

EXERCÍCIOS BANCO DE DADOS II – VISÕES (VIEWS) (EM DUPLA)

- Utilizando o pgAdmin **crie o esquema** abaixo e faça o que se pede:

```
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));  
  
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),  
  foreign key (codFilme) references Filme(codFilme));  
  
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));  
  
create table Reserva (  
  codReserva char(3) not null,  
  codCliente char(4),  
  codFilme char(6),  
  codCopia char (4),  
  data_reserva date,  
  primary key (codReserva),  
  foreign key (codCliente) references  
  Cliente(codCliente),  
  foreign key (codFilme) references Filme(codFilme));  
  
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),  
  foreign key (codCopia) references  
  Copia(codCopia));
```

→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', null);

insert into Reserva values ('002','0003','000005','0007', null);

insert into Reserva values ('003','0004','000004','0006', null);

insert into Copia (codCopia, tipo, preco, codFilme) values ('0001','DVD',3.00,'000001');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0002','DVD',3.00,'000001');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0003','DVD',3.00,'000002');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0004','DVD',3.00,'000002');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0005','DVD',3.00,'000003');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0006','DVD',3.00,'000004');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0007','DVD',3.00,'000005');

insert into Copia (codCopia, tipo, preco, codFilme) values ('0008','DVD',3.00,'000006');

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000001','0001','0002',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000002','0001','0003',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000003','0002','0001',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000004','0003','0006',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000005','0002','0008',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000006','0004','0002',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000007','0003','0004',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000008','0003','0001',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000009','0003','0007',3.00);

insert into Aluguel (codAluguel, codCliente, codCopia, valorPago) values ('00000010','0001','0007',3.00);

EXERCÍCIOS:

1. a) Crie uma visão “filmes_locadora” que retorne o título de todos os filmes existentes na locadora e a categoria dos mesmos.

b) Faça uma consulta que retorne quantos filmes de comédia existem na locadora, a partir da visão “filmes_locadora”.

2. a) Crie uma visão “cliente_filme” que retorne o título de um filme e o nome de um cliente, a partir da junção das tabelas CLIENTE, ALUGUEL, FILME e COPIA.

b) Faça uma consulta dos filmes que o cliente FERNANDO alugou, a partir da visão “cliente_filme” criada na letra A.

c) Faça uma consulta que retorne quais clientes alugaram o filme Matrix, a partir da visão “cliente_filme” criada na letra A.

3. Considere a seguinte visão e responda as questões:

```
CREATE VIEW tamanho_area_cidades_uf_view (uf, nome, area_municipio, area_estado) AS
SELECT a.uf_sigla, b.nome, b.tamanho_area, a.tamanho_area
FROM unidade_da_federacao a JOIN cidade b ON a.codigo_ibge=b.codigo_ibge
```

a) Crie uma outra visão de nome “areas_view” a partir da visão criada acima e que contenha somente os **campos renomeados** relacionados às áreas territoriais (do município e do estado).

b) Faça uma consulta para recuperar os dados:

- i. da visão do enunciado;
- ii. da visão criada na letra a.

c) Escreva o comando para excluir a visão criada na letra a.

4. (VUNESP - 2009 - CETESB - Analista de TI - Administração de Dados - adaptado) Julgue o item a seguir como sendo V (verdadeiro) ou F (falso) e **justifique sua resposta**.

() O comando SQL para criar uma visão chamada Eye, abrangendo os atributos A1 e A2 da tabela Pen, é: CREATE VIEW Pen AS SELECT A1, A2 FROM Eye.

5. Com base no exercício número 5 da revisão e nas respostas disponibilizadas, crie (invente) 2 consultas com visões, que contenha junções de pelo menos 2 tabelas.