

Háttéralkalmazások

Haladó SQL nyelvi elemek és
platformfüggő sajátosságok (Microsoft
SQL Server)



Automatizálási és
Alkalmazott
Informatikai Tanszék

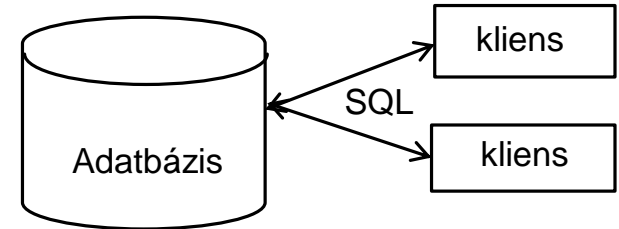
Tartalom

- Adatbázisok szerver oldali programozása
- Transact-SQL nyelv
 - > Kurzorok
 - > Tárolt eljárások és függvények
 - > Hibakezelés
 - > Triggerek

Adatbázisok szerveroldali programozása

Adatbázis szerepe

- Adatmodell: táblák
- Adatmanipuláció: SQL nyelv



- Feladatok, amelyek kimutatnak a relációs adatmodellből
 - > Üzleti logikai réteg
 - > Adatréteg → **Adatbázis szerveroldali programozása**

Szerveroldali programozás: motiváció

- Példa: Neptun rendszer vizsga jelentkezés
 - > Vizsga tábla: ID, Kurzus, Dátum, MaxLétszám
 - > Vizsgajelentkezés tábla: VizsgaID, HallgatóID
- Pszeudokóddal:

```
func vizsgaJelentkezés(vizsgaid, hallgatoid) {  
    jelentezettek_db = számol vizsgajelentkezés ID==vizsgaid  
    if jelentezettek_db < vizsga.maxletszam  
        beszur vizsgajelentkezés vizsgaid, hallgatoid  
    else hiba  
}
```

Szerveroldali programozás: motiváció

- Példa: Neptun rendszer vizsga jelentkezés
- T-SQL tárolt eljárással

```
create procedure vizsgajelentkezes
@vizsgaid int, @hallgatoid int
as
begin
    declare @jelentezett_db int =
        select count(*) from vizsgajelentkezes where ID=@vizsgaid
    declare @vizsgakorlat int
    select @vizsgakorlat = maxletszam from vizsga where ID=@vizsgaid
    if @jelentezett_db < @vizsgakorlat
        insert into vizsgajelentkezes values(@vizsgaid, @hallgatoid)
    else
        throw 51005, 'megengedett letszam atlepv'e'
end
```

Szerveroldali programozás: motiváció

- Példa: Neptun rendszer vizsga jelentkezés
- T-SQL tárolt eljárással

```
create procedure vizsgajelentkezes  
@vizsgaid int, @hallgatoid int  
as  
begin  
    declare @jelentezettek_db int =  
        select count(*) from vizsgajelentkezes where ID=@vizsgaid  
    declare @vizsgakorlat int  
    select @vizsgakorlat = maxletszam from vizsga where ID=@vizsgaid  
    if @jelentezettek_db < @vizsgakorlat  
        insert into vizsgajelentkezes values(@vizsgaid, @hallgatoid)  
    else  
        throw 51005, 'megengedett letszam atlepv'e'  
end
```

Tárolt eljárás ~ metódus

Paraméterek

Lokális változó

Elágazás (control flow vezérlés)

Hibajelzés

Procedurális kód

Szerveroldali programozás előnyei – 1

- Adatbázis felelős a konzisztenciáért
 - > Adatbázis szerepe megváltozik
 - Adatforrás + Szolgáltatások
- Biztonság
 - > Adatmódosítás csak definiált „interfészen” keresztül
 - Pl. ez lesz a tárolt eljárás
 - > Zárt futtató környezet
 - Elkerülhető, hogy az adat kikerüljön az adatbázisból

Szerveroldali programozás előnyei – 2

- Teljesítmény növelés
 - > Csökkenő hálózati forgalom
 - > Adatbázis cache-elés
- Termelékenység
 - > Több komponens hívhatja a kódot
 - > Egyszerűbb karbantartás -> adatbázisban kell javítani

Szerveroldali programozás hátrányai

- Nem szabványos
 - > Platformfüggő nyelvi elemek
 - > Platformfüggő megoldások
- Interpretált
- Növeli a szerver terhelését
- Nem ill. nehezen skálázható

Transact-SQL nyelv

T-SQL nyelv

- Transact-SQL / T-SQL
- Microsoft SQL Server nyelve
- SQL szabványnak megfelelő utasítások
 - + változók
 - + utasítás blokkok
 - + ciklusok
 - + strukturált hibakezelés
 - + új konstrukciók

T-SQL nyelv

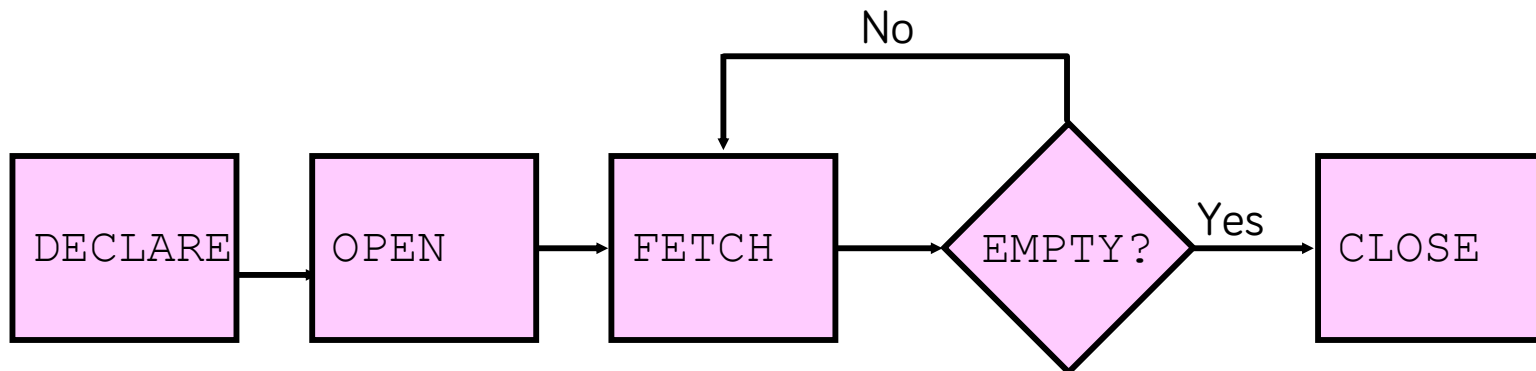
- SQL nyelv ismételés
 - MSSQL specifikus szintaktika
 - Szerveroldali programozás
- > Részleteket lásd az írásos jegyzetben (más tárgy vonatkozó része)

<https://bmeviauac01.github.io/datadriven/hu/lecture-notes/mssql/sql/>

<https://bmeviauac01.github.io/datadriven/hu/lecture-notes/mssql/server-side-programming/>

Kurzor

- Több rekordot visszaadó lekérdezés eredményének feldolgozása egyesével
- „Iterátor”



Tárolt eljárás és függvény

- Tárolt eljárás: az adatbázis szerverben tárolt programkód, amit T-SQL kódból hívhatunk
 - > Eljárás: tipikusan nincs visszatérési értéke (bár lehet)
- Tárolt függvény: van visszatérési értéke, de az adatbázist csak olvashatja (nem írhatja)
 - > Scalar-valued
 - > Table-valued
 - > (Aggregate: .NET Assembly-ben írható)

Trigger: probléma felvetés

- Legyen nyoma egy termék törlésének (audit naplózás)
 - > Megoldás 1: üzleti logikában az adatbázisból való törlés után naplózás
 - Probléma: kikerülhető, elfelejthető
 - > Megoldás 2: adatbázisban törlés „esemény” elkapása
 - Trigger
- A rendelésben legyen egy összegzés, hogy mennyi az összértéke
 - > Kiszámítható a megrendelés tételeiből, de legyen külön is elmentve (denormalizáció)
 - > Amikor a megrendelés módosul, karban kell tartani

Trigger: események

- **DML esemény**
 - > Insert, update, delete
 - > Táblához kötődik
- **Instead of triggerek**
 - > Speciális DML trigger
 - > Nézetek adatmódosítása
- **Rendszerezemény**
 - > Logon, logoff, SysError, ...
- **DDL trigger**
 - > Create, alter, drop, ...
 - > Sémához kötődik