



Instrukcja do zajęć laboratoryjnych

Bazy danych, ORACLE

wersja 3.0

Nr ćwiczenia:	2
Temat:	Program SQL*Plus
Cel ćwiczenia:	Celem ćwiczenia jest zapoznanie studenta z zasadą działania oraz obsługą programu SQL*Plus.
Uwagi:	Bardzo pomocna będzie prawdopodobnie oryginalna dokumentacja firmy ORACLE.

Uwaga: warunkiem uzyskania w ćwiczeniu takich jak poniżej wyników jest praca na schemacie `SUMMIT2` z niezmienionymi danymi. Dlatego też należy upewnić się, że skrypt `summit2.sql` wykonał się bezbłędnie.

1. Uwagi wstępne

SQL*Plus jest prostym programem obsługiwanym z linii poleceń i dołączanym standardowo do każdej instalacji ORACLE na każdej platformie sprzętowej i systemowej. Umożliwia on łączenie się z bazą danych ORACLE, wydawanie poleceń w językach SQL oraz PL/SQL oraz wykonywanie skryptów z zapisanymi poleceniami w tychże językach.

Program, jak na obecne czasy, jest dość ubogi w przyjazne mechanizmy interakcji z użytkownikiem, jednak naprawdę warto się go nauczyć. Choćby tylko dlatego, że gdy nie posiadamy w danym momencie żadnego innego programu do komunikowania się z bazą ORACLE, ten program może nam taką komunikację zapewnić. Jest niemal pewne, że program ten został wcześniej zainstalowany przez administratora w trakcie instalowania systemu ORACLE.

Bardzo istotne jest również i to, że praktycznie wszystkie skrypty instalacyjne (firmy ORACLE oraz innych niezależnych producentów oprogramowania) dostarczane są jako skrypty programu SQL*Plus. Często więc bez znajomości tego programu nie poradzimy sobie z „przebrnięciem” przez proces instalacji wymaganego oprogramowania.

2. Uruchamianie i zakańczanie programu SQL*Plus

Uruchamia się go wydając polecenie `sqlplus`. Po uruchomieniu programu należy podać nazwę użytkownika oraz hasło. Po pomyślnym przebiegu autoryzacji użytkownika na ekranie powinniśmy otrzymać ekran podobny do poniższego (w twojej instalacji pewne napisy mogą wyświetlać się w języku innym niż angielski – np. polskim – zależy to od tzw. *ustawień regionalnych* systemu ORACLE). Interfejs w języku angielskim:

```
D:\oracle\ora11g\BIN>set NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
D:\oracle\ora11g\BIN>sqlplus
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Sun Mar 18 21:07:03 2012

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: summit2
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
```

Po ustawieniu zmiennej systemowej `NLS_LANG` odpowiedniej dla języka polskiego SQL*Plus komunikuje się z nami w ojczystym języku:

```
D:\oracle\ora11g\BIN>set NLS_LANG=POLISH_POLAND.EE8PC852

D:\oracle\ora11g\BIN>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on N Mar 18 21:08:10 2012

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Proszę podać nazwę użytkownika: summit2
Proszę podać hasło:

Połączono z:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
```

Aby podać nazwę użytkownika i jego hasło wprost w wywołaniu programu SQL*Plus piszemy: `sqlplus <nazwa_uzytkownika>/<haslo>`, czyli na przykład:

```
D:\oracle\ora11g\BIN>sqlplus summit2/summit2

SQL*Plus: Release 11.2.0.1.0 Production on Sun Mar 18 21:09:28 2012

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL>
```

Możemy też uruchomić program w trybie niewymagającym od razu podawania nazwy użytkownika oraz hasła. Wówczas, po uruchomieniu programu, należy wydać polecenie `CONNECT`:

```
D:\oracle\ora11g\BIN>sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Sun Mar 18 21:10:06 2012

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect summit2
Enter password:
Connected.
SQL>
```

Inne opcje uruchamiania programu można znaleźć w dokumentacji. Krótkie podsumowanie otrzymamy podając parametr „-” (znak minus, poniżej pokazano tylko fragment zwracanego tekstu):

```
D:\oracle\orallg\BIN>sqlplus -
SQL*Plus: Release 11.2.0.1.0 Production
Copyright (c) 1982, 2010, Oracle. All rights reserved.
Use SQL*Plus to execute SQL, PL/SQL and SQL*Plus statements.
Usage 1: sqlplus -H | -V
    -H          Displays the SQL*Plus version and the
                usage help.
    -V          Displays the SQL*Plus version.
Usage 2: sqlplus [ [<option>] [{logon | /nolog}] [<start>] ]
    <option> is: [-C <version>] [-L] [-M "<options>"] [-R <level>] [-S]
    -C <version> Sets the compatibility of affected commands to the
                version specified by <version>. The version has
                the form "x.y[.z]". For example, -C 10.2.0
    -L          Attempts to log on just once, instead of
                reprompting on error.
...
Refer to the SQL*Plus User's Guide and Reference for more information.
D:\oracle\orallg\BIN>
```

W celu opuszczenia programu SQL*Plus wykonujemy polecenie:

```
SQL> EXIT
```

lub

```
SQL> QUIT
```

3. Logowanie, wylogowywanie, zmiana hasła

W trakcie pracy z programem SQL*Plus możemy w dowolnym momencie zalogować się na konto innego użytkownika. Nie musimy przy tym wylogowywać się z konta poprzednio zalogowanego użytkownika – robi to za nas automatycznie program SQL*Plus. Wydajemy wówczas polecenie:

```
SQL> CONNECT <nazwa_uzytkownika>/<haslo>
```

Jeżeli chcemy natomiast wylogować się bez ponownego logowania, wydajemy polecenie:

```
SQL> DISCONNECT
```

Ten sam skutek osiągniemy opuszczając po prostu program SQL*Plus.

Wydając polecenie:

```
SQL> PASSWORD [nazwa_uzytkownika]
```

mamy możliwość zmiany hasła bieżącego użytkownika (wydajemy polecenie `PASSWORD` bez żadnego parametru) lub też innego użytkownika (podajemy opcjonalny parametr `nazwa_uzytkownika`). Zmiana hasła innego użytkownika niż bieżący jest możliwa tylko wówczas,

jeżeli jesteśmy zalogowani na konto administratora lub też administrator nadał nam uprawnienia pozwalające zmieniać hasła innym użytkownikom. Zaletą tego polecenia jest to, że na ekranie nie pojawia się treść zmienianego hasła (bezpieczeństwo!).

Przykładowo zmieniając swoje hasło (w tym przypadku użytkownik nazywa się `SUMMIT2`) na ekranie możemy otrzymać taki tekst:

```
SQL> PASSWORD
Changing password for SUMMIT2
Old password:
New password:
Retype new password:
Password changed
SQL>
```

Uwagi:

Poszczególne polecenia można skracać. Przykładowo zamiast `PASSWORD` możemy napisać `PASS` a zamiast `CONNECT` `CONN`. Pełną listę dopuszczalnych skrótów znajdziemy w dokumentacji. Ponadto polecenia nie są czułe na wielkość liter.

4. Zapisywanie skryptów w celu ponownego użycia

Aby uniknąć wielokrotnego wpisywania tych samych poleceń dobrze jest zapisywać je w plikach (nazywamy je często *skryptami sql-owymi*). Można więc wydać polecenie:

```
SQL> EDIT nazwa_pliku
```

Efektom wykonania tego polecenia jest uruchomienie edytora tekstu (w systemach *Windows* będzie to zwykle program *NOTEPAD* a w systemie *UNIX* program *vi*). W edytorze tym można wpisać tekst programu PL/SQL lub treść polecenia SQL.

Po zakończeniu pracy należy zapisać plik i opuścić edytor tekstowy. Nastąpi automatyczny powrót do poziomu programu SQL*Plus, gdzie można następnie wydać polecenie:

```
SQL> @nazwa_pliku
```

lub

```
SQL> START nazwa_pliku
```

które spowoduje wykonanie wszystkich poleceń zapisanych w pliku. Trzeba zatem wpisać znak @ („małpa”) lub polecenie `START` – oba sposoby uruchamiania skryptów są równoważne i możemy stosować je zamiennie – a następnie nazwę pliku i wcisnąć *Enter*. Domyślnie przyjmowanym rozszerzeniem nazwy pliku jest `.sql` i dlatego można je pominąć.

W systemach *Windows* po uruchomieniu pliku za pomocą polecenia @ („małpa”) lub `START` czasami pojawia się komunikat:

```
Dane wejściowe obcięte do n znaków.
```

lub w wersji angielskiej:

```
Input truncated to n characters.
```

Można bezpiecznie go zignorować, gdyż w rzeczywistości plik wejściowy nie został wcale obcięty. Komunikat ten pojawia się wtedy, gdy ostatnią linią programu nie jest linia pusta. Aby pozbyć się tego komunikatu należy otworzyć plik i na końcu wstawić pustą linię.

5. Praca z programem SQL*Plus

5.1. Podstawy

Wiele poleceń SQL*Plus posiada formy skrócone, a jedynym sposobem ich poznania jest zapoznanie się z dokumentacją. Przykładowo w poprzednim punkcie zamiast pełnych nazw poleceń `CONNECT` i `DISCONNECT` można użyć ich skróconych wersji `CONN` i `DISC`. Dodatkowo wiele poleceń ma parametry opcjonalne, z których zwykle jeden jest przyjmowany domyślnie. I tak w przypadku polecenia `EXIT` można podać opcjonalny parametr `COMMIT` lub `ROLLBACK` (gdy nie podamy żadnego parametru, pierwszy z nich jest przyjmowany jako domyślny). Szczegóły można znaleźć w dokumentacji.

Programu SQL*Plus możemy używać z powodzeniem do wydawania wszystkich SQL-owych poleceń kierowanych do bazy ORACLE. Przykładowo wpisując następujące polecenie otrzymamy na ekranie:

```
SQL> SELECT * FROM emp WHERE title='President';

          ID LAST_NAME                FIRST_NAME                USERID
-----
START_DATE
-----
COMMENTS
-----
MANAGER_ID TITLE                DEPT_ID                SALARY COMMISSION_PCT
-----
          1 Velasquez                Carmen                cvelasqu
03-MAR-90

          President                50                2500

SQL>
```

Powyższy wynik trudno jest uznać za czytelny. Wiersze zostały po prostu pozawijane, co skutecznie utrudnia czytanie. Program SQL*Plus domyślnie ustawia szerokość wiersza na 80 znaków a długość strony na 12 linii. Aby zmienić wartości tych parametrów powinniśmy wydać polecenia:

```
SQL> SET LINESIZE 500
SQL> SET PAGESIZE 100
```

Ponieważ powyższe polecenia są poleceniami programu SQL*Plus a nie poleceniami języka SQL nie ma potrzeby zakańczać ich średnikami (choć, gdy postawimy średniki, to nic złego się nie stanie).

Wartości liczbowe mogą być oczywiście inne – parametr `LINESIZE`: od 1 do 32767 a parametr `PAGESIZE`: od 1 do 50000. Podane wartości graniczne dla parametru `LINESIZE` dotyczą programu SQL*Plus pracującego pod systemem *Windows*. Dla innych systemów operacyjnych mogą być one nieco inne. Po wydaniu powyższych poleceń wydaje się, że wynik jest jeszcze mniej czytelny:

```
SQL> SELECT * FROM emp WHERE title='President';

          ID LAST_NAME                FIRST_NAME                USERID  START_DATE
T COMMENTS
-----
          1 Velasquez                Carmen                cvelasqu
03-MAR-90

          President                50                2500

          MANAGER_ID TITLE                DEPT_ID                SALARY COMMISSION_PCT
-----
```

```
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
0          1 Velasquez          Carmen          cvelasqu 03-MAR-9
          President          50          2500
SQL>
```

Tym razem oczywiście „winnym” jest okno konsoli tekstowej, gdzie trzeba zmienić rozmiar bufora ekranu (na belce okna klikamy prawym klawiszem myszy a następnie wybieramy z menu kontekstowego pozycję *Właściwości*. W zakładce *Układ* dokonujemy stosownych zmian). Po zmianie szerokości na wartość 500 (lub inną, byle nie za małą) otrzymamy następujący wynik:

```
SQL> SELECT * FROM emp WHERE title='President';

  ID LAST_NAME          FIRST_NAME          USERID  START_DA  COMMENTS
-----
  1 Velasquez          Carmen              cvelasqu 90/03/03
```

W programie SQL*Plus jest dostępnych ponadto wiele innych poleceń i zmiennych. Ich pełen opis można znaleźć w oryginalnej dokumentacji firmy ORACLE jak również w innych opracowaniach na temat bazy danych ORACLE – patrz spis literatury. W przykładzie powyżej użyto polecenia SET do ustawienia wartości zmiennych LINESIZE oraz PAGESIZE.

Często popełnianym błędem przez początkujących użytkowników programu SQL*Plus jest zapominanie o średniku kończącym polecenie SQL (lub blok programowy w języku PL/SQL). Otrzymujemy wówczas na ekranie coś takiego:

```
SQL> SELECT * FROM emp WHERE title='President '
2
```

Jeżeli teraz podamy znak ; (średnik) lub / (ukośnik) to nastąpi wykonanie polecenia. Jeżeli chcemy zrezygnować z wykonywania polecenia należy nacisnąć znak . (kropka).

Wykonanie programu nastąpiło po wpisaniu znaku / (ukośnik), który jest poleceniem programu SQL*Plus i informuje go, aby wykonał zapamiętane w buforze aktualne polecenie SQL (lub tzw. *blok anonimowy* w języku PL/SQL). W celu powtórnego wykonania ostatnio wydawanego polecenia należy ponownie wprowadzić znak ukośnika.

Wydając z kolei komendę:

```
SQL> RUN
```

wykonujemy ostatnio wydawane polecenie. Od polecenia /(ukośnik) różni się jednak tym, że jest również listowane samo wykonywane polecenie, czyli:

```
SQL> /

  ID LAST_NAME          FIRST_NAME          USERID  START_DA
-----
  1 Velasquez          Carmen              cvelasqu 03-MAR-9

SQL>
SQL>
SQL> RUN
 1* SELECT * FROM emp WHERE title='President '
```

ID	LAST_NAME	FIRST_NAME	USERID	START_DA
1	Velasquez	Carmen	cvelasqu	03-MAR-9

SQL>

Aby móc oglądać wyniki działania procedur i funkcji zapisanych w języku PL/SQL musimy jeszcze wydać polecenie:

```
SQL> SET SERVEROUTPUT ON
```

Polecenie to powoduje wyświetlanie na ekranie informacji wyjściowej poleceń DBMS_OUTPUT (np. DBMS_OUTPUT.PUT_LINE – odpowiednik np. polecenia printf w języku C). Opcja ta pozostaje włączona aż do chwili opuszczenia programu SQL*Plus lub wydania polecenia:

```
SQL> SET SERVEROUTPUT OFF
```

Aby sprawdzić działanie tej opcji najlepiej jest napisać najprostszy program w języku PL/SQL (jak poniżej) i samodzielnie wypróbować go:

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('hello, world');
END;
```

Każde polecenie SQL-a lub PL/SQL-a może być wpisywane w jednej linii lub, dla czytelności, może być „łamane” w wielu miejscach. SQL*Plus zwiększa numer linii po każdym wciśnięciu klawisza *Enter*. Przykładowo wprowadzając pokazany wyżej przykładowy program w języku PL/SQL na ekranie widzimy coś takiego (jeżeli na ekranie nie zostanie wyświetlona oczekiwana informacja, należy upewnić się, czy wykonano polecenie SET SERVEROUTPUT ON):

```
SQL> BEGIN
  2   DBMS_OUTPUT.PUT_LINE('hello, world');
  3   END;
  4   /
```

PL/SQL procedure successfully completed.

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> /
hello, world
```

PL/SQL procedure successfully completed.

```
SQL>
```

W linii 4 wpisano znak ukośnika informując tym samym program SQL*Plus, że należy wykonać wpisane w poprzednich liniach polecenie (polecenia).

Gdybyśmy omyłkowo zapomnieli podać w linii 3 średnika po słowie END wygenerowany zostanie następujący błąd:

```
SQL> BEGIN
  2   DBMS_OUTPUT.PUT_LINE('hello, world');
  3   END
  4   /
END
*
ERROR at line 3:
ORA-06550: line 3, column 3:
PLS-00103: Encountered the symbol "end-of-file" when expecting one of the
following:
; <an identifier> <a double-quoted delimited-identifier>
```

```
The symbol ";" was substituted for "end-of-file" to continue.
```

```
SQL>
```

5.2. Zapisywanie wyników zapytań i poleceń SQL w plikach

Często zachodzi potrzeba zapisania wyników zapytań SQL do pliku, by móc je potem edytować. W taki sposób można tworzyć raporty. Ponadto wyniki długo wykonujących się skryptów mogą być umieszczone w plikach, które potem można sprawdzać, weryfikować itd. Komenda `SPOOL` zapisuje wyniki generowane przez SQL*Plus do pliku (mówimy wówczas, że wyniki zapytania są przekierowane do pliku). Wydając więc np. polecenia:

```
SQL> SPOOL c:\temp\raport.txt
SQL> SELECT * FROM DEPT;
SQL> SPOOL OFF
```

otrzymamy plik tekstowy z następującą zawartością:

```
SQL> SELECT * FROM DEPT;

      ID NAME                REGION_ID
-----
      10 Finance                1
      31 Sales                   1
      32 Sales                   2
      33 Sales                   3
      34 Sales                   4
      35 Sales                   5
      41 Operations              1
      42 Operations              2
      43 Operations              3
      44 Operations              4
      45 Operations              5
      50 Administration          1

12 rows selected.

SQL> SPOOL OFF
```

Gdy nie jest określona pełna ścieżka dostępu do pliku, wówczas jest on zapisywany w katalogu roboczym. Jeśli nie zostało podane rozszerzenie, domyślnie przyjmowane jest `.LST`. Aby zatrzymać proces pisania do pliku i dokonać jego zamknięcia należy wydać polecenie:

```
SQL> SPOOL OFF
```

Z kolei polecenie

```
SQL> SPOOL OUT
```

zatrzymuje generowanie wyników i przesyła je bezpośrednio na domyślną drukarkę.

5.3. Definiowanie zmiennych użytkownika

W wielu przypadkach zachodzi potrzeba, aby wykonywane polecenie SQL (najczęściej w formie skryptu) mogło operować na wartościach pobieranych w trakcie jego wykonywania. Chcemy więc stworzyć zmienne definiowane przez użytkownika, do których będziemy mogli odwołać się w skrypcie. Musimy wówczas zdefiniować zmienną użytkownika. Używamy do tego polecenia `ACCEPT` w postaci:

```
SQL> ACCEPT nazwa-zmiennej
```

Wykonywanie skryptu zawierającego to polecenie jest wstrzymywane, a użytkownik zostaje poproszony o podanie wartości zmiennej. Po jej wprowadzeniu, wykonywanie skryptu jest

wznawiane od następnego wiersza. Każde wystąpienie zmiennej zostanie zamienione podaną wartością. Polecenie to jest nieocenione w przypadku zapytań, które wykonywane są wiele razy z różnymi warunkami. Zamiast pisać nowy skrypt za każdym razem, gdy wymagana jest inna wartość zmiennej, możemy w trakcie wykonywania poprosić o jej wprowadzenie użytkownika.

Po utworzeniu zmiennej, można się do niej odwołać w skrypcie, poprzedzając jej nazwę znakiem '&'. W przypadku wystąpienia tej samej nazwy zmiennej po raz drugi, użytkownik nie zostanie zapytany ponownie o jej wartość. W wyniku wykonania poniższego skryptu:

```
PROMPT
PROMPT Podaj zarobki:
ACCEPT zarobki
SELECT
    first_name, last_name, salary "płaca < &zarobki"
FROM
    emp
WHERE
    salary < &zarobki;
```

otrzymamy na ekranie (na twoim komputerze wynik może wyglądać nieco inaczej – patrz uwagi poniżej):

```
SQL> @c:\temp\zarobki.sql

Podaj zarobki:
1000
old  2:  first_name, last_name, salary "płaca < &zarobki"
new  2:  first_name, last_name, salary "płaca < 1000"
old  6:  salary < &zarobki
new  6:  salary < 1000

FIRST_NAME                LAST_NAME                p3aca < 1000
-----
George                     Smith                    940
Vikram                     Patel                    795
Chad                       Newman                   750
Alexander                  Markarian                850
Eddie                      Chang                    800
Radha                      Patel                    795
Bela                       Dancs                    860

7 rows selected.

SQL>
```

Powyższy wynik jest dość mało czytelny ze względu na dużą liczbę informacji systemowych. Program SQL*Plus posiada pewną liczbę poleceń pomagających „zapanować” nad sposobem wyświetlania wyników. Ich pełną listę można oczywiście znaleźć w dokumentacji. My użyjemy tutaj tylko czterech najczęściej używanych w praktyce. Obecnie na początku skryptu dodamy następujące komendy:

```
SET ECHO OFF
SET TERM ON
SET FEEDBACK OFF
SET VERIFY OFF
```

Wynik będzie więc następujący:

```
SQL> @c:\temp\zarobki.sql

Podaj zarobki:
1000
```

FIRST_NAME	LAST_NAME	p3aca < 1000
George	Smith	940
Vikram	Patel	795
Chad	Newman	750
Alexander	Markarian	850
Eddie	Chang	800
Radha	Patel	795
Bela	Dancs	860

SQL>

Samodzielnie zmieniaj wartości ON i OFF i zaobserwuj różnice. W razie wątpliwości zaglądaj do dokumentacji. Pamiętaj również, że gdy w ramach bieżącej sesji ustawisz wartość któregoś z parametrów, pozostanie on w tym stanie do czasu kolejnej jawnej zmiany. Nie wystarczy więc przykładowo usunąć ze skryptu polecenia SET TERM ON, aby kolejne wywołanie tego skryptu „widziało” już wartość OFF. Trzeba to zrobić jawnie wpisując po prostu SET TERM OFF. Czy potrafisz precyzyjnie nazwać, które elementy zostały „wygaszone” (w porównaniu do poprzedniego wyniku) ?

5.4. Przekazywanie parametrów do skryptów SQL*Plus

Często chcemy, aby utworzony przez nas skrypt otrzymywał wymagane parametry bezpośrednio w linii wywołania i nie angażował użytkownika tym samym użytkownika. W jednym wywołaniu przekazanych może być wiele parametrów. Aby omówić to zagadnienie rozważmy następujący skrypt:

```
SELECT
  first_name, last_name, salary
FROM
  emp
WHERE
  salary > &1 AND salary <&2;
```

Używa on dwu zmiennych. Wartości tych zmiennych mogą być określone poprzez przekazanie ich jako parametrów w linii wywołania:

```
SQL> @c:\temp\pracownicy_zarobki.sql 800 900
```

Skrypt będzie podstawiał parametry za zmienne w kolejności ich podawania: &1 zostanie zastąpione parametrem o wartości 800 a &2 parametrem o wartości 900. W wyniku wywołania powyższego skryptu otrzymany na ekranie:

```
SQL> @c:\temp\pracownicy_zarobki.sql 800 900
```

FIRST_NAME	LAST_NAME	SALARY
Alexander	Markarian	850
Bela	Dancs	860

SQL>

Jeżeli przekazanych w linii poleceń parametrów jest mniej niż zmiennych (lub nie podano wcale parametrów), wykonywanie skryptu jest wstrzymywane do momentu podania brakujących zmiennych. Dodatkowe zmienne skrypt ignoruje. Parametry wiązane są ze zmiennymi w kolejności ich podawania i tak: &1 zastępowane jest pierwszym parametrem, &2 drugim itd. Podając więc np.:

```
SQL> @c:\temp\pracownicy_zarobki.sql 800 900 1000
```

otrzymamy ten sam wynik, co powyżej, gdyż trzeci („nadmiarowy”) parametr został zignorowany. Gdy natomiast nie podamy żadnego parametru otrzymamy na ekranie poniższy obraz:

```
SQL> UNDEFINE 1
```

```
SQL> UNDEFINE 2
SQL> @d:\temp\pracownicy_zarobki.sql
Enter value for 1: 800
Enter value for 2: 900
old 6: salary > &1 AND salary <&2
new 6: salary > 800 AND salary <900
```

FIRST_NAME	LAST_NAME	SALARY
Alexander	Markarian	850
Bela	Dancs	860

```
SQL>
```

Polecenie UNDEFINE użyto, aby usunąć wcześniej zdefiniowane zmienne. Zgodnie bowiem z dokumentacją mamy:

```
UNDEFINE
-----

Deletes one or more user variables that you defined either
explicitly (with the DEFINE command), or implicitly (with a START
command argument).

UNDEF[INE] variable ...
```

W zależności od ustawień zmiennych systemowych programu SQL*Plus na ekranie możemy zobaczyć różne informacje „systemowe”, na przykład:

```
SQL> SET ECHO ON
SQL> SET TERM ON
SQL> SET FEEDBACK ON
SQL> SET VERIFY ON
SQL>
SQL>
SQL> @d:\temp\pracownicy_zarobki.sql 800 900
SQL> SELECT
  2   first_name, last_name, salary
  3 FROM
  4   emp
  5 WHERE
  6   salary > &1 AND salary <&2;
old 6: salary > &1 AND salary <&2
new 6: salary > 800 AND salary <900
```

FIRST_NAME	LAST_NAME	SALARY
Alexander	Markarian	850
Bela	Dancs	860

```
2 rows selected.

SQL>
```

Wprowadźmy teraz następujące wartości zmiennych:

```
SQL> SET ECHO OFF
SQL> SET TERM OFF
SQL> SET FEEDBACK OFF
SQL> SET VERIFY OFF
SQL> @d:\temp\pracownicy_zarobki.sql 800 900
SQL>
```

Tym razem chyba nieco przesadziliśmy, gdyż nie pojawia się żaden widoczny wynik. Nie oznacza to oczywiście, że zapytanie SQL nie wykonało się. Ono wykonało się, jednak program SQL*Plus

nic nie wyświetlił na ekranie, gdyż takie dostał od użytkownika wytyczne. Czy potrafisz powiedzieć, które zmienne systemowe odpowiadają za wyświetlenie, których informacji „systemowych”?

6. Polecenia do edycji komend SQL i bloków PL/SQL

Program SQL*Plus towarzyszy systemowi ORACLE od początków jego istnienia, czyli od końca lat 70. Odziedziczył po tym okresie wiele komend, które obecnie, wobec rozwoju programów do edycji tekstów, mają już znaczenie niemal symboliczne. Praca z tymi poleceniami, jak na obecne czasy, jest bardzo nieefektywna. Są to np. polecenia `APPEND`, `CHANGE`, `DEL`, `INPUT`, `LIST` i jeszcze parę innych. Przypominają one trochę polecenia *UNIX-owego* edytora *vi*. Obecnie raczej nie będziemy ich używać, preferując bardziej wygodne metody edycji plików (przykładowo poczciwy edytor *NOTEPAD* jest o niebo bardziej funkcjonalny niż wspomniane tu polecenia). Szczegółowy opis tych poleceń można znaleźć w dokumentacji.

7. Uzyskiwanie pomocy

Wydając polecenie:

```
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	XQUERY
CONNECT	PASSWORD	SHOW	

```
SQL>
```

oraz

```
SQL> HELP <nazwa_polecenia>
```

możemy uzyskać w szybki sposób listę dostępnych poleceń oraz składnię każdego z nich. Ponadto każde polecenie jest wyczerpująco opisane (często z przykładami użycia) w oryginalnej dokumentacji ORACLE. Zachęcamy do jej studiowania.

8. Przeglądanie i modyfikowanie zmiennych systemowych programu SQL*Plus

W programie SQL*Plus istnieje wiele zmiennych, które sterują jego zachowaniem. Przykładami są omówione już wcześniej zmienne `LINESIZE` oraz `PAGESIZE`. Często chcemy znać wartości tych zmiennych oraz zmieniać je w zależności od potrzeb. Wartości zmiennych zmieniamy poleceniem:

```
SQL> SET nazwa-zmiennej wartość-zmiennej
```

Chcąc wyświetlić aktualne wartości wszystkich zmiennych programu SQL*Plus należy wydać polecenie:

```
SQL> SHOW ALL
```

Powinniśmy wówczas otrzymać wynik podobny do poniższego:

```
appinfo is OFF and set to "SQL*Plus"
arraysize 15
autocommit OFF
autoprint OFF
autorecovery OFF
autotrace OFF
blockterminator "." (hex 2e)
btitle OFF and is the first few characters of the next SELECT statement
cmdsep OFF
colsep " "
compatibility version NATIVE
concat "." (hex 2e)
copycommit 0
COPYTYPECHECK is ON
define "&" (hex 26)
describe DEPTH 1 LINENUM OFF INDENT ON
echo OFF
editfile "afiedt.buf"
embedded OFF
escape OFF
escchar OFF
exitcommit ON
feedback OFF
flagger OFF
flush ON
heading ON
headsep "|" (hex 7c)
instance "local"
linesize 80
lno 5
loboffset 1
logsource ""
long 80
longchunksize 80
markup HTML OFF HEAD "<style type='text/css'> body {font:10pt Arial,Helvetica,sans-serif; color:black; background:White;} p {font:10pt Arial,Helvetica,sans-serif; color:black; background:White;} table,tr,td {font:10pt Arial,Helvetica,sans-serif; color:Black; background:#f7f7e7; padding:0px 0px 0px 0px; margin:0px 0px 0px 0px;} th {font:bold 10pt Arial,Helvetica,sans-serif; color:#336699; background-color:#cccc99; padding:0px 0px 0px 0px;} h1 {font:16pt Arial,Helvetica,Geneva,sans-serif; color:#336699; background-color:White; border-bottom:1px solid #cccc99; margin-top:0pt; margin-bottom:0pt; padding:0px 0px 0px 0px;} h2 {font:bold 10pt Arial,Helvetica,Geneva,sans-serif; color:#336699; background-color:White; margin-top:4pt; margin-bottom:0pt;} a {font:9pt Arial,Helvetica,sans-serif; color:#663300; background:#ffffff; margin-top:0pt; margin-bottom:0pt; vertical-align:top;}</style><title>SQL*Plus Report</title>" BODY "" TABLE "border='1' width='90%' align='center' summary='Script output'" SPOOL OFF ENTMAP ON P
REFORMAT OFF
newpage 1
null ""
numformat ""
numwidth 10
pagesize 14
PAUSE is OFF
pno 1
recsep WRAP
recsepchar " " (hex 20)
release 1102000100
repfooter OFF and is NULL
repheader OFF and is NULL
```

```

securedcol is OFF
serveroutput OFF
shiftinout INVISIBLE
showmode OFF
spool OFF
sqlblanklines OFF
sqlcase MIXED
sqlcode 0
sqlcontinue "> "
sqlnumber ON
sqlpluscompatibility 11.2.0
sqlprefix "#" (hex 23)
sqlprompt "SQL> "
sqlterminator ";" (hex 3b)
suffix "sql"
tab ON
termout OFF
timing OFF
trimout ON
trimspool OFF
ttitle OFF and is the first few characters of the next SELECT statement
underline "-" (hex 2d)
USER is "SUMMIT2"
verify OFF
wrap : lines will be wrapped
errorlogging is OFF
SQL>

```

9. Tworzenie raportów z użyciem SQL*Plus

Program SQL*Plus, jakkolwiek dość siermiężny w obsłudze, bardzo dobrze sprawdza się w sytuacjach, gdy chcemy „na szybko” przygotować ładnie wyglądający raport (oparty np. o polecenie `SELECT` lub procedurę PL/SQL-ową). Poniżej zamieszczono skrypt, który wykonuje pewien raport. Wynik jest wyświetlany na ekranie i dodatkowo zapisywany w pliku `raport.txt`. Skrypt celowo nie jest w najmniejszym stopniu skomentowany. Opis nieznanych poleceń należy samodzielnie znaleźć w dokumentacji.

```

SET ECHO OFF
SET TERM OFF
SET FEEDBACK OFF
SET VERIFY OFF
SET UNDERLINE =
SET LINESIZE 80
SET PAGESIZE 42
SET COLSEP "|"
SET HEADING ON
TTITLE OFF
BTITLE OFF

COL col_dzisiaj NOPRINT NEW_VALUE dzisiejsza_data
SELECT TO_CHAR(sysdate,'dd-mm-yyyy') col_dzisiaj FROM dual;
TTITLE CENTER 'Lista pracowników i ich zarobki' SKIP 1 -
      CENTER '-----'
BTITLE CENTER 'Tajne, do użyciu wewnętrznego' SKIP 1 -
      CENTER '-----' SKIP 2 -
      LEFT 'Strona:' FORMAT 999 SQL.PNO RIGHT dzisiejsza_data

COLUMN id FORMAT 9999 HEADING Numer|ID
COLUMN first_name FORMAT A12 HEADING Imię
COLUMN last_name HEADING Nazwisko FORMAT A15
COLUMN start_date HEADING Data|przyjęcia
COLUMN dept_id FORMAT 99999 HEADING Dział

```

```
COLUMN salary FORMAT 99G999D99 HEADING Zarobki
```

```
SPOOL C:\temp\raport.txt
```

```
BREAK ON REPORT
```

```
COMPUTE SUM LABEL "Suma:" OF salary ON REPORT
```

```
SELECT
```

```
  RPAD(id,5), first_name, UPPER(last_name),  
  SUBSTR(UPPER(first_name),1,1)||'. '||  
  SUBSTR(UPPER(last_name),1,1)||'. ' "Inic.",  
  dept_id,  
  TO_CHAR(start_date,'dd-mm-yyyy'), salary
```

```
FROM
```

```
  emp;
```

```
SPOOL OFF
```

```
SET UNDERLINE '-'
```

```
TTITLE OFF
```

```
BTITLE OFF
```

```
SET ECHO ON
```

Po wykonaniu powyższego skryptu powinniśmy otrzymać plik C:\temp\raport.txt z następującą zawartością:

Lista pracowników i ich zarobki						
Numer	Imię	Nazwisko	Inic.	Dział	Data przyjęcia	Zarobki
1	Carmen	VELASQUEZ	C. V.	50	03-03-1990	2,500.00
2	LaDoris	NGAO	L. N.	41	08-03-1990	1,450.00
3	Midori	NAGAYAMA	M. N.	31	17-06-1991	1,400.00
4	Mark	QUICK-TO-SEE	M. Q.	10	07-04-1990	1,450.00
5	Audry	ROPEBURN	A. R.	50	04-03-1990	1,550.00
6	Molly	URGUHART	M. U.	41	18-01-1991	1,200.00
7	Roberta	MENCHU	R. M.	42	14-05-1990	1,250.00
8	Ben	BIRI	B. B.	43	07-04-1990	1,100.00
9	Antoinette	CATCHPOLE	A. C.	44	09-02-1992	1,300.00
10	Marta	HAVEL	M. H.	45	27-02-1991	1,307.00
11	Colin	MAGEE	C. M.	31	14-05-1990	1,400.00
12	Henry	GILJUM	H. G.	32	18-01-1992	1,490.00
13	Yasmin	SEDEGHI	Y. S.	33	18-02-1991	1,515.00
14	Mai	NGUYEN	M. N.	34	22-01-1992	1,525.00
15	Andre	DUMAS	A. D.	35	09-10-1991	1,450.00
16	Elena	MADURO	E. M.	41	07-02-1992	1,400.00
17	George	SMITH	G. S.	41	08-03-1990	940.00
18	Akira	NOZAKI	A. N.	42	09-02-1991	1,200.00
19	Vikram	PATEL	V. P.	42	06-08-1991	795.00
20	Chad	NEWMAN	C. N.	43	21-07-1991	750.00
21	Alexander	MARKARIAN	A. M.	43	26-05-1991	850.00
22	Eddie	CHANG	E. C.	44	30-11-1990	800.00
23	Radha	PATEL	R. P.	34	17-10-1990	795.00
24	Bela	DANCS	B. D.	45	17-03-1991	860.00
25	Sylvie	SCHWARTZ	S. S.	45	09-05-1991	1,100.00
Suma:						31,377.00

Tajne, do użytku wewnętrznego

Strona: 1 18-03-2012

LITERATURA

*SQL*Plus User's Guide and Reference* – oryginalna dokumentacja dołączana do systemu ORACLE

*SQL*Plus Quick Reference* – oryginalna dokumentacja dołączana do systemu ORACLE

Dokumentacja dostępna jest na stronie:

<http://www.oracle.com/technetwork/indexes/documentation/index.html>