OBLIKOVANJE BAZA PODATAKA

Vježba 3

Uvod u okidače

- Okidači (engl. ***triggers***) su posebna vrsta procedura, imaju slične karakteristike (npr. plan izvršavanja se čuva)

- Svaki okidač je vezan uz jednu tablicu ili pogled

- **Ne poziva ih korisnik već RDBMS** kao reakciju na događaje
  - Događaj (akcija) uzrokuje pozivanje okidača (reakcija)
  - Događaj i svi pokrenuti okidači su unutar iste transakcije

- Postoje dvije vrste okidača
  - DDL okidači reagiraju na izmjenu strukture baze – rjeđe se koriste
  - DML okidači reagiraju na promjenu podataka – češće se koriste

DML okidači

- Okidač je uvijek vezan uz točno jednu tablicu (ili pogled)

- Događaji uz koje je vezan i koji uzrokuju njegovo pozivanje:

  INSERT naredba, UPDATE naredba, DELETE naredba

- Svaki okidač može biti vezan uz bilo koju kombinaciju događaja

- Dvije podvrste:
  - **INSTEAD OF** – izvršavaju se **umjesto** pokretačke naredbe
  - **AFTER** – izvršavaju se **nakon** pokretačke naredbe

DML AFTER okidači

- U ovom kolegiju nas zanimaju samo **DML AFTER okidači**

- Specifičnosti okidača:
  - Ne možemo ih pozivati direktno, već to za nas radi RDBMS kao odgovor na određene događaje
  - **Ne mogu primati parametre i ne mogu vraćati podatke**
  - Mogu umetati, mijenjati i brisati podatke iz bilo koje tablice

T-SQL za rad s okidačima

- Kreiranje / izmjena okidača:

  CREATE / ALTER TRIGGER *shema_tablice.naziv_okidača*
  ON *shema_tablice.tablica*
  AFTER INSERT, UPDATE, DELETE
  AS
  niz_naredbi

- Možemo navesti bilo koju kombinaciju INSERT, UPDATE i DELETE događaja odvojenu zarezima

- Umjesto **AFTER** možemo koristiti **FOR** (sinonimi)

- Uklanjanje okidača: **DROP TRIGGER** *naziv_okidača*

Specijalne tablice

- Unutar okidača postoje dvije specijalne tablice:
  - tablica **inserted**
  - tablica **deleted**
  - **ne postoji tablica updated**

- Tablice uvijek postoje: mogu biti prazne ili popunjene

- Struktura im je jednaka strukturi tablice na kojoj je okidač

- **Tablice i podatke u njima moguće je samo čitati**

- Tablice postoje samo unutar okidača, završetkom okidača iz memorije nestaju i tablice

Retci u specijalnim tablicama

Postojanje redaka u specijalnim tablicama je određeno događajem za koji je okidač pozvan:

INSERT

- Tablica **inserted** sadržava sve upravo umetnute retke
- Tablica **deleted** je prazna

DELETE

- Tablica **inserted** je prazna
- Tablica **deleted** sadržava sve upravo obrisane retke

UPDATE

- Tablica **inserted** sadrži nove verzije svih ažuriranih redaka
- Tablica **deleted** sadrži stare verzije svih ažuriranih redaka

Broj redaka u specijalnim tablicama

- Specijalne tablice sadržavaju onoliko redaka koliko je bilo pogođenom originalnom SQL naredbom

INSERT INTO Osoba (Ime) VALUES ('Miro')

- inserted sadržava 1 redak
- deleted sadržava 0 redaka

DELETE FROM Osoba WHERE Ime = 'Iva'

- inserted sadržava 0 redaka
- deleted sadržava onoliko redaka koliko ima Iva u tablici

UPDATE Osoba SET Placa = Placa*1.1 WHERE Placa < 5000.0

- inserted i deleted sadržavaju onoliko redaka koliko ima zapisa s plaćom manjom od 5000 kuna

Broj redaka u specijalnim tablicama

- Kod UPDATE naredbe SET dio može promijeniti jedan ili više stupaca
- Treba nam način kako saznati koji stupci su pogođeni SET dijelom naredbe
- Sistemska funkcija **UPDATE (*naziv stupca*)**
  kao parametar prima naziv stupca
  te vraća **true** ili **false**

Skripte i skupine naredbi

- T-SQL skripta je **datoteka** s nastavkom **.sql** i sastoji se od proizvoljnog broja naredbi

- Skriptu možemo podijeliti u manje cjeline:
  - **Skupina naredbi** (engl. *batch*) je niz naredbi s karakteristikama:
    - Prije izvršavanja se sve naredbe iz te skupine kompajliraju i izrađuje se plan izvršavanja (engl. *execution plan*)
    - Naredbe se izvršavaju jedna po jedna
  - Varijable iz jedne skupine nisu vidljive drugoj skupini
  - Skupine naredbi se kreiraju korištenjem naredbe `GO`
  - Određene T-SQL naredbe se **moraju** nalaziti u vlastitoj skupini: `CREATE PROC`, `CREATE VIEW`, ...

Primjer skripte sa skupinama naredbi

```
SELECT TOP 3 * FROM Kupac
GO

DECLARE @Ime nvarchar(50)

SELECT @Ime = Ime
FROM Kupac
WHERE IDKupac = 9821

PRINT @Ime
GO

PRINT @Ime -- Greška!
```

Skupina 1

Skupina 2

Skupina 3

Skripta

Drugi primjer skripte sa skupinama naredbi

```
SELECT TOP 3 * FROM Kupac
SELECT TOP 3 * FROM Racun
GO

SELECT TOP 3 * FROM Kupac
SELECT TOP 3 Nepostojece FROM Kupac
GO

DELETE FROM Racun
SELECT TOP 3 * FROM Racun
```

- Koliko skupina ovdje vidimo i što će se desiti izvršavanjem cijele skripte?

Varijable

- Deklariranje varijable:
  `DECLARE @naziv tip` -- bilo koji SQL tip
- Varijabla je vidljiva samo u onoj skupini naredbi (*batchu*) u kojoj je deklarirana
- Tri načina dodjeljivanja vrijednosti:

1. **Direktno**:

```
DECLARE @ime nvarchar(50)
SET @ime = 'Ana'
```

2. **Pomoću podupita** koji vraća skalar:

```
DECLARE @prodano int
SET @prodano = (SELECT
SUM(Kolicina) FROM Stavka)
```

3. **Pomoću SELECT upita** koji bi trebao vratiti **jedan redak**:

```
DECLARE @NazivProizvoda nvarchar(50)
DECLARE @BojaProizvoda nvarchar(50)
SELECT
  @NazivProizvoda = p.Naziv,
  @BojaProizvoda = p.Boja
FROM Proizvod AS p WHERE p.IDProizvod=741
```

- Ako SELECT vrati **nula redaka**, obje varijable će sadržavati NULL vrijednosti
- Ako SELECT vrati **više redaka**, varijable će poprimiti vrijednosti **iz zadnjeg vraćenog retka**

Varijable

Vježbe se rade na bazi **AdventureWorksOBP**.

1. Deklarirajte varijable @Ime i @Prezime i dodijelite im neke vrijednosti. Ispišite dodijeljene vrijednosti.

2. Deklarirajte varijable @Ime i @Prezime i dodijelite im vrijednosti iz tablice Kupac za IDKupac jednak 8812. Ispišite dodijeljene vrijednosti.

3. Deklarirajte varijable @Ime i @Prezime i dodijelite im vrijednosti iz tablice Kupac tako da odaberete sve retke iz tablice. Ispišite dodijeljene vrijednosti.

IF-ELSE i SCOPE_IDENTITY()

- Uvjetno izvršavanje dijelova kôda:

```
IF logički_uvjet_1 = TRUE
    BEGIN
        niz_naredbi_1
    END
ELSE IF logički_uvjet_2 = TRUE
    BEGIN
        niz_naredbi_2
    END
ELSE
    BEGIN
        niz_naredbi_3
    END
```

- Sistemska funkcija **SCOPE_IDENTITY()** vraća zadnje eksplicitno kreiranu IDENTITY vrijednost
- Dobra praksa je odmah staviti tu vrijednost u varijablu

```
DECLARE @ID int

INSERT INTO Drzava (Naziv) VALUES ('Indija')

SET @ID = SCOPE_IDENTITY()

INSERT INTO Grad (Naziv, DrzavaID)
VALUES
('Agra', @ID),
('Delhi', @ID)
```

IF-ELSE i SCOPE_IDENTITY()

4. Provjerite broj zapisa u tablici Kupac. Ako ih ima više ili jednako 20000, ispišite “Postoji više od 20000 kupaca :)”. Ako ih ima manje, ispišite “Još nismo dostigli 20000 kupaca :(”

5. Umetnite zapis u tablicu Drzava, generiranu IDENTITY vrijednost dodijelite nekoj varijabli pa je ispišite.

6. Umetnite zapis u tablicu Drzava i u varijablu spremite generiranu IDENTITY vrijednost. Iskoristite tu vrijednost da biste za tu državu umetnuli dva grada.

Uključivanje i isključivanje okidača

- Uključeni (engl. ***enabled***) okidači se pozivaju od strane RDBMS-a kao reakcija na događaje

- Isključeni (engl. ***disabled***) okidači se ne pozivaju – dobije se efekt kao da nisu niti kreirani
  - Isključivanje okidača se obično radi pri umetanju većih količina podataka u tablicu

- Sintaksa:

  DISABLE TRIGGER *naziv_okidača* ON *tablica*

  ENABLE TRIGGER *naziv_okidača* ON *tablica*

Okidač vezan uz više događaja

- Jedan okidač može biti vezan uz 1, 2 ili sva 3 događaja
- U okidaču vezanom za 2 ili 3 događaja moramo moći razlikovati koji se događaj desio:
  - Ako postoje retci u inserted, a ne postoje u deleted, dogodio se INSERT
  - Ako ne postoje retci u inserted, a postoje u deleted, dogodio se DELETE
  - Ako postoje retci i u inserted i u deleted, dogodio se UPDATE

Više okidača na istoj tablici

- Na istoj tablici može biti više okidača vezanih uz isti događaj - RDBMS tada poziva sve vezane okidače
- **Predefinirano ponašanje** RDBMS-a je da poziva okidače **redoslijedom kako su kreirani**
- Možemo utjecati na izvođenje tako da definiramo koji okidač se izvršava **prvi**, a koji **posljednji** (na ostale ne možemo utjecati):

```
EXEC sp_settriggerorder
    @triggername = 'naziv_okidača',
    @order = 'FIRST | LAST | NONE'
    @stmttype = 'INSERT | UPDATE | DELETE'
```

Napredne postavke okidača

- **Ugniježđeni** (engl. *nested*) okidač je uzrokovan drugim okidačem
- **Rekurzivni** (engl. *recursive*) okidač akcijom nad tablicom poziva sam sebe
- Kod obje vrste okidača **postoji opasnost beskonačne petlje**, RDBMS-ovi na razne načine rješavaju taj problem
  - SQL Server dopušta najviše 32 poziva ugniježđenih okidača

Okidači (1)

Napravite tablicu Zapisnik (IDZapisnik, Poruka, Vrijeme).

1. Napravite okidač kojim ćete svako umetanje retka u tablicu Grad zapisati u tablicu Zapisnik. Umetnite redak.

2. Promijenite okidač tako da zapiše ID i naziv umetnutog grada u Zapisnik. Umetnite redak.

3. Promijenite okidač tako da se veže uz sve događaje i u Zapisnik zapisuje broj redaka u inserted i deleted tablicama. Umetnite dva nova grada, promijenite im države i na kraju ih obrišite.

4. Promijenite okidač tako da upisuje staru i novu vrijednost promijenjenog naziva grada u Zapisnik. Promijenite naziv jednom gradu.

Okidači (2)

5. Onemogućite okidač iz prethodnih primjera i promijenite neki redak u tablici. Ponovno ga omogućite. Napravite opet promjenu. Uklonite okidač.

6. Dodajte novi okidač na tablicu Grad i vežite ga uz sva tri događaja. U okidaču saznajte koji događaj ga je pozvao i tu informaciju upišite u Zapisnik. Napravite umetanje, izmjenu i brisanje nekog retka. Uklonite okidač.

7. Dodajte novi okidač i vežite ga uz UPDATE događaj na tablici Grad. Neka okidač zapiše u Zapisnik da se desio događaj samo ako je promijenjen stupac DrzavaID. Uklonite okidač.

Okidači (3)

8. Dodajte 4 nova okidača na tablici Grad koji u zapisnik ispisuju "Pozdrav iz broja n" nakon umetanja retka. Umetnite redak. Posložite redoslijed okidača tako da bude 4, 2, 3, 1. Umetnite redak. Vratite originalni redoslijed. Umetnite redak. Uklonite okidače.

9. Napravite tablice Tbl1 i Tbl2 s proizvoljnim stupcima. Na Tbl1 napravite okidač vezan uz INSERT koji umeće retke u Tbl2 i u Zapisnik. Na Tbl2 napravite okidač vezan uz INSERT koji umeće retke u Tbl1 i u Zapisnik. Umetnite jedan redak u Tbl1. Što piše u svakoj od tablica?