**Checklist 1 - SQL e PL - GDI EC 2016.2**

**CENTRO DE INFORMÁTICA – UFPE**

**IF685/EC - GERENCIAMENTO DE DADOS E INFORMAÇÕES**

**Alunos:** Andreza Dantas Layme Pifano de Moura (adlpm), Filipe Cordeiro de Medeiros Azevedo (fcma), Flavio Lins da Mota Nascimento (flmn), Jen Horng Liu (jhl), Lucas Rodrigues de Araujo (lra2) e Victor Crisostomo Mellia (vcm)

**SQL - básicas**

-- 1 e 2

SELECT pessoa\_cpf

FROM evento

WHERE local\_id BETWEEN 3 AND 14

AND data BETWEEN to\_date ('13/07/2016','dd/mm/yyyy') AND to\_date ('17/07/2016','dd/mm/yyyy');

-- 3, 4, 5 e 6

SELECT atleta\_cpf

FROM equipe

WHERE tecnico\_cpf IN(SELECT cpf FROM tecnico WHERE diploma IS NOT NULL)

AND nome\_equipe LIKE '%BRA'

ORDER BY atleta\_cpf;

-- 7

CREATE VIEW pessoa\_func AS

SELECT nome, pais\_id, cpf

FROM pessoa;

-- 8

SELECT nome, pais\_id

FROM pessoa\_func

WHERE pais\_id < 691;

-- 9

DROP VIEW pessoa\_func;

-- 10, 11 e 12 na criação de tabela

--13

DROP TABLE equipamento\_perdido;

CREATE TABLE equipamento\_perdido (

local\_id NUMBER,

descricao VARCHAR2(30) DEFAULT 'SEMDESCRICAO',

CONSTRAINT equipamento\_perdido\_fk FOREIGN KEY (local\_id) REFERENCES local (id)

);

DROP TABLE equipamento\_perdido;--desfazer

-- 14

ALTER TABLE atleta

ADD nome\_atleta VARCHAR2(30);

-- 15

ALTER TABLE atleta

MODIFY nome\_atleta CHAR(10);

-- 16

ALTER TABLE atleta

DROP COLUMN nome\_atleta;

-- 17 e 18

SELECT AVG(id) - 1

FROM local;

-- 20

SELECT DISTINCT pais\_id

FROM pessoa;

-- 19 e 21

SELECT pais\_id COUNT(\*)

FROM pessoa

GROUP BY pais\_id

HAVING COUNT(\*) > 2;

-- 22

SELECT tipo, COUNT(\*)

FROM medalha

GROUP BY(tipo)

HAVING COUNT(\*) < (SELECT COUNT(DISTINCT tipo) FROM medalha);

-- 23

SELECT tipo, COUNT(\*)

FROM medalha

WHERE premiacao > 1000

GROUP BY tipo

HAVING COUNT(\*) >= 1;

-- 24

SELECT P.cpf, P.nome, P.pais\_id

FROM pessoa P, tecnico T

WHERE P.cpf = T.cpf;

-- 25

SELECT P.nome, T.telefone, A.numero

FROM pessoa P, telefone\_pessoa T, atleta A

WHERE P.cpf = T.cpf AND T.cpf = A.cpf;

-- 26

SELECT \*

FROM pessoa INNER JOIN atleta on pessoa.cpf = atleta.cpf;

-- 27

SELECT \*

FROM pessoa LEFT JOIN atleta on pessoa.cpf = atleta.cpf

WHERE pais\_id = 729;

-- 28

SELECT \*

FROM equipe RIGHT JOIN atleta on equipe.atleta\_cpf = atleta.cpf

WHERE equipe.nome\_equipe IS NOT NULL;

-- 29

SELECT \*

FROM atleta FULL JOIN pessoa on atleta.cpf= pessoa.cpf;

-- 30 e 31

SELECT telefone

FROM telefone\_pessoa

WHERE cpf = ANY (SELECT cpf FROM pessoa WHERE cpf <> ALL (SELECT cpf FROM tecnico));

-- 32 e 34

SELECT cpf

FROM pessoa

WHERE EXISTS (SELECT A.cpf FROM atleta A WHERE A.cpf = cpf) UNION (SELECT T.cpf FROM tecnico T WHERE T.cpf = cpf);

-- 33

SELECT A.cpf AS atletacpf

FROM atleta A

WHERE A.cpf IN(SELECT P.cpf FROM pessoa P WHERE P.cpf = A.cpf) AND

A.numero > 15;

-- 35 e 36

((SELECT cpf

FROM pessoa)

INTERSECT

(SELECT cpf

FROM tecnico))

MINUS

(SELECT P.cpf

FROM pessoa P, tecnico T

WHERE P.cpf = T.cpf AND P.cpf > 60000000000);

-- 37

INSERT INTO evento(pessoa\_cpf, local\_id, competicao\_id, data) values((SELECT DISTINCT tecnico\_cpf FROM equipe WHERE nome\_equipe LIKE ‘%FRA’ ), 14, 100, '19-07-2016');

-- 38

UPDATE evento

SET data = (SELECT data FROM evento WHERE data = '18-07-2016')

WHERE local\_id = 14 AND

data <> to\_date ('18/07/2016','dd/mm/yyyy');

-- 39

DELETE

FROM evento

WHERE pessoa\_cpf IN(SELECT DISTINCT tecnico\_cpf FROM equipe WHERE nome\_equipe LIKE '%FRA');

-- 40 e 41

GRANT UPDATE ON pessoa TO usuario;

REVOKE UPDATE ON pessoa TO usuario;

**SQL - Avançadas**

-- 42

SELECT nome

FROM (SELECT nome, pais\_id FROM pessoa WHERE pais\_id = 291)

WHERE pais\_id = 291;

-- 43

SELECT M.tipo, M.premiacao

FROM medalha

WHERE premiacao BETWEEN (SELECT min(premiacao) + 100 FROM medalha) AND (SELECT avg(premiacao) FROM medalha);

-- 44

SELECT M.tipo, M.premiacao

FROM medalha M

WHERE M.premiacao BETWEEN (SELECT min(premiacao) FROM medalha) AND (SELECT avg(premiacao) FROM medalha);

-- 45

SELECT P.nome, T.telefone

FROM pessoa P INNER JOIN atleta A on P.cpf = A.cpf INNER JOIN telefone\_pessoa T on P.cpf = T.cpf;

-- 46

SELECT \*

FROM pessoa

ORDER BY pais\_id, nome, cpf;

-- 47

SELECT P.nome

FROM pessoa P

WHERE EXISTS (SELECT A.pais\_id FROM pessoa A WHERE P.pais\_id = A.pais\_id);

**PL - BÁSICAS**

-- 48 e 49

DECLARE

nome2 pessoa.nome%TYPE;

BEGIN

SELECT p.nome into nome2 FROM pessoa p WHERE p.cpf = '09660138405';

DBMS\_OUTPUT.PUT\_LINE(nome2);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN RAISE\_APPLICATION\_ERROR (-20123, 'mais de uma pessoa retornada');

END;

/

-- 50 e 51

DECLARE

total1 number;

total2 number;

BEGIN

select count(\*) into total1 from pessoa where pais\_id = 291;

select count(\*) into total2 from pessoa where pais\_id = 311;

if(total1 > total2) then

DBMS\_OUTPUT.PUT\_LINE('Tem mais brasileiros que americanos');

elsif(total1 < total2) then

DBMS\_OUTPUT.PUT\_LINE('Tem mais americanos que brasileiros');

else

DBMS\_OUTPUT.PUT\_LINE('Numero de brasileiros e americanos sao iguais');

END IF;

END;

/

-- 52

DECLARE

p CHAR(1) := UPPER('&p');

premio VARCHAR2(20);

BEGIN

premio :=

CASE

WHEN p = 'O' THEN 'Ouro'

WHEN p = 'P' THEN 'Prata'

WHEN p = 'B' THEN 'Bronze'

ELSE 'Desconhecido'

END;

DBMS\_OUTPUT.PUT\_LINE('Premio: '||premio);

END;

/

-- 53, 56, 58 e 59

DECLARE

CURSOR pessoa\_nome IS

SELECT nome

FROM pessoa;

var\_pessoa pessoa\_nome%ROWTYPE;

BEGIN

OPEN pessoa\_nome;

LOOP

FETCH pessoa\_nome INTO var\_pessoa;

EXIT WHEN pessoa\_nome%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Nome: '||var\_pessoa.nome);

END LOOP;

CLOSE pessoa\_nome;

END;

/

-- 54, 57 e 60

DECLARE

CURSOR atleta\_cursor IS

SELECT \* FROM atleta;

reg\_atleta atleta\_cursor%ROWTYPE;

BEGIN

OPEN atleta\_cursor;

FETCH atleta\_cursor INTO reg\_atleta;

WHILE atleta\_cursor%FOUND LOOP

DBMS\_OUTPUT.PUT\_LINE(reg\_atleta.cpf);

FETCH atleta\_cursor INTO reg\_atleta;

END LOOP;

CLOSE atleta\_cursor;

END;

/

-- 55 e 62

DECLARE

BEGIN

FOR c IN (SELECT cpf FROM tecnico) LOOP

DBMS\_OUTPUT.PUT\_LINE(c.cpf);

END LOOP;

END;

/

-- 61

DECLARE

CURSOR cpfatleta (nomequipe equipe.nome\_equipe%TYPE) IS

SELECT atleta\_cpf

FROM equipe

WHERE nome\_equipe = nomequipe;

v\_cpf cpfatleta%ROWTYPE;

BEGIN

FOR v\_cpf IN cpfatleta('Equipe BRA') LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_cpf.atleta\_cpf);

END LOOP;

END;

/

-- 63

CREATE OR REPLACE PROCEDURE Cpf7 IS

n tecnico.cpf%TYPE;

BEGIN

SELECT cpf INTO n

FROM tecnico

WHERE diploma = 7;

DBMS\_OUTPUT.PUT\_LINE('cpf: '||n);

END;

/

-- 64, 65 e 66

CREATE OR REPLACE PROCEDURE trocaNumero(cpfa IN atleta.cpf%TYPE, numeroNovo IN OUT atleta.numero%TYPE)IS

temp atleta.numero%TYPE;

BEGIN

SELECT numero INTO temp

FROM atleta

WHERE cpf = cpfa;

UPDATE atleta

SET numero = numeroNovo

WHERE cpf = cpfa;

numeroNovo := temp;

END;

/

--67

DECLARE

Var1 atleta.cpf%TYPE;

Var2 atleta.numero%TYPE;

BEGIN

Var1:='09660138405';

Var2:='001';

trocaNumero(Var1,Var2);

END;

/

-- 68

CREATE OR REPLACE FUNCTION media\_premiacao RETURN medalha.premiacao%TYPE IS

media medalha.premiacao%TYPE;

BEGIN

SELECT AVG(premiacao) INTO media

FROM medalha;

RETURN media;

END;

/

-- 69

CREATE FUNCTION retornaNome(cpf\_atleta IN NUMBER) RETURN NUMBER IS

numero\_atleta NUMBER;

BEGIN

SELECT numero INTO numero\_atleta

FROM atleta

WHERE cpf = cpf\_atleta;

RETURN numero\_atleta;

END;

/

-- 73, 76, 80 e 82

CREATE OR REPLACE TRIGGER reducao\_premiacao

BEFORE UPDATE OF premiacao ON medalha

FOR EACH ROW

BEGIN

IF (:OLD.premiacao > :NEW.premiacao) THEN

RAISE\_APPLICATION\_ERROR (-20001, 'Reducao da premiacao nao permitida');

END IF;

END;

/

-- Teste

UPDATE medalha

SET premiacao = 25000.00

WHERE id = 222;

-- 74 e 77

CREATE OR REPLACE TRIGGER Ok

AFTER INSERT ON local

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Insercao feita com sucesso!');

END;

/

-- Teste

INSERT INTO local(id, nome, capacidade) values(16, 'Ilha do Retiro', 370000);

-- 75, 78 e 84

CREATE OR REPLACE TRIGGER insere\_comp

AFTER INSERT ON competicao

FOR EACH ROW

DECLARE new\_id NUMBER;

BEGIN

SELECT id INTO new\_id

FROM competicao

WHERE id = :NEW.id;

INSERT INTO competicao\_modalidade(competicao\_id, modalidade\_id) values(new\_id, 207);

END;

/

**PL - avançadas**

-- 87

SELECT tipo

FROM medalha

WHERE premiacao > media\_premiacao;

-- Usando

CREATE OR REPLACE FUNCTION media\_premiacao RETURN medalha.premiacao%TYPE IS

media medalha.premiacao%TYPE;

BEGIN

SELECT AVG(premiacao) INTO media

FROM medalha;

RETURN media;

END;

/

-- 91

/\*CREATE VIEW pessoa\_view AS

SELECT nome, pais\_id, cpf

FROM pessoa;

CREATE OR REPLACE TRIGGER up\_pessoa

INSTEAD OF UPDATE ON pessoa\_view

FOR EACH ROW

BEGIN

IF :OLD.nome <> :NEW.nome THEN

DBMS\_OUTPUT.PUT\_LINE('Old: '||:OLD.nome||'New: '||:NEW.nome);

END IF;

END;

/\*/