Gyakorlati óravázlat 14 – Jogosultságok, Tranzakciók

1. Paraméteres utasítás
2. Módosítás nézeten keresztül
3. Jogosultságok
4. Tranzakciók

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

**Koncepció**

Az óra elején visszatérünk a nézetekhez, és a korábban nem tárgyalt jogosultságkezelést megnézzük a segítségükkel. A folytatásban a hallgatók párban dolgozzanak, egymásnak adnak jogosultságokat illetve egymás tábláival fognak műveletek végezni, így áttérünk a tranzakciókezelésre is egy kicsit.

Kivételesen a közös anyag megbeszélése után rögtön megírjuk a sprintet (a videó végén megmutatom, hogyan kell átállítani a csoportra egyénileg vonatkozó határidőt. Az utolsó sprint megírása után mindenki véglegsítse a döntését a kihívással kapcsolatban (hétfőn törölni is fogom a „még nem döntöttem el” opciót és átnevezem előjelentkezésről jelentkezésre a választás eseményét. A jelentkezés végső határideje május 11. csütörtök 17:30. Aki eddig az időpontig nem jelzi, hogy írni szeretne, az nem jöhet el a pénteki zh-ra. Terembeosztást csütörtök este fogok tudni csinálni.

A kötelező részek után az óra további része konzultáció, amihez első sorban a közös feladatgyűjtemény feladatait ajánlom felhasználni. Kérlek bíztassátok a hallgatókat, hogy már a következő óráig töltsenek fel oda feladatokat, illetve az órán megoldott feladatok megoldását kommenteljék a feladat alá, vagy ha már van ott megoldás, akkor vessék azzal össze a saját megoldásukat. Még egyszer kérem azt is, hogy TI IS töltsetek fel néhány feladatot!

Kiindulásnak egy GYEREK nevű táblát hozunk létre, feltöltjük néhány adattal, majd létrehozunk hozzá néhány nézetet, amin keresztül DML utasításokat hajtunk létre az eredeti táblán.

**Kiegészítések**

Paraméteres utasításhoz a paraméter megadása kettőspont és a paraméter neve (spec karakter nélkül). Kliensprogram függő, hogy a paramétereket hogy kéri be (részletek a videón).

Amit még nem mutattunk meg korábban, hogy hogy lehet átállítani az aktuális session paramétereit. A dátumoknál szeretnénk pontos időpontot is kiírni.

**Feladatok**

1. Hozzuk létre a gyerek táblát az alábbi utasítással!
2. Paraméteres utasítással vigyünk be 2 fiút és 2 lányt a táblába, csak a nevet és a nemet adjuk meg, a többi alapértelmezett.

**Nézetek**

A nézeteken keresztül van lehetőség DML utasítások futtatására is. A művelet mindig a forrás táblára fog vonatkozni. Nyilvánvalóan attól függően, hogy a nézetet milyen definíció hozta létre, nem minden nézet alkalmas minden DML utasításra. Nem teljeskörűen de erre nézünk néhány példát!

A nézet végén a WITH kulcsszó segítségével direktben is meg lehet adni, hogy hogy viselkedjen. A WITH READ ONLY azt fogja jelenteni, hogy nem lehet rajta keresztül módosítani az eredeti tábla tartalmát. A WITH CHECK OPTION pedig az jelenti, hogy csak úgy lehet a nézeten keresztül módosítani, hogy a nézetet létrehozó lekérdezésben szereplő feltételeknek megfeleljenek a keletkező adatok (insert és update esetén is)

**Feladatok**

1. Hozzunk létre egy bevitel nevű nézetet, amelyen keresztül majd más is tud adatot bevinni a táblánkba. A felvitte és a felvitel adatok mindenképpen csak alapértelmezettek lehessenek!
2. Vigyünk be további 2 rekordot a nézeten keresztül!
3. Készítsünk bevitel2 néven különböző nézeteket és próbáljuk ki rajtuk keresztül az adatbevitelt illetve más DML utasításokat!
4. Készítsünk olyan nézetet, amelybe biztosan nem lehet adatot bevinni!
5. Készítsünk olyan nézetet, amelyben csak a lányok szerepelnek! Ezután próbáljunk fiút bevinni a nézeten keresztül!

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: bizonyos felhasználói köröknek nem adunk jogosultságot az eredeti táblák olvasására/módosítására, csak a táblákból készített nézetekre. A felhasználó az objektum nevéből, viselkedéséből nem fogja tudni, hogy amit használ az éppen tábla vagy nézet. Fontos végiggondolni, hogy a használt nézetek alkalmasak-e DML utasítások szereplőjének. Ez megtiltható explicit is, de vannak nézetek, amelyeken keresztül biztosan nem tudunk DML utasításokat futtatni (pl. aggregát adatokat tartalmaznak).

Ha a nézet megadása végén a WITH READ ONLY kódot adjuk meg, akkor csak olvasható nézetet hozunk létre, ha WITH CHECK OPTION-t, akkor a nézeten keresztül csak úgy módosíthatunk, hogy a nézetet létrehozó feltétel igaz maradjon, ha egyiket sem adjuk meg, akkor bevihetünk/módosíthatunk úgy is adatot, hogy az a nézetet létrehozó select feltételének nem felel meg, csak az eredeti tábla kényszereinek.

**Egymás tábláinak használata**

Jelenlegi jogosultsági beállítások szerint minden hallgatónak hallgato szerepe (role) van, a hallgato szerepnek pedig van select any table rendszerjogosultsága. Ezért le tudjuk kérdezni bárki tábláit, nézetetit. Az adatok írásához, módosításához viszont nincs jogunk. Jogok adása a GRANT utasítással történik, visszavonása pedig a REVOKE-kal. A GRANT szintaktikája: GRANT mit (vesszővel elválasztva több jogot is adhatunk egyszerre) ON objektum TO felhasználó vagy szerep.

REVOKE mit ON objektum FROM felhasználó vagy szerep.

A feladatok végrehajtásához osszuk párokra, szükség esetén 3-as csoportba a hallgatókat. A feladatokban kihagyott részekre (\_\_\_) mindig a társ sémanevét kell majd beírni (érdemes kitenni vágólapra és végig beilleszteni). Akik 3-an vannak, azoknak mindent duplán kell csinálniuk, hogy minden irányban működjön a dolog. Ezt elkerülendő, párataln számú hallgató esetén egy hallgató lehet a mi párunk. Kérjük meg a hallgatókat, hogy ne szúrjanak ki egymással, mindenkinek, aki dolgozni akar, olyan párja legyen, aki szintén halad velünk.

Ezeket a feladatokat tanári vezénylet mellett azonos tempóban kell végrehajtani, hogy kijöjjenek azok a dolgok, amiket meg szeretnénk mutatni.

**Feladatok**

1. Kérdezzük le a szomszédunk gyerek táblájának és bevitel nézetének tartalmát! Ehhez szükségünk lesz az ő sémanevére! Az alulvonások helyére mindig a társunk sémanevét írjuk be! Miért tudjuk lekérdezni?
2. Vigyünk be a szomszédunk gyerek táblájába a bevitel nézeten keresztül egy gyereket!

Hibaüzenetet kapunk jogusultsághiány miatt.

1. Adjunk jogosultságot a szomszédunknak a saját táblánkra adatbevitelre és módosításra!
2. Szúrjunk be egy új rekordot a szomszéd táblájába! (Ismételjük meg a 9. feladatot!)
3. Szúrjuk be a saját bevitel nézetünk adatait a szomszéd bevitel nézetébe! Ne véglegesítsük! Kérdezzük le a szomszéd gyerek tábláját és a sajátunkat is! Ismételjük meg a két lekérdezést azután is, hogy mindketten kiadtuk a commit parancsot!

A véglegesítés előtti lekérdezésnél úgy tűnik, mintha a társunk lusta lenne. Mikor az ő tábláját kérdezzük le, akkor látjuk az általunk odaírtakat is, mikor a sajátunkat, akkor a társunk adatai hiányoznak. Miután mindketten comittoltunk, akkor látszik az összes adat itt is ott is. Eltérés nyilván lesz: az azonosító sorszámban illetve a felvitel dátumában. Itt érdemes erről eldiskurálni egy kicsit, hogy mi lett volna ha a sorszámokat is át akarjuk vinni (nyilván akkor kulcsegyezés). Hogyan lehetett volna ezt megelőzni: ha mondjuk a kulcsérték mindenkinek a saját sémájában nem egyszerűen a számláló következő értéke, hanem a felvivő felhasználónév és a sorszám együtt.

1. Módosítsunk a saját táblánkban minden fiút, hogy ne legyen iskolás! NE COMMITTÁLJUNK! Módosítsuk a szomszédunk nézetében a Pisti nevű fiúkat úgy, hogy iskolások legyenek!

Forgatókönyv:

1. Mindketten módosítják a fiúkat commit nélkül!
2. Mindketten kiadják a Pistit módosító utasítást és mindketten várnak (mert az előző módosítás miatt minden fiú rekordon zár van.
3. egy rövid idő után a pár egyik tagjának hibaüzenettel leáll az utasítás: amelyiknél előbb észreveszi a rendszer, hogy holtpont (deadlock) van, nevezzük őt A felhasználónak, a még mindig várakozót pedig B-nek.
4. Beszéljük meg, hogy mi történt és hogy mi lehet a megoldás. A megoldás az, ha A megszűnteti a zárat. Ezt kétféleképpen teheti meg: vagy commit-tal vagy rollback-kel befejezi a tranzakciót. Mivel az eredeti szándék szerint a tranzakciója két lépésből állt volna, azért nem lenne szerencsés fél úton commit-tal fejezni be. Az A felhasználó adja ki a Rollback utasítást, a B felhasználónak abban a pillanatban le fog futni az utasítása.
5. B kész a teljes tranzakcióval, most kiadhatja a COMMIT-ot.
6. Ezután A lefuttathatja mindkét utasítását egymás után és ő is kiadhatja a COMMIT-ot.

Nem volt sorbarendezhető a két tranzakció, mert A szándéka az volt, hogy a saját táblájában senki ne legyen iskolás és a B táblájában Pistik iskolások legyenek. Miután mindkét tranzakció lefutott a következő az eredmény:

|  |  |  |
| --- | --- | --- |
| Idő | A | B |
| T1 | A fiúi nem iskolásak | B fiúi nem iskolásak |
| T2 | B Pistik iskolásak - várakozás | A Pistik iskolásak - várakozás |
| T3 | Deadlock hibaüzenet, utasítás szintű rollback | várakozás |
| T4 | Rollback | A Pistik iskolásak |
| T5 | A fiúi nem iskolásak (Pistik sem!) |  |
| T6 | B Pistik iskolásak |  |
| Végeredmény | A fiúi egyáltalán nem iskolásak | B fiúi a Pistik kivételével nem iskolásak. |

Így A eredeti szándéka teljesült, B eredeti szándéka viszont nem. Ha A T4-kor commit-ált volna és csak a második utasítását futtaja újra, akkor B eredeti szándéka teljesült volna és A-é nem.

**Módosítások a csoport sémájában**

Minden csoportnak a saját sémájában (G01\_SZALLASHELY, stb.) is létrehoztam a GYEREK táblát és a BEVITEL nézetet, valamint az utóbbira INSERT és UPDATE jogot adtam a HALLGATO szerepnek. A kihagyott részek (\_\_\_) helyére most mindenkinek majd a csoport sémanevét kell bemásolgatnia (érdemes vágólapról)

1. Vigyük be a saját bevitel nézetünk tartalmát a csoport sémánk bevitel nézetébe és commitáljunk!
2. Módosítsa mindenki a csoport sémánk bevitel nézetén keresztül a Pisti nevű gyerekek adatait, hogy biztosan iskolásak legyenek!

Ha van kedvetek egy kis tornaórához (én az enyémekkel biztosan meg fogom csinálni): én adom ki az utasítát először, nekem lefut és nem commitálok. A többiek is kiadják az utasítást és mindenki vár. Az egész csoport beállt egy várakozási sorba. Amikor én commitálok, akkor egy valakinek lefut, ő álljon fel, majd állva adja ki a commit-ot. Akkor megint egy embernek lefut. Feláll, commit, és így tovább. A játék végére mindenki állni fog. Ez volt az utolsó közös feladat.

Kivételesen a sprint a közös feladatok előtt legyen. A videón ott van, hogy hogy tudjátok az időt lezárni csak a saját csoportotok számára. A sprint után mindenki véglegesítse a kihívás jelentkezését. Ennek a legvégső határideje csütörtök 17:30. Aki addig nem jelöli, hogy megírja a kihívást, az nem írhatja meg. Mindezek után, aki akar távozhat, a többieknek konzultáció.

További vizsgainfók a videó végén.

**Önálló feladatok – a feltöltött feladatgyűjtemény feladatai**