

SQL

1.rész

SQL története, szabványok

- Szabvány adatbázis-kezelő nyelv: SQL
- SQL (angol kiejtésben SEQUEL) uis az SQL előfutára IBM fejlesztette ki a 70-es években: SEQUEL → SQL más is volt pl. Ingres : QUEL (ez kalkulus alapú lekérdezés)
- Szabványok (ANSI, ISO)
SQL86, SQL89, SQL92 (SQL2), **SQL:1999** (SQL3),
SQL: 2003, SQL:2006, SQL:2008
- Nyelvjárások (**Oracle**, Sybase, DB2, Progress, MSSQL, mySQL, SQL Server, PostgreSQL, Access,...)
- Az SQL megvalósítások között vannak különbségek, gyakorlatokon az **Oracle SQL**-t nézzük meg részletesen.

SQL fő komponensei

- **Sémaleíró nyelv, DDL** (Data Definition Language)
CREATE, ALTER, DROP
- **Adatkezelő nyelv, DML** (Data Manipulation Language)
INSERT, UPDATE, DELETE, SELECT
-- Az SQL elsődlegesen lekérdező nyelv (Query Language)
SELECT utasítás (az adatbázisból információhoz jussunk)
- **Adatvezérlő nyelv, DCL** (Data Control Language)
GRANT, REVOKE
- **Tranzakció-kezelés**
COMMIT, ROLLBACK, SAVEPOINT
- **Procedurális kiterjesztések**
Oracle PL/SQL (Ada alapján), SQL/PSM (PL/SQL alapján)

Adatbázis relációsémák definiálása

- Az SQL tartalmaz **adateleíró részt (DDL)**, az adatbázis **objektumainak** a leírására és megváltoztatására. **Objektumok** leíró parancsa a **CREATE** utasítás.
- A relációt az SQL-ben táblának (TABLE) nevezik, az SQL alapvetően háromféle táblát kezel:
 - Alaptáblák (permanens) CREATE TABLE
 - Nézet táblák CREATE VIEW
 - Átmeneti munkatáblák (WITH utasítás)
- **Alaptáblák** megadása: **CREATE TABLE**

Tábla/reláció sémák SQL-ben

- A legegyszerűbb formája:

```
CREATE TABLE relációnév (  
    Attribútum deklarációk listája,  
    További kiegészítések  
);
```

- Az attribútum deklaráció legalapvetőbb elemei:

Attribútumnév típus [kiegészítő lehetőségek]

- itt: a **típus** olyan, amit az SQL konkrét megvalósítása támogat (gyakorlaton Oracle környezetben nézzük meg),
Típusok, pl: INTEGER, REAL, CHAR, VARCHAR, DATE
- A **kiegészítő lehetőségek** például [PRIMARY KEY] vagy [DEFAULT érték] (köv.lapon példa)

Egyszerű példák táblák létrehozására

```
CREATE TABLE Sörözők (  
    név CHAR(20) ,  
    város VARCHAR2(40) ,  
    tulaj CHAR(30) ,  
    engedély DATE DEFAULT SYSDATE  
);
```

```
CREATE TABLE Felszolgál (  
    söröző CHAR(20) ,  
    sör VARCHAR2(20) ,  
    ár NUMBER(10,2) DEFAULT 100  
);
```

Az SQL értékekről (bővebben gyakorlaton)

- INTEGER, REAL, stb, a szokásos értékek, számok.
- STRING szintén, de itt egyes-aposztróf közé kell tenni a 'szöveget' (vagyis nem „macskaköröm” közé).
Két egyes-aposztróf = egynek felel meg, például
'Joe' 's Bar' megfelel a Joe's Bar szövegnek.
- DATE és TIME típusok is vannak az SQL-ben.
- A dátum formátumát meg kell adni DATE 'yyyy-mm-dd'
Például: DATE '2007-09-30' (2007. szept. 30)
- Az idő formátumát is meg kell adni TIME 'hh:mm:ss'
Például: TIME '15:30:02.5' (délután fél 4 múlt két és fél másodperccel)
- Bármely érték lehet **NULL** hiányzó érték:

Hiányzó értékek: NULL

- Az SQL lehetővé teszi a táblákban **a hiányzó értékeket**, vagyis a relációk soraiban az attribútum értéke ne legyen megadva, hanem egy speciális **NULL** nullérték legyen.
- **A nullérték értelmezésére** több lehetőségünk is van:
 - **Nem-ismert érték**: például tudom, „Joe’s Bár”-jának van valamilyen címe, de nem tudom, hogy mi az.
 - **Nem-definiált érték**: például a házastárs attribútumnak egyedülálló embereknél nincs olyan értéke, aminek itt értelme lenne, nincs házastársa, ezért nullérték.
 - stb (van olyan cikk, amely több százféle okot felsorol)

Táblák létrehozása után a táblák feltöltése adatokkal

Eddig láttuk, hogy a CREATE TABLE utasítással hogyan tudunk létrehozni táblákat és megadni a kulcsokat:

- **SQL DDL: sémaleíró nyelv** (Data Definition Language)
CREATE TABLE, ALTER TABLE, DROP TABLE

Most nézzük meg a táblák tartalmának módosítását, hogyan tudjuk INSERT utasítással a táblát feltölteni adatsorokkal:

- **SQL DML: adatkezelő nyelv** (Data Manipulation Language)
INSERT, UPDATE, DELETE, SELECT

Adatbázis tartalmának módosítása

- **Lekérdező utasítás** SELECT - lekérdezés
- **A módosító utasítások** nem adnak vissza eredményt, mint a lekérdezések, hanem az adatbázis tartalmát változtatják meg.
- 3-féle módosító utasítás létezik:
 - INSERT** - sorok beszúrása
 - DELETE** – sorok törlése
 - UPDATE** – sorok komponensei értékeinek módosítása

Beszúrás (insert into)

- Két alakja van: 1.) ha egyetlen sort szúrunk be:
INSERT INTO <reláció>
VALUES (<konkrét értékek listája>);
- 2.) ezt majd a lekérdezések után nézzük meg, hogyan tudunk több sort beolvasni a táblába, egy lekérdezés eredményét alkérdés segítségével:
INSERT INTO <reláció>
(<alkérdés>);
- **INSERT INTO** 1.) alakjára példa: a Szeret(név, sör) táblában rögzítjük, hogy Zsu szereti a Bud sört.
INSERT INTO Szeret
VALUES('Zsu', 'Bud');

Attribútumok megadása

- A reláció neve után megadhatjuk az attribútumait.
- Ennek alapvetően két oka lehet:
 1. Nem emlékszünk, hogy a reláció definíciójában, milyen sorrendben szerepeltek az attribútumok.
 2. Nincs minden attribútumnak értéke, és azt szeretnénk, ha a hiányzó értékeket NULL vagy default értékkel helyettesítenék.

Példa:

```
INSERT INTO Szeret(sör, név)  
VALUES('Bud', 'Zsu');
```

Default értékek megadása

- A CREATE TABLE utasításban az oszlopnevet **DEFAULT** kulcsszó követheti és egy érték.
- Ha egy beszúrt sorban hiányzik az adott attribútum értéke, akkor a default értéket kapja.

```
CREATE TABLE Sörivók (  
    név CHAR(30) PRIMARY KEY,  
    cím CHAR(50) DEFAULT 'Sesame St'  
    telefon CHAR(16) );  
INSERT INTO Sörivók(név)  
VALUES ('Zsu');
```

Az eredmény sor:

név	cím	telefon
Zsu	Sesame St	NULL

Tankönyv példa: Filmek séma

Filmek(

cím:string,
év:integer,
hossz:integer,
műfaj:string,
stúdióNév:string,
producerAzon:integer)

FilmSzínész(

név:string,
cím:string,
nem:char,
születésiDátum:date)

Stúdió(

név:string,
cím:string,
elnökAzon:integer)

Mit jelentenek az aláhúzások?

Tankönyv példája, hibás fordítás:

title=(film)cím és address=(lak)cím

Tervezéssel később foglalkozunk, ez a példa hibás, az elnevezések, de így jó lesz, hogy a lekérdezéseknél megnézzük hogyan kezeljük.

SzerepelBenne(

filmCím:string,
filmÉv:integer,
színészNév:string)

GyártásIrányító(

név:string,
cím:string,
azonosító:integer,
nettóBevétel:integer)

Példa megszorításokra: Kulcs

- Előző példában: attribútumok aláhúzása mit jelent?
- Filmek: elvárjuk, hogy ne legyen a megengedett előfordulásokban két különböző sor, amelyek megegyeznek cím, év attribútumokon.
- Egyszerű kulcs egy attribútumból áll, de egy kulcs nem feltétlenül áll egy attribútumból, ez az összetett kulcs. Például a **Filmek** táblában a cím és év együtt alkotják a kulcsot, nem elég a cím, ugyanis van például (King Kong, 1933), (King Kong, 1976) és (King Kong, 2005).
- A kulcsot aláhúzás jelöli: **Filmek** (cím, év, hossz, ...)

Kulcsra vonatkozó megszorítások

- Az attribútumok egy halmaza egy **kulcsot** alkot egy relációra nézve, ha a reláció **bármely előfordulásában** nincs két olyan sor, amelyek a kulcs összes attribútumának értékein megegyeznének.

- Formális megadása:

$$R(U), X \subseteq U, U = \{A_1, \dots, A_n\}, X = \{A_{j_1}, \dots, A_{j_k}\}$$

$$t \in R, t[X] = \langle A_{j_1} : t(A_{j_1}), \dots, A_{j_k} : t(A_{j_k}) \rangle$$

ezzel a jelöléssel mit jelent, hogy X kulcs elvárás?

ha $t_1 \in R, t_2 \in R$ és $t_1[X] = t_2[X]$ akkor $t_1 = t_2$

Kulcs megadása

- **PRIMARY KEY** vagy **UNIQUE**
- Nincs a relációnak két olyan sora, amely a lista minden attribútumán megegyezne.
- Kulcs esetén nincs értelme a DEFAULT értéknek.
- Kulcsok megadásának két változata van:
 - Egyszerű kulcs (egy attribútum) vagy
 - Összetett kulcs (attribútumok listája)

Egyszerű kulcs megadása

- Ha a kulcs egyetlen attribútum, akkor ez az attribútum deklarációban megadható

<attribútumnév> <típus> **PRIMARY KEY**

vagy <attribútumnév> <típus> **UNIQUE**

- Példa:

```
CREATE TABLE Sörök (  
    név          CHAR(20) UNIQUE,  
    gyártó      CHAR(20)  
);
```

Összetett kulcs megadása

- Ha a kulcs több attribútumból áll, akkor a CREATE TABLE utasításban az attribútum deklaráció után a kiegészítő részben meg lehet adni további tábla elemeket: **PRIMARY KEY (attrnév₁, ... attrnév_k)**
- Példa:

```
CREATE TABLE Felszolgal (
    söröző      CHAR(20) ,
    sör         VARCHAR2(20) ,
    ár          NUMBER(10,2) ,
    PRIMARY KEY (söröző, sör)
);
```

PRIMARY KEY vs. UNIQUE

- Csak egyetlen **PRIMARY KEY** lehet a relációban, viszont **UNIQUE** több is lehet.
- PRIMARY KEY egyik attribútuma sem lehet **NULL érték** egyik sorban sem. Viszont UNIQUE-nak deklarált attribútum lehet NULL értékű, vagyis a táblának lehet olyan sora, ahol a UNIQUE attribútum értéke **NULL** vagyis **hiányzó érték**.
- az SQL lekérdezésnél adjuk meg hogyan kell ezzel a speciális értékkel gazdálkodni, hogyan lehet NULL-t kifejezésekben és hogyan lehet feltételekben használni
- Következő héten visszatérünk a megszorítások és a hivatkozási épség megadására.

Idegen kulcsok megadása

- Az első előadáson a táblák létrehozásához veszünk kiegészítő lehetőségeket: **Kulcs és idegen kulcs (foreign key) hivatkozási épség megadása**
- Az egyik tábla egyik oszlopában szereplő értékeknek szerepelnie kell egy másik tábla bizonyos attribútumának az értékei között.
- **A hivatkozott attribútumoknak** a másik táblában kulcsnak kell lennie! (PRIMARY KEY vagy UNIQUE)
- **Példa: Felszolgál(söröző, sör, ár)** táblára megszorítás, hogy a sör oszlopában szereplő értékek szerepeljenek a **Sörök(név, gyártó)** táblában a név oszlop értékei között.

Idegen kulcs megadása: attribútumként

REFERENCES kulcsszó használatának két lehetősége:
attribútumként vagy sémaelemként lehet megadni.

1.) Attribútumonként (egy attribútumból álló kulcsra)

Példa:

```
CREATE TABLE Sörök (  
    név      CHAR(20) PRIMARY KEY,  
    gyártó   CHAR(20) );
```

```
CREATE TABLE Felszolgál (  
    söröző   CHAR(20) ,  
    sör      CHAR(20) REFERENCES Sörök(név) ,  
    ár       REAL );
```

Idegen kulcs megadása: sémaelemként

2.) Sémaelemként (egy vagy több attr.-ból álló kulcsra)

FOREIGN KEY (attribútum lista)

REFERENCES relációnév (attribútum lista)

Példa:

```
CREATE TABLE Sörök (  
    név      CHAR(20),  
    gyártó   CHAR(20),  
    PRIMARY KEY (név) );
```

```
CREATE TABLE Felszolgal (  
    söröző   CHAR(20),  
    sör      CHAR(20),  
    ár       REAL,  
    FOREIGN KEY (sör) REFERENCES Sörök (név) );
```

Kérdés/Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?
- **Oracle gyakorlaton:** Az attribútumok típusának megadásakor (például az Oracle implementációban) **milyen standard típusok** közül választhatunk, mi a különbség a CHAR és a VARCHAR között?
- A táblákat a létrehozásuk után feltöltjük adatsorokkal
- **Következő alkalommal:** Lekérdezések SQL SELECT