

Banco de Dados I

Normalização

- O conceito de normalização foi introduzido por E. F. Codd em 1970.
- O processo de identificação dos agrupamentos necessários e da localização correta de cada atributo consiste num conjunto de técnicas designadas por normalização.
- Metodologia para organizar os atributos nas tabelas de modo que a redundância entre os atributos não-chave seja eliminada.

Normalização – 1ª Forma Normal

Dizemos que um esquema de relação R está na primeira forma normal (1FN) se os domínios de todos os atributos de R são atômicos.

- Definida para NÃO permitir atributos multivalorados e atributos compostos.

Normalização – 1ª Forma Normal

1ª FORMA NORMAL (1ªFN)

Considere a tabela tbCliente abaixo sem normalização:

CLI_COD	CLI_NOME	CLI_TELEFONE	CLI_ENDERECO
1	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12.536-965
2	Maria	3265-8596	Rua Onze, 64 Moema 65.985-963
3	Jânio	8545-8956 9598-6301	Praça Ramos Liberdade 68.858-633

multivalorado

composto

Normalização – 1ª Forma Normal

Normalizando

CLI_COD	CLI_NOME	CLI_TELEFONE	CLI_RUA	CLI_NUM	CLI_BAIRRO	CLI_CEP
1	José	9563-6352 9847-2501	Rua Seis	85	Morumbi	12.536-965
2	Maria	3265-8596	Rua Onze	64	Moema	65.985-963
3	Jânio	8545-8956 9598-6301	Praça Ramos	NULL	Liberdade	68.858-633

multivalorado ↗

Ainda não está na 1ª Forma Normal.

Normalização – 1ª Forma Normal

Normalizando

CLI_COD	CLI_NOME	CLI_RUA	CLI_NUM	CLI_BAIRRO	CLI_CEP
1	José	Rua Seis	85	Morumbi	12.536-965
2	Maria	Rua Onze	64	Moema	65.985-963
3	Jânio	Praça Ramos	NULL	Liberdade	68.858-633

CLT_CODIGO	CLI_COD (FK)	CLT_TELEFONE
1	1	9563-6352
2	1	9847-2501
3	2	3265-8596
4	3	8545-8956
5	3	9598-6301

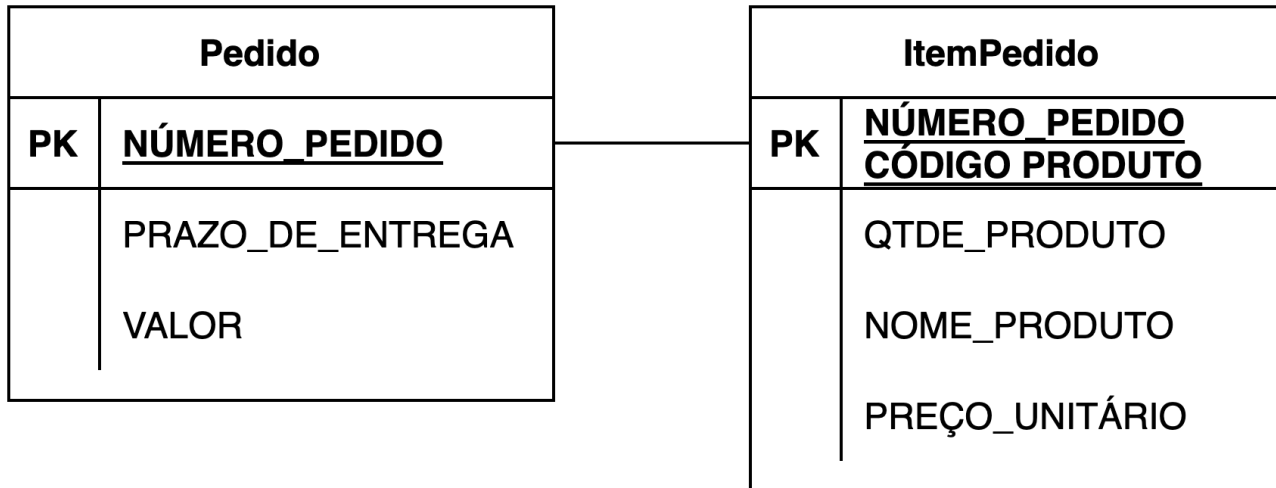
Normalização – 2ª Forma Normal

Uma relação está em 2ª FN se todos os atributos não chave forem totalmente dependentes da chave primária (dependente de toda a chave e não apenas de parte dela).

- Assim, cada atributo não chave deve depender da chave.
- Esse caso aplica-se para entidades cuja chave primária é composta.

Normalização – 2ª Forma Normal

Dependência Funcional



Normalização – 2ª Forma Normal

Dependência Funcional

Uma coluna ou conjunto de colunas A é dependente funcional de um outra coluna B, se a cada valor de B existir nas linhas da tabela um único valor de A. Em outras palavras, A depende funcionalmente de B.

Exemplo:

Considere uma tabela PEDIDO: a coluna PRAZO_DE_ENTREGA depende funcionalmente da coluna NÚMERO_PEDIDO. Em outras palavras, o NÚMERO_PEDIDO determina o PRAZO_ENTREGA.

Normalização – 2ª Forma Normal

Dependência Funcional Total

Na ocorrência de uma chave primária composta, dizemos que um atributo ou conjunto de atributos depende de forma completa ou total desta chave primária composta quando para cada valor da chave está associado um valor para cada atributo.

Exemplo: Na entidade ItemPedido, o atributo QTDE_PRODUTO depende de forma total da chave primária composta (NÚMERO_PEDIDO + CÓDIGO_PRODUTO).

Normalização – 2ª Forma Normal

Dependência Funcional Parcial

Quando uma coluna ou conjunto de colunas A depende de outra coluna B que faz parte da chave primária composta. Para cada valor da coluna B existe um valor associado para a coluna A.

A dependência funcional parcial só existe quando a tabela possui chave primária composta.

Exemplo: Na entidade ItemPedido, as colunas NOME_PRODUTO, PRECO_UNITARIO dependem de forma parcial da chave primária composta (NÚMERO_PEDIDO + CÓDIGO_PRODUTO) através da coluna CÓDIGO_PRODUTO.

Normalização – 2ª Forma Normal

Considere a tabela tbProjetoEmpregado abaixo:

CódProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Sílvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
PAG02	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

Normalização – 2ª Forma Normal

Normalizando

Proj:

CódProj	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

Emp:

CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Sílvio	A2	4
1214	Carlos	A2	4
8191	Mário	A1	4
4112	João	A2	4
6126	José	B1	9

ProjEmp:

CódProj	CodEmp	DataIni	TempAI
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
PAG02	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

Normalização – 3ª Forma Normal

Uma relação está em 3ª FN se todos os atributos que não são chave são independentes, isto é, não existe funções que definam um ao outro. Portanto, sempre a chave deve definir toda a tabela.

- Atributos não-chaves não podem definir outros atributos não-chaves. Mais formalmente, a 3ª FN não permite que um atributo não-chave seja dependente de outro.

Normalização – 3ª Forma Normal

Considere a tabela tblItemNota abaixo (sem normalização).

<u>ITE_COD</u>	PRO_COD	ITE_PRECO	ITE_QTDE	ITE_TOTAL
1	1	0,70	2	1,40
2	2	1,20	4	4,80
3	3	0,50	2	1,00

$$\text{ITE_TOTAL} = \text{ITE_PRECO} * \text{ITE_QTDE}$$

Normalização – 3ª Forma Normal

Normalizando

tblItemNota

ITE_COD	PRO_COD	ITE_PRECO	ITE_QTDE
1	1	0,70	2
2	2	1,20	4
3	3	0,50	2

Integridade dos Dados

Integridade dos Dados

Um dos mais importantes princípios de um modelo de banco de dados é o princípio da integridade dos dados, realizado através de restrições que são condições obrigatórias impostas pelo modelo para manter a consistência dos dados.

As restrições de integridade são:

- Integridade de domínio;
- Integridade de entidade;
- Integridade de chave;
- Integridade referencial.

Integridade dos Dados

- **Restrição de integridade de domínio:**

Especificam que o valor de um atributo deve obedecer a definição dos valores admitidos para a coluna, ou seja, devem pertencer ao domínio deste atributo.

- **Restrição de integridade de entidade:**

Uma chave primária não pode assumir valor nulo em qualquer tupla da relação.

- **Restrição de integridade de chave:**

Cada atributo da chave primária deve assumir valor único em todas as tuplas da relação.

Integridade dos Dados

- Restrição de integridade referencial:**

Uma tupla em uma relação que se refere a outra relação, deve se referenciar a uma tupla existente nesta relação. A chave primária numa relação A1 é chave estrangeira numa relação A2.

Empregado
nome
salário
cod_depto (FK)

Departamento
nome
cod_depto (PK)
nome_chefe
num_depto



Criação do Banco de Dados

SQL (*Structured Query Language*)

A sigla SQL refere-se a *Structured Query Language*.

A SQL é dividida nos seguintes componentes:

- **DCL (Linguagem de Controle de Dados):** Provê a segurança interna do banco de dados.
- **DDL (Linguagem de Definição de Dados):** Permite a criação dos componentes do banco de dados.
- **DML (Linguagem de Manipulação de Dados):** Permite a manipulação dos dados armazenados.
- **DQL (Linguagem de Consulta de Dados):** Permite extrair dados do banco de dados.

Criação do Banco de Dados

DDL: Linguagem de Definição de Dados

- A linguagem SQL, por meio da DDL, disponibiliza um conjunto de comandos para criação (CREATE), alteração (ALTER) e remoção (DROP) de tabelas e outras estruturas.
- Embora a maioria dos SGBDs disponibiliza ferramentas que permitem a criação de Banco de Dados, é possível criar o próprio banco de dados a partir de comandos SQL.

Criação do Banco de Dados



<https://www.microsoft.com/pt-br/sql-server/sql-server-downloads>

<https://sqliteonline.com/>

Criação do Banco de Dados

-Para criar o banco de dados, precisamos ficar atentos para os seguintes itens:

- Tipo de dados;
- *Null e not null*;
- Chaves primárias, estrangeiras e únicas.

Criação do Banco de Dados

Tipo de dados	Valores aceitáveis
bigint	Valores inteiros de -2^{63} a $2^{63}-1$
int	Valores inteiros de -2^{31} a $2^{31}-1$
smallint	Valores inteiros de -2^{15} a $2^{15}-1$
bit	Valores inteiros com valor 0 ou 1
decimal	Valores fracionários de $-10^{38}+1$ a $10^{38}-1$
numeric	Valores fracionários de $-10^{38}+1$ a $10^{38}-1$
money	Valores monetários de -2^{63} a $2^{63}-1$
datetime	Valores de data e hora de 1/1/1753 a 31/12/9999
char	Valores de caracteres de comprimento definido (n: 1 a 8000 caracteres)
varchar	Valores de caracteres de comprimento variável (n: 1 a 8000 caracteres)
text	Valores de caracteres de comprimento variável (máx $2^{31}-1$ caracteres)
varbinary	Valores binários de comprimento variável (máx 8000 bytes)
image	Dados de comprimento variável (máx $2^{31}-1$ bytes)

Criação do Banco de Dados

Null e not null

Os atributos do banco de dados podem permitir null ou não.

-Permitir null: significa que o campo **pode ser vazio**. Por exemplo: Ao armazenar os dados do cliente, o telefone pode não ser informado, então telefone deve permitir null.

-Não permitir null: significa que o atributo **sempre será preenchido**. Por exemplo: Ao armazenar os dados do cliente, o nome sempre será informado, então nome não deve permitir null.

Criação do Banco de Dados

Chaves primárias representam o identificador da tabela;

- Em uma tabela, apenas um atributo pode ser marcado como chave primária;
- A chave primária não pode aceitar null;
- Por padrão, identifica-se um campo código auto-incremento (identidade) para ser a chave primária.

Uma tabela pode ter somente uma coluna de identidade. Além disso, o valor de uma coluna desse tipo é gerado automaticamente pelo SQL Server e não pode ser alterado ou inserido.

Criação do Banco de Dados

Chaves estrangeiras representam a chave primária de outra tabela.

Regras de exclusão:

- Sem Ação
- Cascata
- Definir nulo
- Definir padrão

Unique

Como em uma tabela apenas um atributo é definido como chave primária (identificador que não se repete), existem algumas tabelas que outro atributo não pode se repetir.

Criação do Banco de Dados

Como criar o script?

Primeiro precisamos criar o banco de dados. Para isso utilizaremos o seguinte script:

```
CREATE DATABASE SistemaAcademico
```

representa o nome do banco de dados

Criação do Banco de Dados

Como criar o script?

Depois criaremos as tabelas e os atributos, definindo o tipo, se permite null ou não, chaves primárias, únicas e estrangeiras (caso existam).

Criação do Banco de Dados

```
CREATE TABLE tbAluno
```

```
(
```

```
    ALU_CODIGO int not null primary key identity,
```

```
    ALU_NOME varchar(50) not null,
```

```
    ALU_RG varchar(20) not null unique,
```

```
    ALU_RUA varchar(50) not null,
```

```
    ALU_NUMERO int null,
```

```
    ALU_CIDADE varchar(50) not null,
```

```
    ALU_UF varchar(2) not null
```

```
)
```

```
CREATE TABLE tbAlunoTelefone
```

```
(
```

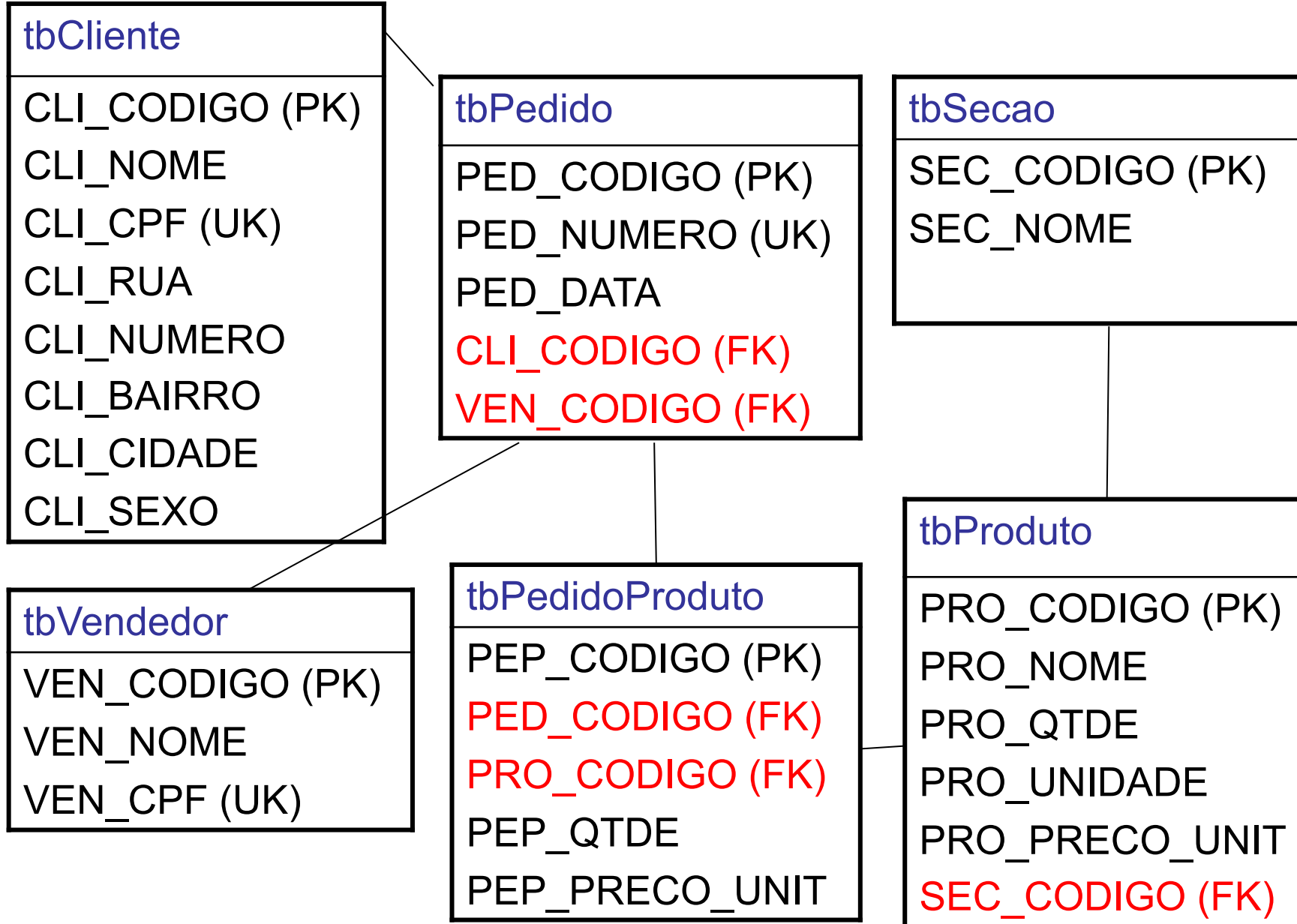
```
    ALT_CODIGO int not null primary key identity,
```

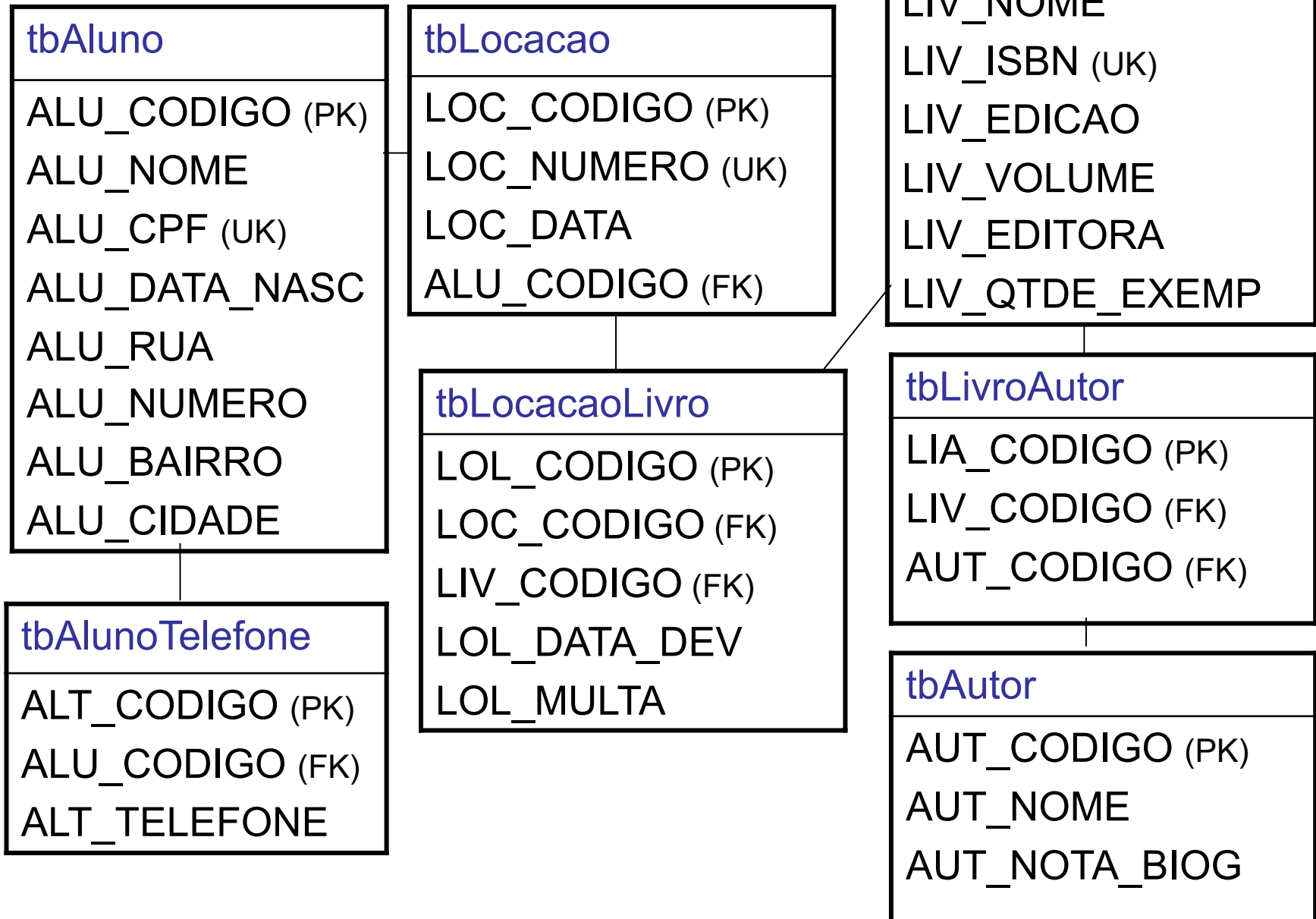
```
    ALT_TELEFONE varchar(13) null,
```

```
    ALU_CODIGO int not null foreign key references tbAluno
```

```
(ALU_CODIGO) ON DELETE CASCADE ON UPDATE CASCADE
```

```
)
```



Criação do Banco de Dados

Para adicionar uma coluna:

```
ALTER TABLE tbAluno  
ADD ALU_RUA varchar(100)
```

Para alterar uma coluna:

```
ALTER TABLE tbAluno  
ALTER COLUMN ALU_RUA varchar(150)
```

Para apagar uma coluna:

```
ALTER TABLE tbAluno  
DROP COLUMN ALU_RUA
```

Criação do Banco de Dados

Para apagar uma chave

```
ALTER TABLE tbAluno  
DROP CONSTRAINT UQ__tbAluno__08EA5793
```

Para adicionar uma chave única

```
ALTER TABLE tbAluno  
ADD CONSTRAINT UK_ALUNO_NOME UNIQUE (ALU_NOME)
```

Para adicionar uma chave estrangeira:

```
ALTER TABLE tbAlunoTelefone  
ADD CONSTRAINT FK_tbAluno_tbAlunoTelefone FOREIGN KEY  
(ALU_CODIGO) REFERENCES tbAluno (ALU_CODIGO)
```

Para apagar uma tabela utilize:

```
DROP TABLE tbAlunoTelefone
```

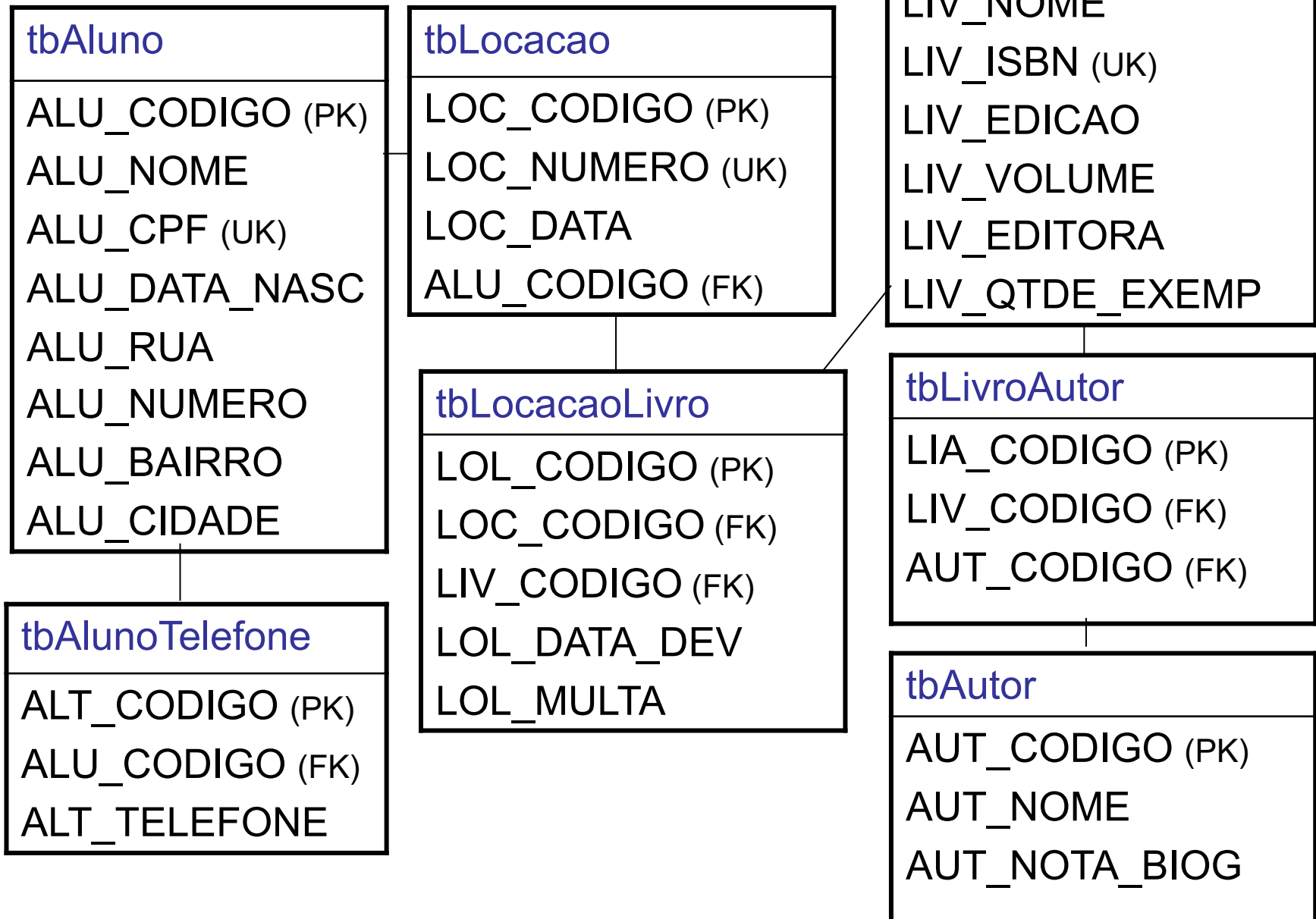
tbCurso

CUR_CODIGO
CUR_NOME
CUR_CARGA_HORARIA (UK)

tbDisciplina

DIS_CODIGO (PK)
DIS_NOME
DIS_SIGLA

- Adicionar os atributos DIS_CARGA_HORARIA e CUR_CODIGO na tabela tbDisciplina.
- Alterar o tipo de dados do atributo CUR_NOME para varchar(50).
- Remover o atributo DIS_SIGLA.
- Adicionar a chave primária ao atributo CUR_CODIGO da tabela tbCurso.
- Adicionar a chave estrangeira ao atributo CUR_CODIGO da tabela tbDisciplina.
- Adicionar a chave única ao atributo CUR_NOME.
- Remover a chave única do atributo CUR_CARGA_HORARIA.
- Remover a tabela tbDisciplina.



tbPecaOrdemServ

POS_CODIGO (PK)
POS_VALOR
POS_QTDE
PEC_CODIGO
ORD_CODIGO (FK)

tbOrdemServico

ORD_CODIGO (PK)
ORD_DATA
CAR_CODIGO (FK)

tbServicoOrdemServ

SOS_CODIGO (PK)
SOS_VALOR
SER_CODIGO (FK)
ORD_CODIGO (FK)

tbPeca

PEC_DESCRICAO
PEC_UNIDADE
PEC_VALOR

tbCarro

CAR_CODIGO (PK)
CAR_PLACA
CAR_MODELO (UK)
CAR_ANO
CAR_CHASSI
CAR_COR

tbServico

SER_CODIGO (PK)
SER_VALOR
SER_TEMPO

Criação do Banco de Dados

- 1) Adicionar o atributo SOS_QTDE a tabela tbServicoOrdemServ.
- 2) Remover o atributo SER_TEMPO da tabela tbServico.
- 3) Alterar o atributo SER_VALOR da tabela tbServico para decimal(10,2).
- 4) Adicionar o campo PEC_CODIGO na tabela tbPeca e adicionar a chave primária a este campo.
- 5) Adicionar a chave estrangeira ao atributo PEC_CODIGO da tabela tbPecaOrdemServ.
- 6) Adicionar as chaves únicas aos campos CAR_PLACA e CAR_CHASSI da tabela tbCarro.
- 7) Remover a chave única do atributo CAR_MODELO da tabela tbCarro.
- 8) (Tentar) remover o atributo SER_CODIGO da tabela tbServico.
- 9) (Tentar) remover a tabela tbServico.
- 10) Remover a tabela tbPecaOrdemServ.

Obs.: Justifique a execução dos itens 8 a 10, ressaltando as tentativas sem sucesso dos itens 8 e 9. Elabore maneiras que seria possível a execução desses itens.

tbPecaOrdemServ

POS_CODIGO (PK)
POS_VALOR
POS_QTDE
PEC_CODIGO (FK)
ORD_CODIGO (FK)

tbOrdemServico

ORD_CODIGO (PK)
ORD_DATA
CAR_CODIGO (FK)

tbServicoOrdemServ

SOS_CODIGO (PK)
SOS_VALOR
SOS_QTDE
SER_CODIGO (FK)
ORD_CODIGO (FK)

tbPeca

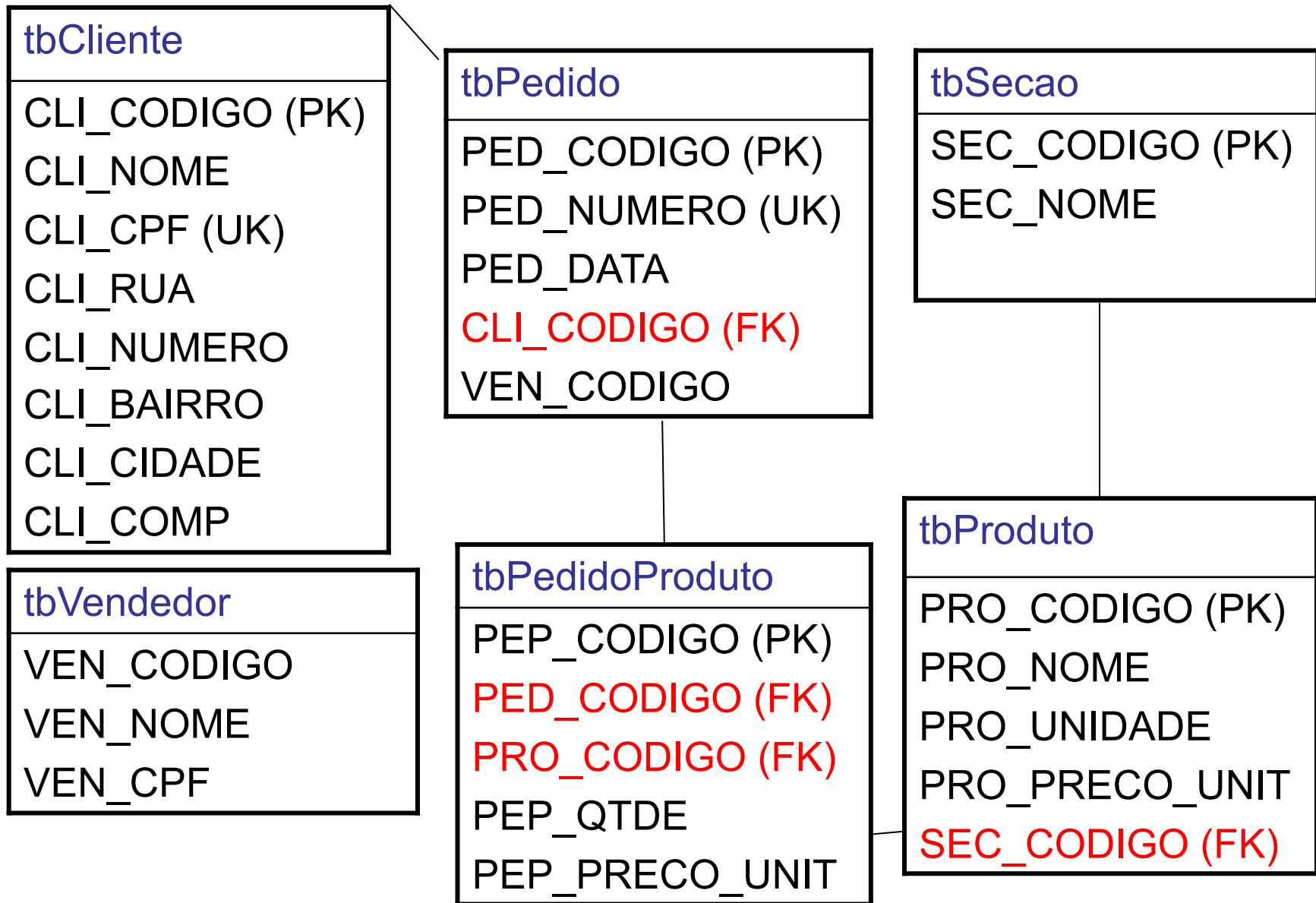
PEC_CODIGO (PK)
PEC_DESCRICAO
PEC_UNIDADE
PEC_VALOR

tbCarro

CAR_CODIGO (PK)
CAR_PLACA (UK)
CAR_MODELO
CAR_ANO
CAR_CHASSI (UK)
CAR_COR

tbServico

SER_CODIGO (PK)
SER_VALOR
SER_TEMPO



Criação do Banco de Dados

1. Crie o banco de dados através de scripts.
2. Adicionar o atributo PRO_QTDE na tabela tbProduto.
3. Remova o atributo CLI_COMP da tabela tbCliente.
4. Adicione a chave única ao atributo VEN_CPF da tabela tbVendedor.
5. Adicione a chave primária ao atributo VEN_CODIGO da tabela tbVendedor.
6. Adicione a chave estrangeira ao atributo VEN_CODIGO da tabela tbPedido.
7. Remova a chave única do atributo CLI_CPF da tabela tbCliente.
8. Remova a tabela tbSecao.

tbAluno

ALU_CODIGO (PK)
 ALU_NOME
 ALU_CPF (UK)
 ALU_DATA_NASC
 ALU_RUA
 ALU_NUMERO
 ALU_BAIRRO
 ALU_CIDADE

tbAlunoTelefone

ALT_CODIGO (PK)
 ALU_CODIGO (FK)
 ALT_TELEFONE

tbLocacao

LOC_CODIGO (PK)
 LOC_NUMERO (UK)
 LOC_DATA
 ALU_CODIGO (FK)

tbLocacaoLivro

LOL_CODIGO (PK)
 LOC_CODIGO (FK)
 LIV_CODIGO (FK)
 LOL_DATA_DEV
 LOL_MULTA

tbLivro

LIV_CODIGO (PK)
 LIV_NOME (UK)
 LIV_ISBN
 LIV_VOLUME
 LIV_EDITORA
 LIV_QTDE_EXEMP

tbLivroAutor

LIA_CODIGO (PK)
 LIV_CODIGO (FK)
 AUT_CODIGO

tbAutor

AUT_CODIGO
 AUT_NOME
 AUT_NOTA_BIOG

Criação do Banco de Dados

1. Crie o banco de dados através de scripts.
2. Adicionar o atributo LIV_EDICAO na tabela tbLivro.
3. Alterar o tipo de dados do atributo LOC_DATA da tabela tbLocacao para datetime.
4. Remover o atributo LIV_EDITORA.
5. Adicionar a chave primária ao atributo AUT_CODIGO da tabela tbAutor.
6. Adicionar a chave estrangeira ao atributo AUT_CODIGO da tabela tbLivroAutor.
7. Adicionar a chave única ao atributo LIV_ISBN da tabela tbLivro.
8. Remover a chave única de LIV_NOME na tabela tbLivro.
9. Remover a tabela tbAlunoTelefone
10. Adicionar o atributo ALU_TELEFONE na tabela tbAluno.