

Banco de Dados – Oracle PL/SQL

Prof. Washington Almeida, MSc, ISF 27002

PL/SQL

- **PL/SQL** é a **linguagem procedural** do banco de dados Oracle para SQL.
- O **PL/SQL** está integrado ao banco de dados, suportando todas as instruções, funções e tipos de dados Oracle SQL.
- Os aplicativos gravados nas APIs do banco de dados podem chamar subprogramas armazenados **PL/SQL** e enviar blocos de código **PL/SQL** ao banco de dados para execução.

PL/SQL

- O **PL/SQL** fornece uma linguagem processual armazenada no servidor, fácil de usar, integrada com SQL, robusta, portátil e segura.
- É possível **acessar e manipular dados** do banco de dados usando objetos procedurais chamados unidades **PL/SQL**.
- O compilador e intérprete **PL/SQL** estão incorporados no **Oracle SQL Developer**, oferecendo aos desenvolvedores um modelo de desenvolvimento consistente e avançado no cliente e no servidor.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** TRT - 2ª REGIÃO (SP) **Prova:** FCC - 2018 - TRT - 2ª REGIÃO (SP) - Técnico Judiciário - Tecnologia da Informação

Em um banco de dados Oracle aberto e em condições ideais há uma tabela chamada func de onde se deseja exibir os nomes dos funcionários (campo nome) e os códigos de departamentos (campo codDep) de todos os funcionários dos departamentos 10 ou 20 em ordem alfabética crescente por nome. Para isso, utilizando PL/SQL, um Técnico de TI deve utilizar a instrução SELECT nome, codDep FROM func WHERE:

- a) codDep IN (10,20) ORDER BY nome ASC;
- b) codDep CONTAIN(10 OR 20) ORDER BY nome ASC;
- c) codDep=10 OR codDep=20 ORDER ASC nome;
- d) codDep BELONG(10 OR 20) ORDER BY nome ASCENDING;
- e) codDep BETWEEN(10,20) ORDER BY nome ASC;

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** TRT - 2ª REGIÃO (SP) **Prova:** FCC - 2018 - TRT - 2ª REGIÃO (SP) - Técnico Judiciário - Tecnologia da Informação

Em um banco de dados Oracle aberto e em condições ideais há uma tabela chamada func de onde se deseja exibir os nomes dos funcionários (campo nome) e os códigos de departamentos (campo codDep) de todos os funcionários dos departamentos 10 ou 20 em ordem alfabética crescente por nome. Para isso, utilizando PL/SQL, um Técnico de TI deve utilizar a instrução SELECT nome, codDep FROM func WHERE:

- a) **codDep IN (10,20) ORDER BY nome ASC;**
- b) codDep CONTAIN(10 OR 20) ORDER BY nome ASC;
- c) codDep=10 OR codDep=20 ORDER ASC nome;
- d) codDep BELONG(10 OR 20) ORDER BY nome ASCENDING;
- e) codDep BETWEEN(10,20) ORDER BY nome ASC;

PL/SQL

- As unidades **PL/SQL** geralmente são categorizadas da seguinte maneira:
 - Um **subprograma PL/SQL** é um bloco **PL/SQL** armazenado no banco de dados e pode ser chamado pelo nome a partir de um aplicativo. Quando um subprograma é criado, o banco de dados analisa o subprograma e armazena sua representação analisada no banco de dados. É possível declarar um subprograma como um procedimento ou uma função.
 - Um **bloco anônimo PL/SQL** é um bloco **PL/SQL** que aparece no aplicativo e não é nomeado ou armazenado no banco de dados.

SUBPROGRAMA

- Um **subprograma PL/SQL** é um bloco **PL/SQL** nomeado que permite que o chamador forneça parâmetros que podem ser apenas de entrada, saída apenas ou valores de entrada e saída.
- Um **subprograma** é um **procedimento PL/SQL** ou uma **função PL/SQL**.
- **Procedimentos** e **funções** são idênticos, exceto que as funções sempre retornam um valor único para o chamador, enquanto os procedimentos não.

SUBPROGRAMA

- As vantagens da utilização de subprogramas **PL/SQL** incluem:
 - Performance melhorada.
 - Alocação de memória.
 - Maior produtividade.
 - Integridade.
 - Segurança.
 - Privilégios herdados.

SUBPROGRAMA

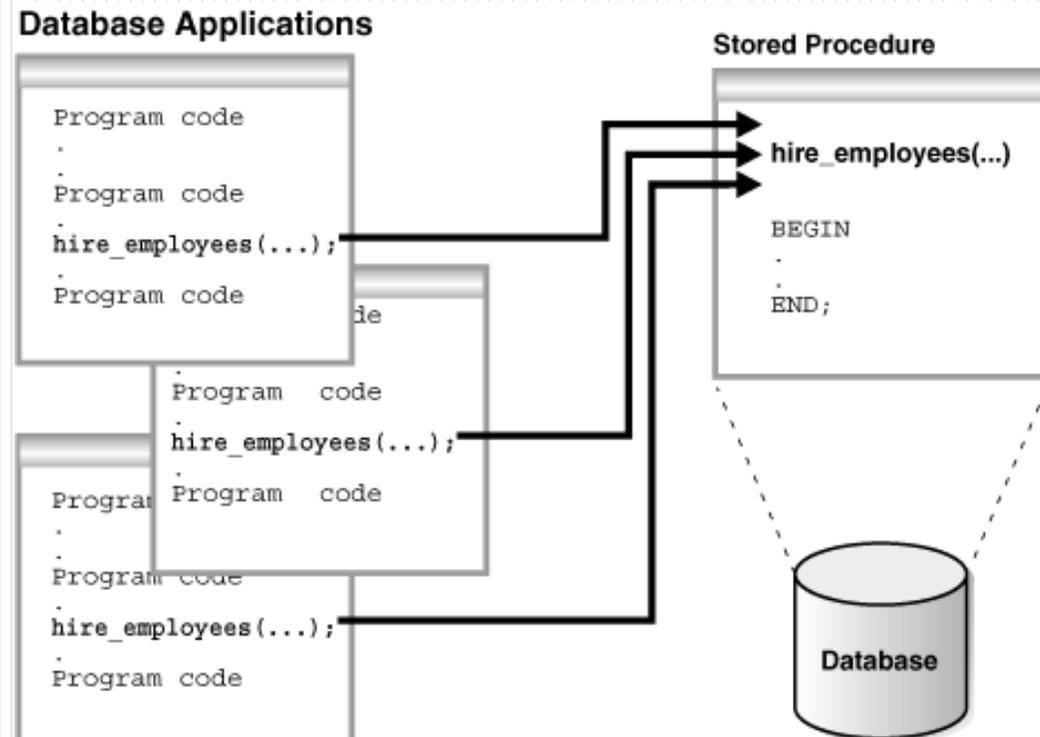
- Um subprograma armazenado independente é um subprograma criado no nível do esquema com a instrução **CREATE PROCEDURE** ou **CREATE FUNCTION**.
- Os subprogramas definidos em um pacote são chamados **subprogramas de pacote** e são considerados parte do pacote.
- O banco de dados armazena subprogramas no **dicionário de dados** como objetos de esquema.

SUBPROGRAMA

```
CREATE PROCEDURE hire_employees
  (p_last_name VARCHAR2, p_job_id VARCHAR2, p_manager_id NUMBER,
   p_hire_date DATE, p_salary NUMBER, p_commission_pct NUMBER,
   p_department_id NUMBER)
IS
BEGIN
.
.
.
  INSERT INTO employees (employee_id, last_name, job_id, manager_id, hire_date,
    salary, commission_pct, department_id)
  VALUES (emp_sequence.NEXTVAL, p_last_name, p_job_id, p_manager_id,
    p_hire_date, p_salary, p_commission_pct, p_department_id);
.
.
.
END;
```

SUBPROGRAMA

- Os usuários podem executar um subprograma interativamente de várias maneiras.



SUBPROGRAMA

- Um procedimento armazenado depende dos objetos mencionados em seu corpo.
- A instrução de exemplo insere um novo registro **TSMITH** na tabela **employees**.
- O banco de dados rastreia e gerencia automaticamente essas dependências.

```
EXECUTE hire_employees ('TSMITH', 'CLERK', 1037, SYSDATE, 500, NULL, 20);
```

PACOTES

- Um **pacote PL/SQL** é um grupo de subprogramas relacionados, juntamente com os cursores e variáveis que eles usam, armazenados juntos no banco de dados para uso continuado como uma unidade.
- Os **subprogramas empacotados** podem ser chamados explicitamente por aplicativos ou usuários.

PACOTES

- O **Oracle Database** inclui muitos pacotes fornecidos que estendem a funcionalidade do banco de dados e fornecem acesso **PL/SQL** aos recursos SQL.
- As vantagens dos **pacotes PL/SQL** incluem:
 - Encapsulamento.
 - Segurança de dados.
 - Melhor performance.

PACOTES

- O **pacote PL/SQL** é criado em duas partes: a **especificação** e o **corpo**.
- A **especificação** do pacote declara todas as construções públicas do pacote, enquanto o **corpo** do pacote define todas as construções (públicas e privadas) do pacote.

PACOTES

- O exemplo mostra parte de uma instrução que cria a especificação do pacote **employees_management**.

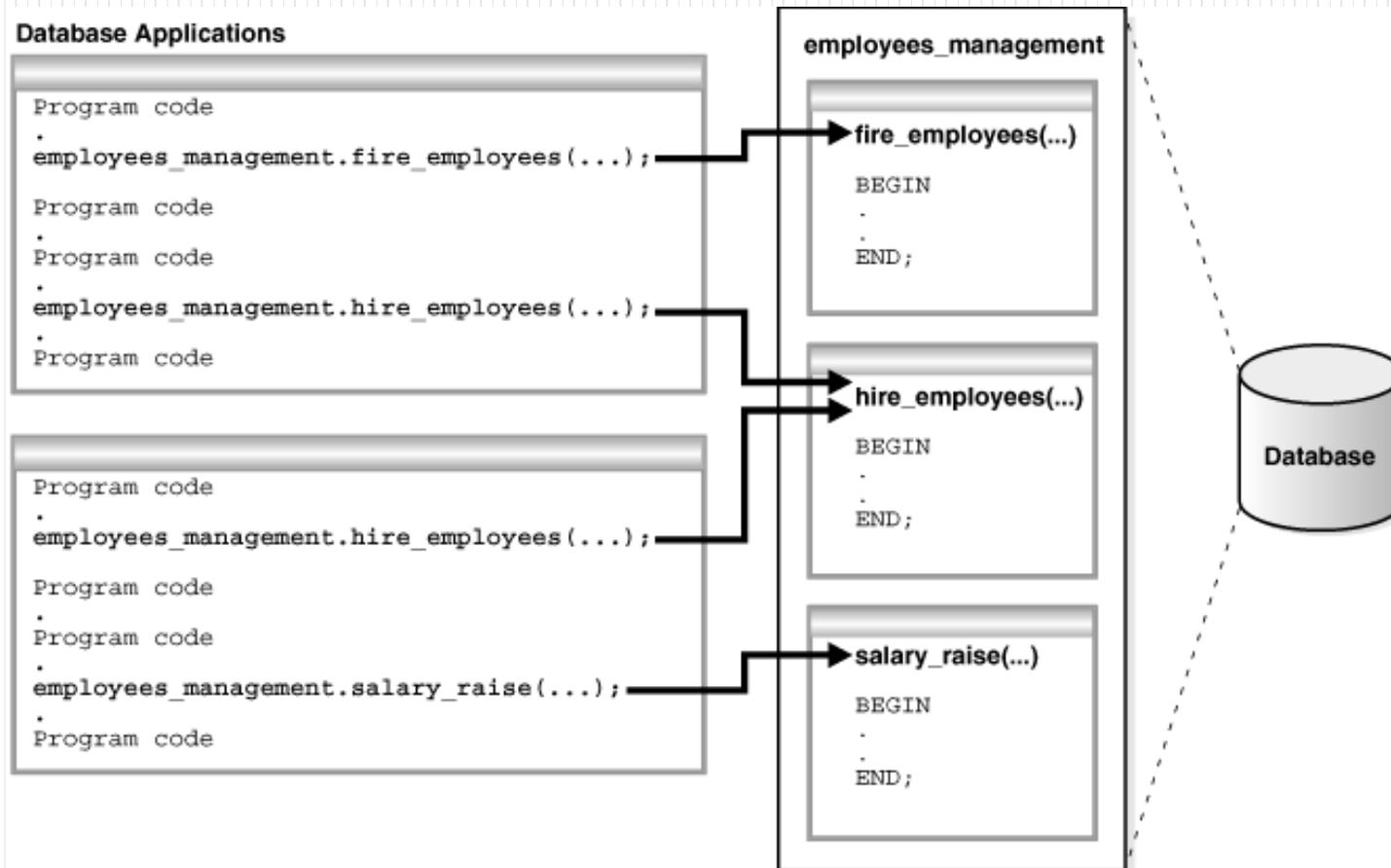
```
CREATE PACKAGE employees_management AS
  FUNCTION hire_employees (last_name VARCHAR2, job_id VARCHAR2, manager_id NUMBER,
    salary NUMBER, commission_pct NUMBER, department_id NUMBER) RETURN NUMBER;
  PROCEDURE fire_employees(employee_id NUMBER);
  PROCEDURE salary_raise(employee_id NUMBER, salary_incr NUMBER);
  .
  .
  .
  no_sal EXCEPTION;
END employees_management;
```

PACOTES

- A instrução **CREATE PACKAGE BODY** define objetos declarados na especificação.
- O corpo do pacote deve ser criado no mesmo esquema que o pacote.

PACOTES

- Execução de Subprogramas de Pacote PL/SQL.



BLOCOS ANÔNIMOS

- Um **bloco anônimo PL/SQL** é uma unidade **PL/SQL** não persistente e sem nome.
- Os usos típicos para **blocos anônimos** incluem:
 - Iniciando chamadas para subprogramas e construções de pacotes.
 - Isolando o tratamento de exceções.
 - Gerenciando o controle aninhando código em outros blocos PL/SQL.
- **Blocos anônimos** não têm as vantagens de reutilização de código dos subprogramas armazenados.

BLOCOS ANÔNIMOS

É a unidade PL / SQL ...	Blocos Anônimos	Subprogramas
Especificado com um nome?	Não	sim
Compilado com cada reutilização?	Não	Não
Armazenado no banco de dados?	Não	sim
Invocável por outras aplicações?	Não	sim
Capaz de retornar valores variáveis de ligação?	sim	sim
Capaz de retornar valores de função?	Não	sim
Capaz de aceitar parâmetros?	Não	sim

BLOCOS ANÔNIMOS

- Um **bloco anônimo** consiste em uma parte declarativa opcional, uma parte executável e um ou mais manipuladores de exceção opcionais.

```
DECLARE
  v_lname VARCHAR2(25);
BEGIN
  SELECT last_name
     INTO v_lname
  FROM employees
 WHERE employee_id = 101;
  DBMS_OUTPUT.PUT_LINE('Employee last name is '||v_lname);
END;
```

BLOCOS ANÔNIMOS

- O **Oracle Database** compila o bloco **PL/SQL** e o coloca no pool compartilhado do SGA, mas não armazena o código-fonte ou a versão compilada no banco de dados para reutilização além da instância atual.
- Diferentemente dos gatilhos, um **bloco anônimo** é compilado toda vez que é carregado na memória.
- O SQL compartilhado permite que os **blocos anônimos PL/SQL** no pool compartilhado sejam reutilizados e compartilhados até que sejam liberados do pool compartilhado.

PL/SQL

- **Construções de Linguagem PL/SQL.**
- Os blocos **PL/SQL** podem incluir uma variedade de construções de linguagem **PL/SQL** diferentes.
- **Variáveis e Constantes:** essas construções podem ser declaradas em um procedimento, função ou pacote.
- **Cursor:** um cursor pode ser declarado explicitamente em um procedimento, função ou pacote para facilitar o processamento orientado a registros dos dados do banco de dados Oracle.
- **Exceções:** o PL/SQL permite lidar explicitamente com as condições de erro internas e definidas pelo usuário, chamadas exceções.

PL/SQL

- **Construções de Linguagem PL/SQL.**
- O **PL/SQL** pode executar instruções **SQL dinâmicas** cujo texto completo não é conhecido até o tempo de execução.
- As instruções **SQL dinâmicas** são armazenadas em cadeias de caracteres inseridas ou criadas pelo programa em tempo de execução.

PL/SQL

- **Coleções e Registros PL/SQL.**
- O PL/SQL fornece os tipos de dados **TABLE** e **VARRAY**, que permitem declarar matrizes associativas, tabelas aninhadas e matrizes de tamanho variável.
- Uma **coleção PL/SQL** é um grupo ordenado de elementos, todos do mesmo tipo.
- Cada elemento possui um índice exclusivo que determina sua posição na **coleção**.
- As **coleções** funcionam como as matrizes.

PL/SQL

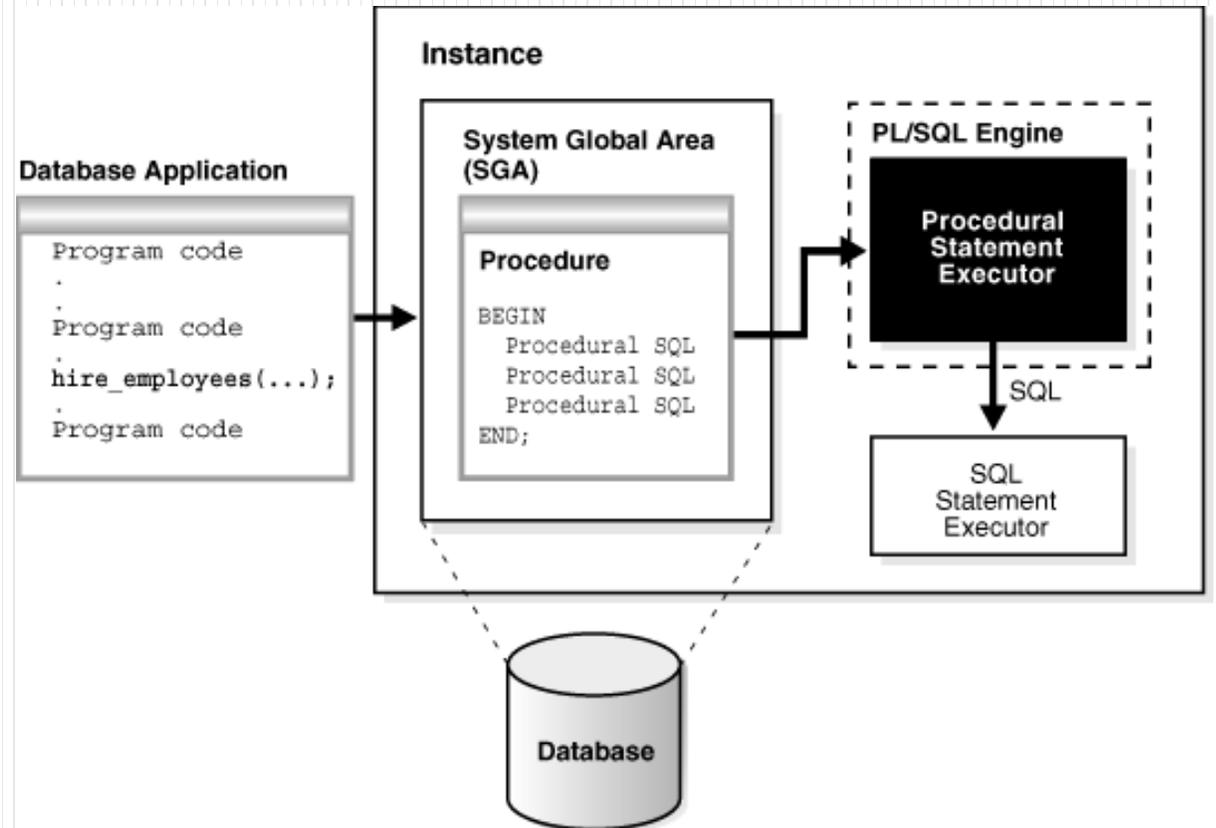
- **Coleções e Registros PL/SQL.**
- Um **registro PL/SQL** é uma variável composta que pode armazenar valores de dados de tipos diferentes.
- Os registros são úteis para reter dados das linhas da tabela ou determinadas colunas das linhas da tabela.

PL/SQL

- **Execução do PL/SQL.**
- O **PL/SQL** suporta **execução interpretada** e **execução nativa**.
- Na **execução interpretada**, o código-fonte **PL/SQL** é compilado na chamada representação de bytecode.
- A **execução nativa** oferece o melhor desempenho em unidades computacionalmente intensivas.

PL/SQL

- **Execução do PL/SQL.**
- O mecanismo **PL/SQL** define, compila e executa unidades PL/SQL.
- Esse mecanismo é um componente especial de muitos produtos Oracle, incluindo o **Oracle Database**.



PL/SQL

- **Execução do PL/SQL.**
- A unidade **PL/SQL** é armazenada em um banco de dados.
- Quando um aplicativo chama um **procedimento armazenado**, o banco de dados carrega a unidade **PL/SQL** compilada no pool compartilhado na área global do sistema (SGA).
- Os executores de instruções **PL/SQL** e SQL trabalham juntos para processar as instruções no procedimento.
- É possível chamar um **procedimento armazenado** de outro bloco PL/SQL, que pode ser um **bloco anônimo** ou outro **procedimento armazenado**.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** Gestão Concurso **Órgão:** EMATER-MG **Prova:** Gestão Concurso - 2018 - EMATER-MG - Analista de Sistemas I

Informe se é verdadeiro (V) ou falso (F) o que se afirma a seguir sobre *Procedural Language Extension to SQL* (PL/SQL).

- () Os programas são combinados em blocos.
- () Comentários podem ser somente de linha única.
- () Faz distinção entre letras maiúsculas e minúsculas.
- () É utilizada para o desenvolvimento do lado do servidor e do lado do cliente.

De acordo com as afirmações, a sequência correta é:

- a) (F); (V); (V); (V).
- b) (V); (V); (F); (F).
- c) (V); (F); (F); (V).
- d) (F); (F); (V); (F).

DIRETO DO CONCURSO

Ano: 2018 **Banca:** Gestão Concurso **Órgão:** EMATER-MG **Prova:** Gestão Concurso - 2018 - EMATER-MG - Analista de Sistemas I

Informe se é verdadeiro (V) ou falso (F) o que se afirma a seguir sobre *Procedural Language Extension to SQL* (PL/SQL).

- () Os programas são combinados em blocos.
- () Comentários podem ser somente de linha única.
- () Faz distinção entre letras maiúsculas e minúsculas.
- () É utilizada para o desenvolvimento do lado do servidor e do lado do cliente.

De acordo com as afirmações, a sequência correta é:

- a) (F); (V); (V); (V).
- b) (V); (V); (F); (F).
- c) **(V); (F); (F); (V).**
- d) (F); (F); (V); (F).

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** DPE-AM **Prova:** FCC - 2018 - DPE-AM - Assistente Técnico de Defensoria - Programador

Considere o código PL/SQL abaixo.

```
DROP TABLE tab;
CREATE TABLE tab (ch CHAR(3 CHAR));

DECLARE
    str VARCHAR2(5 CHAR) := 'DPEAM';
BEGIN
    INSERT INTO tab(ch) VALUES(str);
END;
/
```

Executar o código acima no Oracle 12c, em condições ideais, resulta em

- a) PL/SQL *procedure successfully completed.*
- b) PL/SQL: *numeric or value error: character string buffer too large or too small.*
- c) ORACLE *table successfully created.*
- d) ERROR: *character string buffer too small.*
- e) ERROR: *value too large for column.*

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** DPE-AM **Prova:** FCC - 2018 - DPE-AM - Assistente Técnico de Defensoria - Programador

Considere o código PL/SQL abaixo.

```
DROP TABLE tab;
CREATE TABLE tab (ch CHAR(3 CHAR));

DECLARE
    str VARCHAR2(5 CHAR) := 'DPEAM';
BEGIN
    INSERT INTO tab(ch) VALUES(str);
END;
/
```

Executar o código acima no Oracle 12c, em condições ideais, resulta em

- a) PL/SQL *procedure successfully completed.*
- b) PL/SQL: *numeric or value error: character string buffer too large or too small.*
- c) ORACLE *table successfully created.*
- d) ERROR: *character string buffer too small.*
- e) **ERROR: value too large for column.**

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** DPE-AM **Prova:** FCC - 2018 - DPE-AM - Assistente Técnico de Defensoria - Programador

Considere o código PL/SQL abaixo.

Ao executar o código acima no Oracle 12c em condições ideais, o resultado será

- a) bônus = 0
bônus = 50
- b) bônus = 50
bônus = 500
- c) bônus = 50
bônus = 3000
- d) bônus = 0
bônus = 500
- e) um erro de execução, pois deveria ser usado o comando SELECT e não UPDATE.

```
DECLARE
  PROCEDURE calc (
    vendas NUMBER,
    quota NUMBER,
    emp_id NUMBER
  )
  IS
    bonus NUMBER := 0;
  BEGIN
    IF vendas > (quota + 200) THEN
      bonus := (vendas - quota)/4;
    ELSE
      bonus := 50;
    END IF;

    DBMS_OUTPUT.PUT_LINE('bônus = ' || bonus);

    UPDATE tabEmpregados
    SET salario = salario + bonus
    WHERE empregado_id = emp_id;
  END calc;
BEGIN
  calc(10100, 10000, 120);
  calc(12000, 10000, 121);
END;
/
```

DIRETO DO CONCURSO

Ano: 2018 Banca: FCC Órgão: DPE-AM Prova: FCC - 2018 - DPE-AM - Assistente Técnico de Defensoria - Programador

Considere o código PL/SQL abaixo.

Ao executar o código acima no Oracle 12c em condições ideais, o resultado será

- a) bônus = 0
bônus = 50
- b) **bônus = 50**
bônus = 500
- c) bônus = 50
bônus = 3000
- d) bônus = 0
bônus = 500
- e) um erro de execução, pois deveria ser usado o comando SELECT e não UPDATE.

```
DECLARE
  PROCEDURE calc (
    vendas NUMBER,
    quota NUMBER,
    emp_id NUMBER
  )
  IS
    bonus NUMBER := 0;
  BEGIN
    IF vendas > (quota + 200) THEN
      bonus := (vendas - quota)/4;
    ELSE
      bonus := 50;
    END IF;

    DBMS_OUTPUT.PUT_LINE('bônus = ' || bonus);

    UPDATE tabEmpregados
    SET salario = salario + bonus
    WHERE empregado_id = emp_id;
  END calc;
BEGIN
  calc(10100, 10000, 120);
  calc(12000, 10000, 121);
END;
/
```

DIRETO DO CONCURSO

Ano: 2020 Banca: CESPE Órgão: TJ-PA Prova: CESPE - 2020 - TJ-PA - Analista Judiciário – Programador

```
1.  create or replace trigger TJ_PA.ATUALIZA_CARTORIO INSTEAD OF INSERT OR UPDATE ON
    TJ_PA.CARTORIO FOR EACH ROW
2.  BEGIN
3.      IF USER <> 'TJ_PA' THEN
4.          IF INSERTING THEN
5.              INSERT INTO TJ_PA.CARTORIO@TJBASE.TJPA.JUS.BR (CODIGO,NOME,CNPJ,NOME_TITULAR,DT_POSSE_TITULAR,
                TELEFONE,EMAIL,URL_HOMEPAGE)
6.                  VALUES (:NEW.CODIGO,:NEW.NOME,:NEW.CNPJ,:NEW.NOME_TITULAR,:NEW.DT_POSSE_TITULAR,:NEW.TELEFONE,
                :NEW.EMAIL,:NEW.URL_HOMEPAGE);
7.          END IF;
8.          IF UPDATING THEN
9.              UPDATE TJ_PA.CARTORIO@TJBASE.TJPA.JUS.BR
10.                 SET NOME=:new.NOME,CNPJ=:new.CNPJ,NOME_TITULAR=:new.NOME_TITULAR,
11.                   DT_POSSE_TITULAR=:new.DT_POSSE_TITULAR,
12.                   TELEFONE=:new.TELEFONE,EMAIL=:new.EMAIL,URL_HOMEPAGE=:new.URL_HOMEPAGE
13.                 WHERE CODIGO=:old.CODIGO;
14.          END IF;
15.      END IF;
16.  END;
```

Esse código Oracle PL/SQL

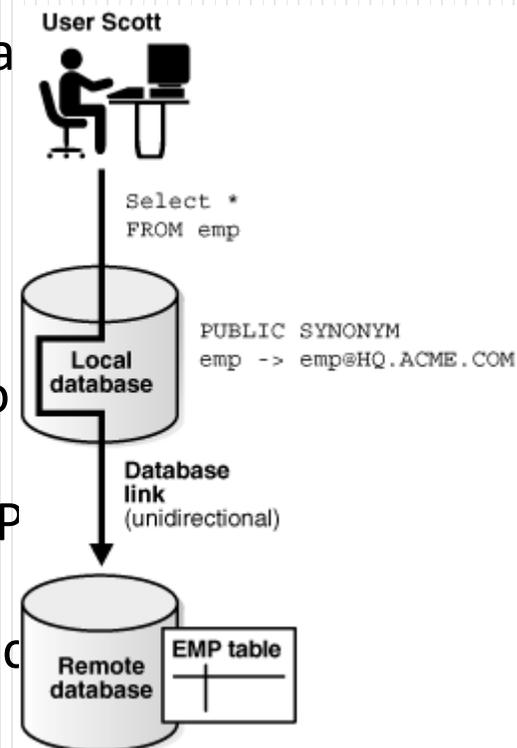
- a) faz uso de *database link*.
- b) tem a função de criar o objeto procedure no banco de dados Oracle
- c) apresenta um erro de sintaxe na linha 3, pois a linguagem PL/SQL não admite atribuir nomes idênticos para usuário e esquema.
- d) envia o conteúdo da tabela TJ_PA para o email CARTORIO@TJBASE.TJPA.JUS.BR, conforme instrução na linha 5.
- e) utiliza a tabela particionada ATUALIZA_CARTORIO, conforme indica o comando INSTEAD OF na linha 1.

DIRETO DO CONCURSO

Ano: 2020 Banca: CESPE Órgão: TJ-PA Prova: CESPE - 2020 - TJ-PA - Analista Programador

LETRA A

- a) faz uso de *database link*.
- b) tem a função de criar o objeto procedure no banco de dados Oracle
- c) apresenta um erro de sintaxe na linha 3, pois a linguagem PL/SQL não nomes idênticos para usuário e esquema.
- d) envia o conteúdo da tabela TJ_PA para o email CARTORIO@TJBASE.TJP instrução na linha 5.
- e) utiliza a tabela particionada ATUALIZA_CARTORIO, conforme indica o c OF na linha 1.



Justificativa:

Cria um **database link (DBLINK)** sendo possível utilizar e referenciar tabelas e visões do outro banco de dados, acrescentando ao final do nome destes objetos @nome_do_dblink.

https://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts002.htm

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** Prefeitura de São Luís - MA **Prova:** FCC - 2018 - Prefeitura de São Luís - MA - Auditor Fiscal de Tributos I - Tecnologia da Informação (TI)

Um Auditor digitou em um banco de dados Oracle, aberto e em condições ideais, a instrução PL/SQL:

```
SELECT id, nome FROM cidadao WHERE id = '001' FOR UPDATE ORDER BY id;
```

A cláusula FOR UPDATE

- a) indica que todos os cidadãos que possuem o valor 001 no campo id poderão ter os dados modificados por qualquer usuário do banco de dados.
- b) altera o valor do campo id do registro selecionado para 001, independentemente do valor existente nesse campo.
- c) indica apenas que o Auditor está selecionando o cidadão que possui valor 001 no campo id para executar uma operação de alteração nos dados.
- d) bloqueia as linhas na tabela cidadao onde o valor do campo id é 001 para que somente o Auditor possa alterar. O bloqueio será liberado apenas depois da emissão de uma instrução ROLLBACK ou COMMIT.
- e) indica que os registros que possuem o valor 001 no campo id serão transferidos para a área de transferência do banco de dados onde poderão ser executadas operações de alteração de dados.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** Prefeitura de São Luís - MA **Prova:** FCC - 2018 - Prefeitura de São Luís - MA - Auditor Fiscal de Tributos I - Tecnologia da Informação (TI)

Um Auditor digitou em um banco de dados Oracle, aberto e em condições ideais, a instrução PL/SQL:

```
SELECT id, nome FROM cidadao WHERE id = '001' FOR UPDATE ORDER BY id;
```

A cláusula FOR UPDATE

- a) indica que todos os cidadãos que possuem o valor 001 no campo id poderão ter os dados modificados por qualquer usuário do banco de dados.
- b) altera o valor do campo id do registro selecionado para 001, independentemente do valor existente nesse campo.
- c) indica apenas que o Auditor está selecionando o cidadão que possui valor 001 no campo id para executar uma operação de alteração nos dados.
- d) **bloqueia as linhas na tabela cidadao onde o valor do campo id é 001 para que somente o Auditor possa alterar. O bloqueio será liberado apenas depois da emissão de uma instrução ROLLBACK ou COMMIT.**
- e) indica que os registros que possuem o valor 001 no campo id serão transferidos para a área de transferência do banco de dados onde poderão ser executadas operações de alteração de dados.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** TRT - 2ª REGIÃO (SP) **Prova:** FCC - 2018 - TRT - 2ª REGIÃO (SP) - Analista Judiciário - Tecnologia da Informação

Para criar uma consulta que pesquise na tabela Trabalhadores de um banco de dados aberto e em condições ideais todos os nomes e telefones de trabalhadores cujos nomes sejam Sofia ou Sophia, um Analista de Sistemas deve utilizar a instrução PL/SQL SELECT nome, telefone FROM Trabalhadores WHERE

- a) REGEXP_LIKE (nome, '^So(f|ph)ia\$');
- b) nome = '^So(f|ph)ia\$';
- c) nome CONTAINS('^So(f|ph)ia\$');
- d) nome REGEXP('%So(f|ph)ia%');
- e) REGEXP(nome, '%So(f|ph)ia%');

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** TRT - 2ª REGIÃO (SP) **Prova:** FCC - 2018 - TRT - 2ª REGIÃO (SP) - Analista Judiciário - Tecnologia da Informação

Para criar uma consulta que pesquise na tabela Trabalhadores de um banco de dados aberto e em condições ideais todos os nomes e telefones de trabalhadores cujos nomes sejam Sofia ou Sophia, um Analista de Sistemas deve utilizar a instrução PL/SQL SELECT nome, telefone FROM Trabalhadores WHERE

- a) **REGEXP_LIKE (nome, '^So(f|ph)ia\$');**
- b) nome = '^So(f|ph)ia\$';
- c) nome CONTAINS('^So(f|ph)ia\$');
- d) nome REGEXP('%So(f|ph)ia%');
- e) REGEXP(nome, '%So(f|ph)ia%');

DIRETO DO CONCURSO

JUSTIFICATIVA:

Para utilização de Expressões Regulares, se faz necessário conhecer alguns meta caracteres (entenda por “curingas” ou “alias”) que são encontrados na utilização de ERs. Para utilização de expressões regulares existem vários meta caracteres que devem ser conhecidos, entre eles:

Metacaracter ----> Função ----> Exemplo

\ ----> Escape ----> \\ Corresponde a \

^ ----> Posição inicial da Linha (string) ----> ^A "procura as strings que iniciem com A"

\$ ----> Posição final da Linha (String) ----> \$B "procura as strings que terminem com B"

[] ----> Lista ----> [abc] a string pode conter qualquer dos caracteres da lista

. ----> Qualquer caractere ----> Funciona como um "coringa" traz qualquer resultado

{ } ----> Número de caracteres

| ----> Alternativa (similar a ou) ----> Alternativa de padrão de busca, por exemplo ('casa' | 'prédio')

REGEXP_LIKE(*x*, *padrão*[*opção_correspondente*]) - Realiza a busca em *x* da expressão regular definida no parâmetro **padrão**. **Opção_correspondente** pode ser definida com os seguintes valores:

'c' - Especifica correspondência com diferenciação de maiúsculas e minúsculas(Padrão – case-Sensitive) 'i' - Especifica a busca sem diferenciação de caracteres maiúsculos e minúsculos(case-Insensitive) 'n' - Que permite usar o operador de correspondência com qualquer caractere. 'm' - Que trata *x* como uma linha múltipla. 'x' - ignora espaços em branco.

A semelhança com o comando LIKE não é mera coincidência, mas o leitor já poderá notar a maior flexibilidade no uso de **REGEXP_LIKE** em relação ao operador LIKE.

DIRETO DO CONCURSO

Ano: 2020 **Banca:** INSTITUTO AOCP **Órgão:** MJSP **Provas:** INSTITUTO AOCP - 2020 - MJSP - Cientista de Dados - Big Data

A linguagem PL/SQL possui um recurso denominado PL/SQL Table. É correto afirmar que esse recurso

Alternativas

- a) permite a construção de uma estrutura capaz de realizar uma varredura em uma tabela, linha por linha, coluna por coluna. O comando 'cursor' é o responsável por criar essa estrutura.
- b) faz uso de dados intrínsecos por meio de vetores homogêneos. O comando 'type' é o responsável por criar essa estrutura.
- c) cria variáveis que podem ser acessadas por meio de referências. O comando 'def' é o responsável por definir essa estrutura.
- d) é dinâmico e oferece a possibilidade de executar um comando SQL a partir de uma string. O comando 'declare' é o responsável por definir essa estrutura.
- e) disponibiliza um comando que permite fazer iterações de execução do código tantas vezes quantas forem necessárias. O comando 'loop' é o responsável por criar essa estrutura.

DIRETO DO CONCURSO

Ano: 2020 **Banca:** INSTITUTO AOCP **Órgão:** MJSP **Provas:** INSTITUTO AOCP - 2020 - MJSP - Cientista de Dados - Big Data

A linguagem PL/SQL possui um recurso denominado PL/SQL Table. É correto afirmar que esse recurso

Alternativas

- a) permite a construção de uma estrutura capaz de realizar uma varredura em uma tabela, linha por linha, coluna por coluna. O comando 'cursor' é o responsável por criar essa estrutura.
- b) faz uso de dados intrínsecos por meio de vetores homogêneos. O comando 'type' é o responsável por criar essa estrutura.**
- c) cria variáveis que podem ser acessadas por meio de referências. O comando 'def' é o responsável por definir essa estrutura.
- d) é dinâmico e oferece a possibilidade de executar um comando SQL a partir de uma string. O comando 'declare' é o responsável por definir essa estrutura.
- e) disponibiliza um comando que permite fazer iterações de execução do código tantas vezes quantas forem necessárias. O comando 'loop' é o responsável por criar essa estrutura.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** CESPE / CEBRASPE **Órgão:** SEFAZ-AL **Prova:** CESPE / CEBRASPE - 2021 - SEFAZ-AL - Auditor Fiscal de Finanças e Controle de Arrecadação da Fazenda Estadual

Acerca de Oracle e linguagem de programação em banco de dados (PL/SQL), julgue o item que se segue.

Um cursor é um objeto de esquema que agrupa tipos, variáveis, constantes, subprogramas, cursores e exceções relacionados logicamente.

Alternativas

Certo

Errado

DIRETO DO CONCURSO

Ano: 2021 **Banca:** CESPE / CEBRASPE **Órgão:** SEFAZ-AL **Prova:** CESPE / CEBRASPE - 2021 - SEFAZ-AL - Auditor Fiscal de Finanças e Controle de Arrecadação da Fazenda Estadual

Acerca de Oracle e linguagem de programação em banco de dados (PL/SQL), julgue o item que se segue.

Um cursor é um objeto de esquema que agrupa tipos, variáveis, constantes, subprogramas, cursores e exceções relacionados logicamente.

Alternativas

Certo

Errado

O '**Cursor**' é a construção PL/SQL que permite ao usuário nomear a área de trabalho e acessar as informações nela armazenadas. A principal função de um cursor **é recuperar dados, uma linha de cada vez, de um conjunto de resultados**, ao contrário dos comandos SQL que operam em todas as linhas do conjunto de resultados de uma só vez.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** Quadrix **Órgão:** CFT **Prova:**
Quadrix - 2021 - CFT - Analista de Tecnologia da
Informação Júnior

A partir do código PL/SQL (Procedural Language/Structured Query Language) acima, julgue o item.

Há um erro de sintaxe no código. A palavra reservada ELSIF foi escrita de forma errada. O correto seria ELSEIF.

Alternativas

Certo

Errado

```
CREATE OR REPLACE FUNCTION mostra_cargo_CFT (
    p_RENDIMENTO IN FUNCIONARIO.RENDIMENTO%type
)
RETURN FUNCIONARIO.CARGO%type
IS
    v_CARGO FUNCIONARIO.CARGO%type;
BEGIN
    IF p_RENDIMENTO < 2000 THEN
        v_CARGO := 'ESTAGIARIO';
    ELSIF p_RENDIMENTO < 5000 THEN
        v_CARGO := 'COORDENADOR';
    ELSIF p_RENDIMENTO < 7000 THEN
        v_CARGO := 'ASSESSOR';
    ELSE
        v_CARGO := 'GERENTE';
    END IF;
    RETURN v_CARGO;
END;
```

DIRETO DO CONCURSO

Ano: 2021 **Banca:** Quadrix **Órgão:** CFT **Prova:**
Quadrix - 2021 - CFT - Analista de Tecnologia da
Informação Júnior

A partir do código PL/SQL (Procedural Language/Structured Query Language) acima, julgue o item.

Há um erro de sintaxe no código. A palavra reservada ELSIF foi escrita de forma errada. O correto seria ELSEIF.

Alternativas

Certo

Errado

```
CREATE OR REPLACE FUNCTION mostra_cargo_CFT (
    p_RENDIMENTO IN FUNCIONARIO.RENDIMENTO%type
)
RETURN FUNCIONARIO.CARGO%type
IS
    v_CARGO FUNCIONARIO.CARGO%type;
BEGIN
    IF p_RENDIMENTO < 2000 THEN
        v_CARGO := 'ESTAGIARIO';
    ELSIF p_RENDIMENTO < 5000 THEN
        v_CARGO := 'COORDENADOR';
    ELSIF p_RENDIMENTO < 7000 THEN
        v_CARGO := 'ASSESSOR';
    ELSE
        v_CARGO := 'GERENTE';
    END IF;
    RETURN v_CARGO;
END;
```

DIRETO DO CONCURSO

Ano: 2021 **Banca:** VUNESP **Órgão:** Sema de Piracicaba - SP **Prova:** VUNESP - 2021
- Sema de Piracicaba - SP - Programador Junior

Em um comando do PL/SQL, é possível inserir comentários que ocupem várias linhas. Nesse caso, os delimitadores de início e fim desse tipo de comentário são, correta e respectivamente,

Alternativas

- a) // e \\.
- b) | e ||.
- c) <= e =>.
- d) /* e */.
- e) /# e #/.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** VUNESP **Órgão:** Sema de Piracicaba - SP **Prova:** VUNESP - 2021
- Sema de Piracicaba - SP - Programador Junior

Em um comando do PL/SQL, é possível inserir comentários que ocupem várias linhas. Nesse caso, os delimitadores de início e fim desse tipo de comentário são, correta e respectivamente,

Alternativas

- a) // e \\.
- b) | e ||.
- c) <= e =>.
- d) /* e */.
- e) /# e #/.

Referências

- <https://docs.oracle.com/en/database/oracle/oracle-database/index.html>
- <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/index.html>
- <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/cncpt/introduction-to-oracle-database.html#GUID-A42A6EF0-20F8-4F4B-AFF7-09C100AE581E>
- <http://wiki.icmc.usp.br/images/f/f1/Mat09o.pdf>
- <https://www.devmedia.com.br/guia-rapido-para-uso-do-rman/24131>