



Adatbázisrendszerek

11. gyakorlat

Data Definition Language (DDL)

- Az SQL adatdefiníciós (DDL) utasításai segítségével alakítható ki az adatok tárolására szolgáló séma.
 - A DDL utasítások többek között
 - táblák,
 - táblák közötti kapcsolatok,
 - indexek,
 - megszorítások
- létrehozására, módosítására és törlésére használhatók.

Tábla létrehozása

- Egy tábla létrehozásának általános alakja:
 - **CREATE TABLE** táblanév (
oszlopdefiníció
[, oszlopdefiníció]...
[, táblaszintű megszorítás]...
) ;

Tábla létrehozása

```
CREATE TABLE táblanév (  
    oszlopdefiníció  
    [, oszlopdefiníció]...  
    [, táblaszintű megszorítás]...  
);
```

- Oszlopdefiníció:
 - oszlopnév típus [oszlopszintű_megszorítások]
- Oszlop szintű megszorítások: Egyetlen oszlopra vonatkoznak. Ezek az oszlopdefiníció végén adhatók meg.
 - [**CONSTRAINT** név] megszorítás [megszorítás]...

Tábla létrehozása

```
CREATE TABLE táblanév (  
    oszlopdefiníció  
    [, oszlopdefiníció]...  
    [, táblaszintű megszorítás]...  
);
```

- Táblaszintű megszorítások: Több oszlopra is vonatkozhatnak. A **CREATE TABLE** utasításban az oszlopdefiníciók után kell állniuk.
 - [**CONSTRAINT** név] megszorítás (oszlopnév [, oszlopnév]...)

Tábla létrehozása - megszorítások

- **Elsődleges kulcs megszorítás (PRIMARY KEY):**
 - Minden táblának egy elsődleges kulcsa lehet, azonban ez lehet összetett (azaz több attribútumból álló) is.
 - Az elsődleges kulcsot alkotó attribútumok között nem lehet olyan, amelynek az értéke NULL és nem szerepelhet két olyan sor a táblában, amelyek esetében az elsődleges kulcsot alkotó attribútumok értéke rendre megegyezik.
 - Összetett kulcs esetében egy-egy attribútumot tekintve lehet azonosság, de az összes attribútumot tekintve a kulcsnak egyedinek kell lennie.

Tábla létrehozása - megszorítások

- **Egyediség megszorítás (UNIQUE):**
 - Az értékeknek egyedieknek kell lenniük, de NULL érték is szerepelhet.
- Több ilyen megszorítás is szerepelhet egy táblára vonatkozóan.
- **NOT NULL megszorítás:**
 - A NOT NULL megszorítás tiltja, hogy egy oszlopban NULL érték szerepeljen.
 - A táblaszintű megszorítások között nem szerepelhet.

Tábla létrehozása - megszorítások

- **Külső kulcs megszorítás:**
 - Két tábla közti kapcsolat szabályozására szolgál.
 - Biztosítja, hogy a külső kulcsot tartalmazó tábla külső kulcsot alkotó attribútumaiban csak NULL érték vagy olyan értékek állhatnak, mely a hivatkozott tábla (amelyhez ez a tábla kapcsolódik) hivatkozott oszlopaiban ténylegesen szerepel.

Tábla létrehozása - megszorítások

- **Külső kulcs megszorítás:**

- Állhat oszlop után írva oszlopmegszorításként:

- [**CONSTRAINT** megszorításnév] **REFERENCES**
hivatkozott_tábla (oszlopnév)

- Vagy állhat az oszlopdefiníciók után táblaszintű megszorításként:

- [**CONSTRAINT** megszorításnév] **FOREIGN KEY**
(oszlopnév [, oszlopnév]...) **REFERENCES**
hivatkozott_tábla (oszlopnév [, oszlopnév]...)

Tábla létrehozása - megszorítások

- **Általános megszorítás:**

- Ezzel biztosítható, hogy egy adott oszlopba csak egy feltételnek eleget tevő értékek kerülhessenek be.
- A feltétel több oszlopra is vonatkozhat, ekkor táblaszintű megszorításként kell megadni.
 - [**CONSTRAINT** megszorításnév] **CHECK** (feltétel)

Tábla módosítása

- Oszlop hozzáadásának általános alakja:
 - **ALTER TABLE** táblanév
ADD oszlopdefiníció [, oszlopdefiníció]... ;
- Táblaszintű megszorítás hozzáadásának általános alakja:
 - **ALTER TABLE** táblanév
ADD tsz_megszorítás [, tsz_megszorítás]... ;
- Oszlop módosításának általános alakja:
 - **ALTER TABLE** táblanév
MODIFY oszlopnév új_típus új_megszorítás
[, oszlopnév új_típus új_megszorítás]... ;

Tábla módosítása



- Oszlop átnevezésének általános alakja:
 - **ALTER TABLE** táblanév
RENAME COLUMN oszlopnév **TO** új_oszlopnév;
- Oszlop törlésének általános alakja:
 - **ALTER TABLE** táblanév
DROP COLUMN oszlopnév [, oszlopnév]... ;

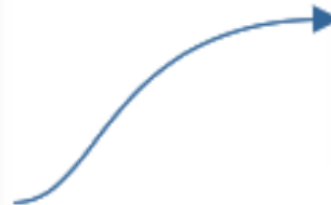
Tábla törlése


- Egy tábla törlésének általános alakja:
 - **DROP TABLE** táblanév;
- Egy tábla teljes tartalmának végleges törlése, a tábla törlése nélkül:
 - **TRUNCATE TABLE** táblanév;

Példa

- Hozzuk létre az alábbi sémát!

HALLGATO		
	NEPTUN	varchar2(6)
	VEZETEKNEV	varchar2(128)
	KERESZTNEV	varchar2(128)
	SZULETESI_DATUM	date
NOT NULL 	SZAK_AZON	number(2)



SZAK		
	AZON	number(2)
NOT NULL	MEGNEVEZES	varchar2(128)

Példa

- A HALLGATO tábla létrehozása:
 - ```
CREATE TABLE GY_HALLGATO (
 NEPTUN VARCHAR2 (6) PRIMARY KEY,
 VEZETEKNEV VARCHAR2 (128) ,
 KERESZTNEV VARCHAR2 (128) ,
 SZULETESI_DATUM DATE,
 SZAK_AZON NUMBER (2) NOT NULL
);
```

# Példa

- A SZAK tábla létrehozása:
  - ```
CREATE TABLE GY_SZAK (  
    AZON NUMBER(2),  
    MEGNEVEZES VARCHAR2(128) NOT NULL,  
    CONSTRAINT SZAK_PK PRIMARY KEY (AZON)  
);
```


Példa

- A HALLGATO tábla módosítása:
 - ```
ALTER TABLE GY_HALLGATO
ADD CONSTRAINT HALLGATO_FK
FOREIGN KEY (SZAK_AZON)
REFERENCES GY_SZAK (AZON) ;
```

# Példa

- A HALLGATO tábla létrehozása (második változat):

```
CREATE TABLE GY_HALLGATO (
 NEPTUN VARCHAR2 (6) PRIMARY KEY,
 VEZETEKNEV VARCHAR2 (128) ,
 KERESZTNEV VARCHAR2 (128) ,
 SZULETESI_DATUM DATE,
 SZAK_AZON NUMBER (2) NOT NULL
 REFERENCES GY_SZAK (AZON)
);
```

# Data Manipulation Language (DML)

- Az SQL adatmanipulációs (DML) utasításai segítségével tölthető fel és módosítható egy tábla tartalma.
  - INSERT: sorok beszúrása a táblába
  - UPDATE: sorok tartalmának frissítése
  - DELETE: sorok törlése
- Az adatmanipulációs utasítások eredménye nem kerül automatikusan rögzítésre az adatbázisban.
  - COMMIT: tranzakciók véglegesítése
  - ROLLBACK: tranzakciók visszagörgetése

# Data Manipulation Language (DML)

- INSERT: sorok beszúrását teszi lehetővé egy táblába.
- Általános alak:
  - `INSERT INTO táblanév[ (mezőnév1, mezőnév2, ... ) ]  
VALUES (érték1, érték2, ... )`
- Példa:
  - `INSERT INTO GY_SZAK VALUES (1, 'PTI BSc');`  
`INSERT INTO GY_SZAK VALUES (2, 'GI BSc');`  
`INSERT INTO GY_SZAK VALUES (3, 'MI BSc');`
- Feladat: A fentiek mintájára illesszük be a 4 – PTI MSc, 5 – GI MSc, 6 – MI MSc sorokat is az adatbázisba!

# Data Manipulation Language (DML)

- UPDATE: sorok frissítését teszi lehetővé egy táblában.
- Általános alak:
  - `UPDATE táblanév SET mezőnév = érték  
[, mezőnév = érték] [WHERE feltétel]`
- Példa:
  - `UPDATE GY_HALLGATO SET SZULETESI_DATUM =  
TO_DATE('1996-06-20', 'YYYY-MM-DD')  
WHERE NEPTUN = 'CBA321';`
- Feladat: A fentiek mintájára frissítsük Nagy Szabolcs születési dátumát 1994-09-19-re!

# Data Manipulation Language (DML)

- DELETE: sorok törlését teszi lehetővé egy táblából.
- Általános alak:
  - `DELETE FROM táblanév [WHERE feltétel]`
  - Ha nincs megadva feltétel, az összes sort törli a táblából!
- Példa:
  - `DELETE FROM GY_HALLGATO WHERE SZULETESI_DATUM IS NULL;`