Nézettáblák

Gyorsításra, vagy egyszerűsítésre

Nézettáblának nevezzük az olyan táblákat, amelyek egy elkérdezés (SELECT) eredményeként jönnek létre. Tényleges tárolásra nem kerülnek, csak az őket előállító utasítást tárolja az adatbázisrendszer. Amikor szükség van rá, akkor a tárolt utasítást végrehajtva létrehozza az adatbázisrendszer a nézettáblát az adatbázis pillanatnyi tartalmának megfelelően.
A nézettáblák lekérdezésre ugyanúgy használhatók, mint a táblák.
Nézettáblára is létrehozható nézettábla: A nézettábla – néhány kivételtől eltekintve – teljesen ugyanúgy használható, mint a tárolt táblák, így például nézettáblák létrehozásában is szerepelhetnek.

Nézettábla definiálása:

CREATE VIEW Nézettáblanév (attributúmok neve) **AS SELECT** …;

Feladat:
A „telepules” táblából hozzunk létre egy „telepulesek1” nevű nézetet, amely tartalmazza a települések nevét, a megye, járás és lakosság adataikat.

A nézettáblában szereplő mezők elnevezése:

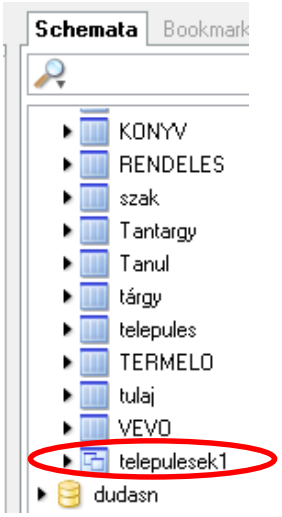
```
CREATE VIEW telepulesek1(Név, Megye, Járás, Lakosság)
  AS SELECT helysegnev, megye, jaras, lakossag FROM telepules;
```

vagy

```
CREATE VIEW telepulesek1
  AS SELECT helysegnev AS Név, megye AS Megye, jaras AS Járás, lakossag AS Lakosság
     FROM telepules;
```

vagy a nézettáblában szereplő mezők neve azonos lesz az eredeti táblában szereplő mezőnevekkel:

```
CREATE VIEW telepulesek1
  AS SELECT helysegnev, megye, jaras, lakossag FROM telepules;
```




```
SELECT * FROM telepulesek1;
```

Nézettábla törlése: **DROP VIEW** nézetnév;

Feladat1:

Korábban már találkoztunk az alábbi feladattal:

Állítsuk elő megyénként a legkisebb lakosságú települések listáját! (megye, településnév, lakosság) rendezés: lakosság, megye, településnév.

```
SELECT megye,helysegnev,lakossag FROM telepules
  WHERE (megye,lakossag) IN
   (SELECT megye,MIN(lakossag) FROM telepules GROUP BY megye)
  ORDER BY 3,1,2;
```

Tapasztalhatjuk, hogy a végrehajtás „sok" időt igényel.

Hozzuk létre a belső lekérdezés (SELECT) által adott eredménytáblát nézettáblaként:

```
CREATE VIEW megyeim AS
  SELECT megye,MIN(lakossag) FROM telepules GROUP BY megye;
```

Ezt a nézettáblát felhasználva alakítsuk át a „lassú" lekérdezést:

```
SELECT megye,helysegnev,lakossag FROM telepules
  WHERE (megye,lakossag) IN
    (SELECT * FROM megyeim)
  ORDER BY 3,1,2;
```

Az eredmény ugyanaz, de lényegesen gyorsabban!!!

Feladat2

Tegyük fel, hogy Ön a járási hivatal dolgozója, és gyakran van szüksége az egyes járások településeinek az adatára, amiket – de csak a saját járásához tartozókat – módosíthat is.

Ekkor célszerű lehet (minden járásra más-más) nézettáblát létrehozni, és a későbbiekben ezt használni.

A „telepules" táblában minden település adatai szerepelnek. Hozzon tehát létre egy „jarasom" nézettáblát, amelyben csak a saját járásához tartozó települések neve, típusa, és lakossága adatai szerepelnek.

A „jaras" mezőnév – ebben a táblában – állandó lévén ide nem kell (gondolhatnánk)?!

```
CREATE VIEW jarasom AS
  SELECT helysegnev,tipus,lakossag FROM telepules
  WHERE jaras='Adonyi';
```

De: a magyar ékezetes járásneveknél nem tudja létrehozni a nézetet az adatbázisrendszer!
Ezért konvertálni kell:

```
CREATE VIEW jarasom2 AS
  SELECT helysegnev,tipus,lakossag FROM telepules
    WHERE jaras=CAST('Tamási' AS CHAR CHARACTER SET utf8);
```

CAST(): Általános típuskonverziós függvény - **CAST** (kifejezés **AS** adattípus)

Nézzük meg a tartalmát:

```
SELECT * FROM jarasom;
```

Nézettáblán keresztüli módosítás

Módosítsunk a „jarasom”nézettáblában: *Tegyük fel, hogy Kulcson hármas ikrek születtek. (Kulcs lakossága: 2669 fő.)*

```
UPDATE jarasom SET lakossag=lakossag+3 WHERE helysegnev='Kulcs';
```

Nézzük meg a nézettáblában és az alaptáblában is:

```
SELECT * FROM jarasom;

SELECT * FROM telepules WHERE helysegnev='Kulcs';
```

Megjegyzés: *Nem minden nézettáblán keresztül lehet módosítani!*

Néhány szabály, ha nézettáblán keresztül szeretnénk módosítani:

- A nézettábla definíciójában a SELECT után nem szerepelhet DISTINCT.
- A FROM záradékban csak az alaptábla, és az is egyszer szerepelhet.
- A WHERE záradékban az alaptábla nem szerepelhet egy alkérdésben sem.

Szúrjunk be a nézettáblába:

Tegyük fel, hogy új település csatlakozott a járáshoz (Újfalu, mely falu típusú és 10 fő a lakossága):

A nézettáblára kiadjuk a következő utasítást:

```
INSERT INTO jarasom VALUES('Újfalu','Falu',10);
```

Nézzük meg:

```
SELECT * FROM jarasom;
```

A beszúrt sor a nézettáblában nem jelenik meg!

Az adatbázisrendszer felismeri, hogy a „jarasom" egy nézettábla. Előveszi a definiáló utasítását, és a fenti utasítást átfordítja az alaptáblára vonatkozó utasítássá. Majd ezt végrehajtja, azaz bekerül az új falu a települések közé.

Nézzük az alaptáblában:

```
SELECT * FROM telepules;
```

Megvan! De nincs járás adata, hisz ilyen a nézettáblában nincs is?!

Ha a módosítást az adatbázisrendszerre bízzuk, akkor a nézet létrehozásakor a SELECT záradéklistájában elegendő attribútumot kell megadnunk ahhoz, hogy az alaptáblában végrehajtható legyen.

Oldjuk meg a nézettáblán keresztül való beszúrási feladatot ismét!

Két tipikus megoldási módszer: a nézettábla bővítése a létrehozásában szerepet játszó attribútumokkal, (vagy alkalmas trigger elkészítése – később).

Töröljük tehát a „járásom" nézettáblát:

```
DROP VIEW jarasom;
```

Hozzuk létre úgy, hogy minden olyan attribútum szerepeljen benne, ami a létrehozásában szerepet játszik:

```
CREATE VIEW jarasom AS
  SELECT helysegnev,tipus,lakossag, jaras FROM telepules
  WHERE jaras='Adonyi';
```

Szúrjunk be új sort a nézettáblába:

```
INSERT INTO jarasom VALUES('Újfalu2','Falu',15, 'Adonyi');
```

S ez természetesen megjelenik a nézettáblában:

```
SELECT * FROM jarasom;
```

és persze az alaptáblában is

```
SELECT * FROM telepules;
```