



# OBLIKOVANJE BAZA PODATAKA

Vježba 08

# Pretvaranje tipova podataka

*CAST(izraz AS tip\_podataka)*

*CONVERT(tip\_podataka, izraz [ , stil ])*

1. Dohvatite brojeve računa i datume izdavanja za kupca s ID-em 378. Datume izdavanja formatirajte na hrvatski način.
2. Napišite proceduru za umetanje zapisa u tablicu Drzava. Neka procedura kroz izlazni parametar vrati vrijednost primarnog ključa novog zapisa. Pozovite proceduru i ispišite vraćenu vrijednost u formatu: "Umetnuta je država s ID-em  $n$ ", gdje je  $n$  vrijednost primarnog ključa.
3. Dohvatite nazive svih proizvoda i uz svaki naziv u zagradi ispišite i njegov ID te cijenu, npr. "HL Road Rear Wheel (ID = 828, Cijena = 357.06)"

# Naredbe CASE i WHILE

```
CASE
    WHEN uvjet1 THEN rezultat1
    WHEN uvjet2 THEN rezultat2
    ...
    ELSE rezultat
END
```

```
WHILE uvjet
BEGIN
    naredbe
END
```

4. Dohvatite nazive svih proizvoda i uz svaki naziv ispišite i naziv potkategorije. Ako neke potkategorije nema, napišite “Nije definirana”.
5. Dohvatite naziv i cijene svih proizvoda. Za cijene koje su ispod 1000, napišite “Jeftino”, između 1000 i 2000 napišite “Prihvatljivo”, za sve ostale napišite “Skupo”.
6. Napravite tablicu Proba sa stupcima IDProba (primarni ključ i IDENTITY) i Vrijednost (int). U stupac Vrijednost unesite vrijednosti između 10.000.000 i 10.015.000.

# Strukturirano hvatanje pogrešaka

Svaka greška u SQL Serveru ima svoju razinu ozbiljnosti (engl. severity level):

- o Razine 1 do 10 predstavljaju informacije

- o **Razine 11 do 16 predstavljaju korisničke greške**

- **11 – traženi objekt ne postoji (npr. DROP TABLE NemaMe)**
- **15 – upit nema ispravnu sintaksu (npr. SELECT \* FRMO Student)**
- **16 – sve ostale korisničke greške (npr. PRINT 17/0)**

- o **Razine 17 do 19 predstavljaju softverske greške koje zahtijevaju intervenciju administratora**

- **17 – nema prostora na disku za traženu operaciju**

- o Razine 20 do 25 predstavljaju fatalne greške i mogu rezultirati zatvaranjem konekcije

- T-SQL nudi mogućnost strukturiranog hvatanja grešaka, za greške 11 do 19

# Strukturirano hvatanje pogrešaka

- T-SQL nudi mogućnost strukturiranog hvatanja grešaka
  - Hvataju se **greške razine 11 do 19** (1-10 i 20-25 se ne hvataju)

- Sastoji se od dva bloka, TRY i CATCH
  - Ako se u TRY **desi** greška, izvođenje se prebacuje u CATCH
  - Ako se u TRY **ne desi** greška, CATCH blok neće biti izvršen

- Sintaksa:

```
BEGIN TRY  
    niz_naredbi  
END TRY  
BEGIN CATCH  
    niz_naredbi  
END CATCH
```



# Strukturirano hvatanje pogrešaka

- U CATCH bloku su nam korisne sljedeće sistemske funkcije:

- **ERROR\_MESSAGE()**

- Vraća tekst greške

- **ERROR\_NUMBER()**

- Vraća broj greške (svaka greška ima svoj jedinstveni broj)

- **ERROR\_SEVERITY()**

- Vraća razinu greške koja se desila

- **ERROR\_LINE()**

- Vraća liniju u kojoj se desila greška

- **ERROR\_PROCEDURE()**

- Vraća naziv procedure u kojoj se desila greška

# Strukturirano hvatanje pogrešaka

7. Napravite proceduru koja prima dva broja i kroz izlazni parametar vraća prvi broj podijeljen drugim. Ako se desi greška, neka procedura u izlazni parametar upiše 0 i neka ispiše tekst greške. Pozovite proceduru i ispišite rezultat dijeljenja.
8. Napravite tablicu Zivotinja koja ima stupce IDZivotinja (primarni ključ, ali nije IDENTITY) i Naziv. Napravite proceduru koja prima IDZivotinja i Naziv i umeće ih u tablicu. Pozovite proceduru 2x s vrijednostima 20 i "Čaplja".
  - Implementirajte TRY/CATCH izvan procedure i pozovite je.
  - Implementirajte TRY/CATCH unutar procedure i pozovite je.

# Procedure vs funkcije - podsjetnik

- 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
  - ...



# CRUD operacije

- CRUD operacije:

**C** = Create   **R** = Retrieve   **U** = Update   **D** = Delete

9. Napravite procedure koje rade CRUD operacije na tablici Student tako da svakoj operaciji dodijelite posebnu proceduru. Neka C procedura kroz izlazni parametar vrati novi IDStudenta. Iskoristite procedure za umetanje, izmjenu, dohvaćanje i brisanje zapisa.
10. Napravite procedure koje rade CRUD operacije na tablici Student tako da operacije umetanja i izmjene obavite u jednoj proceduri, a druge dvije operacije obavite u posebnim procedurama. Iskoristite procedure za umetanje, izmjenu, dohvaćanje i brisanje zapisa.
11. Napravite procedure koje rade CRUD operacije na tablici Student tako da operacije umetanja, izmjene i brisanja obavite u jednoj proceduri, a dohvaćanje u drugoj. Iskoristite procedure za umetanje, izmjenu, dohvaćanje i brisanje zapisa.

# CRUD operacije

12. Za tablicu Student koja se sastoji od IDStudent, Ime, Prezime i JMBAG implementirajte tabličnom funkcijom operaciju SELECT. Iskoristite funkciju za dohvaćanje zapisa.