

# Índice

## **Exercícios**

01-Introdução ao PL/SQL .....	2
02-Declarando Variáveis PL/SQL .....	3
03-Instruções DML no PL/SQL .....	5
04-Estruturas de Controle.....	7
05-Registros PL/SQL .....	8

## **Soluções**

01-Introdução ao PL/SQL .....	9
02-Declarando Variáveis PL/SQL .....	10
03-Instruções DML no PL/SQL .....	11
04-Estruturas de Controle.....	12

## **Lista de Exercícios**

Esses exercícios estão disponíveis no final de cada apostila.

### **Cap 01-Introdução ao PL/SQL**

1. Qual dos seguintes blocos PL/SQL é executado corretamente?
  - a. BEGIN  
END;
  - b. DECLARE  
amount INTEGER(10);  
END;
  - c. DECLARE  
BEGIN  
END;
  - d. DECLARE  
amount INTEGER(10);  
BEGIN  
DBMS\_OUTPUT.PUT\_LINE(amount);  
END;
2. Crie e execute um bloco anônimo simples que exiba "Hello World". Execute e salve este script como lab\_01\_02\_soln.sql.

## **Cap 02-Declarando Variáveis PL/SQL**

1. Identifique se os nomes dos identificadores são válidos ou inválidos:
  - a. today
  - b. last\_name
  - c. today's\_date
  - d. Number\_of\_days\_in\_February\_this\_year
  - e. Isleap\$year
  - f. #number
  - g. NUMBER#
  - h. number1to7
2. Identifique se a declaração de variável e a inicialização é válida ou inválida:
  - a. printer\_name      constant VARCHAR2(10);
  - b. deliver\_to        VARCHAR2(10):=Johnson;
  - c. by\_when            DATE:= SYSDATE+1;
3. Examine o seguinte bloco anônimo e escolha a resposta correta apropriada.

```
SET SERVEROUTPUT ON
DECLARE
    fname VARCHAR2(20);
    lname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
    DBMS_OUTPUT.PUT_LINE( FNAME || ' ' ||lname);
END;
/
```

  - a. O bloco será executado com sucesso mas não será impresso nada.
  - b. O bloco retornará um erro porque a variável fname é declarada sem ser iniciada.
  - c. O bloco será executado com sucesso e irá imprimir 'fernandez'.
  - d. O bloco retornará um erro porque você não pode usar a palavra-chave DEFAULT para inicializar uma variável do tipo VARCHAR2.
  - e. O bloco retornará um erro porque a variável FNAME não foi declarada.
4. Crie um bloco anônimo. Carregue o script da aula 01 lab\_01\_02\_soln.sql, que você criou na pergunta 2 do exercício 1.
  - a. Adicione a seção DECLARE nesse bloco PL/SQL. Na seção DECLARE declare as seguintes variáveis:
    1. Variável today do tipo DATE. Inicializar today com SYSDATE
    2. Variável tomorrow do tipo today. Use o atributo%TYPE para declarar essa variável.
  - b. Na seção executável inicialize a variável tomorrow com uma expressão que calcula a data de tomorrow (adicione um ao valor de today). Exiba o valor de today e tomorrow após de imprimir 'Hello World'
  - c. Execute e salve este script como lab\_02\_04\_soln.sql. Um exemplo da saída é mostrada a seguir.

```
Hello World
TODAY IS : 12-JAN-04
TOMORROW IS : 13-JAN-04
PL/SQL procedure successfully completed.
```

5. Edite o script lab\_02\_04\_soln.sql.
  - a. Crie as bind variables basic\_percent e pf\_percent do tipo NUMBER.
  - b. Na seção executável do bloco PL/SQL atribua os valores 45 e 12 a basic\_percent e pf\_percent, respectivamente.
  - c. Encerre o bloco PL/SQL com "/" e exiba o valor das bind variables utilizando o comando PRINT.
  - d. Execute e salve seu arquivo de script como lab\_02\_05\_soln.sql.

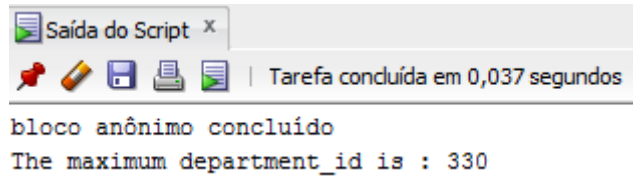
Na seção executável do bloco PL/SQL atribua os valores 45 e 12 a basic\_percent e pf\_percent, respectivamente.

Encerre o bloco PL/SQL com "/" e exibir o valor das variáveis de ligação usando o comando PRINT.

Execute e salve seu arquivo de script como lab\_02\_05\_soln.sql.

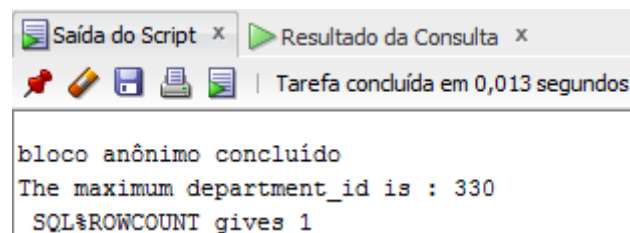
### Cap 03-Instruções DML no PL/SQL

1. Crie um bloco PL/SQL que seleciona o número máximo de departamento na tabela `DEPARTMENTS` e o armazena em uma variável. Imprima os resultados na tela. Salve o bloco PL/SQL em um arquivo nomeado `lab_04_01_soln.sql`.



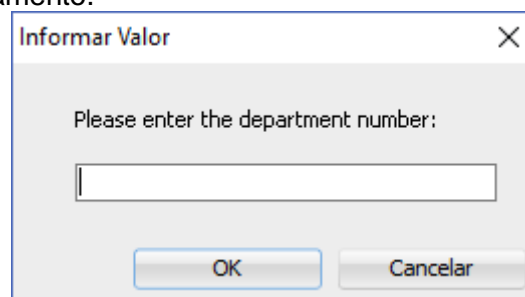
```
bloco anônimo concluído
The maximum department_id is : 330
```

2. Modifique o bloco PL/SQL que você criou no exercício 1 (`lab_04_01_soln.sql`) para inserir um novo departamento na tabela `DEPT`. Salve o bloco PL/SQL em um arquivo nomeado (`lab_04_02_soln.sql`).
  - a. Em vez de imprimir o número do departamento recuperado do exercício 1, adicione 10 a ele (Se o número retornado foi 330 então adicione 10) e use-o como o número do departamento do novo departamento. O novo departamento deve ser definido como: `EDUCATION`
  - b. Use uma variável de substituição do tipo `BIND VARIABLE` para definir o número do departamento novo departamento.
  - c. Deixe um valor nulo nas colunas como `MANAGER_ID` e como o `LOCATION_ID` do novo departamento.
  - d. Execute o bloco PL/SQL.
  - e. Exiba as seguintes mensagens como a seguir demonstrado. (Utilize o atributo de cursor `SQL%ROWCOUNT` para exibir a quantidade de linhas manipuladas.)



```
bloco anônimo concluído
The maximum department_id is : 330
SQL%ROWCOUNT gives 1
```

3. Crie um bloco PL/SQL que atualize a localização (`location_id`) para um departamento existente (`department_id`) (Tabela: `DEPARTMENTS`). Salve o bloco PL/SQL em um arquivo denominado `lab_04_03_soln.sql`.
  - a. Use uma variável de substituição (`VARIABLE` ou `ACCEPT`) para informar o número de departamento.

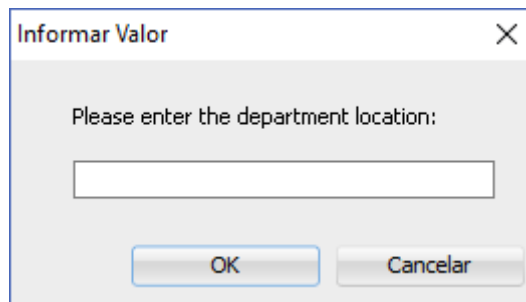


Informar Valor

Please enter the department number:

OK Cancelar

- b. Use uma variável de substituição (`VARIABLE` ou `ACCEPT`) para informar a localização de departamento. (Consulte a tabela `LOCATIONS` para ter um número de localização válido)



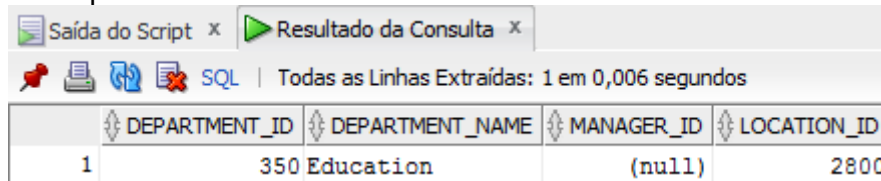
Informar Valor

Please enter the department location:

OK Cancelar

c. Teste o bloco PL/SQL.

d. Exiba o nome e o número do departamento, além da localização do departamento atualizado.

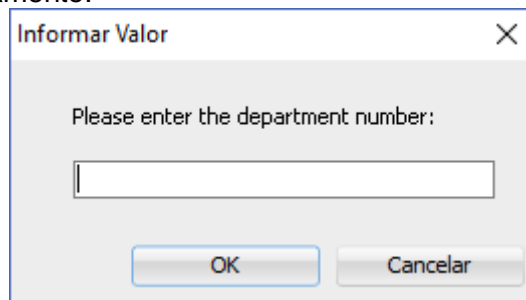


SQL | Todas as Linhas Extraídas: 1 em 0,006 segundos

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	350	Education	(null)	2800

4. Crie um bloco PL/SQL que delete o departamento criado no exercício 2. Salve o bloco PL/SQL em um arquivo denominado lab\_04\_04\_soln.sql.

a. Use uma variável de substituição (VARIABLE ou ACCEPT) para informar para o número do departamento.



Informar Valor

Please enter the department number:

OK Cancelar

b. Imprima o número de linhas afetadas na tela. (Para isso utilize uma `BIND VARIABLE` e exiba essa mensagem após o teste do bloco. Não use `dbms_output.put_line`)

c. Teste o bloco PL/SQL.

d. O que acontece se você informar um número de departamento inexistente?

e. Confirme que o departamento foi deletado.

## Cap 04-Estruturas de Controle

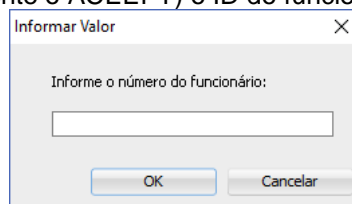
1. Execute o comando a seguir para criar a tabela messages.

```
CREATE TABLE MESSAGES  
(TEXT VARCHAR(50));
```

- a. Crie um bloco PL/SQL que insira na tabela MESSAGES os números de 1 até 10, excluindo os números 6 e 8.

Após a execução do bloco faça SELECT na tabela MESSAGES para verificar se os números foram inseridos corretamente.

2. Crie um bloco PL/SQL que premie um funcionário, anexando um asterisco à coluna STARS da tabela EMP para cada US\$1000 do salário do funcionário. Para isso deverá ser aceito via teclado (utilize obrigatoriamente o ACCEPT) o ID do funcionário.



Se o funcionário recebe um salário de US\$8000, a string de asteriscos deve conter oito asteriscos. Se o funcionário recebe um salário de US\$12500, a string de asteriscos deve conter 12 asteriscos.

Atualize a coluna STARS do funcionário com a string de asteriscos, de acordo com a regra anterior.

Exemplos de seleção de funcionários:

```
SELECT EMPLOYEE_ID, SALARY, STARS, LENGTH(STARS) STARS_COUNT  
FROM EMP  
WHERE EMPLOYEE_ID in (100,170,107);
```

EMPLOYEE_ID	SALARY	STARS	STARS_COUNT
100	24000	*****	24
107	4200	***	4
170	9600	*****	10

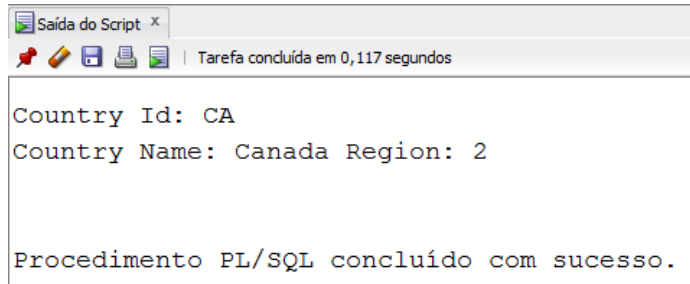
Utilize o script a seguir para criar a tabela EMP que será uma cópia da tabela EMPLOYEES

```
drop table emp;  
create table emp  
as select * from employees;
```

```
alter table emp  
add stars varchar(50);
```

## Cap 05-Registros PL/SQL

1. Escreva um bloco PL/SQL para imprimir informações sobre um determinado país. Declare um registro PL/SQL baseado na estrutura da tabela de countries.
  - a. Use o comando DEFINE para definir uma variável c **countryid**. Atribua CA a countryid. Passe o valor para o bloco PL/SQL através de uma variável de substituição.
  - b. Na seção declare, use o atributo %ROWTYPE e declare a variável country\_record de tipo countries.
  - c. Na seção executável, obter todas as informações da tabela **countries** usando countryid. Mostra as informações selecionadas sobre o país.



```
Country Id: CA
Country Name: Canada Region: 2

Procedimento PL/SQL concluído com sucesso.
```

The screenshot shows a window titled 'Saída do Script' with a toolbar containing icons for running, saving, and printing. Below the toolbar, the output of the script is displayed in a monospaced font, showing the country details for Canada and a success message.



# **Lista de Respostas**

## **Cap 01-Introdução ao PL/SQL**

1.

Alternativa D

2.

```
SET SERVEROUTPUT ON
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello World');
END;
/
```

## **Cap 02-Declarando Variáveis PL/SQL**

1.

- a. válida
- b. válida
- c. inválida
- d. inválida
- e. válida
- f. inválida
- g. válida
- h. válida

2.

- a. inválida
- b. inválida
- c. válida

3.

Alternativa c

4.

```
SET SERVEROUTPUT ON
DECLARE
    today DATE:=SYSDATE;
    tomorrow today%TYPE;
BEGIN
    tomorrow:=today +1;
    DBMS_OUTPUT.PUT_LINE(' Hello World ');
    DBMS_OUTPUT.PUT_LINE('TODAY IS : '|| today);
    DBMS_OUTPUT.PUT_LINE('TOMORROW IS : ' || tomorrow);
END;
```

5.

```
VARIABLE basic_percent NUMBER
VARIABLE pf_percent NUMBER

SET SERVEROUTPUT ON
DECLARE
    today DATE:=SYSDATE;
    tomorrow today%TYPE;
BEGIN
    :basic_percent:=45;
    :pf_percent:=12;

    tomorrow:=today +1;
    DBMS_OUTPUT.PUT_LINE(' Hello World ');
    DBMS_OUTPUT.PUT_LINE('TODAY IS : '|| today);
    DBMS_OUTPUT.PUT_LINE('TOMORROW IS : ' || tomorrow);
END;
/
PRINT basic_percent
PRINT pf_percent
```

## **Cap 03-Instruções DML no PL/SQL**

1.

```
SET SERVEROUTPUT ON
DECLARE
    max_deptno NUMBER;
BEGIN
    SELECT MAX(department_id) INTO max_deptno FROM
        departments;
    DBMS_OUTPUT.PUT_LINE('The maximum department_id is : ' ||
max_deptno);
END;
```

2.

```
VARIABLE dept_id NUMBER
SET SERVEROUTPUT ON
DECLARE
    dept_name departments.department_name%TYPE:= 'Education';
    max_deptno NUMBER;
BEGIN
    SELECT MAX(department_id) INTO max_deptno FROM departments;
    DBMS_OUTPUT.PUT_LINE('The maximum department_id is : ' ||
max_deptno);
    :dept_id := 10 + max_deptno;
    INSERT INTO departments (department_id, department_name,
location_id)
    VALUES (:dept_id,dept_name, NULL);
    DBMS_OUTPUT.PUT_LINE (' SQL%ROWCOUNT gives ' || SQL%ROWCOUNT);
END;
/
SELECT * FROM departments WHERE department_id=:dept_id;
```

3.

```
ACCEPT p_department_id PROMPT 'Please enter the department
number'
ACCEPT p_location_id PROMPT 'Please entre the department
location'
DECLARE
    v_department_id
departments.department_id%TYPE:=&p_department_id;
    v_location_id departments.location_id%TYPE:=&p_location_id;
BEGIN
    UPDATE departments
    SET location_id=v_location_id
    WHERE department_id=v_department_id;
END;
/
```

4.

```
ACCEPT p_department_id PROMPT 'Please enter the department
number'
DECLARE
    v_department_id
departments.department_id%TYPE:=&p_department_id;
BEGIN
    delete from departments
    WHERE department_id=v_department_id;
END;
/
```

## **Cap 04-Estruturas de Controle**

1.

```
BEGIN
FOR i in 1..10 LOOP
  IF i = 6 or i = 8 THEN
    null;
  ELSE
    INSERT INTO messages(results)
      VALUES (i);
  END IF;
END LOOP;
COMMIT;
END;
/
SELECT * FROM messages;
```

2.

```
ACCEPT p_empno PROMPT 'Informe o número do funcionário: '
DECLARE
  v_empno      emp.employee_id%TYPE := TO_NUMBER(&p_empno);
  v_asterisk   emp.stars%TYPE := NULL;
  v_sal        emp.salary%TYPE;
BEGIN
  SELECT NVL(ROUND(salary/1000), 0)
    INTO v_sal
  FROM emp
  WHERE employee_id = v_empno;

  FOR i IN 1..v_sal LOOP
    v_asterisk := v_asterisk || '*';
  END LOOP;

  UPDATE emp
  SET stars = v_asterisk
  WHERE employee_id = v_empno;

  COMMIT;
END;
/
```