Gyakorlati óravázlat 03 – Egyszerű lekérdezések, dual tábla, szöveg-szám és dátum az sql kifejezésekben

1. A SELECT utasítás alap szintaxisa, vetítés, szűrés és rendezés
2. Tábla „másolása” insert utasítás subselect-tel
3. DUAL tábla, hogy néz ki, mire való
4. Describe utasítás
5. Lekérdezésekben kifejezések, összetett feltételek, dátumkezelés
6. NULL még nem ma lesz, csak a jövő héten, mert szerintem sok lenne

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

**Kiegészítő információk gyakvezéreknek**

SELECT a megjelenítendő oszlopokat(!) határozza meg, az eredmény értékét nem befolyásolják. Természetesen, a függvények módosíthatják a visszaadott oszlopban található értéket, de a teljes eredményhalmazra, annak minden adott oszlopbeli értékére lefut. Ha nem minden sort(!) akarunk megjeleníteni, akkor ennek helye a WHERE feltételekben van. Ha valamelyik sor **igazzá teszi**(!) a WHERE feltételben megfogalmazott kifejezést, akkor és csak akkor kerülhet az eredményhalmazba. Az üres feltételre minden illeszkedik, azaz minden igazzá teszi.

SQL és a relációs elméletről: a relációs elmélet nem SQL és az SQL nem a relációs adatbázisok matematikáját követi. Kicsi az eltérés, de óriási jelentőségű: a reláció halmaz, az SQL-ben az adatok zsákot (bag) alkotnak. Előbbi definíció szerint mentes az ismétlődésektől, az utóbbiban lehetnek ismétlődő elemek. Ebből következően az SQL matematikája nem mindig zárt, illetve egzakt. Igaz, nem fogunk ebbe a problémába ütközni, ez a kényszerek feladásában – pontosabban a funkcionális függőségek (nem) kezelésében – érhető majd tetten, illetve ezért van UNIQUE, de nincs KEY tulajdonságunk. A kommunikálandó szöveg általában ilyenkor az, hogy az SQL azért mégis relációs, mert létezik az a bizonyos ROWID, amitől persze minden sor egyedi lesz. A valóságban a praktikusság dominál: néha egyszerűbb nem számokat nyilvántartani az adatbázisban, hanem a gyakoriság mutatja pl. a készletmennyiséget.

TO\_CHAR / TO\_DATE: egyetlen dátumfüggvény sem szabványos, mindenki máshogy használja, sajnos, például az MS-SQL-ben CONVERT van helyette. Ezt leszámítva nagyjából azonosak. A dátum elemeit jelző betűkódok teljesen különbözőek lehetnek.

Dátumkezelő függvények tudása hasznos, mutassuk meg, de az aritmetika és a konvertálás a két kiemelt jelentőségű elem.

A dátumokról röviden: a dátum tárolása egy sajátos adatszerkezetben valósul meg, egy programozásból talán ismerős struktúra, a rekord (struct) formájában. Fontos, hogy hiába adunk meg csak bizonyos paramétereket – pl. percet, órát – attól még a DATE és TIMESTAMP a többi elemet kitölti, és fizikailag is tárolódik a háttértáron. Érdemes megnézni, hogy melyek az alapértelmezett kötődések. A legkisebb egységtől felfelé haladva nap szintig általában a legkisebb érték az alapértelmezett, felette – így a hónap, év, század – a mai naphoz igazodik.

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

A SELECT utasítás alap felépítése:

SELECT szelktlista FROM tábla WHERE feltétel ORDER BY oszlop

a szelektlista tartalmazhat oszlopokat, az oszlopokkal végzett műveleteket vagy konstansokat vesszővel elválasztva (vetítés relációs művelet megvalósítása, alapértelmezetten zsák szemantikával, ha a SELECT után beírjuk, hogy DISTINCT, akkor lesz belőle halmaz szemantika)

az oszlopoknak lehet címkét adni: szóközzel, vagy az AS kulcsszóval. A címke egyrészt az eredménytáblázat fejlécében jelenik meg, másrészt a sorba rendezésnél lehet használni (máshol nem!)

a táblánál meg lehet adni a séma nevét, amiben van, de ha a saját sémánkban van, akkor nem kötelező. a táblának is lehet aliast adni, szóközzel elválasztva. Ennek akkor lesz jelentősége, ha már több táblából kérdezünk le (néhány alkalom múlva)

a WHERE után egy logikai kifejezésnek kell állnia, ami lehet összetett is. Azokat a sorokat fogja kilistázni az Oracle, amelyikek igazzá teszik a feltételt. (kiválasztás relációs művelet megvalósítása)

az order by után megadhatjuk az oszlop nevét, címkéjét vagy sorszámát. Vesszővel elválasztva többszintű rendezést is megadhatunk. A DESC kulcsszóval tudunk csökkenő rendezést produkálni.

**Feladatok**

1. Kérdezzük le a MINTA sémában található alkalmazott tábla teljes tartalmát!
2. Kérdezzük le csak a neveket (vetület)!
3. Kérdezzük le csak a születési éveket zsák és halmaz szemantikával is!
4. Kérdezzük le a születési évet és a nevet, a fejléc SZÜLETÉSI ÉV és NÉV legyen! Csak azok szerepeljenek a listában, akik 1980 után születtek! (a születési évben lévő szóköz miatt dupla idézőjelbe kell tennünk a címkét, így viszont már számít a kis és nagybetű – szokjuk meg, hogy mindig legyen csupa nagy)
5. Kérdezzük le ugyanazt, mint az előbb, az eredményt rendezzük év, azon belül név szerinti sorrendbe. Próbáljunk ki több variációt, illetve növekvő és csökkenő rendezést is (a nyelvi beállítások miatt tudja a magyar ábécét a rendezésnél, nem úgy mint a Moodle ☺)
6. Kérdezzük le az alkalmazottak adatait és azt, hogy kb. hány évesek (csak a születési évet vonjuk ki 2017-ből)! Itt mindenképpen érdemes címkét adni az oszlopnak, különben a képlet lesz a fejléc.
7. Kérdezzük le az alkalmazottak nevét és mindegyik mellé írjuk ki, hogy Alkalmazott! A plusz oszlop címkéje legyen Beosztás! (Szöveg konstanst egyszeres aposztrófok közé kell tenni!)

A DUAL tábla a SYS sémában lévő rendszertábla, ami létrejött az adatbázis létrehozásakor. Mivel automatikusan létrehozták az azonos nevű publikus szinonimát, ezért nem kell megadni a sémanevet a lekérdezéskor (egyébként a katalógus táblákat is ezért lehet sémamegjelölés nélkül lekérdezni).

Az SQLDeveloperben sql utasítások mellett SQL\*Plus utasításokat is tudunk futtatni. Ezeknek a végére nem kell pontosvessző, és mindig egy sorosak (ezért nem kell a pontosvessző). A DESCRIBE utasítás megjeleníti a scrip outputon a kért tábla szerkezetét. Rövidítve is használható: DESC táblanév.

**Feladatok**

1. Nézzük meg a DUAL tábla szerkezetét!
2. Kérdezzük le a DUAL tábla tartalmát! Mire jó ez? (konstansok lekérdezésére)
3. Kérdezzünk le szám és szöveg konstansokat!

Számok: tizedes pontot kell használni az sql kódban! Hogy miképp jelenik meg a lekérdezés eredménye, az formátum beállítás kérdése.

Szövegek: egyszeres idézőjelek között! Számít a kis és nagybetű!

1. Használjuk az Oracle-t számológépnek, kérdezzük le különböző számítások eredményét! (A példa lekérdezést írjuk át!)

Nézzük a dátumokat! Nehezebb téma: nem hagyhatjuk figyelmen kívül a formátumot. Beírni nem tudunk dátum értéket, csak szöveget, ami automatikus típuskonverzió áldozata lesz. Az, hogy a szöveg melyik része miként konvertálódik, az az adott session nyelvi beállításaitól függ. Abban meg soha nem lehetünk biztosak, kivéve ha épp most adtuk meg mi magunk. Törekedjünk a kifejezések írásakor arra, hogy függetlenítsük magunkat a session nyelvi beállításaitól. Mindenképpen használjuk a konvertáló függvényeket! Ha beírunk valamit, akkor szövegből kell dátummá konvertálni: TO\_DATE függvény (ez mindig fontos!). Ha kiíratunk egy dátumértéket, akkor a dátumból kell szöveggé konvertálni: TO\_CHAR (előfordulhat, hogy elfogadjuk az aktuális beállításokat, bármi is legyen az). Összehasonlításánál mindenképpen azonos adattípus szerepeljen mindkét oldalon! Ha dátumok sorrendje a kérdés, akkor mindenképpen dátum szerepeljen mindkét oldalon (ne szövegeket rendezzünk abc sorrendbe). A dátum formátum kódokat nézzük meg a dokumentációból (ismétlés). Bevitelnél biztonságosabb a számok használata, mert a megnevezéseknél a nyelvi beállítások megint okozhatnak fennakadást (milyen nyelven írjuk ki a hónapot pl., hány karakteresre van kiegészítve szóközökkel).

1. Futtassuk le az alábbi utasításokat a dátumok különböző megjelenítésére:

SELECT SYSDATE FROM dual;

Ezt az első utasítást mi magunk futtassuk le a táblán különböző session-ökben (még az óra előtt nyissunk külön session-öket, és mindegyikben állítsunk be eltérő formátumot. Pl.:   
ALTER SESSION SET NLS\_DATE\_FORMAT='MM.DD.RR';  
ALTER SESSION SET NLS\_DATE\_FORMAT='YY. MONDD. HH:MI';  
ALTER SESSION SET NLS\_DATE\_FORMAT='DD. DAY HH24:MI:SS';  
)

SELECT TO\_CHAR(SYSDATE, 'HH24:MI:SS.') FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'YYYY.MM.DD.') FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'DDD' ) AS evnapja FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'IW' ) AS evhete FROM DUAL;

SELECT TO\_CHAR(SYSDATE, 'Q' ) AS negyedev FROM DUAL;

Táblát létre tudunk hozni „másolással” is. A CREATE TABLE után egy select utasítást használva. Másoljuk le a DOLGOZO táblát a MINTA sémából. A további feladatok erre vonatkoznak majd.

1. Hozzuk létre a DOLGOZO táblát az azonos nevű tábla másolásával a MINTA sémából! Nézzük meg a szerkezetét!

CREATE TABLE dolgozo AS SELECT \* FROM minta.dolgozo;

DESC dolgozo

SELECT \* FROM dolgozo;Kérdezzük le márciusban született dolgozók adatait!

1. Kérdezzük le márciusban született dolgozók adatait!

SELECT \* FROM dolgozo

WHERE to\_char(szuldat,'MM')='03'

;

1. Írassuk ki azokat a dolgozókat, akik 2010 július elseje előtt kerültek a céghez!

SELECT \* FROM dolgozo

WHERE felv\_nap<to\_date('10-07','RR-MM')

ORDER BY felv\_nap DESC; --2010 február, március és májusi adatok is bekerültek

/\*Egy H I B Á S megoldás: (mit okozhat, ha dátumsorrend helyett abc sorrendet

vizsgálunk)\*/

SELECT \* FROM dolgozo

WHERE to\_char(felv\_nap,'RR.MONTH.DD')<'10.JÚLIUS .01'

ORDER BY felv\_nap DESC; --2010-ből csak február maradt benne, mert F előbb van, mint J. Az M viszont hátrébb...

/\*Nem csak szöveges és szám átírás miatt:\*/

SELECT \* FROM dolgozo

WHERE felv\_nap<to\_date('10.JÚLIUS .01','RR.MONTH.DD')

ORDER BY felv\_nap DESC; --Így is jó!

**Önálló feladatok**

1. Kérdezzük le a férfi dolgozókat a felvételük napja szerint csökkenő sorrendben.
2. Kérdezzük le a női dolgozók keresztneveit abc szerint növekvő sorrendben!
3. Kérdezzük le ismét a női dolgozók keresztneveit, de most ismétlődés nélkül!
4. Kérdezzük le azokat a dolgozókat, akiknek a keresztneve Krisztina, vagy a vezetékneve Kiss!
5. Kérdezzük le a januári születésű férfi dolgozók vezetéknevét és keresztnevét!
6. Írassuk ki a 2010 és 11 között hozzánk került dolgozókat!
7. Írassuk ki azokat a dolgozókat, akik elsején kerültek a céghez, jelenítsük meg a felvétel napját is!
8. Írassuk ki a dolgozók azonosítóját, nevét és a születésnapjukat kiírt hónapnévvel!
9. Írassuk ki az előző listát születésnap szerinti sorrendben! (Vigyázat! Januárral kezdődjön, ne áprilissal!)
10. Írassuk ki a dolgozókról, hogy milyen napon születtek (hétfő, kedd stb.)
11. Írassák ki, hogy a saját születésnapjuk milyen napra esett!
12. Írassuk ki, hogy az egyes dolgozók hány napja dolgoznak a cégnél! A dátumot idő nélkül a TRUNC függvénnyel kapjuk meg.
13. Kérdezzük le azokat a dolgozókat, akik 1980 előtt születtek, de 2010 után kerültek a céghez.
14. Kérdezzük le azokat a dolgozókat, akik 20 éves koruk előtt kerültek a céghez!
15. Kérdezzük le azokat a férfi dolgozókat, akik a születésnapjuk hónapjában kerültek a céghez!