

EXERCÍCIO 1 - TRIGGERS - BANCO DE DADOS II (entregar na próxima aula)

1) Considere o esquema abaixo (é um novo; diferente do exercício de visão), e faça o que se pede:

a) Criar um gatilho de nome "removeReserva" para, assim que um cliente alugar determinado filme, remover a reserva desse filme que o cliente porventura tenha feito.

b) Criar um gatilho de nome "mudaAtributo" para, quando uma cópia for alugada, seu atributo "estaAlugada" ser modificado para "sim".

c) Considere que cada cliente tenha associado a si um registro de PONTOS obtidos para futuras promoções, computado da seguinte maneira: para cada registro de locação, ele ganha um ponto. Sendo assim, crie um trigger de nome "adicionaLinha" que, para cada novo cliente cadastrado, implica na inserção de uma linha associada a ele na tabela de pontos.

```
create database Locadora_Trigger;
```

```
create table Cliente (  
  codCliente char(4) not null,  
  nome varchar(40),  
  cidade varchar(40),  
  estado char(2),  
  sexo char(1),  
  primary key (codCliente));
```

```
create table FilmeCategoria (  
  codCategoria char(2) not null,  
  nomeCategoria varchar(20),  
  primary key (codCategoria));
```

```
create table Filme (  
  codFilme char(6) not null,  
  titulo varchar(100),  
  anoProducao integer,  
  produtora varchar(50),  
  duracao integer,  
  codCategoria char(2),  
  primary key (codFilme),  
  foreign key (codCategoria) references  
  FilmeCategoria(codCategoria) ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

```
create table Ator (codAtor char(6) not null,  
  nome varchar(40),  
  primary key (codAtor));
```

```
create table AtorFilme ( codAtor char(6) not null,  
  codFilme char(6),  
  nomePersonagem varchar(40),  
  primary key (codAtor,codFilme),  
  foreign key (codAtor) references Ator(codAtor) ON  
  DELETE CASCADE ON UPDATE CASCADE,  
  foreign key (codFilme) references Filme(codFilme)  
  ON DELETE CASCADE ON UPDATE CASCADE);
```

```
create table Pontos (  
  codCliente char(4) not null,  
  pontos INT NOT NULL DEFAULT '0',  
  primary key (codCliente),  
  foreign key (codCliente) references Cliente  
  (codCliente) ON DELETE CASCADE ON UPDATE  
  CASCADE);
```

```
id SERIAL,  
  codCliente char(4) NOT NULL,  
  pontos INT NOT NULL DEFAULT '0',  
  PRIMARY KEY (id),  
  FOREIGN KEY (codCliente) REFERENCES Cliente  
  (codCliente) ON DELETE CASCADE ON UPDATE  
  CASCADE);
```

```
create table Copia (  
  codCopia char (4) not null,  
  tipo varchar(10),  
  preco numeric (8,2) not null,  
  codFilme char(6),  
  estaAlugada char(3) default 'nao',  
  primary key (codCopia),  
  foreign key (codFilme) references Filme(codFilme)  
  ON DELETE CASCADE ON UPDATE CASCADE);
```

```
create table Reserva (  
  codReserva char(3) not null,  
  codCliente char(4),  
  codFilme char(6),  
  codCopia char (4),  
  primary key (codReserva),  
  foreign key (codCliente) references  
  Cliente(codCliente) ON DELETE CASCADE ON UPDATE  
  CASCADE,  
  foreign key (codFilme) references Filme(codFilme)  
  ON DELETE CASCADE ON UPDATE CASCADE);
```

```
create table Aluguel (  
  codAluguel char(8) not null,  
  codCliente char(4),  
  codCopia char(4),  
  valorPago numeric(8,2),  
  primary key (codAluguel),  
  foreign key (codCliente) references  
  Cliente(codCliente) ON DELETE CASCADE ON UPDATE  
  CASCADE,  
  foreign key (codCopia) references Copia(codCopia)  
  ON DELETE CASCADE ON UPDATE CASCADE);
```

VALORES PARA INSERIR E TESTAR...

```
insert into Cliente values ('0001', 'Cleber', 'Campina Grande', 'PB', 'M');
insert into Cliente values ('0002', 'Paula', 'Campina Grande', 'PB', 'F');
insert into Cliente values ('0003', 'Fernando', 'Areia', 'PB', 'M');
insert into Cliente values ('0004', 'Ricardo', 'Patos', 'PB', 'M');
```

```
insert into FilmeCategoria values ('01', 'Romance');
insert into FilmeCategoria values ('02', 'Comedia');
insert into FilmeCategoria values ('03', 'Policial');
insert into FilmeCategoria values ('04', 'Suspense');
insert into FilmeCategoria values ('05', 'Aventura');
insert into FilmeCategoria values ('06', 'Ficcao');
insert into FilmeCategoria values ('07', 'Drama');
```

```
insert into Filme values ('000001', 'Matrix', 2000, NULL, NULL, '05');
insert into Filme values ('000002', 'Pulp Fiction', 1994, NULL, NULL, '03');
insert into Filme values ('000003', 'Forrest Gump', 1994, NULL, NULL, '07');
insert into Filme values ('000004', 'Constantine', 2005, NULL, NULL, '05');
insert into Filme values ('000005', 'Ace Ventura', 2001, NULL, NULL, '02');
insert into Filme values ('000006', 'O Mentiroso', 2002, NULL, NULL, '02');
```

```
insert into Ator values ('000001', 'Keanu Reeves');
insert into Ator values ('000002', 'Jim Carrey');
insert into Ator values ('000003', 'John Travolta');
insert into Ator values ('000004', 'Tom Hanks');
insert into Ator values ('000005', 'Samuel L. Jackson');
insert into Ator values ('000006', 'Carrie-Ann Moss');
```

```
insert into AtorFilme values ('000001', '000001', 'Neo');
insert into AtorFilme values ('000006', '000001', 'Trinity');
insert into AtorFilme values ('000003', '000002', 'Vincent Vega');
insert into AtorFilme values ('000005', '000002', 'Jules Winfield');
insert into AtorFilme values ('000001', '000004', 'Constantine');
insert into AtorFilme values ('000002', '000005', 'Ace Ventura');
insert into AtorFilme values ('000004', '000003', 'Forrest Gump');
```

```
insert into Reserva values ('001', '0001', '000002', '0003');
insert into Reserva values ('002', '0003', '000005', '0007');
insert into Reserva values ('003', '0004', '000004', '0006');
```

```
insert into Copia (codCopia, tipo, preco, codFilme) values ('0001', 'DVD', 5.00, '000001');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0002', 'DVD', 5.00, '000001');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0003', 'DVD', 5.00, '000002');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0004', 'DVD', 5.00, '000002');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0005', 'DVD', 5.00, '000003');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0006', 'DVD', 5.00, '000004');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0007', 'DVD', 5.00, '000005');
insert into Copia (codCopia, tipo, preco, codFilme) values ('0008', 'DVD', 5.00, '000006');
```

```
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000001', '0001', '0002', 5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000002', '0001', '0003', 5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000003', '0002', '0001', 5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000004', '0003', '0006', 5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000005', '0002', '0008', 5.00);
```

```
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000006','0004','0002',5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000007','0003','0004',5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000008','0003','0001',5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000009','0003','0007',5.00);
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000010','0001','0007',5.00);
```

```
insert into Pontos (id, CodCliente, pontos) values (1,'0001',3);
insert into Pontos (id, CodCliente, pontos) values (2,'0002',2);
insert into Pontos (id, CodCliente, pontos) values (3,'0003',4);
insert into Pontos (id, CodCliente, pontos) values (4,'0004',1);
```

OBSERVAÇÕES:

1) COMANDOS PARA **INSERIR NA TRIGGER** É SEMELHANTE À INSERÇÃO EM TABELAS, mas não se esqueça de usar o **operador NEW** nos valores de atributos. Exemplo:

INSERT INTO (atributo1, atributo2, ..., atributo N) VALUES (valor1, valor 2, ..., valor N);

2) DA MESMA FORMA, COMANDOS PARA **EXCLUIR NA TRIGGER** É SEMELHANTE À EXCLUSÃO EM TABELAS, mas não se esqueça de usar o **operador OLD** no(s) atributo(s) da condição. Exemplo:

DELETE FROM <TABLE>

WHERE <CONDIÇÃO>

a)

#Linhas de inserção para testes

```
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000011', '0001', '0003', 5.00)
```

```
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000012', '0001', '0001', 5.00)
```

#linha de inserção para teste: o mesmo cliente faz 2 reservas (000001)

```
insert into Reserva values ('004', '0001', '000001', '0001')
```

b)

```
SELECT * FROM COPIA #verificar que copia 0008 está “não”
```

#Linha de inserção para teste

```
insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000013', '0004', '0008', 5.00)
```

```
SELECT * FROM COPIA #verificar que copia 0008 está “sim”, isto é, foi alugada, conforme comando acima
```

c)

#Linha de inserção para testes

```
insert into Cliente (codCliente, nome, cidade, estado, sexo) values ('0005', "Camila", "Itauna", 'MG', 'F')
```

```
SELECT * FROM PONTOS #verificar se a cliente Camila, código 0005 está na tabela pontos!
```
