



PL09 – Triggers, Funções e Procedimentos

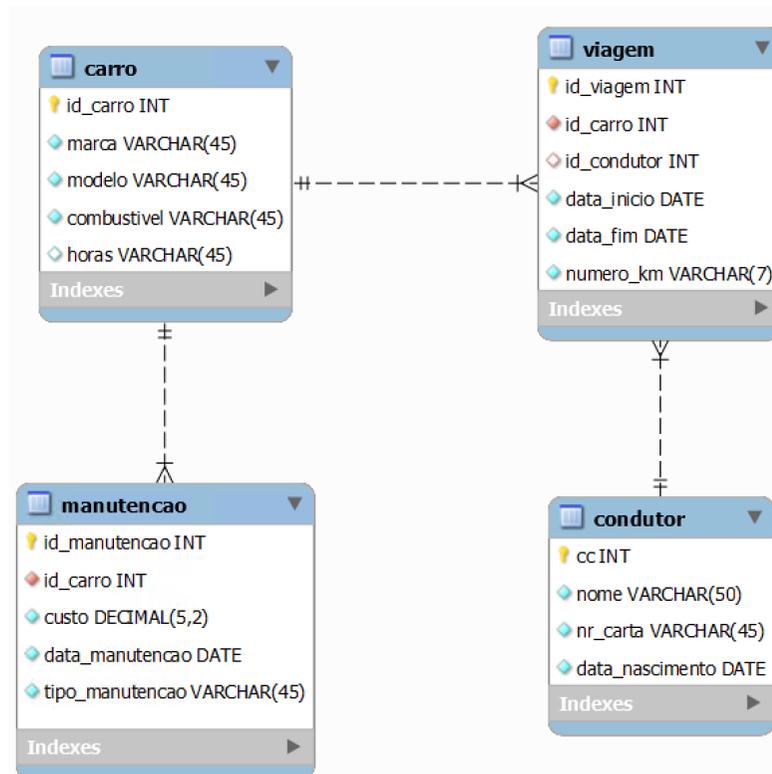
BDCGH - Licenciatura em Engenharia Biomédica

Plano de Aula - PL09

 Triggers, Procedimentos e Funções - Exemplos

 Ficha Exercícios (fe05)

Base de Dados



Triggers

Criar 2 triggers:

- 1) Verificar a idade do condutor ao inserir ou atualizar**
- 2) Atualizar a quantidade de horas de um veículo após uma viagem**

Triggers

1) Verificar a idade do condutor ao inserir ou atualizar

```
DELIMITER $$
CREATE TRIGGER verificar_idade_condutor_ins
BEFORE INSERT ON ubers.condutor
FOR EACH ROW
BEGIN
IF TIMESTAMPDIFF(YEAR, NEW.data_nascimento, CURDATE()) < 18 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Condutor deve ter pelo menos 18 anos de idade.';
END IF;
END$$

DELIMITER ;

DELIMITER $$
CREATE TRIGGER verificar_idade_condutor_upd
BEFORE UPDATE ON ubers.condutor
FOR EACH ROW
BEGIN
IF TIMESTAMPDIFF(YEAR, NEW.data_nascimento, CURDATE()) < 18 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Condutor deve ter pelo menos 18 anos de idade.';
END IF;
END$$

DELIMITER ;
```

Triggers

1) Verificar a idade do condutor ao inserir ou atualizar

```
insert into condutor values (12345, 'User1', 1234, '2020-01-01');
```

```
insert into condutor values (12345, 'User1', 1234, '2000-01-01');
```

```
Update condutor set data_nascimento = '2020-01-01' where cc = 12345;
```

```
Update condutor set data_nascimento = '2020-01-01' where cc = 12345;
```

Triggers

2) Atualizar a quantidade de horas de um veículo após uma viagem

```
DELIMITER $$

CREATE TRIGGER atualizar_horas_carro
AFTER INSERT ON ubers.viagem
FOR EACH ROW
BEGIN
    UPDATE ubers.carro
    SET horas = ADDTIME(horas, TIMEDIFF(NEW.data_fim, NEW.data_inicio))
    WHERE id_carro = NEW.id_carro;
END$$

DELIMITER ;
```

Triggers

2) Atualizar a quantidade de horas de um veículo após uma viagem

```
ALTER TABLE ubers.viagem  
CHANGE COLUMN data_inicio data_inicio DATETIME NOT NULL,  
CHANGE COLUMN data_fim data_fim DATETIME NOT NULL;
```

```
insert into uber.viagem values (60, 1, 12345, '2023-01-01 11:00:00', '2023-01-01 12:00:00', 20);
```

Funções

Criar 2 funções:

- 1) Calcular a Idade do Condutor**
- 2) Calcular o custo total de manutenção de um carro**

Funções

1) Calcular a idade do condutor

```
DELIMITER $$  
  
CREATE FUNCTION calcular_idade(data_nasc DATE)  
RETURNS INT  
BEGIN  
    RETURN TIMESTAMPDIFF(YEAR, data_nasc, CURDATE());  
END$$  
  
DELIMITER ;
```

```
select calcular_idade(data_nascimento) from condutor;
```

Funções

2) Calcular o custo total de manutenção de um carro

```
DELIMITER $$
```

```
CREATE FUNCTION custo_total_manutencao(id_car INT)
RETURNS DECIMAL(10,2)
BEGIN
    DECLARE total DECIMAL(10,2);
    SELECT SUM(custo) INTO total FROM ubers.manutencao WHERE id_carro = id_car;
    RETURN total;
END$$
```

```
DELIMITER ;
```

```
select calcular_idade(data_nascimento) from condutor;
```

Procedimentos

Criar 2 procedimentos:

- 1) Inserir um condutor caso ele tenha a idade mínima**
- 2) Adicionar uma entrada de manutenção para um carro**

Procedimentos

1) Inserir um condutor caso ele tenha a idade mínima

DELIMITER \$\$

```
CREATE PROCEDURE adicionar_condutor(in_cc INT, in_nome VARCHAR(50), in_nr_carta VARCHAR(45), in_data_nasc DATE)
BEGIN
  IF calcular_idade(in_data_nasc) < 18 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Condutor deve ter pelo menos 18 anos de idade.';
  ELSE
    INSERT INTO ubers.condutor (cc, nome, nr_carta, data_nascimento) VALUES (in_cc, in_nome, in_nr_carta, in_data_nasc);
  END IF;
END$$
```

DELIMITER ;

Procedimentos

2) Adicionar uma entrada de manutenção para um carro atualizando o custo total de manutenção

```
ALTER TABLE ubers.carro  
ADD COLUMN custo_manutencao_total DECIMAL(10,2) DEFAULT 0;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE registrar_manutencao(in_id_carro INT, in_custo DECIMAL(5,2), in_data DATE, in_tipo VARCHAR(45))  
BEGIN  
    -- Insere o novo registro de manutenção  
    INSERT INTO ubers.manutencao (id_carro, custo, data_manutencao, tipo_manutencao) VALUES (in_id_carro, in_custo,  
in_data, in_tipo);  
  
    -- Atualiza o custo total acumulado de manutenção no registro do carro  
    UPDATE ubers.carro  
    SET custo_manutencao_total = custo_manutencao_total + in_custo  
    WHERE id_carro = in_id_carro;  
END$$
```

```
DELIMITER ;
```



Ficha de Exercícios 05



PL09 – Triggers, Funções e Procedimentos

BDCGH - Licenciatura em Engenharia Biomédica