

1. ***Apollo um fã de livros de terror e fantasia queria comprar um livro chamado “Sangue entre lágrimas” na Biblioteca After no Endereço y\_. Então com o objetivo de organizar a agenda de Apollo crie uma tabela em Sql que corresponda a preferência do gênero de livros de Apollo, biblioteca(Nome), e endereço da biblioteca.*** Obs: O endereço será de sua escolha, mas lembre-se de utilizar as regras de Normalização.

```
CREATE TABLE genero(  
  genero VARCHAR (1000) NOT NULL,  
  Id_genero INT NOT NULL,  
  PRIMARY KEY (Id_genero)  
);
```

```
CREATE TABLE Biblioteca(  
  CNPJ INT NOT NULL,  
  Nome VARCHAR (1000) NOT NULL,  
  Rua VARCHAR (1000) NOT NULL,  
  Livros_ID_FK INT NOT NULL,  
  Bairro VARCHAR (1000) NOT NULL,  
  Numero iNT NOT NULL,  
  CEP iNT NOT NULL,  
  PRIMARY KEY (CNPJ)  
);
```

```
CREATE TABLE Livro_Biblioteca(  
  Livros_ID_FK iNT NOT NULL,  
  CNPJ iNT NOT NULL,  
  FOREIGN KEY (Livros_ID_FK) REFERENCES Livro(Livros_ID),  
  FOREIGN KEY (CNPJ) REFERENCES biblioteca(CNPJ)  
);
```

```
CREATE TABLE Livro(  
  Setor INT NOT NULL,  
  Livros_ID INT NOT NULL,  
  Estado_de_Conservacao VARCHAR (1000) NOT NULL,  
  Id_genero INT NOT NULL,  
  Status VARCHAR(100) NOT NULL,  
  PRIMARY KEY (Livros_ID),  
  FOREIGN KEY (Id_genero) REFERENCES genero(Id_genero)  
);
```

2. Durante o Expediente de Nyrx uma consultora de elenco de atores, pediu ao seu estagiário produzir uma tabela que retrata a vida profissional dos artista principais da obra a qual estava trabalho. No entanto, a mesma ordenou que fosse feito em Sql, tendo que possuir as seguinte informações: Nome do Ator, data de nascimento, locais trabalhados anteriormente, como se vê dentro de 10 anos, principais papéis anteriores e outras .

```
CREATE TABLE Atores(  
Data_de_nascimento VARCHAR (10000) NOT NULL,  
Quant_anosTrabalhados Int not null,  
Locais_Trabalhados VARCHAR (10000) NOT NULL,  
Visao_Futura VARCHAR (10000) NOT NULL,  
Papeis_anteriores VARCHAR (10000) NOT NULL,  
Curriculo VARCHAR (10000) NOT NULL,  
ID INT NOT NULL,  
PRIMARY KEY (ID)  
);
```

```
CREATE TABLE Telefone(  
Telefone INT Not null,  
Id_telefone Int not null,  
PRIMARY KEY (Id_telefone)  
);
```

```
CREATE TABLE Principais_referencias(  
Principais_referencias VARCHAR (10000) NOT NULL,  
ID_Principais_referencias int NOT NULL,  
PRIMARY KEY (ID_Principais_referencias)  
);
```

```
CREATE TABLE telefone_Atores(  
Atores_ID_FK iNT NOT NULL,  
Id_telefone_FK iNT NOT NULL,  
FOREIGN KEY (Atores_ID_FK) REFERENCES Atores(ID),  
FOREIGN KEY (Id_telefone_FK) REFERENCES Telefone(Id_telefone)  
);
```

```
CREATE TABLE Principais_referencias_Atores(  
Atores_ID_FK iNT NOT NULL,  
Principais_referencias_FK iNT NOT NULL,  
FOREIGN KEY (Principais_referencias_FK) REFERENCES  
Principais_referencias(ID_Principais_referencias),  
FOREIGN KEY (Atores_ID_FK) REFERENCES Atores(ID)  
);
```

3. Baltazar trabalha com uma filial de cinemas há alguns anos, ele possui certa dificuldade para organizar todos os filmes que entram e saem de cartaz. Então, Baltazar te contratou para ajudá-lo nessa tarefa, crie um banco de que possa orientá-lo. Não se esqueça que esse banco de dados deve conter o título do filme, ano de lançamento, gênero(s), duração, diretor e elenco principal, além de uma ligação com as filiais e o tipo de sala que serão exibidos(exemplos: sessão VIP ou 3D).

```
SELECT * FROM demo;
Create table Filiais(
  Rua Varchar(40) Not NULL,
  Bairro Varchar(20) Not NULL,
  edificio Varchar(20) Not NULL,
  cnpj_pk int Not NULL,
  nome Varchar(50) not null,
  Primary key (cnpj_pk )
)
Create table FilmesCartaz(
  Id_Filme_PK int Not NULL,
  título Varchar(50) Not NULL,
  AnoLancamento int not null,
  duracao Varchar(04) not null,
  diretor Varchar(50) not null,
  exibicaoSala Varchar(20) not null,
  cnpj_fk int not null,
  Primary key (Id_Filme_PK ),
  FOREIGN key (cnpj_fk) references filiais (cnpj_pk)
)
Create table GeneroFilme(
  GeneroFilme Varchar(20) Not NULL,
  Id_Filme_fK int NOT NULL,
  FOREIGN key (Id_Filme_fK) references FilmesCartaz (Id_Filme_PK)
)
Create table elenco(
  elenco Varchar (100) not NULL,
  Id_Filme_fK int not NULL,
  FOREIGN key (Id_Filme_fK) references FilmesCartaz (Id_Filme_PK)
)
```

4. Alice tem uma cafeteria chamada CafeManiacos, porém possui dificuldade em organizar todos os seus pedidos. Ajude Alice criando para ela um banco de dados para a sua cafeteria. Obrigatoriamente, esse banco de dados deve possuir as informações sobre a cafeteria, sobre o pedido e sobre o cliente.

```
SELECT * FROM demo;
```

```

Create table Cafeteria(
  cnpj_pk int Not NULL,
  telefone Varchar(12) Not NULL,
  Rua Varchar(40) Not NULL,
  Bairro Varchar(20) Not NULL,
  edificio Varchar(20) Not NULL,
  numero int Not NULL,
  Primary key (cnpj_pk )
)
Create table Pedido(
  Id_pedido_PK int Not NULL,
  cnpj_fk int not null,
  numeroMesa int Not NULL,
  tipoPedido Varchar (20) not null,
  quantidade Varchar(50) not null,
  Primary key (Id_pedido_PK),
  FOREIGN key (cnpj_fk) references cafeteria (cnpj_pk)
)
Create table Cliente(
  Nome Varchar(50) Not NULL,
  numeroMesa_fk int NOT NULL,
  formaPagamento Varchar(10) NOT NULL,
  id_cliente_pk int NOT NULL,
  cnpj_fk int Not NULL,
  Id_pedido_fk int Not NULL,
  Primary key (id_cliente_pk),
  FOREIGN key (cnpj_fk) references cafeteria (cnpj_pk),
  FOREIGN key (Id_pedido_fk) references pedido (Id_pedido_pK)
)

```

**5.Você foi contratado para desenvolver um sistema de gerenciamento em um Database. O sistema deve conter uma lista com nomes de clientes para concorrer a prêmios de algumas lojas de marcas como a Nike, Calvin Klein e Apple. Observações: Cada lista deve conter todos os dados do cliente com telefones e o profissional que atendeu este cliente, inclusive o nome da Empresa em que o ele está atuando, não esqueça que da ligação entre a empresa e o profissional, utilizar as regras de normalização.**

```

CREATE TABLE Empresa(
  nome_empresa VARCHAR(500) not null,
  CPNJ_empresa int not null,
  telefone_empresa int not null,
  ID_empresa int not null,
  PRIMARY KEY(ID_empresa))

```

```
CREATE TABLE clientes(  
  nome_clientes VARCHAR(100) not null,  
  telefone_clientes int not null,  
  cep_clientes int not null,  
  idade int,  
  ID_clientes int not null,  
  PRIMARY KEY(ID_clientes))
```

```
CREATE TABLE profissionais(  
  nome_profissionais VARCHAR(50) not NULL,  
  id_profissionais int not NULL,  
  bloco_atuacao int,  
  id_clientes_fk int not NULL,  
  PRIMARY KEY(id_profissionais),  
  FOREIGN KEY(id_clientes_fk) REFERENCES clientes(id_clientes))
```

```
CREATE TABLE profissionais_empresa(  
  id_profissionais_FK int not null,  
  id_empresa_FK int not null,  
  FOREIGN KEY (id_profissionais_FK) REFERENCES profissionais(id_profissionais),  
  FOREIGN KEY (id_empresa_FK) REFERENCES empresa(id_empresa))
```

```
INSERT into empresa(nome_empresa,CPNJ_empresa,telefone_empresa, ID_empresa)  
VALUES('Nike', 595465150,111111,1),  
      ('Blohsh', 132510131,151021,2);
```

```
INSERT INTO clientes(nome_clientes,telefone_clientes,cep_clientes,ID_clientes)  
VALUES('Luke', 1112, 2000, 10),  
      ('Drew', 01010, 334411, 20);
```

```
INSERT INTO profissionais(nome_profissionais,id_profissionais,id_clientes_fk)  
VALUES('William',100,20),  
      ('Oconnel', 500,10);
```

```
INSERT INTO profissionais_empresa(id_profissionais_FK,id_empresa_FK)  
VALUES(100,2),  
      (500,1);
```

```
SELECT * FROM clientes;  
SELECT * FROM profissionais;  
SELECT * FROM empresa;  
SELECT * FROM profissionais_empresa
```