

Guia de Consulta Rápida

Oracle 10g PL / SQL

Celso H. Poderoso de Oliveira

Novatec Editora

Copyright © 2005 da Novatec Editora Ltda.

Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/1998. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: RUBENS PRATES

ISBN: 85-7522-067-5

NOVATEC EDITORA LTDA.

Rua Cons. Moreira de Barros 1084 – Conj. 01

02018-012 São Paulo SP – Brasil

Tel.: +55 11 6959-6529

Fax: +55 11 6950-8869

E-mail: novatec@novateceditora.com.br

Site: www.novateceditora.com.br

Introdução	5
Novos recursos da versão 10g	5
Bloco PL/SQL	5
Funcionamento do engine PL/SQL	6
Parâmetros de inicialização do PL/SQL	7
Diferença entre SQL e PL/SQL	8
SQL*Plus	9
SQL*Plus Worksheet	10
iSQL*Plus	11
Comandos de edição do SQL*Plus	11
Configuração do ambiente	12
Comandos do SQL*Plus	13
Estrutura geral da linguagem	17
Delimitadores	17
Literais	18
Comentários	18
Tipos de dados	19
Subtipos	20
Manipulação de datas	21
Operações com datas	21
Compostos, Referência e LOB	22
Subtipos definidos pelo usuário	22
Conversão de dados	22
Seção de Declaração	23
Escopo e visibilidade	24
Expressões e comparações	25
Expressões booleanas	25
Dicas de utilização de expressões booleanas	26
Expressões CASE	27
Expressões regulares	28
Estruturas de controle	29
Controle de fluxo – IF	29
Controle de fluxo – CASE	30
Controle de repetição – LOOP	31
FOR-LOOP	33
Controle de seqüência – GOTO e NULL	34
Exemplo de estruturas de controle	34
Collections e records	37
Uso de collections	37
Manipulação de collections	38
PL/SQL e comandos SQL	39
Associative Array	40
Métodos utilizados em collections	40
Manipulação de erros de collections	41
Operadores de conjunto	42
Melhorando o desempenho	42
Definição de mais dimensões	44
Uso de records	44
Manipulação de records	45
Simplificando código	45
Cursors	46
Cursors explícitos	46
Cursors implícitos	48
Utilização do comando FOR	48
Variáveis do tipo Cursor	49
Subqueries x Cursors explícitos	50
Expressões CURSOR	51
Transações	53
Two Fase Commit	54
Mecanismos de bloqueio	54
Transações autônomas – PRAGMA	54
LOCK TABLE	55
DBMS_LOCK	55
Tratamento de erros	56
Exceções predefinidas	57

Exceções criadas pelo usuário	58
Uso do RAISE_APPLICATION_ERROR	59
Considerações sobre o tratamento de erros	59
Subprogramas	60
Alertas de compilação	60
Procedimentos	61
Comandos relacionados a procedimentos	62
Funções	63
Comandos relacionados a funções	65
Declaração de subprogramas	65
Chamando programas externos	66
Atribuindo direitos de execução	67
Visões relacionadas com subprogramas armazenados	67
Pacotes	68
Especificação do pacote	68
Corpo do pacote	69
Comandos relacionados a pacotes	70
Estados do pacote	71
Objetos	72
Utilizando o Object Type	72
Declaração do Object Type	73
Comparação de objetos	74
Troca de atributos e métodos	75
Exclusão de objetos	75
Método construtor	75
Acessando métodos – notação de ponto	76
Manipulação de objetos	76
Utilização de VALUE	77
Utilização de REF e Deref	78
Regras gerais para utilização de Object Type	78
Triggers	82
Tipos de triggers	82
Triggers DML	82
Triggers Instead of	84
Triggers DDL	85
Triggers de banco de dados	86
Comandos para gerenciamento de triggers	86
Dynamic SQL	87
Quando utilizar o Dynamic SQL	87
Utilizando o Dynamic SQL	88
Consultas com mais de uma linha de retorno	90
Utilização de atributos de cursor	92
Parâmetros de retorno	92
Utilizando BULK DYNAMIC SQL	93
Utilizando Java com PL/SQL	94
Oracle JVM	94
Passos para criação de procedimentos Java no Oracle	95
Importando as classes Java	95
Apagando classes Java	98
Publicando classes Java	99
Parâmetros IN OUT e OUT	101
Ferramenta OJVMJAVA	102
Conexão com bancos de dados usando JDBC	102
Utilização do SQLJ	103
Apêndices	105
Ferramentas	105
PL/SQL Wrapper	106
Pacotes fornecidos com o banco de dados (PL/SQL Built-ins) ..	107
Limites do PL/SQL	112
Palavras reservadas do PL/SQL e SQL	113
Informações adicionais	115
Onde obter mais informações	115
Notação utilizada neste guia	115
Comentários e sugestões	115
Sobre o autor	115

Introdução

PL/SQL é a linguagem utilizada para programar o banco de dados Oracle. É uma extensão do SQL padrão. Desta forma, é possível utilizar comandos SQL, como INSERT, DELETE, UPDATE e SELECT para manipular dados, além de comandos para controlar a lógica dos processos.

A programação em PL/SQL guarda muita semelhança com linguagens procedurais. Uma característica importante é a divisão de um bloco em pequenos blocos de comandos. Desse modo, as unidades básicas (procedimentos e funções) são baseadas em blocos lógicos, e estes, por sua vez, podem conter outros blocos.

O fato de ser chamada procedural pode, para muitos, parecer que é uma linguagem de concepção “antiga”. Na realidade, a linguagem PL/SQL vem sendo aprimorada pela Oracle ao longo do tempo. Possui desde o “mal falado” comando GOTO até a possibilidade de criar e manter objetos, atendendo aos conceitos da Programação Orientada a Objetos. Contudo, cada método do objeto deverá, naturalmente, ser programado de maneira procedural, como em todas as linguagens.

Novos recursos da versão 10g

Nesta nova versão, além dos recursos introduzidos na release 2 do Oracle 9i, como herança de objetos, houve uma significativa melhora no desempenho dos subprogramas executados no 10g. Documentos da Oracle indicam que esta versão executa o mesmo código, sem modificações, cerca de duas vezes mais rápido que a versão anterior e três vezes mais rápido que a versão 8, uma das mais populares, ainda, nas instalações Oracle.

Em termos de linguagem, os avanços que a Oracle implementou nesta nova versão podem ser resumidos em:

- a) Novos tipos de dados do IEEE.
- b) Expressões regulares.
- c) Operações de grupo em tabelas aninhadas.
- d) Inserção de caracteres especiais definidos pelo usuário.
- e) Modificações na sintaxe do comando FORALL.

Bloco PL/SQL

Toda programação PL/SQL está baseada em blocos de comandos. Esses blocos fazem parte dos procedimentos e funções (tratados como subprogramas neste Guia), pacotes, triggers e objetos. É uma estrutura muito simples e é necessário que o programador a tenha sempre em mente. Cada uma das seções será analisada neste Guia.

As seções que compõem um bloco PL/SQL são:

- a) Declaração de variáveis – Cláusula DECLARE.
- b) Execução de comandos e lógica – Cláusula BEGIN.
- c) Exceção ou tratamento de erros – Cláusula EXCEPTION.

A estrutura do bloco PL/SQL é a seguinte:

DECLARE

declarações de variáveis;

...

BEGIN

comandos;

...

EXCEPTION

tratamento de erros;

...

END;

A única seção obrigatória é a de Execução, as demais são opcionais. Conforme dito, dentro de um bloco é possível haver outros. Abaixo do BEGIN, pode-se incluir nova seção de declaração, execução e tratamento de erros.

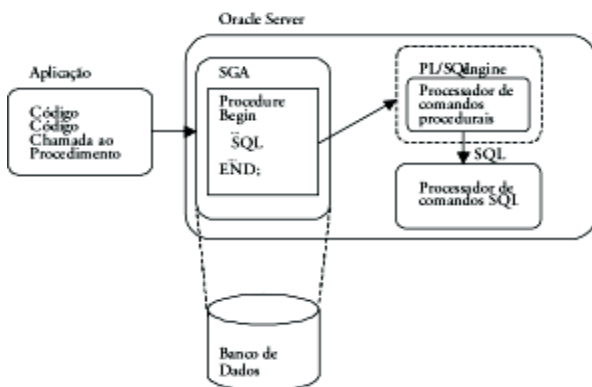
Utilidade do bloco PL/SQL

Utiliza-se o bloco PL/SQL para criar:

- Triggers: são programas que devem ser executados após eventos que ocorrem no banco de dados, como inclusão (INSERT), exclusão (DELETE) e atualização (UPDATE).
- Procedimentos e funções: programas armazenados no banco de dados que são chamados por aplicações, triggers ou outras ferramentas.
- Blocos anônimos: blocos enviados por outros programas, como o próprio SQL*Plus ou Enterprise Manager que são compilados e executados no Oracle Server.
- Pacotes e objetos: armazenam e agrupam funções, procedimentos, cursors e variáveis.

Funcionamento do engine PL/SQL

Os subprogramas (procedimentos e funções) são compilados e armazenados no banco de dados. Quando uma aplicação externa chama um procedimento armazenado, este é requisitado para uma área da SGA (System Global Area – área de memória que o Oracle utiliza no servidor) chamada shared pool (área de memória compartilhada). Por sua vez, cada comando procedural é executado no engine PL/SQL que, conseqüentemente, requisita, sempre que necessário, o processador de comandos SQL. Note que ambos os processadores, tanto o do PL/SQL quanto do SQL, trabalham de maneira integrada, o que garante melhor performance na execução dos blocos. A utilização de subprogramas armazenados melhora o desempenho das aplicações e da rede como um todo. É altamente recomendável que se utilize subprogramas quando se trabalha com Oracle.



Este engine está disponível nas seguintes ferramentas Oracle:

- Oracle Server.
- Oracle Forms (a partir da versão 3).
- SQL*Menu (a partir da versão 5).
- Oracle Reports (a partir da versão 2).
- Oracle Graphics (a partir da versão 2).

Um procedimento armazenado pode ser chamado por um bloco anônimo (não armazenado no banco de dados, mas executado diretamente nele) ou por outro procedimento armazenado. Pode-se passar, também, blocos anônimos para o Oracle, utilizando as seguintes ferramentas:

- Pré-compiladores Oracle.
- Oracle Call Interfaces (OCIs).
- SQL*Plus.
- Server Manager.
- Oracle Enterprise Manager.

Existe uma série de pré-compiladores Oracle para diversas linguagens como C, C++, Cobol etc. Outras ferramentas utilizam as OCIs para se comunicarem com o Oracle, como o PHP.

Parâmetros de inicialização do PL/SQL

Alguns parâmetros do banco de dados são importantes aos desenvolvedores PL/SQL (sublinhado indica o valor default):

Parâmetro	Descrição
PLSQL_CODE_TYPE = { <u>INTERPRETED</u> NATIVE}	Ao indicar NATIVE, o código será compilado no código nativo do computador.
PLSQL_DEBUG = {TRUE <u>FALSE</u> }	Se for indicado TRUE, o código será compilado como interpretado.
PLSQL_OPTIMIZE_LEVEL={0 1 2}	Indica o nível de otimização que será utilizado no código PL/SQL. Zero desabilita o recurso e 2 é o nível mais alto de otimização.
PLSQL_WARNINGS	Será visto em detalhes em Subprogramas (página 60).

Diferença entre SQL e PL/SQL

Como visto, o PL/SQL aceita praticamente todos os comandos SQL. Há apenas uma grande diferença na utilização do comando SELECT, conforme veremos a seguir.

Comando SELECT INTO

A principal diferença na utilização de comandos SQL em blocos PL/SQL está na utilização do comando SELECT.

Sabe-se que o comando SELECT é utilizado para realizar buscas no banco de dados. No PL/SQL é necessário colocar o resultado de qualquer busca em variáveis.

Isso se faz especificando a cláusula INTO após a lista de campos, seguida de tantas variáveis quantos sejam os campos que buscamos na tabela.

Como resultado da busca podemos ter uma, várias ou nenhuma linha. Sempre que utilizamos a cláusula INTO, a expectativa do PL/SQL é que haja uma única linha de retorno.

As duas outras situações deverão ser convenientemente tratadas. Faremos a verificação dessas situações não esperadas pelo PL/SQL será visto no tópico de Tratamento de erros (página 56).

Sintaxe:

SELECT *lista_campos* **INTO** *lista_destino* **FROM** *tabela* ...

Parâmetro	Descrição
<i>lista_campos</i>	Lista de campos que serão selecionadas na tabela.
<i>lista_destino</i>	Lista de variáveis que receberão o conteúdo da lista de campos.
<i>tabela</i>	Tabela base de busca.

Note que o número de elementos da *lista_destino* deve coincidir com o número de elementos da *lista_campos*. Caso contrário, o PL/SQL reportará um erro na tentativa de execução do comando.

Utilização de pseudocolunas

Pseudocolunas são colunas que não fazem parte da tabela, mas que têm características de colunas. Toda vez que for feita uma referência a uma dessas pseudocolunas, elas terão o mesmo comportamento de outras colunas, retornando valores, por exemplo. O que não se pode fazer é manipulá-las com instruções do tipo DDL (criação, alteração etc.) nem alguns comandos DML (INSERT e UPDATE).

No PL/SQL, pode-se utilizar as seguintes pseudocolunas:

Pseudocoluna	Descrição
CURRVAL	Valor corrente de uma seqüência armazenada no banco de dados.
NEXTVAL	Valor do próximo valor da seqüência armazenada no banco de dados.
LEVEL	Nível na estrutura da árvore em bancos de dados.
ROWID	Coluna que especifica a localização da linha (definido pelo Oracle).
ROWNUM	Número da linha selecionada em uma tabela. Não é afetada pela cláusula ORDER BY.

SQL*Plus

Esta ferramenta está disponível na quase-totalidade das instalações Oracle. É um programa que permite executar desde comandos SQL simples até criar subprogramas. Ele envia as informações ao servidor que executa e retorna as informações requisitadas. Desde a versão 9i é possível realizar todas as operações de DBA utilizando esta ferramenta.

Existe tanto em ambiente gráfico (apesar de possuir uma interface muito pobre) e também em ambiente caractere. O funcionamento de ambos é exatamente igual.

Principais funções que podem ser executadas no SQL*Plus:

- a) criar, editar, recuperar e executar comandos SQL;
- b) criar, editar, armazenar, recuperar e executar blocos PL/SQL;
- c) criar relatórios simples;
- d) verificar informações de tabelas;
- e) enviar e receber mensagens de usuário;
- f) administrar o banco de dados Oracle.

Para executar o SQL*Plus no prompt do sistema operacional, você deve digitar:

SQLPLUS

Será solicitado o nome do usuário, senha e string de conexão ao servidor. Existe a possibilidade de você executar o programa e definir um usuário, senha e string no próprio comando. Nesse caso, digite:

SQLPLUS *usuário/senha@string*

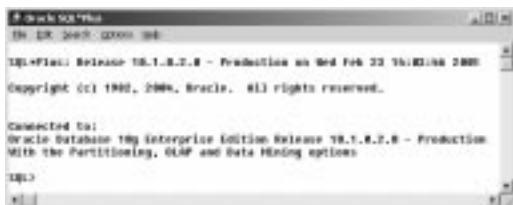
Você poderá executar uma seqüência de comandos armazenados em um arquivo (veja comando @ adiante), bastando especificar o @ e o nome do arquivo. Caso você não forneça a senha, esta será solicitada no momento da execução do comando.

SQLPLUS *usuário/senha@string @arquivo.sql*

Caso você esteja em ambiente gráfico (Windows, por exemplo), localize o programa e execute-o. As mesmas informações serão solicitadas.



Esta é a tela de logon do SQL*Plus. Após informar User Name (usuário), Password (senha) e Host String (conexão com o servidor), pressione OK e uma tela como esta será mostrada:



Caso a sua versão seja diferente da apresentada na figura, poderá haver alguma diferença, mas apenas na versão do Oracle. É importante que, para executar os exemplos do tópico sobre Java, esteja habilitada a função “JServer Release...”.

Um comando SQL será enviado ao servidor e o SQL*Plus aguardará o retorno mostrando o resultado na tela. Um comando SQL só é terminado com ponto-e-vírgula (;) ou barra (/) na última linha de comando. Até que encontre um dos dois, sempre que você teclar Enter será acrescentada uma linha sem encerrar o comando. Caso você deixe em branco alguma linha, o comando ou bloco não será executado, mas estará no buffer do SQL*Plus. Basta, no prompt, colocar a barra (/) e o comando ou o bloco será executado. Você também pode encerrar o modo de edição explicitamente, bastando colocar um ponto (.) em uma nova linha.

Para criar um bloco PL/SQL, digite os comandos um após o outro e no final encerre o bloco com a barra (/). O bloco será encerrado, compilado e, no caso de blocos autônomos, executado imediatamente.

Apenas o último comando ou bloco ficará no buffer do SQL*Plus.

SQL*Plus Worksheet

Esta é uma ferramenta GUI semelhante ao SQL*Plus e oferece algumas facilidades para editar e diferenciar comandos por meio de cores. Facilita a gravação e recuperação dos arquivos gravados.

Ele é instalado com o Oracle Enterprise Manager até a versão 8i – release 8.1.5 – e na instalação de administrador de banco de dados para a release 8.1.6 em diante. O SQL*Plus Worksheet é dividido em duas partes: a parte superior onde você irá digitar e editar os comandos para serem executados e a parte de baixo onde aparecerão os resultados. A partir da versão 10g a Oracle direciona todas as ferramentas para utilização em browser (vide iSQL*Plus a seguir).

A tela de entrada do SQL*Plus Worksheet é esta:



Basicamente são solicitadas as mesmas informações quando você entra no SQL*Plus. A diferença é que no Worksheet é possível conectar-se diretamente ao ambiente do Oracle Management Server. Para o uso comum, basta utilizar a conexão direta ao banco de dados, fornecendo usuário, senha e serviço (que é a mesma string de conexão informada no host string do SQL*Plus).

Você poderá gravar os comandos utilizando o menu *Arquivo > Salvar Entrada Como*, recuperar arquivo gravado no sistema operacional com o menu *Arquivo > Abrir* e gravar o resultado da execução (a parte de baixo da janela) com *Arquivo > Salvar Saída Como*.



A qualquer instante, você pode trocar o usuário utilizando o menu *Arquivo > Alterar Conexão de Banco de Dados* ou pressionar o botão no formato de uma tomada à esquerda.

Para executar os comandos que você digitar, pressione o botão no formato de raio à esquerda, utilizando o menu *Folha de Trabalho > Executar* ou ainda pressionando a tecla F5. Para ver a lista dos comandos executados no Worksheet, basta *Folha de Trabalho > Histórico de Comandos*, pressionando o botão no formato de formulários ou pressionando as teclas Control + H.

iSQL*Plus

Todo ambiente de administração do banco de dados Oracle está migrando para utilização de browser. Com o SQL*Plus não poderia ser diferente. Desde a versão 9i o iSQL*Plus é instalado automaticamente. A porta onde é instalado varia de acordo com a versão do Oracle. Na versão 10g a porta padrão é a 5560. Um exemplo de conexão é: <http://localhost:5560/isqlplus>. No Windows o serviço OracleOraDb10g_home1iSQL*Plus deve estar ativo para que o iSQL*Plus funcione corretamente.

Comandos de edição do SQL*Plus

A tabela a seguir mostra os comandos de edição que podem ser utilizados no SQL*Plus:

Comando	Abreviação	Descrição
APPEND <i>texto</i>	A <i>texto</i>	Adiciona texto no final da linha atual.
CHANGE / <i>old/new</i> C/ <i>old/new</i>		Troca a cadeia de caracteres <i>old</i> pela <i>new</i> .
CHANGE / <i>texto</i> C / <i>texto</i>		Apaga o texto da linha.
CLEAR BUFFER	CL BUFF	Apaga todas as linhas.
DEL	DEL	Apaga a linha atual.
DEL <i>n</i>	DEL <i>n</i>	Apaga a linha <i>n</i> .
DEL *	DEL *	Apaga e linha atual.
DEL <i>n</i> *	DEL <i>n</i> *	Apaga desde a linha <i>n</i> até a linha atual.
DEL LAST	DEL LAST	Apaga a última linha.
DEL <i>m n</i>	DEL <i>m n</i>	Apaga desde a linha <i>m</i> até a linha <i>n</i> .
DEL * <i>n</i>	DEL * <i>n</i>	Apaga desde a linha atual até a linha <i>n</i> .
INPUT	I	Acrescenta uma linha a partir da posição atual.
INPUT <i>texto</i>	I <i>texto</i>	Acrescenta uma linha com o texto especificado.
LIST	L	Mostra o conteúdo do buffer.
LIST *	L *	Mostra a linha atual.
LIST <i>n</i>	L <i>n</i>	Mostra a linha <i>n</i> .
LIST <i>n</i> *	L <i>n</i> *	Mostra desde a linha <i>n</i> até a linha atual.
LIST LAST	L LAST	Mostra a última linha.
LIST <i>m n</i>	L <i>m n</i>	Mostra o intervalo de linhas entre <i>m</i> e <i>n</i> .
LIST * <i>n</i>	L * <i>n</i>	Mostra desde a linha atual até a linha <i>n</i> .

Configuração do ambiente

Alguns comandos são utilizados para configurar o ambiente do SQL*Plus. A lista a seguir não contempla todas as configurações possíveis do ambiente, mas aquelas que são normalmente utilizada pelos programadores PL/SQL.

SET VERIFY

Sempre que trabalhar com variáveis do SQL*Plus, o padrão é mostrar o valor anterior e o novo valor da variável. Você pode modificar este padrão com este comando.

SET VERIFY [ON|OFF]

Parâmetro	Descrição
ON	Ativa a exibição dos valores anterior e posterior da variável.
OFF	Desativa a exibição dos valores.

SET SERVEROUTPUT

Habilita a saída do pacote DBMS_OUTPUT para o console da ferramenta.

SET SERVEROUT[PUT] {ON | OFF} [SIZE *n*] [FOR[MAT] {WRA[PPED] | WOR[D_WRAPPED] | TRU[NCATED]]]

Parâmetro	Descrição
ON	Ativa a saída de texto no console.
OFF	Desativa a saída de texto no console.
SIZE	Indica o tamanho máximo do buffer de texto.
FORMAT	Indica a forma como o texto será apresentado no console.

SET LINESIZE

Indica o tamanho da linha da ferramenta. O padrão no *i*SQL*Plus é de 150 caracteres e no SQL*Plus é de 80 caracteres.

SET LIN[ESIZE] {80 | *n*}

SET PAGESIZE

Indica a quantidade de linhas que será mostrada no console antes que se inicie um novo cabeçalho para as colunas.

SET PAGES[IZE] {14 | *n*}

SET TIMING

Habilita a visualização das estatísticas baseadas em tempo de execução (veja comando TIMING na página 14).

SET TIMI[NG] {ON|OFF}

Comandos do SQL*Plus

Você deve notar a diferença existente entre os comandos do PL/SQL e dos comandos do SQL*Plus. Não esqueça que o SQL*Plus é uma ferramenta de acesso ao banco de dados. Por este motivo, há diversos comandos utilizados para realizar as funções internas da ferramenta que não funcionam necessariamente no PL/SQL. Não há pretensão neste Guia de indicar todos os comandos do SQL*Plus, mas apenas aqueles que são importantes para que o programador PL/SQL possa extrair o necessário para programar. Para maiores informações sobre o SQL*Plus, consulte o manual SQL*Plus Quick Reference disponível no site da Oracle.

EXECUTE

Executa blocos PL/SQL no SQL*Plus.

Sintaxe:

EXEC[UTE] *comando*

Exemplos

```
EXECUTE CriaMusica( 1, 'Roda Viva' )
```

```
EXECUTE :nValor := TotalVendas( SYSDATE )
```

```
EXECUTE :n := 1
```

O primeiro exemplo executa um procedimento armazenado chamado CriaMusica, passando os parâmetros necessários. O segundo executa uma função armazenada colocando o resultado na variável *bind* (vinculada) nValor. O terceiro apenas atribui um valor a uma variável *bind*. Nos dois últimos exemplos, você deve criar a variável utilizando o comando VAR.

SHOW

Mostra o valor de uma variável de sistema ou do ambiente SQL*Plus.

Sintaxe:

SHOW *opção*

Opção	Descrição
<i>variável de sistema</i>	Qualquer variável que seja definida pelo comando SET.
ALL	Lista todos os parâmetros do comando SHOW. Executando esta opção você poderá ver as variáveis de sistema disponíveis.
ERR[ORS]	Mostra os erros de compilação de procedimentos, funções, pacotes, triggers, views e collections. É possível especificar qual função ou procedimento com as palavras chave: FUNCTION, PROCEDURE, PACKAGE, PACKAGE BODY, TRIGGER, VIEW, TYPE e TYPE BODY seguido do nome do objeto.
PARAMETERS	Mostra o valor de um ou mais parâmetros de inicialização do banco de dados. Você pode especificar o parâmetro que quer ver. Caso não especifique todos serão mostrados.
SGA	Mostra informações relacionadas a System Global Area do Oracle.
SQLCODE	Retorna o último código de erro ocorrido no SQL*Plus.
USER	Retorna o usuário conectado no banco de dados pelo SQL*Plus.

Exemplos

```
SHOW ERRORS PROCEDURE CriaMusica
SHOW SGA
SHOW PARAMETERS service_names
```

TIMING

Registra o tempo decorrido entre dois eventos no banco de dados. Possibilita fornecer um label para o registro do tempo ou listar todos os tempos calculados.

Sintaxe:

TIMING] [START *texto* | SHOW | STOP]

Opção	Descrição
START <i>texto</i>	Inicializa a contagem do tempo e atribui um nome à contagem.
SHOW	Mostra o nome e os dados da contagem atual.
STOP	Mostra o nome e os dados da contagem atual e apaga a contagem. Se houver mais de um TIMING ativo o mais recente é que passa a ser o atual.

Exemplos

```
TIMING START tempo_proc
TIMING SHOW
TIMING STOP
```

SAVE

Grava o comando ou bloco PL/SQL do buffer para um arquivo do sistema operacional. Se for especificado APPEND, o comando atual será acrescentado no final do arquivo existente.

Sintaxe:

SAVE *arquivo* [APPEND]

Exemplo

```
SQL>SAVE MeuBloco.sql
```

EDIT

Chama um editor de texto para criar e manter os comandos SQL e blocos PL/SQL.

Sintaxe:

EDIT *arquivo*

Exemplo

```
SQL>EDIT MeuBloco.sql
```

GET

Abre um arquivo de comandos gravados no sistema operacional. O conjunto de linhas será trazido para o buffer do SQL*Plus.

Sintaxe:

GET *arquivo*

Exemplo

```
SQL>GET MeuBloco.sql
```

START / @

Este comando abre e executa um arquivo armazenado no sistema operacional.

Sintaxe:

START *arquivo*

Exemplos

SQL>START MeuBloco.sql

SQL>@MeuBloco.sql

Variáveis e Variáveis de ligação

Pode-se utilizar variáveis em blocos autônomos apenas especificando o caractere & antes da variável. Sempre que o Oracle identifica uma variável não-definida, é solicitado ao usuário que informe o conteúdo da variável. Se forem utilizados os caracteres &&, será requisitado apenas uma vez o conteúdo da variável. A partir daí, o mesmo valor será utilizado em todas as chamadas à variável.

VARIABLE

O comando VARIABLE define uma variável de ambiente no SQL*Plus que pode ser utilizada como parâmetro de saída para subprogramas ou retorno de função executada diretamente na ferramenta. É muito utilizado para realizar a ligação entre o servidor e o cliente (SQL*Plus). Quando for utilizá-la como variável de ligação em blocos PL/SQL ou em subprogramas, deve estar precedida por dois pontos (:).

Sintaxe:

VAR[**IABLE**] [*variável* [**NUMBER** | **CHAR** | **CHAR** (*n*) | **VARCHAR2** (*n*) | **NCHAR** | **NCHAR** (*n*) | **NVARCHAR2** (*n*) | **CLOB** | **NCLOB** | **REFCURSOR**]]

Exemplo

VARIABLE nValor NUMBER

DEFINE

Especifica um valor inicial para a variável. Ao definir um valor a uma variável não haverá necessidade de o usuário informá-lo.

Sintaxe:

DEFINE *variável* = *valor*;

Parâmetro

Descrição

variável

Nome da variável a que está sendo atribuído valor.

valor

Conteúdo da variável.

Exemplo

DEFINE nCdAutor = 2

DEFINE cNmAutor = 'Chico Buarque'

BEGIN

INSERT INTO ATORES (CDATOR, NMATOR) VALUES (&nCdAutor,
'&cNmAutor');

END;

/