4. FEJEZET

Interaktív környezet

Elméleti összefoglaló

Különböző feladatok megoldásához az SQL*Plus-környezetben változókat hozhatunk létre. E változók segítségével a felhasználó adatokat adhat meg egy SQL- vagy SQL*Plus-utasítás (vagy mint később látni fogjuk akár egy PL/SQL program) számára, és ily módon interaktív szkript programokat készíthet. Ezek az értékek az úgynevezett felhasználói (más néven helyettesítő, vagy makró) változók segítségével eltárolhatók az SQL*Plus-környezetben, majd az onnan való kilépéskor törlődnek. Kezelésük az & és az && jelekkel, valamint az `ACCEPT`, a `DEFINE` és az `UNDEFINE` utasításokkal történik.

Felhasználói (helyettesítő) változók használata

A felhasználói változók hivatkozási jelei (& és &&)

Az SQL*Plus-környezetben felhasználói változókat jelölhetünk és definiálhatunk, és karaktersorozat (`CHAR` típusú) értékkel láthatunk el *&változó* és *&&változó* módon.

Ha korábban e változók még nem kaptak értéket, akkor az &, illetve az && jelek adatbekérést is kezdeményeznek, azonban míg az *&változó* módon bekért érték nem kerül eltárolásra az SQL*Plus-környezetben, addig az *&&változó* módon bekért értékét az SQL*Plus-környezet megőrzi mindaddig, amíg ki nem lépünk az SQL*Plus-környezetből, vagy egy `UNDEFINE` utasítással ki nem töröljük (*lásd* alább). Tehát az && jellel egy felhasználói változó definiálható, melynek értéke csak egy `DEFINE` vagy `ACCEPT` utasítással változtatható meg (*lásd* alább).

A már értékkel ellátott felhasználói változókra bármely SQL- vagy SQL*Plus-utasításban hivatkozhatunk. A hivatkozás eredménye egyszerű behelyettesítés (ezért nevezik helyettesítő változónak is). Tehát, ha egy utasításban egy értékkel rendelkező helyettesítő változóra hivatkozunk, akkor az utasítás előbb felveszi a behelyettesítést követő alakot, és csak azután kerül végrehajtásra. Ennek megfelelően, bár egy & vagy && módon bekért helyettesítő változó belső tárolása karaktersorozat formában (`CHAR` típusként) történik, használatánál ugyanazok a szabályok, mintha a tartalma közvetlenül konstansként lett volna beírva.

Például, ha karakteres konstansként vagy dátumkonstansként szerepel, akkor a hivatkozás módja `'&változó'`, vagyis a hivatkozást aposztrófok (') közé kell tenni (mint például az `&név` hivatkozást a

```
SELECT * FROM emp WHERE ename = '&név';
```

utasításban), míg numerikus konstansként vagy függvénynévként elég közvetlenül szerepelnie *&változó* módon (mint például a

```
SELECT * FROM emp WHERE sal<&minfiz AND hiredate<&mainap;
```

utasításban, feltéve, hogy a `minfiz` változóba egy számot, a `mainap` nevű változóba pedig a `sysdate` karaktersorozatot írtuk).

További alkalmazási szabály, hogy ha a *változó* által tartalmazott értéket közvetlenül követné valamilyen karaktersorozat (*lásd* például alább a `PROMPT` utasításnál), akkor a hivatkozás (*&változó*) és az azt követő karaktersorozat közé egy pont (`.`) karaktert kell tenni.

Az értékkel ellátott (tárolt) felhasználói változóra kiadott *&&változó* módon történő hivatkozás a tárolt értéket használja fel.

`DEFINE` utasítás

Felhasználói változók definiálására szolgál. Az utasítás alakja:

`DEFINE` *változó = érték*	Létrehoz egy `CHAR` típusú felhasználói *változót*, és egy *értéket* rendel hozzá.
`DEFINE` *változó*	Megjeleníti a *változót*, az értékét és az adattípusát.
`DEFINE`	Megjeleníti az összes felhasználói *változót*, értékeivel és adattípusával együtt.

Megjegyzés
Egy `DEFINE` segítségével létrehozott felhasználói változó az értékét mindaddig megőrzi, amíg ki nem lépünk az SQL*Plus-környezetből vagy egy `UNDEFINE` utasítással ki nem töröljük. A *változó* értéke csak egy újabb `DEFINE`, vagy `ACCEPT` utasítással változtatható meg, és arra *&változó* módon hivatkozhatunk.

`ACCEPT` utasítás

Az `ACCEPT` utasítás definiál egy felhasználói változót, küld egy üzenetet a felhasználónak, majd a felhasználó válaszát eltárolja az általa definiált változóban (a felhasználói választ el is tudja rejteni, így alkalmas jelszó bevitelére). Az utasítás alakja:

`ACCEPT` *változó* [*adattípus*] [`FORMAT` *'formátummaszk'*] [`PROMPT` *'üzenet'*] [`HIDE`]

ahol:

- *változó* — a változó neve (ha még nem létezik ez a változó, akkor létrehozza),
- *adattípus* — `NUMBER`: esetleg mínusz előjelet és/vagy tizedespontot tartalmazó szám,
  `CHAR`: tetszőleges karaktersorozat (maximum 240 bájt),
  `DATE`: érvényes dátumot tartalmazó karakterlánc,

- FORMAT *'formátummaszk'* kijelöli a használandó formátummaszkot (*lásd* 1. fejezet: *Dátumok és számok formázott megjelenítése* c. pont),
- PROMPT *'üzenet'* a felhasználó adatmegadása előtt megjeleníti az üzenetet,
- HIDE elrejti a felhasználó által megadott adatot (jelszó).

Megjegyzés

- Az ACCEPT utasítás az SQL*Plus fejlesztésének egy későbbi szakaszában keletkezett, ez már (ellentétben az && vagy a DEFINE segítségével definiált változókkal) nem csupán karakteres típust rendelhet az általa definiált változóhoz. Mivel e változók belső reprezentációja már lehet NUMBER vagy DATE is, ezért egy *&változó* hivatkozás használatánál ezt figyelembe kell venni.
- Egy ACCEPT segítségével létrehozott felhasználói változó az értékét mindaddig megőrzi, amíg ki nem lépünk az SQL*Plus-környezetből, vagy egy UNDEFINE utasítással ki nem töröljük. A *változó* értéke csak egy újabb DEFINE vagy ACCEPT utasítással változtatható meg, és arra *&változó* módon hivatkozhatunk.

PROMPT utasítás

Egy szkript programból való üzenet kiírására a PROMPT utasítás önmagában is kiadható a

```
PROMPT [szöveg]
```

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

Megjegyzés

- Figyeljünk fel arra, hogy ebben az esetben a kiírandó szöveget nem határolja aposztróf (') karakter.
- A PROMPT utasításban egy felhasználói változóra *&változó* módon hivatkozhatunk (*lásd* feljebb *A felhasználói változók hivatkozási jelei* c. pontban leírtakat).
- Ha a szövegben szerepel egy hivatkozás (*&változó*), akkor azt egy pont (.) karakternek kell lezárnia (például a PROMPT Ez a(z) &index.-ik utasítás hatására a 8-as számot tartalmazó index változó esetén a kiírt szöveg: Ez a(z) 8-ik).

UNDEFINE utasítás

Az ACCEPT, a DEFINE vagy az && előtaggal definiált SQL*Plus felhasználói változók mindaddig megőrzik értéküket, míg egy UNDEFINE utasítással nem töröljük őket, vagy ki nem lépünk az SQL*Plus-környezetből. Az utasítás alakja:

```
UNDEFINE változó
```

4.1. példa

Írjon szkript programot, mely a felhasználó által megadott lekérdező utasítást futtatja.

Megoldás

```
-- Futtató.sql szkript program eleje
SET numwidth 6
ACCEPT utasítás PROMPT 'Adjon meg egy lekérdező utasítást: '
SELECT * FROM (&utasítás);
SET numwidth 10
-- Futtató.sql szkript program vége
```

Eredmény

```
Adjon meg egy lekérdező utasítást: select * from emp where deptno=10
régi   1: SELECT * FROM (&utasítás)
új   1: SELECT * FROM (select * from emp where deptno=10)

 EMPNO ENAME      JOB          MGR HIREDATE     SAL   COMM DEPTNO
------ ---------- --------- ------ --------- ------- ------ ------
  7839 KING       PRESIDENT       81-NOV-17   5000             10
  7782 CLARK      MANAGER     7839 81-JÚN-09   2450             10
  7934 MILLER     CLERK       7782 82-JAN-23   1300             10
```

Az SQL*Plus-környezet beállítása

A környezet lekérdezése

Az SQL*Plus-környezet lekérdezése a

```
SHOW ALL
```

utasítással, egy rendszerváltozó lekérdezése pedig a

```
SHOW Rendszerváltozó
```

segítségével történik.

A környezet beállítása

Az SQL*Plus-környezet beállítása a

```
SET Rendszerváltozó Érték
```

utasítással, vagy az Opciók / Környezet menüpontban történhet.

Az SQL*Plus-utasítások lekérdezése (HELP)

Az SQL*Plus környezetbeállító utasításainak lekérdezése a

HELP *SQL*Plus_utasítás*

utasítással történhet.

4.2. példa

Az alábbiakban bemutatjuk az SQL*Plus környezetbeállító utasításainak használatát.

1. A HELP használata

```
SQL> HELP

 HELP
 ----

 Accesses this command line help system. Enter HELP INDEX for a list
 of topics.
 In iSQL*Plus, click the Help button to display iSQL*Plus help.

 HELP [topic]
```

2. Az SQL*Plus-utasítások lekérdezése

```
SQL> HELP INDEX

Enter Help [topic] for help.

 @             COPY          PAUSE                    SHUTDOWN
 @@            DEFINE        PRINT                    SPOOL
 /             DEL           PROMPT                   SQLPLUS
 ACCEPT        DESCRIBE      QUIT                     START
 APPEND        DISCONNECT    RECOVER                  STARTUP
 ARCHIVE LOG   EDIT          REMARK                   STORE
 ATTRIBUTE     EXECUTE       REPFOOTER                TIMING
 BREAK         EXIT          REPHEADER                TTITLE
 BTITLE        GET           RESERVED WORDS (SQL)     UNDEFINE
 CHANGE        HELP          RESERVED WORDS (PL/SQL)  VARIABLE
 CLEAR         HOST          RUN                      WHENEVER OSERROR
 COLUMN        INPUT         SAVE                     WHENEVER SQLERROR
 COMPUTE       LIST          SET
 CONNECT       PASSWORD      SHOW
```

3. A SET utasítások lekérdezése

```
SQL> HELP SET

SET
---

Sets a system variable to alter the SQL*Plus environment settings
for your current session, for example:
    -   display width for data
    -   turn on HTML formatting
    -   enabling or disabling printing of column headings
    -   number of lines per page

In iSQL*Plus, you can also use the System Variables screen to set
system variables.

SET system_variable value

where system_variable and value represent one of the following clauses:

  APPI[NFO]{OFF|ON|text}                  *NEWP[AGE] {1|n|NONE}
  ARRAY[SIZE] {15|n}                       NULL text
  AUTO[COMMIT] {OFF|ON|IMM[EDIATE]|n}      NUMF[ORMAT] format
  AUTOP[RINT] {OFF|ON}                     NUM[WIDTH] {10|n}
  AUTORECOVERY {ON|OFF}                    PAGES[IZE] {24|n}
  AUTOT[RACE] {OFF|ON|TRACE[ONLY]}        *PAU[SE] {OFF|ON|text}
    [EXP[LAIN]] [STAT[ISTICS]]             RECSEP {WR[APPED] |
  BLO[CKTERMINATOR] {.|c}                    EA[CH]|OFF}
  CMDS[EP] {;|c|OFF|ON}                    RECSEPCHAR {_|c}
  COLSEP {_|text}                          SERVEROUT[PUT] {OFF|ON}
  COM[PATIBILITY] {V7|V8|NATIVE}             [SIZE n] [FOR[MAT]
  CON[CAT] {.|c|OFF|ON}                      {WRA[PPED] |
  COPYC[OMMIT] {0|n}                         WOR[D_WRAPPED] |
  COPYTYPECHECK {OFF|ON}                     TRU[NCATED]}]
  DEF[INE] {&|c|OFF|ON}                   *SHIFT[INOUT] {VIS[IBLE] |
  DESCRIBE [DEPTH {1|n|ALL}]                 INV[ISIBLE]}
    [LINENUM {ON|OFF}] [INDENT {ON|OFF}]  *SHOW[MODE] {OFF|ON}
  ECHO {OFF|ON}                           *SQLBL[ANKLINES] {ON|OFF}
 *EDITF[ILE] file_name[.ext]                SQLC[ASE] {MIX[ED] |
  EMB[EDDED] {OFF|ON}                        LO[WER] | UP[PER]}
  ESC[APE] {\|c|OFF|ON}                   *SQLCO[NTINUE] {> | text}
  FEED[BACK] {6|n|OFF|ON}                 *SQLN[UMBER] {OFF|ON}
  FLAGGER {OFF|ENTRY|INTERMED[IATE]|FULL}  SQLPLUSCOMPAT[IBILITY] {x.y[.z]}
 *FLU[SH] {OFF|ON}                        *SQLPRE[FIX] {#|c}
```

```
HEA[DING] {OFF|ON}                          *SQLP[ROMPT] {SQL>|text}
HEADS[EP] {|c|OFF|ON}                        SQLT[ERMINATOR]
INSTANCE [instance_path|LOCAL]                 {;|c|OFF|ON}
LIN[ESIZE] {80|n} ({150|n} iSQL*Plus)       *SUF[FIX] {SQL|text}
LOBOF[FSET] {n|1}                            TAB {OFF|ON}
LOGSOURCE [pathname]                         TERM[OUT] {OFF|ON}
LONG {80|n}                                  TI[ME] {OFF|ON}
LONGC[HUNKSIZE] {80|n}                       TIMI[NG] {OFF|ON}
MARK[UP] HTML [ON|OFF]                       TRIM[OUT] {OFF|ON}
  [HEAD text] [BODY text] [TABLE text]       TRIMS[POOL] {ON|OFF}
  [ENTMAP {ON|OFF}]                          UND[ERLINE] {-|c|ON|OFF}
  [SPOOL {ON|OFF}]                           VER[IFY] {OFF|ON}
  [PRE[FORMAT] {ON|OFF}]                     WRA[P] {OFF|ON}
```

An asterisk (*) indicates the SET option is not supported in iSQL*Plus.

4. A numerikus mezőszélesség beállítása és hatása

```
SQL> SHOW numwidth
numwidth 10

SQL> SELECT empno, sal
  2    FROM emp;

     EMPNO        SAL
---------- ----------
      7369        800
      7499       1600
...
14 sor kijelölve.
```

Állítsuk át a kiírandó számjegyek számát 5-re

```
SQL> SET numwidth 5
SQL> SELECT empno, sal
  2    FROM emp;
SQL> SET numwidth 10

EMPNO   SAL
----- -----
 7369   800
 7499  1600
 ...
 7934  1300

14 sor kijelölve.
```

Megjegyezzük, hogy alapértelmezésben, ha a lista hat, vagy annál több sorból áll, akkor kiírja a listázott sorok számát. Ez a `feedback` rendszerváltozóval módosítható; ki- és bekapcsolható (`SET feedback {6|n|OFF|ON}` – *lásd* feljebb).

Az SQL*Plus formázási utasításai

Az SQL*Plus-környezet lehetőséget ad arra, hogy az eredménylisták megjelenését beállítsuk. A formázási utasítások egy része az eredménylista egészére vonatkozik (fejléc, lábléc stb.), a másik része az egyes (karakteres, numerikus és dátum típusú) oszlopok szélességét és megjelenését állítja be. Lehetőség van a formázási utasítások letiltására, újra engedélyezésére és törlésére is.

LISTA SZINTŰ FORMÁZÁSOK

`TTITLE` [*"szöveg"* | `OFF` | `ON`]	Minden oldal tetején megjelenő fejléc megadása, letiltása, engedélyezése.
`BTITLE` [*"szöveg"* | `OFF` | `ON`]	Minden oldal alján megjelentő lábléc megadása, letiltása, engedélyezése.
`BREAK [ON` {*oszlopnév* | *oszlopkifejezés*} [`ON` {*oszlopnév* | *oszlopkifejezés*}]...]	Kiszűri az ismétlődő értékek megjelenítését az egymást követő sorokban, és sortöréssel tagolja az adatsorokat.

OSZLOP SZINTŰ FORMÁZÁSOK

A `COLUMN` utasítás az oszlopok és az oszlopfejlécek megjelenési formáját szabályozza a paramétereknek megfelelően. A paraméterek megadhatók egymás után vesszővel elválasztva, vagy külön `COLUMN` utasításokkal is. A `COLUMN` utasítás szintaktikája:

```
COLUMN {oszlopnév | oszlopkifejezés}
       [FORMAT 'formátummaszk']
       [HEADING "szöveg"]
       [JUSTIFY {LEFT | CENTER | RIGHT} ]
```

illetve

```
COLUMN oszlopnév {PRINT | NOPRINT | OFF | ON | CLEAR}
```

ahol:

`FORMAT` *'formátummaszk'*	Megadja az oszlopadatok megjelenítési formáját. Karakteres adatok esetén az *Aszám* formátum (pl. `A15`) azt jelenti, hogy az adat számára hány karakternyi helyet engedélyezünk, numerikus és dátum típusú adatok esetén a korábban ismertetett formátummaszk elemek használhatók (*lásd* 1. fejezet: *A dátumok és számok formázott megjelenítése* c. rész).

`HEADING "szöveg"`	Egy oszlop fejlécének beállítására szolgál. Alapértelmezés az oszlopnév. Ha ezen változtatni akarunk, meg kell adni a szöveget. Ha e szövegben üres vagy speciális karakter van, aposztrófok közé kell tenni. Ha a szöveg tördelőkaraktert (|) tartalmaz, akkor annak helyén a szöveget tördelve jeleníti meg.
`JUSTIFY {LEFT | CENTER | RIGHT}`	Az *oszlopfejléc* igazítása. Alapértelmezés: számoknál jobbra igazítás, minden más típusnál balra igazítás.
`NOPRINT`, illetve `PRINT`	Letiltja (elrejti), illetve engedélyezi az oszlop megjelenítését.
`OFF`, illetve `ON`	Kikapcsolja, illetve bekapcsolja az oszlophoz rendelt formázást.
`CLEAR`	A megadott oszlop formátumának törlése.

Megjegyzés

- Ügyeljünk arra, hogy a formázó utasításokat mindig egyetlen sorba írjuk, illetve, ha több sorba írjuk, akkor az egyes sorok végére *sorfolytató jelet* (-) tegyünk (ugyanis ezek SQL*Plus-utasítások, melyeknél ez követelmény – *lásd* 3. melléklet).
- A formázott listázások után állítsuk vissza a környezeti beállításokat, mert különben a későbbi listázásoknál is megjelenhetnek a beállított formátumok.
- Sajnálatos módon a `COLUMN` utasítás az Unicode karakterkészlet esetén hibásan működik még a 9.2. verzióban is! (Telepítéskor ezért az úgynevezett `default` karakterkészletet célszerű választani. *Lásd* 1. melléklet.)

FORMÁZÁSOK TÖRLÉSE

```
CLEAR {BREAKS | COLUMNS | SCREEN}
```

ahol:

- a `CLEAR BREAKS` utasítással törölhető a `BREAK` utasítás hatása,
- a `CLEAR COLUMNS` utasítással az összes oszlopformázás törölhető,
- a `CLEAR SCREEN` utasítással törölhető az SQL*Plus-környezet képernyője.

4.3. példa

Állítsuk át a fizetés értéket a helyi pénznemre. Listázzuk ki a 30-as részlegben dolgozók nevét, fizetését és jutalékát.

Megoldás

```
SET feedback OFF
COLUMN sal FORMAT L99999
SELECT ename, sal, comm  FROM emp  WHERE deptno=30;

COLUMN sal OFF
SELECT ename, sal, comm  FROM emp  WHERE deptno=30;
COLUMN sal ON
```

```
COLUMN sal NOPRINT
SELECT ename, sal, comm  FROM emp  WHERE deptno=30;
COLUMN sal PRINT

COLUMN sal CLEAR
SET feedback ON
```

Eredmény

```
ENAME                    SAL       COMM
---------- ---------------- ----------
BLAKE                 Ft2850
MARTIN                Ft1250       1400
ALLEN                 Ft1600        300
TURNER                Ft1500          0
JAMES                  Ft950
WARD                  Ft1250        500

ENAME             SAL       COMM
---------- ---------- ----------
BLAKE             2850
MARTIN            1250       1400
ALLEN             1600        300
TURNER            1500          0
JAMES              950
WARD              1250        500

ENAME             COMM
---------- ----------
BLAKE
MARTIN            1400
ALLEN              300
TURNER               0
JAMES
WARD               500
```

4.4. példa

Legyen az ename listázandó oszlop felirata "A dolgozók neve", de a neve már új sorban legyen (sortörés). Igazítsuk középre.

Megoldás

```
COLUMN ename HEADING "A dolgozók|neve" JUSTIFY CENTER PRINT
SET numwidth 5
SELECT * FROM emp ORDER BY ename;
SET numwidth 10
```

Eredmény

```
        A dolgozók
EMPNO   neve    JOB         MGR HIREDATE    SAL  COMM DEPTNO
----- ---------- --------- ----- --------- ----- ----- ------
 7876 ADAMS      CLERK      7788 83-JAN-12  1100            20
 7499 ALLEN      SALESMAN   7698 81-FEB-20  1600   300      30
 .... .....      .......    .... .........  ....   ...      ..
```

Most igazítsuk jobbra:

```
COLUMN ename HEADING "A dolgozók|neve" JUSTIFY RIGHT
SET numwidth 5
SELECT * FROM emp ORDER BY ename;
SET numwidth 10
```

Eredmény

```
        A dolgozók
EMPNO       neve JOB         MGR HIREDATE    SAL  COMM DEPTNO
----- ---------- --------- ----- --------- ----- ----- ------
 7876 ADAMS      CLERK      7788 83-JAN-12  1100            20
 7499 ALLEN      SALESMAN   7698 81-FEB-20  1600   300      30
 .... .....      .......    .... .........  ....   ...      ..
```

Tiltsuk le az `empno` **és** `hiredate` **oszlop kiírását.**

```
COLUMN empno NOPRINT
COLUMN hiredate NOPRINT
SET numwidth 5
SELECT * FROM emp ORDER BY ename;
SET numwidth 10
```

Eredmény

```
A dolgozók
      neve JOB         MGR   SAL  COMM DEPTNO
---------- --------- ----- ----- ----- ------
ADAMS      CLERK      7788  1100            20
ALLEN      SALESMAN   7698  1600   300      30
.....      .......    ....  ....   ...      ..
```

Megjegyzés

Sose feledkezzünk meg a formázási és rendszerbeállítások visszaállításáról:

```
COLUMN empno OFF
COLUMN hiredate CLEAR
CLEAR BREAKS
TTITLE OFF
```

```
BTITLE OFF
COLUMN ename CLEAR
```

vagy

```
CLEAR COLUMNS
```

4.5. példa

Listázza ki az emp táblából a felhasználó által megadott foglalkozású dolgozókat.

Megoldás

```
SET linesize 50
BREAK ON job
TTITLE "Lista a foglalkozás szerinti| dolgozók fizetéséről"
COLUMN ename FORMAT A15 HEADING "A dolgozó|neve" JUSTIFY LEFT
COLUMN sal FORMAT 999999999999 HEADING "A dolgozó|havi jövedelme neve"
JUSTIFY CENTER

SELECT job,
       ename,
       sal
  FROM emp
  WHERE UPPER(job) = UPPER('&foglalkozás');
```

Eredmény

```
Adja meg a(z) foglalkozás értékét: clerk
régi   5:   WHERE UPPER(job) = UPPER('&foglalkozás')
új   5:   WHERE UPPER(job) = UPPER('clerk')

K.  Okt 05                                        lap    1
           Lista a foglalkozás szerinti
              dolgozók fizetéséről

           A dolgozó            A dolgozó
JOB        neve            havi jövedelme neve
---------  --------------- ------------------
CLERK      SMITH                          800
           ADAMS                         1100
           JAMES                          950
           MILLER                        1300
```

Ne felejtsük el visszaállítani a beállításokat:

```
SET linesize 400
CLEAR BREAKS
TTITLE OFF
```

```
COLUMN ename OFF
COLUMN sal OFF
CLEAR COLUMNS
```

4.6. példa

Listázza ki az emp táblából a felhasználó által megadott foglalkozású dolgozók főnökeinek nevét.

Megoldás

```
SET verify OFF

SELECT dolgozó.ename  AS dolgozóNév,
       dolgozó.job    AS dolgozóFoglalkozás,
       dolgozó.sal    AS dolgozóFizetés,
       főnök.ename    AS főnökNév
  FROM emp  főnök,
       (SELECT *
          FROM emp
          WHERE UPPER(job) = UPPER('&foglalkozás'))  dolgozó
  WHERE dolgozó.mgr = főnök.empno;

SET verify ON
```

Eredmény

```
Adja meg a(z) foglalkozás értékét: clerk

DOLGOZÓNÉV DOLGOZÓFO DOLGOZÓFIZETÉS FŐNÖKNÉV
---------- --------- -------------- ----------
SMITH      CLERK                800 FORD
ADAMS      CLERK               1100 SCOTT
JAMES      CLERK                950 BLAKE
MILLER     CLERK               1300 CLARK
```

4.7. példa

Listázza ki a felhasználó által megadott főnök beosztottjainak hierarchiaszintjét, azonosítóját, nevét, a főnökének kódját és részlegét.

Megoldás

```
SELECT LEVEL    AS "Beosztási szint",
       empno    AS azonosító,
       ename    AS neve,
       mgr      AS főnök_kódja,
       deptno   AS részleg
```

```
FROM emp
CONNECT BY mgr = PRIOR empno
START WITH UPPER(ename) = UPPER('&név')
ORDER BY deptno;
```

Eredmény

```
Adja meg a(z) név értékét: Jones

Beosztási szint  AZONOSÍTÓ NEVE       FŐNÖK_KÓDJA    RÉSZLEG
--------------- ---------- ---------- ----------- ----------
              1       7566 JONES             7839         20
              2       7788 SCOTT             7566         20
              3       7876 ADAMS             7788         20
              2       7902 FORD              7566         20
              3       7369 SMITH             7902         20
```

4.8. példa

Listázza ki a felhasználó által megadott oszlop szerinti, valamint a felhasználó által megadott belépési időintervallumba eső dolgozók átlagfizetését.

Megoldás

```
ACCEPT oszlop PROMPT 'Az oszlop neve: '
ACCEPT dátum1 PROMPT 'Kezdő dátum (éééé.hh.nn): '
ACCEPT dátum2 PROMPT 'Záró dátum (éééé.hh.nn):  '
SELECT &oszlop,
       ROUND(AVG(sal))  AS "Átlagfizetés"
  FROM emp
  WHERE hiredate BETWEEN TO_DATE('&dátum1','YYYY.MM.DD')
                     AND TO_DATE('&dátum2','YYYY.MM.DD')
  GROUP BY &oszlop;
```

Eredmény

```
Az oszlop neve: deptno
Kezdő dátum (éééé.hh.nn): 1980.01.01
Záró dátum (éééé.hh.nn):  1990.01.01
régi   1: SELECT &oszlop,
új   1: SELECT deptno,
régi   4:   WHERE hiredate BETWEEN TO_DATE('&dátum1','YYYY.MM.DD')
új   4:   WHERE hiredate BETWEEN TO_DATE('1980.01.01','YYYY.MM.DD')
régi   5:                      AND TO_DATE('&dátum2','YYYY.MM.DD')
új   5:                    AND TO_DATE('1990.01.01','YYYY.MM.DD')
régi   6:   GROUP BY &oszlop
új   6:   GROUP BY deptno
```

```
    DEPTNO Átlagfizetés
---------- ------------
        10         2917
        20         2175
        30         1567
```

Szkript programok felhasználóvezérelt hívása

Bár az SQL*Plus környezetének szkript programjai nem összetett feladatok megoldására készültek (erre a magasszintű programozási nyelvek szolgálnak; PL/SQL, Delphi, Java, C# stb.), azért van lehetőség különböző hívási szerkezetek létrehozására. Az alábbi példában ezt mutatjuk be. Ennek során fel fogjuk használni a jelen fejezetben bemutatott felhasználói változókat és az interaktivitást lehetővé tevő `ACCEPT` utasítást.

4.9. példa

Írjon egy `aa.sql` nevű szkript programot, mely felkínálja a felhasználónak a választást a *b* és a *c* módszer között (előbbit az `ab.sql`, utóbbit az `ac.sql` szkript program „futtatja"), majd a választól függően meghívja (a vele azonos könyvtárban lévő) megfelelő szkript programot. Ha a felhasználó válasza hibás, akkor írja ki, hogy „A válasz hibás!", és kínálja fel újra a választást. Helyes válaszok esetén az egyes meghívott szkript programok írják ki a nevüket, majd ezekből visszatérve az `aa.sql` búcsúzzon el a felhasználótól, és lépjen ki az SQL*Plus-környezetbe.

Megoldás
(A példabeli feladatot az alábbi öt szkript program segítségével oldottuk meg. Mindegyiket a megadott nevű fájlba kell tenni, és azzal a fájlnévvel meghívni!)

```
-- Az aa.sql szkript program eleje
PROMPT Ez az aa.sql szkript program:
DEFINE OK = 'NEM'
ACCEPT válasz PROMPT 'Melyik módszert választja (b, c): '
PROMPT
@@a&válasz
@@a&OK
-- Az aa.sql szkript program vége

-- Az ab.sql szkript program eleje
PROMPT Ez az ab.sql szkript program:
DEFINE OK = 'IGEN'
PROMPT Most az a&válasz..sql szkript program fut (ab)
-- további utasítások
PROMPT
-- Az ab.sql szkript program vége
```

```
-- Az ac.sql szkript program eleje
PROMPT Ez az ac.sql szkript program:
DEFINE OK = 'IGEN'
PROMPT Most az a&válasz..sql szkript program fut (ac)
-- további utasítások
PROMPT
-- Az ac.sql szkript program vége

-- Az aNEM.sql szkript program eleje
PROMPT
PROMPT Ez az OK = 'NEM' válaszhoz tartozó aNEM.sql szkript program:
PROMPT A(z) &válasz válasz hibás!
PROMPT
@@aa
-- Az aNEM.sql szkript program vége

-- Az aIGEN.sql szkript program eleje
PROMPT Ez az OK = 'IGEN' válaszhoz tartozó aIGEN.sql szkript program:
PROMPT Viszontlátásra!
-- Az aIGEN.sql szkript program vége
```

Eredmény

```
SQL> @c:\próba\aa
Ez az aa.sql szkript program:
Melyik programot választja (b, c): x

SP2-0310: nem lehet megnyitni a(z) "ax.sql" fájlt

Ez az aNEM.sql szkript program:
A(z) x válasz hibás!

Ez az aa.sql szkript program:
Melyik programot választja (b, c): b

Ez az ab.sql szkript program:
Most az ab.sql szkript program fut (ab)

Ez az aIGEN.sql szkript program:
Viszontlátásra!
```