

# **Banco de Dados Relacionais e não Relacionais**

Prof. Henrique Batista da Silva

# Introdução à Banco de Dados

# Armazenamento dos dados

Por muito tempo banco de dados Relacionais tem sido o principal meio de armazenamento de dados.

O modelo de dados relacional foi introduzido por Edgar Codd em 1970 (IBM).

# Armazenamento dos dados

A ideia de modelo relacional era representar **entidade e relacionamento** de maneira uniforme.

Os SGBDs Relacionais mais conhecidos hoje são MySQL (Oracle), Oracle (Oracle) e SQL Server (Microsoft).

# Exemplo de um modelo Relacional

Modelo de Relacional (organizado em tuplas, normalizado, e possui integridade referencial)

Tabela: Cliente	
Id	Nome
1	Marcos

Tabela: Pedido		
Id	IdCliente	IdEndEntrega
1	1	1

Tabela: Produto	
Id	Nome
10	Laptop

Tabela: ItemPedido			
Id	IdPedido	IdProduto	Preço
1	2	10	350,00

Tabela: Endereço				
Id	Logradouro	Cidade	Estado	CEP
1	Av. Sen. Salgado Filho	Natal	RN	59.056-000

# Alguns desafios para persistências de dados

**A**tomicidade (a transação é executada totalmente ou **não** é executada), **C**onsistência (sistema sempre consistente após uma operação), **I**solamento (transação não sofre interferência de outra transação concorrente), e **D**urabilidade (o que foi salvo não é mais perdido)

- Força a consistência ao final de cada transação

# Modelo Relacional



# Modelo Relacional

A ideia de modelo relacional era representar ambos, **entidade e relacionamento**, de maneira uniforme.

O modelo de dados relacional é uma linguagem unificada para definição e manipulação de dados.



# Modelo Relacional

Quando uma relação é pensada como uma tabela, cada linha na tabela representa uma coleção de valores de dados relacionados.

No modelo relacional, cada linha na tabela corresponde a uma entidade ou relacionamento do mundo real.

# Modelo Relacional

As colunas no modelo relacional (atributos de uma entidade), especificam como interpretar os valores de dados em cada linha.

Todos os valores de uma coluna são do mesmo tipo de dados.

# Modelo Relacional

## Terminologia:

- Uma linha é chamada ***Tupla***.
- Um cabeçalho de coluna é chamado ***Atributo***.
- A tabela é chamada ***Relação***.
- O conjunto de valores que um atributo pode ter se chama ***Domínio***.

# Modelo Relacional

Algumas relações(tabelas) podem representar fatos sobre **entidades**, enquanto outras podem representar fatos sobre **relacionamentos**.

# Modelo Relacional

## Os atributos e as tuplas de uma relação ALUNO

The diagram illustrates the components of a relational table. The label 'Nome da relação' points to the first column header 'ALUNO'. The label 'Atributos' points to the column headers 'Nome', 'SSN', 'FoneResidencia', 'Endereco', 'FoneEscritorio', 'Idade', and 'MPG'. The label 'Tuplas' points to the rows of data in the table.

ALUNO	Nome	SSN	FoneResidencia	Endereco	FoneEscritorio	Idade	MPG
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	<i>null</i>	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	<i>null</i>	18	2.89
	Dick Davidson	422-11-2320	<i>null</i>	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	<i>null</i>	19	3.25

# Modelo Relacional

## Os atributos e as tuplas de uma relação ALUNO

Cada valor em uma tupla é atômico: não são permitidos atributos multivalorados ou compostos

ALUNO	Nome	SSN	FoneResidencia	Endereco	FoneEscritorio	Idade	MPG
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	<i>null</i>	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	<i>null</i>	18	2.89
	Dick Davidson	422-11-2320	<i>null</i>	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	<i>null</i>	19	3.25

Os atributos multivalorados devem ser representados em relações separadas

# Modelo Relacional

## Os atributos e as tuplas de uma relação ALUNO

**Restrições de domínio:** O valor de uma coluna deve obedecer a definição dos tipos de dados admitidos para esta coluna

ALUNO	Nome	SSN	FoneResidencia	Endereco	FoneEscritorio	Idade	MPG
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	<i>null</i>	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	<i>null</i>	18	2.89
	Dick Davidson	422-11-2320	<i>null</i>	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	<i>null</i>	19	3.25

Assim, é também especificado se a coluna pode ou não ter valores nulos



# Modelo Relacional

## Os atributos e as tuplas de uma relação ALUNO

**Restrições de chave:** Relação é um conjunto de tuplas. Por definição, todos os elementos de um conjunto são distintos.

ALUNO	Nome	SSN	FoneResidencia	Endereco	FoneEscritorio	Idade	MPG
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Assim, duas tuplas não podem ter a mesma combinação de valores para todos os seus atributos.



# Restrições do Modelo Relacional

A faint, light-colored illustration in the background of the teal slide. It features several interlocking gears of different sizes, a lightbulb with a cross inside, and a circular component with a serrated edge, possibly a flywheel or a part of a machine. The illustration is rendered in a simple, line-art style.

# Restrições do Modelo Relacional

Uma **chave** é um conjunto mínimo de valores dos atributos que identifica unicamente uma tupla (linha).

Garante a **restrição de unicidade** entre as tuplas de uma relação

Exemplo:

**Matrícula** é uma chave de ALUNO pois dois ou mais alunos não podem ter a mesma matrícula.

Não podemos indicar o “nome do aluno” como chave pois é possível existirem dois alunos com o mesmo nome.

# Restrições do Modelo Relacional

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	João	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	NULL

# Restrições do Modelo Relacional

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	João	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	NULL

Existem dois empregados com o mesmo nome, porém não há problemas, desde que o atributo **Nome** não seja o **Atributo Chave** da relação (e nem contenha a restrição Unique).

# Restrições do Modelo Relacional

**CodEmp** é o **Atributo Chave** da relação. Note que seus valores não se repetem. O atributo **CodEmp** garante a unicidade entre as tuplas da relação.

Empregado					
<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	João	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	NULL

Existem dois empregados com o mesmo nome, porém não há problemas, desde que o atributo **Nome** não seja o **Atributo Chave** da relação (e nem contenha a restrição Unique).

# Restrições do Modelo Relacional

Restrições de chave (Chave Primária):

É uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela.

Podem ser Chaves únicas ou Chaves compostas.

# Restrições do Modelo Relacional

Chave  
Única

CódigoEmp	Nome	CódigoDepto
E3	Paulo	D1
E1	Luciana	D2
E2	Antonio	D1
E4	Lucia	D3

Chave  
Composta

CódigoEmp	NumDepend	Nome	Tipo
E1	01	Luis	Filho
E1	02	Marta	Esposa
E2	01	Ana	Esposa
E2	02	Carlos	Filho

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições de Integridade Referencial





# Restrições do Modelo Relacional

Diagrama para  
o esquema do  
banco de dados  
relacional  
EMPRESA

**EMPREGADO**

PNome	Minicial	UNome	<u>SSN</u>	Datanasc	Endereco	Sexo	Salario	SuperSSN	DNO
-------	----------	-------	------------	----------	----------	------	---------	----------	-----

**DEPARTAMENTO**

DNome	<u>DNumero</u>	GerSSN	GerDataInicio
-------	----------------	--------	---------------

**DEPTO\_LOCALIZACOES**

<u>DNumero</u>	<u>DLocalizacao</u>
----------------	---------------------

**PROJETO**

PJNome	<u>PNumero</u>	PLocalizacao	DNum
--------	----------------	--------------	------

**TRABALHA\_EM**

<u>ESSN</u>	<u>PNO</u>	HORAS
-------------	------------	-------

**DEPENDENTE**

<u>ESSN</u>	<u>NOME_DEPENDENTE</u>	SEXO	DATANASC	PARENTESCO
-------------	------------------------	------	----------	------------

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições do Modelo Relacional

Restrições de Integridade Entidade:

A chave primária de uma relação **NÃO** pode ter valor **NULO**.

Se isso fosse permitido, então estaríamos admitindo que existam tuplas que não se diferenciam, violando a regra básica da chave primária.

# Restrições do Modelo Relacional

Restrições de Integridade Referencial:

É classificada entre duas relações e usada para manter consistência entre as tuplas das duas relações.

Informalmente:

Uma tupla em uma relação, que faz referência a outra relação, **deve-se referir a uma tupla existente nesta relação.**

# Restrições do Modelo Relacional

## Exemplo (Restrição Integridade Referencial):

O valor do atributo DNO (faz referencia ao departamento em que o empregado trabalha) de EMPREGADO deve sempre corresponder ao valor de DNUMERO em alguma tupla de Departamento

EMPREGADO	PNOME	MINICIAL	UNOME	SSN	DATANASC	ENDERECO	SEXO	SALARIO	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPARTAMENTO	DNO	DNUMERO	GERSSN	GERDATAINICIO
Pesquisa	5	333445555	1988-05-22	
Administração	4	987654321	1995-01-01	
Sede administrativa	1	888665555	1981-06-19	

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições do Modelo Relacional

Formalmente (Restrição Integridade Referencial):

Usa-se o conceito de **chave estrangeira (foreign key)** para definir as restrições de integridade referencial.

A chave estrangeira (FK) faz referência a uma relação se satisfazer as duas regras:

Os atributos de FK de  $R_1$  têm o mesmo domínio da PK de  $R_2$ .

Um valor de FK de uma tupla  $t_1$  deve ser igual a um valor de PK para uma tupla  $t_2$ , ou ser null.

# Restrições do Modelo Relacional

*A integridade referencial estabelece que todo valor de chave estrangeira numa relação deve corresponder a um valor de chave primária de uma segunda relação ou deve ser nulo.*

# Restrições do Modelo Relacional

## Restrição Integridade Referencial

Chave Primária (PK)

CodigoDepto	Nome
D1	Compras
D2	Vendas
D3	Engenharia

R2 - DEPARTAMENTO

Chave Estrangeira (FK)

CódigoEmp	Nome	CódigoDepto
E3	Paulo	D1
E1	Luciana	D2
E2	Antonio	D1
E4	Lucia	NULL

R1 - EMPREGADO

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições do Modelo Relacional

## Restrição Integridade Referencial (Exemplo)

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDeppto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	José	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	4

**Departamento**

<u>CodDeppto</u>	Nome
1	Matemática
2	Física
3	Computação

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.



# Restrições do Modelo Relacional

## Restrição Integridade Referencial (Exemplo)

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	José	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	4

**Departamento**

<u>CodDepto</u>	Nome
1	Matemática
2	Física
3	Computação

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições do Modelo Relacional

## Restrição Integridade Referencial (Exemplo)

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	José	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	4

**Departamento**

<u>CodDepto</u>	Nome
1	Matemática
2	Física
3	Computação

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições do Modelo Relacional

## Restrição Integridade Referencial (Exemplo)

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepdo
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	José	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	4

**Departamento**

<u>CodDepdo</u>	Nome
1	Matemática
2	Física
3	Computação

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Restrições do Modelo Relacional

## Restrição Integridade Referencial (Exemplo)

**Empregado**

<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepdo
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	José	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	4

**Departamento**

<u>CodDepdo</u>	Nome
1	Matemática
2	Física
3	Computação

Empregado 8, não faz referência a nenhum departamento.

# Restrições do Modelo Relacional

## Restrição Integridade Referencial (Exemplo)

**Empregado**

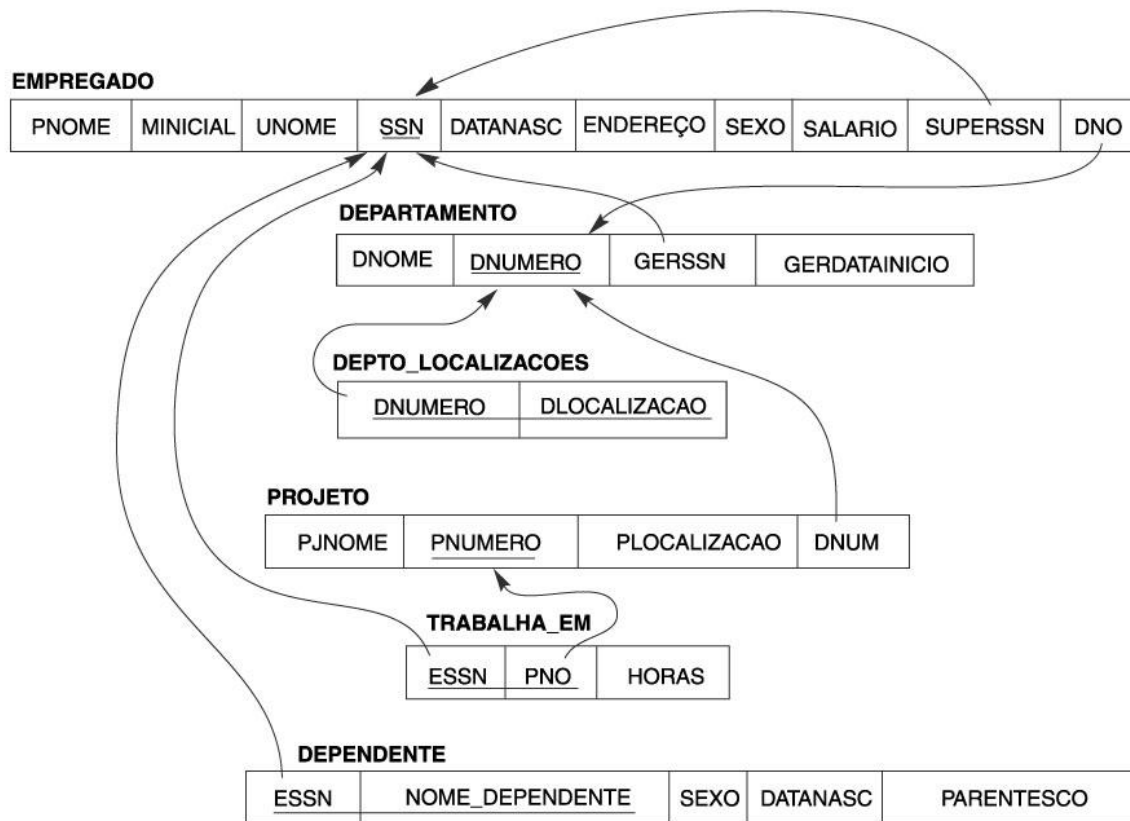
<u>CodEmp</u>	Nome	DataNasc	Cidade	Estado	CodDepto
1	José	21/04/1980	BH	MG	1
2	Alberto	22/04/1980	BH	MG	1
3	Maria	05/08/1970	BH	MG	3
4	Ana	24/04/1980	BH	MG	2
5	Pedro	05/08/1970	BH	MG	1
6	Antônio	06/08/1970	BH	MG	2
7	Maria	07/08/1970	BH	MG	3
8	José	28/04/1980	BH	MG	NULL
9	Carlos	29/04/1980	BH	MG	4

**Departamento**

<u>CodDepto</u>	Nome
1	Matemática
2	Física
3	Computação

Não existe o departamento de código igual a 4. Portanto, **valor não Permitido.**

# Restrições do Modelo Relacional



Restrições de integridade referencial exibidas no esquema de um banco de dados relacional EMPRESA

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Operações de Atualização sobre Relações

# Operações de Atualização sobre Relações

As operações de atualização precisam ser aplicadas sem violar a integridade do banco de dados.

Há três operações básicas.

- Inserir (INSERT) – Usada para inserir uma nova tupla.
- Alterar (UPDATE) – usada para modificar os valores dos atributos em tuplas existentes.
- Excluir (DELETE) – usada para remover uma tupla.



# Operações de Atualização sobre Relações

**INSERT** fornece uma lista de atributos para uma nova tupla  $t$  que está sendo inserida em uma relação  $R$ .

Inserir pode violar qualquer um dos tipos de restrição:

- Domínio

- Chave

- Entidade

- Integridade referencial

# Operações de Atualização sobre Relações

Violação de **Domínio** (insert): violada se algum atributo não estiver contido no domínio definido:

INSERT →

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDepto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
3	Maria	F	BH	5	3

O valor a ser inserido deve ser conter apenas duas letras e não pode ser valor numérico.

# Operações de Atualização sobre Relações

Violação de **Domínio** (insert): violada se algum atributo não estiver contido no domínio definido:

INSERT →

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDepto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
2	Maria	F	BH	MG	3

O valor a ser inserido no atributo chave deve ser único, não podendo haver valores repetidos para tuplas diferentes.

# Operações de Atualização sobre Relações

Violação de **Domínio** (insert): violada se algum atributo não estiver contido no domínio definido:

INSERT →

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDepto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
NULL	Maria	F	BH	MG	3

O valor a ser inserido no atributo chave não pode ser um valor nulo.

# Operações de Atualização sobre Relações

Violação de **Domínio** (insert): violada se algum atributo não estiver contido no domínio definido:

INSERT →

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDepto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
3	Maria	F	BH	MG	4

Departamento

<u>CodDepto</u>	Nome
1	Matemática
2	Física
3	Computação

→ O valor a ser inserido deve ser um valor válido (existente) na relação que está sendo referenciada.

# Operações de Atualização sobre Relações

A operação **DELETE** é usada para excluir tuplas.

A única restrição de integridade que a operação **DELETE** pode violar é a **integridade referencial**:  
Isso ocorre quando a tupla que está sendo excluída é referenciada por chaves estrangeiras em outras tuplas no BD.

# Operações de Atualização sobre Relações

Violação de **Integridade referencial** (delete):

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDepto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
3	Maria	F	BH	MG	3

Departamento

<u>CodDepto</u>	Nome
1	Matematica
2	Física
3	Computação

DELETE →

A tupla a ser apagada  
está sendo referenciada  
por outra tupla em uma  
outra relação.

# Operações de Atualização sobre Relações

Existem três comportamentos possíveis quando uma operação de exclusão causa uma violação da integridade:

Bloqueio (restrict): quando a tupla a ser excluída é referenciada em outras relações (pela FK).

Propagação (cascade): todas as tuplas que referenciam a tupla a ser excluída (pela FK) são excluídas também automaticamente.



# Operações de Atualização sobre Relações

Substituição por nulo (set null): todas as tuplas que referenciam a tupla a ser excluída (pela FK) têm os valores dos atributos da chave estrangeira modificados para nulo (se for permitido nulo) e a exclusão é efetuada.

# Operações de Atualização sobre Relações

## Propagação (cascade):

Empregado

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDeppto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
3	Maria	F	BH	MG	3

Departamento

<u>CodDeppto</u>	Nome
1	Matematica
2	Física
3	Computação

DELETE →

←

# Operações de Atualização sobre Relações

## Propagação (cascade):

Empregado

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDepto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
3	Maria	F	BH	MG	3

Departamento

<u>CodDepto</u>	Nome
1	Matematica
2	Física

DELETE →



# Operações de Atualização sobre Relações

## Propagação (SET NULL):

Empregado

<u>CodEmp</u>	Nome	Sexo	Cidade	Estado	CodDeppto
1	José	M	BH	MG	1
2	Alberto	M	BH	MG	2
3	Maria	F	BH	MG	NULL

SET NULL →

Departamento

<u>CodDeppto</u>	Nome
1	Matematica
2	Física

DELETE →

# Operações de Atualização sobre Relações

A operação de atualização (**UPDATE**) muda os valores de alguns atributos em tuplas existentes.

É necessário especificar uma condição para indicar que tuplas devem ser alteradas.

Exemplo: altere o endereço do empregado, cujo o valor de sua matrícula = 10.

# Operações de Atualização sobre Relações

As seguintes violações podem ser causadas (update):

- Domínio: violada se algum atributo não estiver contido no domínio definido.
- Chave e Entidade: modificar o valor de uma chave primária é equivalente a excluir uma tupla e incluir outra;
  - as restrições e problemas de INSERT e DELETE têm que ser observadas.

# Operações de Atualização sobre Relações

- Integridade referencial: possivelmente violada se algum atributo de chave estrangeira for alterado;
  - o SGBD deve garantir que o novo valor se refira a algum valor existente na relação referenciada.

# Modelagem ER - R: Banco de Dados Empresa



# Exemplo Banco de Dados Empresa

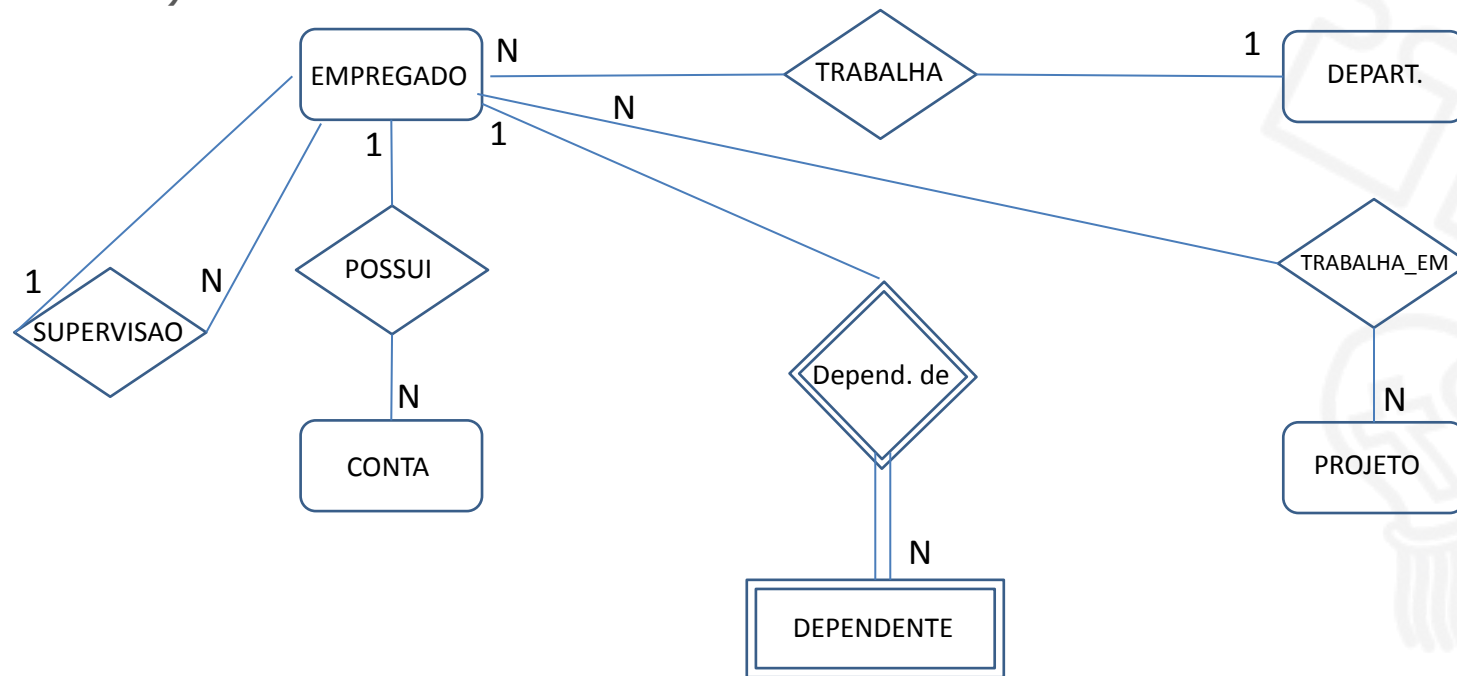
A empresa é organizada em departamentos. Cada **departamento** possui um nome único. Cada **empregado** trabalha em uma certa quantidade de **projetos**, cada um deles possuindo nome único, código único e uma única localização. Cada **empregado** possui **dependentes** com nome, sexo e data de nascimento. O empregado também possui **conta** bancária.

# Exemplo Banco de Dados Empresa

Cada **empregado** é acompanhado e tem seu nome, número da carteira de trabalho, endereço, salário, sexo e data de nascimento registrados. Cada **empregado** é alocado a um **departamento**, porém pode trabalhar em diversos **projetos**. É necessário acompanhar a quantidade de horas que cada **empregado** dedica a cada **projeto** semanalmente. Também é necessário acompanhar o supervisor direto de cada **empregado**.

# Exemplo Banco de Dados Empresa

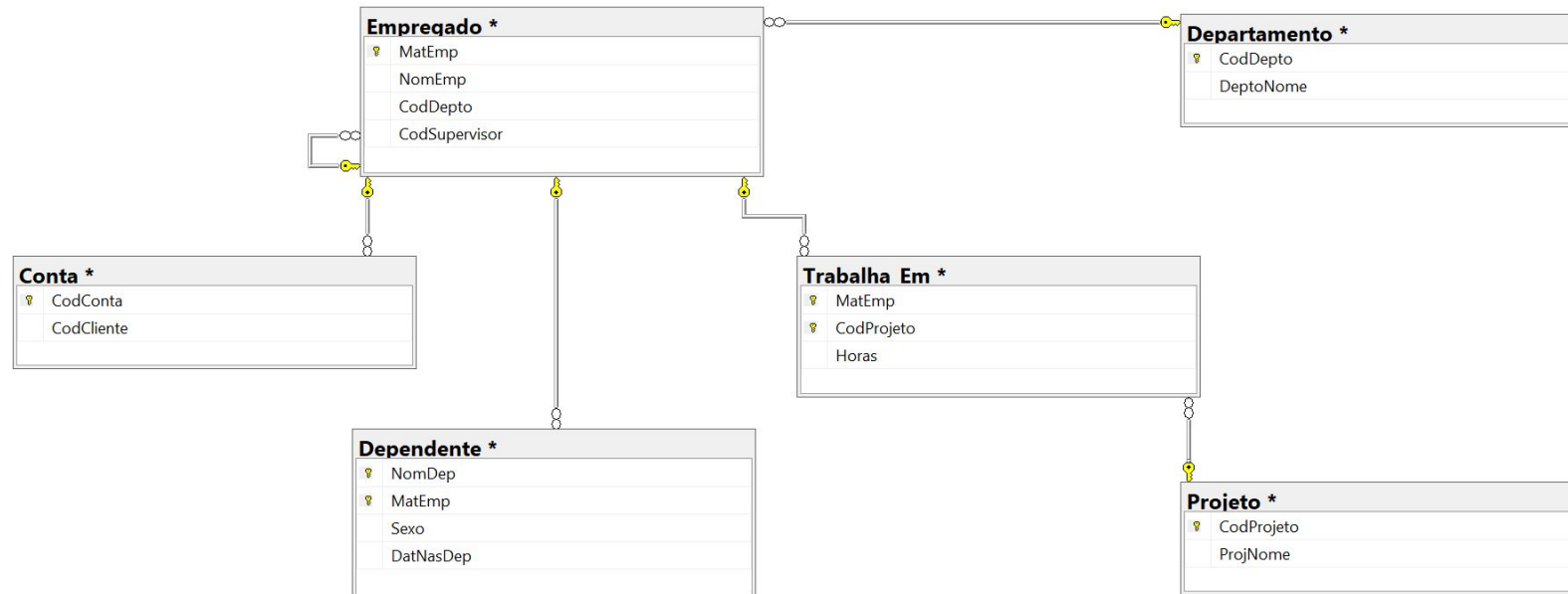
Modelo Entidade-Relacionamento (versão adaptada - atributos foram omitidos)



Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Exemplo Banco de Dados Empresa

Modelo Relacional (alguns atributos foram omitidos)



Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# Normalização



# Normalização

Processo através do qual esquemas de relação são sucessivamente decompostos até que satisfaçam determinadas propriedades ou formas normais.

Proposto por Codd (1972), o processo avalia cada relação sob os critérios de cada forma normal e as decompõe, se necessário.

# Primeira forma normal

Primeira forma normal (1FN): eliminação de atributos compostos e/ou multivalorados.

1FN impede as “relações dentro de relações”


# Primeira forma normal

A relação Departamento não está na 1FN

(a)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations



(b)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}



Remover o atributo DLOCATION (multivalorado) que viola a 1FN e coloca-lo em uma relação separada.



# Segunda forma normal

Segunda forma normal (2FN): Uma relação R está na 2FN se todo atributo de R não pertencente a uma de suas chaves **for totalmente dependente da chave primária (mesmo ela sendo composta)**

O esquema de relação

EMP-PROJ (**SSN**, PNUMBER, HOURS, **ENAME**, PNAME, PLOCATION)

não está na 2FN, porque:

**ENAME** está associado a **SSN** e

{PNAME, PLOCATION} está associado a PNUMBER

# Terceira forma normal

Terceira forma normal (3FN): Eliminar dependência transitiva de atributo **não**-chave.

# Terceira forma normal

Exemplo:

EMPRESTIMO(CodEmprest, DataEmprest, DataDevol, CodUsuario, NomeUsuario, DataNasc, CodLivro, Titulo, AnoPublic, Edicao, NumPaginas)

Dependências:

CodEmprest □ DataEmprest, DataDevol, CodUsuario, CodLivro.

CodUsuario □ NomeUsuario, DataNasc.

CodLivro □ Titulo, AnoPublic, Edicao, NumPaginas.

# Definição Geral

Uma relação  $R$  está normalizada (ou seja, nas 3FNs) se todos os seus atributos são dependentes exclusivamente de suas chaves.

# Oracle Apex



# Oracle Apex


Acesse o link abaixo e crie sua conta gratuita para usar o Oracle APEX

<https://apex.oracle.com/en/>

Atenção: nenhuma informação sobre cobrança será necessária. Se for exigido dados de cartão de crédito, não os forneça.

# Oracle Apex

Clique em Sign In

 APEX Platform Solutions Learn Community 

Sign In

 Oracle DevLive São Paulo - April 3, 2024

## Build enterprise apps 20x faster with 100x less code.

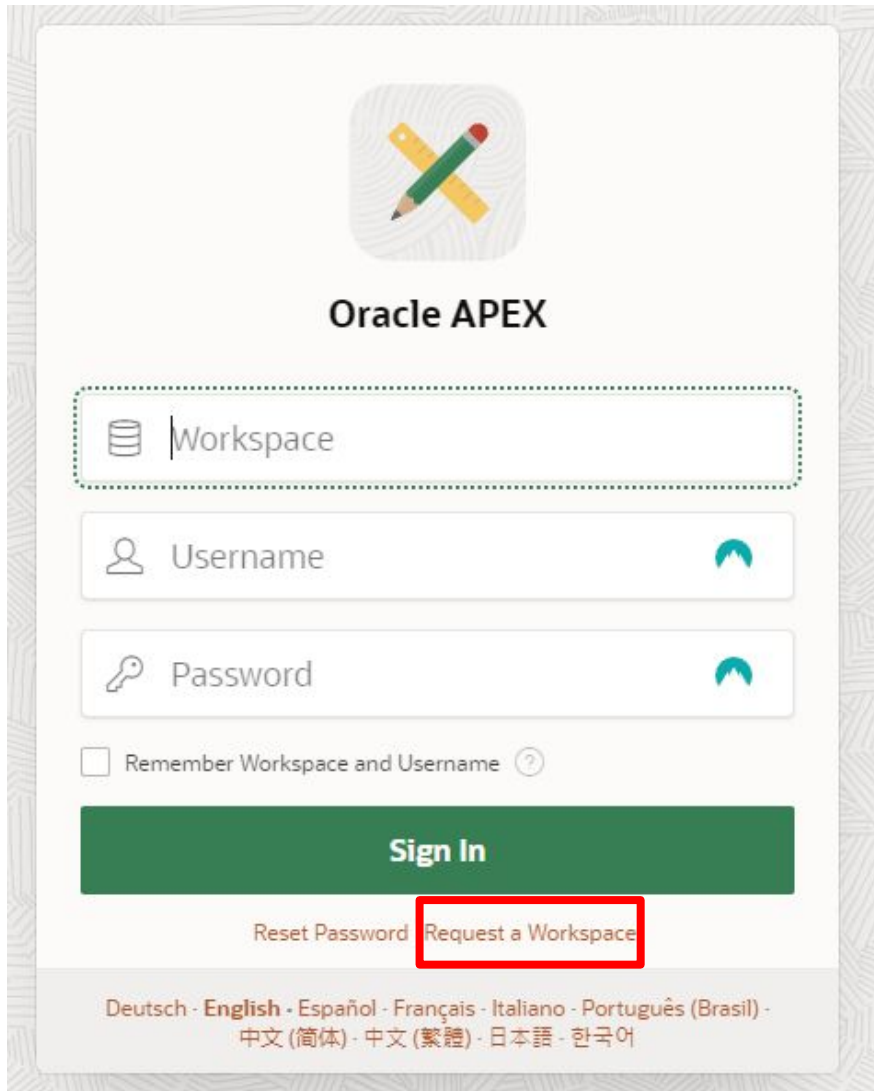
Oracle APEX is the world's most popular enterprise low-code application platform.

Start for Free Today

View Pricing

Get your free cloud service or install it on premises.

# Oracle Apex



The image shows the Oracle APEX login page. At the top is the Oracle APEX logo, which consists of two crossed pencils (one yellow, one green) inside a rounded square. Below the logo is the text "Oracle APEX". There are three input fields: "Workspace" (with a database icon), "Username" (with a person icon), and "Password" (with a key icon). Each of the last two fields has a small blue eye icon to its right. Below these fields is a checkbox labeled "Remember Workspace and Username" with a question mark icon to its right. A large green "Sign In" button is below the checkbox. At the bottom, there are two links: "Reset Password" and "Request a Workspace", the latter of which is highlighted with a red rectangular box. The footer contains a list of languages: Deutsch, English, Español, Français, Italiano, Português (Brasil), 中文 (简体), 中文 (繁體), 日本語, and 한국어.

Oracle APEX

Workspace

Username

Password

☐ Remember Workspace and Username ?

**Sign In**

[Reset Password](#) [Request a Workspace](#)

Deutsch · **English** · Español · Français · Italiano · Português (Brasil) · 中文 (简体) · 中文 (繁體) · 日本語 · 한국어

Clique em Request a Workspace



# Oracle Apex

## Request a Workspace

Identification

First Name

Last Name

Email   
A link to activate your workspace will be sent to this email address.

Workspace   
This workspace name uniquely identifies your development environment and is used when signing in.

---

Country

Usage ☒ Personal ☐ Organization

[Oracle Privacy Policy](#)

Cancel

Next >

Preencher com seus dados. Você precisará criar um nome para o seu workspace (guarde esta informação).

# Oracle Apex

## Request a Workspace



Please help us by filling out this short survey prior to signing up for a workspace. The information provided by you will be collected and used pursuant to the [Oracle Privacy Policy](#)

\* Are you new to Oracle APEX?

☒ Yes

☐ No

\* Do you plan to use APEX for education or training?

☒ Yes

☐ No

<

Cancel

Next >

# Oracle Apex

## Request a Workspace



This information helps the Oracle APEX administrator understand how you intend to use this service.

\* Why are you requesting this service? 

I plan to use APEX for education.



Cancel

Next >

# Oracle Apex

## Request a Workspace



### ORACLE APEX SERVICE AGREEMENT

PLEASE SCROLL DOWN AND READ ALL OF THE FOLLOWING TERMS OF USE CAREFULLY. THIS IS A LEGAL AGREEMENT ("AGREEMENT") BETWEEN YOU AND ORACLE AMERICA, INC. STATING THE TERMS AND CONDITIONS THAT GOVERN YOUR USE OF THE ORACLE APEX SERVICE. IF YOU ARE ENTERING INTO THIS AGREEMENT ON BEHALF OF A COMPANY OR LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS AND CONDITIONS, IN WHICH CASE THE TERMS "YOU" AND "YOUR" SHALL REFER TO SUCH ENTITY. BY CLICKING "I ACCEPT" OR BY USING THE ORACLE APEX SERVICE, YOU ARE AGREEING TO ALL OF THE TERMS AND CONDITIONS STATED HEREIN. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT CLICK "I ACCEPT." AND DO NOT USE THE ORACLE APEX SERVICE

☒ I accept the terms



Cancel

Next >

# Oracle Apex

## Request a Workspace



First Name **Henrique**

Last Name **Batista**

Email

Workspace **henriquebatistaworkspace**

Country **Brazil**

Usage **Personal**

<

Cancel

Submit Request

# Oracle Apex

## **Workspace Requested!**



Você receberá um e-mail para confirmar (pode levar alguns minutos até o envio do e-mail)

You will receive an email to activate your workspace once this request has been approved.

# Oracle Apex

## Request a Workspace



Workspace Successfully Created

Your request for an account has been approved.

[Continue to Sign In Screen >](#)

Clique em “Continue to Sign In Screen”

# Oracle Apex



Acesse o link abaixo e faça login na sua conta para usar o Oracle APEX

<https://apex.oracle.com/en/>



# Oracle Apex

Clique em Sign In

 APEX Platform Solutions Learn Community 

Sign In

 Oracle DevLive São Paulo - April 3, 2024

## Build enterprise apps 20x faster with 100x less code.

Oracle APEX is the world's most popular enterprise low-code application platform.

Start for Free Today

View Pricing

Get your free cloud service or install it on premises.

# Oracle Apex



Oracle APEX

 Workspace

 Username

 Password

☐ Remember Workspace and Username [?](#)

**Sign In**

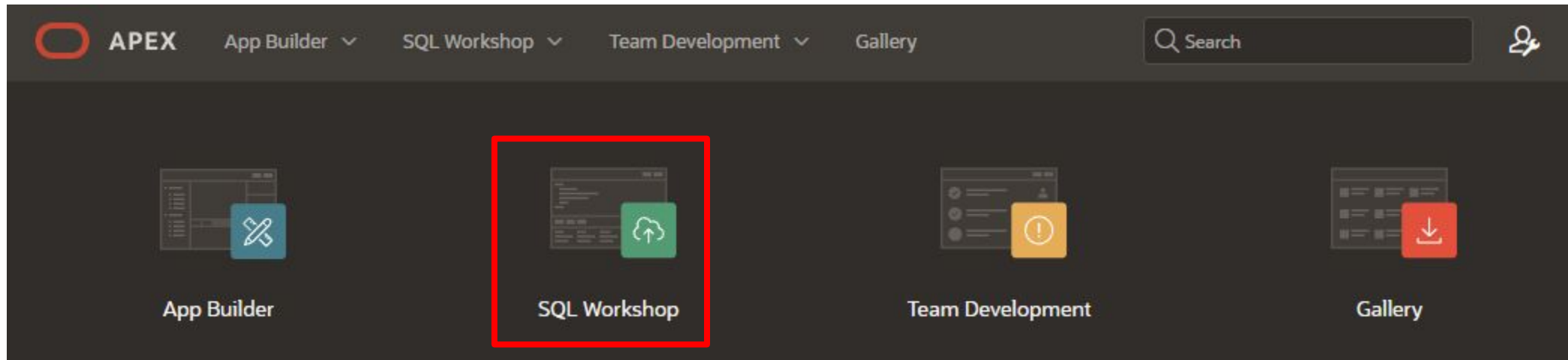
[Reset Password](#) [Request a Workspace](#)

Deutsch · **English** · Español · Français · Italiano · Português (Brasil) ·  
中文(简体) · 中文(繁體) · 日本語 · 한국어

Faça o login informando o nome do seu workspace criado anteriormente.

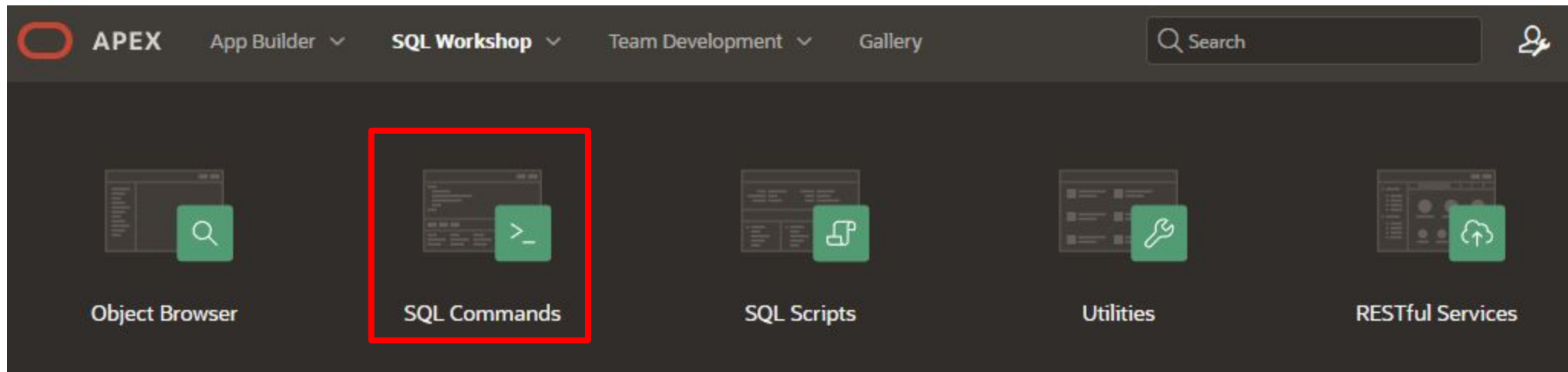
# Oracle Apex

Após o login, esta será a tela inicial, clique em SQL Workshop



# Oracle Apex

Na tela seguinte, clique em SQL Commands



# Introdução linguagem SQL



# Linguagem SQL

Utilize o Oracle Apex para criar uma ou mais tabelas no banco de dados.

Insira alguns elementos (atenção às violações de restrições) na tabela e realize consultas.

# Introdução

Junto com o modelo Relacional de Codd, foi posposto uma linguagem chamada DSL/Alpha para manipulação dos dados no modelo.

A IBM criou uma simplificação desta linguagem, chamada Square.

Aprimoramentos na Square levaram a uma linguagem chamada SEQUEL, posteriormente renomeada para SQL.

# Introdução

Linguagem padrão para lidar com Banco de Dados (BD).

Praticamente todos SGBDs (Oracle, SQLServer, MySQL, etc.) que estão no mercado suportam a linguagem SQL. Muitos bancos de dados NoSQL suportam o padrão SQL Like

SQL tem uma ligação com o modelo relacional pois o resultado de uma consulta SQL é uma tabela (também chamada de conjunto resultado).



# Classes de Comandos SQL

- A linguagem SQL se divide em três subgrupos:
  - Linguagem de Definição de Dados (**DDL - *Data Denition Language***):
    - Usada para definição dos esquemas
  - Linguagem de Manipulação de Dados (**DML - *Data Manipulation Language***):
    - Utilizada pelos usuários para manipulação dos dados (inclusão, alteração e exclusão).

# Classes de Comandos SQL

- Linguagem de Controle de Dados (**DCL - *Data Control Language***):
  - Utilizada para conceder e retirar privilégios de usuários de BD em objetos de BD
- Linguagem de consulta de Dados (**DQL - *Data Query Language***):
  - Usada para recuperação de dados
- Linguagem de controle de transação (**TCL - *transaction Control Language***):
  - Utilizada para lidar com transações no banco de dados.

# Classes de Comandos SQL

Por exemplo, para criar uma nova tabela é necessário utilizar o comando CREATE TABLE (definição de dados).

```
CREATE TABLE Aluno (  
    matricula INT,  
    nome VARCHAR(30),  
    CONSTRAINT pk_matricula PRIMARY KEY  
    (matricula)  
);
```

Essa instrução cria uma tabela com duas colunas, *matricula* (chave primária da tabela) e *nome*

# Classes de Comandos SQL

Para popular o banco de dados é necessário o comando INSERT.

```
INSERT INTO Aluno (matricula, nome) VALUES (10, 'Maria');
```

Essa instrução adiciona um linha à tabela Aluno com o valor 10 na coluna *matricula* e o valor 'Maria' na coluna *nome*

# Classes de Comandos SQL

Com o banco populado é possível recuperar valores com o comando **SELECT** (manipulação de dados):

```
SELECT nome  
FROM Aluno  
WHERE matricula = 10;
```



Nome
Maria

# Criando Tabelas no Banco



# DDL (Linguagem de Definição de Dados)

- É a linguagem que define a ESTRUTURA do BD.
- Seus comandos permitem a criação, alteração e exclusão de objetos em um banco de dados.
- Seus principais comandos são:

Comando	Definição	Exemplos
<b>CREATE</b>	Cria objetos do BD.	Criação da tabela CLIENTE
<b>ALTER</b>	Altera objetos do BD.	Alteração da tabela CLIENTE incluindo a coluna TELEFONE
<b>DROP</b>	Remove objetos do BD.	Exclusão da tabela CLIENTE

# Comando Create table

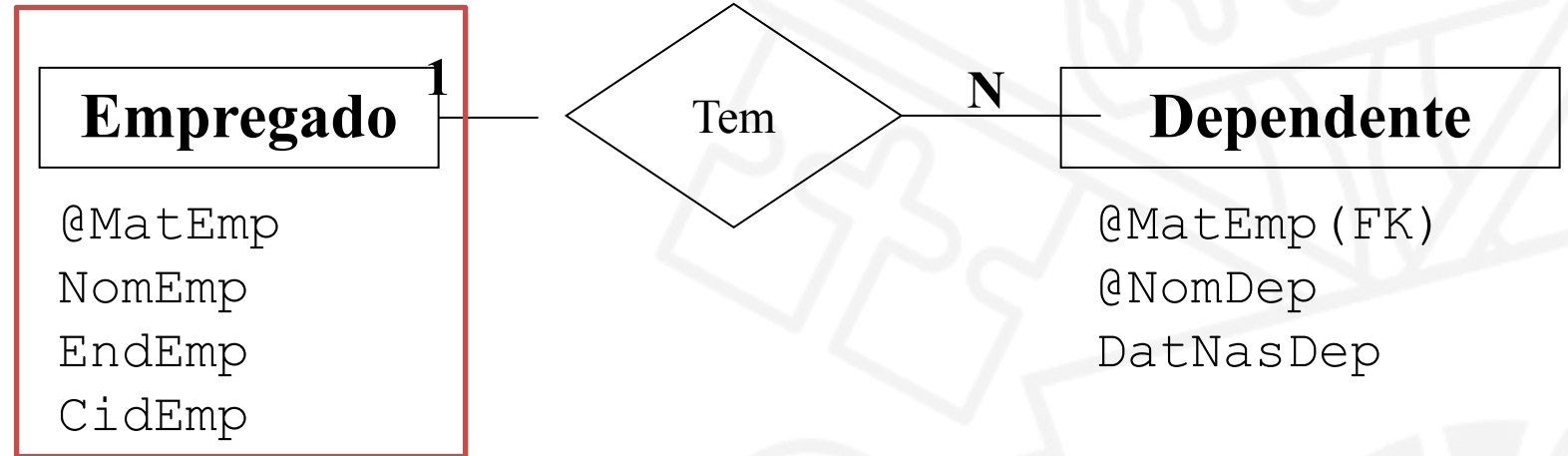
- Comando utilizado para criar uma nova relação (tabela), dando a ela:
  - Nome
  - Especificando seus atributos (suas colunas)
    - Nome e tipo de dado (domínio).
    - Restrições do atributo (not null)
  - Especificando suas restrições (Tipo, chave).
    - Especificado depois que os atributos forem declarados.
    - Ou com o comando ALTER TABLE.



# Comando Create table

Create table Empregado

```
(  
  MatEmp smallint not null,  
  NomEmp char(30) not null,  
  EndEmp char(80) null,  
  CidEmp char(20) null,  
  Constraint PK_EMP primary Key(MatEmp)  
);
```

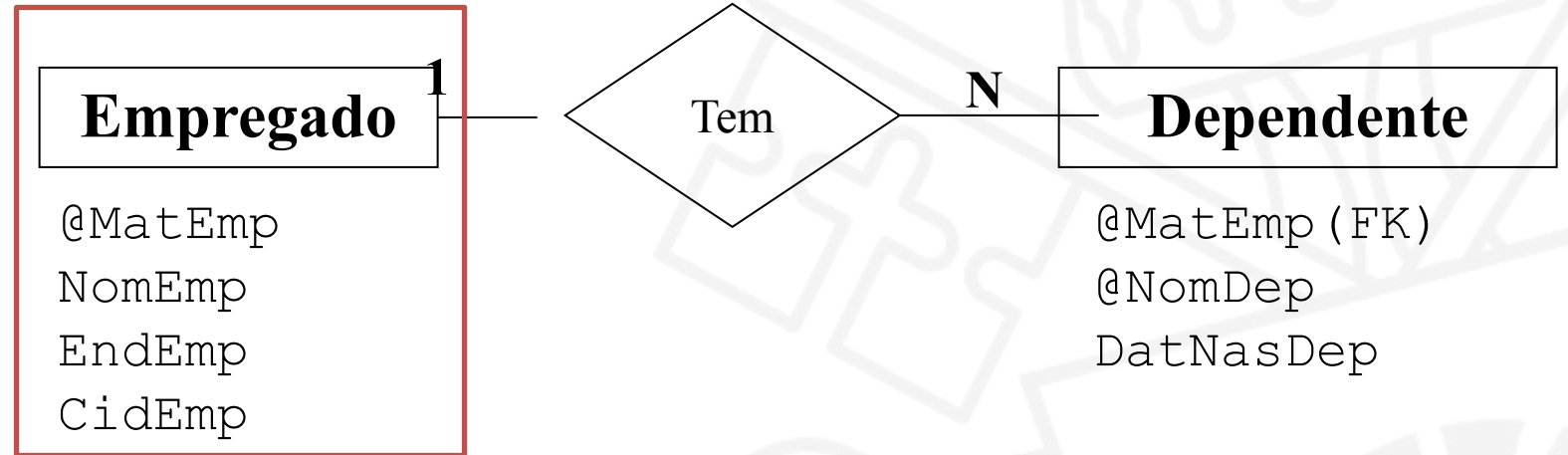


# Comando Create table

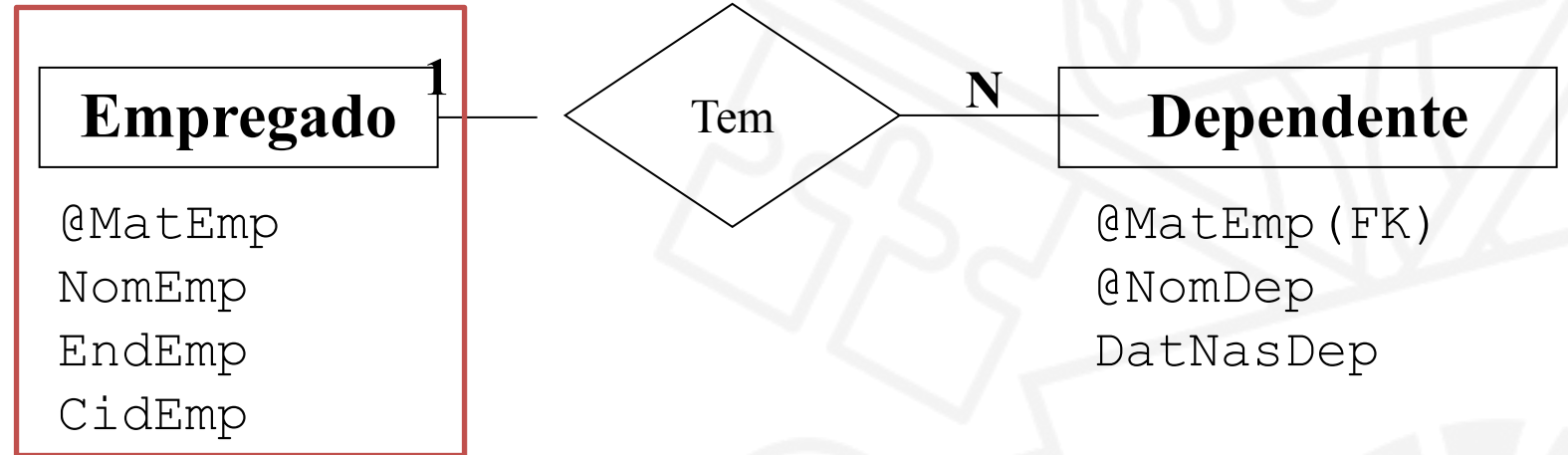
Constraint é uma restrição de chave primária, criada sobre a coluna MatEmp e recebe o nome de PK\_EMP (são definidas com a cláusula PRIMARY KEY).

**Create table** Empregado

```
(  
  MatEmp smallint not null,  
  NomEmp char(30) not null,  
  EndEmp char(80) null,  
  CidEmp char(20) null,  
  Constraint PK_EMP primary Key(MatEmp)  
);
```



# Comando Create table



Create table Empregado

```
(  
  MatEmp smallint not null,  
  NomEmp char(30) not null,  
  EndEmp char(80) null, ← Não é necessário especificar  
  CidEmp char(20) null,  
  Constraint PK_EMP primary Key(MatEmp)  
);
```

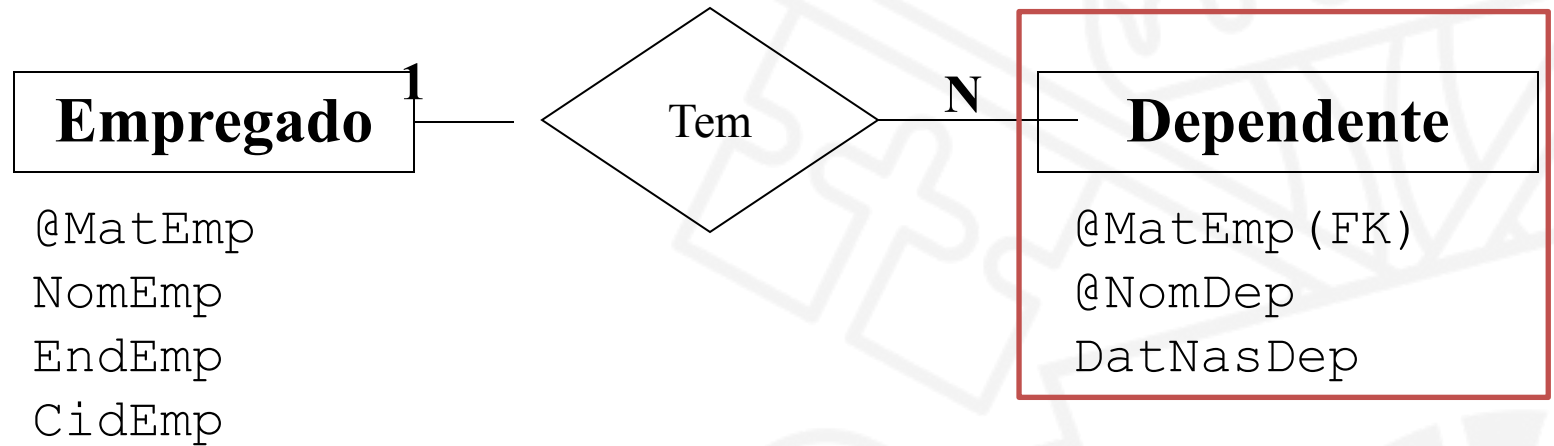
# Atributo Chave Estrangeira

- Similar a instrução para criar a tabela Empregado, mas com algumas exceções:
  - Como empregados diferentes podem ter dependentes com o mesmo nome, é preciso mais do que a coluna MatEmp para garantir a unicidade, portanto a tabela tem chave primária composta de duas colunas.

# Atributo Chave Estrangeira

- A tabela dependente contém outro tipo de restrição, denominada **restrição de chave estrangeira** (definida com a cláusula FOREIGN KEY).
  - Restringe os valores da coluna MatEmp na tabela Dependente para que sejam incluídos apenas valores existentes na tabela Empregado.

# Atributo Chave Estrangeira



```
Create table Dependente
(
  NomDep char(30),
  DatNasDep date,
  MatEmp smallint,
  Constraint PK_DEP primary Key (MatEmp, NomDep),
  Constraint FK_EMP_DEP foreign Key (MatEmp)
    References Empregado (MatEmp)
);
```

# Modificando e excluindo tabelas



# Comandos para Alterações no Esquema SQL

- Alter table
  - Altera a definição de uma tabela no banco de dados
- Ações:
  - Adicionar uma coluna
  - Retirar uma coluna
  - Adicionar restrições de tabela
  - Retirar restrições de tabela



# Comandos para Alterações no Esquema SQL

## Adicionar uma Coluna

**ALTER TABLE** <nome da tabela> **ADD** <nome da coluna>  
<tipo da coluna> [<restrição do atributo>] ;

## Exemplo

```
ALTER TABLE EMPREGADO ADD UF_EMP CHAR(2) NULL
```

# Alter Table (Adicionar Coluna)

É necessário entrar com um valor para o novo atributo UF\_EMP de cada tupla de EMPREGADO. Pode ser feito pela especificação da cláusula *default* ou pelo comando UPDATE.

Neste caso, se não for especificado nenhuma cláusula *default*, o novo atributo será inserido como NULL automaticamente, a restrição NOT NULL não é permitida neste caso.

# Alter Table (Remover Coluna)

Remover uma coluna:

Para remover uma coluna, deve-se optar por CASCADE ou RESTRICT em termos de comportamento para eliminação.

```
ALTER TABLE <nome da tabela> DROP COLUMN <nome da coluna>
```

Exemplo

```
ALTER TABLE EMPREGADO DROP  
COLUMN UF_EMP
```

Alguns SGBDs exigem a **COLUMN**, após **DROP**

# Alter Table (Modificar Restrições)

Ainda é possível modificar as restrições especificadas em uma tabela, adicionando ou eliminando uma restrição.

Eliminando uma restrição:

Para ser eliminada, a restrição precisa ter recebido um nome quando foi especificada.

```
ALTER TABLE <nome da tabela> DROP CONSTRAINT <nome>;
```

# Alter Table (Modificar Restrições)

Excluindo uma Primary Key:

```
ALTER TABLE DEPENDENTE DROP  
CONSTRAINT PK_DEP
```

Excluindo uma Foreign Key:

```
ALTER TABLE DEPENDENTE DROP CONSTRAINT FK_EMP_DEP
```

# Alter Table (Modificar Restrições)

Adicionando uma Restrição:

É realizada pela palavra chave **ADD**, seguida da nova restrição:  
Se for uma Primary Key:

```
ALTER TABLE <nome da tabela> ADD CONSTRAINT <nome>  
PRIMARY KEY (<nomes da colunas>);
```

Exemplo

```
ALTER TABLE EMPREGADO ADD CONSTRAINT PK_EMP  
PRIMARY KEY (MATEMP)
```

# Alter Table (Modificar Restrições)

Adicionando uma Restrição:  
Se for uma Foreign Key:

```
ALTER TABLE <nome da tabela> ADD CONSTRAINT <nome>  
FOREIGN KEY (<nome coluna>) REFERENCES <nome tabela>;
```

Exemplo

```
ALTER TABLE DEPENDENTE ADD CONSTRAINT FK_EMP_DEP  
FOREIGN KEY (MATEMP) REFERENCES EMPREGADO (MatEmp)
```

Obs.: para criar uma restrição, a coluna que será PK, já tem que ter sido criada anteriormente como uma coluna normal

Referência: Navathe. **Sistemas de Banco de Dados** – 2011, 6ª Ed.

# DROP Table

Pode ser usado para eliminar uma tabela de um banco de dados  
Sintaxe:

```
DROP TABLE <nome da tabela> ;
```

Exemplo

```
DROP TABLE  
DEPENDENTE
```



# DML (Linguagem de Manipulação de Dados)

# DML (Linguagem de Manipulação de Dados)

Linguagem que manipula os dados no banco.  
Seus comandos permitem a recuperação, inserção, alteração e exclusão de dados.  
Seus principais comandos são:

Comando	Definição	Exemplos
INSERT	Adiciona dados a uma tabela.	Inserir o João como novo cliente na tabela CLIENTE
UPDATE	Atualiza os dados de uma tabela.	Alterar o número do telefone do cliente João na tabela CLIENTE
DELETE	Exclui dados de uma tabela.	Exclui o cliente Geraldo da tabela CLIENTE

# Comando INSERT

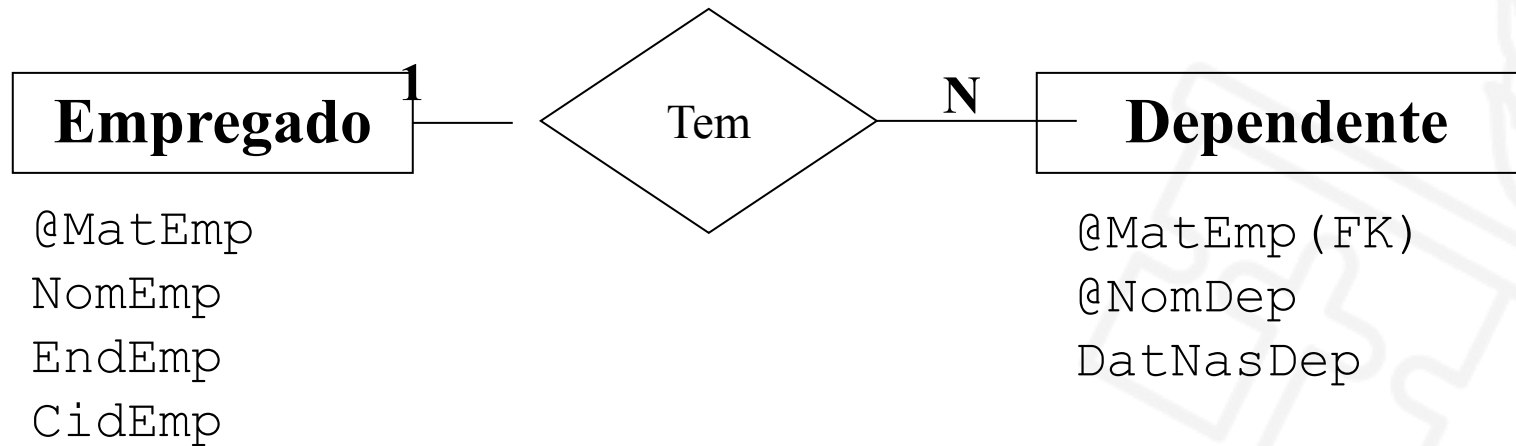
- É usado para adicionar uma única tupla (linha) em uma relação (tabela).
- Existem três componentes principais no comando INSERT:
  - O nome da tabela na qual serão adicionado os dados.
  - Os nomes das colunas que serão populadas dentro da tabela.
  - Os valores que serão usados para popular as colunas.

# Comando INSERT

- Os valores devem ser relacionados na mesma ordem em que as colunas foram especificados no comando CREATE TABLE, caso o nome das colunas não seja explicitados.
- Sintaxe:

**INSERT INTO** <nome da tabela>[(<nome da coluna 1>, <nome da coluna 2>, <nome da coluna n>)] **VALUES** (<valor constante 1>, <valor constante 2>, <valor constante n>)

# Comando INSERT



```
INSERT INTO Empregado (MatEmp, NomEmp, EndEmp , CidEmp )  
VALUES (10, 'Marcos Ramos', 'Rua X', 'Itabira');  
INSERT INTO Empregado VALUES (11, 'Marcos Ramos', null, 'SP');  
INSERT INTO Empregado (NomEmp, EndEmp , MatEmp, CidEmp )  
VALUES ('Mario Andrade', 'Rua 4', 12, 'Belo Horizonte');
```

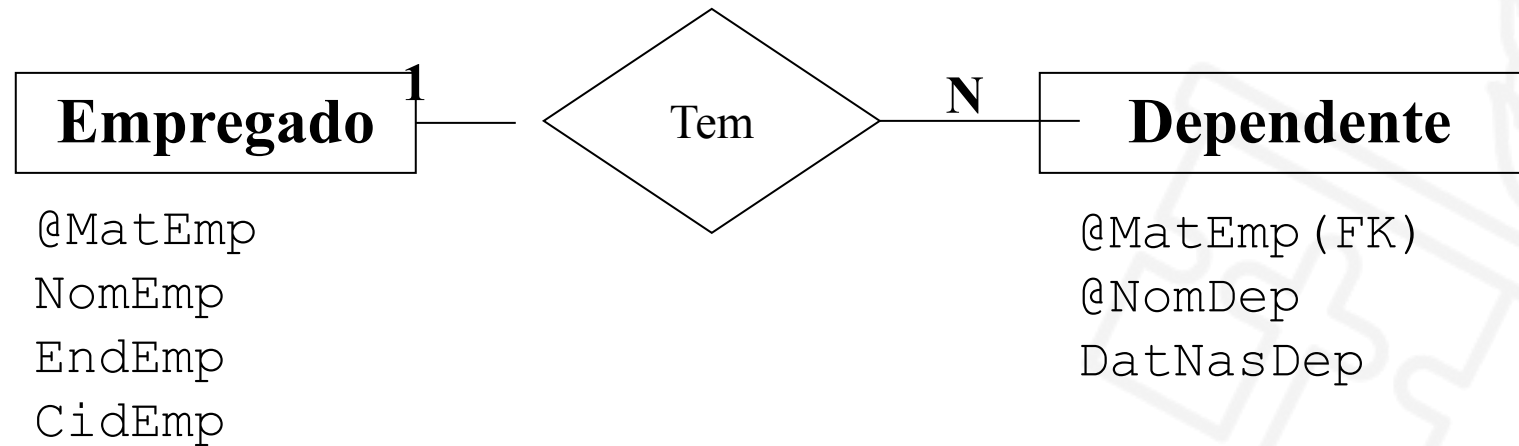
```
INSERT INTO Dependente (NomDep, DatNasDep, MatEmp)  
VALUES ('Carlos Ramos', '2003-01-10', 10);  
INSERT INTO Dependente (NomDep, DatNasDep, MatEmp)  
VALUES ('Joao Andrade', '2003-10-01', 11);
```

# Comando DELETE

- Remove tuplas (linhas) de uma relação (tabela).
- Possui a cláusula **WHERE** para filtrar as linhas a serem excluídas.
  - Poderão se excluídas nenhuma, uma ou várias linhas em um único comando.
- As remoções poderão propagar-se nas tuplas de outras relações.
- Cláusula WHERE omitida indica que todas as linhas na tabela devem ser excluídas (resulta em tabela vazia).
- Sintaxe:

**DELETE FROM** <nome da tabela>  
[**WHERE** <condição da seleção>];

# Comando DELETE



```
DELETE FROM EMPREGADO  
WHERE MatEmp = 12;
```

```
DELETE FROM EMPREGADO;  
--Apaga Todas as linhas da relação empregado  
(tabela empregado se torna vazia).
```

# Comando UPDATE

- Colunas de linhas existentes em uma tabela podem ser modificadas.
- Possui a cláusula **WHERE** para filtrar as linhas a serem alteradas.
- Uma cláusula adicional **SET** dentro do comando UPDATE, **especifica os atributos que serão modificados e seus novos valores.**
- Sintaxe:

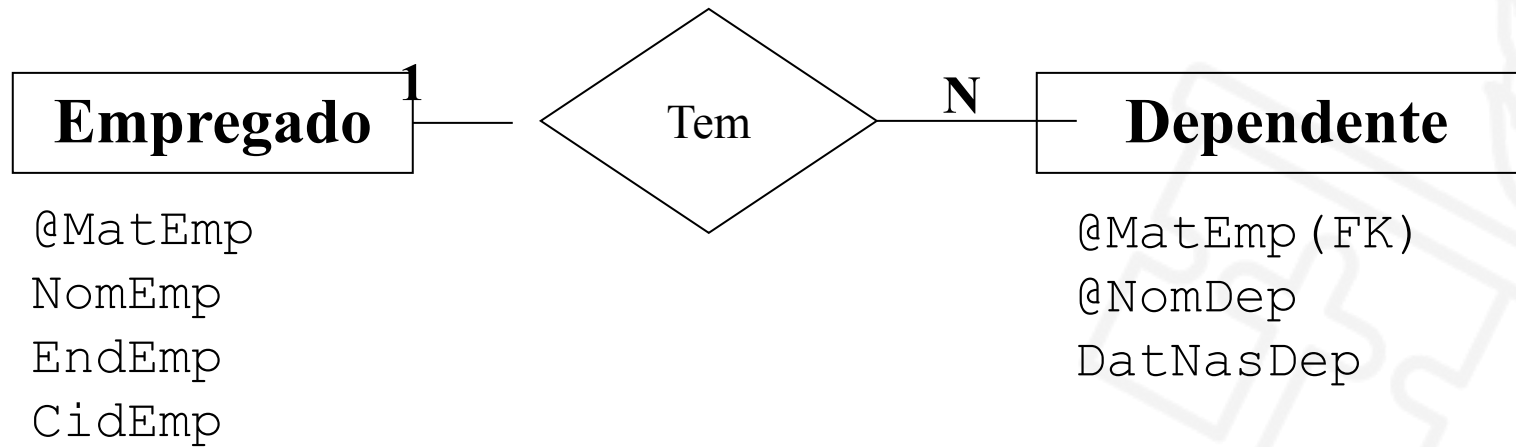
**UPDATE** <nome da tabela>

**SET** <nome da coluna 1> = <expressão do valor 1>, <nome da coluna n> =  
<expressão do valor n>

[**WHERE** <condição da seleção>];



# Comando DELETE



```
UPDATE Empregado
SET EndEmp = 'Rua Agua Santa'
WHERE MatEmp = 10;
```

# Estrutura do comando SELECT



# Comando SELECT

Comando SELECT é a instrução Básica para recuperar informações.

O formato básico da declaração SELECT é composto por três cláusulas (SELECT, FROM e WHERE):

Sintaxe:

```
SELECT <lista atributos>  
FROM <lista de tabelas>  
WHERE <condição>
```

# Comando SELECT

<lista atributos> é uma lista de nomes de atributos cujo valores são recuperados.

<lista de tabelas> é uma lista dos nomes das relações necessárias para o processamento de consulta.

<condição> é uma expressão condicional que identifica as tuplas que serão recuperadas.

# Comando SELECT

Exemplo:

```
SELECT *  
FROM Empregado;
```

Esta consulta seleciona todas as colunas da tabela EMPREGADO.

```
SELECT NomEmp, CidEmp  
FROM Empregado  
WHERE MatEmp = 11;
```

Esta consulta seleciona as colunas **NomEmp** e **cidEmp** da tabela EMPREGADO, cujo valor do atributo MatEmp seja igual a 11.

# Comando SELECT

Esta consulta seleciona as colunas **NomEmp** e **cidEmp** da tabela EMPREGADO, cujo valor do atributo MatEmp seja igual a 11.

Obs.: Como a coluna MatEmp é atributo chave a consulta irá retornar apenas uma linha, a linha do empregado com MatEmp igual a 11 (pois não há outra linha com valor 11 para MatEmp).

# Comando SELECT

É uma das ultimas cláusulas a ser avaliada pelo SGBD.

Antes de gerar o resultado, é necessário conhecer todas as colunas que podem fazer parte do resultado.

A cláusula SELECT **determina quais entre todas as colunas possíveis deverão ser incluídas no conjunto resultado da consulta.**

# Comando SELECT

A cláusula FROM define as tabelas usadas em uma consulta.

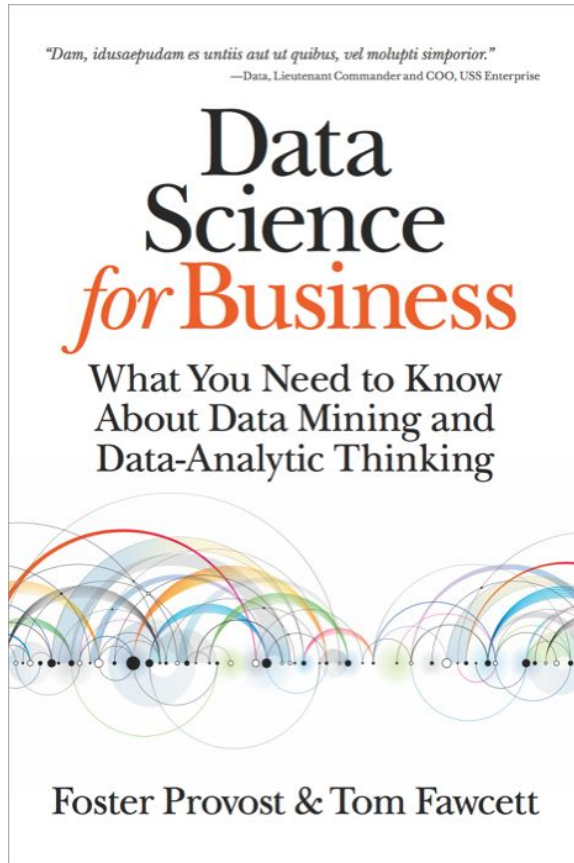
```
SELECT *  
FROM Empregado
```



# Referências



# Principais Referências



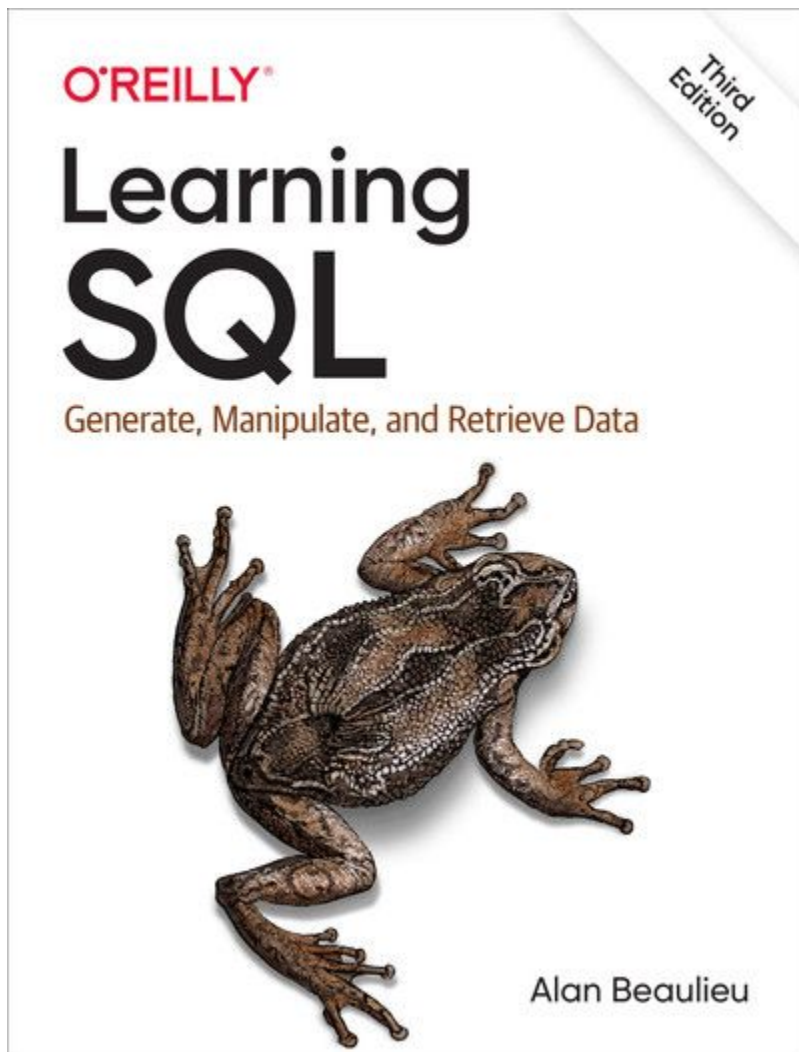
Foster Provost; Tom Fawcett. **Data Science for Business: *What You Need to Know about Data Mining and Data-Analytic Thinking***. Ed. 1; O'Reilly. 2013.

# Principais Referências



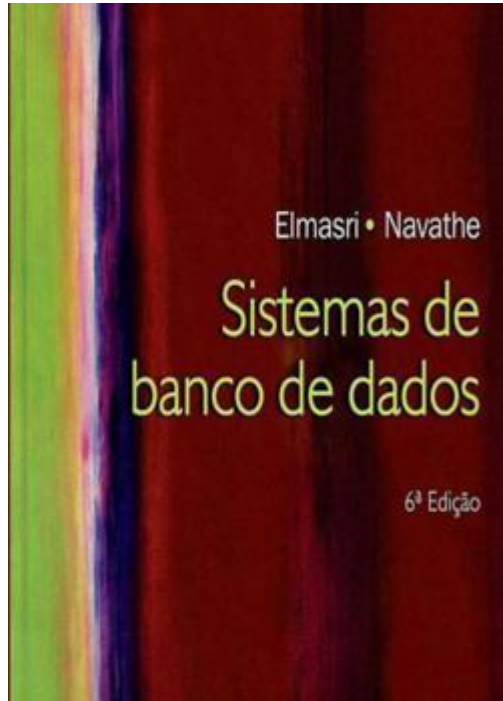
Ramesh Sharda, Dursun Delen, Efraim Turban. **Business Intelligence e Análise de Dados para Gestão do Negócio**. Bookman. 2019.

# Principais Referências



Alan Beaulieu. **Learning SQL**. Ed. 3;  
O'Reilly. 2020.

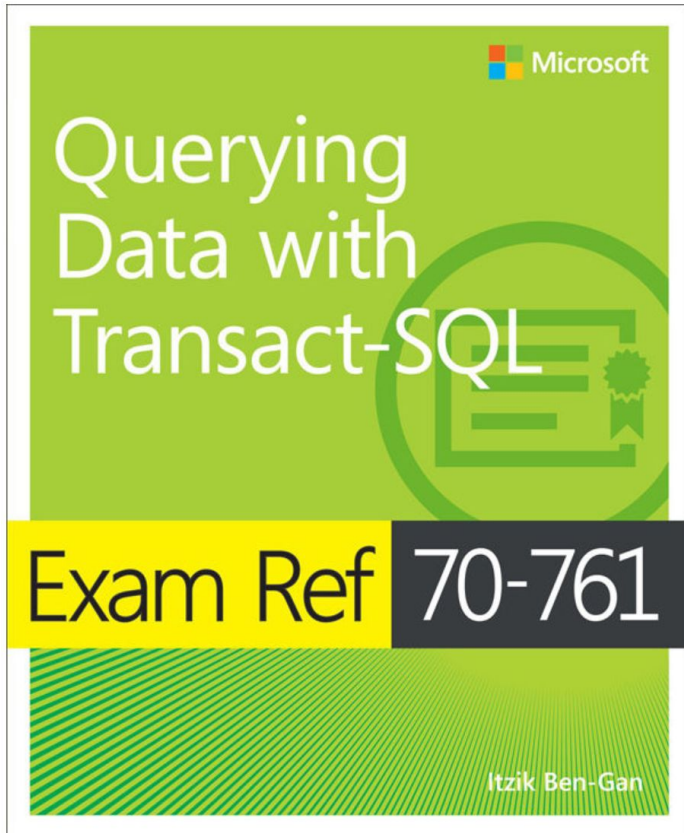
# Principais Referências



Navathe. **Sistemas de Banco de Dados** - 6ª Ed.

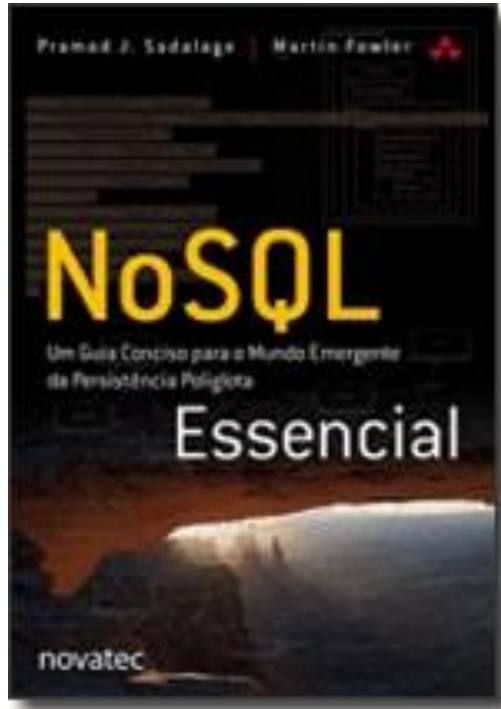


# Principais Referências



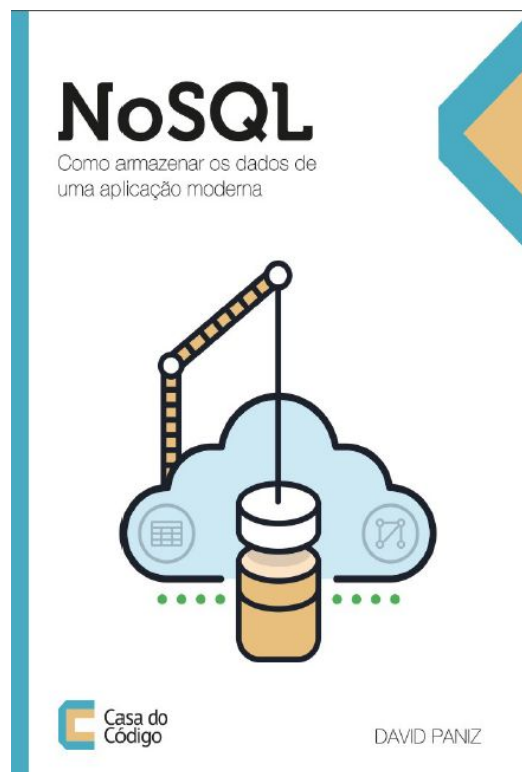
Itzik Ben-Gan. **Exam Ref 70-761  
Querying Data with Transact-SQL.**  
Microsoft Press; Edição: 1 (4 de abril  
de 2017)

# Principais Referências



Pramod J.; Sadalage, Martin Fowler.  
**NoSQL Essencial: Um Guia Conciso  
para o Mundo Emergente da  
Persistência Poliglota.** Novatec  
Editora, 2013.

# Principais Referências



Paniz, David. NoSQL: **Como armazenar os dados de uma aplicação moderna**. Casa do Código, 2017.



# Principais Referências



Boaglio, Fernando. MongoDB: **Construa novas aplicações com novas tecnologias**. Casa do Código, 2017.

# Principais Referências

Armazenando dados com

## Redis

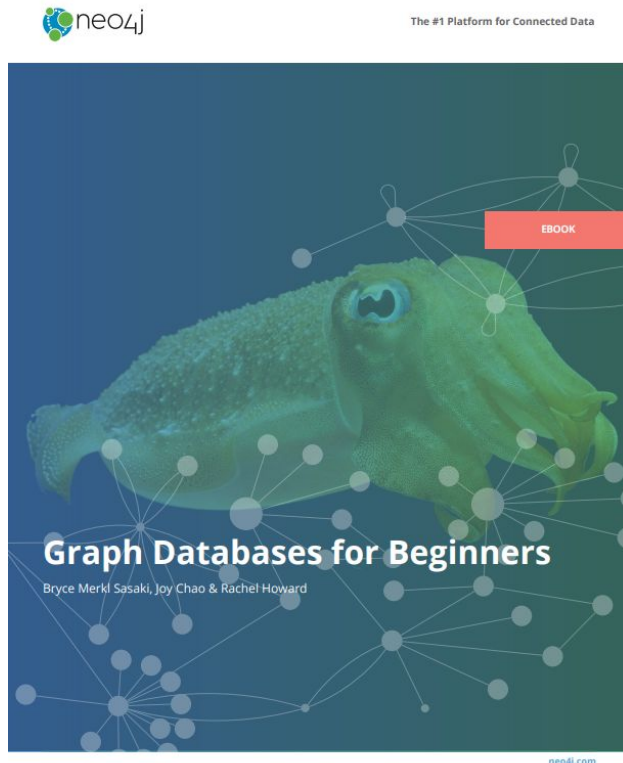


 Casa do  
Código

RODRIGO LAZOTI

Lazoti, Rodrigo. **Armazenando dados com Redis**. Casa do Código, 2017

# Principais Referências



Sasaki, B.; Chao, J.; Howard, R..  
**Graph Databases for Beginners.**  
Neo4j.com.

Download gratuito em:  
<https://neo4j.com/whitepapers/graph-databases-beginners-ebook/?ref=blog>