

DDL-utasítások

A táblák létrehozását, szerkezetének módosítását, oszlopok beszúrását, oszloptípusok módosítását, törlését, átnevezését végző SQL-utasításokat a DDL (Data Definition Language) nyelv tartalmazza. Megjegyezzük, hogy a DDL-utasításokat a táblákon kívül egyéb adatbázis-objektumokra (nézettábla, index, szinonima stb.) is használjuk.

Tábla létrehozása (közvetlen létrehozás)

Az utasítás alakja:

```
CREATE TABLE [felhasználó.]táblanév
  (oszlopnév adattípus [DEFAULT kifejezés] [oszlopmegszorítások]
  [, oszlopnév adattípus [DEFAULT kifejezés] [oszlopmegszorítások]...]
  [, táblamegszorítások]);
```

ahol az *adattípus* (többek között) a következők valamelyike lehet:

NUMBER(<i>m</i> [, <i>t</i>])	változó hosszúságú numerikus adat, ahol <i>m</i> a mezőszélesség, mely az esetleges előjelet és tizedespontot is tartalmazza, <i>t</i> ebből a tizedesek száma,
CHAR(<i>n</i>)	kötött hosszúságú karaktersorozat, vagyis a definícióban megadottnál rövidebb karaktersorozatokat balra tömöríti és jobbról szóköz karakterekkel tölti fel, az <i>n</i> maximális értéke 2000, alapértelmezett értéke 1,
VARCHAR2(<i>n</i>)	változó hosszúságú karaktersorozat, vagyis a definícióban megadottnál rövidebb karaktersorozatokat a tényleges hosszúságuk szerint kezeli (például jeleníti meg), az <i>n</i> maximális értéke 4000,
DATE	dátumok tárolására alkalmas típus,

továbbá az *oszlopmegszorítások* helyén egy vagy több, szóközzel elválasztott *oszlopmegszorítás*, a *táblamegszorítások* helyén pedig egy vagy több, vesszővel elválasztott *táblamegszorítás* állhat.



Megjegyzés

- A továbbiakban általában nem jelöljük a táblanév előtt a felhasználóra való hivatkozás lehetőségét, de azért az természetesen minden esetben lehetséges.
- Az *oszlopmegszorítás* és a *táblamegszorítás* fogalmát lásd később (jelen fejezet *Integritási megszorítások* c. pontja).
- A további alkalmazható adattípusokat lásd [12], [16] és [26].

Tábla létrehozása allekérdezéssel (származtatott létrehozás)

```
CREATE TABLE táblanév
  AS allekérdezés;
```

**Megjegyzés**

Ezzel az utasítással készíthetünk egy tábláról másolatot. Például az emp táblából egy emp1 nevű táblát a

```
CREATE TABLE emp1
AS SELECT * FROM emp;
```

utasítással hozhatunk létre. Ügyeljünk arra, hogy az így létrehozott tábla csak adattartalmában azonos az eredetivel, annak megszorításai közül azonban *nem mindegyiket* tartalmazza (lásd később).

Tábla módosítása

ÚJ OSZLOP HOZZÁADÁSA

```
ALTER TABLE táblanév
ADD (oszlopnév adattípus [DEFAULT kifejezés] [oszlopmegszorítások]);
```

OSZLOP MÓDOSÍTÁSA

```
ALTER TABLE táblanév
MODIFY (oszlopnév [adattípus] [DEFAULT kifejezés] [oszlopmegszorítások]);
```

OSZLOP TÖRLÉSE (ELDOBÁSA)

```
ALTER TABLE táblanév
DROP COLUMN oszlopnév [CASCADE CONSTRAINTS];
```

**Megjegyzés**

A CASCADE CONSTRAINTS opció esetén az oszlophoz rendelt esetleges (PRIMARY KEY, FOREIGN KEY) kulcsmegszorítások ellenére megtörténik a törlés (lásd később).

OSZLOP ÁTNEVEZÉSE

```
ALTER TABLE táblanév
RENAME COLUMN RégiOszlopnév TO ÚjOszlopnév;
```

**Megjegyzés**

Csak az Oracle 9i verziójától alkalmazható az oszlop átnevezése.

TÁBLA TÖRLÉSE (ELDOBÁSA)

DROP TABLE táblanév;

TÁBLA ÁTNEVEZÉSE

RENAME RégiTáblanév TO ÚjTáblanév;

Integritási megszorítások

Az integritási megszorítások az adattáblára vonatkozó olyan szabályok, melyek a tábla oszloppaira vonatkoznak (függőségek). Ezek biztosítják, hogy a DML-utasítások a szabályok megsértése esetén ne hajtódjanak végre.

Az integritási megszorítások a tábladefiníció részét alkotják, de utólag is megadhatók. Utólag módosíthatók, törölhetők, és lehetőség van arra is, hogy ideiglenesen letiltjuk, illetve újra engedélyezzük.

Az integritási megszorítások, mint a nevük is mutatja, az adatbázisok integritásának (hitelességének, megbízhatóságának) legfontosabb eszközei. Általában csak egyetlen táblára vonatkoznak (kivételek a FOREIGN KEY, azaz idegen kulcs és a REFERENCES, azaz hivatkozási megszorítás), és jellemzően a tábla minden sorára érvényes szabályok megfogalmazására alkalmasak csupán. A gyakorlatban előfordul, akár az adatbázis tervezésekor (a normalizálás során), akár már használatba vett adatbázisra vonatkozó alkalmi felhasználói igények esetén, hogy szükségesnek mutatkozik több táblára, vagy csak a tábla néhány sorára vonatkozó szabályok érvényesítése. Az ilyen feladatok megoldására használjuk (egyebek mellett) a triggereket (lásd 10. fejezet).

MEGSZORÍTÁSOK DEFINÍCIÓJA**Oszlopmegszorítás**

A tábladefinícióban, az oszlopdefiníció részeként szerepel, de utólag is megadható. Segítségével szabályt definiálhatunk az adott oszlopra.

Táblamegshorítás

A tábladefiníció részeként, az oszlopdefiníció után következik, de utólag is megadható. Segítségével szabályt definiálhatunk bármely oszlopra vagy oszlopok kombinációjára.

Az oszlop- és táblamegshorítások megadása

[CONSTRAINT MegszorításNév] MegszorításTípus

**Megjegyzés**

Ha egy megszorításnak nem adunk nevet, akkor az Oracle-rendszer automatikusan generál számára egy SYS_C00xxxx alakú nevet, ahol xxxx egy sorszám.

A megszorítások típusai

- **NULL** *NULL megszorítás.* Az adott oszlop tartalmazhat NULL értéket. Ez az alapértelmezés. Csak oszlopmegszorításban szerepelhet.

Megadása:

NULL

- **NOT NULL** *NOT NULL megszorítás.* Az adott oszlop *nem* tartalmazhat NULL értéket. Csak oszlopmegszorításban szerepelhet.

Megadása:

NOT NULL

- **UNIQUE**, *Egyedi kulcs megszorítás.* Az oszlop minden sorában különböző értéknek kell lennie. Tartalmazhat NULL értéket is.

Megadása táblamegszorításként:

UNIQUE

(*vonatkozó_oszlopnév* [, *vonatkozó_oszlopnév*]...)

ahol a *vonatkozó_oszlopnév* az adott tábla oszlopát jelöli, és több oszlop kijelölése esetén az egyediség csupán a megadott oszlopokból álló vektor értékeire, és nem az egyes oszlopok értékeire vonatkozik.

Megadása oszlopmegszorításként:

UNIQUE

- **PRIMARY KEY** *Elsődleges kulcs megszorítás.* Az elsődleges kulcs biztosítja, hogy minden sor egyértelműen azonosítható legyen, így minden sorban különböző értékű, és nem tartalmazhat NULL értéket.

Megadása táblamegszorításként:

PRIMARY KEY

(*vonatkozó_oszlopnév* [, *vonatkozó_oszlopnév*]...)

ahol több oszlop kijelölése adja az összetett kulcsot (az egyediségre vonatkozóan lásd a fenti megjegyzést).

Megadása oszlopmegszorításként:

PRIMARY KEY

- **REFERENCES** *Hivatkozási megszorítás.* Egyaránt vonatkozhat saját vagy más táblára. Az általa jelölt oszlop értékeinek szerepelnie kell a hivatkozott tábla hivatkozott (elsődleges vagy egyedi kulcs) oszlopának értékei között. Önállóan csak oszlopmegszorításban szerepelhet, egyébként az idegen kulcs megszorítás részeként.

Megadása az idegen kulcs megszorítás részeként:

(lásd alább az idegen kulcs megszorításnál),

Megadása önálló megszorításként:

REFERENCES *hivatkozott_táblanév*

(*hivatkozott_oszlopnév*) [ON DELETE CASCADE]

ahol az ON DELETE CASCADE megadása esetén a hivatkozási integritás biztosítása érdekében a megadott oszlop értékének törlése esetén a törlés továbbgyűrűzhet a hivatkozott oszlop vonatkozó értékére is.

- **FOREIGN KEY** *Idegen kulcs (külső vagy másodlagos kulcs) megszorítás.*

Egy oszlopot, illetve oszlopok egy halmazát jelöli ki idegen kulcs kapcsolatként a hivatkozott tábla hivatkozott (nem feltétlenül azonos nevű, elsődleges vagy egyedi kulcs) osz-

lopára, illetve oszlopaire vonatkozóan. Az idegen kulcs megszorítás (speciális esetként) használható saját táblára is. Ekkor „idegen kulcsként” természetesen a saját elsődleges kulcsra kell hivatkozni (lásd 5.3. példa). Csak táblamegszorításban szerepelhet.

Megadása:

```
FOREIGN KEY hivatkozott_táblanév
  (vonatkozó_oszlopnév [, vonatkozó_oszlopnév]...)
REFERENCES hivatkozott_táblanév
  (hivatkozott_oszlopnév [, hivatkozott_oszlopnév]...)
[ON DELETE CASCADE]
```

ahol az ON DELETE CASCADE megadása esetén a hivatkozási integritás biztosítása érdekében egy saját (vonatkozó) oszlop értékének törlése esetén a törlés továbbgyűrűzhet a külső (hivatkozott) oszlopok megfelelő értékeire is.

- CHECK *Feltételes megszorítás*. Oszlop- és táblamegszorításban egyaránt szerepelhet.

Megadása:

```
CHECK (feltétel)
```

ahol a *feltétel* tetszőleges, az adott tábla oszlopaire vonatkozó logikai kifejezés lehet, és annak igaz (TRUE) értéke esetén engedélyezi az adatbevitelt, illetve módosítást. Ügyelni kell arra, hogy a megadott logikai kifejezés a teljes táblára érvényes legyen.



Megjegyzés
Az adattábláknak a

```
CREATE TABLE táblanév AS allekérdezés;
```

utasítással való másolása esetén a megszorítások közül csak a NOT NULL másolódik át, a megszorítások nem.

MEGSZORÍTÁS HOZZÁADÁSA

```
ALTER TABLE táblanév
  ADD Oszlopmegszorítások;
```

MEGSZORÍTÁS TÖRLÉSE

```
ALTER TABLE táblanév
  DROP CONSTRAINT MegszorításNév [CASCADE];
```

ahol a CASCADE opció megadása esetén minden olyan megszorítás törlődik, mely függ a törölt megszorítástól.

MEGSZORÍTÁSOK LEKÉRDEZÉSE

A megszorítások adatai az adatszótár nézeteiből kaphatók meg. A megszorítások nevét és típusát a `user_constraints`, a megszorítások nevét és a hozzájuk tartozó oszlopok nevét pedig a `user_cons_columns` adatszótár nézetből kérdezhetjük le.

A lekérdezések általános alakja:

```
SELECT constraint_name, constraint_type
   FROM user_constraints
  WHERE table_name = 'táblanév';
```

illetve

```
SELECT constraint_name, column_name
   FROM user_cons_columns
  WHERE table_name = 'táblanév';
```

MEGSZORÍTÁS FELFÜGGESZTÉSE (LETILTÁSA)

Egy megszorítást ideiglenesen felfüggeszthetünk a `DISABLE` utasításrésszel. A `CASCADE` opció használata esetén egy másik táblára ható (tovagyűrűző) integritási megszorítást is felfüggeszthetünk.

A megszorítás letiltásának általános alakja

```
ALTER TABLE táblanév
  DISABLE CONSTRAINT MegszorításNév [CASCADE];
```

ahol a `CASCADE` opció megadása esetén minden olyan megszorítás felfüggesztésre kerül, mely függ a felfüggesztett megszorítástól.

MEGSZORÍTÁS ENGEDÉLYEZÉSE

A tábla definíciójában megadott, de felfüggesztett megszorításokat engedélyezhetjük az `ENABLE` utasításrésszel. (Egyedi vagy elsődleges kulcs engedélyezésekor újragenerálódik az egyedi index a táblához.)

A megszorítás engedélyezésének általános alakja

```
ALTER TABLE táblanév
  ENABLE CONSTRAINT MegszorításNév;
```