

SQL kulcsszavak (külön nézzük majd a TÁBLÁK MEZŐINEK módosítását | ATTRIBÚTUMOK módosítását)

CREATE	Létrehozás
SELECT	Kiválasztás
UPDATE	Adatok frissítése, módosítás a táblákban
DELETE	Adatok törlése a táblákból
INSERT INTO	Rekord beszúrása a táblába
DROP	Tábla vagy adatbázis törlése

UPDATE és DELETE NEM VISSZAVONHATÓ! Ha valamit elrontottunk, akkor az úgy is marad!

Nézzük először a TÖRLÉST:

A DELETE parancs törli a tábla feltételnek megfelelő sorait (NEM VISSZAVONHATÓ):

(feltétel megadása nélkül mindent)

DELETE FROM táblanév

[WHERE feltétel]

Pl.:

DELETE FROM tanulo

WHERE evfolyam=12;

Másodjára nézzük a rekord(ok) MÓDOSÍTÁSÁT:

Az UPDATE parancs módosítja a tábla meghatározott mezőinek értékét (NEM VISSZAVONHATÓ):

UPDATE táblanév

SET (attr1=adat1,attr2=adat2)

[WHERE feltétel]

UPDATE dolgozo SET fizet=1.1*fizet

WHERE fizet<200000;

Harmadjára vizsgáljuk meg az új rekord BESZÚRÁSÁT:

Az INSERT INTO parancs új rekordot szúr be a táblába:

Ha minden értéket megadunk, akkor a szintaktika a következő:

INSERT INTO táblanév(attribútumlista)

VALUES(értéklista)

INSERT INTO tanulo (kod,nev,evf,oszt)

VALUES(009,"Nagy János","12","b");

Használhatjuk a NULL értéket

Az új rekord az utolsó után tárolódik

Figyelni kell az ÉRTÉKLISTÁNÁL az értékek sorrendjére. Meg kell egyeznie az ATTRIBÚTUMLISTÁNÁL alkalmazott sorrenddel!

Ha csak pár mező értékét adjuk meg (NEM MINDET), akkor a szintaktika a következő:

```
INSERT INTO tábla (vezeteknev, keresztnév) VALUES ('Kiss', 'Janos');
```

ebben az esetben is fontos a mezők (általunk definiált) sorrendje

Végül hozzunk létre adatbázist és táblákat.

Adatbázis létrehozásnál ugye figyelni kell a karakterkódolásra és a nyelvi illesztésre.

```
CREATE DATABASE adatbázis neve DEFAULT CHARACTER SET utf8 COLLATE utf8_hungarian_ci;
```

```
CREATE DATABASE orszagok DEFAULT CHARACTER SET utf8 COLLATE utf8_hungarian_ci;
```

TÁBLÁK létrehozásakor sok mindenre figyelniünk kell.

Adattábla létrehozása

```
CREATE TABLE táblanév
```

```
(attribnev1 adattip1 (méret1) [NOT NULL]
```

```
(attribnev2 adattip2 (méret2) [NOT NULL];
```

```
CREATE TABLE tanulo
```

```
(kod INTEGER(8) , nev VARCHAR(16), evf CHAR(3), oszt INT(1), pont(INT(5)));
```

Ahol a **kod** a mező neve, az **INTEGER** a mező típusa és maximum 8 számjegyet tartalmazhat stb. A példában egyik mező megadását se jelöltük meg KÖTELEZŐEN KITÖLTENDŐNEK. Adjuk meg a **kod** mezőt KÖTELEZŐNEK.

```
CREATE TABLE tanulo
```

```
(kod INTEGER(8) NOT NULL, nev VARCHAR(16), evf CHAR(3), oszt INT(1), pont(INT(5)));
```

Adjuk meg, hogy a **kod** mező értéke automatikusan növekedjen minden létrehozott rekordnál 1-el! Ilyenkor a rekord feltöltésekor NEM KELL megadni a KOD értékét, azt automatikusan fogja felvenni! Az automatikus növeléshez az is kell, hogy elsődleges kulcs legyen az a mező, amelyre alkalmazni szeretnénk. (PRIMARY KEY) És egy táblában CSAK EGY mezőre lehet beállítani az **AUTO_INCREMENT** tulajdonságot!

```
CREATE TABLE tanulo
```

```
(kod INTEGER(8) NOT NULL AUTO_INCREMENT, nev VARCHAR(16), evf CHAR(3), oszt INT(1),  
pont(INT(5)), PRIMARY KEY (kod));
```

Az IDEGEN KULCSHOZ egy másik táblát vegyünk mintának, fáradt vagyok új dolgok kitalálásához.

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

Az Orders táblában a PersonID mező lesz az idegen kulcs, aminek a segítségével összekapcsolható az Orders és a Person tábla, mint látható a Person táblában is létezik a PersonID mező! A REFERENCES paranccsal HIVATKOZUNK az IDEGEN TÁBLA NEVÉRE és a megfelelő MEZŐJÉRE!

MEGSZORÍTÁSOK:

Név	Leírás
NOT NULL	az oszlop értéke nem lehet NULL (üres)
UNIQUE	az oszlop összes értékének egyedinek kell lenni
PRIMARY KEY	a NOT NULL és a UNIQUE kombinációja, minden rekordnak egyértelműen beazonosíthatónak kell lennie
FOREIGN KEY	egyértelműen beazonosíthatónak kell lennie egy rekordnak egy másik táblában
CHECK	az oszlop értékének meg kell felelnie a meghatározott feltételnek
DEFAULT	egy alapértelmezett értéket állít be az oszlopnak, ha a felhasználó nem adott volna meg értéket

A MEZŐK által gyakran felvett adattípusok:

INT(*n*): egész szám, *n* darab számjegy vagy hosszabban írva INTEGER(*n*)

VARCHAR(*hossz*): változó hosszú, de maximálisan *hossz* hosszú karaktersorozat

CHAR(*hossz*): *hossz* hosszú karaktersorozat, fix hosszú!!!

DATE: dátum

TIME: időpont

DATETIME : dátum és időpont

DECIMAL: Lebegőpontos szám (egészek száma, tizedesek száma)