Gyakorlati óravázlat 10 – Nézetek, beágyazott lekérdezések

1. Beágyazott lekérdezések
2. Nézetek
3. Módosítás nézeten keresztül

\_\_\_\_\_\_\_\_\_\_\_\_

**Nézetek**

A nézeteknek két nagy felhasználási köre van: egyrészt egyszerűsíthetik a lekérdezéseket: a beágyazott lekérdezést megcsináljuk külön és névvel látjuk el, majd felhasználjuk abból való lekérdezésre.

Másrészt jogosultságkezelésre is lehet alkalmas. A mai órán csak a lekérdezések írásának megkönnyítésére fogjuk használni.

A lekérdezés eredményeként létrejött relációt pontosan ugyanúgy használhatjuk lekérdezésekben, mint a táblákat.

Nézetekkel kapcsolatos utasítások:

**Létrehozás**

CREATE VIEW nézetnév [(oszlopnevek)] AS SELECT …;

Ha nem adunk meg oszlopneveket, akkor a select-ben szereplő oszlopnevekkel jön létre a nézet, de ebben az esetben kötelező a kifejezéseknek aliast adni a select listában.

Ha a nézetet definiáló select-et szeretnénk **módosítani**, akkor:

CREATE OR REPLACE VIEW nézetnév AS SELECT …

Általában minden nézetet ezzel az utasítással szoktunk létrehozni.

Van módosító utasítás is, de az a nézet jellemzőit, illetve a nézetre létrehozott kényszereket módosítja. Nem fogunk vele dolgozni, de a dokumentációban megnézhető: ALTERT VIEW

**Törlés**: DROP VIEW nézetnév.

Az jogosultsági kérdés, hogy hozhatunk-e létre nézetet más sémájában lévő táblák lekérdezésével. Jelenleg ez nem engedélyezett. Ezért ma létrehozunk táblákat a saját sémájában mindenkinek, hogy aztán lehessen a nézetesdit kipróbálni.

Tehát a 0. feladat, hogy dobjunk el mindent és másolással hozzuk magunknál létre a 3 kölcsönzős táblát (a nem tábla nem fontos). Mivel ma csak lekérdezünk, nincs jelentősége annak, hogy a kényszereket a másolás nem hozza át.

**Feladatok**

1. Készítsünk az mszemely-ből két nézetet, az egyik tartalmazza a férfiakat, a másik a nőket. Kérdezzük is le a nézeteink tartalmát!
2. Kérdezzük le az 50 évesnél idősebb nőket!

Ezt a lekérdezést simán meg tudnánk írni kapásból a szemely táblára is, de tételezzük fel, hogy mi olyan felhasználók vagyunk, akik a szemely tábla létezéséről nem is tudunk, számunkra csak a ferfiak és a nok táblák léteznek. A lekérdezéseket író felhasználók gyakran nem is tudják, hogy az általuk használt táblák valójában táblák-e vagy nézetek. Az ő szemszögükből nincs is ennek jelentősége.

1. Kérdezzük le az azonos évben született nőket és férfiakat!
2. Ismételjük meg a 2. feladatot nézet használata nélkül, beágyazott lekérdezéssel!

Annyi történik, hogy lemásoltuk a 2. feladat megoldását. Kikommenteztük belőle a nézet nevét és helyette zárójelben beírtuk azt a lekérdezést, amivel a nézet létrejött. A beágyazott lekérdezést mindig így kell elképzelni. Gyakran azért van szükség a beágyazott lekérdezésre, mert nincs is jogosultságunk a szükséges nézetet létrehozni.

**Beágyazott lekérdezések**

A lekérdezések eredménye mindig reláció. Ez az eredményreláció lehet több soros, több oszlopos, de nem ritkán egyetlen oszlop (gyakorlatilag egy lista), vagy akár egyetlen érték a lekérdezés eredménye. A lekérdezések eredményét fel lehet használni egy újabb lekérdezésen belül.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

1. a FROM után, akár JOIN részeként: több soros, több oszlopos
2. WHERE ill. HAVING feltételben: lista, vagy érték lekérdezés. Lista jellemzően az IN operátorral. Előfordulhat az is feltételben, hogy több értékből álló feltételünk van, akkor az IN után is többoszlopos SELECT szerepelhet
3. extrém esetben akár a select listában: értéklekérdezést, az új lekérdezés minden sorában ugyanaz az érték fog megjelenni

A beágyazott lekérdezések néha a táblák összefűzésének alternatívái.

**Feladatok**

1. Kérdezzük le, hogy az egyes személyek hányszor kölcsönöztek ki korhatár nélküli filmeket! Azokat is lássuk, akik egyszer sem!

A csoportfüggvényt a beágyazott lekérdezésben kell használni és azt külső joinnal a szemely táblához kapcsolni, különben a where feltétel elrontaná a külső joint. A beágyazott lekérdezés helyett használhatunk nézetet is.

1. Mi van ha más besorolásal kapcsolatban vagyunk kíváncsiak ugyanerre?

Ha beágyazott lekérdezésünk van, akkor egyszerűen át tudjuk írni a WHERE feltételt. Ha nézetet használtunk, akkor viszont a nézetet kell módosítanunk. Ezért jó pl. az OR REPLACE létrehozási mód, mert akkor csak a nézet definíciójában írjuk át a feltételt és újrafuttatjuk. Aztán futtathatjuk újra az ebből készült lekérdezést.

1. Listázzuk ki az átlagnál nagyobb országokat!

A megoldáshoz a múlt órán is használt ORSZ szinonímát használjuk. Ha feltételben van a beágyazás, ott nem is kényelmesebb helyette a nézet használata. Gyakoribb, hogy tényleg megírjuk a select-et.

1. Melyik kontinensen a legkisebb az országok átlag területe?

HAVING feltételben van a beágyazás.

1. Ismételjük meg a 7. feladatot, de írjuk ki az átlagérétéket is minden rekordban, illetve, hogy mekkora az eltérés.

Ennél a feladatnál a selectlistába kerül egy értéket eredményül adó allekérdezés. A kiszámolt értékkel további számításokat is végezhetünk, pl. különbség, vagy arány, kerekítés. Látszik pl. hogy Oroszország mérete majdnem 25-szöröse az átlagméretnek.

A beágyazott lekérdezések használhatók a JOIN helyett is. Ezeket a lekérdezéseket semi-join-nak is szokták nevezni.

1. Kérdezzük le azokat a személyeket, akik még soha nem kölcsönöztek! Ezúttal ne használjunk külső joint!

A megoldás előtt megnézzük a 8. heti, külső joinos megoldásunkat erre a feladatra. Ehelyett lekédezhetjük csak a SZEMELYEK táblát úgy, hogy a feltételben egy allekérdezést használunk a másik táblából. Az allekérdezés egy listát ad eredményül, amit a feltételben úgy használunk fel, hogy azokra van szükségünk, akik nem szerepelnek a listában, tehát az operátor: NOT IN.

1. Kérdezzük le azokat a filmeket minden adatukkal, amelyeket legalább 5-ször kikölcsönöztek már. Ezuttal ne használjunk joint!

**Korrelált subselect**

Előadáson a DML-nél volt róla szó, de sima select-ben és DML utasításban egyarán előfordulhat, hogy beágyazott lekérdezést használunk és néha szükséges, hogy a beágyazott lekérdezés a külső utasítás által érintett minden rekordra külön-külön lefusson. Azaz a beágyazott lekérdezésben olyan feltétel van, ami a külső utasítás adott rekordjának az egyik mezőjétől függ. Éppen ezért nézünk példát select-ben lévő correlált subselectre is, illetve DML utasításban levőre is. Korrelált subselect önmagában nem futtatható le, mert hivatkozik a külső utasítás táblájának egy mezőjére.

1. Kérdezzük le azokat a filmeket, amelyek imdb pontszáma magasabb, mint az adott rendező filmjeinek pontátlaga.
2. Legyenek VIP ügyfelek, akik legalább egyszer kikölcsönözték valamelyik Keresztapa filmet! – ez nem lesz korrelált!
3. Legyenek vip ügyfelek, akik a születési évüknél régebbi filmet is kölcsönöztek már ki! – ez már korrelált

**Nézet használata joinok helyett**

Az utolsó feladatban hozzunk létre 2 nézetet a 3 táblánkból, ami alapból megkönnyíti az összekapcsolásos lekérdezéseket. Az egyik nézetben csak a tényleges kölcsönzések minden adata szerepeljen. A másikban minden olyan személy és DVD is, amelyek nem szerepeltek egyetlen kölcsönzési eseménynél sem.

1. Hozzunk létre két nézetet a tábláinkból, az egyikben minden kapcsolódó adat szerepeljen, a másikban összesen minden adat szerepeljen a tábláinkból!

Hangsúlyozzuk a hallgatóknak, hogy a 2 héttel ezelőtti feladatok ezekből ugyan könnyebben megoldhatóak, de mindenképpen gyakorolják be úgy is, hogy ők megírják az összes joint. Azért nem univerzális megoldás mindig egy ilyen teljes nézet létrehozása, mert egy valódi adatbázisban nem 3-4 tábla szokott lenni, hanem nagyságrendekkel több (akár több 100). Természetesen a gyakran együtt szereplő táblákra létezni szoktak ezek a nézetek és a tipikus lekérdezéseket ezekből szoktuk létrehozni. De mindig lehetnek olyan extra feladatok, amelyekhez másik táblákat is össze kell kapcsolni.

Azt is el lehet mondani (bár a következő órán erre még biztosan visszatérünk), hogy abszolút életszerű az is, hogy egy feladat megoldását a szükséges táblák felderítésével kell kezdeni. Nem triviális az, hogy az embernek a fejében van a komplett adatszerkezet.

**Önálló feladatok:**

1. Kérdezzük le a VIP férfi ügyfeleinket. Használjuk az óra elején létrehozott nézetet!
2. Kérdezzük le a 40 évesnél fiatalabb férfiakat és a náluk fiatalabb nőket! Rendezzük a listát a férfiak kora szerint, azon belül a nők kora szerint! Használjuk az óra elején létrehozott nézeteket!
3. Kérdezzük le, hogy az egyes filmeket hány alkalommal kölcsönözték ki 3 napnál rövidebb időre! Csak azokat listázzuk ki, amelyeknél egyáltalán volt ilyen kölcsönzés!
4. Ismételjük meg az előző feladatot, kérdezzük le, hogy az egyes filmeket hány alkalommal kölcsönözték ki 3 napnál rövidebb időre! Azok is jelenjenek meg a lekérdezésben, amelyeknél ez a szám 0!
5. Listázzuk ki az átlaghossználh hosszabb filmeket!
6. Listázzuk ki azokt a filmeket, amelyek hosszabbak a leghosszabb korhatár nélküli filmnél!
7. Listázzuk ki azokat a filmeket amelyek hosszabak a saját műfajuk átlaghosszánál (csak a műfaj1 oszlopot vegyük figyelembe!)
8. (Nehéz) Listázzuk ki azokat a filmeket amelyek hosszabak a saját műfajuk átlaghosszánál! Vegyük azonos műfajúnak minden olyan filmet, amelynek a műfaj1 vagy műfaj 2 besorolása megegyezik az adott film műfaj1 vagy műfaj2 besorolásával!
9. Listázzuk ki, hogy melyik filmet hányszor kölcsönözték ki összesen, és hányszor kölcsönözték ki 4 napnál hosszabb időre! Ha egyszer sem, az is jelenjen meg az eredményben!
10. Listázzuk ki, hogy melyik ügyfelünk hány saját magánál öregebb filmet kölcsönzött már ki!