

Trabalho de Banco de Dados

Grupo 3: Companhia Aérea

Funções e Gatilhos no Banco de Dados

| | |
|--------------------------------------|------------|
| Gustavo Fernandes Carneiro de Castro | - 11369684 |
| Guilherme Martiniano de Oliveira | - 11215765 |
| Mateus Miquelino da Silva | - 11208412 |
| Renan Gomes Carneiro de Freitas | - 11391085 |

1ª Função:

Parâmetros:

(p_cod voo.cod_voo%TYPE)

Objetivo:

Esta função imprime quais funcionários participam da tripulação do voo escolhido, e retorna 'true'. Caso não ache nenhum tripulante no voo escolhido, retorna 'false'.

Instrução:

```
CREATE OR REPLACE FUNCTION list_trip (p_cod voo.cod_voo%TYPE)
RETURNS boolean
AS
$$
    DECLARE
        cs_tripu CURSOR FOR SELECT tripulação_funcionário_id_func FROM
Tripulação_voo WHERE voo_cod_voo = p_cod;
        v_func Funcionário%ROWTYPE;
        v_tripu Tripulação_voo.tripulação_funcionário_id_func%TYPE;

    BEGIN
        OPEN cs_tripu;

        FETCH cs_tripu INTO v_tripu;

        IF v_tripu IS null THEN

            RETURN 'f';

        ELSE

            WHILE FOUND LOOP
                SELECT INTO v_func * FROM Funcionário WHERE v_tripu = id_func;
                RAISE NOTICE 'ID FUNC: % Nome: % % Tipo: %', v_func.id_func,
v_func.prim_nome, v_func.sobrenome, v_func.tipo;
                FETCH cs_tripu INTO v_tripu;
            END LOOP;

            RETURN 't';

        END IF;

        CLOSE cs_tripu;
    END
$$
LANGUAGE 'plpgsql';
```

Resultados:

```
ciaaerea=> select list_trip(1);
NOTICE: ID FUNC: 14 Nome: Ivete Sangalo Tipo: Pilota
NOTICE: ID FUNC: 13 Nome: Shakira Mebarak Tipo: Copilota
NOTICE: ID FUNC: 2 Nome: Manfred von Richtoffen Tipo: Piloto
list_trip
-----
t
(1 registro)
```

```
ciaaerea=> select list_trip(0);
list_trip
-----
f
(1 registro)
```

2ª Função:

Parâmetros:

(p_cod_voo Voo.cod_voo%TYPE, preco_novo NUMERIC)

Objetivo:

Esta função altera o preço da passagem. Quando alterada, mostra o preço antigo, o preço atual e de quanto foi o desconto/aumento do preço em porcentagem e retorna 'true'. Caso o voo não exista, retorna 'false'.

Instrução:

```
CREATE OR REPLACE FUNCTION mudar_preco_pass (p_cod_voo Voo.cod_voo%TYPE,
preco_novo NUMERIC)
RETURNS boolean
AS
$$
DECLARE
    porcentagem FLOAT;
    preco_antigo Voo.preco_passag%TYPE;

BEGIN
    SELECT INTO preco_antigo preco_passag FROM Voo WHERE cod_voo = p_cod_voo;
    porcentagem := ((preco_novo / preco_antigo) - 1) * 100;
    IF preco_antigo IS null THEN
        RETURN 'f';
    ELSE
        IF porcentagem = 0 THEN
            RAISE NOTICE 'Preço não alterado!';
        ELSE

            UPDATE Voo SET preco_passag = preco_novo WHERE cod_voo = p_cod_voo;

            IF porcentagem > 0 THEN
                RAISE NOTICE 'Preço Antigo: % Preço Novo: % Aumento de % %%',
preco_antigo, preco_novo, porcentagem;
```

```

        ELSE
            RAISE NOTICE 'Preço Antigo: % Preço Novo: % Desconto de % %%',
                preco_antigo, preco_novo, ABS(porcentagem);
        END IF;

    END IF;

    RETURN 't';

END IF;
END
$$
LANGUAGE 'plpgsql';

```

Resultado:

```

ciaaerea=> select mudar_preco_pass(1, 5000);
NOTICE:  Preço Antigo: 2500  Preço Novo: 5000  Aumento de 100 %
mudar_preco_pass
-----
t
(1 registro)

```

```

ciaaerea=> select mudar_preco_pass(1, 5000);
NOTICE:  Preço não alterado!
mudar_preco_pass
-----
t
(1 registro)

```

```

ciaaerea=> select mudar_preco_pass(0, 5000);
mudar_preco_pass
-----
f
(1 registro)

```

3ª Função:

Parâmetros:

(dia DATE)

Objetivo:

Esta função tem por objetivo contabilizar a quantidade de passagens vendidas em um único dia, retornando este valor.

Instrução:

```
CREATE OR REPLACE FUNCTION vendas_diarias (dia DATE)
RETURNS NUMERIC
AS
$$
DECLARE
    v_cod_voo Voo%ROWTYPE;
    qtde NUMERIC;
    somatorio NUMERIC;

BEGIN

    qtde := 0;
    somatorio := 0;

    FOR v_cod_voo IN SELECT * FROM Voo WHERE data_i = dia LOOP
        qtde := qtde + 1;
        somatorio := v_cod_voo.preco_passag + somatorio;
    END LOOP;

    RAISE NOTICE 'Quantidade de Passagens: % Lucro Liquido: %', qtde, somatorio;

    RETURN somatorio;
END;
$$
LANGUAGE 'plpgsql';
```

Resultado:

```
ciaaerea=> select vendas_diarias('2020-06-10');
NOTICE: Quantidade de Passagens: 1 Lucro Liquido: 2600
 vendas_diarias
-----
                2600
(1 registro)
```

```
ciaaerea=> select vendas_diarias('2020-06-11');
NOTICE: Quantidade de Passagens: 0 Lucro Liquido: 0
 vendas_diarias
-----
                0
(1 registro)
```

1º Gatilho:

Condição:

BEFORE DELETE ON Funcionário FOR EACH ROW

Objetivo:

Este gatilho e função tem por objetivo deletar as tuplas de diferentes tabelas relacionadas ao funcionário deletado, ao deletar um funcionário.

Instrução:

```
CREATE OR REPLACE FUNCTION demitir()
RETURNS TRIGGER
AS
$$
    BEGIN
        DELETE FROM experiencia_linguas WHERE funcionario_id_func =
        OLD.id_func;
        DELETE FROM presencial WHERE funcionario_id_func = OLD.id_func;
        DELETE FROM tripulacao WHERE funcionario_id_func = OLD.id_func;
        DELETE FROM especialidade_voo WHERE
        tripulacao_funcionario_id_func = OLD.id_func;
        RETURN OLD;

    END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER excluir_func
BEFORE DELETE ON Funcionario FOR EACH ROW
EXECUTE FUNCTION demitir();
```

Resultado:

Antes:

```
ciaaerea=> select * from funcionario;
id_func | setor | prim_nome | sobrenome | sexo | data_nasc | est_civil | rg | cpf | nacionalidade | tipo
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | Um | Thomas | Jefferson | Masc | 1990-04-30 | Solteiro | 1234 | 4321 | Estadunidense | Presencial
9 | Um | John | Lewis | Masc | 1987-12-09 | Solteiro | 53264 | 12582 | Estadunidense | Presencial
2 | B05 | Manfred | von Richtoffen | Masc | 1892-03-02 | Solteiro | 2222 | 2222 | Polonês | Piloto
10 | B05 | James | Doolittle | Masc | 1896-12-14 | Solteiro | 23 | 31556 | Estadunidense | Piloto
11 | B05 | Erich | Hartmann | Masc | 1922-04-19 | Solteiro | 34 | 31256 | Alemão | Piloto
12 | B05 | Roberto | Carlos | masc | 1941-04-19 | Viuvo | 6969 | 96969 | Brasileiro | Aeromoço
14 | B05 | Ivete | Sangalo | fem | 1972-05-27 | Casada | 1225 | 66566 | Brasileira | Pilota
6 | Um | Luisa | Guedes | Fem | 1990-06-04 | Solteira | 43413 | 53435 | Brasileira | Telemarketing
4 | M77 | Guilherme | Oliveira | Masc | 2000-02-19 | Solteiro | 4444 | 4444 | Brasileiro | Presencial
7 | Um | Nikishna | Ilyinishna | Fem | 1990-10-10 | Solteira | 82456 | 54324 | Russa | Faxineiro
3 | F09 | Gustavo | Castro | Masc | 2000-12-27 | Casado | 3333 | 3333 | Brasileiro | Presencial
8 | Um | Marcelo | Antonio | Masc | 1956-07-04 | Solteiro | 18365 | 50854 | Inglês | Executivo
5 | M77 | José | Codsworth | Masc | 1944-06-06 | Solteiro | 5555 | 5555 | Brasileiro | Presencial
13 | B05 | Shakira | Mebarak | fem | 1977-02-02 | Casada | 5656 | 5665 | Colombiana | Copilota
99 | txt | teste | teste | Masc | 2000-01-01 | Solteiro | 9999 | 9999 | Brasileiro | Piloto
(15 registros)

ciaaerea=> select * from tripulacao;
funcionario_id_func | passaporte
-----+-----
2 | 1212
10 | 13412
11 | 1661
12 | 16991
13 | 5545
14 | 548885
99 | 9999
(7 registros)
```

Depois:

```
ciaaerea=> delete from funcionario where id_func = 99;
DELETE 1
ciaaerea=> select * from funcionario;
id_func | setor | prim_nome | sobrenome | sexo | data_nasc | est_civil | rg | cpf | nacionalidade | tipo
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | Um | Thomas | Jefferson | Masc | 1990-04-30 | Solteiro | 1234 | 4321 | Estadunidense | Presencial
9 | Um | John | Lewis | Masc | 1987-12-09 | Solteiro | 53264 | 12582 | Estadunidense | Presencial
2 | B05 | Manfred | von Richtoffen | Masc | 1892-03-02 | Solteiro | 2222 | 2222 | Polonês | Piloto
10 | B05 | James | Doolittle | Masc | 1896-12-14 | Solteiro | 23 | 31556 | Estadunidense | Piloto
11 | B05 | Erich | Hartmann | Masc | 1922-04-19 | Solteiro | 34 | 31256 | Alemão | Piloto
12 | B05 | Roberto | Carlos | masc | 1941-04-19 | Viuvo | 6969 | 96969 | Brasileiro | Aeromoço
14 | B05 | Ivete | Sangalo | fem | 1972-05-27 | Casada | 1225 | 66566 | Brasileira | Pilota
6 | Um | Luisa | Guedes | Fem | 1990-06-04 | Solteira | 43413 | 53435 | Brasileira | Telemarketing
4 | M77 | Guilherme | Oliveira | Masc | 2000-02-19 | Solteiro | 4444 | 4444 | Brasileiro | Presencial
7 | Um | Nikishna | Ilyinishna | Fem | 1990-10-10 | Solteira | 82456 | 54324 | Russa | Faxineiro
3 | F09 | Gustavo | Castro | Masc | 2000-12-27 | Casado | 3333 | 3333 | Brasileiro | Presencial
8 | Um | Marcelo | Antonio | Masc | 1956-07-04 | Solteiro | 18365 | 50854 | Inglês | Executivo
5 | M77 | José | Codsworth | Masc | 1944-06-06 | Solteiro | 5555 | 5555 | Brasileiro | Presencial
13 | B05 | Shakira | Mebarak | fem | 1977-02-02 | Casada | 5656 | 5665 | Colombiana | Copilota
(14 registros)

ciaaerea=> select * from tripulacao;
funcionario_id_func | passaporte
-----+-----
2 | 1212
10 | 13412
11 | 1661
12 | 16991
13 | 5545
14 | 548885
(6 registros)
```

2º Gatilho:

Condição:

AFTER INSERT ON Voo FOR EACH ROW

Objetivo:

Este gatilho e função têm por objetivo definir o valor de duração do voo, conforme os horários de chegada e saída do voo adicionado, ao inserir um novo voo.

Instrução:

```
CREATE OR REPLACE FUNCTION duracao_voo()
RETURNS TRIGGER
AS
$$
BEGIN

    UPDATE Voo SET duracao_voo = NEW.horario_f - NEW.horario_i WHERE cod_voo =
    NEW.cod_voo;

    RETURN OLD;
END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER tg_duracao_voo
AFTER INSERT ON Voo FOR EACH ROW
EXECUTE FUNCTION duracao_voo();
```

Resultado:

```
ciaaerea=> select cod_voo, horario_i, horario_f, duracao_voo from voo;
cod_voo | horario_i | horario_f | duracao_voo
-----+-----+-----+-----
      2 | 21:00:00 | 22:00:00 | 01:00:00
      3 | 23:35:00 | 04:35:00 | 05:00:00
      4 | 17:50:00 | 22:00:00 | 07:00:00
      5 | 07:40:00 | 08:30:00 | 18:00:00
      1 | 22:00:00 | 02:00:00 | 04:00:00
(5 registros)
```

```
INSERT 0 1
ciaaerea=> select horario_i, horario_f, duracao_voo from voo;
horario_i | horario_f | duracao_voo
-----+-----+-----
21:00:00 | 22:00:00 | 01:00:00
23:35:00 | 04:35:00 | 05:00:00
17:50:00 | 22:00:00 | 07:00:00
07:40:00 | 08:30:00 | 18:00:00
22:00:00 | 02:00:00 | 04:00:00
05:00:00 | 11:00:00 | 06:00:00
(6 registros)
```

3º Gatilho:

Condição:

AFTER INSERT ON Tripulação_voo FOR EACH ROW

Objetivo:

Este gatilho tem por objetivo imprimir o valor por viagem ganho por cada membro da tripulação (Pilot%, Copilot% e Aeromoç%), com pesos diferentes para cada membro, ao atribuir um tripulante a um voo.

Instrução:

```
CREATE OR REPLACE FUNCTION pagamento_tripu()
RETURNS TRIGGER
AS
$$
DECLARE
    cs_tripu CURSOR FOR SELECT * FROM Tripulação_voo WHERE voo_cod_voo =
NEW.voo_cod_voo;
    reg RECORD;
    pagam NUMERIC;
    info_func Funcionário%ROWTYPE;

BEGIN
    OPEN cs_tripu;
    FETCH cs_tripu INTO reg;

    WHILE FOUND LOOP

        SELECT INTO info_func * FROM Funcionário WHERE id_func =
reg.tripulação_funcionário_id_func;

        IF info_func.tipo LIKE 'Pilot%' THEN

            SELECT INTO pagam 50 * EXTRACT(epoch from duracao_voo)/3600 FROM Voo
WHERE cod_voo = reg.voo_cod_voo;

        ELSE
            IF info_func.tipo LIKE 'Copilot%' THEN

                SELECT INTO pagam 42 * EXTRACT(epoch from duracao_voo)/3600 FROM Voo
WHERE cod_voo = reg.voo_cod_voo;

            ELSE
                IF info_func.tipo LIKE 'Aeromoç%' THEN

                    SELECT INTO pagam 35 * EXTRACT(epoch from duracao_voo)/3600 FROM
Voo WHERE cod_voo = reg.voo_cod_voo;

                END IF;
            END IF;
        END IF;

        RAISE NOTICE 'Pagamento de % % Profissão: % em R$', info_func.prim_nome,
info_func.sobrenome, info_func.tipo, pagam;

        FETCH cs_tripu INTO reg;
```



```

        END LOOP;

        CLOSE cs_tripu;
        RETURN OLD;
    END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER tg_pagamento_tripu
AFTER INSERT ON Tripulação_voo FOR EACH ROW
EXECUTE FUNCTION pagamento_tripu();

```

Resultado:

```

ciaaerea=> INSERT INTO tripulação_voo values(1, 10);
NOTICE:  Pagamento de James Doolittle  Profissão: Piloto  em R$200
NOTICE:  Pagamento de Ivete Sangalo  Profissão: Pilota  em R$200
NOTICE:  Pagamento de Shakira Mebarak  Profissão: Copilota  em R$168
NOTICE:  Pagamento de Manfred von Richtoffen  Profissão: Piloto  em R$200
INSERT 0 1

```