**SCC0241 - Laboratório de Bases de Dados**

**Relatório de execução da aula prática 5**

**Aluno NUSP**

Giancarlo Malfate Caprino 12725025

Adriano da Silva de Carvalho 13692400

**Exercício 1**

**➔ Comentário**

Este espaço destina-se às questões discursivas; apenas algumas questões têm comentários.

**➔ Código**

SELECT \*

FROM DUAL;

**➔ Saída**

Resultados apresentados no terminal da interface SQL; exemplo: status de execução, mensagens de erro, dados (em forma tabular), e outras informações pertinentes. Pode-se enfatizar o texto com negrito ou itálico, mas não underline.

# **Observações:**

* O código será copiado deste documento para execução, portanto, finalize cada linha com um ponto e vírgula;
* Comentários também devem ser colocados junto com o código e iniciados pelos caracteres de comentário “--”;
* O código deve ser enfatizado com a fonte Courier New, de preferência com uma cor diferente de preto;
* Faça a cópia do exemplo acima para cada exercício, mantendo a aparência do relatório ao longo dos exercícios;
* Inicie cada exercício (mas não cada item) em uma página nova usando CTRL+ENTER;
* Itens que não valem nota, não precisam ser reportados;
* Itens curtos podem ser agrupados quando conveniente;
* Objetive a fácil leitura do documento.

**Exercício 1**

**Item a**

**➔ Código**

ALTER TABLE L01\_PAIS

ADD VITORIAS NUMBER DEFAULT 0;

CREATE OR REPLACE PROCEDURE atualizar\_vitorias\_pais AS

BEGIN

-- Atualiza o número de vitórias para cada país

UPDATE L01\_PAIS P

SET VITORIAS = (

SELECT COUNT(\*)

FROM L05\_DISPUTA D

WHERE D.VENCEDOR = P.NOME

);

-- Commit para garantir que as alterações sejam salvas

COMMIT;

END;

/

BEGIN

atualizar\_vitorias\_pais;

END;

/

SELECT \* FROM L01

**Item b**

**➔ Código**

CREATE OR REPLACE TRIGGER TRG\_UPDATE\_VITORIAS

AFTER INSERT OR UPDATE OR DELETE ON L05\_DISPUTA

FOR EACH ROW

BEGIN

-- Adicionar uma vitória

IF INSERTING THEN

UPDATE L01\_PAIS

SET VITORIAS = VITORIAS + 1

WHERE NOME = :NEW.VENCEDOR;

-- Remover uma vitória (atualização ou exclusão)

ELSIF UPDATING OR DELETING THEN

-- Remover vitória se a disputa anterior foi vencida por outro país

IF :OLD.VENCEDOR IS NOT NULL THEN

UPDATE L01\_PAIS

SET VITORIAS = VITORIAS - 1

WHERE NOME = :OLD.VENCEDOR;

END IF;

-- Adicionar vitória se a disputa atual for vencida por outro país

IF :NEW.VENCEDOR IS NOT NULL AND :NEW.VENCEDOR <> :OLD.VENCEDOR THEN

UPDATE L01\_PAIS

SET VITORIAS = VITORIAS + 1

WHERE NOME = :NEW.VENCEDOR;

END IF;

END IF;

END;

/

--UPDATE

SELECT \* FROM L05\_DISPUTA

WHERE ID = 2

-- Atualizando o vencedor da disputa com ID = 2 - China antiga vencedora

UPDATE L05\_DISPUTA

SET VENCEDOR = 'Japão'

WHERE ID = 2;

-- Conferindo

SELECT \* FROM L01

--DELETE

SELECT \* FROM L05\_DISPUTA

WHERE ID = 7

-- Excluindo a disputa com ID = 11 - Brasil vencedor

DELETE FROM L05\_DISPUTA

WHERE ID = 7;

-- Conferindo

--INSERT

-- Inserindo uma nova disputa - mais uma vitoria para o Brasil

INSERT INTO L05\_DISPUTA (ID, PAIS1, PAIS2, VENCEDOR, MODALIDADE, OLIMPIADA, DATA\_HORA, LOCAL)

VALUES (11, 'Itália', 'Brasil', 'Brasil', 1, 2008, TO\_DATE('2008-08-15 17:00:00', 'YYYY-MM-DD HH24:MI:SS'), 4);

-- Conferindo

SELECT \* FROM L01

**Exercício 2**

**Item a**

**➔ Código**

CREATE OR REPLACE TRIGGER TRG\_VALIDAR\_LOCAL\_DISPUTA

BEFORE INSERT OR UPDATE ON L05\_DISPUTA

FOR EACH ROW

DECLARE

v\_pais\_olimpiada L02\_OLIMPIADA.PAIS%TYPE;

v\_pais\_local L04\_LOCAL.PAIS%TYPE;

BEGIN

-- Obter o país onde a Olimpíada está ocorrendo

SELECT O.PAIS

INTO v\_pais\_olimpiada

FROM L02\_OLIMPIADA O

WHERE O.ANO = :NEW.OLIMPIADA;

-- Obter o país onde o local da disputa está situado

SELECT L.PAIS

INTO v\_pais\_local

FROM L04\_LOCAL L

WHERE L.ID = :NEW.LOCAL;

-- Verificar se o país do local da disputa corresponde ao país da Olimpíada

IF v\_pais\_local <> v\_pais\_olimpiada THEN

RAISE\_APPLICATION\_ERROR(-20001, 'O local da disputa deve estar no país onde a Olimpíada está ocorrendo.');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Local ou Olimpíada não encontrados.');

END;

/

INSERT INTO L05\_DISPUTA (ID, PAIS1, PAIS2, VENCEDOR, MODALIDADE, OLIMPIADA, DATA\_HORA, LOCAL)

VALUES (17, 'Japão', 'Itália', 'Itália', 1, 2008, TO\_DATE('2008-08-10 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 2);

**Item b**

**➔ Código**

CREATE OR REPLACE TRIGGER TRG\_VALIDAR\_DATA\_DISPUTA

BEFORE INSERT OR UPDATE ON L05\_DISPUTA

FOR EACH ROW

DECLARE

v\_data\_inicio L02\_OLIMPIADA.DATA\_INICIO%TYPE;

v\_data\_encerramento L02\_OLIMPIADA.DATA\_ENCERRAMENTO%TYPE;

BEGIN

-- Buscar o período da Olimpíada

SELECT DATA\_INICIO, DATA\_ENCERRAMENTO

INTO v\_data\_inicio, v\_data\_encerramento

FROM L02\_OLIMPIADA

WHERE ANO = :NEW.OLIMPIADA;

-- Verifica se a 'DATA\_HORA' inserida está entre o período da Olimpiada.

IF :NEW.DATA\_HORA < v\_data\_inicio OR :NEW.DATA\_HORA > v\_data\_encerramento THEN

RAISE\_APPLICATION\_ERROR(-20001, 'A data da disputa deve estar dentro do período da Olimpíada.');

END IF;

END;

/

INSERT INTO L05\_DISPUTA

(ID, PAIS1, PAIS2, VENCEDOR, MODALIDADE, OLIMPIADA, DATA\_HORA, LOCAL)

VALUES

(333777, 'Brasil', 'Alemanha', 'Brasil', 2, 2016, TO\_DATE('2016-08-31 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), 1);

**Item c**

**➔ Código**

CREATE OR REPLACE TRIGGER TRG\_VALIDAR\_MODALIDADE\_ATLETA

BEFORE INSERT ON L07\_JOGA

FOR EACH ROW

DECLARE

v\_modalidade\_atleta L06\_ATLETA.MODALIDADE%TYPE;

v\_modalidade\_disputa L05\_DISPUTA.MODALIDADE%TYPE;

BEGIN

-- Buscar a modalidade do atleta na tabela L06\_ATLETA

SELECT MODALIDADE

INTO v\_modalidade\_atleta

FROM L06\_ATLETA

WHERE PASSAPORTE = :NEW.ATLETA;

-- Buscar a modalidade da disputa na tabela L05\_DISPUTA

SELECT MODALIDADE

INTO v\_modalidade\_disputa

FROM L05\_DISPUTA

WHERE ID = :NEW.DISPUTA;

-- Verificar se a modalidade da disputa é a mesma da modalidade do atleta

IF v\_modalidade\_atleta <> v\_modalidade\_disputa THEN

RAISE\_APPLICATION\_ERROR(-20002, 'O atleta só pode participar de disputas na modalidade registrada.');

END IF;

END;

/

INSERT INTO L07\_JOGA (ATLETA, DISPUTA) VALUES ('FR13578', 41);

**Exercício 3**

**➔ Código**

CREATE TABLE output (

id NUMBER,

operacao VARCHAR2(30),

descricao VARCHAR2(100)

);

CREATE SEQUENCE op\_seq

START WITH 1

INCREMENT BY 1;

CREATE OR REPLACE TRIGGER TRG\_AUDITAR\_DISPUTA

AFTER INSERT OR UPDATE OR DELETE ON L05\_DISPUTA

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

operacao VARCHAR2(30);

BEGIN

-- Identificar o tipo de operação

IF INSERTING THEN

operacao := 'INSERT';

ELSIF UPDATING THEN

operacao := 'UPDATE';

ELSIF DELETING THEN

operacao := 'DELETE';

END IF;

-- Registrar os valores antigos (para UPDATE e DELETE) e novos (para INSERT e UPDATE)

IF DELETING OR UPDATING THEN

INSERT INTO output VALUES(op\_seq.NEXTVAL, operacao, :OLD.PAIS1 || ' vs ' || :OLD.PAIS2 || ' - ' || :OLD.VENCEDOR);

END IF;

IF INSERTING OR UPDATING THEN

INSERT INTO output VALUES(op\_seq.NEXTVAL, operacao, :NEW.PAIS1 || ' vs ' || :NEW.PAIS2 || ' - ' || :NEW.VENCEDOR);

END IF;

-- Inserir um separador para as operações no log

INSERT INTO output VALUES(0, '------------', '-----------');

-- Commit para confirmar a transação autônoma

COMMIT;

END;

/

CREATE OR REPLACE TRIGGER TRG\_PROTECT\_OUTPUT

BEFORE UPDATE OR DELETE ON output

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(-20001, 'Operação não permitida: a tabela de auditoria não pode ser modificada.');

END;

/

INSERT INTO L05\_DISPUTA (ID, PAIS1, PAIS2, VENCEDOR, MODALIDADE, OLIMPIADA, DATA\_HORA, LOCAL)

VALUES (2000, 'Japão', 'Brasil', 'Brasil', 1, 2008, TO\_DATE('2008-08-11 10:30:00', 'YYYY-MM-DD HH24:MI:SS'), 3);

-- Checando existencia

SELECT \* FROM L05\_DISPUTA

WHERE ID = 529

UPDATE L05\_DISPUTA SET VENCEDOR = 'Reino Unido', DATA\_HORA = TO\_DATE('2021-07-29 10:00', 'YYYY-MM-DD HH24:MI') WHERE ID = 529;

UPDATE L05\_DISPUTA SET PAIS1 = 'Brasil', DATA\_HORA = TO\_DATE('2021-07-29 10:00', 'YYYY-MM-DD HH24:MI') WHERE ID = 529;

--necessario dropar constraint para rodar o delete e checar o output

ALTER TABLE L11\_TRANSMITE DISABLE CONSTRAINT FK\_DISPUTA;

DELETE FROM L05\_DISPUTA WHERE ID = 529;

-- Conferindo

SELECT \* FROM L05\_DISPUTA

WHERE ID = 529

-- Conferindo

SELECT \* FROM output;

**Exercício 4**

**➔ Código**

-- Realizando uma consulta para ver o estado anterior do nome do país

SELECT \* FROM L01\_PAIS WHERE NOME = 'Brasil';

SELECT \* FROM L02\_OLIMPIADA WHERE PAIS = 'Brasil';

SELECT \* FROM L04\_LOCAL WHERE PAIS = 'Brasil';

SELECT \* FROM L05\_DISPUTA WHERE PAIS1 = 'Brasil' OR PAIS2 = 'Brasil' OR VENCEDOR = 'Brasil';

SELECT \* FROM L06\_ATLETA WHERE PAIS = 'Brasil';

CREATE OR REPLACE TRIGGER TRG\_UPDATE\_PAIS\_CASCADE

AFTER UPDATE OF NOME ON L01\_PAIS

FOR EACH ROW

BEGIN

-- Atualiza o país nas tabelas relacionadas

-- Atualiza L02\_OLIMPIADA (referência ao país)

UPDATE L02\_OLIMPIADA

SET PAIS = :NEW.NOME

WHERE PAIS = :OLD.NOME;

-- Atualiza L04\_LOCAL (referência ao país)

UPDATE L04\_LOCAL

SET PAIS = :NEW.NOME

WHERE PAIS = :OLD.NOME;

-- Atualiza L05\_DISPUTA (referências PAIS1, PAIS2 e VENCEDOR)

UPDATE L05\_DISPUTA

SET PAIS1 = :NEW.NOME

WHERE PAIS1 = :OLD.NOME;

UPDATE L05\_DISPUTA

SET PAIS2 = :NEW.NOME

WHERE PAIS2 = :OLD.NOME;

UPDATE L05\_DISPUTA

SET VENCEDOR = :NEW.NOME

WHERE VENCEDOR = :OLD.NOME;

-- Atualiza L06\_ATLETA (referência ao país)

UPDATE L06\_ATLETA

SET PAIS = :NEW.NOME

WHERE PAIS = :OLD.NOME;

END;

/

-- Necessario desabilitar a constraint e os trigger para rodar corretamente esse trigger

ALTER TRIGGER TRG\_UPDATE\_VITORIAS DISABLE;

ALTER TRIGGER TRG\_VALIDAR\_LOCAL\_DISPUTA DISABLE;

ALTER TRIGGER TRG\_VALIDAR\_DATA\_DISPUTA DISABLE;

ALTER TABLE L05\_DISPUTA DISABLE CONSTRAINT CK\_VENC;

-- Atualizar o nome do país de 'Brasil' para 'Brasil Atualizado'

UPDATE L01\_PAIS

SET NOME = 'Brasil Atualizado'

WHERE NOME = 'Brasil';

-- Habilitando novamente

ALTER TRIGGER TRG\_UPDATE\_VITORIAS ENABLE;

ALTER TRIGGER TRG\_VALIDAR\_LOCAL\_DISPUTA ENABLE;

ALTER TRIGGER TRG\_VALIDAR\_DATA\_DISPUTA ENABLE;

ALTER TABLE L05\_DISPUTA ENABLE CONSTRAINT CK\_VENC;

-- Verificar a tabela L01\_PAIS

SELECT \* FROM L01\_PAIS WHERE NOME = 'Brasil Atualizado';

-- Verificar a tabela L02\_OLIMPIADA

SELECT \* FROM L02\_OLIMPIADA WHERE PAIS = 'Brasil Atualizado';

-- Verificar a tabela L04\_LOCAL

SELECT \* FROM L04\_LOCAL WHERE PAIS = 'Brasil Atualizado';

-- Verificar a tabela L05\_DISPUTA

SELECT \* FROM L05\_DISPUTA WHERE PAIS1 = 'Brasil Atualizado' OR PAIS2 = 'Brasil Atualizado' OR VENCEDOR = 'Brasil Atualizado';

-- Verificar a tabela L06\_ATLETA

SELECT \* FROM L06\_ATLETA WHERE PAIS = 'Brasil Atualizado';

**Exercício 5**

**Item a**

**➔ Código**

CREATE TABLE event\_table (

event\_type VARCHAR2(30),

object\_name VARCHAR2(30),

object\_type VARCHAR2(20),

object\_owner VARCHAR2(30),

event\_timestamp DATE

);

**Item b**

**➔ Código**

CREATE OR REPLACE TRIGGER log\_ddl\_events

AFTER DDL ON SCHEMA

DECLARE

v\_event VARCHAR2(30);

v\_obj\_name VARCHAR2(30);

v\_obj\_type VARCHAR2(20);

v\_obj\_owner VARCHAR2(30);

BEGIN

-- Captura o evento DDL

v\_event := ora\_sysevent;

-- Captura o nome do objeto DDL

v\_obj\_name := ora\_dict\_obj\_name;

-- Captura o tipo do objeto DDL

v\_obj\_type := ora\_dict\_obj\_type;

-- Captura o proprietário do objeto DDL

v\_obj\_owner := ora\_dict\_obj\_owner;

-- Inserção na tabela de eventos

INSERT INTO event\_table (event\_type, object\_name, object\_type, object\_owner, event\_timestamp)

VALUES (v\_event, v\_obj\_name, v\_obj\_type, v\_obj\_owner, SYSDATE);

END;

/

-- Criar uma tabela para ver se o log funciona

CREATE TABLE teste\_ddl (

id NUMBER

);

-- Alterar a tabela

ALTER TABLE teste\_ddl ADD nome VARCHAR2(50);

-- Dropar a tabela

DROP TABLE teste\_ddl;

-- Consultar o log para ver as operações registradas

SELECT \* FROM event\_table;