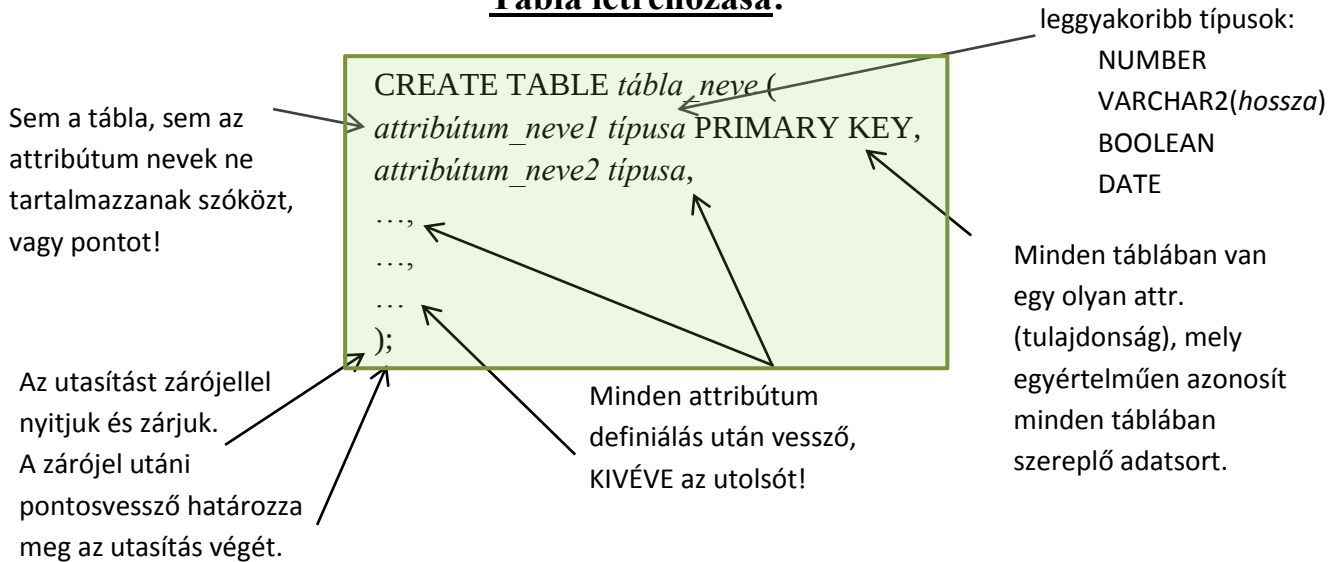


Adatbázis kezelés

Összefoglaló

Tábla létrehozása:

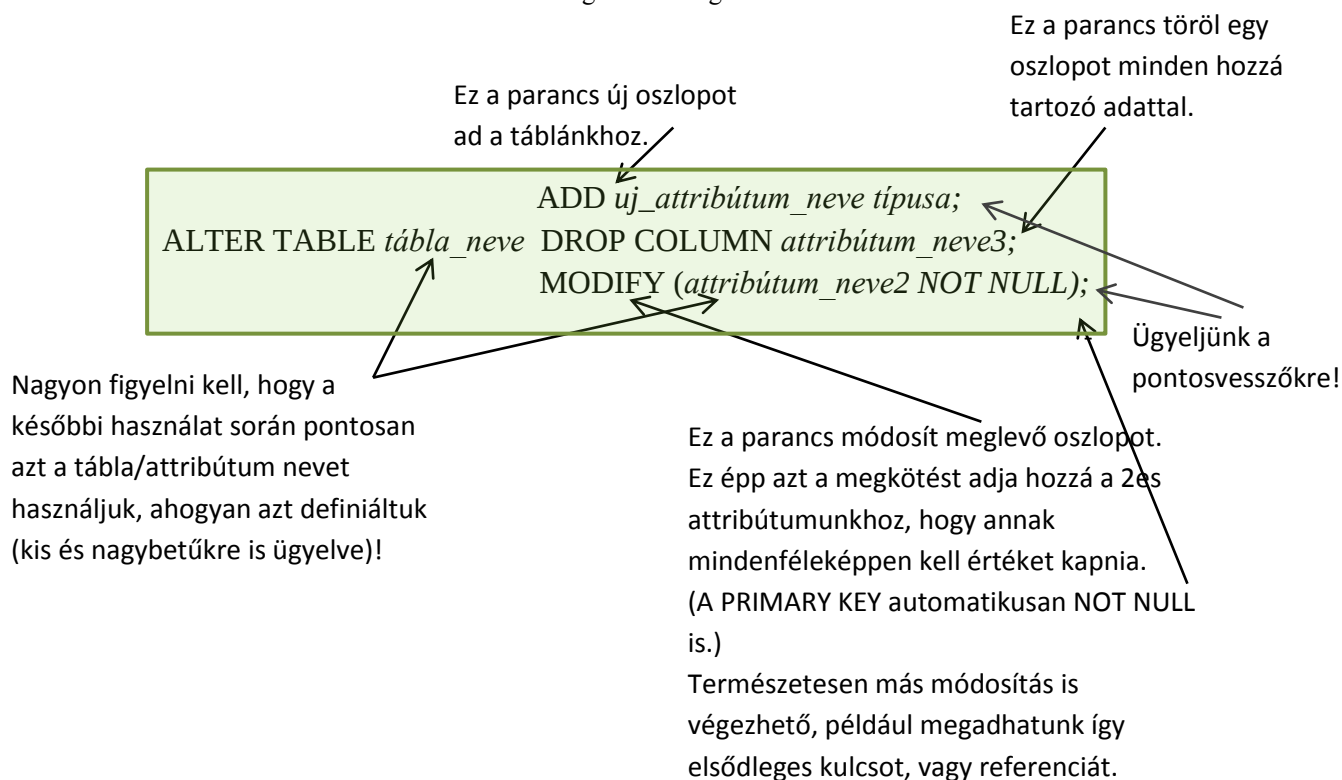


Tábla módosítása:

Ha egy tábla létrejött, ugyanazzal a névvel másik táblát nem lehet létrehozni.

Ha egy létező táblán módosítanánk, és a táblát létrehozó parancsban átírjuk a kívánt dolgot, majd lefuttatjuk azt ismét, hibát kapunk, hiszen a parancs nem csak a módosítást hajtja végre, hanem létre akarja hozni a táblát a módosított paraméterekkel. Erre egy megoldás, hogy ha módosítani szeretnénk valamit, a meglévő táblát töröljük (lásd később), majd létrehozzuk a módosítottat.

Azonban van sokkal elegánsabb megoldás is:



Hivatkozás más tábla elemeire

Adatbázisunkban általában több táblában tárolunk adatokat, melyek valahogyan mégis összetartoznak. Például a gyakorlaton látott példában egy táblában tároljuk a betegek adatait és egy másikban a kezeléseket. A feladatunk valahogy összekötni ezt a két táblát, eltávolítva ezzel azt, hogy mely páciens számára milyen kezelés van előírva. Ennek megoldására –például- a kezeléseket tartalmazó táblában elhelyezünk egy oszlopot, amibe azt írjuk, hogy a kezelés melyik beteghez tartozik. Ahhoz, hogy ez valódi, visszakereshető összeköttetést jelentsen, ennek az adatnak olyannak kell lenni, ami szerepel a betegeket tartalmazó táblában. Erre használjuk a referenciát.

Tábla létrehozásakor:

```
CREATE TABLE tabla_nev (  
  atr1 típus PRIMARY KEY,  
  atr2 típus REFERENCES tábla_neve(összekötő_attribútum),  
  ...,  
  ...  
);
```

Az új táblában az attribútum neve akármilyen lehet, tehát nem kell, hogy ugyanaz legyen, mint a meghivatkozott attribútum.

Nagyon fontos, hogy az összekötő attribútum (*külső, vagy idegenkulcs*) típusa ugyanaz legyen, mint amire hivatkozunk vele!

annak az attribútumnak a neve, amire hivatkozni szeretnénk

annak a táblának a neve, amiben van az attribútum, amire hivatkozni szeretnénk

Tábla módosításakor:

```
ALTER TABLE tabla_nev ADD FOREIGN KEY attr3  
REFERENCES tábla_neve(összekötő_attribútum);
```

Létrehoz a táblában egy új attribútumot, és az lesz az idegen kulcs.

Vagy:

```
ALTER TABLE tabla_nev MODIFY attr2 FOREIGN KEY  
REFERENCES tábla_neve(osszekoto_attributum);
```

A táblában egy, már meglevő attribútumot (*attr2*) tesz idegen kulccsá.

Adatok táblába töltése

Eddig egy tábla fejlécét szerkesztettük meg, most pedig a megfelelő adatokat tesszük majd bele.

```
INSERT INTO tabla_nev(attr1, attr2...) VALUES ('adat attr1-hez', 'adat attr2-höz', szám, '...');
```

A feltöltendő tábla
neve

Az attribútumok megadása, melyekhez szeretnénk adatot rendelni. Nem muszáj, hogy az összes oszlopot kitöltsük, DE azokat, melyek PRIMARY KEY, vagy NOT NULL megkötéssel bírnak, igen!
Ha nem adjuk meg, mely attribútumokat töltjük fel, az egyenértékű azzal, mint hogy ha azt írnánk, hogy mindet, és abban a sorrendben, melyben az attribútumok a tábla létrehozásakor szerepeltek.

Az adatokat olyan sorrendben kell megadni, amilyenben a VALUES előtt az attribútumokat. Szöveges adatokat '...' közé kell írni, a számokat nem.

```
INSERT INTO tabla_nev VALUES ('adat attr1-hez', 'adat attr2-höz', szám, '...');
```

Ügyeljünk a
pontosvesszőkre!

Dátum adat hozzáadása: VALUES után a megfelelő helyre a következő kerül:
to_date('2012-02-22' 'yyyy-mm-dd')

Tábla törlése

```
DROP TABLE tabla_nev PURGE;
```

```
DROP TABLE tábla_név CASCADE CONSTRAINT;
```

Ha olyan táblát próbálunk törölni, melyre más táblák hivatkoznak, hibaüzenetet kapunk, hiszen ezzel a művelettel „szétesne” az adatbázisunk.
Emiatt ha egy nagyobb adatbázisból törölünk táblát, figyelni kell a sorrendre: először töröljük az, mely hivatkozik valamely másikkra, majd azt, amire hivatkozott a korábbi.
Összetett adatbázisokban ez elég bonyolult tud lenni. Ezen segít ez a parancs, mely kitöröl minden hivatkozást is, melyben szerepel a törlendő táblára.

A parancs a PURGE nélkül is működne, így azonban a tábla nem a szemétkébe, hanem az éterbe kerül, megóvva minket a későbbi esetleges memóriaszeméttől.

Ügyeljünk a
pontosvesszőkre!