Gyakorlati óravázlat 08 – Táblák összekapcsolása lekérdezésben

1. Oracle join szintaxis
2. Különböző join szintaxisok
3. Non equi-join
4. Tábla csatolása önmagával
5. Külső illesztések

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

**Szinonímák lekérdezése**

A Minta sémában található kolcsonzo,nem, dvd és kolcsonzes táblákara a DBA létrehozott ún. publikus szinonímákat mszemely,mnem, mdvd és mkolcsonzes néven. Ha ezeket a neveket használjuk a lekérdezéseinkben, akkor nem szükséges a sémanevet eléírnunk. (Ha azonos néven van saját objektumunk is, akkor a lekérdezés a sajátunkra fog vonatkozni. Ezért igyekeztem olyan neveket adni, amelyek nem voltak használatban.)

**JOIN – különböző szintaxisok**

Volt előadáson, de az már régen volt. Igyekeztem didaktikusan felépíteni a példákat és lehetőleg minden variációt bemutatni. Ezt nagyon kell nekik tudni!

**Feladatok:**

1. Kérdezzük le az mszemely és mnem táblákból a személyek adatait!

Kezdjük az Oracle szintaxissal, mert azzal jól meg tudjuk mutatni, hogy a join valójában mit is jelent: a keresztszorzat szűrése az összetartozó értékekre.

Az első példafeladaton azt is meg tudjuk nézni, hogy ha az összekapcsolt tábláknak vannak azonos nevű oszlopaik, akkor azokat minősítenünk kell, azaz elé kell írnunk, hogy melyik tábla ilyen nevű oszlopára gondolunk. (Ráadásul ebben a példában az azonos név nem is azonos tartalmú oszlop, de ha azonos tartalmú lenne, a minősítés akkor is kéne.) A minősítésre nem csak a táblák teljes nevét lehet felhasználni, hanem a lekérdezésben megadott rövidített nevet is.

Mivel a kapcsoló oszlop neve nem egyezik a két táblában, a join szintaxisok közül csak az ON-osat használhatjuk ebben a feladatban. Az inner szó elhagyható, azért írtam megjegyzésbe. ON után nem kötelező a zárójel és akár összetett feltétel is jöhet utána, de azért átláthatóbb, ha csak a join feltételt írjuk és külön WHERE után írjuk az egyéb megszorítást (lásd a következő feladatban).

SELECT \* FROM

mszemely,mnem; -- Vigyázat, ez így keresztszorzat! Mindenki egyszerre nő és férfi.

--Nekünk csak azok kellenek, ahol a nem stimmel!

DESC mszemely

DESC mnem

SELECT \* FROM

mszemely, mnem

WHERE mszemely.nem=mnem.kod;

--Igazából a nemek kódját felesleges kiírni, pláne kétszer.

SELECT kid, nev, nem

FROM mszemely, mnem

WHERE mszemely.nem=mnem.kod; -- miért nem fut le?

SELECT kid, nev, mnem.nem

FROM mszemely, mnem

WHERE mszemely.nem=mnem.kod;

--vagy

SELECT s.kid, s.nev, n.nem

FROM mszemely s, mnem n

WHERE s.nem=n.kod;

-- milyen más módon tudjuk még lekérdezni ugyanezt?

SELECT s.kid, s.nev, n.nem

FROM mszemely s /\*inner\*/ JOIN mnem n

ON (s.nem=n.kod);

-- mivel nem egyezik az oszlopnév, más mód nincs.

1. Kérdezzük le, hogy melyik dvd-t kik kölcsönözték ki (most elég a kölcsönző kódja)! Csak az 1970-ben, vagy azután készült filmekre vagyunk kiváncsiak. Jelenítsük meg az éveket is. (Próbáljuk meg a lekérdezést különböző szintaktikákkal!)

Azt lehet tudni a két tábláról (most higyjék el nektek, ne kérdezgessük le katalógustáblákból), hogy mindkét táblának a sorszam az elsődleges kulcsa, de a dvd táblában a film is egyedi (be is van rá állítva a unique kényszer) és a kolcsonzes táblában a film oszlop a dvd tábla film oszlopára hivatkozik idegen kulcsként. A két tábla sorszam oszlopainak semmi köze nincs egymáshoz!

Kezdjük megint az Oracle szintaxissal. Látjuk azt is, hogy ha feltétel is van, akkor a join feltételhez azt AND-del tudjuk hozzáfűzni. Az ON-os megoldásnál nem and-del írtam (úgy is működött volna, de nem logikus), hanem külön where-be. Mivel azonos az oszlopnév, USING is használható. A USING (és a NATURAL JOIN) szintaxis használatánál arra kell vigyázni, hogy ezekben az esetekben a kapcsoló oszlop(ok) az eredményrelációban csak egyszer szerepelnek (akkor is, ha \* a select lista). Így viszont nem kell, sőt nem is szabad minősíteni ezeket az oszlopokat.

A NATURAL JOIN viszont nem jó, mert az az összes azonos nevű oszlop alapján illeszt, azaz annak felene meg, hogy WHERE d.film=k.film AND d.sorszam=k.sorszam. Ennek jelen esetben egyetlen rekord sem felel meg, de ha véletlenül lenne is ilyen sor, annak sem lenne semmi értelme (a dvd sorszáma megegyezik a kölcsönzés sorszámával).

DESC mdvd

DESC mkolcsonzes

SELECT \* FROM

mdvd d,mkolcsonzes k

WHERE d.film=k.film

AND ev >= 1970;

SELECT d.film,d.ev,k.sorszam,k.kolcsonzo FROM

mdvd d JOIN mkolcsonzes k

ON d.film=k.film

WHERE ev >= 1970;

--Mivel azonos az oszlopnév, lehet using is.

SELECT d.film,d.ev,k.sorszam,k.kolcsonzo FROM

mdvd d JOIN mkolcsonzes k

USING (film)

WHERE ev >= 1970;

--miért nem jó?

SELECT film,d.ev,k.sorszam,k.kolcsonzo FROM

mdvd d JOIN mkolcsonzes k

USING (film)

WHERE ev >= 1970;

--Mi történik, ha a using-ot is elhagyjuk és natural join-t használunk?

SELECT film,ev,sorszam,kolcsonzo FROM

mdvd NATURAL JOIN mkolcsonzes

WHERE ev >= 1970

;

--Miért nincs eredménye? Mi volt most az összekapcsoló feltétel?

1. Kérdezzük le mégegyszer, hogy kik kölcsönözték ki az 1970-ben, vagy azután készült filmeket, de most neveket is szeretnénk látni.

Nem csak kettő táblát lehet összekapcsolni. Ilyenkor úgy kell elképzelni, hogy két tábla összekapcsolásával keletkezik egy reláció, amihez aztán hozzákapcsolhatunk egy újabb relációt és így tovább. A JOIN műveleteket is lehet zárójelezni. Inner join-nál nem feltétlenül van jelentősége, de outer joinnál mindenképpen. Több tábla összekapcsolásánál kombinálhatjuk is a különböző szintaxisokat.

SELECT d.film,d.ev, k.sorszam, s.nev

FROM mdvd d, mkolcsonzes k, mszemely s

WHERE d.film=k.film AND k.kolcsonzo=s.kid

AND ev >= 1970

ORDER BY film,nev;

SELECT film,d.ev, k.sorszam, s.nev

FROM

(mdvd d JOIN mkolcsonzes k USING(film))

JOIN mszemely s ON(k.kolcsonzo=s.kid)

AND ev >= 1970

ORDER BY film,nev;

1. Összekapcsolás nem csak egyenlőség mentén lehet. Tételezzük fel (csak a példa kedvéért), hogy úgy szeretnénk filmeket ajánlani az ügyfeleinknek, hogy a nevük kezdőbtűjével megegyező kezdőbetűjű filmeket ajánljuk nekik. Kinek mit ajánlunk?

Ez a példa úgynevezett non equijoint tartalmaz. Az előadáson az hangzott el, hogy ez igen ritka, és egy fizetési kategóriás betweenes példa szerepelt. Ebből a hallgatók jó része azt szűrte le, hogy ha nem egyenlőség, akkor between. Az igazság az, hogy itt bármi lehet. Kisebb, nagyobb, bármiféle függvények, transzformációk a két oldalon. Az lehet még példa, ha tervezési hiba miatt nem azonos az adattípus a két táblában (az idegen kulcs nem is építhető fel), pl. az egyik táblában fix hosszúságú, míg a másikban változó hosszúságú szöveg van. Ebben az esetben a fix hosszúságút TRIM függvénnyel kell „kezelni”.

SELECT s.nev,d.film

FROM mszemely s, mdvd d

WHERE substr(s.nev,1,1)=substr(d.film,1,1)

ORDER BY 1,2;

--Menne join-nal is?

SELECT s.nev,d.film

FROM mszemely s JOIN mdvd d

ON (substr(s.nev,1,1)=substr(d.film,1,1))

ORDER BY 1,2;

1. Táblákat önmagukkal is összekapcsolhatunk. Kérdezzük le az azonos nevű ügyfeleinket! Minden párost csak egyszer írjunk ki!

Ha a második feltételt nem írjuk hozzá, akkor mindenkit összepárosít saját magával is, illetve oda vissza is kiírja a párosokat. (Kikommentezéssel ki lehet próbálni. Megkapjuk mind a 62 személyt önmagával párosítva, meg még azt a 2-t, ahol tényleg névegyezés van, de őket meg oda vissza is, így 66 eredménysorunk lesz.)

SELECT m1.nev, m1.kid, m2.kid FROM

mszemely m1, mszemely m2

WHERE m1.nev=m2.nev

AND m1.kid<m2.kid

ORDER BY m1.nev;

-- ugyanez joinnal

SELECT m1.nev, m1.kid, m2.kid FROM

mszemely m1 JOIN mszemely m2

ON (m1.nev=m2.nev)

WHERE m1.kid<m2.kid

ORDER BY m1.nev;

1. Listázzuk ki a 70-ben, vagy azután készült filmeket és azt, hogy hányadik kölcsönzés alkalmával kölcsönözték ki őket! Ezúttal szerepeljenek a listában azok a filmek is, amelyeket még nem kölcsönöztek ki!

Itt jutunk el az OUTER joinhoz. Az OUTER szót szintén nem kötelező kiírni. Azzal, hogy a left, right vagy full szavak valamelyike szerepel, már egyértelmű, hogy outerről van szó. Ahol nincs érték a kapcsolódó táblában, ott null értéket kapunk. Idegen kulccsal összekapcsolt tábláknál nyilván csak az egyik oldali külső joinnak van értelme, a másik oldalról a kényszer miatt egyértelmű, hogy a kapcsolódó táblában is mindig lesz adat. Az ilyen irányban írt külső join logikai hibának számít, még akkor is, ha az eredményben konrétan nem jelentkezik hiba. Hívjátok fel rá a hallgatók figyelmét, hogy csak akkor használjanak külső joint, ha azt a feladat valóban megkívánja.

select d.film, d.ev,k.sorszam kölcsönzés\_sorszám

from mdvd d left /\*outer\*/ join mkolcsonzes k

on (d.film=k.film)

where d.ev >= 1970

order by d.film;

-- mi szerepel a kölcsönzés\_sorszám helyén a még ki nem vett filmeknél?

--ugyanez fordítva:

select d.film, d.ev,k.sorszam kölcsönzés\_sorszám

from mkolcsonzes k right join mdvd d

on (d.film=k.film)

where d.ev >= 1970

order by d.film;

1. Listázzuk ki csak azokat a filmeket, amelyeket még senki nem kölcsönzött ki!

Itt megint külső join kell, a feltételben pedig kihasználjuk, hogy a kapcsolódó táblában null érték szerepel. Fontos, hogy a kulcs oszlopban (de mindenképpen not null oszlopban) vizsgáljuk a null értéket, különben olyan is előfordulhat, hogy van kapcsolódó rekord, csak éppen a vizsgált mező értéke nem volt megadva.

select d.sorszam,film,rendezo,ev

from mdvd d left join mkolcsonzes k

using(film)

where k.sorszam is null

order by ev;

1. Van-e olyan ügyfelünk, aki soha nem kölcsönzött még? Kik azok?

Ugyanaz pepitában. Annyit mutat be pluszban a feladat, hogy a \* elé is lehet minősítést írni (kivéve USING és NATURAL esetében).

select s.\*

from mszemely s left join mkolcsonzes k

on (s.kid=k.kolcsonzo)

where k.sorszam is null;

Az önálló feladatok a csoportsémákra vonatkoznak. Felírom a hallgatóknak, hogy melyik típusú séma melyik konkrét sémákban szerepel. Így gyakorlásképpen a többiben is megoldhatják a feladatokat. Fontos, hogy nem mindegyik csoport egyforma logikával oldotta meg a tervezést, így ugyanannak a feladatnak is eltérő megoldásai lehetnek a különböző sémákban. Érdemes mindegyik megoldás előtt felderíteni az adott séma szerkezetét. Lehet azt csinálni, hogy Ti döntsétek el, hogy melyik szállás, melyik egyetem és melyik webshop sémát javasoljátok a hallgatóknak. De lehet azt is mondani nekik, hogy az a legjobb gyakorlás, ha otthon megcsinálják mindegyikre. A szálláshelyes feladatok az 1.-től kezdődnek, az egyetemesek a 7.-től, a webhelyesek a 13.-tól.

Mindegyik csoportból az első 4 feladat alap, csuklóból meg kell tudniuk majd oldani a vizsgán. Az utolsó feladatok trükkösebbek és nem is egyformák, úgyhogy azokat mindenképpen javasoljátok otthon megoldásra az eltérő csoportokban is azok számára, akik 4-es 5-ösre pályáznak. Illetve ott a nehézségek sem sikerültek egyformára. Előfordulhatnak olyan feladatok is, amelyekhez hiányzik adat egy-egy sémából, akkor másikban próbáljuk meg megoldani.

Én az első feladatokat megoldottam két-két sémában (bocs azoktól, akiké kimaradt). A többit már csak egyben-egyben.

**Önálló feladatok:**

1. Listázzuk ki a szálláshelyfoglaló oldalunk ügyfeleit és azt, hogy mikori érkezéssel foglaltak szállást! (Csak azok az ügyfelek kellenek, akik legalább egyszer foglaltak.)\*/
2. Listázzuk ki a szálláshelyfoglaló oldalunkon szereplő szálláshelyeket és azok szobáit. Csak a 3 csillagos szálláshelyek klímás szobáit szeretnénk látni.
3. Listázzuk ki azokat a szálláshelyeket, amelyekhez még nincsenek felvíve a szobák az adatbázisunkban! Van ilyen?
4. Listázzuk ki a szálláshelyfoglaló oldalunk azon ügyfeleit, akik még nem foglaltak!
5. Ajánljunk az ügyfeleinknek olyan szállodát, ami a saját városukban van (az irányítószám a címek első 4 karaktere)
6. Listázzuk ki azokat az ügyfeleket, akik a születésnapjukat a szállodánkban töltik!
7. Listázzuk ki az egyetemi adatbázisunkból a hallgatókat és a tárgyak kódját, amiket felvettek. Azt is, hogy az adott tárgyat melyik félévben vették fel. (Csak azok a hallgatók kellenek, akik vettek már fel tárgyat.)
8. Listázzuk ki az egyetemi adatbázisunkból a tantárgyfelvételeket és az azokhoz tartozó erdményeket. Csak a 2015-16-os első félév 5-ös osztályzataira vagyunk kiváncsiak.
9. Listázzuk ki egyetemünk azon hallgatóit, akik még nem vettek fel tárgyat! Van ilyen?
10. Listázzuk ki azokat a tárgyakat, amelyeket még senki nem vett fel!
11. Ajánljunk hallgatóinknak olyan tárgyat, ahol a tantárgyfelelős vezetékneve megegyezik az övékkel!
12. Listázzuk ki azokat a hallgatókat, akik a 2015-16-os tanév második félévben a 2. vizsgájukon legalább 3 jeggyel jobbat értek el, mint az elsőn!
13. Listázzuk ki a webshopunk adatbázisából az ügyfeleket és azt, hogy melyik napokon rendeltek. (Csak azok az ügyfelek kellenek, akiknek volt már rendelésük.)
14. Listázzuk ki a webshopunk adatbázisából az iskolaszer kategóriájú termékeket és a hozzájuk tartozó rendelések azonosítóját és a darabszámokat! Csak azok a rendelések kellenek, ahol legalább 50db-ot rendeltek.
15. Listázzuk a webáruházunkból azokat a vevőket, akik még soha nem rendeltek. Van ilyen?
16. Listázzuk ki a webáruházunkból azokat a termékeket, amelyekből még soha nem rendeltek!
17. Csak a gyakorlás kedvéért ajánljunk az ügyfeleinknek olyan termékeket, amelyek listaára alacsonyabb, mint az ügyfelek életkora.
18. Azt tervezzük, hogy a születésnapos ügyfeleinknek 10% kedvezményt adunk. Hogy lássuk, ez az eddigi rendeléseinket hogyan érintette volna, ha már korábban bevezettük volna, listázzuk ki azokat a rendeléseket, amelyeket az ügyfél a születésnapja előtti egy hétben adott fel, és a tételek ára mellett irassuk ki a feltételezett kedvezmény összegét is.