Gyakorlati óravázlat 04 – Szövegkezelés, oszlopfüggvények a lekérdezésben

1. Dátumfüggvények (a konvertáláson túl is), dátumaritmetika
2. Like operátor
3. Szövegfüggvények
4. Aritmetikai függvények

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

**Dátum aritmetika**

A múlt órán kiveséztük a dátumok ide-oda alakítását. Amiről nem volt szó, hogy milyen módon tárolja az Oracle a dátumokat. Az úgynevezett Julián naptárszámítást veszi alapul, aminek a lényege, hogy a napok sorszámok „az idők kezdetétől” és a napon belül az időt törtszámokkal tároljuk, pl. 0.5 az pont 12:00.

Az idők kezdete Excelben pl. 1900. január 1., de remélhetőleg köztudott, hogy az Excel dátumkezelése nem éppen világbajnok. Ezzel szemben az Oracle ennél kicsit nagyobb távlatú. Le tudjuk kérdezni a Julián formátumként értelmezett 1-et, és megkapjuk, hogy az első nap I.E. 4712. január 01. Ha megpróbálnánk 0-t írni, akkor hibaüzenetet kapunk, amiből a legnagyobb szám is kiderül (I.SZ. 9999. december 31.). Ez a kettő szerepel az első mintalekérdezésben.

Ha ezt tudjuk, akkor a dátumokkal különböző műveleteket is tudunk végezni. Pl. kivonni őket egymásból, számértéket hozzáadni. Természetesen az osztásnak és a szorzásnak nincs értelme illetve két dátumot sem tudunk összeadni, így ezek a műveletek nem is alkalmazhatóak.

Nézzünk néhány dátum függvényt! (Dokumentáció: 7. fejezet, Single-row funcitons – magyarul oszlopfüggvénynek hívják, ami egy elég buta elnevezés, Datetime functions). Nagyon hasznosak tudnak lenni ezek a függvények pl. határidők számítgatásánál. Az Oracle pl. abszolút tudja, hogy mikor van szökőév. Amiket jó lenne ismerni:

* SYSDATE, SYSTIMESTAMP: ezekkel már dolgoztunk, nincs argumentumuk
* ADD\_MONTHS: ha az eredményhónap rövidebb, mint a kiinduló dátum napja, akkor az adott hónap utolsó napját kapjuk. Csak egész hónapokat lehet kivonni és hozzáadni.
* LAST\_DAY: az adott dátum hónapjának utolsó napja
* MONTHS\_BETWEEN: két dátum között eltelt idő hónapokban, törthónapokat is eredményül ad. Életkor meghatározására a legbiztosabb módszer, a 365.25-tel való osztás is csak közelítő pontosságú, de ha a köztes hónapokat osztjuk 12-vel, az teljesen pontos (nincs igazán jelentősége, ritkán lehet szükség tűpontos életkor meghatározására)
* NEXT\_DAY: csak a nyelvi beállítások ismeretében működik, mert a hét napját szövegesen kell megadni
* TRUNC: levágás az adott dátumelemig (az adott időszak első napjának 0:00:00 idejét adja vissza)
* ROUND: az adott dátumelemig lefelé vagy felfelé kerekít.

**Aritmetikai függvények**: itt is van ROUND és TRUNC, egyébként nem kell vele túl sokat foglalkozni, nézzük meg a listát a dokumentációban. Látni fogjuk, hogy mindenféle matematika függvények megtalálhatóak.

**Feladatok**

1. Kérdezzük le az Oracle-ben tárolható legkisebb és legnagyobb dátumot!
2. Kérdezzük le, hogy milyen nap van ma, mi volt tegnap és mi lesz holnap!
3. Kérdezzük le, hogy az aktuális hónap utolsó napja milyen napra esik.
4. Kérdezzük le kétféleképpen, hogy mi volt a dátum 4 évvel ezelőtt (a közbeeső szökőév figyelembe vételével és anélkül).
5. Kérdezzük le a MINTA.IRODALOM táblából az írók nevét, születési és halálozási dátumát (teljesen kiírva) és kétféleképpen azt, hogy hány évet éltek. Az évek számát 3 tizedesre kerekítve írjuk ki. (az lesz a tanulság, hogy kár ezzel szórakozni, mert minimális az eltérés)

**Szövegkezelés**

Előfordul, hogy olyan karaktert kell használnunk valahol, amely önmagában vezérlést segít. Az alapértelmezett ún. escape, azaz kivételkezelő karakter az aposztróf ('), ami amúgy a szöveghatároló elem is, merthogy egy normál szövegben kizárólag ez, az aposztróf zavarná az Oracle-t. Tehát Jeanne d'Arc Oracle megfelelőjének leírása így néz ki:

SELECT 'Jeanne d''Arc' AS Orleans\_i\_szuz FROM DUAL;

A szövegösszefűző operátor: ||, nézzünk szövegfüggvényeket is. Amiket tudni kell: **LENGTH, UPPER, LOWER, INITCAP, SUBSTR**, **TRIM**, LTRIM, RTRIM, LPAD, RPAD, INSTR, REPLACE. Amit vastaggal írtam, azt teljesen fejből kell tudni, a többihez szabad puskázni a dokumentációból.

A like irányított operátor, a reguláris kifejezés csak a jobb oldalon állhat! Említsük meg a regexp\_like-ot is, de nem fogjuk számon kérni. Dokumentációban önszorgalomból meg lehet nézni, hogy hogyan működik, mert sokszor hasznos lehet. (6. fejezet: conditions, azon belül Pattern-matching Conditions – mintaillesztéses feltételek)

Like-nál % helyettesít akárhány karaktert, \_ helyettesít egyetlen karaktert.

Csak akkor használjuk a like-ot, ha tényleg mintát akarunk illeszteni, különben elég lesz az = is.

**Gyakvezéreknek:**

Előfordul, hogy olyan karaktert kell használnunk valahol, amely önmagában vezérlést segít. Az alapértelmezett ún. escape, azaz kivételkezelő karakter az aposztróf ('), ami amúgy a szöveghatároló elem is, merthogy egy normál szövegben kizárólag ez, az aposztróf zavarná az Oracle-t. Tehát Jeanne d'Arc Oracle megfelelőjének leírása így néz ki:

SELECT 'Jeanne d''Arc' AS Orleans\_i\_szuz FROM DUAL;

Eddig minden szép és jó, de a LIKE esetében két speciális karakter még van, az \_ és a %. Nos, itt általában nem lehet az aposztróf kivételkarakter, mert nem tudnánk megkülönböztetni a szöveg végétől. Ezért vezették be az ESCAPE kulcsszót mellé:

SELECT 'Lefutott' AS megmi FROM DUAL

WHERE 'VAT 18%' LIKE '%18\%%' ESCAPE '\';

Fontos tudni, hogy a LIKE irányított operátor, azaz reguláris kifejezés csak a jobb oldalon állhat, a baloldali operandust minden körülmények között szövegként értelmezi:

SELECT 'Lefutott' AS megmi FROM DUAL

WHERE '18%' LIKE '1880';

Nem kell tudni, de az NLS miatt van LIKEC, LIKE2, LIKE4 operátor is – első az UTF, az utolsó kettő pedig a Unicode Microsoft implementációját, az UCSC-2 és UCSC-4 szerinti illesztést támogatja. Mondtam, hogy nem kell tudni :).

Lehet ma már ennél bonyoltabb reguláris kifejezést is megadni a 11g változattól kezdve, mert amúgy ez nagyon-nagyon hasznos szövegfeldolgozásoknál. Sszámos szövegfüggvénynek van REGEXP alakja is, például:

SELECT orszag FROM országok

WHERE REGEXP\_LIKE( orszag, '^Ausztr(\*|ál)ia$' );

A reguláris kifejezést a Perl szintaxisának megfelelően lehet előírni – ebbe nem fogunk belemenni.

Mi történik, ha a where feltétel konstans kifejezést tartalmaz? Ha konstans, akkor nyilván el tudjuk dönteni, hogy igaz vagy hamis. Ha igaz, akkor a dual táblából megkapjuk egyszer a szelektlistában szereplő értéket. Ha hamis, akkor 0 sort kapunk eredményül. (A példákban szereplő fejléc, a MEGMI történelmi maradvány Kardkovács Zsolttól. Azért hagytam benne a példákban ezt a rettenetes szóviccet, mert sajnos a tesztfeladatinkban is ez szerepel, és nem lesz energiám az összeset átírni. Így viszont jobb, ha ugyanazt látják már órán is. Természetesen nincs jelentősége.)

A kisbetű-nagybetű váltó függvények használatának jelentőségéről: egyrészt kiíráskor esztétikai kérdés, hogy meghagyjuk az eredeti írásmódot, vagy egységesítünk. Másrészt feltétel megfogalmazásakor fontos, mert a szövegfüggvények case sensitivek. Ha tehát pl. szövegegyezést, vagy mintaillesztést nézünk és azt szeretnénk, hogy azokat bármilyen írásmóddal felismerje, akkor az attribútum írásmódját mindenképpen egységesíteni kell, pl. csupa kisbetűsre és akkor a feltétel másik felében is csupa kisbetűsnek kell lennie. (A dual-os mintapéldákban a függvény működésének megértése miatt konstans szövegekre is használjuk ezeket a függvényeket, de valódi lekérdezéseknél ez ugye felelsleges, írhatjuk eleve a kívánt írásmóddal a szöveget – hibának persze nem hiba, de minek)

A legtöbbhelyen működő bal, jobb, közép függvények helyett oracle-ben SUBSTR van csak (résszszöveg), de azzal mindhárom probléma megoldható. Ha a második paraméter negatív, akkor hátulról számolja a karatereket, így nem szükséges a LENGTH függvény beágyazása.

A TRIM függvénynek csak az alapesetét kell fejből tudni, amikor az elejéről meg a végéről a szóközöket szedi le. A példába írtam egy olyat is, amikor kötőjeleket kell leszedni, de azt elég a dokumentációból tudni (ki is akadtam, hogy ez így működik!).

**Feladatok**

1. Nézzük meg az sql fájlban lévő mintapéldákat és fejtsük meg őket!
2. Fejtsük meg az alábbi lekérdezéseket. Mi történik, ha a WHERE feltétel konstans kifejezést tartalmaz?
3. Kérdezzük le a minta séma dolgozo táblájából azokat, akiknek a nevéből kiderül, hogy feleségek! Rendezzük névsorba!
4. Kérdezzük le ugyanazt, mint az előbb, de egységes írásmóddal irassuk ki a neveket!
5. Kérdezzük le a minta séma dolgozo táblájából az A és Á kezdőbetűs keresztnevű dolgozókat!
6. Kérdezzük le a minta séma dolgozo táblájából a Tiborokat!
7. Kérdezzük le a minta séma dolgozo táblájából a dolgozók nevét összefűzve és kiegészítve 20 karakterre \_ karakterekkel!

**Önálló feladatok**

1. Hozzuk létre az ORSZAGADATOK táblát a MINTA séma táblájának másolásával. A további feladatok egy része erre fog vonatkozni.
2. Írjuk ki az országok neveinek hosszát!
3. Írjunk lekérdezést, amely az országok neveinek csak a kezdőbetűjét hagyja meg nagybetűnek, a többi kisbetűs legyen!
4. Listázzuk ki a váltópénzek neveit – az ott lévő számok nélkül!
5. Listázzuk ki az európai országokat! (LIKE operátor használatával!)
6. Kérdezzük le az ALL\_TABLES katalógustáblából azokat a kurzusrésztvevőket, akiknek van ÁRFOLYAM táblájuk (találjuk meg azokat is, akik az első betűt máshogy írták)!
7. Kérdezzük le a MINTA séma CÉGEK táblájából azokat a cégeket, ahol a cégnévben szerepel a a buda szórészlet!
8. Kérdezzük le a MINTA séma CÉGEK táblájából azokat a nem budapesti cégeket, akiknek a nevében szerepel a buda szórészlet!
9. Kérdezzük le a MINTA séma CÉGEK táblájából a részvénytársaságokat! Vigyázat, a Zrt-k és Nyrt-k kellenek, de azok a cégek ne kerüljenek bele a lekérdezésbe, amelyeknek a nevében véletlenül van benne az rt betűpáros!
10. Kérdezzük le a MINTA séma CÉGEK táblájából azokat a cégeket, amelyeknek a nevében szerepel a magyar, vagy Hungary megjelölés és ebben az évezredben alapították őket!
11. Kérdezzük le a MINTA séma táblájából a cégek adatait és azt, hogy idén mikor lesz az üzleti év vége!
12. Listázzuk ki a MINTA séma ALKALMAZOTT táblájából a neveket kötőjelek között úgy, hogy a keletkező szöveg hossza 32 karakter legyen, és pont középen van a név!   
    Például: „Gipsz Jakab”-ból legyen: „----------Gipsz Jakab----------” (az idézőjelek nélkül)!
13. Kérdezzük le a MINTA séma IRODALOM táblájából a neveket szétbontva vezeték és keresztnévre!
14. Számoljuk ki helyesen az SQL segítségével, évtől függetlenül azt az időpontot, amikor az óraátállítást el kell végezni, óra pontossággal. (Március és október utolsó szombatján, 2h-kor, illetve 3h-kor kell átállítani az órát.