

Praca z SQL Plus

Agenda

- Dlaczego SQL*Plus
- Wersje SQL*Plus w Oracle na systemie operacyjnym Windows
- Uruchamianie SQL*Plus i łączenie się z bazę danych
- Rozwiązywanie problemu ORA-01034 ORACLE not available
- Wykonywanie zapisanych zapytań
- Domyślny katalog dla SQL*Plus
- Polecenie host
- Praca z edytora tekstu z poziomu SQL*Plus
- Czyszczenie ekranu za pomocą CLEAR SCREEN

Agenda

- Jak efektywnie używać SQL*Plus
- Przydatne ustawienia
 - Jak zmienić rozmiar i nazwy czcionki w SQL*Plus
 - Jak zignorować puste linie w SQL*Plus?
- SQL*Plus i zmienne związane
- Jak zapisać wyeksportować wyniku zapytania do plik Excel
- Jak zakończyć pracę z SQL*Plus
- Dokumenty, zasoby

Dlaczego SQL *Plus

- Miejsce SQL*Plus w środowisko Oracle
- SQL*Plus jest programem dostępu do bazy danych Oracle w trybie znakowym, który jest instalowany przy każdej instalacji bazy danych
- Graficzne narzędzia, wygodne,
 - nie zawsze dostępne,
 - komercyjne, różne interfejsy różnych producentów
- iSQLPlus - dostępny w Oracle 10g
- W starszych wersjach SQL * Plus - podstawowe narzędzie do administracji wraz z Server Manager (svrmgr do wersji bazy danych Oracle 8.2) (teraz Oracle Enterprise Manager)

Historia SQL*Plus

- Oryginalna nazwa UFI (User Friendly Interface)
- Autor narzędzia SQL*Plus Bruce Scott.
Każdy DBA i użytkownik zaawansowany zna użytkownika Scott z hasło „tiger” - właściciel schematu Scott z przykładowymi tabelami, które są instalowane z każdą wersją bazy danych Oracle, którą instalujemy z przykładowymi schematami
 - początkowo wprowadzanie kwerend i otrzymywanie ich wyników;
 - następne
 - możliwości skryptowe - raz przetestować, zachować zapytania
 - formatowanie wyników zapytań

Do czego jest używany SQL*Plus

- Wsadowe wykonywanie poleceń SQL (wykonywanie skryptów SQL);
- Interaktywne wykonywanie poleceń SQL - wprowadzanie, redagowanie, zachowywanie, drukowanie, wykonywanie;
- Wykonywanie bloki PL/SQL
/Blok PL/SQL - grupa powiązanych operatorów PL/SQL, które tworzą blok anonimowy (nienazwany);
- Formatowanie wyników zapytań w raporty i eksportowanie ich do Excela, do plików tekstowych lub strony HTML;
- Uzyskanie informacji o obiektach przechowywane w bazy danych (describe);

Do czego jest używany SQL*Plus

- Kopiowanie danych między instancjami (copy);
- Administracja instancji bazy danych (teraz przeważnie z użyciem Oracle Enterprise Manager) - uruchamianie skryptów podczas instalacji (aktualizacji, migracji) produktów Oracle;
- Eksport (ekstrakcja) informacji z bazy danych do płaskiego pliku.
 - dane o stałej szerokości kolumn;
 - comma-delimited file - plik stosujący przecinek „,” przy eksporcie informacji, która później jest wprowadzana do Excela lub Ms Access.
- Tworzenie kodu SQL poprzez zapytania SQL.

Ograniczenia SQL*Plus

- Co musimy brać pod uwagę?
- SQL*Plus nie jest w stanie zaprezentować informacji która jest w
 - binarnej formie ani
 - typu LOB (multimedialna, dźwiękowa, graficzna)
- W SQL*Plus brakuje stwierdzenie **IF**
- Nie ma możliwości budowy pętli
- Jest bardzo ograniczone przetwarzanie błędów
- Jest bardzo ograniczone wsparcie kontroli wprowadzonych przez użytkownika wartości.

Typy poleceń SQL*Plus

Według autorów książki „Oracle 8 server. Księga eksperta”^{*} można wyróżnić sześć typów poleceń SQL*Plus:

- inicjujące środowisko SQL*Plus;
 - polecenia wykonywalne SQL*Plus;
 - polecenie edycyjne SQL*Plus;
 - polecenia formatujące SQL*Plus;
 - polecenia dostępu do różnych baz danych;
 - inne polecenia.
-
- ^{*} Joe Greene, „Oracle 8 server. Księga eksperta”, Helion 2000, s. 526

Typy poleceń SQL*Plus

- Data Definiton Language (DDL) polecenia, które umożliwiają nam konserwować i modyfikować bazę danych
 - CREATE TABLE tworzyć tabele,
 - CREATE TABLESPACE tworzyć przestrzenie tabel (tablespaces)
 - CREATE INDEX
 - CREATE USER
- Data Manipulation Commands (DMC)
 - INSERT, UPDATE, DELETE, SELECT

Wersje programu SQL*Plus w systemach operacyjnych Windows

Są dwie wersje programu SQL*Plus w systemach operacyjnych Windows:

- sqlplusw.exe – wersja dla systemu Windows
- sqlplus.exe – wersja dla systemu Ms DOS.

Uruchamianie SQL*Plus i łączenie się do bazy danych

- W celu uruchomienia SQL*Plus należy:
 - wybrać program SQL*Plus z programów
 - wybrać skrót do programu SQL *Plus
 - wybrać ścieżka i program
- Przy uruchomieniu programu bez parametrów się pojawia okno dialogowe w którym należy wprowadzić
 - User (Użytkownik)
 - Password (Hasło)
 - Service (Nazwa usługi łączenia do bazy danych)
- Program SQL*Plus można uruchomić również z parametrami
sqlplusw user/password@serwis na przykład
sqlplusw student/student@WNTTEST
- Uwaga w systemach Linux/Unix polecenie sqlplus musi być wprowadzone małymi literami, ponieważ te systemy odróżniają małe i duże litery i Root <>root, SqlPlus<>sqlplus.

Uruchamianie SQL*Plus i łączenie się do bazy danych

- Jeżeli chcemy użyć opcji roli (SYSDBA, SYSOPER) w systemie operacyjnym Ms Windows „user/password@instance role” muszą być ujęte w „”. Z poziomu MS-DOS wpisujemy:
Sqlplusw „student/student@WNTTEST as SYSDBA”
- sqlplusw /nolog uruchamia SQL*Plus bez łączenia się do bazy danych
- Należy wykonać polecenia CONNECT
USER/PASSWORD@SERVICE
w celu połączenia z bazą danych.
SQL>CONNECT student/student@WNTTEST;
SQL> CONNECT scott/tiger@WNTTEST;
- Poprzez polecenie CONNECT również możemy się przełączać między instancjami bazy danych na przykład
SQL>CONNECT student/student@LNTEST;

Uruchamianie SQL*Plus i łączenie się do bazy danych

- Można również ukazać w wierszu poleceń przy uruchamianiu SQL*Plus nazwa pliku, który ma być wykonany przy uruchamianiu SQL*Plus
`sqlplusw user/password@service @file_name.SQL`
`sqlplusw @file_name`
- Ze względów bezpieczeństwa
`sqlplusw user/password@service @file_name.SQL`
wyżej podane logowanie należy unikać.
- Jeżeli przy uruchomieniu SQL*Plus wstawimy „-S” program SQL*Plus nie wyświetli informacji o wersji bazy danych do której się połączyliśmy.
`Sqlplusw -S „user/password@service”`

Co jeżeli nie możemy się połączyć z instancją bazy danych

- Jeżeli się pojawi przy próbach uruchomienia SQL*Plus:
- ORA-01034 ORACLE not available
/ baza danych Oracle jest niedostępna/
- ORA-27101 Shared memory realm does not exist
- należy najpierw z poziomu Oracle Enterprise Manager uruchomić (startup) instancji bazy danych i wtedy się połączyć za pomocą SQL * Plus
- Administrator bazy danych może ograniczyć dostęp użytkowników do bazy danych Oracle z użyciem SQL*Plus poprzez tabelę PRODUCT_USER_PROFILE lub poprzez PROFILES.

Jak uzyskać pomoc?

- SQL>HELP
- SQL>HELP INDEX
pokazuje listy dostępnych tematów pomocy
- SQL>HELP TOPIC pokazuje listy dostępnych tematów pomocy,
każdy na osobnym wierszu.
- Następne możemy wpisać polecenia dla którego chcemy
uzyskać pomoc na przykład
- SQL>HELP [POLECENIE]
SQL>HELP CL
lub
- SQL>HELP CLEAR

Polecenia SQL*Plus

- Wszystkie polecenia SQL*Plus są niezależnie od wielkości liter - możemy stosować małe, duże lub małe i duże litery.

Polecenia

SQL>clear screen

SQL>CLEAR SCREEN

SQL>CLEAR screen

są równoważne.

- Uwaga nazwy plików w skryptach SQL w zależności od systemu operacyjnego są lub nie zależnie od wielkości liter. W Windows nie mamy zależności, natomiast w Unix/Linux jest zależność nazwy plików od wielkości liter i „**Nasze_zapytanie.sql**” nie jest jednakowe z „**nasze_zapywanie.sql**”.

Wykonywanie poleceń SQL*Plus

- Polecenia SQL*Plus nie muszą być zakończone „;”, ale można na koniec wstawiać „;”.
- HOST
lub
- HOST;

Bufor SQL*Plus

- SQL*Plus przechowuje w buforze kopię ostatnie wykonanej kwerendy (zapytania) lub blok PL/SQL. Tam również jest przechowywana treść wprowadzonego zapytania.
- W buforze są przechowywane polecenia SQL, bloki kodu PL/SQL ale nie polecenia SQL*Plus. Na przykład w buforze nie będzie przechowywane
SET SERVEROUTPUT ON.
- Zawartość bufora można zachować na dysku w plik, który później można wykonać jako skrypt SQL.
SAVE nazwa_pliku[.ext] [CRE[ATE] | REP[LACE] | APP[END]]

Skrypt SQL

- Skrypt SQL jest plik tekstowy, który zawiera jedna lub więcej kwerend (zapytań). W nim może być odwołanie do kolejnych skryptów SQL.
- W momencie wykonywania zapytań z pliku SQL one są wykonywane tak, jak by były wprowadzane z klawiatury.
- Skrypt SQL może zawierać zapytania SQL, polecenia SQL*Plus, bloki SQL.
- W skryptach SQL możemy przechowywać naszej wiedzy dotyczącej administracji, strojenia, ładowania danych, określonej wiedzy biznesowej... Skrypty oszczędzają czas i pomylek przy ponownym wprowadzeniu zapytania.
- Skrypty SQL są używane przez firmy Oracle przy instalacji bazy danych, jej aktualizacji, migracji, przy instalacji nowych produktów.

Wykonywanie zapisanych zapytań, kwerend

- SQL>START d:\oracle\product\10.1.0\WNTTEST\BIN\test
- SQL>@nazwa_zapytania.???
- /jeżeli rozszerzenie ??? Zapytania jest sql ??? Może być ominięte/
- Na przykład jeżeli mamy
 - zapytanie_1.sql
 - zapytanie_2.txt
- w celu ich wykonania należy:
- SQL >@zapytanie_1.sql
- jest jednakowe z
- SQL >@zapytanie_1
- SQL >@zapytanie_2.txt

Wykonywanie zapisanych zapytań, kwerend

- @@ - działa tak samo jak @ z tym różnicą, że poszukuje skryptu w katalogu z którym był zapuszczony główny skrypt.
- Jest używane do wykonywania zapytań SQL, które zawierają odwołania do kilku plików SQL w tym samym katalogu.
- Na przykład plik wykonaj.sql posiada
 - @@wykonaj101
 - @@wykonaj102
 - @@wykonaj103

Wykonywanie zapisanych zapytań, kwerend

- W SQL*Plus zapytania są zakończone:
 - ;
 - /
 - pusty wiersz
- Symbol „;” oznacza wykonywanie zapytania.
- Symbol „/” oznacza, że trzeba wykonać zapytania.
- Pusty wiersz oznacza, że jest zakończone wprowadzanie zapytania ale jeszcze nie należy go wykonywać.
Uwaga - puste wiersze w długich zapytań w celu czytelności zapytania. Ustawienia środowiskowe SET SQLBLANKLINES.

Wstawianie komentarzy

W skryptach SQL w celu ich łatwiejsze czytanie, uporządkowanie należy stosować nagłówki, które określają:

- do czego jest używany dany skrypt,
- w której wersji bazy danych
- jeżeli jest z Internetu, z której strony
- autor skryptu
- data utworzenia skryptu
- grupa do której należy dany skrypt
- przykładowe wywołanie skryptu
- przykładowe wyniki działania skryptu

Wstawianie komentarzy

- REM [ARK] - komentarz na jednym wierszu
- /*
• Komentarze na kilku wierszach obowiązkowo z spacji po /* i spacji przed */
• */
- -- komentarz na jednym wierszu
- W komentarzach nie wolno używać &

Wstawianie komentarzy

- Rem - Zapytanie o placach w schemacie Scott
 - -- To też jest komentarz
 - SELECT ename, sal from scott.emp;
 - /* Teraz wstawiamy komentarze
 - na dwóch wierszach
 - */
-
- @d:\oracle\product\10.1.0\WNTTEST\BIN\komentarze

Polecenie host

Poleceniem:

SQL>host

lub

SQL>ho

lub

SQL>\$

widzimy nasz katalog roboczy.

Jesteśmy w stanie wykonać polecenia MS DOS i wrócić z powrotem do SQL*Plus po wpisaniu polecenia exit.

Również, możemy wykonać polecenie systemu operacyjnego i wrócić do SQL*Plus automatycznie po wykonaniu polecenia
\$ dir

Domyślny katalog dla SQL*Plus

Dlaczego jest ważne ustawianie domyślnego katalogu?
Stosując domyślny katalog nie musimy wprowadzać pełną ścieżkę dostępu do skryptu, którego chcemy wykonać, uruchomić. W domyślnym katalogu są zapisywane pliki tymczasowe przez SQL*Plus.

Katalog BIN przechowuje programy i nie jest zalecane aby tam były przechowywane skrypty SQL. W produkcyjnych środowiskach możemy mieć podkatalogi na przykład
TUNING,
EXPORT

...

Domyślny katalog dla SQL*Plus

- Jaki jest domyślny katalog dla SQL*Plus?
 - Domyślnym katalogiem dla SQL*Plus jest ten katalog,
 - - w którym użytkownik był w momencie uruchomienia SQL*Plus;
 - - \$ORACLE_HOME\BIN.
- ORACLE_HOME oznacza główny katalog, w którym jest zainstalowane oprogramowanie Oracle.

Domyślny katalog dla SQL*Plus

- Katalogi, które są przeszukiwane o skryptach SQL są określone w zmiennej środowiskowej SQLPATH.
- W celu jej zmiany należy edytować rejestr Windows.
 - Uruchamiamy cmd\regedit32.
 - Wyszukujemy SQLPATH i edytujemy tę zmienną.
- Inną możliwością jest przejście do katalogu w którym są przechowywane nasze pliki sql i uruchomienie stamtąd SQL*Plus poprzez wykonywanie polecenia
ORACLE_HOME\bin\sqlplusw.exe w naszym przypadku
Oracle\client\bin\sqlplusq.exe

Polecenie DESCRIBE

Poleceniem DESCRIBE (DESC) uzyskujemy opis tabeli, widoku (perspektywy), synonimu, opis funkcji lub procedury.

DESCRIBE schema.obiekt@serwis

Naprzkład chcemy uzyskać opis tabeli hr.regions. Orzynamy opis kolumn, typ danych, długość (ile znaków po przecinku)

DESCRIBE hr.regions@WNTTEST

lub

DESC hr.regions@WNTTEST

Polecenie DESCRIBE

- Druga metoda uzyskania informacji jest Oracle Data Dictionary - zbiór tabel systemowych, które zbierają informacji o obiektach w bazie danych (repozytorium informacji o obiektach w bazie danych). Właściciel tabeli opisujących obiektów w bazie Oracle jest użytkownik **SYS**.
- W celu uzyskaniu informacji o tabeli hr.regions należy zadać pytanie do dwóch tabel z Oracle Data Dictionary:
 - ALL_TABLES;
 - ALL_TAB_COLUMNS.

Polecenie DESCRIBE

- Jeżeli nas interesuje kwestia bezpieczeństwa - kto ma dostęp do danej tabeli, to możemy uzyskać odpowiedź z widoków:
 - ALL_TAB_PRIVS
 - ALL_COL_PRIVS
 - DBA_TAB_PRIVS_MADE
 - DBA_COL_PRIVS_MADE
- W celu dostępu do powyższych widoków, kiedy chcemy uzyskać informację o tabelach, które mają inny właściciel niż my musimy posiadać uprawnienia DBA.

Polecenie DESCRIBE

- W celu uzyskaniu informacji o indeksach w bazie należy użyć tabeli:
 - ALL_INDEXES
 - ALL_IND_COLUMNS.
- Możemy uzyskać informacji o
 - wszystkich indeksach związanych z daną tabelą lub
 - opis jednego indeksu

Polecenie DESCRIBE

- Wyzwalacze (triggers)
- Informacja o wyzwalaczach (triggers) jest przechowywana w:
 - ALL_TRIGGERS;
 - ALL_TRIGGER_COLS
- Synonimy (alternatywna nazwa tabeli)
- Kiedy w naszych programach zamiast tabeli używamy ich synonimy zabezpieczamy się przed zmian w nazwy, właściciela, miejsca tabeli.
- W celu otrzymania informacji o synonimach należy zadać zapytanie do tabeli ALL_SYNONYMS.

Polecenie PROMPT

- Wyświetla na monitorze pusty wiersz lub zawartość tekstu.
- PRO [MPT] <tekst>
- Polecenie PROMPT jest stosowane do opisu jakiej informacji dostarczą dany skrypt, zapytanie, raport.
- Polecenie PROMPT jest stosowane wraz z ACCEPT w celu wyświetlania informacji dotyczącej wprowadzania niezbędnej informacji.

Polecenie ACCEPT

- Polecenie ACCEPT prosi użytkownika o podaniu wartości zmiennej i zapisuje tej wartości w zmiennej użytkownika.
- Polecenie ACCEPT pozwala w pewnym stopniu kontrolować wartości, których użytkownik wprowadza (typ danych). Na przykład użytkownik nie może wprowadzić litery, tam gdzie są wymagane wartości.
- ACC [EPT] <nazwa_zmiennej> [<typ zmiennej>]
FOR [MAT] <format> [DEF [AULT] <wartość_domyślna>]
typ_zmiennej ::= NUM [BER] | CHAR | DATE

SQL> PROMPT

SQL> PROMPT Proszę wprowadzić minimalnej płacy (850)

SQL> ACCEPT sal_min number prompt 'Minimalna_płaca:'

Polecenie ACCEPT

- SQL> PROMPT
- SQL> PROMPT Proszę wprowadzić minimalnej płacy (850)
- Proszę wprowadzić minimalnej płacy (850)
- SQL> ACCEPT sal_min number prompt 'Minimalna_płaca:'
- 'Minimalna_płaca:'870
- SQL> SELECT ename, sal from scott.emp where sal>&sal_min;
- old 1: SELECT ename, sal from scott.emp where sal>&sal_min
- new 1: SELECT ename, sal from scott.emp where sal> 870
- ENAME Płaca
- -----
- ALLEN 1,600
- WARD 1,250
- JONES 2,975
- 13 rows selected.

Metodologia tworzenia raportów w SQL*Plus

- Zbieranie informacji, podejmowanie decyzji wraz z użytkownikiem dla którego jest przeznaczony dany raport
 - Jakie informacje potrzebujemy
 - Jaki tytuł raportu
 - Data czas wykonania raportu
 - Jaka częstotliwość wykonania
 - Jakie zmienne będą określać zakres raportu
 - Jakie kolumny w raporcie
 - Czy będzie suma i suma częściowa w raporcie (TOTAL, SUBTOTAL)
 - Nagłówek raportu
 - Które tabele będziemy używać?

Metodologia tworzenia raportów w SQL*Plus

- Kto będzie wykonywał dany raport (jako użytkownik biznesowy, administrator, użytkownik z większymi uprawnieniami)
- Czy tworzyć skrót do automatycznego wykonywania raportu w którym są przechowywane użytkownik i hasło?
- Zasada bezpieczeństwa, role...
- Jaka wersja SQL*Plus w Windows będzie używana sqlplus lub sqlplusw?
- Jaka nazwa skryptu tworzącego raportu?
- W jakim katalogu będzie przechowywany?
- Uwagi, które wyjaśniają kwestie związane z raportu w celu uniknięcia dwuznaczności w jego interpretacji.

Metodologia tworzenia raportów w SQL*Plus

- Cały proces tworzenia raportu - zapytań SQL zachowujemy w pliku dla późniejszego wykonania lub zmiany.
- Proces tworzenia raportu jest interaktywny proces
 - budowa zapytania;
 - formatowanie wyników zapytania;
 - rozmiar strony - parametr ile wiersze na strony
SET NEWPAGE 0
SET PAGESIZE 55
 - ile symbole na wierszu
SET LINESIZE 80
 - nagłówek raportu
TTITLE CENTER 'NAZWA RAPORTU' SKIP 2 -
LEFT 'ODDZIAŁ'
LEFT '===== ' -
'====='

Metodologia tworzenia raportów w SQL*Plus

- stopka raportu

```
BTITLE LEFT '-----' -  
          '-----'
```

SKIP 1

```
LEFT 'Uruchomił ' SQL.USER
```

```
RIGHT 'Strona ' FORMAT 999 SQL.PNO
```

- formatowanie nagłówków kolumn

```
COLUMN pole_1 HEADING 'Nagłówek 1' FORMAT 999
```

```
COLUMN pole_2 HEADING 'Nagłówek 2' FORMAT A20 -  
WORD_WRAPED
```

```
COLUMN pole_3 HEADING 'Data ' FORMAT A11
```

Metodologia tworzenia raportów w SQL*Plus

- - po zakończeniu raportu wyłączenie tytułów i nagłówków
TTITLE OFF
BTITLE OFF
- - przerwy na stronach - BREAK
 - test pierwszego wariantu raportu;
 - wstawianie sum, sum częściowych (TOTAL, SUBTOTAL)
polecenia BREAK i COMPUTE
 - akceptacja i proponowanie zmian przez użytkownika;
 - kolejny wariant raportu

Metodologia tworzenia raportów w SQL*Plus

- Komentarze, teraz wiemy co zrobiliśmy:
 - Czy za parę miesięcy będziemy pamiętać?
 - Praca w grupie

- Nazwa pliku :sql
- Działanie
- Data Utworzenia
- Data Modyfikacji
- Autor
- Wymagana informacja wejściowa
- Przykładowy wynik

Formatowanie wyników zapytania

- W SQL*Plus możemy formatować wyników zapytania w celu określenia jak mają być wyświetlane dane, nazwa kolumn, nagłówki stron.
- Obecnie są dostępne graficzne narzędzia do edycji zapytań i eksportu wyników do Excelu (Toad) lub tworzenia raportów (Oracle Reports).
- W iSqlPlus wyniki zapytania są lepiej przedstawione niż w SQL*Plus.
- Podstawa formatowania wyników w SQL*Plus jest polecenie
COLUMN
COL [UMN].

Formatowanie wyników zapytania

- COL nazwa_kolumna format 9,999.99
ukazuje, że format kolumny nazwa_kolumna będzie liczbowy z dwóch znaków po przecinku i , oddzielająca tysiącach.
- COL nazwa_kolumna heading „NAGŁÓWEK KOLUMNY”
oznacza, że nazwa_kolumna będzie miała nagłówek pod nazwy „NAGŁÓWEK KOLUMNY”
- Naprzykład
- COL sal format 9,999.99
- COL sal heading „Płaca”
- Select sal, ename from scott.emp;

Formatowanie wyników zapytania

- `SET LINESIZE 80`
ustała szerokość linii raportu na 80 symboli
- `SET PAGESIZE 50`
ustała liczbę linii (50) na stronie raportu.
- `SET TERMOUT OFF`
zapobiega wyświetlaniu wyników zapytania na monitorze. Ta opcja jest używana przy korzystaniu z polecenia `SPOOL`.
- `SET ECHO OFF`
nie wyświetla poleceń SQL.

Zapis w pliku i wydruk wyników wykonania zapytania

- Do tego celu jest wykorzystywane polecenie SPOOL
- SPO [OL] nazwa_pliku.??? OFF | OUT
- Standardowe rozszerzenie nazwy pliku jest LST lub LIS
- Polecenie SPOOL OFF wyłącza spooling
- Polecenie SPOOL OUT wysyła wynik na drukarce.
- Jeżeli ustawimy zmienną środowiskową SET TERMOUT OFF wynik zapytania nie będzie wyświetlany na monitorze.
- W Windows domyślny katalog plików jest \$ORACLE_HOME\BIN. Ten katalog nie jest najlepsze miejsce dla przechowywaniu plików ze skryptami.

Praca z edytora tekstu z poziomu SQL*Plus

- Jeżeli nam się nie podoba wbudowany w SQL*Plus edytor tekstu możemy stosować domyślny edytor (w Windows **notepad**; w Unix/Linux - **VI**)
- Kiedy jesteśmy w SQL * Plus i chcemy uruchomić domyślnego edytora należy wpisać:
- SQL> edit nazwa_zapytania.sql
lub
- SQL>ed plik.sql
- Uruchomi się domyślny edytor SQL

Czyszczenie ekranu za pomocą **CLEAR SCREEN**

- Jeżeli chcemy wyczyścić ekran należy zastosować polecenia

CLEAR SCREEN

lub

CL SCR

Strojenie zapytań SQL

- Uzyskanie informacji o czasie wykonywania zapytań
- SET AUTOTRACE ON | OFF

Uzyskanie informacji o czasie wykonywania zapytań

- W celu uzyskaniu informacji o czasie niezbędnym dla wykonania poszczególnych zapytań (kwerend) i planowane ich wykonywaniu (późniejsza ich optymalizacji) należy użyć polecenia TIMING.
- TIMI[NG] START [nazwa zegara] | SHOW | STOP
- Opcja START - uruchamia mierzenie czasu wykonywania kwerendy.
- Opcja SHOW pokazuje czas, który upłynął od uruchomienia zegara.
- Opcja STOP zatrzymuje zegara i go usuwa z bufora.

Uzyskanie informacji o czasie wykonywania zapytań

timig start zegar

```
select * from hr.regions;
```

timing show

timig stop

W celu usunięcia wszystkich uruchomionych zegarów należy zastosować polecenia `CLEAR TIMING`.

Jeżeli ustawimy opcji środowiska `SET AUTOTRACE ON` będziemy utrzymywać po każdym wykonaniu zapytania jaki czas był potrzebny.

SET AUTOTRACE ON | OFF

- Jeżeli mamy zapytania, o których uważamy, że są źle zoptymalizowane możemy uzyskać informacje jak one są przetwarzane przez Oracle w celu optymalizacji ich wykonania.
- Stosując ustawienia zmiennej środowiskowej SQL*Plus **AUTOTRACE** na **ON** uzyskujemy analiza wykonania zapytania (EXPLAIN PLAN).
- SET AUTOTRACE ON
- Wykonanie zapytania, dla którego chcemy uzyskać planu wykonania
- Utrzymanie na monitorze planu wykonania zapytania
- SET AUTOTRACE OFF - wyłącza tej opcji
- Przed użyciem opcji AUTOTRACE musi być utworzona tabela **PLAN_TABLE**, jeżeli nie istnieje z użyciem skryptu **UTLXPLAN.SQL**.
- Problemy związane z użyciem tej metody napotykamy wtedy, kiedy chcemy uzyskać plany wykonania zapytań, które mają długi czas wykonania.

Przydatne ustawienia

- Co jest przydatne zmienić, dodać w SQL*Plus:
- Jeżeli się łączymy do różnych instancji Oracle (różnych) baz danych jest przydatne wiedzieć do której bazy się łączymy
- Również jest przydatne wiedzieć czas, który jest potrzebny do wykonania poszczególnych kwerend.
- Zmienić domyślnego edytora w Windows z notatnik na Wordpad.exe lub vi.exe (dla pracujących pod Unix/Linux)
- Jak to zrobić (osiągnąć)?
- W pliku konfiguracyjnym GLOGIN.SQL jest przechowywana informacja o personalizacji środowiska SQL*Plus, która obowiązuje dla wszystkich użytkowników danego komputera, którzy się logują do bazy Oracle przy użyciu SQL*Plus.
Plik GLOGIN.SQL jest wykonywany automatycznie przy każdym uruchomieniu SQL*Plus.
- W pliku konfiguracyjnym LOGIN.SQL jest przechowywana informacja o profilu użytkownika.

Przydatne ustawienia - nazwa instancji bazy danych

- Przechodzimy od katalogu ORACLE_HOME **d:\Oracle\client**
- Przechodzimy do katalogu SQL Plus **\sqlplus\admin**
- Edytujemy lub tworzymy plik **Glogin.sql**
- Dodajemy następujący tekst

```
set lines 100
set pages 100
set termout off
col dbname new_value prompt_dbname
select substr(global_name,1,instr(global_name,'.')->1) dbname
from global_name;
set sqlprompt „&&prompt_dbname> „
set termout on
set time on
```

- Herve Deschamps, Useful Prompt in SQL*Plus

Przydatne ustawienia - zmiana domyślnego edytora z notatnik na wordpad

- Zmienić domyślnego edytora w Windows z notatnik na
- Wordpad.exe lub
- vi.exe (dla pracujących pod Unix/Linux) jest przydatne
- Należy skopiować edytora z
Program Files\Windows NT\Accessories do
ORACLE_HOME\BIN
- Należy dodać wiersz do glogin.sql
- define_editor=„wordpad.exe”

Polecenie DEFINE

Polecenie DEFINE bez parametrów pokazuje ustawień zmiennych użytkownika

- edytor
- wersja instancji bazy danych
- identyfikator serwisu (instancji)
- użytkownika
- rola

```
SQL>define a=text
```

```
SQL>define a
```

```
DEFINE A          = "text" (CHAR)
```

UNDEFINE <nazwa_zmiennej> - parametr traci przyswojoną mu wartość poprzez polecenia DEFINE - usuwa zmienne użytkownika wprowadzone poprzez polecenia DEFINE.

Jak zmienić rozmiar i nazwy czcionki w SQL*Plus?

- Jeżeli uruchomimy sqlplus.exe (nie sqlplusw.exe) to wybieramy lewy górny kąt i lewym przyciskiem myszy wybieramy właściwości. Możemy zmienić czcionki, kolory, układ.
- Jeżeli chcemy zmienić czcionki dla sqlplusw (SQL*Plus dla Windows) to musimy zrobić te zmiany w rejestrach Windows.
- Uruchamiamy cmd\regedit32.
- Wybieramy
HKEY_LOCAL_MACHINE \SOFTWARE \ ORACLE \ ORACLE_HOME.
- Tworzymy nowe wpisy w rejestrze:
 - SQLPLUS_FONT i wprowadzamy czcionki, której chcemy używać;
 - SQLPLUS_FONT_SIZE i określamy jaki rozmiar czcionki nam odpowiada.
- Zamykamy regedit i uruchamiamy sqlplusw.exe w celu zobaczenia czy jest efekt wskutek dokonanych przez nas zmian.

Jak zignorować puste linie w SQL*Plus?

- Jeżeli ten parametr nie jest ustawiony i mamy puste linie przy jego wykonaniu otrzymamy błąd.
- W celu uniknięcia błędów, kiedy mamy puste linie w skrypcie należy wprowadzić w SQL*Plus
- Set sqlblanklines on

Wykonywanie bloków PL/SQL

- PL/SQL jest język programowania opracowany i rozwijany przez firmę Oracle, który jest proceduralne rozszerzenie języka SQL. PL/SQL jest używany do pisania przechowywanych w bazie danych procedur, funkcji, pakietów, wyzwalaczy i zaczynając od Oracle8 również typy obiektowe.
- SQL*Plus był na początku jedynym narzędziem używanym do pisaniu kodu PL/SQL, jego weryfikacji i wykonaniu.

Wykonywanie bloków PL/SQL

- W celu wprowadzenia bloków PL/SQL trzeba być w reżimie PL/SQL. SQL*Plus przechodzi w tym stanie jeżeli:
 - W wierszu poleceń SQL*Plus jest wprowadzone jedno z kluczowych słów
DECLARE lub **BEGIN**.
 - Wprowadzone jest polecenie SQL, które jest używane do tworzenia obiektów programowych, na przykład
CREATE PROCEDURE
CREATE PACKAGE
- W celu zakończeniu i wykonaniu bloku PL/SQL należy wprowadzić na nowym wierszu symbol „/”.

Wykonywanie bloków PL/SQL

- W celu wyświetlenia wyników procedury (kodu) bloku anonimowego należy wprowadzić polecenie
SET SERVEROUTPUT ON
- Ta opcja jest włączona do momentu zakończenia pracy z SQL*Plus lub wydania polecenia
SET SERVEROUTPUT OFF
- Przykładowy blok anonimowy (nienazwany) kodu PL/SQL.
Wykonywany jest pakiet DBMS.OUTPUT.
BEGIN
 DBMS.OUTPUT.PUT_LINE('Witam Was');
END;
Po zakończeniu wprowadzania kodu należy wpisać „/” na nowej linii.

Wprowadzenie wartości parametrów w czasie wykonania

- Przy wykonywaniu skryptów SQL możemy podawać wartości parametrów w wierszu poleceń. SQL*Plus podstawia podaną przez nas wartość parametru w zapytaniu. Tak uzyskujemy bardziej inteligentny kod, nie musimy zmieniać zapytania, jak się zmieniają wartości parametrów stosowanych do filtrowania, ograniczenia wyniku zapytania.
- @nazwa_skryptu parametr1, parametr 2.... Parametr n
- Chcemy uzyskać informacji o pracownikach, którzy pracują w departamencie 10
SQL>clear buffer
/wyczyszczamy bufora/
SQL> input
SQL>select ename, sal from scott.emp where deptno=&1
SQL>save d:\oracle\product\10.1.0\WNTTEST\BIN\placa
SQL>@d:\oracle\product\10.1.0\WNTTEST\BIN\placa 10

Wprowadzenie wartości parametrów w czasie wykonania

- SQL>clear buffer
/wyczyszczamy bufora/
SQL> input
select ename, sal from scott.emp where deptno=&1 and sal>&2
- save d:\oracle\product\10.1.0\WNTTEST\BIN\placa2
- SQL>@d:\oracle\product\10.1.0\WNTTEST\BIN\placa2 10 2500
chcemy uzyskać informacje o pracownikach, którzy pracują w 10 departamencie i mają płacą powyżej 2500

Wprowadzenie wartości parametrów w czasie wykonania

- SQL>select &func.(&col.) from &tab;
- Enter value for func: MAX
- Enter value for col: SAL
- Enter value for tab: scott.emp
- old 1: select &func.(&col.) from &tab
- new 1: select MAX(SAL) from scott.emp
- MAX(SAL)
- -----
- 5000
- Uwaga jeżeli zmienna jest przed symbolem bez odstępu należy po nią zastosować „.”

Wprowadzenie wartości parametrów w czasie wykonania

- SQL>Select max(&col), avg(&col), min(&col) from &tab;
- Enter value for col: sal
- Enter value for col: sal
- Enter value for col: sal
- Enter value for tab: scott.emp
- old 1: Select max(&col), avg(&col), min(&col) from &tab
- new 1: Select max(sal), avg(sal), min(sal) from scott.emp

| MAX(SAL) | AVG(SAL) | MIN(SAL) |
|----------|------------|----------|
| 5000 | 2073,21429 | 800 |

Wprowadzenie wartości parametrów w czasie wykonania

- W celu tylko jednorazowego wprowadzania wartości zmiennej, która jest używana kilkakrotnie należy zamiast & zastosować &&
- SQL>Select max(&&col), avg(&&col), min(&&col) from &tab;
- Enter value for col: sal
- Enter value for tab: scott.emp
- old 1: Select max(&&col), avg(&&col), min(&&col) from &tab
- new 1: Select max(sal), avg(sal), min(sal) from scott.emp
- MAX(SAL) AVG(SAL) MIN(SAL)
- ----- ----- -----
- 5000 2073,21429 800
- Uwaga w stosowaniu symbolu „&” w komentarzach!

SQL*Plus i zmienne związane (bind variables)

- Zmienne związane (bind variables) są używane w celu podania wartości dla skryptu SQL w którym nie chcemy żeby optymalizator zmieniał planu wykonania zapytania.
- Zmienne związane (bind variables) są zmienne, które mają typ (number, varchar2 ...) i rozmiar, długość (20..).
- Stosując zmienne związane (bind variables) w skryptach SQL*Plus możemy podawać wartości które utrzymujemy w wyniku wykonania bloku PL/SQL z powrotem do SQL*Plus.

SQL*Plus i zmienne związane (bind variables)

```
var gr1 number;  
var gr2 varchar2(20);  
exec :gr1 :=7369;  
exec :gr2 :='GARRY';  
print gr1;  
print gr2;  
select :gr1, :gr2 from dual;  
select * from scott.emp where empno= :gr1;
```

- SQL*Plus Tips, <http://www.oracleadvice.com/Tips/sqlplus.htm>

Wybór tabeli do eksportu do Excela

- ```
select owner, table_name
from all_tables
where rownum<3;
```
- ```
select *  
from hr.regions;
```

Jak zapisać wyeksportować wyniku zapytania do plik Excel

- Ten skrypt pokazuje jak wyeksportować wyniku zapytania w SQL *Plus do pliku arkusza kalkulacyjnego

```
SET LINESIZE 999 VERIFY OFF FEEDBACK OFF
```

```
SET MARKUP HTML ON ENTMAP ON SPOOL ON PREFORMAT OFF
```

```
DEFINE table_name=&l
```

Enter value for l:**hr.regions**

/tabela **REGIONS** należąca do użytkownika **HR**/

```
SPOOL &table_name..xls
```

```
SELECT * FROM &table_name;
```

```
SPOOL OFF
```

```
SET MARKUP HTML OFF ENTMAP OFF SPOOL OFF PREFORMAT ON
```

```
SET LINESIZE 80 VERIFY ON FEEDBACK ON
```

- Start Excel z SQL*Plus w graficznym trybie (sqlplusw.exe)

```
HOST START excel.exe &table_name..xls
```


Jak zapisać wyeksportować wyniku zapytania do plik Excel

SET MARKUP HTML [ON | OFF] [HEAD text] [BODY text]
[ENTMAP {ON | OFF}] [SPOOL {ON | OFF}]
[PRE[FORMAT] {ON | OFF}]

- podstawa do eksportu do arkusza kalkulacyjnego Excel

Administracja przy pomocy SQL*Plus

- Przy pomocy SQL*Plus (oraz Oracle Enterprise Manager) możemy:
 - startować, uruchamiać bazę danych
STARTUP
 - zatrzymywać bazę danych
SHUTDOWN [NORMAL]
SHUTDOWN IMMEDIATE
SHUTDOWN ABORT
 - zmieniać trybu archiwizacji
SHOW LOGSOURCE
SHOW AUTORECOVERY
 - monitorować użycie pamięci - System Global Area (SGA)
SHOW SGA
 - zmieniać parametry instancji

Polecenie SHOW

- Stosując polecenia SHOW uzyskuje informacje o parametrach bazy danych i zmiennych środowiska
- SHO[W]
ALL
PARAMETERS [parameter_name]
SGA - wymaga uprawnień DBA
SPOO[L]
SQLCODE
USER
LNO
PNO
REL[EASE]
RECYC[LEBIN] [oryginalna_nazwa]
BTI[TLE] / TTI[TLE]

Polecenie SET

Jak zakończyć pracę z SQL*Plus?

W celu zakończenia pracy programu SQL*Plus należy wpisać:

- SQL>exit
- SQL>quit
- zamykanie okna SQL*Plus

Tematy, które nie rozpatrywaliśmy

- Polecenie COMPUTE
- Polecenie BREAK
- Polecenie PAUSE
- Polecenia redagowania w SQL*Plus /Append, Changes, Input, Del .../
oraz polecenie SAVE - zachowywanie zawartości bufora do pliku
- Polecenie REPHEADER, REPFOOTER
- Polecenie COPY - kopiowanie danych z jednej do drugiej bazy danych
- Polecenie SAVE - zachowanie zawartości buffera w pliku
- Ustawień środowiska SQL*Plus
- Zachowanie w pliku zawartości ustawień środowiska SQL*Plus -
STORE SET
- SHOW ALL - wyświetlenie zmiennych środowiska SQL*Plus
- Administracja bazy danych przy użyciu SQL*Plus

Dokumentacja na temat SQL*Plus

<http://www.oracle.com/technology/documentation/database10g.html>

http://otn.oracle.com/h/sql_plus

<http://asktom.oracle.com>

http://otn.oracle.com/pls/db10g/portal.portal_demo3?selected=3#index-SQL

SQL*Plus® Quick Reference Release 10.1 December 2003 Part No. B12171-01

SQL*Plus Tips, <http://www.oracleadvice.com/Tips/sqlplus.htm>

Jonathan Gennick, Oracle SQL*Plus: The Definitive Guide, O'Reilly, ISBN:1-56592-578-5

Bill Pribyl, Steven Feurstein, Oracle PL/SQL, Helion, 2002, s. 40-44

- Joe Grene, Oracle 8 Server, Księga Eksperta, Helion, 2000, s. 526 - 544

Informacja w Internecie na temat SQL*Plus

- <http://www.oraFAQ.com/faqplus.htm>
- <http://www.oreilly.com/catalog/orsqplus/>
- http://otn.oracle.com/pls/db10g/portal.portal_demo3?selected=1
-