno 200 ako osoba ne postoji. Pozovite proceduru i ispišite RETURN vrijednost.
12. Promijenite proceduru iz prethodnog zadatka tako da bude zaštićena.
13. Uklonite proceduru.