



# PL/SQL ORACLE

## Exercícios

Prof. Lúcio Camilo  
Email: [luciocamilo@gmail.com](mailto:luciocamilo@gmail.com)  
<http://www.itnerante.com.br/profile/LucioCamilo>



# Lúcio Camilo

- Resumo – CV
- Analista de Sistemas TCE/RJ
- Pós Graduado em Engenharia de Software
- Autor do Livro “Android para Desenvolvedores”, Editora Brasport
- MBA Gerenciamento de Projetos
- Certificações Profissionais:
  - SCJP, OCWD, OCJA Part I
  - RHSA, Big IP Essentials e Advanced

# Contatos:

- [luciocamilo@gmail.com](mailto:luciocamilo@gmail.com)
- [www.itnerante.com.br/profile/luciocamilo](http://www.itnerante.com.br/profile/luciocamilo)



# Fundamentos



PL/SQL é a linguagem para consulta de banco de dados padrão ANSI (American National Standards Institute) utilizado pelos principais sistemas gerenciadores de banco de dados (SGBDs) do mercado.

Certo

Errado

PL/SQL é a linguagem para consulta de banco de dados padrão ANSI (American National Standards Institute) utilizado pelos principais sistemas gerenciadores de banco de dados (SGBDs) do mercado.

Certo

➡ Errado

Em relação ao PL/SQL, é INCORRETO afirmar:

- A) Pode utilizar todos os comandos SQL.
- B) Aplicações executáveis em qualquer plataforma com base Oracle instalada.
- C) Programas obedecem, sempre, a uma estrutura de blocos de indentação, hierarquicamente definidos.
- D) Todo comando SELECT pede a cláusula INTO para associar valores de colunas da base Oracle a variáveis PL/SQL.
- E) Variáveis não precisam ser declaradas antes de serem utilizadas.

Em relação ao PL/SQL, é INCORRETO afirmar:

A) Pode utilizar todos os comandos SQL.

B) Aplicações executáveis em qualquer plataforma com base Oracle instalada.

C) Programas obedecem, sempre, a uma estrutura de blocos de indentação, hierarquicamente definidos.

D) Todo comando SELECT pede a cláusula INTO para associar valores de colunas da base Oracle a variáveis PL/SQL.

➡ E) Variáveis não precisam ser declaradas antes de serem utilizadas.



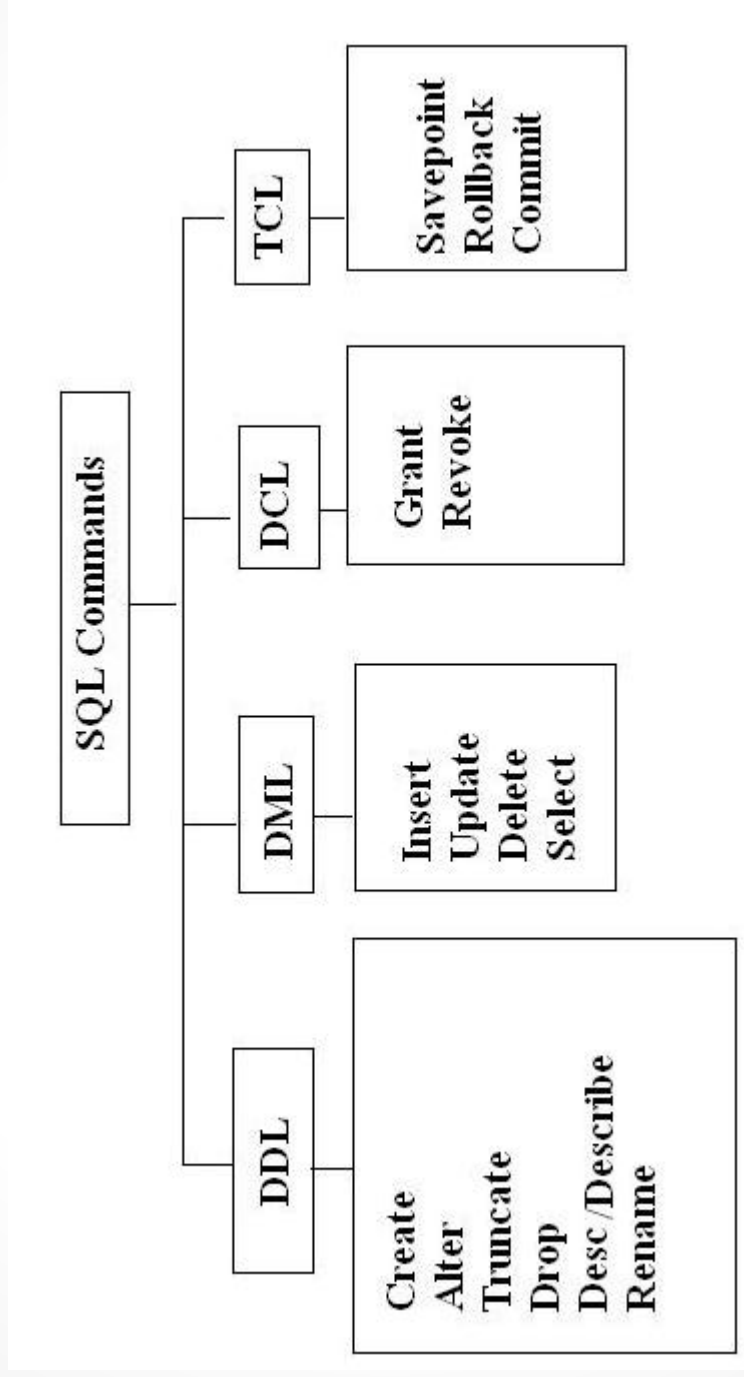
## Questão 03 – 2012 – FCC – TRE/CE

Sobre Bancos de Dados, analise:

- I. PL/SQL combina a facilidade e flexibilidade da SQL com a funcionalidade de procedimentos de uma linguagem de programação estruturada, tal como *IF... THEN*, *WHILE* e *LOOP*.
- II. As declarações DML permitem que um usuário conceda a outros os privilégios, ou direitos, de acesso ao banco de dados e a objetos específicos dentro do banco de dados.
- III. Os *triggers* de banco de dados podem ser usados para automatizar a geração de dados, fazer auditoria das modificações, implantar as restrições complexas de integridade e personalizar as autorizações complexas de segurança.
- IV. As restrições de integridade são definidas com uma tabela e armazenadas como parte dessa tabela, de forma centralizada no dicionário de dados de um banco de dados, de modo que todos os aplicativos devem obedecer ao mesmo conjunto de regras.

Está correto o que consta em

- A) I, II, III e IV.
- B) I, III e IV, apenas.
- C) I e IV, apenas.
- D) II e III, apenas.
- E) III e IV, apenas.



## Questão 03 – 2012 – FCC – TRE/CE

Sobre Bancos de Dados, analise:

- I. PL/SQL combina a facilidade e flexibilidade da SQL com a funcionalidade de procedimentos de uma linguagem de programação estruturada, tal como *IF... THEN*, *WHILE* e *LOOP*.
- II. As declarações DML permitem que um usuário conceda a outros os privilégios, ou direitos, de acesso ao banco de dados e a objetos específicos dentro do banco de dados.
- III. Os *triggers* de banco de dados podem ser usados para automatizar a geração de dados, fazer auditoria das modificações, implantar as restrições complexas de integridade e personalizar as autorizações complexas de segurança.
- IV. As restrições de integridade são definidas com uma tabela e armazenadas como parte dessa tabela, de forma centralizada no dicionário de dados de um banco de dados, de modo que todos os aplicativos devem obedecer ao mesmo conjunto de regras.

Está correto o que consta em

A) I, II, III e IV.

➡ B) I, III e IV, apenas.

C) I e IV, apenas.

D) II e III, apenas.

E) III e IV, apenas.

PL/SQL é uma linguagem de programação que permite a inclusão de unidades de programa no SGBD para a manipulação de dados.

Certo

Errado

PL/SQL é uma linguagem de programação que permite a inclusão de unidades de programa no SGBD para a manipulação de dados.

➡ Certo

Errado

A linguagem PL/SQL, introduzida nos gerenciadores de banco de dados ORACLE,

- A) aumenta a capacidade não-procedural da SQL, oferecendo e combinando blocos de construtores procedurais.
- B) constitui uma interface básica pela qual pode-se entrar e executar comandos SQL para manipulações genéricas de um banco.
- C) trata-se de uma linguagem que identifica quais as informações necessárias e não como buscá-las.
- D) tem os seus comandos executados pelo executor de SQL do *Kernel*.
- E) tem os seus comandos enviados para serem processados pelo SGBD um por vez.

A linguagem PL/SQL, introduzida nos gerenciadores de banco de dados ORACLE,

- ➡ A) aumenta a capacidade não-procedural da SQL, oferecendo e combinando blocos de construtores procedurais.
- B) constitui uma interface básica pela qual pode-se entrar e executar comandos SQL para manipulações genéricas de um banco.
- C) trata-se de uma linguagem que identifica quais as informações necessárias e não como buscá-las.
- D) tem os seus comandos executados pelo executor de SQL do *Kernel*.
- E) tem os seus comandos enviados para serem processados pelo SGBD um por vez.

## Questão 06 – 2009 – COSEAC - DATAPREV

Na linguagem PL/SQL, pacotes são conjuntos de:

- A) índices aplicados sobre partições;
- B) tipos definidos por usuário;
- C) triggers pré-compiladas;
- D) visões e tabelas;
- E) procedimentos e/ou funções agrupados.



## Questão 06 – 2009 – COSEAC - DATAPREV

Na linguagem PL/SQL, pacotes são conjuntos de:

A) índices aplicados sobre partições;

B) tipos definidos por usuário;

C) triggers pré-compiladas;

D) visões e tabelas;

➡ E) procedimentos e/ou funções agrupados.

Um código PL/SQL anônimo é aquele:

- A) que é compilado ao ser salvo no banco;
- B) armazenado em um cliente e enviado ao servidor para ser executado;
- C) estruturado na forma de pacote;
- D) armazenado no dicionário de dados do banco;
- E) armazenado no banco na forma de bloco de código nomeado.

Um código PL/SQL anônimo é aquele:

- ➡ A) que é compilado ao ser salvo no banco;
- B) armazenado em um cliente e enviado ao servidor para ser executado;
- C) estruturado na forma de pacote;
- D) armazenado no dicionário de dados do banco;
- E) armazenado no banco na forma de bloco de código nomeado.

## Questão 08 – 2014 – FUMARC - PBH

Quanto ao desenvolvimento PL/SQL utilizando objetos do banco de dados Oracle, todas as afirmativas a seguir estão corretas, EXCETO:

- A) Assim como procedimentos e funções em pacotes PL/SQL, métodos em tipos de objetos (object types) podem ser sobrecarregados (overloaded).
- B) Métodos de objetos são funções ou procedimentos que podem ser especificados e codificados em tipos de objeto (object types) para se implementarem operações ou comportamentos em objetos
- C) Os atributos de um tipo de objeto (object type) podem ser definidos com %ROWTYPE.
- D) Os tipos de objeto (object types) do banco de dados Oracle podem ser utilizados em procedimentos e funções PL/SQL

## Questão 08 – 2014 – FUMARC - PBH

Quanto ao desenvolvimento PL/SQL utilizando objetos do banco de dados Oracle, todas as afirmativas a seguir estão corretas, EXCETO:

A) Assim como procedimentos e funções em pacotes PL/SQL, métodos em tipos de objetos (object types) podem ser sobrecarregados (overloaded).

B) Métodos de objetos são funções ou procedimentos que podem ser especificados e codificados em tipos de objeto (object types) para se implementarem operações ou comportamentos em objetos

➡ C) Os atributos de um tipo de objeto (object type) podem ser definidos com %ROWTYPE.

D) Os tipos de objeto (object types) do banco de dados Oracle podem ser utilizados em procedimentos e funções PL/SQL

## Questão 09 – 2009 – COSEAC - DATAPREV

Marque o único objeto da lista abaixo relativo à PL/SQL:

- A) restrição;
- B) tabela;
- C) visão;
- D) papel;
- E) procedimento.

## Questão 09 – 2009 – COSEAC - DATAPREV

Marque o único objeto da lista abaixo relativo à PL/SQL:

A) restrição;

B) tabela;

C) visão;

D) papel;

➡ E) procedimento.

## Questão 10 – 2008 – FCC – MPE/RS

A linguagem PL/SQL é uma estrutura em blocos, compostos basicamente das partes declarativa, executável e manipulação de exceções, as quais são, respectivamente, de uso

- A) opcional, para todas as partes.
- B) obrigatório, para todas as partes.
- C) opcional, obrigatório e obrigatório.
- D) obrigatório, obrigatório e opcional.
- E) opcional, obrigatório e opcional.



## Questão 10 – 2008 – FCC – MPE/RS

A linguagem PL/SQL é uma estrutura em blocos, compostos basicamente das partes declarativa, executável e manipulação de exceções, as quais são, respectivamente, de uso

- A) opcional, para todas as partes.
- B) obrigatório, para todas as partes.
- C) opcional, obrigatório e obrigatório.
- D) obrigatório, obrigatório e opcional.

➡ E) opcional, obrigatório e opcional.

**DECLARE**

*Variable Declarations (Optional)*

**BEGIN**

*Executable Statements (Mandatory)*

**EXCEPTION**

*Exception Handlers (Optional)*

**END ;**

Trechos de memória alocada para processar as declarações *Select* definidos pelo próprio PL/SQL são denominados

- A) extensões.
- B) coleções aninhadas.
- C) cursores explícitos.
- D) coleções *varrays*.
- E) cursores implícitos.

Trechos de memória alocada para processar as declarações *Select* definidos pelo próprio PL/SQL são denominados

A) extensões.

B) coleções aninhadas.

C) cursores explícitos.

D) coleções *varrays*.

➡ E) cursores implícitos.

Em PL/SQL, parâmetros cujo tipo não esteja explicitamente declarado são considerados como do tipo IN.

Certo

Errado

# Tipos de Parâmetros

- IN => Enviam informações para as stored procedures (Default)
- OUT => Similar ao return existente em functions
- IN OUT => Enviam e recuperam valores das stored procedures

Em PL/SQL, parâmetros cujo tipo não esteja explicitamente declarado são considerados como do tipo IN.

⇒ Certo

Errado

## Questão 13 – 2011 - FCC- INFRAERO

Na passagem de parâmetros para Procedures e Funções em PL/SQL, o tipo de parâmetro que tem por objetivo enviar valores para as *storedprocedures* é:

- A) IN-OFF.
- B) OUT.
- C) OFF.
- D) ON.
- E) IN.

## Questão 13 – 2011 - FCC- INFRAERO

Na passagem de parâmetros para Procedures e Funções em PL/SQL, o tipo de parâmetro que tem por objetivo enviar valores para as *storedprocedures* é:

A) IN-OFF.

B) OUT.

C) OFF.

D) ON.

➡ E) IN.



## Questão 14 – 2011 - FCC- TRT19

Com relação aos modos de parâmetros de *procedures* PL/SQL é correto afirmar:

- A) Se o modo de um parâmetro não for especificado, ele será OUT (*valor default*).
- B) Existem três modos de parâmetros: IN, OUT e IN OUT.
- C) Parâmetros IN são utilizados para retornar um valor.
- D) Parâmetros OUT são utilizados para passar um valor para a *procedure* que está sendo invocada.
- E) Uma *procedure* deve ter pelo menos um parâmetro do modo IN e um parâmetro do modo OUT.

## Questão 14 – 2011 - FCC- TRT19

Com relação aos modos de parâmetros de *procedures* PL/SQL é correto afirmar:

A) Se o modo de um parâmetro não for especificado, ele será OUT (*valor default*).

➡ B) Existem três modos de parâmetros: IN, OUT e IN OUT.

C) Parâmetros IN são utilizados para retornar um valor.

D) Parâmetros OUT são utilizados para passar um valor para a *procedure* que está sendo invocada.

E) Uma *procedure* deve ter pelo menos um parâmetro do modo IN e um parâmetro do modo OUT.

Com relação à linguagem PL/SQL (*Procedural Language/ Structured Query Language*), uma extensão da linguagem SQL, analise as afirmativas a seguir.

- I. Com o uso da PL/SQL armazenado geralmente o desempenho do sistema como um todo tende a aumentar.
- II. Como o código PL/SQL pode ser armazenado de forma centralizada em um banco de dados, o tráfego de rede entre o aplicativo e o banco é reduzido.
- III. PL/SQL é totalmente compatível com os principais SGBDs do mercado por ser baseada em padrões abertos e livres, facilitando possíveis migrações entre diferentes SGBDs.

Assinale:

- A) se somente a afirmativa I estiver correta.
- B) se somente as afirmativas I e II estiverem corretas.
- C) se somente as afirmativas I e III estiverem corretas.
- D) se somente as afirmativas II e III estiverem corretas.
- E) se todas as afirmativas estiverem corretas.

Com relação à linguagem PL/SQL (*Procedural Language/ Structured Query Language*), uma extensão da linguagem SQL, analise as afirmativas a seguir.

- I. Com o uso da PL/SQL armazenado geralmente o desempenho do sistema como um todo tende a aumentar.
- II. Como o código PL/SQL pode ser armazenado de forma centralizada em um banco de dados, o tráfego de rede entre o aplicativo e o banco é reduzido.
- III. PL/SQL é totalmente compatível com os principais SGBDs do mercado por ser baseada em padrões abertos e livres, facilitando possíveis migrações entre diferentes SGBDs.

Assinale:

- A) se somente a afirmativa I estiver correta.
- ➡ B) se somente as afirmativas I e II estiverem corretas.
- C) se somente as afirmativas I e III estiverem corretas.
- D) se somente as afirmativas II e III estiverem corretas.
- E) se todas as afirmativas estiverem corretas.

## Questão 16 – 2009 – FCC – TRT16

Os programas PL/SQL são constituídos por blocos que executam operações lógicas e cada bloco tem três partes que definem as:

- I. declarações de variáveis e itens.
- II. instruções procedurais e SQL.
- III. instruções de tratamento de erros.

No bloco é obrigatória a presença da seção que se afirma em

- a) I e II, apenas.
- b) II e III, apenas.
- c) I, II e III.
- d) I, apenas.
- e) II, apenas.

## Questão 16 – 2009 – FCC – TRT16

Os programas PL/SQL são constituídos por blocos que executam operações lógicas e cada bloco tem três partes que definem as:

- I. declarações de variáveis e itens.
- II. instruções procedurais e SQL.
- III. instruções de tratamento de erros.

No bloco é obrigatória a presença da seção que se afirma em

- a) I e II, apenas.
- b) II e III, apenas.
- c) I, II e III.
- d) I, apenas.

➡ e) II, apenas.

## Questão 17 – 2013 – FEPESE - JUCESC

Assinale a alternativa que indica uma forma válida de tratar exceções em PL/SQL, considerando as palavras-chave necessárias para tal, na sequência **correta**.

A) Begin; Exception; End.

B) Begin; End; Exception.

C) When; Exception; End.

D) Exception; While; End.

E) Não há uma forma de tratar exceções em PL/ SQL.

## Questão 17 – 2013 – FEPESE - JUCESC

Assinale a alternativa que indica uma forma válida de tratar exceções em PL/SQL, considerando as palavras-chave necessárias para tal, na sequência **correta**.

➡ A) Begin; Exception; End.

B) Begin; End; Exception.

C) When; Exception; End.

D) Exception; While; End.

E) Não há uma forma de tratar exceções em PL/ SQL.



A criação de *stored procedure*, que são procedimentos armazenados dentro do próprio banco de dados, não é possível na linguagem PL/SQL.

Certo

Errado

A criação de *stored procedure*, que são procedimentos armazenados dentro do próprio banco de dados, não é possível na linguagem PL/SQL.

Certo

➡ Errado

## Questão 19 – 2009 – FCC – TRE/PI

No PL SQL, uma procedure pode ser chamada a partir de uma

I. aplicação.

II. *outra procedure.*

III. *uma query.*

IV. *trigger.*

Está correto o que se afirma em

A) I e II, apenas.

B) I, II e III, apenas.

C) I, II, III e IV.

D) I e III, apenas.

E) II, III e IV, apenas.

## Questão 19 – 2009 – FCC – TRE/PI

No PL SQL, uma procedure pode ser chamada a partir de uma

I. aplicação.

II. *outra procedure.*

III. *uma query.*

IV. *trigger.*

Está correto o que se afirma em

A) I e II, apenas.

B) I, II e III, apenas.

➡ C) I, II, III e IV.

D) I e III, apenas.

E) II, III e IV, apenas.

## Questão 20 – 2014 – FUMARC – PBH

Sobre o tratamento de exceções em PL/SQL, analise as seguintes sentenças, identificando-as com (V) ou (F), conforme sejam verdadeiras ou falsas.

- ( ) Uma exceção pode ser tratada por mais de um manipulador de exceções.
- ( ) Exceções predefinidas são automaticamente lançadas para certos erros, envolvendo variáveis e operações de banco de dados.
- ( ) Exceções podem ser declaradas e lançadas explicitamente pelo desenvolvedor.
- ( ) Exceções lançadas dentro de manipuladores de exceções não podem ser tratadas.

A sequência CORRETA, de cima para baixo, é:

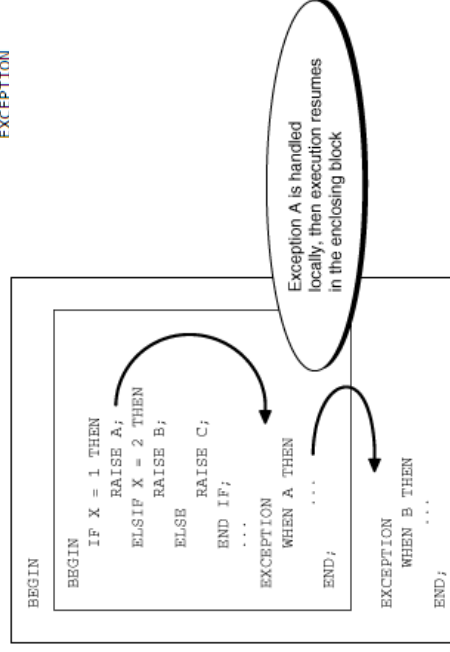
- A) F, F, V, V
- B) F, V, V, F
- C) V, V, F, F
- E) V, F, F, V

Exception	Oracle Error	SQLCODE Value
ACCESS_INTO_NULL	ORA-06530	-6530
CASE_NOT_FOUND	ORA-06592	-6592
COLLECTION_IS_NULL	ORA-06531	-6531
CURSOR_ALREADY_OPEN	ORA-06511	-6511
DUP_VAL_ON_INDEX	ORA-00001	-1
INVALID_CURSOR	ORA-01001	-1001
INVALID_NUMBER	ORA-01722	-1722
LOGIN_DENIED	ORA-01017	-1017
NO_DATA_FOUND	ORA-01403	+100
NOT_LOGGED_ON	ORA-01012	-1012
PROGRAM_ERROR	ORA-06501	-6501
ROWTYPE_MISMATCH	ORA-06504	-6504
SELF_IS_NULL	ORA-30625	-30625
STORAGE_ERROR	ORA-06500	-6500
SUBSCRIPT_BEYOND_COUNT	ORA-06533	-6533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	-6532
SYS_INVALID_ROWID	ORA-01410	-1410
TIMEOUT_ON_RESOURCE	ORA-00051	-51
TOO_MANY_ROWS	ORA-01422	-1422
VALUE_ERROR	ORA-06502	-6502
ZERO_DIVIDE	ORA-01476	-1476

```

DECLARE
    past_due EXCEPTION;
    acct_num NUMBER;
BEGIN
    DECLARE ----- sub-block begins
        past_due EXCEPTION; -- this declaration prevails
        acct_num NUMBER;
    BEGIN
        ...
        IF ... THEN
            RAISE past_due; -- this is not handled
        END IF;
    END; ----- sub-block ends
EXCEPTION
    ie THEN -- does not handle RAISED exception

```



## Questão 20 – 2014 – FUMARC – PBH

Sobre o tratamento de exceções em PL/SQL, analise as seguintes sentenças, identificando-as com (V) ou (F), conforme sejam verdadeiras ou falsas.

- ( ) Uma exceção pode ser tratada por mais de um manipulador de exceções.
- ( ) Exceções predefinidas são automaticamente lançadas para certos erros, envolvendo variáveis e operações de banco de dados.
- ( ) Exceções podem ser declaradas e lançadas explicitamente pelo desenvolvedor.
- ( ) Exceções lançadas dentro de manipuladores de exceções não podem ser tratadas.

A sequência CORRETA, de cima para baixo, é:

A) F, F, V, V

➡ B) F, V, V, F

C) V, V, F, F

E) V, F, F, V

## Questão 21 – 2012 – FEPESE - FATMA

Assinale a alternativa que define corretamente a exceção predefinida no Oracle 10g que corresponde ao erro interno do PL/SQL:

- A) ZERO\_DIVIDE
- B) ACCESS\_INTO\_NULL
- C) PROGRAM\_ERROR
- D) TOO\_MANY\_ROWS
- E) CASE\_NOT\_FOUND



Exception	Raised when ...
ACCESS_INTO_NULL	Your program attempts to assign values to the attributes of an uninitialized (atomically null) object.
CASE_NOT_FOUND	None of the choices in the WHEN clauses of a CASE statement is selected, and there is no ELSE clause.
COLLECTION_IS_NULL	Your program attempts to apply collection methods other than EXISTS to an uninitialized (atomically null) nested table or varray, or the program attempts to assign values to the elements of an uninitialized nested table or varray.
CURSOR_ALREADY_OPEN	Your program attempts to open an already open cursor. A cursor must be closed before it can be reopened. A cursor FOR loop automatically opens the cursor to which it refers. So, your program cannot open that cursor inside the loop.
DUP_VAL_ON_INDEX	Your program attempts to store duplicate values in a database column that is constrained by a unique index.
INVALID_CURSOR	Your program attempts an illegal cursor operation such as closing an unopened cursor.
INVALID_NUMBER	In a SQL statement, the conversion of a character string into a number fails because the string does not represent a valid number. (In procedural statements, VALUE_ERROR is raised.) This exception is also raised when the LIMIT-clause expression in a bulk FETCH statement does not evaluate to a positive number.
LOGIN_DENIED	Your program attempts to log on to Oracle with an invalid username and/or password.
NO_DATA_FOUND	A SELECT INTO statement returns no rows, or your program references a deleted element in a nested table or an uninitialized element in an index-by table. SQL aggregate functions such as AVG and SUM always return a value or a null. So, a SELECT INTO statement that calls an aggregate function never raises NO_DATA_FOUND. The FETCH statement is expected to return no rows eventually, so when that happens, no exception is raised.
NOT_LOGGED_ON	Your program issues a database call without being connected to Oracle.
PROGRAM_ERROR	PL/SQL has an internal problem.
ROWTYPE_MISMATCH	The host cursor variable and PL/SQL cursor variable involved in an assignment have incompatible return types. For example, when an open host cursor variable is passed to a stored subprogram, the return types of the actual and formal parameters must be compatible.
SELF_IS_NULL	Your program attempts to call a MEMBER method on a null instance. That is, the built-in parameter SELF (which is always the first parameter passed to a MEMBER method) is null.
STORAGE_ERROR	PL/SQL runs out of memory or memory has been corrupted.
SUBSCRIPT_BEYOND_COUNT	Your program references a nested table or varray element using an index number larger than the number of elements in the collection.
SUBSCRIPT_OUTSIDE_LIMIT	Your program references a nested table or varray element using an index number (-1 for example) that is outside the legal range.
SVS_INVALID_ROWID	The conversion of a character string into a universal rowid fails because the character string does not represent a valid rowid.
TIMEOUT_ON_RESOURCE	A time-out occurs while Oracle is waiting for a resource.
TOO_MANY_ROWS	A SELECT INTO statement returns more than one row.
VALUE_ERROR	An arithmetic, conversion, truncation, or size-constraint error occurs. For example, when your program selects a column value into a character variable, if the value is longer than the declared length of the variable, PL/SQL aborts the assignment and raises VALUE_ERROR. In procedural statements, VALUE_ERROR is raised if the conversion of a character string into a number fails. (In SQL statements, INVALID_NUMBER is raised.)
ZERO_DIVIDE	Your program attempts to divide a number by zero.

## Questão 21 – 2012 – FEPESE - FATMA

Assinale a alternativa que define corretamente a exceção predefinida no Oracle 10g que corresponde ao erro interno do PL/SQL:

A) ZERO\_DIVIDE

B) ACCESS\_INTO\_NULL

➡ C) PROGRAM\_ERROR

D) TOO\_MANY\_ROWS

E) CASE\_NOT\_FOUND

Exception	Oracle Error	SQLCODE Value
ACCESS_INTO_NULL	ORA-06530	-6530
CASE_NOT_FOUND	ORA-06592	-6592
COLLECTION_IS_NULL	ORA-06531	-6531
CURSOR_ALREADY_OPEN	ORA-06511	-6511
DUP_VAL_ON_INDEX	ORA-00001	-1
INVALID_CURSOR	ORA-01001	-1001
INVALID_NUMBER	ORA-01722	-1722
LOGIN_DENIED	ORA-01017	-1017
NO_DATA_FOUND	ORA-01403	+100
NOT_LOGGED_ON	ORA-01012	-1012
PROGRAM_ERROR	ORA-06501	-6501
ROWTYPE_MISMATCH	ORA-06504	-6504
SELF_IS_NULL	ORA-30625	-30625
STORAGE_ERROR	ORA-06500	-6500
SUBSCRIPT_BEYOND_COUNT	ORA-06533	-6533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	-6532
SYS_INVALID_ROWID	ORA-01410	-1410
TIMEOUT_ON_RESOURCE	ORA-00051	-51
TOO_MANY_ROWS	ORA-01422	-1422
VALUE_ERROR	ORA-06502	-6502
ZERO_DIVIDE	ORA-01476	-1476

No tipo Escalares, NÃO é uma das quatro famílias de tipos de dados predefinidos do PL/SQL:

- A) referências.
- B) data e hora.
- C) números.
- D) caracteres.
- E) booleanos.

No tipo Escalares, NÃO é uma das quatro famílias de tipos de dados predefinidos do PL/SQL:

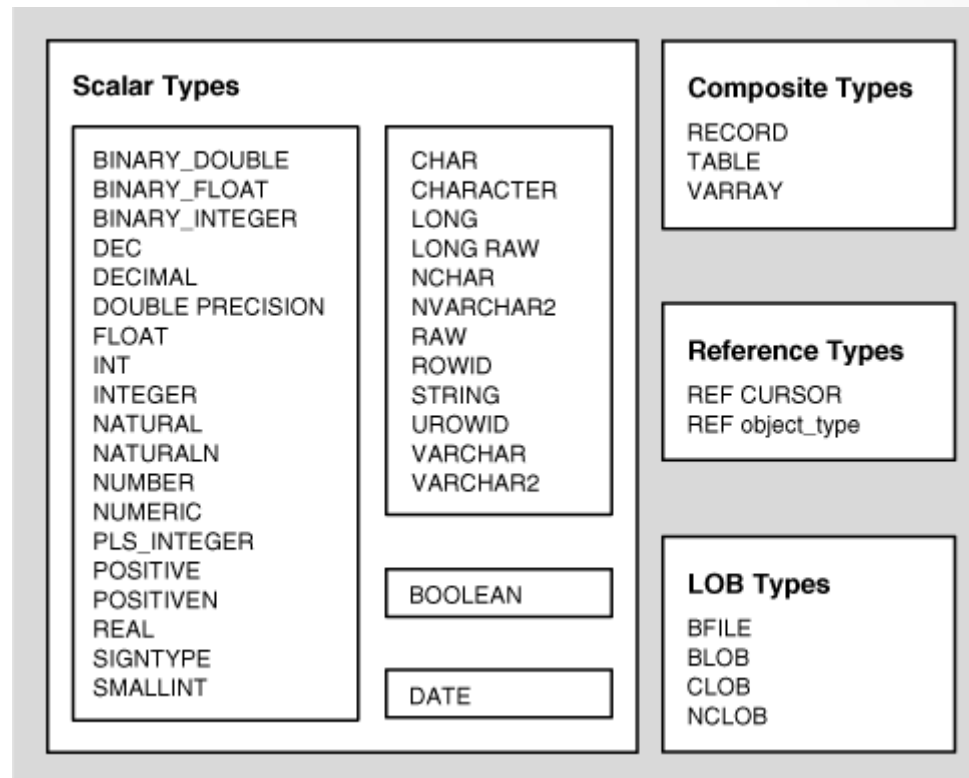
➡ A) referências.

B) data e hora.

C) números.

D) caracteres.

E) booleanos.



## Questão 23 – 2009 – FCC – TRT3

Para passar resultados de query entre subprogramas PL/SQL, ou seja, quando um subprograma executa uma query e o resultado será processado por outro subpro- grama, utiliza-se uma

- A) view.
- B) subquery.
- C) variável cursor.
- D) variável bind.
- E) variável locator.

## Questão 23 – 2009 – FCC – TRT3

Para passar resultados de query entre subprogramas PL/SQL, ou seja, quando um subprograma executa uma query e o resultado será processado por outro subpro- grama, utiliza-se uma

A) view.

B) subquery.

➡ C) variável cursor.

D) variável bind.

E) variável locator.

## Questão 24 – 2012 - FCC- TJ/PE

Na parte declarativa de qualquer bloco PL/SQL, subprograma ou pacote pode-se criar coleções (*collections*), definindo-se o tipo de coleção

A) NESTED TABLE ou ARRAY.

B) TABLE ou ARRAY.

C) TABLE ou VARRAY.

D) NESTED ou ARRAY.

E) NESTED ou VARRAY.

## Questão 24 – 2012 - FCC- TJ/PE

Na parte declarativa de qualquer bloco PL/SQL, subprograma ou pacote pode-se criar coleções (*collections*), definindo-se o tipo de coleção

A) NESTED TABLE ou ARRAY.

B) TABLE ou ARRAY.

➡ C) TABLE ou VARRAY.

D) NESTED ou ARRAY.

E) NESTED ou VARRAY.



# Triggers

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
    Declaration-statements
BEGIN
    Executable-statements
EXCEPTION
    Exception-handling-statements
END;
```

- Create [Or Replace] – cria ou substitui uma trigger que já existe.
- {Before | After | Instead Of} – especifica quando a trigger deverá ser executada.
- {Insert [Or] | Update [Or] | Delete} – especifica a operação DML.
- [Of col\_name] – especifica a coluna que terá que sofrer modificação.
- [Referencing OLD AS o NEW AS n] – permite que referencia novos e antigos valores para as declarações DML.
- [For Each Row] – especifica o nível do trigger.
- When (condition) – provê uma condição para as linhas que o trigger será acionado.

Um bloco PL/SQL que está associado a um evento ocorrido no banco de dados Oracle é do tipo

- A) função.
- B) gatilho.
- C) anônimo.
- D) procedimento.
- E) dinâmico.

Um bloco PL/SQL que está associado a um evento ocorrido no banco de dados Oracle é do tipo

A) função.

➡ B) gatilho.

C) anônimo.

D) procedimento.

E) dinâmico.

## Questão 26 – 2008 – FGV – SENADO

**Triggers** constituem construções especiais em **PL/SQL**, semelhantes a *procedures*. Enquanto uma *procedure* é executada explicitamente via uma *procedure call*, um *trigger* é executado implicitamente se um determinado evento acontece, a partir de um dos seguintes comandos:

- A) INSERT, DELETE ou UPDATE.
- B) DELETE, UPDATE ou COMMIT.
- C) COMMIT, ROLLBACK ou INSERT.
- D) ROLLBACK, INSERT ou DELETE.
- E) UPDATE, COMMIT ou ROLLBACK.

**Triggers** constituem construções especiais em **PL/SQL**, semelhantes a *procedures*. Enquanto uma *procedure* é executada explicitamente via uma *procedure call*, um *trigger* é executado implicitamente se um determinado evento acontece, a partir de um dos seguintes comandos:

- ➡ A) INSERT, DELETE ou UPDATE.
- B) DELETE, UPDATE ou COMMIT.
- C) COMMIT, ROLLBACK ou INSERT.
- D) ROLLBACK, INSERT ou DELETE.
- E) UPDATE, COMMIT ou ROLLBACK.

No ORACLE, um *trigger* é um bloco PL/SQL armazenado e associado a uma tabela, esquema ou banco de dados que pode ser criado com o comando CREATE TRIGGER.

Certo

Errado

No ORACLE, um *trigger* é um bloco PL/SQL armazenado e associado a uma tabela, esquema ou banco de dados que pode ser criado com o comando CREATE TRIGGER.

 Certo

Errado

## Questão 28 – 2014 – FCC – TJ/AP

Na linguagem PL/SQL, deseja-se desabilitar um *trigger* denominado Abc, mas não excluí-lo do banco de dados. O comando que realiza esta tarefa é

- A) DROP TRIGGER Abc = 0;
- B) CHANGE TRIGGER Abc → 0;
- C) MAKE TRIGGER Abc DOWN;
- D) PUT TRIGGER Abc OFF;
- E) ALTER TRIGGER Abc DISABLE;



## Questão 28 – 2014 – FCC – TJ/AP

Na linguagem PL/SQL, deseja-se desabilitar um *trigger* denominado Abc, mas não excluí-lo do banco de dados. O comando que realiza esta tarefa é

A) DROP TRIGGER Abc = 0;

B) CHANGE TRIGGER Abc → 0;

C) MAKE TRIGGER Abc DOWN;

D) PUT TRIGGER Abc OFF;

➡ E) ALTER TRIGGER Abc DISABLE;

## Questão 29 – 2011 – FCC – TRT19

Para uma dada tabela, foram criados 4 *triggers* utilizando-se PL/SQL, conforme abaixo:

- I. *Trigger 1* – BEFORE <*statement*>
- II. *Trigger 2* – BEFORE <*for each row*>
- III. *Trigger 3* – AFTER <*statement*>
- IV. *Trigger 4* – AFTER <*for each row*>

A sequência correta em que os *triggers* acima serão disparados é:

- A) I, II, III e IV.
- B) II, I, III e IV.
- C) I, II, IV e III.
- D) I, IV, II e III.
- E) IV, III, II e I.

## Questão 29 – 2011 – FCC – TRT19

Para uma dada tabela, foram criados 4 *triggers* utilizando-se PL/SQL, conforme abaixo:

- I. *Trigger 1* – BEFORE <*statement*>
- II. *Trigger 2* – BEFORE <*for each row*>
- III. *Trigger 3* – AFTER <*statement*>
- IV. *Trigger 4* – AFTER <*for each row*>

A sequência correta em que os *triggers* acima serão disparados é:

- A) I, II, III e IV.
- B) II, I, III e IV.
- ➡ C) I, II, IV e III.
- D) I, IV, II e III.
- E) IV, III, II e I.

## Questão 30 – 2014 – UFBA – UFSBA

A execução apresentada do script deflagrará a execução da trigger trg\_01, construída em PL/SQL, que é demonstrada a seguir.

```
update tabela_a
    set coluna_a = 'X'
    where coluna_b = 'Y';

create trigger trg_01
    before delete on tabela_a
    begin
        insert into tabela_b values ('W', 'K');
    end;
```

Certo

Errado

## Questão 30 – 2014 – UFBA – UFSBA

A execução apresentada do script deflagrará a execução da trigger trg\_01, construída em PL/SQL, que é demonstrada a seguir.

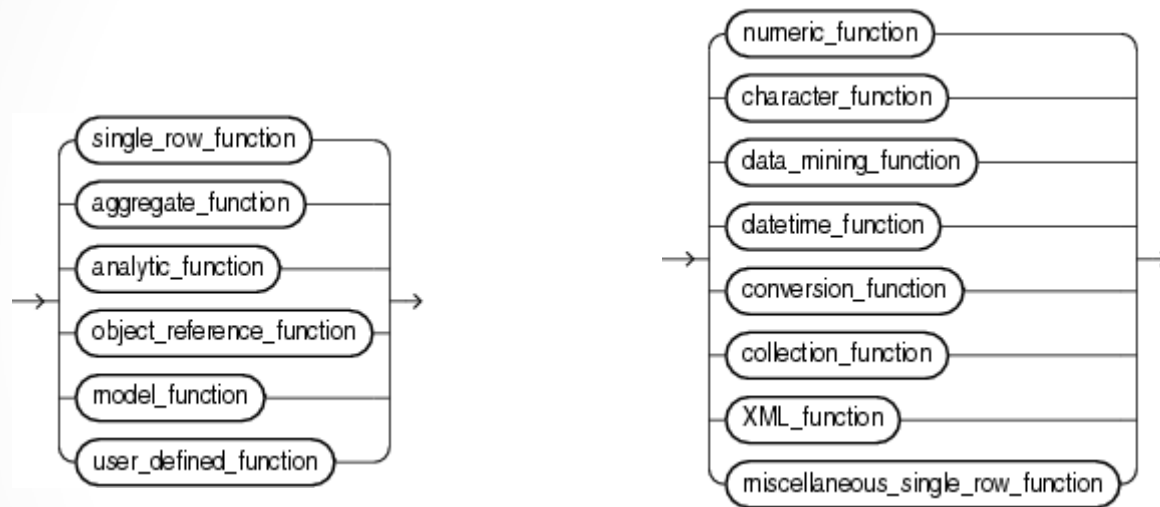
```
update tabela_a
    set coluna_a = 'X'
    where coluna_b = 'Y';

create trigger trg_01
    before delete on tabela_a
    begin
        insert into tabela_b values ('W', 'K');
    end;
```

Certo

 Errado

# Funções



A função de caractere pré-definida do PL/SQL que retorna um valor numérico é

- A) ASCII (string).
- B) LOWER (string).
- C) UPPER (string).
- D) SOUNDEX (string).
- E) INITCAP (string).

A função de caractere pré-definida do PL/SQL que retorna um valor numérico é

- ➡ A) ASCII (string).
- B) LOWER (string).
- C) UPPER (string).
- D) SOUNDEX (string).
- E) INITCAP (string).



Para verificar se um valor à esquerda está contido na lista à direita utiliza-se, no PL/SQL do ORACLE, o operador de comparação

- A) SOME
- B) ANY
- C) LIKE
- D) IN
- E) BETWEEN

Para verificar se um valor à esquerda está contido na lista à direita utiliza-se, no PL/SQL do ORACLE, o operador de comparação

A) SOME

B) ANY

C) LIKE

➡ D) IN

E) BETWEEN

## Questão 26 – 2014 – FCC – TRT16

Considere o comando PL/SQL a seguir:

```
select nome, cargo, salario, comissao, (salario*12)+NVL(comissao,2) from  
funcionario;
```

A função NVL foi usada para

- A) elevar os valores do campo comissao ao quadrado.
- B) converter valores nulos do campo comissao para 2.
- C) obter os valores do campo comissao com duas casas decimais.
- D) obter apenas os dois primeiros valores da direita do campo comissao.
- E) calcular a raiz quadrada dos valores do campo comissao.

## Questão 26 – 2014 – FCC – TRT16

Considere o comando PL/SQL a seguir:

```
select nome, cargo, salario, comissao, (salario*12)+NVL(comissao,2) from  
funcionario;
```

A função NVL foi usada para

- A) elevar os valores do campo comissao ao quadrado.
- ➡ B) converter valores nulos do campo comissao para 2.
- C) obter os valores do campo comissao com duas casas decimais.
- D) obter apenas os dois primeiros valores da direita do campo comissao.
- E) calcular a raiz quadrada dos valores do campo comissao.

## Questão 27 – 2014 – FCC – TRT2

Considere uma tabela em uso chamada func, de um banco de dados Oracle contendo, dentre outros, os campos para armazenar o número do funcionário (FunNo) e a data de admissão (FDatAdm).

Para exibir o número do funcionário, a data de admissão e o número de meses entre a data de admissão e a data atual, utiliza-se a instrução PL/SQL

- A) select FunNo, FDatAdm, MONTHS(SYSDATE, FDatAdm) from func;
- B) select FunNo, FDatAdm, MONTHS\_BETWEEN(DATE(), FDatAdm) from func;
- C) select FunNo, FDatAdm, MONTHS-BETWEEN(SYSDATE, FDatAdm) from func;
- D)select FunNo, FDatAdm, MONTHS(CURRENT\_DATE, FDatAdm) from func;
- E) select FunNo, FDatAdm, MONTHS\_BETWEEN(SYSDATE, FDatAdm) from func;

## Questão 27 – 2014 – FCC – TRT2

Considere uma tabela em uso chamada func, de um banco de dados Oracle contendo, dentre outros, os campos para armazenar o número do funcionário (FunNo) e a data de admissão (FDatAdm).

Para exibir o número do funcionário, a data de admissão e o número de meses entre a data de admissão e a data atual, utiliza-se a instrução PL/SQL

- A) select FunNo, FDatAdm, MONTHS(SYSDATE, FDatAdm) from func;
- B) select FunNo, FDatAdm, MONTHS\_BETWEEN(DATE(), FDatAdm) from func;
- C) select FunNo, FDatAdm, MONTHS-BETWEEN(SYSDATE, FDatAdm) from func;
- D)select FunNo, FDatAdm, MONTHS(CURRENT\_DATE, FDatAdm) from func;
- ➡ E) select FunNo, FDatAdm, MONTHS\_BETWEEN(SYSDATE, FDatAdm) from func;

## Questão 28 – 2012 – FCC – TRE/CE

Considere a existência de uma tabela chamada funcionario que contém os campos abaixo:

id – inteiro, não nulo, chave primária.

nome – cadeia de caracteres, não nulo.

cargo – cadeia de caracteres, não nulo.

salario – número real, não nulo.

Utilizando a PL/SQL, para exibir o nome do funcionário, o salário, e o salário aumentado (reajustado) em 30%, arredondado com 0 casas decimais e com legenda de coluna Salário Reajustado, utiliza-se a instrução

- A) select nome, salario, arred((salario \* 1.3),0) "Salário Reajustado" from funcionario;
- B) select nome, salario, round((salario\* 0.30),0) as "Salário Reajustado" from funcionario;
- C) select nome, salario, nvl(salario\* 30%,0) as "Salário Reajustado" from funcionario;
- D) select nome, salario, round((salario \* 1.3),0) "Salário Reajustado" from funcionario;
- E) select nome, salario, arred((salario + 30%),0) as "Salário Reajustado" from funcionario;

## Questão 28 – 2012 – FCC – TRE/CE

Considere a existência de uma tabela chamada funcionario que contém os campos abaixo:

id – inteiro, não nulo, chave primária.  
nome – cadeia de caracteres, não nulo.  
cargo – cadeia de caracteres, não nulo.  
salario – número real, não nulo.

Utilizando a PL/SQL, para exibir o nome do funcionário, o salário, e o salário aumentado (reajustado) em 30%, arredondado com 0 casas decimais e com legenda de coluna Salário Reajustado, utiliza-se a instrução

- A) select nome, salario, arred((salario \* 1.3),0) "Salário Reajustado" from funcionario;
- B) select nome, salario, round((salario\* 0.30),0) as "Salário Reajustado" from funcionario;
- C) select nome, salario, nvl(salario\* 30%,0) as "Salário Reajustado" from funcionario;
- ➡ D) select nome, salario, round((salario \* 1.3),0) "Salário Reajustado" from funcionario;
- E) select nome, salario, arred((salario + 30%),0) as "Salário Reajustado" from funcionario;



# Scripts

ORACLE Database Express Edition

User: JOHN :JOHN

Home > SQL > SQL Scripts > Script Editor

Script Name: DB\_PATCH

Cancel Download Delete Save Run

Undo Redo Find

```
1 CREATE TABLE Customer_Addresses (customer_id number(10) NOT NULL, address_id number(10) NOT NULL,
2 ALTER TABLE Computer_Part ADD discount number(8, 2);
3 CREATE TABLE Part_Usage (usage_id number(10) NOT NULL, description varchar2(1024), part_id number
4 ALTER TABLE Customer_Addresses ADD CONSTRAINT FKCustomer_A330997 FOREIGN KEY (customer_id) REFERE
5 ALTER TABLE Part_Usage ADD CONSTRAINT FKPart_Usage45987 FOREIGN KEY (part_id) REFERENCES Computer
6 CREATE SEQUENCE seq_Customer_Addresses;
7 CREATE SEQUENCE seq_Part_Usage;
```

Connections

hr @ orcl

Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL | SYNONYMS

Actions...

```
1 CREATE TABLE "EMPLOYEES"
2 ( "EMPLOYEE_ID" NUMBER(6,0),
3 "FIRST_NAME" VARCHAR2(20 BYTE),
4 "LAST_NAME" VARCHAR2(25 BYTE) CONSTRAINT "EMP_LAST_NAME" NOT NULL,
5 "EMAIL" VARCHAR2(25 BYTE) CONSTRAINT "EMP_EMAIL_NN" NOT NULL,
6 "PHONE_NUMBER" VARCHAR2(20 BYTE),
7 "HIRE_DATE" DATE CONSTRAINT "EMP_HIRE_DATE_NN" NOT NULL,
8 "JOB_ID" VARCHAR2(10 BYTE) CONSTRAINT "EMP_JOB_NN" NOT NULL,
9 "SALARY" NUMBER(8,2),
10 "COMMISSION_PCT" NUMBER(2,2),
11 "MANAGER_ID" NUMBER(6,0),
12 "DEPARTMENT_ID" NUMBER(4,0),
13 CONSTRAINT "EMP_SALARY_MIN" CHECK (salary > 0) ENABLE,
14 CONSTRAINT "EMP_EMAIL_UK" UNIQUE ("EMAIL") ENABLE,
15 CONSTRAINT "EMP_EMP_ID_PK" PRIMARY KEY ("EMPLOYEE_ID") ENABLE,
16 CONSTRAINT "EMP_DEPT_FK" FOREIGN KEY ("DEPARTMENT_ID")
17 REFERENCES "DEPARTMENTS" ("DEPARTMENT_ID") ENABLE,
18 CONSTRAINT "EMP_JOB_FK" FOREIGN KEY ("JOB_ID")
19 REFERENCES "JOBS" ("JOB_ID") ENABLE,
20 CONSTRAINT "EMP_MANAGER_FK" FOREIGN KEY ("MANAGER_ID")
21 REFERENCES "EMPLOYEES" ("EMPLOYEE_ID") ENABLE
22 );
23
```

## Questão 29 – 2009 – FGV– MEC

```
CREATE OR REPLACE TRIGGER tr_fatura_pk  
  BEFORE INSERT ON FATURA  
  FOR EACH ROW  
  BEGIN  
    SELECT nvl(curr_fat,0) + 1 INTO :next_nr FROM FILIAL  
    WHERE cod_fil = :new.cod_fil;  
  END;
```

De acordo com o código PL SQL, é correto afirmar que:

- A) gera chave primária para a tabela Filial, mas dependendo da tabela Fatura.
- B) permite a alteração na tabela Filial, mas dependendo da última fatura da filial na tabela Fatura.
- C) permite a alteração na tabela Fatura, mas dependendo da última fatura da filial na tabela Filial.
- D) gera chave primária para a tabela Fatura, mas dependendo da última fatura da filial na tabela Filial.
- E) gera chave estrangeira para a tabela Fatura, mas dependendo da última fatura da filial na tabela Filial.

## Questão 29 – 2009 – FGV– MEC

```
CREATE OR REPLACE TRIGGER tr_fatura_pk
  BEFORE INSERT ON FATURA
  FOR EACH ROW
  BEGIN
    SELECT nvl(curr_fat,0) + 1 INTO :next_nr FROM FILIAL
    WHERE cod_fil = :new.cod_fil;
  END;
```

De acordo com o código PL SQL, é correto afirmar que:

- A) gera chave primária para a tabela Filial, mas dependendo da tabela Fatura.
- B) permite a alteração na tabela Filial, mas dependendo da última fatura da filial na tabela Fatura.
- C) permite a alteração na tabela Fatura, mas dependendo da última fatura da filial na tabela Filial.
- ➡ D) gera chave primária para a tabela Fatura, mas dependendo da última fatura da filial na tabela Filial.
- E) gera chave estrangeira para a tabela Fatura, mas dependendo da última fatura da filial na tabela Filial.

## Questão 30 – 2012 – FCC – TRT6

Considere os seguintes comandos em PL/SQL:

```
CREATE TABLE VALORES (ID INT NOT NULL PRIMARY KEY, VALOR INT);  
INSERT INTO VALORES VALUES (1,5);  
INSERT INTO VALORES VALUES (1,6);  
INSERT INTO VALORES VALUES (NULL,7);  
INSERT INTO VALORES VALUES (2,1);  
INSERT INTO VALORES VALUES (2,8);  
INSERT INTO VALORES VALUES (3,9);  
INSERT INTO VALORES VALUES (NULL,10);  
SELECT COUNT(VALOR) FROM VALORES WHERE VALOR >= 5;
```

A consulta retornará o valor

- A) 2.
- B) 14.
- C) 6.
- D) 29.
- E) 5.

## Questão 30 – 2012 – FCC – TRT6

Considere os seguintes comandos em PL/SQL:

```
CREATE TABLE VALORES (ID INT NOT NULL PRIMARY KEY, VALOR INT);  
INSERT INTO VALORES VALUES (1,5);  
INSERT INTO VALORES VALUES (1,6);  
INSERT INTO VALORES VALUES (NULL,7);  
INSERT INTO VALORES VALUES (2,1);  
INSERT INTO VALORES VALUES (2,8);  
INSERT INTO VALORES VALUES (3,9);  
INSERT INTO VALORES VALUES (NULL,10);  
SELECT COUNT(VALOR) FROM VALORES WHERE VALOR >= 5;
```

A consulta retornará o valor

- ➡ A) 2.  
B) 14.  
C) 6.  
D) 29.  
E) 5.

## Questão 31 – 2014 – FEPESE - MPE/SC

Suponha a seguinte consulta PL/SQL em uma tabela denominada *empregado*

```
SELECT empno,  
       enome,  
       funcao,  
       salario,  
       LAG(salario, 1, 0) OVER (ORDER BY salario) AS XX  
FROM empregado;
```

Nesta consulta, XX irá conter:

- A) O maior valor de salário da consulta.
- B) O menor valor de salário da consulta.
- C) O valor de salário do registro anterior.
- D) O valor de salário do próximo registro.
- E) O valor de salário da mediana da consulta.

## Questão 31 – 2014 – FEPESE - MPE/SC

Suponha a seguinte consulta PL/SQL em uma tabela denominada *empregado*

```
SELECT empno,  
       enome,  
       funcao,  
       salario,  
       LAG(salario, 1, 0) OVER (ORDER BY salario) AS XX  
FROM empregado;
```

Nesta consulta, XX irá conter:

- A) O maior valor de salário da consulta.
- B) O menor valor de salário da consulta.
- ➡ C) O valor de salário do registro anterior.
- D) O valor de salário do próximo registro.
- E) O valor de salário da mediana da consulta.

## Questão 32 – 2014 – FUMARC - PBH

Qual dos trechos PL/SQL abaixo permite identificar **corretamente** quantas linhas foram removidas pelo comando DELETE?

A)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
Delete from empregado where salario > 5000;  
v_quant_del := SQL%ROWCOUNT; . . .  
END;
```

B)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
v_quant_del := Delete from empregado where salario > 5000; . . .  
END;
```

C)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
Delete from empregado where salario > 5000 into v_quant_del; . . .  
END;
```

D)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
Delete into v_quant_del from empregado where salario > 5000; . . .  
END;
```



## Questão 32 – 2014 – FUMARC - PBH

Qual dos trechos PL/SQL abaixo permite identificar **corretamente** quantas linhas foram removidas pelo comando DELETE?

A)



```
DECLARE  
v_quant_del number(10);  
BEGIN  
Delete from empregado where salario > 5000;  
v_quant_del := SQL%ROWCOUNT; . . .  
END;
```

B)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
v_quant_del := Delete from empregado where salario > 5000; . . .  
END;
```

C)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
Delete from empregado where salario > 5000 into v_quant_del; . . .  
END;
```

D)

```
DECLARE  
v_quant_del number(10);  
BEGIN  
Delete into v_quant_del from empregado where salario > 5000; . . .  
END;
```

## Questão 33 – 2013 – FCC – MPE/CE

```
DECLARE
    TYPE tabela IS TABLE OF NUMBER;
    list NUMBER_TABLE := tabela(1,2,3,4,5,6,7,8);
BEGIN
    list.DELETE(2);
    FOR i IN 1..list.COUNT LOOP
        IF list.EXISTS(i) THEN
            dbms.output.put(''||list(i)');
        END IF;
    END LOOP;
    dbms_output.new_line;
END;
```

O código acima imprime na console:

- A) [1] [3] [4] [5] [6] [7] [8]
- B) (1) (2) (3) (4) (5) (6) (7) (8)
- C) [1] [2] [3] [4] [5] [6] [7] [8]
- D) [3] [4] [5] [6] [7] [8]
- E) 1 3 4 5 6 7 8

## Questão 33 – 2013 – FCC – MPE/CE

```
DECLARE
    TYPE tabela IS TABLE OF NUMBER;
    list NUMBER_TABLE := tabela(1,2,3,4,5,6,7,8);
BEGIN
    list.DELETE(2);
    FOR i IN 1..list.COUNT LOOP
        IF list.EXISTS(i) THEN
            dbms.output.put(''||list(i)');
        END IF;
    END LOOP;
    dbms_output.new_line;
END;
```

O código acima imprime na console:

- ➡ A) [1] [3] [4] [5] [6] [7] [8]
- B) (1) (2) (3) (4) (5) (6) (7) (8)
- C) [1] [2] [3] [4] [5] [6] [7] [8]
- D) [3] [4] [5] [6] [7] [8]
- E) 1 3 4 5 6 7 8

## Questão 34 – 2010 – CESPE – TRT21

O trecho de programa em PL/SQL a seguir possibilita aumentar o valor da variável *salary* em 15%, sempre que esta alcançar valor abaixo da variável *media\_salary*.

```
UPDATE servidor SET salary = salary *1.1  
WHERE salary < media_salary
```

Certo

Errado

## Questão 34 – 2010 – CESPE – TRT21

O trecho de programa em PL/SQL a seguir possibilita aumentar o valor da variável *salary* em 15%, sempre que esta alcançar valor abaixo da variável *media\_salary*.

```
UPDATE servidor SET salary = salary *1.1  
WHERE salary < media_salary
```

Certo

 Errado

## Questão 35 – 2010 – VUNESP – CEAGESP

Considere o seguinte comando do PL/SQL:

```
DECLARE abc NUMBER := 20;
```

Esse comando declara uma variável chamada abc, do tipo NUMBER,

A) tendo seu valor mínimo igual a 20.

B) tendo seu valor máximo igual a 20.

C) com 20 posições.

D) com 20 casas após a vírgula.

E) e lhe atribui o valor 20.

## Questão 35 – 2010 – VUNESP – CEAGESP

Considere o seguinte comando do PL/SQL:

```
DECLARE abc NUMBER := 20;
```

Esse comando declara uma variável chamada abc, do tipo NUMBER,

A) tendo seu valor mínimo igual a 20.

B) tendo seu valor máximo igual a 20.

C) com 20 posições.

D) com 20 casas após a vírgula.

➡ E) e lhe atribui o valor 20.

# Questões Diversas



**PL/SQL**  
programming



## Questão 36 – 2010 – FCC – TRT20

A principal diferença na utilização de comandos SQL em blocos PL/SQL está na utilização do comando SELECT, com uma cláusula para colocar o resultado da consulta em variáveis. Tal cláusula é

A) INTO.

B) FETCH

C) ORDER BY.

D) HAVING.

E) GROUP BY.

## Questão 36 – 2010 – FCC – TRT20

A principal diferença na utilização de comandos SQL em blocos PL/SQL está na utilização do comando SELECT, com uma cláusula para colocar o resultado da consulta em variáveis. Tal cláusula é

- ➡ A) INTO.
- B) FETCH
- C) ORDER BY.
- D) HAVING.
- E) GROUP BY.

## Questão 37 – 2010 – CESPE – TRT21

Julgue o item abaixo, relativo à linguagem PL/SQL.

Caso seja utilizado o comando de manipulação de dados SELECT, da linguagem de consulta estruturada (SQL), em um bloco PL/SQL, deverá ser incluído, obrigatoriamente, a cláusula INTO.

Certo

Errado

## Questão 37 – 2010 – CESPE – TRT21

Julgue o item abaixo, relativo à linguagem PL/SQL.

Caso seja utilizado o comando de manipulação de dados SELECT, da linguagem de consulta estruturada (SQL), em um bloco PL/SQL, deverá ser incluído, obrigatoriamente, a cláusula INTO.

 Certo

Errado

## Questão 38 – 2010 – FGV – BADESC

A PL/SQL é uma linguagem procedural que estende a linguagem padrão SQL e sua unidade básica é um bloco.

A seção utilizada para declaração de variáveis, tipos e subprogramas locais de um bloco PL/SQL é definida pela palavra reservada:

- A) var.
- B) using.
- C) define.
- D) include.
- E) declare.

## Questão 38 – 2010 – FGV – BADESC

A PL/SQL é uma linguagem procedural que estende a linguagem padrão SQL e sua unidade básica é um bloco.

A seção utilizada para declaração de variáveis, tipos e subprogramas locais de um bloco PL/SQL é definida pela palavra reservada:

A) var.

B) using.

C) define.

D) include.

➡ E) declare.

## Questão 39 – 2014 – FCC – TJ/AP

Suponha que tenha sido criada uma *Procedure* na linguagem PL-SQL do Oracle 11g, denominada Paper, sem parâmetros de entrada. Uma das formas de invocar essa *Procedure* é por meio do comando

- A) SET Paper (\_, \_) . . .
- B) JOIN Paper . . .
- C) CURSOR Paper . . .
- D) CALL Paper . . .
- E) UNIQUE Paper (null) . . .

## Questão 39 – 2014 – FCC – TJ/AP

Suponha que tenha sido criada uma *Procedure* na linguagem PL-SQL do Oracle 11g, denominada Paper, sem parâmetros de entrada. Uma das formas de invocar essa *Procedure* é por meio do comando

A) SET Paper (\_, \_) . . .

B) JOIN Paper . . .

C) CURSOR Paper . . .

➡ D) CALL Paper . . .

E) UNIQUE Paper (null) . . .



## Questão 40 – 2010 – VUNESP - CREMESP

No PL/SQL do sistema gerenciador de bancos de dados Oracle 10g, há a opção de se colocar *labels* em um *loop*. A sintaxe adequada para a inserção de *labels* é:

- A) < label >
- B) |– label –|
- C) << label >>
- D) -- label --
- E) \*\* label \*\*

## Questão 40 – 2010 – VUNESP - CREMESP

No PL/SQL do sistema gerenciador de bancos de dados Oracle 10g, há a opção de se colocar *labels* em um *loop*. A sintaxe adequada para a inserção de *labels* é:

A) < label >

B) |– label –|

➡ C) << label >>

D) -- label --

E) \*\* label \*\*

## Questão 41 – 2010 – VUNESP - CREMESP

Considere o seguinte trecho de um comando do tipo **CASE** no PL/SQL do sistema gerenciador de bancos de dados Oracle 10 g.

```
CASE teste  
  X 'A' ...  
  X 'B' ...  
  ELSE ...  
END CASE
```

O valor correto de **X** para completar o comando **CASE** é:

- A) NULL.
- B) WHEN.
- C) ENDIF.
- D) ELSEIF.
- E) IF THEN.

## Questão 41 – 2010 – VUNESP - CREMESP

Considere o seguinte trecho de um comando do tipo **CASE** no PL/SQL do sistema gerenciador de bancos de dados Oracle 10 g.

```
CASE teste  
  X 'A' ...  
  X 'B' ...  
  ELSE ...  
END CASE
```

O valor correto de **X** para completar o comando **CASE** é:

A) NULL.

➡ B) WHEN.

C) ENDIF.

D) ELSEIF.

E) IF THEN.

## Questão 42 – 2010 – VUNESP - CREMESP

No PL/SQL do sistema gerenciador de bancos de dados Oracle 10 g, os cursores do tipo *explícito* possuem 4 atributos. O atributo utilizado para verificar se um cursor está aberto é o

- A) %GET.
- B) %FOUND.
- C) %ISOPEN.
- D) %ISFIRST.
- E) %NOTOPEN.

## Questão 42 – 2010 – VUNESP - CREMESP

No PL/SQL do sistema gerenciador de bancos de dados Oracle 10 g, os cursores do tipo *explícito* possuem 4 atributos. O atributo utilizado para verificar se um cursor está aberto é o

A) %GET.

B) %FOUND.

➡ C) %ISOPEN.

D) %ISFIRST.

E) %NOTOPEN.

## Questão 43 – 2008 – FCC – MPE/RS

A estrutura de repetição PL/SQL que executa uma seqüência de comandos por um número fixo de vezes, estabelecido na própria estrutura, é implementada pelo comando

*A) FOR LOOP.*

*B) WHILE LOOP.*

*C) LOOP EXIT.*

*D) LOOP WHEN.*

*E) CASE LOOP.*

# Controles Iterativos

- LOOP 

```
LOOP
    --ação que deverá ser executada
END LOOP;
```
- FOR-LOOP 

```
FOR i in 1..100 LOOP
    dbms_output.put_line(i);
END LOOP;
```
- WHILE-LOOP 

```
WHILE sal <= 10000 LOOP
    --atualiza a tabela com o aumento de salário
END LOOP;
```
- EXIT-WHEN 

```
LOOP
    --ação que deverá ser executada
    EXIT WHEN valor > 100;
END LOOP;
```
- CONTINUE\_WHEN 

```
LOOP
    --ação que deverá ser executada
    CONTINUE WHEN valor > 100;
END LOOP;
```



## Questão 43 – 2008 – FCC – MPE/RS

A estrutura de repetição PL/SQL que executa uma seqüência de comandos por um número fixo de vezes, estabelecido na própria estrutura, é implementada pelo comando

➡ *A) FOR LOOP.*

*B) WHILE LOOP.*

*C) LOOP EXIT.*

*D) LOOP WHEN.*

*E) CASE LOOP.*

A estrutura de controle iteração pode ser utilizada em PL/SQL com os comandos

A) LOOP, CASE-LOOP, WHILE-LOOP e FOR-LOOP.

B) LOOP, CASE-LOOP e WHILE-LOOP.

C) LOOP, CASE-LOOP e FOR-LOOP.

D) CASE-LOOP, WHILE-LOOP e FOR-LOOP.

E) LOOP, WHILE-LOOP e FOR-LOOP.

A estrutura de controle iteração pode ser utilizada em PL/SQL com os comandos

A) LOOP, CASE-LOOP, WHILE-LOOP e FOR-LOOP.

B) LOOP, CASE-LOOP e WHILE-LOOP.

C) LOOP, CASE-LOOP e FOR-LOOP.

D) CASE-LOOP, WHILE-LOOP e FOR-LOOP.

➡ E) LOOP, WHILE-LOOP e FOR-LOOP.

Nas estruturas de controle de iteração dos programas PL/SQL são utilizadas apenas as instruções

- A) LOOP e IF.
- B) IF e CASE.
- C) GOTO e NULL.
- D) CASE e GOTO.
- E) LOOP e EXIT.

Nas estruturas de controle de iteração dos programas PL/SQL são utilizadas apenas as instruções

A) LOOP e IF.

B) IF e CASE.

C) GOTO e NULL.

D) CASE e GOTO.

➡ E) LOOP e EXIT.

## Questão 46 – 2010 – VUNESP - CREMESP

Considerando o PL/SQL do sistema gerenciador de bancos de dados Oracle 10 g, indique a alternativa que apresenta o formato de um comentário válido.

- A) /\* comentário do PL/SQL \*/
- B) || comentário do PL/SQL ||
- C) << comentário do PL/SQL >>
- D) <= comentário do PL/SQL =>
- E) % comentário do PL/SQL %

## Questão 46 – 2010 – VUNESP - CREMESP

Considerando o PL/SQL do sistema gerenciador de bancos de dados Oracle 10 g, indique a alternativa que apresenta o formato de um comentário válido.

- ➡ A) /\* comentário do PL/SQL \*/
- B) || comentário do PL/SQL ||
- C) << comentário do PL/SQL >>
- D) <= comentário do PL/SQL =>
- E) % comentário do PL/SQL %

## Questão 47 – 2013 - FEPESE - JUCESC

Assinale a alternativa que indica corretamente o comando PL/SQL que lista os atributos com respectivos tipos de dados de uma dada tabela.

- A) GET
- B) LIST
- C) SHOW
- D) SELECT
- E) DESCRIBE



## Questão 47 – 2013 - FEPESE - JUCESC

Assinale a alternativa que indica corretamente o comando PL/SQL que lista os atributos com respectivos tipos de dados de uma dada tabela.

A) GET

B) LIST

C) SHOW

D) SELECT

➡ E) DESCRIBE

No PL/SQL, é um laço utilizado quando não se tem certeza de quantas vezes o ciclo deve ocorrer e não se quer executar o laço no mínimo uma vez:

- A) simples.
- B) while.
- C) infinito.
- D) numérico.
- E) reverso.

No PL/SQL, é um laço utilizado quando não se tem certeza de quantas vezes o ciclo deve ocorrer e não se quer executar o laço no mínimo uma vez:

A) simples.

➡ B) while.

C) infinito.

D) numérico.

E) reverso.

É INCORRETO dizer que dentre as declarações DDL (Data Definition Language ) presentes em PL/SQL Oracle, se inclua

- A) ALTER TABLE.
- B) MERGE.
- C) GRANT.
- D) RENAME.
- E) REVOKE.

É INCORRETO dizer que dentre as declarações DDL (Data Definition Language ) presentes em PL/SQL Oracle, se inclua

A) ALTER TABLE.

➡ B) MERGE.

C) GRANT.

D) RENAME.

E) REVOKE.

As pseudo colunas SQL: CURRVAL, NIVEL, NEXTVAL, ROWID, e ROWNUM são permitidas em PL/SQL somente no comando

- A) delete.
- B) update.
- C) select.
- D) insert.
- E) de teste condicional.

As pseudo colunas SQL: CURRVAL, NIVEL, NEXTVAL, ROWID, e ROWNUM são permitidas em PL/SQL somente no comando

A) delete.

B) update.

➡ C) select.

D) insert.

E) de teste condicional.

# Gabarito

## Fundamentos

1 - ERRADO	9 - E	17 - A
2 - E	10 - E	18 - CERTO
3 - B	11 - E	19 - C
4 - CERTO	12 - CERTO	20 - B
5 - A	13 - E	21 - C
6 - E	14 - B	22 - A
7 - A	15 - B	23 - C
8 - C	16 - E	24 - C

## Triggers

25 - B	27 - CERTO
26 - A	28 - D

## Funções

24 - A	27 - E
25 - D	28 - D
26 - B	

## Scripts

29 - D	33 - A
30 - A	34 - ERRADO
31 - C	35 - E
32 - A	

## Diversas

36 - A	43 - A
37 - CERTO	44 - E
38 - E	45 - E
39 - D	46 - A
40 - C	47 - E
41 - B	48 - B
42 - C	49 - B
	50 - C