

8.-11. gyakorlat – emlékeztető

Sidló Csaba

2005. november 21.

1. SQL*Plus kliens

ld. gyakorlat valamint más segédanyagok

2. Oracle SQL

A következőkben az Oracle adatbázis SQL nyelvének SQL'92 szabványnak megfelelő részhalma-zára nézünk feladatokat, bár néhány SQL bővítés túlmutat az SQL'92 szabványon. A témakö-rökön végighaladva viszonylag használható SQL tudást szedhetünk fel.

2.1. Adatbázis objektumok, sémák

Az adatbázis szolgáltatásait adatbázis-objektumokon keresztül vesszük igénybe. Számunkra lé-nyeges objektumok a *táblák*, *nézettáblák*, *indexek* és a *szinonímák*. Az objektumokat ún. sémákba csoportosítjuk, egy séma pedig egy adott felhasználóhoz tartozik. A user létrehozásával implicit létrejön egy, a nevével megegyező séma, aminek segítségével minősíthetjük a user objektumait (például SCS.SZERET: SCS séma SZERET táblája).

2.2. SQL utasítások csoportjai

Az SQL nyelv utasításkészlete több csoportra bontható:

- lekérdezés (*QL*): lekérdező műveletek **SELECT**,
- adatdefiníció (*DDL*): relációs séma szintű utasítások, mint **CREATE TABLE**, **CREATE INDEX**, **ALTER TABLE**, **DROP TABLE**, ...,
- adatmanipuláció (*DML*): relációs séma szintű utasítások, mint **INSERT**, **UPDATE**, ...,
- adatvezérlő nyelv (*DCL*): jogosultságok, tranzakciókezelés, mint **GRANT**, **COMMIT**, **ROLLBACK**, ...

2.3. Kommentek

Egy SQL utasítás ellátható kommentekkel,

- **/*** kommentálok ***/** : többsoros komment,
- **-** kometntálok : egysoros komment
- **REMARK** kommentálok: SQL*Plus utasítás, egysoros komment (pl. utasítások közé csak ilyen tehető)

2.4. Literálok

Literálok: konstans értékek.

- karakterlánc literálok: 'Valami'
- egész szám: 1230, -20
- tetszőleges szám: +6.34, 0.5, 25e-03 stb.

2.5. Alapvető beépített adattípusok

- CHAR(méret): fix hosszú karakterlánc; hossz megadása kötelező.
- VARCHAR2(maximális méret): változó hosszúságú karakterlánc; hossz megadása kötelező.
- NUMBER(precision, scale): fixpontos szám értékek; precision a számjegyek száma, scale a tizedespont jobb oldalán álló számjegyek száma. Megkötés nélkül is használható (NUMBER). Negatív scale érték esetén a szám kerekített az adott pontossággal. A NUMBER(p) ekvivalens NUMBER(p, 0)-val, valamint NUMBER(36) ekvivalens INTEGER-rel és INT-tel.
- FLOAT: lebegőpontos szám.
- DATE: dátum típus; mindig egy (év, hónap, nap, óra, perc, másodperc) időpillanat megfelelője.
- BLOB: Binary Large Object; tetszőleges, akár nagy mennyiségű, struktúrátlan adat tárolására alkalmas típus (pl. képek, hangok stb.). Maximum 2GB méretű lehet egy-egy eleme. CLOB: byte-sorozat (character) large object.

2.6. Adatbázis táblák

Alapvető adatbázis objektumként mi a táblákkal foglalkozunk legtöbbit. A tábla megfelel a relációs adatmodell reláció fogalmának, ennek megfelelően beszélhetünk a szerkezetéről mint relációsémáról. Ezt az SQL*Plus környezet DESC parancsával tudjuk lekérdezni (pl. DESC scs.szeret). A táblák adatmodellje a következőkben különbözik a hagyományos relációs adatmodelltől:

- A tábla úgynevezett "multihalmaz", tehát egy sor többször is előfordulhat benne.
- A tábla típusos, tehát az attribútumokhoz egyértelműen tartozik egy-egy SQL adattípus.

Adott user tábláit lekérdezhethetjük az úgynevezett rendszertáblák segítségével, például `select table_name from dba_tables where owner='SCS' ;`.

2.7. SELECT és alkérdések

A SELECT utasítás adatbázis táblák lekérdezését szolgálja, és durván nagy közelítésben a következőképp nézhet ki:

```
SELECT
    attribútumlista
FROM
    táblák listája
WHERE
```

feltételek
GROUP BY
 csoportosító attribútumok
HAVING
 csoportra vonatkozó feltételek
ORDER BY
 rendezés attribútumai

Ebben bárhová, ahol táblák (relációk) szerepelhetnek, (némi megkötésekkel) behelyettesíthető tábla helyett alkérdés (subquery) is, tehát egy újabb select utasítással definiált reláció. Így beágyazható **SELECT** a táblák listájába és a **WHERE** feltételek közé is. Példák beágyazott lekérdezésekre:

- ... FROM (SELECT nev FROM szeret) nevek ...,
- ... WHERE eletkor <= (SELECT eletkor FROM hallgatok WHERE csoport=4) ...,
- ... WHERE gyumolcs NOT IN (SELECT gyumolcs FROM szeret WHERE nev='Malac') ...,
- ... WHERE EXISTS (SELECT gyumolcs FROM szeret sz2 WHERE nev = sz2.nev) ...
- ... (SELECT nev FROM szeret1) UNION (SELECT nev FROM szeret2) ...

2.8. Szinonímák

A szinoníma adatbázis objektumok alternatív elnevezése. Például

- **CREATE SYNONYM szeret FOR scs.szeret** : user-re lokális szinoníma scs szeret táblájára,
- **CREATE PUBLIC SYNONYM szeret FOR scs.szeret** : publikus, minden userre kiterjedő szinoníma.

2.9. SELECT: ismétlődések kiszűrése

Fontos fogalom a *multihalmaz*: az adatbáziskezelő ennek megfelelő relációs modellt használ, tehát két azonos sor is szerepelhet egy relációban. A **SELECT DISTINCT** utasítás olyan típusú lekérdezés, ami halmaz eredményt ad.

2.10. Feltételek

Feltételek szerepelhetnek például a **SELECT** utasítás **WHERE** tagja után (vagy a későbbiekben **DELETE**, **UPDATE** utasításoknál, valamint a **SELECT ... GROUP BY ... HAVING ...** konst-runkcióban).

Csoportjai:

- egyszerű összehasonlító feltételek: kifejezések közötti =, !=, <>, <, >, <=, >=,
- csoportos összehasonlítás feltételek: ANY, SOME, ALL,
- tartalmazás feltétel: IN, NOT IN,
- tartomány feltételek: BETWEEN,

- null feltételek: IS NULL, IS NOT NULL,
- lebegőpontos feltételek: IS (NOT) NAN, IS (NOT) INFINITE,
- egyéb: EXISTS, NOT EXISTS, IS NULL, IS NOT NULL, LIKE, REGEXP_LIKE,
- valamint a logikai kifejezésekkel összekapcsolt kifejezések (AND, OR).

2.11. SELECT: attribútumok átnevezése

2.12. Direktszorzat és összekapcsolás

Alap változat: táblák felsorolása a FROM kulcsszó után (eddig direktszorzat), majd az összekapcsolási feltételek megadása a WHERE után, valamint a projektálandó oszlopok kiválasztása a SELECT után.

Az összekapcsolásban részt vevő táblák megkülönböztetésére azokat át tudjuk nevezni, ennek segítségével hivatkozhatunk rájuk a SELECT utasításon belül, például

```
SELECT
    t.nev, m.megye_nev
FROM
    ksh_telep t, ksh_megye m
WHERE
    t.megye_id = m.megye_id
```

2.13. DUAL tábla

A DUAL tábla SYSTEM tulajdonában van, de közvetlenül hivatkozható bárki által, egy DUMMY nevű attribútummal rendelkezik valamint egy adatsorral, ami egy 'X' értéket tartalmaz. Példa DUAL tábla használatára:

```
SELECT
    3*5 Eredmény
FROM
    DUAL
```

2.14. NULL érték

A NULL érték egy olyan speciális érték, amit bármely adattípusú attribútum felvehet értékül, jelentése pedig az, hogy az adott adat nem ismert. Nem ekvivalens a 0-val, összehasonlítani sem lehet más adatokkal (az eredmény mindig UNKNOWN). Az egyetlen összehasonlító operátor a kezeléséhez a NULL, NOT NULL páros.

2.15. Beépített függvények

- Sorfüggvények:
 - szám → szám függvények: pl. ABS, FLOOR, SIN, TAN, ROUND, TRUNC, MOD, ...
 - karakter → karakter függvények: pl. CHR, UPPER, SUBSTR, REPLACE, ...

- karakter → szám függvények: pl. LENGTH, INSTR, ...
- dátum függvények: pl. SYSDATE, ...
- konverziós függvények: pl. TO_DATE, TO_CHAR, TO_NUMBER, ...
- egyéb: pl. USER, UID, ...
- Aggregáló függvények: több sor → egy sor; pl. AVG, COUNT, MAX, MIN, SUM, VARIANCE, ...
Soknál lehetséges a DISTINCT illetve az ALL opciók használata, pl. COUNT(DISTINCT nev).

Példa sorfüggvény használatára:

```
SELECT
    sz.nev, LENGTH(sz.nev) NÉV_HOSSZ
FROM
    szeret sz
```

Példa aggregáló- és sorfüggvény használatára:

```
SELECT
    COUNT(DISTINCT nev) ALMA_SZERETŐK_SZÁMA
FROM
    szeret
WHERE
    UPPER(gyumolcs) = 'ALMA'
```

2.16. Operátorok

Rendelkezésünkre állnak unáris és bináris operátorok, aritmetikai (+, -, * ...), összehasonlító (=, <, >, <>, LIKE, ...), logikai (AND, OR, ...), és halmaz-operátorok (UNION, INTERSECT, ...).

A jegyzetben máshol nem előforduló, de hasznos operátorok közül kiemelném a || operátort, aminek segítségével karakterláncokat fűzhetünk össze.

2.17. Konverziók

Adatkonverziós függvényekre adatfeldolgozás során időnként kikerülhetetlenül szükségünk van. Gyakran az adatbázis szerver implicit végrehajt konverziókat, de néha explicit módon nekünk kell ezt megfogalmaznunk. Néhány péda konverzióra:

- TO_CHAR(SYSDATE, 'YYYY'): az épp aktuális év,
- TO_DATE('20040901', 'YYYYMMDD'): sztring átalakítása dátummá,
- TO_NUMBER('198.5'): karakterláncból szám stb.

A TO_CHAR, TO_DATE konverziókhoz kapcsolódó fogalom az ún. *format model* (pl. 'YYYYMMDD' format modelnek megfelelő dátum az '19981104'), használatához lásd az Oracle doksit.

2.18. Halmazműveletek

Azonos sémájú relációk (pontosabban sorrendben azonos típusú attribútumokkal rendelkező lekérdezések, táblák között) értelmezhetők a következő halmazműveleteket:

- UNION: a két operandus sorainak halmaza,
- UNION ALL: multihalmaza,
- INTERSECT: metszete,
- MINUS: különbsége.

Az operátorok precedenciája megegyezik, ha nincs zárojelezéssel egyértelműsítve, akkor a kiértékelés balról jobbra történik.

2.19. SELECT: rendezés

Egy SELECT lekérdezés sorait rendezhetjük az ORDER BY után megadott attribútumoknak megfelelően.

- ORDER BY nev: név szerinti növekvő sorrend,
- ORDER BY nev DESC: név szerinti csökkenő sorrend,
- ORDER BY nev ASC, kor DESC: név szerinti növekvő, ezen belül kor szerint csökkenő sorrend,
- ORDER BY 2 ASC, 3 DESC: 2. attribútum szerinti növekvő, ezen belül a 3. attr. szerint csökkenő sorrend,

Példa rendezésre:

```
SELECT
    nev, gyumolcs
FROM
    szeret
ORDER BY
    nev ASC, gyumolcs DESC
```

2.20. Csoportosítás

A GROUP BY ... HAVING ... záradék használatával lehetőségünk nyílik egy lekérdezés eredményében csoportokat kialakítani. A GROUP BY után megadhatunk attribútumokat, ezek szerint jönnek létre a csoportok, majd a csoportokra a HAVING segítségével feltételeket is adhatunk. A lekérdezés eredménye a csoportosító attribútumok felvett értékei, valamint esetleges csoporton belüli aggregációk eredményei lesznek. A SELECT után ezért nem sorolhatunk fel csak csoportosító attribútumokat, vagy a csoportokra vonatkozó aggregációs függvényeket.

Példa csoportosításra:

```

SELECT
    nev, COUNT(DISTINCT gyumolcs) HÁNY_GYÜMÖLCSÖT_SZERET
FROM
    szeret
GROUP BY
    nev
HAVING
    COUNT(DISTINCT gyumolcs) > 1

```

A csoportosításhoz hasznos kiegészítések a CUBE, ROLLUP operátorok, valamint a GROUPING függvény.

2.21. Külső összekapcsolások

Két tábla külső összekapcsolása esetén azokat a sorokat kapjuk eredményként, melyek teljesítik az összekapcsoló és egyéb feltételeket, valamint azokat, melyekhez az egyik táblához kapcsolt másik táblában nincs megfelelője. A külső összekapcsolásra a WHERE összekapcsoló feltételében a megfelelő oszlop minősítésével ((+) operátor) nyílik lehetőségünk. Ha az összekapcsolás több feltétellel történik, mindegyiknél minősíteni kell a megfelelő attribútumokat.

Például, listázzuk ki a Magyar Posta adatbázisából a település-részeket, valamint a hozzájuk tartozó népeiséget a KSH adatbázisa alapján. Jelenítsük meg azonban azokat a település-rész neveket is, amelyek valamilyen oknál fogva nem szerepelnek a KSH nyilvántartásában:

```

SELECT
    isz.telepules_resz, t.nepesseg
FROM
    ksh_telep t, ir_szam isz
WHERE
    isz.telepules_resz = t.nev (+)

```

A NULL attribútumértékek megjelenítéséhez hasznos lehet az NVL(expr1, expr2) függvény, ami visszaadja expr1-et, ha az nem NULL, és expr2-t, ha expr1 NULL.

- 2.22. Tábla létrehozása, törlése
- 2.23. Tábla módosítása
- 2.24. Beszúrás táblába, törlés táblából
- 2.25. Tábla adatainak módosítása
- 2.26. Nézetablák
- 2.27. Indexek
- 2.28. Tranzakciókezelés
- 2.29. Objektum szintű jogosultságok
- 2.30. Összekapcsolások és félig-összekapcsolások a szabványos JOIN operátorral
- 2.31. Rekurzív lekérdezések