

3. MELLÉKLET

Bevezetés az SQL*Plus környezet, és az SQL nyelv használatába

Tartalomjegyzék

Bevezetés.....	1
Belépés az SQL*Plus környezetbe	2
Felhasználói táblák lekérdezése.....	4
Felhasználói katalógus és adatszótár	4
Felhasználói adattábla lekérdezése	4
Felhasználó váltás.....	5
Szkript programok írása, futtatása	5
Szövegszerkesztés az SQL*Plus környezetben	5
A login szkript létrehozása	5
A login szkript futtatása	6
Általános szkript program létrehozása és futtatása	6
Kiírás szkript programból	7
Megjegyzések használata	7
A sorfolytató jel használata	7
Változók használata	8
Az SQL*Plus editorának lecserélése	9
Tábla-létrehozás-feltöltés-törlés, dátumformák.....	9

Bevezetés

Az alábbiakban rövid bevezetést adunk az Oracle adatbázis-kezelő rendszer fejlesztői (egygépes telepítésű) Personal Oracle változata, az SQL*Plus környezet, valamint e környezetben az SQL nyelv használatába. Ennek az anyagrésznek a feldolgozását célszerű számítógépen követni. Előtte azonban javasoljuk, hogy az Olvasó tanulmányozza át a 2. mellékletben az Oracle rendszer alaptábláinak felépítéséről és értelmezéséről leírtakat.

A továbbiakban bemutatásra kerülő utasítások egyrészt részletesen megtalálhatóak a [16]-ban, másrészt e példatár különböző fejezeteiben.

Belépés az SQL*Plus környezetbe

Az SQL*Plus környezet egy interaktív parancsnyelvi felület, melynek segítségével SQL parancsok közvetlenül végrehajthatók, formázott listák készíthetők, több utasításból álló, fájlban tárolható szkript programok szerkeszthetők és futtathatók. E szkript programokban változók definiálhatóak, melyek révén interaktívvá tehetők, sőt még az Oracle szerveroldali programozását lehetővé tevő PL/SQL programok is elhelyezhetők benne.

Megjegyezzük, hogy e parancsnyelvi felület a Windows operációs rendszer alatt "windows"-osan viselkedik (létezik ugyanis DOS alatt futó változata is). Tehát például a PgUp, PgDn, ↑, ↓ billentyűkkel hátra-előre lehet mozogni a felületen, a Ctrl-C és Ctrl-V billentyű-kombinációkkal a szokásos módon lehet akár e felületen belül, akár kifelé (bármilyen szöveges fájlba, vagy fájlból) szövegeket másolni (ráadásul igen kellemes, oszlopkivágásos módon).

Az SQL*Plus környezet tulajdonképpen egy fejlesztői eszköz, melynek segítségével ötletek, megoldások kipróbálhatók annak érdekében, hogy azok később valamilyen alkalmazói rendszer részei legyenek. Előfordul persze az is, hogy egy hirtelen jött alkalmazói feladatot kell megoldani, például egy szépen formázott, jól dokumentált listát készíteni, vagy egy kis interaktív programot készíteni, amely néhány tipikus feladat megoldására alkalmas, mégis könnyen kezelhető, stb. Az SQL*Plus környezet és szkript programjai erre is remekül megfelelnek. Mindebből pedig már következik, hogy az SQL*Plus környezet mind az alapszintű SQL tanulásához, mind a professzionális Oracle megismeréséhez ideális eszköz.

Nézzük tehát, hogyan lehet belépni az SQL*Plus környezetbe! A Personal Oracle telepítése után az SQL*Plus programot a következőképpen találhatjuk meg:

- egy kattintás a "Start" gombra,
- egy kattintás a "Programok" menüpontra,
- egy kattintás az "Oracle – Orahome 92" mappa névre,
- egy kattintás az "Application Development" mappa névre és máris megtaláltuk az "SQL Plus" ikont.

Ezekután vagy kettőt kattintunk erre az ikonra (elindítva ezzel az SQL*Plus programot), vagy (és a magunk részéről inkább ezt javasolnánk) levonszoljuk az asztalra (desktop) a Ctrl billentyű és az egér bal gombjának egyidejű nyomva tartásával. Amikor később használni akarjuk, elég erre az ikonra kattintani.

Az SQL*Plus program indításakor megjelenik a Bejelentkezés ablak. Ennek értelmezéséhez tudni kell, hogy az Oracle egy professzionális rendszer, ezért csak pontosan tisztázott jogkörű felhasználók léphetnek be, ők is csak felhasználói név és jelszó megadása után. A bejelentkező felület

ahol Personal Oracle esetén csak az első két mező kitöltése szükséges. (Az SQL*Plus program kliens-szerver telepítés esetén is használható, akkor azonban a harmadik mezőt is ki kell tölteni, ezzel most nem foglalkozunk.)

A legtöbb joggal az úgynevezett adatbázis-adminisztrátor rendelkezik. Ő tetszés szerint (az adott számítógép hardver lehetőségein és az operációs rendszer képességeinek korlátain belül) átkonfigurálhatja az adatbáziskezelő rendszert, új felhasználókat definiálhat, ezeket jogokkal láthatja el, új adatbázis-területeket hozhat létre, stb. (Lásd a 11. fejezetben.) Adatbázis-adminisztrátorként alapértelmezésben a Sys felhasználó névvel kellett belépni a `change_on_install` jelszóval. A 9.2. verzióban azonban a telepítő rendszer e jelszó használatát már nem engedi (túl sokan ismerik), és a telepítés során egy új jelszót kell megadni. A Sys felhasználó név esetén a bejelentkezési felületen jelszóként megadható a

TelepítésiJelszó as sysdba

karaktársorozat, ahol a *TelepítésiJelszó* a telepítés során a Sys felhasználóhoz általunk bevezetett jelszó.

Korlátozottabb lehetőségekkel, de felhasználói és tábladefiníciós jogokkal rendelkezik a System felhasználó, akinek az alapértelmezés szerinti manager jelszavát szintén a 9.2. verzió már nem engedélyezi, helyette valami mást kell megadni telepítéskor. Ha system felhasználóként lépünk be az SQL*Plus környezetbe, akkor jelszóként a System felhasználóhoz általunk bevezetett jelszót kell beírni.

Az Oracle rendszer leggyakrabban használt mintadattábláinak tulajdonosa a Scott felhasználó, akinek *tiger* jelszava szerencsére nem változott az utolsó tíz évben, így ezzel már könnyűszerrel birtokba veheti az Olvasó az SQL*Plus környezetet.

Ha már valamelyik felhasználóként bejutottunk az SQL*Plus környezetbe, akkor bármelyik másik felhasználó környezetébe átléphetünk a

```
CONNECT FelhasználóNév/jelszó [AS {SYSDBA|SYSOPER}]
```

SQL*Plus utasítással, ahol az AS SYSDBA, illetve a AS SYSOPER záradékok valamelyikének megadása a Sys felhasználó esetén szükséges.

A sikeres belépés után megjelenik az SQL*Plus prompt jele (SQL>). Minden SQL és SQL*Plus utasítást ez után kell írni.

Az SQL*Plus környezetben (mint már jeleztük) egyaránt kiadhatók SQL utasítások, és a környezeti beállításokat végző, és a szkript programokat szerkesztő és futtató SQL*Plus utasítások. Megjegyezzük, hogy az SQL utasításokat pontosvesszővel (";") kell lezárni, míg az SQL*Plus utasításokat közvetlenül a sorvégjel zárja. Az utasítások használatával kapcsolatos további tudnivaló, hogy az SQL utasítások tördelhetők több sorba (sőt az áttekinthetőség érdekében célszerű is tördelni), míg az SQL*Plus utasításokat egyetlen sorba kell írni.

Felhasználói táblák lekérdezése

Felhasználói katalógus és adatszótár

Belépés után kérdezzük le a *felhasználói katalógust* annak érdekében, hogy megtudjuk a felhasználónak milyen adattáblái, illetve tábla jellegű objektumai vannak. Ezek lekérdezhetők az Oracle adatbázis nézeteiből. (Az alábbiakban az SQL*Plus környezetben kiadott utasítások elé odaírjuk az SQL*Plus készenléti, azaz prompt jelét, az "SQL> " karaktersorozatot)

```
SQL> SELECT * FROM user_catalog;
```

illetve a régebbi verziókban:

```
SQL> SELECT * FROM cat;
```

Ily módon megkapjuk mindazon táblákat, nézeteket, stb., amelyek az adott felhasználó tulajdonában vannak.

A későbbiek során majd érdekes lehet az egyéb objektumokat is tartalmazó *felhasználói adatszótár*, melyet a

```
SQL> SELECT * FROM user_objects;
```

utasítás segítségével kérdezhetünk le.

Felhasználói adattábla lekérdezése

Kérdezzük le ezután a Scott felhasználó tulajdonában levő emp tábla tartalmát:

```
SQL> SELECT * FROM emp;
```

Ha ez lesz az Olvasó első lekérdezése, akkor láthatja, hogy a lista több sorba tördelt, nem áttekinthető. Miért? Azért, mert az SQL*Plus környezetben a listázandó sor hossza nem megfelelő e tábla listázására. Kérdezzük le az SQL*Plus környezet sorhossz beállítását.

```
SQL> SHOW linesize
```

Az eredmény 80. Láthatóan ez kevés ahhoz, hogy az emp tábla minden oszlopa egy sorba kerüljön, azaz egy rekord egy sorban jelenjen meg. Állítsuk be a sorhosszt, mely egy SQL*Plus környezeti rendszerváltozó.

```
SQL> SET linesize 400
SQL> SELECT * FROM emp;
```

Az ekkor megjelenő lista már majdnem jó, mert egy rekordot egy sorba ír, függőlegesen azonban kétszer írja a fejléct, ez pedig zavaró. Tehát kérdezzük le a lapméretet:

```
SQL> SHOW pagesize
```

A kapott érték nyilván kevés az emp tábla egy fejléccel való megjelenítéséhez, tehát növeljük meg a lapméret rendszerváltozót.

```
SQL> SET pagesize 40
```

majd újra kérdezzük le az emp tábla teljes tartalmát:

```
SQL> SELECT * FROM emp;
```

Felhasználó váltás

Nézzük meg mi történik, ha újraindítjuk az SQL*Plus programot, de nem Scott felhasználóként lépünk be, hanem System-ként! Ha megpróbáljuk lekérdezni az emp táblát, nem fog sikerülni. Ennek az az oka, hogy az emp tábla a Scott "tulajdona". Lehetőségünk van azonban arra, hogy átlépünk egy másik felhasználó környezetébe (persze csak akkor, ha tudjuk a jelszavát). Adjuk ki az alábbi SQL*Plus utasítást

```
SQL> CONNECT scott/tiger  
Kapcsolódva.
```

Most lekérdezve az emp táblát, már meglelenik a szokásos lista.

Szkript programok írása, futtatása

Az iménti lista már megfelelő lesz számunkra (lásd a 2. mellékletet). Annak érdekében, hogy minden SQL*Plus környezetbe való belépéskor ez legyen a munkakörnyezet, létre kell hoznunk egy login.sql szkript programot a c:\oracle\ora92\bin alkönyvtárban (ha a C: meghajtóra és ora92 néven telepítettük az Oracle rendszert). Ez a szkript az SQL*Plus minden indításakor automatikusan lefut, ezáltal inicializálva a környezetet. Az alábbiakban megmutatjuk, hogy miként lehet egyszerűen létrehozni és futtatni egy tetszőleges SQL*Plus szkript programfájlt.

Szövegszerkesztés az SQL*Plus környezetben

A login szkript létrehozása

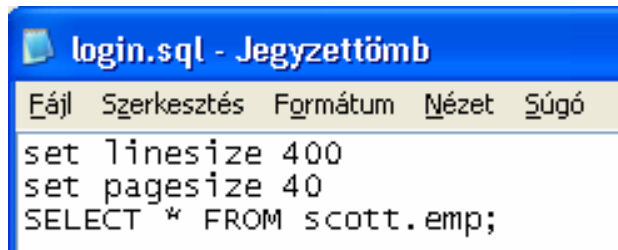
Mivel az SQL*Plus rendelkezik saját szövegszerkesztővel, a legegyszerűbb, ha azt használjuk. Ezt az ed *fájlnév* SQL*Plus utasítással hívhatjuk meg. Az általános gyakorlási célokra válasszunk valamilyen egyszerű programfájlnévet, például abc, vagy aaa, illetve a login.sql létrehozása érdekében login. Az

```
SQL> ed login
```

utasítás kiadása után a beugró ablakban megnyíló szövegszerkesztőbe írjuk be például a

```
SET linesize 400  
SET pagesize 40  
SELECT * FROM scott.emp;
```

sorokat



majd zárjuk be a szövegszerkesztőt (például Alt-F4 + Enter paranccsal, vagy a szokásos grafikus módon). Megjegyezzük, hogy az így szerkesztett szkript fájl automatikusan megkapja az sql kiterjesztést. (Ezúttal egyébként a Scott felhasználó tulajdonában lévő emp táblára scott.emp módon hivatkoztunk, amely módon már a System felhasználó is elérheti.)

A login szkript futtatása

Ha kilépünk az SQL*Plus környezetből (az EXIT, vagy a QUIT utasítással), majd újra belépünk, akkor már észlelhetjük a login szkriptünk hatását, de a benne szereplő SQL lekérdező utasítás révén a közvetlen futtatás is elég látványos. Adjuk ki a

```
SQL> @login
```

utasítást. Ezzel lefuttatjuk a login szkriptet, és megjelenik az emp tábla listája.

Általános szkript program létrehozása és futtatása

Megjegyezzük, hogy általában "nem illik" egy telepített program rendszerkönyvtáraiba bármit is írni (bár azért a login szkript kivétel), különösen nem gyakorlási célokat szolgáló minta programokat. Természetesen tetszőleges útvonal is megadható mind az ed szerkesztő, mind az @ futtató parancshoz. Tehát kiadható például az

```
SQL> ed c:\gyakorlás\abc
```

és az

```
SQL> @c:\gyakorlás\abc
```

utasítás, csak ezeket sokszor leírni nem éppen kellemes, ezért ha mindig csak ugyanazt a néhány (például aa1, aa2, stb.) programfájlnevet használjuk, akkor a bin alkönyvtárba írás tekinthető akár bocsánatos bűnnek is...

A továbbiakban az utasítások elé már nem fogjuk odaírni az SQL*Plus prompt jelét.

Kiíratás szkript programból

Gyakran előfordul, hogy valamilyen értelmező szöveget szeretnénk kiírni egy lista előtt vagy után. Erre szolgál az SQL*Plus környezetben a PROMPT utasítás.

Pontos alakja:

```
PROMPT [szöveg]
```

ahol a megadott szöveg megjelenik a felhasználó képernyőjén, illetve ennek hiánya esetén egy üres sor (azaz egy soremelés).

Megjegyzések használata

Egy programba az áttekinthetőség és a dokumentálás érdekében megjegyzéseket helyezhetünk el, melyeket az Oracle a futtatható kód előállításakor figyelmen kívül hagy. Ezt háromféleképpen is megtehetjük:

A REM TÍPUSÚ MEGJEGYZÉS

Csak SQL*Plus szkriptben használható, PL/SQL nyelvű kódban nem. A sorban nem előzheti meg más, csak szóköz karakter.

```
rem E megjegyzés előtt csak szóköz állhat a sorban
```

A -- TÍPUSÚ MEGJEGYZÉS

Egyaránt használható SQL*Plus szkriptben és PL/SQL nyelvű kódban. A sorban ezt sem előzheti meg más, csak szóköz karakter.

```
-- E megjegyzés előtt is csak szóköz lehet
```

A /* */ TÍPUSÚ MEGJEGYZÉS

Csak PL/SQL nyelvű kódban használható. A sorban előtte és utána is lehet utasítás, és az ilyen megjegyzés több soros is lehet. Gyakran alkalmazzuk programok tesztelésénél, amikor egy (akár több soros) kódrészt szeretnénk átmenetileg kihagyni a futtatásból.

```
utasítás /* Ez előtt és után is állhat utasítás,  
és akár több soros is lehet. */
```

A sorfolytató jel használata

Az SQL*plus utasításai úgynevezett egysoros utasítások, azaz elvben egyetlen sorba el kellene férniük. Mivel a sorok gyakorlatilag tetszőleges hosszúak lehetnek (bár az alkalmazott programszöveg-szerkesztőknek általában van erre felső korlátjuk), ez nem jelent elvi korlátot, ám igen megnehezíti egy hosszabb leírt utasítás áttekintését.

A fenti probléma elkerülése érdekében vezették be a "sorfolytató" jelet, mely egy, a sor végére írt "--" karakter. Az SQL*Plus parancsértelmező programja ekkor megvárja a következő sort is, és csak annak beolvasása után (ha ez már nem tartalmaz további

sorfolytató jelet) értelmezi az utasítást. A sorfolytató jele(ke)t az SQL*Plus parancsértelmezője természetesen figyelmen kívül hagyja.

Ennek értelmében a

```
SET linesize 400
SELECT * FROM emp;
```

és a

```
SET -
linesize -
400
SELECT * FROM emp;
```

utasításpárok egyenértékűek.

Változók használata

A már definiált változókat és azok értékét lekérdezhetjük a

```
DEFINE
```

utasítással. Változókat (karakteres típusúakat) definiálhatunk a

```
DEFINE változó = érték
```

utasítással, és az

```
ACCEPT változó PROMPT szöveg
```

utasítással, mely felhasználói adatbekérésre használható. A definiált változókra hivatkozhatunk az

```
&változó
```

illetve, ha jobbról közvetlenül szöveg követi, akkor

```
&változó.
```

módon (tehát ponttal a végén), és kiírathatjuk a képernyőre a

```
PROMPT &változó
```

utasítással. Egy definiált változót törölhetünk az

```
UNDEFINE változó
```

utasítással. (Mindezekről részletesebben szó esik a 4. fejezetben.)

Az SQL*Plus editorának lecserélése

Mivel az SQL*Plus saját szövegszerkesztője meglehetősen egyszerű, előfordulhat, hogy le szeretnénk cserélni egy általunk jobban kedvelt editorra. Tegyük fel, hogy a kedvenc editorunk a Windows `write.exe` nevű programja. Ekkor ki kell adnunk a

```
DEFINE_EDITOR="write.exe"
```

SQL*Plus utasítást, és az `ed` szerkesztőparancs hatására máris a `write.exe` fog megjeleníteni. Ha azt szeretnénk, hogy a továbbiakban mindig ez jelenjen meg, akkor a fenti utasítást el kell helyeznünk a `login.sql` szkriptbe is (az `oracle\ora92\bin` alkönyvtárba).

Tábla-létrehozás-feltöltés-törlés, dátumformák

Az alábbiakban röviden ismertetjük a táblák létrehozásával kapcsolatos alapvető DDL (Data Definition Language) utasításokat.

1.) Tábla létrehozása, adattípusok

Hozzunk létre egy `minta` nevű adattáblát. Legyen egy `név` adatmezője, mely változó méretű karakteres típusú, egy `szül` adatmezője, mely dátum típusú, és egy `tan_átl` adatmezője, mely egy tizedest tartalmazó numerikus típus:

```
CREATE TABLE minta
(név          VARCHAR2(20),
szül          DATE,
tan_átl       NUMBER(3,1));
```

Eredmény

A tábla létrejött.

Megjegyzés

A fenti `VARCHAR2(20)` definíciós alak egy, legfeljebb 20 karakteres szöveg típusú adat tárolására alkalmas adatmezőt jelöl.

A `NUMBER(3,1)` definíciós alak egy tizedes szám formát jelöl, ahol 3 az összes karakterek száma, 1 pedig a tizedes jegyek száma. Mivel egy karakter szükséges a tizedespont számára, így e módon olyan számok ábrázolhatók, melyek egy egész jegyet és egy tizedest tartalmaznak (például 4.6). Ügyeljünk arra, hogy ha negatív értéket is lehetséges, akkor még egy karakterhelyre szükség van, ekkor a megfelelő alak tehát: `NUMBER(4,1)`.

2.) A tábla felépítésének lekérdezése

```
SET linesize 60
DESC minta
SET linesize 400
```

Eredmény

Név	Üres?	Típus
-----	-----	-----
NÉV		VARCHAR2 (20)
SZÜL		DATE
TAN_ÁTL		NUMBER (3,1)

Megjegyzés

A sorhossz (linesize) átállítására azért volt szükség, mert különben nem láttuk volna az eredménysor végét.

3.) A dátum ábrázolása

Hogyan ábrázolja az Oracle rendszerünk a dátumot? Kérdezzük le a dual segédtábla segítségével:

```
SELECT sysdate
FROM dual;
```

Eredmény

```
SYSDATE
-----
04-DEC-28
```

Tehát <év (kétbetűs formátumban)> – <hónap neve (3 betűs magyar rövidítéssel)> – <a nap száma>. A kétjegyű évszám viszont ma a 20xx-et jelenti. Annak érdekében, hogy az általunk megadott (például múlt századi) dátum biztosan helyes legyen, célszerű az évszázadot is megadni. Egy múlt századi dátum megadása például az '1981-ÁPR-05' alakban történhet.

4.) Tábla feltöltése

Egy tábla feltöltéséhez a tárolandó adatokat a megfelelő formában kell megadni; a karakter adattípust és a dátum adattípust például aposztrófok (') között.

Az imént létrehozott minta nevű adattáblába vigyünk be egy rekordot.

```
INSERT INTO minta
VALUES ('CSÍNOM PALKÓ', '1981-ÁPR-05', 2.9);
```

Eredmény

```
1 sor létrejött.
```

5.) Tábla lekérdezése

Kérdezzük le a minta nevű adattáblánkat.

```
SELECT *
FROM minta;
```

Eredmény

NÉV	SZÜL	TAN_ÁTL
-----	-----	-----
CSÍNOM PALKÓ	81-ÁPR-05	2.9

6.) Rendszer dátum formátumának módosítása

Állítsuk át a rendszer dátumot a szokásos magyar formátumra, és kérdezzük le ismét az adattáblánkat.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY.MON.DD';

SELECT *
FROM minta;
```

Eredmény

A munkamenet módosítva.

NÉV	SZÜL	TAN_ÁTL
CSÍNOM PALKÓ	1981.ÁPR.05	2.9

7.) Tábla törlése

Végül töröljük a minta nevű adattáblánkat.

```
DROP TABLE minta;
```

Eredmény

A tábla eldobva.

8.) Rendszer dátum formátumának visszaállítása

Állítsuk vissza a rendszer dátumot az eredeti (évszázadhiányos) formátumra.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YY-MON-DD';
```

Eredmény

A munkamenet módosítva.