

ADATBÁZIS-KEZELÉS

- TÁBLAKEZELÉS -

Tarcsi Ádám, ade@inf.elte.hu

Tábla létrehozás, eldobás

2

Tábla létrehozása:

```
CREATE TABLE tDiak
  (DAzon INT(4),
   Nev VARCHAR(50),
   Kor INT(3),
   Ofo INT(4));
```

Feltöltés adatokkal:

```
INSERT INTO tDiak VALUES (1, 'Deb Ella',
  '20', 1);
```

Tábla lekérdezése:

```
SELECT * FROM tDiak;
```

Tábla eldobása:

```
DROP TABLE tDiak;
```

3

Adattípusok

Számok: egész, valós

4

Típus	Előjeles	Előjel nélküli
TINYINT(M)	-128 -tól 127 -ig	0 -tól 255 -ig
SMALLINT(M)	-32768 -tól 32767 -ig	0 -tól 65535 -ig
MEDIUMINT(M)	-8388608 -tól 8388607 -ig	0 -tól 16777215 -ig
INT(M)	-2147483648 -tól 2147483647 -ig	0 -tól 4294967295 -ig
BIGINT(M)	-9223372036854775808 -tól 9223372036854775807 -ig	0 -tól 18446744073709551615 -ig
FLOAT(M,D)	-3.402823466E+38 -tól -1.175494351E-38 -ig	1.175494351E-38 -tól 3.402823466E+38 -ig
DOUBLE(M,D)	-1.7976931348623157E+308 -tól -2.2250738585072014E-308-ig	2.2250738585072014E-308 -tól 1.7976931348623157E+308 -ig
DECIMAL(M,D)	A számokat karakterként tárolja, ezért az M és a D összesen 254 lehet legfeljebb.	

- (M): megjelenítési szélesség (de pl.: Oracle: számjegyek száma)
- (M,D): M: hány darab számjegy lehet maximum az egészrészben, míg a D a törtrészt adja meg. M és D összesen 15 lehet.
- UNSIGNED: előjel nélküli szám
- ZEROFILL: megjelenítési szerpe van, nullával való feltöltés.
egy *int(5)* típusú oszlopnak esetén pl.: 00004
UNSIGNED is egyben
- Beíráskor ellenőrzés! Konzisztencia, típus és megszorítás megfeleltetés

Karakteres típusok

5

- ❑ CHAR (hossz) : fix hosszú szöveg. A hossz max. 255 lehet.
- ❑ VARCHAR (hossz) : max n hosszú szöveg. A hossz max. 255 lehet.
- ❑ TEXT (hossz) : hosszabb szövegek tárolására. Max. 65535 bájtot tud tárolni.
- ❑ BLOB: binárisan tárolt nagyméretű objektum. Max. 65535 bájtot tud tárolni
- ❑ MEDIUMBLOB, MEDIUMTEXT (16777215 bájt)
- ❑ LONGBLOB, LONGTEXT (4294967295 bájt, 4GB)

Dátum és idő típusok

6

Típus	Kezdőérték és forma
DATETIME	'0000-00-00 00:00:00'
DATE	'0000-00-00'
TIMESTAMP	'0000-00-00 00:00:00'
TIME	'00:00:00'
YEAR	0000

- a DATETIME és a TIMESTAMP 'év-hónap-nap óra:perc:másodperc' alakú.

Példa

7

```
mysql> SELECT NOW();
+-----+
| NOW() |
+-----+
| 2011-09-23 16:41:30 |
+-----+
1 row in set (0.00 sec)
mysql> CREATE TABLE ido(dt datetime, d date, t time, y year);
Query OK, 0 rows affected (0.01 sec)
mysql> INSERT INTO ido VALUES (NOW(),NOW(), NOW(), NOW());
Query OK, 1 row affected, 1 warning (0.00 sec)
mysql> SELECT * FROM ido;
+-----+-----+-----+-----+
| dt          | d          | t          | y          |
+-----+-----+-----+-----+
| 2011-09-23 16:44:51 | 2011-09-23 | 16:44:51 | 2011      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

8

CREATE TABLE

CREATE TABLE

9

```
CREATE TABLE <név>
(<oszlopnév1> <oszloptípus1>,
 <oszlopnév2> <oszloptípus2>,
 ... ,
 <oszlopnévn> <oszloptípusn>
 [CONSTRAINT <megszorításnév> <típus>
 <típustól függő vég>...])
```

Megszorítások

10

Tábla szintű megszorítások:

- ▣ elsődleges kulcs
- ▣ idegen kulcs
- ▣ egyediség
- ▣ logikai feltétel
- ▣ (AUTO_INCREMENT)

Mezőkre vonatkozó megszorítások:

- ▣ alapértelmezett érték
- ▣ kötelezőség (nem null érték)

Megszorítás - CONSTRAINT

11

1. Elsődleges kulcs – PRIMARY KEY

- egyedi azonosító
- példa: tTanar táblában TAzon, tDiak tablában DAzon
- szokásos elnevezés: pk_tablanev

```
CONSTRAINT <megszorításnév> PRIMARY KEY  
( <mezőnév1>, ..., <mezőnévk> )
```

CONSTRAINT

12

2. Idegen kulcs – FOREIGN KEY

- Másik táblában (hivatkozott tábla) elsődleges kulcsként előforduló értékeket vehet fel
- Nem törölhető olyan rekord, amelyre hivatkoznak.
- példa: tDiak táblában Ofo mező
- szokásos elnevezés:
fk_hivatkozotablanév_hivatkozotttablanév
(fk_tDiak_tTanar)

```
CONSTRAINT <megszorításnév> FOREIGN KEY  
(<mezőnév1>, ..., <mezőnévk> ) REFERENCES  
<hivatkozotttáblanév> (<újmezőnév1>, ...,  
<újmezőnévk>)
```

CONSTRAINT

13

3. Egyediség - UNIQUE

- adott mezőben minden értéknek különbözőnek kell lennie
- szokásos elnevezés: uq_tablanev

```
CONSTRAINT <megszorításnév> UNIQUE (<mezőnév1>, ..., <mezőnévk>)
```

CONSTRAINT

14

4. Feltétel– CHECK

- Adott mező mely halmazból vehesse fel az értékeit
- szokásos elnevezés: ck_tablanev
- Sokféle feltétel: AND, OR, BETWEEN, SELECT, mezőnév>0, ...
- I. **CONSTRAINT <megszorításnév> CHECK (<mezőnév1> IN (<halmaz1>), ..., <mezőnévk> IN (<halmazk>))**
- II. **IN (<halmaz>)**

Példa:

```
CONSTRAINT ck_tablanev CHECK (Nev IN('Pistike','Ferike'))
```

```
CONSTRAINT ck_tablanev2 CHECK (Jegy IN(1,2,3,4,5))
```

5. Alapértelmezett érték - DEFAULT

- Ha egy rekordnál az adott mező értékét nem adjuk meg, akkor az alapértelmezett értéket veszi fel értékként
- DEFAULT <érték vagy kifejezés>

6. Nem nullérték - NOT NULL

- Lehetséges értelmezései
 - Ismeretlen érték: pl.: nem ismerem a címét
 - Alkalmazhatatlan érték: pl.: férfiak esetén a „Leánykori név” mező kitöltése
 - Visszatartott érték: nincs rá jogom, titkos adat esetén
- Elsődleges kulcs NOT NULL

Példa

17

```
CREATE TABLE tTanar
(TAzon INT(5) NOT NULL,
Nev VARCHAR(50) NOT NULL,
CONSTRAINT pk_tTanar PRIMARY KEY (TAzon),
CONSTRAINT uq_tTanar UNIQUE (Nev))
ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci;
```

```
CREATE TABLE tDiak
(DAzon INT(5) NOT NULL AUTO_INCREMENT PRIMARY KEY,
Nev VARCHAR(50) NOT NULL,
Kor INT(3) DEFAULT 18,
Ofo INT(5) NOT NULL,
CONSTRAINT fk_tDiak_tTanar FOREIGN KEY (Ofo) REFERENCES
    tTanar (TAzon),
CONSTRAINT ck_tDiak CHECK (Kor BETWEEN 1 AND 100))
ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci;
```

18

ALTER TABLE

ALTER TABLE

19

- **Tárolómotor módosítás:**

```
ALTER TABLE tTanar ENGINE = InnoDB;
```

- **Karakterkódolás módosítása:**

```
ALTER TABLE tTanar CONVERT TO  
CHARACTER SET charset_name;
```

Táblamódosítás - ALTER TABLE <tábla név> ...;

20

- **Új oszlop hozzáadása a táblához (a tábla „végére”)**
 - ALTER TABLE <tábla név> ADD COLUMN <oszlop név> <oszlop típus>;
 - Adjunk egy Kor nevű egész típusú oszlopot a tTanar táblához!
 - ALTER TABLE tTanar ADD COLUMN Kor INTEGER;
- **Egy oszlop típusának módosítása**
 - ALTER TABLE <tábla név> MODIFY / CHANGE <oszlop név> <új oszlop típus>;
 - Típusproblémák!!!
 - Legyen a tDiak tábla Nev mezőjének típusa 100 hosszú szöveg!
 - ALTER TABLE tDiak MODIFY Nev VARCHAR(100);
- **egy oszlop törlése**
 - ALTER TABLE <tábla név> DROP <oszlop név>;
 - (Ha az oszlop például elsődleges kulcs, akkor a művelet hibához vezet.)
 - Töröljük a tTanar táblából a Kor oszlopot!
 - ALTER TABLE tTanar DROP Kor;

Megszorítások módosítása

21

- **Tábla szintű megszorítások:**

```
ALTER TABLE <tábla név> ADD  
CONSTRAINT <név> <típus>  
<(oszlop)>;
```

- **Feladat:** Adjunk egy olyan megszorítást a tTanar táblához, aminek következtében nem tárolhatunk két ugyanolyan nevű tanárt!

```
ALTER TABLE tTanar ADD CONSTRAINT  
uq_tTanar UNIQUE (Nev);
```

Megszorítások módosítása – 2.

22

- **Megszorítás törlése:**

```
ALTER TABLE <tábla név> DROP CONSTRAINT  
<megszorítás név>;
```

- **Feladat: Dobjuk el az előbbi megszorítást!**

```
ALTER TABLE tTanar DROP CONSTRAINT  
uq_tTanar;
```

vagy

```
ALTER TABLE tTanar DROP INDEX uq_tTanar
```

Táblák eldobása

23

```
DROP TABLE <táblanév>;
```