Salve, pessoal! Seguindo o mesmo esquema do último tópico, vamos tentar subir os códigos respostas das questões abaixo no Git. Se você não conseguir e tiver dificuldade, enviar por zip também é possível! Vamos nessa!

As respostas do tópico "Modelagem MER" devem ser os códigos modelando o MER de cada questão seguindo a sintaxe do site https://dbdiagram.io/home como mostrado na aula! Vocês podem tanto colocar as respostar em arquivos separados(ex_1_2.md, exercicio 2 do tópico 1) ou colocar tudo em um arquivo só(ex_1.md, exercícios do tópico 1), comentando acima de cada código o número da questão("// Questão 1").

Já as respostas do tópico Comandos SQL devem ser os códigos em SQL, cumprindo as demandas de cada questão. O esquema pode seguir a mesma ideia da questão 1, colocar as respostas em arquivos separados(ex_2_1.sql, exercício 1 do tópico 1) ou no mesmo arquivo(ex_2.sql, exercícios da questão 2). Bom estudos e coding a todos!

1 Modelagem MER

Se possível, tentem fazer os campos e tabelas em inglês. O RG, CPF e registros específicos não precisam ser traduzidos.

1.1 BD 1 - Escola

Estevan e Waliff pretendem fundar uma pequena plataforma online de ensino de programação online! Só que para isso, eles precisam da sua ajudar para fazer o BD!

Especificação: "**Waliff:** Bom dia! Então, o modelo do BD possuirá professores(com nome, CPF, RG, sexo, email, data de nascimento, linguagens de programação que dominam, anos de experiência em lecionar), alunos(com nome, CPF, RG, data de nascimento, endereço, cidade, estado de residência). Nessa plataforma, um aluno possui somente um professor."

Obs.: Lembre-se que cidade e estado podem ser tabelas também!!!

1.2 BD 2 - Consultorio dentista

João Victor, Lucca e João Pedro fecharam mais um contratinho top pra Struct com um consultorio! E como sabemos que os trainees são cabeça, o Pedro pediu pra vocês nos ajudarem! O projeto possui a seguinte modelagem:
Especificação: "**Pedro:** Valeu pela sua ajuda maninho! O BD do sistema do consultório possui a seguinte modelagem: no consultório temos vários dentistas(com nome, CPF, RG, sexo, email, data de nascimento, e o código CRO), e cada um pode possuir mais de uma especialização em alguma área(ortodontia, cirurgia geral...). Dentro, temos salas de atendimento, sendo que 1 dentista só pode ocupar uma das salas, mas as salas podem ser ocupadas por mais de 1 dentista. Além dessas informações, cada dentista possui vários clientes, além de que cada cliente pode ser atendido por mais de um dentista do consultório. Nesse sistema, temos a possibilidade de marcar uma consulta(com data, sala, cliente e o dentista que vai atender) também."

Dica.: As tabelas associativas às vezes estão contidas na própria questão! Ou seja, algumas tabelas associativas geradas no modelo podem ser reaproveitadas para assumir alguma entidade da especificação.

1.3 BD 3 - Aeroporto

O Hugo Moai é o gerente de TI do aeroporto de brasília, e foi desafiado pelo seu chefe a fazer um BD em 1 semana. Atarefado, ele pediu sua ajuda!

Especificações:"**Hugo Moai:** Fi, temos um aeroporto que precisa de um BD safe para poder armazernar as informações dos voos. O bagulho é o seguinte: temos voos(possui horário de saída, horário previsto de saída,

horário de chegada e horário previsto de chegada), e precisamos registrar as cidades de origem e destino do voo. Para cada voo, temos 1 piloto(com nome, CPF, RG, email, data de nascimento, código ANAC) e vários passageiros(com nome, CPF, RG, sexo, email, data de nascimento). 1 Piloto pode assumir mais de um voo, e cada voo possui um avião(possui modelo, número de registro do avião e quantidade de assentos). Boa sorte mlk!"

2 Comandos SQL

Agora, vamos explorar um pouco do SQL! Para isso, faça o login com o usuário postgres, e entre no shell do PG.
Vamos didaticamente criar um banco em que temos filmes e atores desses filmes, sabendo que um filme pode ter vários atores e cada ator pode atuar em vários filmes.

2.1 Criar um BD e popular

Crie um banco de dados chamado "hollywood", e crie as tabelas *Movies*(que contém só *movie_name* e *release_year*, nessa ordem), *Actors*(que contém *actor_name* e *gender*, que pode ser um varchar; nessa ordem) e *Characters*, ou seja, personagens(que contém *character_name*, e as chaves estrangeiras de ambas tabelas, *Actors* e *Movies*; nessa ordem), que servirá de tabela associativa entre *Movies* e *Actors*.

Depois popule com os seguintes dados(data é no formato ano, mês e dia):

```
INSERT INTO Movies
   (movie_name, release_year)
   Values
   ('Django Unchained', '2013-01-18'),
   ('Inception', '2010-08-06'),
   ('Pulp Fiction', '1995-02-18');
INSERT INTO Actors
   (actor_name, gender)
    Values
   ('Leonardo DiCaprio', 'Male'),
   ('Samuel L. Jackson', 'Male'),
   ('Uma Turman', 'Female');

INSERT INTO Characters
   (character_name, actor_id, movie_id)
    Values
   ('Calvin Candle', 1, 1),
   ('Cobb', 1, 2),
   ('Jules Winnfield', 2, 3);
```

Certo! Próximo passo: popule o BD com seus 2 filmes favoritos, 1 ator de cada um desses filmes e em qual papel eles atuam no filme seguindo o padrão acima.(na resposta, não é necessário reescrever os *inserts* exemplos acima, só a dos seus filmes favoritos).

2.2 Listar os filmes

Agora que temos um banco de dados preenchido, faça uma listagem dos nomes dos filmes que lançaram antes de 2011.
Faça também uma listagem de todas as informações dos personagens cujo nome começam com 'C'.

2.3 Atualizar o nome

Reparamos que o nome de uma atriz está errada. Vamos consertar alterando o nome da atriz da Tabela *Actors* que tem o nome errado "Uma Turman" para "Uma Thurman", com o h que faltava.

2.4 Adicionar campo na tabela

Para deixar mais completo, vamos adicionar um campo de *birth_date* na tabela dos *Actors*. Logo depois, atualize 1 dos atores do BD com sua respectiva data de nascimento.

2.5 Extra: Brincando com Union

Ok! Vamos aprofundar mais um pouco, agora vamos listar os nomes dos atores e o nome dos papéis em que eles atuam. Mesmo se o ator não atuar em nenhum filme ele deve ser exibido(nesse caso, o campo do character_name estará *null*)