

Centro
Universitário
FMU

FMU

Professor Lima

Banco de Dados



e-Learning



Research



www.professorlima.com



Banco de Dados

Banco de Dados

2 – Modelo Entidade e Relacionamento (MER)

- Conceitos
- Diagramas.

01



3 – Modelo Entidade e Relacionamento (MER)

- Diagramas

02



4 – Modelo Relacional

Elementos Básicos

- Os sistemas de banco de dados relacionais foram desenvolvidos originalmente por causa da familiaridade e simplicidade. Como as tabelas são utilizadas para comunicar ideias em vários campos, a terminologia de tabelas, linhas e colunas é familiar para muitos usuários.

Tabelas

- Um banco de dados relacional consiste em uma coleção de tabelas. Cada tabela tem um cabeçalho ou uma parte de definição, e um corpo, ou uma parte de conteúdo. A parte do cabeçalho é composta do nome da tabela e nomes de colunas. Por exemplo, uma tabela de alunos pode ter colunas de número do CPF, nome, endereço, cidade, estado, CEP, classe (primeiro, segundo, terceiro ano etc.), especialização e média geral de notas. O corpo mostra as linhas da tabela. Cada linha de uma tabela de alunos representa um aluno matriculado em uma universidade. Uma tabela de alunos para a graduação da universidade pode ter mais de 30.000 linhas, linhas demais para serem visualizadas de uma só vez.

Tabelas

- Para entender uma tabela, também é importante ver algumas de suas linhas. Uma listagem de tabela, ou folha de dados (*datasheet*), mostra os nomes de coluna na primeira linha e o corpo nas demais linhas.

CPFAluno	NomeAluno	SobrenomeAluno	CidadeAluno	UFAluno	CEPAluno	Especializacao	Turma	MediaAluno
123-45-6789	HOMER	WELLS	SEATTLE	WA	98121-1111	SI	FR	3,00
124-56-7890	BOB	NORBERT	BOTHELL	WA	98011-2121	FINAN	JR	2,70
234-56-7890	CANDY	KENDALL	TACOMA	WA	99042-3321	CONTB	JR	3,50

Listagem de Amostra da Tabela Aluno

- Como os nomes de coluna muitas vezes são utilizados sem identificar as tabelas associadas, o prefixo permite a fácil associação à tabela. No caso de nomes combinados, são ressaltadas as diferentes partes do nome de uma coluna.

Tabelas

- Pode-se utilizar uma instrução CREATE TABLE para definir a parte do cabeçalho de uma tabela. CREATE TABLE é uma instrução em SQL. Como o SQL é uma linguagem padrão da indústria, a instrução CREATE TABLE pode ser utilizada na maioria dos SGBDs, para criar tabelas.
- Para cada coluna, são especificados o nome da coluna e o tipo de dado. Os tipos de dados indicam a espécie de dado (caractere, numérico, Sim/Não etc.) e as operações permitidas (operações numéricas, operações de cadeia de caracteres etc.) na coluna. Cada tipo de dado possui um nome (por exemplo, CHAR para caractere) e normalmente uma especificação de comprimento.

Tipo de Dados

- ***CHAR(L)*** Para entradas de texto de comprimento fixo, por exemplo, siglas de estado e número do CPF. Cada valor de coluna utilizando CHAR contém o número máximo de caracteres (*L*) mesmo que o comprimento real seja mais curto. A maioria dos SGBDs possui um limite superior de comprimento (*L*), por exemplo, 255.
- ***VARCHAR(L)*** Para texto de comprimento variável, por exemplo, nomes e endereços. Os valores de coluna utilizando VARCHAR contêm somente o número real de caracteres, e não o comprimento máximo das colunas CHAR. A maioria dos SGBDs possui um limite superior de comprimento, por exemplo, 255.

Tipo de Dados

- ***FLOAT(P)*** Para colunas contendo dados numéricos com precisão flutuante, por exemplo, cálculos de taxa de juros e cálculos científicos. O parâmetro de precisão *P* indica o número de dígitos significativos. A maioria dos SGBDs possui um limite superior de *P*, por exemplo, 38. Alguns SGBDs possuem dois tipos de dados, REAL e DOUBLE PRECISION, para números fracionários flutuantes de baixa e alta precisão em vez de precisão variável com o tipo de dado FLOAT.
- ***DATE/TIME*** Para colunas contendo data e hora, por exemplo, data do pedido. Esses tipos de dados não são padronizados entre os SGBDs. Alguns sistemas fornecem suporte a três tipos de dados (DATE, TIME e TIMESTAMP), enquanto outros fornecem suporte a um tipo de dado combinado (DATE) armazenando tanto a data como a hora.

Tipo de Dados

- ***DECIMAL(W, R)*** Para colunas contendo dados numéricos com precisão fixa, por exemplo, valores monetários. O valor *P* indica o número total de dígitos, e o valor *E* indica o número de dígitos à direita da vírgula decimal. Em alguns sistemas, esse tipo de dado também é denominado NUMERIC.
- ***INTEGER*** Para colunas contendo números inteiros (ou seja, números sem vírgula decimal). Alguns SGBDs possuem o tipo de dado SMALLINT para números inteiros bem pequenos e o tipo de dado LONG para números inteiros bem grandes.
- ***BOOLEAN*** Para colunas contendo dados com dois valores, por exemplo, verdadeiro/falso ou sim/não.

SQL

```
CREATE TABLE Aluno (  
    CPFAluno CHAR (11),  
    NomeAluno VARCHAR (50),  
    SobrenomeAluno VARCHAR (50),  
    CidadeAluno VARCHAR(50),  
    UFAluno CHAR(2),  
    CEPAluno CHAR(10),  
    Especializacao CHAR(6),  
    Turma CHAR(6),  
    MediaAluno DECIMAL(3,2)  
)
```

Ligações Entre Tabelas

- Não basta entender cada tabela individualmente. Para entender um banco de dados relacional, é necessário entender também as ligações, ou seja, os relacionamentos existentes entre as tabelas.
- As linhas de uma tabela normalmente estão relacionadas a linhas de outras tabelas. A correspondência entre valores (idênticos) indica os relacionamentos entre as tabelas.
- Considerar a amostra da tabela *Matricula* em que cada linha representa um aluno matriculado em um oferecimento de um curso. Os valores contidos na coluna *CPFAluno* da tabela *Matricula* correspondem aos valores de *CPFAluno* na amostra da tabela *Aluno*.
- Por exemplo, a primeira e a terceira linha da tabela *Matricula* possuem o valor de *CPFAluno* igual à primeira linha da tabela *Aluno*.

Entendendo Bancos de Dados Relacionais

Tabela Matricula

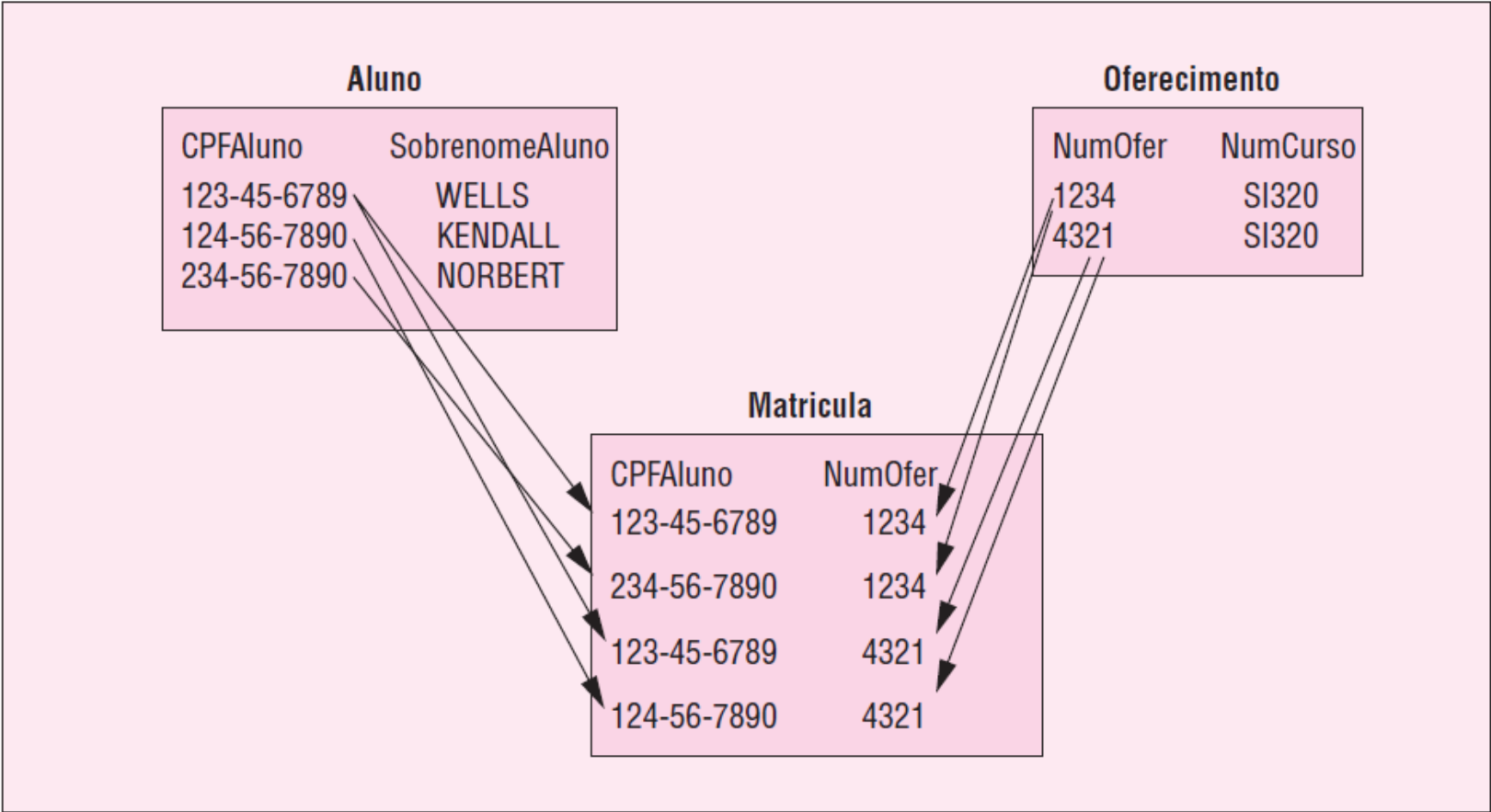
NumOfer	CPFAluno	NotaMatr
1234	123-45-6789	3,3
1234	234-56-7890	3,5
4321	123-45-6789	3,5
4321	124-56-7890	3,2

Tabela Oferecimento

NumOfer	NumCurso	TrimestreOfer	AnoOfer	LocalOfer	HorarioOfer	CPFProf	DiasSemanaOfer
1111	SI320	VERAO	2006	BLM302	10h30		SEG-QUA
1234	SI320	OUTONO	2005	BLM302	10h30	098-76-5432	SEG-QUA
2222	SI460	VERAO	2005	BLM412	13h30		TER-QUI
3333	SI320	PRIMAVERA	2006	BLM214	8h30	098-76-5432	SEG-QUA
4321	SI320	OUTONO	2005	BLM214	15h30	098-76-5432	TER-QUI
4444	SI320	PRIMAVERA	2006	BLM302	15h30	543-21-0987	TER-QUI
5678	SI480	PRIMAVERA	2006	BLM302	10h30	987-65-4321	SEG-QUA
5679	SI480	PRIMAVERA	2006	BLM412	15h30	876-54-3210	TER-QUI
9876	SI460	PRIMAVERA	2006	BLM307	13h30	654-32-1098	TER-QUI

Correspondência de Valores entre as Tabelas Matricula, Oferecimento e Aluno

- Do mesmo modo, os valores contidos na coluna *NumOfer* da tabela *Matricula* coincidem com o valor da coluna *NumOfer* da tabela *Oferecimento*. A Figura mostra a reprodução gráfica dos valores correspondentes.



Definição das Regras de Integridade

- **Integridade de entidade** significa que cada tabela deve ter uma coluna ou uma combinação de colunas com valores únicos. O termo “único” significa que não existem valores iguais em duas linhas de uma tabela. Por exemplo, o valor de *CPFAluno* na tabela *Aluno* é único, e a combinação entre *CPFAluno* e *NumOfer* é única na tabela *Matricula*.
- A integridade de entidade assegura a identificação exclusiva das entidades (pessoas, objetos e eventos) em um banco de dados. Para fins de auditoria, segurança e comunicação, é importante que as entidades de negócios sejam facilmente rastreáveis.

Definição das Regras de Integridade

- **Integridade referencial** significa que os valores da coluna de uma tabela devem corresponder aos valores da coluna em uma tabela relacionada. Por exemplo, o valor de *CPFAluno* em cada linha da tabela *Matricula* deve corresponder ao valor de *CPFAluno* em alguma linha da tabela *Aluno*.
- A integridade referencial assegura a existência de ligações válidas dentro de um banco de dados. Por exemplo, é fundamental a existência de número do CPF de um aluno válido em cada linha da tabela *Matricula*.
- Do contrário, algumas matrículas podem não ter sentido, provocando possivelmente a rejeição da matrícula de alguns alunos por causa da matrícula de alunos inexistentes em seu lugar. Para definir com mais precisão a integridade de entidade e a integridade referencial, são necessárias algumas outras definições. Seguem as definições desses pré-requisitos e definições mais precisas.

Definições

- **Superchave:** uma coluna ou uma combinação de colunas contendo valores únicos para cada linha. A combinação de todas as colunas em uma tabela sempre é uma superchave, porque as linhas de uma tabela devem ser únicas.
- **Chave candidata:** uma superchave mínima. A superchave é mínima se, com a remoção de qualquer coluna, ela deixa de ser única.
- **Valor nulo:** um valor especial que representa a ausência de um valor real. O valor nulo pode significar que o valor real é desconhecido ou não se aplica a determinada linha.
- **Chave primária:** chave candidata especialmente designada. A chave primária de uma tabela não pode conter valores nulos.
- **Chave estrangeira:** uma coluna ou uma combinação de colunas em que os valores devem corresponder aos valores de uma chave candidata. A chave estrangeira deve ter tipo de dado igual ao da chave candidata associada. Na instrução CREATE TABLE do SQL, a chave estrangeira deve ser associada a uma chave primária e não a uma simples chave candidata.

Regras de Integridade

- **Regra de integridade de entidade:** duas linhas de uma tabela não podem conter o mesmo valor de chave primária. Além disso, nenhuma linha pode conter um valor nulo para qualquer coluna componente de uma chave primária.
- **Regra de integridade referencial:** somente dois tipos de valores podem ser armazenados em uma chave estrangeira: um valor correspondente ao valor de uma chave candidata em alguma linha da tabela contendo a chave candidata associada ou um valor nulo

Aplicação das Regras de Integridade

- Para ampliar seu entendimento sobre esse tópico, as regras de integridade serão aplicadas a diversas tabelas do banco de dados de uma universidade. A chave primária de *Aluno* é *CPFAluno*.
- A chave primária pode ser designada como parte da instrução CREATE TABLE. Para designar *CPFAluno* como chave primária de *Aluno*, utiliza-se uma cláusula CONSTRAINT para a chave primária no final da instrução CREATE TABLE.
- O nome de restrição (PKAluno) depois da palavra-chave CONSTRAINT facilita a identificação da restrição se ocorre uma violação quando uma linha é inserida ou atualizada.

SQL

```
CREATE TABLE Aluno (  
    CPFAluno CHAR(11),  
    NomeAluno VARCHAR(50),  
    SobrenomeAluno VARCHAR(50),  
    CidadeAluno VARCHAR(50),  
    UFAluno CHAR(2),  
    CEPAluno CHAR(10),  
    Especializacao CHAR(6),  
    Turma CHAR(2),  
    MediaAluno DECIMAL(3,2),  
  
    CONSTRAINT PKAluno PRIMARY KEY (CPFAluno)  
)
```

Aplicação das Regras de Integridade

- Os números do CPF são atribuídos pelo governo federal, portanto, a universidade não precisa atribuí-los. Em outros casos, os valores primários são atribuídos pela organização. Por exemplo, número de cliente, número de produto e número de funcionário são normalmente atribuídos pela organização controlando o banco de dados subjacente. **Nesses casos, é necessária a geração automática de valores únicos. Alguns SGBDs fornecem suporte à geração automática de valores únicos.**

Variações da Integridade de Entidade

- As chaves candidatas que não são chaves primárias são declaradas com a palavra-chave UNIQUE. A tabela *Curso* contém duas chaves candidatas: *NumCurso* (chave primária) e *DescrCurso* (descrição do curso). A coluna *NumCurso* é a chave primária porque é mais estável que a coluna *DescrCurso*. As descrições de curso podem mudar com o tempo, mas os números de curso permanecem os mesmos.

NumCurso	DescrCurso	CargaHoraCurso
SI320	FUNDAMENTOS DE NEGOCIOS	4
SI460	ANALISE DE SISTEMAS	4
SI470	COMUNICACOES DE DADOS	4
SI480	FUNDAMENTOS DE GERENCIAMENTO DE BANCO DE DADOS	4

SQL

```
CREATE TABLE Curso (  
    NumCurso CHAR(6),  
    DescrCurso VARCHAR(250),  
    CargaHoraCurso SMALLINT,  
  
    CONSTRAINT PKCurso PRIMARY KEY(NumCurso),  
    CONSTRAINT DescrCursoUnico UNIQUE (DescrCurso)  
)
```


Variações da Integridade de Entidade

- Algumas tabelas necessitam de mais de uma coluna na chave primária. Na tabela *Matricula*, a combinação entre *CPFAluno* e *NumOfer* é a única chave candidata. Ambas as colunas são necessárias para identificar uma linha. A chave primária constituída de mais de uma coluna é conhecida como chave primária composta ou combinada.

```
CREATE TABLE Matricula (  
    NumOfer INTEGER,  
    CPFAluno CHAR(11),  
    NotaMatr DECIMAL(3,2),  
    CONSTRAINT PKMatricula PRIMARY KEY(NumOfer, CPFAluno)  
)
```

Variações da Integridade de Entidade

- As superchaves não mínimas não são importantes porque são comuns e contêm colunas que não contribuem para a propriedade da unicidade. Por exemplo, a combinação de *CPFAluno* e *SobrenomeAluno* é única. No entanto, se *SobrenomeAluno* for removido, *CPFAluno* ainda permanece único.

Integridade Referencial

- No caso da integridade referencial, as colunas *CPFAluno* e *NumOfer* são chaves estrangeiras na tabela *Matricula*. A coluna *CPFAluno* refere-se à tabela *Aluno*, e a coluna *NumOfer* refere-se à tabela *Oferecimento*. Uma linha de *Oferecimento* representa um curso ministrado em um período acadêmico (verão, inverno etc.), um ano, um horário, um local e dias da semana. A chave primária de *Oferecimento* é *NumOfer*. Um curso, por exemplo, SI480, terá número de oferecimento diferente cada vez que for oferecido.
- O método de definição de restrições de integridade referencial é semelhante ao de definição de chaves primárias. Por exemplo, para definir as chaves estrangeiras em *Matricula*, deve-se utilizar cláusulas `CONSTRAINT` para chaves estrangeiras no final da instrução `CREATE TABLE`, assim como mostra a instrução revisada `CREATE TABLE` para a tabela *Matricula*.

SQL

```
CREATE TABLE Matricula (  
    NumOfer INTEGER,  
    CPFAluno CHAR(11),  
    NotaMatr DECIMAL(3,2),  
  
    CONSTRAINT PKMatricula PRIMARY KEY(NumOfer, CPFAluno),  
    CONSTRAINT FKNumOfer FOREIGN KEY (NumOfer) REFERENCES Oferecimento,  
    CONSTRAINT FKCPFAluno FOREIGN KEY (CPFAluno) REFERENCES Aluno  
)
```

Integridade Referencial

- Embora a integridade referencial permita a existência de chaves estrangeiras com valores nulos, não é comum chaves estrangeiras terem valores nulos. Quando a chave estrangeira é parte de uma chave primária, não são permitidos valores nulos por causa da regra de integridade de entidade. Por exemplo, não são permitidos valores nulos nem para *Matricula.CPFAluno* nem *Matricula.NumOfer* porque cada coluna é parte da chave primária.

Integridade Referencial

- Embora a integridade referencial permita a existência de chaves estrangeiras com valores nulos, não é comum chaves estrangeiras terem valores nulos. Quando a chave estrangeira é parte de uma chave primária, não são permitidos valores nulos por causa da regra de integridade de entidade. Por exemplo, não são permitidos valores nulos nem para *Matricula.CPFAluno* nem *Matricula.NumOfer* porque cada coluna é parte da chave primária.
- Quando a chave estrangeira não é parte de uma chave primária, o uso determina se são permitidos valores nulos. Por exemplo, *Oferecimento.NumCurso*, chave estrangeira referente a *Curso*, não é parte de uma chave primária, portanto, não são permitidos valores nulos. Na maioria das universidades, um curso não pode ser oferecido antes de ele ser aprovado. Portanto, um oferecimento não deve ser inserido sem um curso relacionado.

Integridade Referencial

- As palavras-chave NOT NULL indicam que uma coluna não pode ter valores nulos, assim como mostra a instrução CREATE TABLE para a tabela *Oferecimento*. As restrições NOT NULL são restrições em linha associadas a uma coluna específica. Em contrapartida, as restrições de chave primária e estrangeira em uma instrução CREATE TABLE para a tabela *Oferecimento* são restrições de tabela em que as colunas associadas devem ser especificadas na restrição. Os nomes de restrição devem ser utilizados tanto com restrições de tabela como com restrições em linha para facilitar a identificação quando ocorre uma violação.

```
CREATE TABLE Oferecimento (  
    NumOfer INTEGER,  
    NumCurso CHAR(6) CONSTRAINT NumCursoOferRequerido NOT NULL,  
    LocalOfer VARCHAR(50),  
    DiasSemanaOfer CHAR(6),  
    TrimestreOfer CHAR(6) CONSTRAINT TrimOferRequerido NOT NULL,  
    AnoOfer INTEGER CONSTRAINT AnoOferRequerido NOT NULL,  
    CPFProf CHAR(11),  
    HorarioOfer DATE,  
    CONSTRAINT PKOferecimento PRIMARY KEY (NumOfer),  
    CONSTRAINT FKNumCurso FOREIGN KEY(NumCurso) REFERENCES Curso,  
    CONSTRAINT FKCPFProf FOREIGN KEY(CPFProf) REFERENCES Professor  
)
```


Entendendo Bancos de Dados Relacionais

- Em contrapartida, *Oferecimento.CPFProf* referente ao professor do oferecimento pode ser nulo. A tabela *Professor* armazena dados sobre os instrutores dos cursos. Um valor nulo em *Oferecimento.CPFProf* significa que ainda não há professor designado para lecionar o oferecimento. Por exemplo, não há instrutor designado na primeira e terceira linhas. Como os oferecimentos geralmente são programados talvez com um ano de antecedência, possivelmente, os instrutores de alguns oferecimentos são conhecidos somente depois de armazenada inicialmente a linha de oferecimento. Assim, a permissão de valores nulos na tabela *Oferecimento* é prudente.

Integridade Referencial para Auto-Relacionamentos (Unários)

- A restrição de integridade referencial envolvendo uma única tabela é conhecida como autorelacionamento ou relacionamento unário. Os auto-relacionamentos não são comuns, mas são importantes quando ocorrem. No banco de dados de uma universidade, um professor pode supervisionar outros professores e ser supervisionado por um professor.

Integridade Referencial para Auto-Relacionamentos (Unários)

- Por exemplo, Victoria Emmanuel (segunda linha) supervisiona Leonard Fibon (terceira linha). A coluna *SupervisorProf* mostra este relacionamento: o valor de *SupervisorProf* na terceira linha (543- 21-0987) corresponde ao valor de *CPFProf* na segunda linha. A restrição de integridade referencial envolvendo a coluna *SupervisorProf* representa o auto-relacionamento. Na instrução CREATE TABLE, a restrição de integridade referencial para um auto-relacionamento pode ser escrita da mesma forma que outras restrições de integridade referencial.

CPFProf	NomeProf	SobrenomeProf	CidadeProf	UFProf	DeptoProf	ClassificacaoProf	SalarioProf	SupervisorProf	DataAdmProf	CEPProf
098-76-5432	LEONARD	VINCE	SEATTLE	WA	ADM	ASSISTENTE	R\$35.000	654-32-1098	01-Abril-95	98111-9921
543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA	ADM	CATEDRATICO	R\$120.000		01-Abril-96	98011-2242
654-32-1098	LEONARD	FIBON	SEATTLE	WA	ADM	ASSOCIADO	R\$70.000	543-21-0987	01-Abril-95	98121-0094
765-43-2109	NICKI	MACON	BELLEVUE	WA	FINAN	CATEDRATICO	R\$65.000		01-Abril-97	98015-9945
876-54-3210	CRISTOPHER	COLAN	SEATTLE	WA	ADM	ASSISTENTE	R\$40.000	654-32-1098	01-Abril-99	98114-1332
987-65-4321	JULIA	MILLS	SEATTLE	WA	FINAN	ASSOCIADO	R\$75.000	765-43-2109	01-Abril-00	98114-9954

SQL

```
CREATE TABLE Professor (  
    CPFProf CHAR(11),  
    NomeProf VARCHAR(50) CONSTRAINT NomeProfRequerido NOT NULL,  
    SobrenomeProf VARCHAR(50) CONSTRAINT SobrenomeProfRequerido NOT NULL,  
    CidadeProf VARCHAR(50) CONSTRAINT CidadeProfRequerido NOT NULL,  
    UFProf CHAR(2) CONSTRAINT UFProfRequerido NOT NULL,  
    CEPProf CHAR(10) CONSTRAINT CEPProfRequerido NOT NULL,  
    DataAdmProf DATE,  
    DeptoProf CHAR(6),  
    ClassificacaoProf CHAR(4),  
    SalarioProf DECIMAL(10,2),  
    SupervisorProf CHAR(11),  
    CONSTRAINT PKProfessor PRIMARY KEY (CPFProf),  
    CONSTRAINT FKSupervisorProf FOREIGN KEY (SupervisorProf) REFERENCES Professor  
)
```

“Seja o que a sua mente pode conceber e acreditar, ela pode conseguir.”

Napoleon Hill



Obrigado!

