

I. ADATMANIPULÁCIÓS NYELV (DML)

INSERT	Új sor(ok) felvitele táblába, nézetbe
UPDATE	Táblában meglévő sor(ok) módosítása
DELETE	Táblában meglévő sor(ok) törlése

Megjegyzések:

- A karbantartó műveletek által kezelt adatoknak ki kell elégíteni az integritási megszorítási szabályokat
- Egy-egy karbantartó utasítással csak egy táblát kezelhetünk ORACLE 8i-ig. Az utasításokban szereplő alselectek, más táblá(k)ra is hivatkozhatnak és korrelációban lehetnek a karbantartó utasítással kezelt táblával
- A rendszer a táblákhoz kapcsolódó index-objektumokat automatikusan karbantartja

I. Adatok felvitele

INSERT INTO {táblanév | nézet} [aliasnév] [(oszlopnév, ...)]
{VALUES (érték, ...) | alselect};

- Oszlopnév - érték pároknak típusban és értéktartományban illeszkedni kell
- Ha az oszlopnév felsorolás elmarad, akkor az összes oszlopra vonatkozik az értékadás a CREATE TABLE utasításban megadott sorrendben
- NULL értékek bevitele ott lehetséges, ahol ez megengedett, nem tiltja megszorítás
- Speciális értékek megadhatók kifejezéssel: SYSDATE+7, USER, DEFAULT stb.
- Dátumkonstans megadása az alapértelmezett dátumformátum szerint vagy TO_DATE konverziót alkalmazva lehetséges. Ha időpontot nem adunk meg, az alapértelmezett idő éjféli
- Az alselect más táblából nyert adatok felvitelét teszi lehetővé

a) Adatfelvitel a tábla minden oszlopába:

```
INSERT INTO telephely VALUES (99, 'IRODAK', 'DEBRECEN');  
Hiba:      TKOD:      70      elsődleges kulcs megsértés  
          TKOD:      NULL     elsődleges kulcs nem lehet NOT NULL
```

b) Adatfelvitel hiányos listával, NULL kulcsszóval

```
INSERT INTO alkalmazott  
      (akod, anev, beosztas, tkod, fizetes)  
VALUES (2000, 'UJ DOLGOZO', 'IGAZGATO', '30', 200000);  
Megjegyzés:      a belépés a SYSDATE lesz
```

Dátum és időérték beszúrása:

```
... VALUES (... TO_DATE('1982-01-31 08:00', 'YYYY-MM-DD HH24:MI'),...)
```

c) Adatfelvitel másik táblából

```
1, Hozzuk létre a TELVIDEK táblát, mely azonos szerkezetű TELEPHELY-el, de  
   üres  
2, Vigyük fel bele a vidéki telephelyeket névsor szerinti sorrendben  
CREATE TABLE telvidek  
      AS SELECT * FROM telephely  
      WHERE 1=2;  
INSERT INTO telvidek  
      SELECT * FROM telephely  
      WHERE varos<>'BUDAPEST'  
      ORDER BY tnev;
```

d) Adatfelvitel skalár értéket adó alselect használatával:

Vigyünk fel egy új könyvet a KONYV táblába, melynek

Szerzője az eddig felvitt legrövidebb nevű szerző

Ára a könyvek eddigi átlagárának 60 %-a

A többi adat tetszőleges (KKOD és CIM NOT NULL)

```
INSERT INTO konyv  
VALUES (2000,(SELECT szerzo FROM konyv  
      WHERE LENGTH(szerzo)=  
      (SELECT MIN(LENGTH(szerzo)) FROM konyv)),  
      'UJ KONYV', 'IRODALOM', 1,  
      (SELECT AVG(ar) FROM konyv)*0.6);
```

- Az `alselect` `select` listájának ugyanannyi oszlopa kell hogy legyen, mint a `VALUES` utasításrész oszlop listájának
- Az alaptábla oszlopaira vonatkozó összes szabályt követnünk kell, hogy az `INSERT` utasítás sikeresen működjön. Például, nem használhatunk többszörös alkalmazotti azonosítót, és nem hagyhatunk ki egyetlen értéket sem a kötelezően nem nulla oszlopra vonatkozóan

II. Adatok módosítása

UPDATE [séma.]{táblanév | nézet} [aliasnév]

SET {oszlop=kifejezés | oszlop=(alselect) | (oszlopnév,...)=(alselect)}, ...
[**WHERE feltétel**];

SET: megadja, hogy mely oszlopok régi értékét mely új értékre kell módosítani
Érték: `NULL`, `DEFAULT`, `USER`, skalár értéket adó `alselect` is lehet
WHERE feltétel: a módosításban résztvevő sorokat válogatja ki

A vidéki dolgozóknak adjunk 10%-os fizetésemelést:

```
UPDATE alkalmazott
SET fizetes=1.1*fizetes
WHERE tkod IN (SELECT tkod FROM telephely
               WHERE varos<>'BUDAPEST');
```

Helyezzük át `KISS`-t az 98-as (nem létező) telephelyre:

```
UPDATE alkalmazott
SET tkod=98 WHERE anev like 'KISS%';
```

Hiba: nincs 98-as telephely a `TELEPHELY` táblában

Az alkalmazottak fizetését növeljük meg az azonos beosztásúak átlagfizetésének 10%-val:

```
UPDATE alkalmazott X
SET fizetes=fizetes+(SELECT 0.1*AVG(fizetes)
                     FROM alkalmazott
                     WHERE X.beosztas=beosztas);
```

III. Adatok törlése

DELETE [FROM] [séma.]{táblanév | nézet} [aliasnév]
[**WHERE feltétel**];

WHERE feltétel: a törlésben résztvevő sorokat válogatja ki

a) Egy sor törlése:

Töröljük a 70 kódú telephelyet:

```
DELETE FROM telephely WHERE tkod=70;
```

Hiba: TKOD: 60 külső kulcs

b) Tábla teljes tartalmának törlése:

```
DELETE FROM tabla;
```

c) Táblarészlet törlése:

```
DELETE FROM telephely
WHERE varos='EGER';
```

Hivatkozási megszorítási hiba: ha olyan sort törlünk egy táblában, amelynek elsődleges kulcsára hivatkozás van egy másik vagy ugyanazon táblában. Szülő táblában olyan sor nem törölhető, melyhez vannak sorok a gyerek táblában!