



EACH

Universidade de São Paulo

Escola de Artes, Ciências e Humanidades

Graduação em Sistemas de Informação

Adriano Barbieri 8921162

Douglas Mizuma 8920964

Laura Castro Vieira 8598822

Virgílio Fernandes Junior 7640870

Artefato 2B

Laboratório de Banco de Dados

Professor Luciano Araújo

São Paulo, SP

2016

Projeto de Laboratório de Bancos de Dados - Regras de negócio e visões

Regras de negócio - gatilhos

1. Cadastro de assentos no momento de criação da aeronave

Uma regra de negócio é definida no registro de aeronaves no sistema. A entidade *Aeronave* possui o campo *Quantidade de Assentos*, que deve ser preenchido no momento da criação de um novo registro de aeronave. O gatilho definido cria registros na tabela *Assentos* de acordo com o campo *Quantidade de Assentos* de cada registro de Aeronave. Assentos não podem ser criados além da capacidade da aeronave.

SQL Padrão:

```
1 CREATE PROCEDURE inserir_assentos(IN n INT, IN aer INT)
2 BEGIN
3     DECLARE v1 INT;
4     SET V1 = 1;
5     WHILE v1 <= n DO
6         INSERT INTO assento(classe, aer_codigo, numero) VALUES ('1', aer, v1);
7         SET v1 = v1 + 1;
8     END WHILE;
9 END;
10
11
12 CREATE TRIGGER aeronave_inserir_assentos
13 AFTER INSERT ON aeronave
14 FOR EACH ROW
15 BEGIN
16     CALL inserir_assentos(new.qtd_assentos, new.aer_codigo);
17 END;
```

Código:

```
1 DELIMITER $$
2 CREATE PROCEDURE inserir_assentos(IN n INT, IN aer INT)
3 BEGIN
4     DECLARE v1 INT;
5     SET V1 = 1;
6     WHILE v1 <= n DO
7         INSERT INTO assento(classe, aer_codigo, numero) VALUES ('1', aer, v1);
8         SET v1 = v1 + 1;
9     END WHILE;
10 END$$
11 DELIMITER ;
12
13 DELIMITER $$
14 CREATE TRIGGER aeronave_inserir_assentos
15 AFTER INSERT ON aeronave
16 FOR EACH ROW
17 BEGIN
18     CALL inserir_assentos(new.qtd_assentos, new.aer_codigo);
19 END$$
20 DELIMITER ;
```

Transcrição:

Este gatilho, denominado "aeronave_inserir_assentos" é disparado após uma inserção de registro na tabela *Aeronave*. Assim como as afirmações, ela também foi especificada através de um delimitador \$\$ do MySQL. Esta trigger executa o procedimento criado na linha 1, chamado "inserir_assentos", que recebe a quantidade de assentos existentes no registro da aeronave inserida (new.qtd_assentos) e o código de tal aeronave (new.aer_codigo) para que possa inserir na tabela *Assento* um registro para cada um dos assentos relacionados àquela aeronave até a quantidade de assentos existentes estipulada no registro da aeronave.

Casos de teste:

- Teste 1: Inserção de companhia, aeronave e disparo do gatilho de criação de assentos

```
INSERT INTO companhia (com_codigo, registro, nome) VALUES (1, '1111',  
'Companhia');  
INSERT INTO aeronave(aer_codigo, modelo, qtd_assentos, com_companhia)  
VALUES (200, 123, 10, 1);  
SELECT * FROM assento A WHERE A.aer_aeronave = 200;  
/* Affected rows: 2 Found rows: 10 Warnings: 0 Duration for 3 queries:  
0,156 sec. */
```

2. Voo indisponível quando atinge a lotação máxima.

A tabela *Voo* possui um campo booleano chamado *Disponível*, que permite a visualização rápida de seu status sem necessidade de consultar a tabela *Reserva* sempre que essa informação for requerida, o que traz otimização de consulta. O gatilho definido atualiza esse campo quando o voo atingir a lotação máxima, o que é medido de acordo com o número de reservas para aquele voo (tabela *Passagem*) e sua capacidade (campo *Quantidade de Assentos* da *Aeronave* associada aquele voo).

SQL Padrão:

```
1 CREATE TRIGGER voo_indisponivel_compra_passagem
2 AFTER INSERT ON passagem
3 FOR EACH ROW
4 BEGIN
5     DECLARE qtd_assentos INTEGER;
6     DECLARE aer_id INTEGER;
7     DECLARE count_assentos INTEGER;
8     SELECT A.aer_codigo INTO aer_id FROM assento A
9     WHERE new.ass_codigo = A.ass_codigo;
10    SELECT A.qtd_assentos INTO qtd_assentos FROM
11    aeronave A WHERE A.aer_codigo = aer_id;
12    SELECT COUNT(*) INTO count_assentos FROM
13    passagem A WHERE A.voo_codigo = new.voo_codigo;
14    IF count_assentos = qtd_assentos
15    THEN
16        UPDATE voo V SET V.disponivel = 0 WHERE
17        V.voo_codigo = new.voo_codigo;
18    END IF;
19 END;
```

Código:

```
1  DELIMITER $$
2  CREATE TRIGGER voo_indisponivel_compra_passagem
3  AFTER INSERT ON passagem
4  FOR EACH ROW
5  BEGIN
6      DECLARE qtd_assentos INTEGER;
7      DECLARE aer_id INTEGER;
8      DECLARE count_assentos INTEGER;
9      SELECT A.aer_codigo INTO aer_id FROM assento A
      WHERE new.ass_codigo = A.ass_codigo;
10     SELECT A.qtd_assentos INTO qtd_assentos FROM
      aeronave A WHERE A.aer_codigo = aer_id;
11     SELECT COUNT(*) INTO count_assentos FROM
      passagem A WHERE A.voo_codigo = new.voo_codigo;
12     IF count_assentos = qtd_assentos
13     THEN
14         UPDATE voo V SET V.disponivel = 0 WHERE
            V.voo_codigo = new.voo_codigo;
15     END IF;
16 END$$
17 DELIMITER ;
```

Transcrição:

Este gatilho, denominado "voo_indisponivel_compra_passagem", também teve seu delimitador definido como \$\$ no MySQL. Este gatilho é disparado após uma inserção na tabela *Passagem*. Na linha 9 obtém-se o código da aeronave, que é atribuído à variável "aer_codigo". A linha 10 trata da obtenção da quantidade de assentos existentes na aeronave utilizada no voo; este valor é atribuído à variável "qtd_assentos". A linha 11, conta quantas passagens possuem referência ao voo em questão, atribuindo este dado à variável "count_assentos". Por fim, compara-se, na linha 12 se o total de assentos relacionados a passagens é equivalente ao número de assentos da aeronave e, em caso afirmativo, é realizada a atualização do campo de disponibilidade daquele Voo (V.disponivel) para 0, tornando-o indisponível.

Casos de teste:

- Inserção de uma Aeronave com capacidade máxima 2 e Voo no banco de dados:

```
INSERT INTO aeronave (modelo, qtd_assentos, com_companhia) VALUES (123, 2, 1);
```

```
INSERT INTO voo(origem, destino, data_hora_ini, data_hora_fim, aer_codigo,
disponivel, preco) VALUES ('GRU', 'SSS', '2017-05-28-13:00:00',
'2017-05-28-20:00:00', 1, 1, 100.00);
```

```
#voo disponivel = 1
```

- Teste 1: Compra de uma passagem pelo passageiro 1

```
INSERT INTO passagem (passagem.status, passagem.psg_codigo,  
passagem.voo_codigo, passagem.ass_codigo) VALUES ('OK', 1, 1, 1);
```

```
SELECT V.disponivel FROM voo V WHERE V.voo_codigo = 1;  
# voo disponivel = 1
```

- **Teste 2: Compra de uma passagem pelo passageiro 2**

```
INSERT INTO passagem (passagem.status, passagem.psg_codigo,  
passagem.voo_codigo, passagem.ass_codigo) VALUES ('OK', 1, 1, 2);
```

```
SELECT V.disponivel FROM voo V WHERE V.voo_codigo = 1;  
# voo disponivel = 0
```