

# ORACLE PL/SQL

## Básico e Intermediário

Autor: Rogério Laureano Martins

<b>Introdução.....</b>	<b>6</b>
<b>Banco de dados.....</b>	<b>7</b>
<b>Linguagem SQL.....</b>	<b>9</b>
DDL - Linguagem de Definição de Dados.....	9
Comandos DDL aplicados ao ORACLE.....	9
DQL - Linguagem de Consulta de Dados.....	16
Comando DQL (Select) aplicados ao ORACLE.....	16
Estudo 1 do comando SELECT.....	16
Estudo 2 do comando SELECT:.....	17
Estudo 3 do comando SELECT:.....	18
Estudo 4 do comando SELECT:.....	19
Estudo 5 do comando SELECT:.....	20
Estudo 6 do comando SELECT:.....	21
Estudo 7 do comando SELECT:.....	22
Estudo 8 - Abaixo exemplos de operações numéricas possíveis para o comando “Select” no ORACLE:.....	23
Estudo 9 – Abaixo operações que podem ser feitas com caracteres à partir do comando “SELECT” no ORACLE:.....	28
Estudo 10 – Abaixo exemplos de operações com tempo que podem ser feitos à partir do comando “select” no ORACLE:.....	31
Estudo 11 – Abaixo exemplos de transformação de dados de números para texto:.....	35
Estudo 12 – Abaixo exemplo de transformação de texto para número.....	37
Estudo 13 – Abaixo exemplo de transformação de texto para data.....	39
Estudo 14 – Abaixo exemplos de transformações de data para texto.....	40
Estudo 15 – Abaixo exemplos de transformações de valores data para número.....	43
Estudo 16 – Abaixo exemplos de comandos no “SELECT” para trazer valores condicionais.....	45
Estudo 17 – Abaixo exemplos de comandos no “SELECT” com consulta à tabelas de dados.....	50
Estudo 18 – Abaixo exemplos de comandos no “SELECT” com cláusula where.....	52
Estudo 19 – Abaixo exemplos de comandos no “SELECT” com like, escape, between, IS NULL.....	53
Estudo 20 – Abaixo exemplos de comandos no “SELECT” com join.....	58

Estudo 21 – Abaixo exemplos de comandos no “SELECT” com sub-query, in e exists.....	60
Estudo 22 – Abaixo exemplos de comandos no “SELECT” com union, minus e intersect.....	63
Estudo 23 – Abaixo exemplos de comandos no “SELECT” com funções de agrupamento (count, max, min, avg, sum).....	65
Estudo 24 – Abaixo exemplos de comandos no “SELECT” com rank.....	68
Estudo 25 – Abaixo exemplos de comandos no “SELECT” com with.....	75
Estudo 26 – Análise de performance e tuning de query.....	77
DML - Linguagem de Manipulação de Dados.....	81
DCL - Linguagem de Controle de Dados.....	86
DTL - Linguagem de Transação de Dados.....	90
<b>Programação PL/SQL.....</b>	<b>92</b>
Exibição de mensagens e textos.....	92
Entrada de informação por parte do operador.....	93
Variáveis.....	93
OPERAÇÕES MATEMÁTICAS.....	94
OPERAÇÕES COM TEXTO.....	100
OPERAÇÕES COM TEMPO.....	106
TRANSFORMAÇÕES DE NÚMERO PARA TEXTO.....	115
TRANSFORMAÇÕES DE TEXTO PARA NÚMERO.....	120
TRANSFORMAÇÕES DE TEXTO PARA DATA.....	123
TRANSFORMAÇÕES DE DATA PARA TEXTO.....	125
TRANSFORMAÇÕES DE DATA PARA NÚMERO.....	132
CONDIÇÕES.....	137
LOOPING.....	149
ARRAY.....	151
PL/SQL com comandos SQL.....	157
CURSORES.....	158
SELECT INTO TYPE.....	160
SELECT BULK COLLECT INTO.....	161
EXISTS.....	162
COUNT.....	163
LIMIT.....	163
FIRST e LAST.....	164
PRIOR e NEXT.....	164
EXTEND.....	166

TRIM.....	166
DELETE com Array.....	166
INSERT.....	167
Insert com campo array.....	167
UPDATE com array.....	168
UPDATE em campo array.....	168
DELETE.....	169
INSERT COM BULK COLLECT.....	169
UPDATE COM BULK COLLECT.....	170
DELETE COM BULK COLLECT.....	171
SAVE EXCEPTIONS .....	172
SCRIPTS DINÂMICOS (EXECUTE IMMEDIATE).....	173
SAIDA.....	173
ENTRADA .....	174
VARIÁVEIS.....	175
ESTRUTURA DECLARE, BEGIN, END DINÂMICOS .....	176
DDL DINÂMICOS.....	176
DML DINÂMICOS .....	179
DCL DINÂMICOS .....	180
BULK COLLECT.....	181
CURSOR DE STRING.....	182
DBMS_JOB e DBMS_IJOB.....	183
SUBMIT.....	183
REMOVE.....	184
LEITURA DE ARQUIVO TEXTO.....	185
ESCRITURA EM ARQUIVO TEXTO.....	186
LEITURA DE XML.....	187
ESCRITURA EM XML.....	188
VIEWS .....	189
VIEWS MATERIALIZADAS.....	190
SEQUENCE.....	192
SYNONYM.....	193
TYPE.....	193
PROCEDURES .....	193
FUNCTIONS.....	194
PACKAGES.....	195

TRIGGERS.....	196
<b>Considerações finais.....</b>	<b>200</b>

## **Introdução**

Quando eu estava cursando a faculdade de processamento de dados, minha professora de Comunicação e Expressão propôs que discutíssemos idéias contidas em um artigo sobre tecnologia. O autor daquele artigo sugeria que a tecnologia desperdiçava o talento humano, pois nivelava a capacidade de execução das tarefas.

Desde que inventaram o trator escavadeira, seria indiferente se o escavador fosse forte como um touro ou fraco como uma pomba o tamanho do buraco seria o mesmo, assim como o tempo dispendido para escavá-lo.

Existe uma grande falha nesses argumentos. Realmente o trator escavadeira nivela a capacidade física dos escavadores. No entanto, a habilidade e conhecimento de como operar este trator diferenciará o mais competente.

Hoje em dia, grande parte dos trabalhadores executa tarefas utilizando computadores. Tenho visto muitos profissionais se destacarem pela perícia com que manuseiam este recurso tão importante no atual cenário mundial.

Caro leitor, se você trabalha em escritórios, utiliza computadores e deseja se destacar entre seus colegas de trabalho, este livro é ideal para alcançar tais objetivos.

Nos capítulos que seguem, você aprenderá utilizar de maneira surpreendente os recursos do banco de dado ORACLE. Estará apto a trabalhar com esta poderosa ferramenta que está em boa parte das empresas.

## **Banco de dados**

Banco de dados é a solução que a informática criou para armazenar informações de maneira estruturada e organizada. Os bancos de dados são compostos de tabelas de dados.

Tabelas de dados são agrupamento de informações referente a uma entidade (pessoa, local, documentos, controles etc.)

Exemplo: Em uma tabela de clientes, teremos as informações de todos os clientes. Em uma tabela de endereços, teremos as informações de todos os endereços relevantes.

A tabela de dados por sua vez é composta de registros. Nos exemplos acima, cada grupo de informações referente a um único cliente ou a um único endereço é chamado de registro.

Por sua vez, os registros são compostos de dados. O dado é a menor parcela de informação em um banco de dados.

Exemplo: O dado do campo chamado “Bairro” é uma parcela da informação de um endereço.

Quanto à estrutura, as tabelas de dados são organizadas em campos. No exemplo acima citamos um campo chamado “Bairro”, onde pode ser armazenado um bairro de um endereço.

Um campo possui atributos que qualificam e criam regras de uso das informações. São eles:

Nome → Define um identificador técnico em que o programador irá referenciá-lo

Descrição → Atributo que descreve de maneira funcional o uso do campo. Ou seja, documenta o para que serve o dado que será colocado no campo.

Tipo → Tipo de dado que pode estar contido no campo. Cada tecnologia utiliza uma gama de tipos de dados. Porém, basicamente os tipos são: Caracter, que permite letras, números ou qualquer caracter como “\*”, “/” ou “,” etc.; Número, que permite valores numéricos e Tempo, que permite Data, Hora, minuto, segundo.

Tamanho → Tamanho do dado suportado pelo campo.

Obrigatoriedade → Atributo de um campo que indica se para que um registro exista é ou não obrigatório o preenchimento do dado. Ou seja, caso haja uma tentativa de criar um registro com um dado nulo em um campo obrigatório, o sistema retornará um erro e não será criado o registro.

Valor Padrão → Atributo de um campo que assume um valor padrão ao dado caso o dado não seja informado na criação do registro.

Exemplo de organização de tabelas:

<i>TabClientes</i>				
<i>Nome</i>	<i>Descrição</i>	<i>Tipo</i>	<i>Tam</i>	<i>Requerido S/N</i>
CodCliente	Identificador interno do cliente	Numéro	10	S
NomCliente	Nome do cliente	Caracter	50	S
NomPai	Nome do pai do cliente	Caracter	50	N
NomMae	Nome da mãe do cliente	Caracter	50	S
DtaNasc	Data de nascimento do cliente	Data		S
RG	RG do cliente	Caracter	40	N
CPF	CPF do cliente	Caracter	40	N

Neste exemplo, poderíamos ter a seguinte tabela:

<i>CodCliente</i>	<i>NomCliente</i>	<i>NomPai</i>	<i>NomMae</i>	<i>DataNasc</i>	<i>RG</i>	<i>CPF</i>
1	João	Pedro	Laura	09/06/1976	1234	54352
2	José	Geraldo	Helena	05/02/1983	3421	32345
3	Maria	Antônio	Neusa	07/01/1994	2345	54673

## Linguagem SQL

A linguagem de programação mais utilizada em banco de dados é a linguagem SQL. A sigla SQL (Structured Query Language) significa (Linguagem de Consulta Estruturada). Ela é dividida em 5 grupos de comandos.

### *DDL - Linguagem de Definição de Dados*

O grupo DDL (Data Definition Language - Linguagem de Definição de Dados) permite criar, alterar ou excluir tabelas (ou qualquer outro tipo de componente) no banco de dados. Ou seja, enquanto a DML manipula registros de uma tabela, a DDL manipula a existência ou não da tabela inteira.

<b>comandos SQL</b>	<b>descrição do comando</b>
CREATE	Cria um novo objeto de banco de dados
ALTER	Altera um objeto pré-existent no banco de dados
DROP	Exclui um objeto pré-existent no banco de dados

### **Comandos DDL aplicados ao ORACLE**

Por questão de continuidade, começaremos a criar exercícios de linguagens DDL. Ou seja, os primeiros exercícios serão a criação de tabelas.

Para criar uma tabela em banco de dados pode ser usada a sintaxe abaixo:

```
CREATE TABLE <nome_da_tabela>
(
[<nome_campo> <tipo_do_campo>(<tamanho_do_campo>) ...]
)
```

Assim para criar a tabela “TabClientes” seria o seguinte:

```
CREATE TABLE TABCLIENTES
(CODCLIENTE NUMBER(10),
 NOMCLIENTE VARCHAR2(50),
 NOMPAI VARCHAR2(50),
 NOMMAE VARCHAR2(50),
 DTANASC DATE,
 RG VARCHAR2(40),
 CPF VARCHAR2(40));
```

Tradicionalmente, as instalações do ORACLE criavam algumas tabelas com o usuário chamado “SCOTT”. As tabelas eram e EMP, DEPT, BONUS, SALGRADE e DUMMY. Como exemplos, segue os scripts de criação destas tabelas:

```
-- Tabela de funcionários
CREATE TABLE EMP
(EMPNO NUMBER(4) NOT NULL,
 ENAME VARCHAR2(10),
 JOB VARCHAR2(9),
 MGR NUMBER(4),
 HIREDATE DATE,
 SAL NUMBER(7, 2),
 COMM NUMBER(7, 2),
 DEPTNO NUMBER(2));

-- Tabela de departamentos
CREATE TABLE DEPT
(DEPTNO NUMBER(2),
 DNAME VARCHAR2(14),
 LOC VARCHAR2(13) );
```

```

-- Tabela de bônus
CREATE TABLE BONUS
    (ENAME VARCHAR2(10),
     JOB   VARCHAR2(9),
     SAL   NUMBER,
     COMM  NUMBER);

-- Tabela de range de salários
CREATE TABLE SALGRADE
    (GRADE NUMBER,
     LOSAL NUMBER,
     HISAL NUMBER);

-- Tabela auxiliar
CREATE TABLE DUMMY
    (DUMMY NUMBER);

```

Os tipos de dados são classificados em 5 grupos:

Escalares → Tipo de dados mais básicos, com valores numéricos, caracteres e de data.

Compostas → Tipo de dados que representam uma estrutura de dados.

LOB → Tipo especial de dados “caracteres” que permite uma grande quantidade de informações

Referência → Tipo de dados que referencia cursores

Objeto → Tipo de dados que referencia objetos do banco

### Grupo escalares

Por sua vez, o grupo escalar é subdividido em 6 famílias:

Família numérica → Tipo de dados em que em seu conteúdo armazena informações de números

Família caracteres → Tipo de dados em que em seu conteúdo armazena informações de texto

Família data → Tipo de dados em que em seu conteúdo armazena informações de datas, horas e intervalos de tempo

Família de ROWID → Tipo de dados em que em seu conteúdo armazena informações de endereço de um registro

Família booleana → Tipo de dados em que em seu conteúdo armazena informações booleanas (Verdadeiro ou Falso)

Família Raw → Tipo de dados em que em seu conteúdo armazena valores Raw. Será detalhado mais adiante.

Tipos escalares		
Família numerica:	Família de caracteres:	Família de data/intervalo
BINARY_INTEGER DEC	CHAR	DATE
DECIMAL	CHARACTER	INTERVAL DAY TO SECOND <sup>5</sup>
DOUBLE PRECISION	LONG	INTERVAL YEAR TO MONTH <sup>3</sup>
FLOAT	NCHAR <sup>1</sup>	TIMESTAMP <sup>3</sup>
INT INTEGER	NVARCHAR2 <sup>1</sup>	TIMESTAMP WITH TIME ZONE <sup>5</sup>
NATURAL	STRING	TIMESTAMP WITH LOCAL TIME ZONE <sup>5</sup>
NATURALN	VARCHAR	
	VARCHAR2	
		Família booleana:
NUMBER	Família de Rowid:	BOOLEAN
NUMERIC	ROWID	
PLS_INTEGER	UROWID <sup>2</sup>	
POSITIVE	Família de confiança:	Família de Raw:
POSITIVEN	MLSLABEL	RAW LONG RAW
REAL		
SIGNTYPE		
SMALLINT		

TIPOS COMPOSTOS	TIPOS LOB	TIPOS DE REFERENCIA	TIPOS DE OBJETO
RECORD NESTED	BFILE <sup>1</sup>	REF CURSOR REF	<i>object type</i> <sup>4</sup>
TABLE INDEX-BY	LOB <sup>1</sup>	<i>object type</i> <sup>4</sup>	SYS.ANYTYPE <sup>3</sup>
TABLE VARRAY <sup>1</sup>	CLOB <sup>1</sup>		SYS.ANYDATA <sup>3</sup>
	NLOB <sup>1</sup>		SYS.ANYDATASET <sup>3</sup>

<sup>1</sup>Esse tipo está disponível no Oracle8 e superior <sup>2</sup>Esse tipo está disponível no Oracle8i e superior <sup>3</sup>Esse tipo está disponível no Oracle9i e superior

Para eliminar uma tabela criada, utilizamos o comando DROP. No exemplo abaixo, o script da eliminação das tabelas do usuário SCOTT:

```
DROP TABLE EMP;  
DROP TABLE DEPT;  
DROP TABLE BONUS;  
DROP TABLE SALGRADE;  
DROP TABLE DUMMY;
```

Para alterar uma tabela pré-existent em banco de dados pode ser usada a sintaxe abaixo:

### **Adicionar uma coluna a uma tabela pré-existente**

```
ALTER TABLE <TABELA>  
  ADD <nome da coluna> <definição da coluna>;
```

Exemplos:

```
ALTER TABLE filial  
  ADD desc_filial varchar2(45);  
  
ALTER TABLE filial  
  ADD municipio varchar2(40) DEFAULT 'São Paulo';
```

### **Adicionar mais de uma coluna em uma tabela pré-existente**

Para adicionar em um único comando mais de uma coluna, deve ser usada a sintaxe abaixo:

```
ALTER TABLE <TABELA>  
  ADD (<COLUNA 1> <DEFINIÇÃO COLUNA>,  
      <COLUNA 2> <DEFINIÇÃO COLUNA>);
```

```
...  
<COLUNA N> <DEFINIÇÃO COLUNA>,);
```

### Exemplo:

```
ALTER TABLE filial  
  ADD (desc_filial varchar2(45),  
       municipio varchar2(40) DEFAULT 'São Paulo');
```

### Modificar coluna em uma tabela

Para alterar as definições de uma coluna pré-existente, deve ser utilizada a seguinte sintaxe:

```
ALTER TABLE <TABELA>  
  MODIFY <COLUNA> <DEFINIÇÕES>;
```

### Exemplo:

```
ALTER TABLE filial  
  MODIFY desc_filial varchar2(100) NOT NULL;
```

Para modificar mais de uma coluna pré-existente em um único comando, deve ser usada a seguinte sintaxe:

```
ALTER TABLE <TABELA>  
  MODIFY (<COLUNA 1> <DEFINIÇÕES>,  
         <COLUNA 2> <DEFINIÇÕES>,  
         ...  
         <COLUNA N> <DEFINIÇÕES>);
```

### Exemplo:

```
ALTER TABLE filial  
  MODIFY (desc_filial varchar2(100) NOT NULL,  
         municipio varchar2(75) DEFAULT 'Seattle' NOT NULL);
```

## **Eliminar coluna dentro de uma tabela pré-existente**

A sintaxe para eliminar uma coluna em uma tabela é a seguinte:

```
ALTER TABLE <TABELA>  
    DROP COLUMN <COLUNA>;
```

Exemplo:

```
ALTER TABLE filial  
    DROP COLUMN desc_filial;
```

## **Renomear coluna dentro de uma tabela pré-existente**

A sintaxe para muda o nome de uma coluna dentro de uma tabela pré-existente é a seguinte:

```
ALTER TABLE <TABELA>  
    RENAME COLUMN <NOME ANTIGO> TO <NOME NOVO>;
```

Exemplo:

```
ALTER TABLE filial  
    RENAME COLUMN desc_filial TO dfilial;
```

## **Mudar o nome da tabela**

A sintaxe para mudar o nome da tabela é a seguinte:

```
ALTER TABLE <TABELA>  
    RENAME TO <NOVO NOME DA TABELA>;
```

Exemplo:

```
ALTER TABLE filial  
  RENAME TO filiais;
```

## ***DQL - Linguagem de Consulta de Dados***

O grupo DQL é a parte da SQL mais utilizada. O comando SELECT permite ao usuário construir uma consulta.

<b>comandos SQL</b>	<b>descrição do comando</b>
SELECT	Permite buscar informações nas tabelas dos banco de dados

### **Comando DQL (Select) aplicados ao ORACLE**

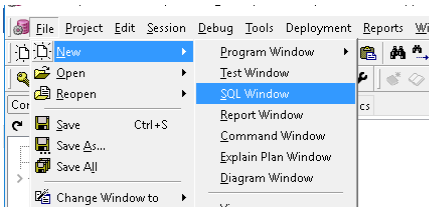
A sintaxe do comando select é a seguinte:

```
SELECT [DISTINCT] {*, coluna [pseudônimo], ... }  
FROM <tabelas>  
WHERE <condições>  
GROUP BY <colunas>  
HAVING <condições>  
ORDER BY {coluna, expr} [ASC|DESC];
```

A tabela virtual “dual” permite que façamos o comando select sem ter uma tabela específica. Siga os seguintes passos.

### **Estudo 1 do comando SELECT**

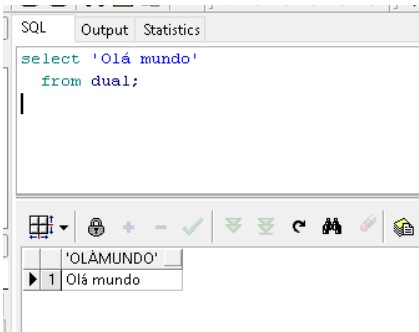
- 1 – Execute o programa PL/SQL Developer e log
- 2 – Clique no menu “File → New → SQL Window”



3 – Na janela de comando digite o seguinte:

```
select 'Olá mundo'  
from dual;
```

4 – Pressione “F8” e o Sistema exibirá a mensagem conforme abaixo:

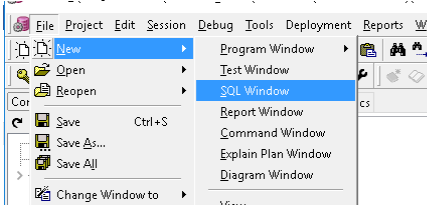


Explicações → No exemplo acima, fica demonstrada a possibilidade de tratamento de valores do tipo caracteres. Estes valores são normalmente utilizados para uma conter informações como mensagens para o usuário, descritivos como nome, endereço etc.

### **Estudo 2 do comando SELECT:**

É possível alterar o cabeçalho do retorno da query colocando um alias.

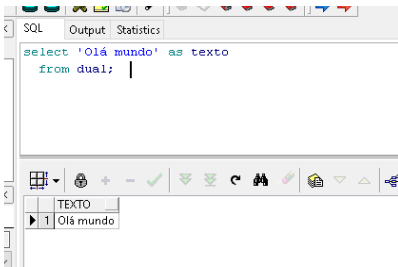
- 1 – Execute o programa PL/SQL Developer e log
- 2 – Clique no menu “File → New → SQL Window”



- 3 – Na janela de comando digite o seguinte:

```
select 'Olá mundo' as texto  
from dual;
```

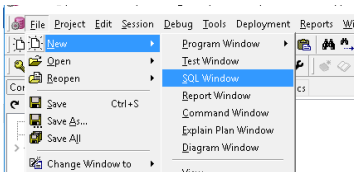
- 4 – Pressione “F8” e o Sistema exibirá a mensagem conforme abaixo:



### **Estudo 3 do comando SELECT:**

É possível retornar duas colunas por vez.

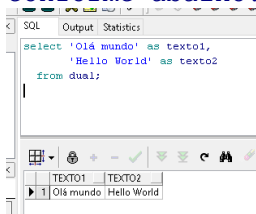
- 1 – Execute o programa PL/SQL Developer e log
- 2 – Clique no menu “File → New → SQL Window”



3 – Na janela de comando digite o seguinte:

```
select 'Olá mundo' as texto1,  
       'Hello World' as texto2  
from dual;
```

4 – Pressione “F8” e o sistema exibirá a mensagem conforme abaixo:



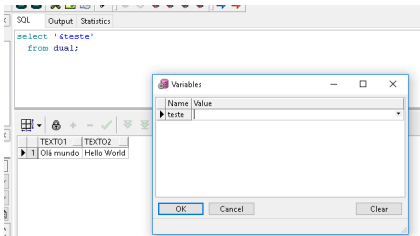
#### **Estudo 4 do comando SELECT:**

É possível receber informações para retornar na query.

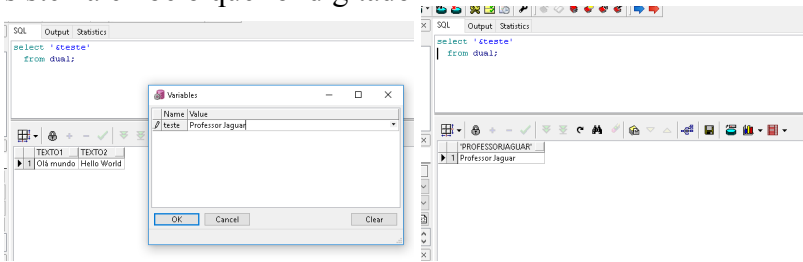
1 – Faça os procedimentos de exibição da janela conforme estudos anteriores e digite a query:

```
select '&teste'  
from dual;
```

2 – Pressione “F8” e o sistema irá pedir o parâmetro “teste” conforme abaixo:



3 – Informe e clique em “ok” ou pressione o “Enter” que o sistema exibe o que foi digitado



### Estudo 5 do comando SELECT:

Outro exemplo de inserção de parâmetros:

1 – Faça os procedimentos de exibição da janela conforme estudos anteriores e digite a query:

```
select '&teste' as texto
from dual;
```

2 – Pressione “F8” e o sistema irá pedir o parâmetro “teste” conforme abaixo: