

# MODELAGEM DE BANCO DE DADOS

ALEXANDRE BARCELOS





# **Modelagem de Banco de Dados**

---

**Alexandre Barcelos**

IESDE BRASIL  
2021

© 2021 – IESDE BRASIL S/A.

É proibida a reprodução, mesmo parcial, por qualquer processo, sem autorização por escrito do autor e do detentor dos direitos autorais.

Projeto de capa: IESDE BRASIL S/A.

Imagem da capa: naulicrea /Envato Elements

CIP-BRASIL. CATALOGAÇÃO NA PUBLICAÇÃO  
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

---

B218m

Barcelos, Alexandre

Modelagem de banco de dados / Alexandre Barcelos. - 1. ed. - Curitiba [PR] : IESDE, 2021.

84 p. : il.

Inclui bibliografia

ISBN 978-85-387-6444-1

1. Banco de dados - Desenvolvimento. 2. Projeto de banco de dados.  
3. Banco de dados relacionais. I. Título.

21-74748

CDD: 005.75

CDU: 004.652.4

---

*Todos os direitos reservados.*



**IESDE BRASIL S/A.**

Al. Dr. Carlos de Carvalho, 1.482. CEP: 80730-200

Batel – Curitiba – PR

0800 708 88 88 – [www.iesde.com.br](http://www.iesde.com.br)

## Alexandre Barcelos

Mestre em Gestão e Tecnologia em Sistemas Produtivos pelo Centro Estadual de Educação Tecnológica “Paula Souza”. Especialista em Docência no Ensino Superior, especialista em Administração de Banco de Dados e especialista em Data Science e Informática para Área da Saúde. Possui ampla experiência como Administrador de Banco de Dados em empresas nacionais e multinacionais de diversos segmentos. Atua como professor em cursos de graduação e de pós-graduação presencial e a distância onde leciona disciplinas relacionadas as áreas de Sistemas Gerenciadores de Banco de Dados, Big Data e Data Science. Produz e revisa material didático para diversos cursos de graduação, pós-graduação e de extensão universitária.

# Vídeos em QR code!



Agora é possível acessar os vídeos do livro por meio de QR codes (códigos de barras) presentes no início de cada seção de capítulo.

Acesse os vídeos automaticamente, direcionando a câmera fotográfica de seu smartphone ou tablet para o QR code.

Em alguns dispositivos é necessário ter instalado um leitor de QR code, que pode ser adquirido gratuitamente em lojas de aplicativos.

# SUMÁRIO

---

<b>1</b>	<b>Introdução ao universo dos dados</b>	<b>9</b>
1.1	Dado, informação e conhecimento	9
1.2	Banco de dados - definição	13
1.3	Conceito de SGBD e BD	17
<b>2</b>	<b>As fases do projeto de banco de dados</b>	<b>23</b>
2.1	Objetivos da modelagem entidade-relacionamento	25
2.2	Modelo conceitual	26
2.3	Modelo lógico	27
2.4	Modelo físico	28
<b>3</b>	<b>Representando o Entendimento do Negócio – diagrama entidade relacionamento (DER)</b>	<b>33</b>
3.1	Definições das convenções	34
3.2	Componentes do diagrama entidade relacionamento (DER)	34
3.3	Realizando a leitura do diagrama entidade relacionamento	46
<b>4</b>	<b>Organizando o projeto de banco de dados – normalização</b>	<b>51</b>
4.1	Relação identificada	52
4.2	Formas normais – definições	53
4.3	Resolvendo a relação de muitos para muitos – N:N	58
4.4	Criação do modelo entidade-relacionamento (MER)	60
<b>5</b>	<b>Do projeto físico ao conceito – engenharia reversa</b>	<b>66</b>
5.1	Engenharia reversa	66
5.2	Identificando as tabelas	71
5.3	Identificando as relações 1:N, 1:1, N:N	75
5.4	Identificando os atributos e os tipos de dados	76
	<b>Resolução das atividades</b>	<b>80</b>





# APRESENTAÇÃO

---



Para que os gestores possam tomar decisões cada vez mais assertivas é necessário que os dados sejam armazenados de maneira correta, permitindo dessa forma que a geração das informações sejam íntegras e confiáveis. É imprescindível que as estruturas de armazenamento sejam criadas de maneira correta na concepção do projeto de banco de dados.

Dessa forma, é primordial que o profissional especializado em dados sabia como modelar as estruturas conceituais, lógicas e físicas que irão armazenar os dados.

Essa obra tem por objetivo apresentar as técnicas utilizadas para criação das estruturas de armazenamento de dados transacionais, ou seja, estruturas que irão armazenar os dados gerados no dia a dia de uma organização.

No primeiro capítulo iremos compreender os conceitos fundamentais para a área de Banco de Dados: Dado, Informação e Conhecimento. Também iremos descrever as características de um sistema gerenciador de banco de dados relacional para entendermos quais são as necessidades que levam uma empresa a adquirir um sistema gerenciador de banco de dados.

No segundo capítulo entenderemos as razões que determinam a criação de um modelo de banco de dados. Nesse capítulo explicaremos as diferenças entre os projetos conceitual, lógico e físico.

No terceiro capítulo iremos construir o Diagrama Entidade Relacionamento-DER. Também estudaremos cada um dos componentes de um DER e, por fim, nós aplicaremos a sintaxe para a leitura desse diagrama.

O quarto capítulo aborda os tipos de relações entre as entidades. Nós iremos entender e aplicar as regras de Formas Normais para obtermos um design de banco de dados que irá permitir o acesso e o armazenamento dos dados de maneira eficiente. Essas etapas minimizam a redundância dos dados e reduzem as possibilidades dos dados se tornarem inconsistentes. Como resultado teremos o Modelo Entidade Relacionamento-MER.

No último capítulo realizaremos o processo de engenharia reversa. Essa atividade permite documentar um banco de dados que já se encontra em produção e que não possui o projeto lógico pois, por meio da utilização desse conceito, será possível entendermos a especificação de um banco de dados caso ela não exista.

Bom estudo!

# 1

## Introdução ao universo dos dados



### Objetivos de aprendizagem

Com o estudo deste capítulo, você será capaz de:

- compreender os conceitos de dado, informação e conhecimento;
- analisar a necessidade de uma empresa adquirir um Sistema Gerenciador de Banco de Dados;
- descrever as características que definem um Sistema Gerenciador de Banco de Dados Relacional.

Quando falamos de banco de dados, devemos entender, primeiro, a definição de dados, informação e conhecimento, pois a modelagem entidade-relacionamento trata da modelagem de um negócio, ou seja, trata-se de modelar os requisitos de dados que atendam às necessidades de determinado negócio, tomando por base as funcionalidades atuais ou desejadas de um sistema de informação.

Para implementarmos um banco de dados que foi modelado anteriormente, necessitamos adquirir um software gerenciador de banco de dados relacional. Dessa forma, para que possamos fazer uma escolha assertiva, é fundamental conhecermos as características desse produto.

## 1.1

### Dado, informação e conhecimento



Vídeo



Nos dias atuais o dado se tornou um dos ativos mais preciosos para as empresas. Ao trabalhar com essa matéria, prima devemos identificar se estamos manipulando dado, informação ou conhecimento.

Os dados são entendidos de diferentes formas em diversos setores. Em sua forma básica, os dados são um conjunto de diferentes símbolos e caracteres cujo significado só se torna claro quando eles se conectam.

Na visão de Rob e Coronel (2014), os dados são brutos, pois eles devem ser processados para terem um significado. Coletar e medir ob-

servações gera dados. Normalmente as máquinas enviam, recebem e processam dados.

Um dado, de maneira genérica, não tem significado, pois ele deve ser colocado dentro de um contexto. Por exemplo, se considerarmos o número 33, ele pode representar a idade de um funcionário, o peso de um determinado produto, uma distância em quilômetros a ser percorrida etc.



Figura 1

Exemplo de dados

Funcionário	Cargo	Idade	Salário
Bob	Engenheiro de dados	42	R\$ 12.500
Meg	Analista de dados	32	R\$ 9.800



Dado

Representa um valor numérico

Fonte: Elaborada pelo autor.

Portanto, podemos definir que o dado é uma sequência simples de sinais e símbolos que não tem um significado.

É necessário que exista um fluxo de aquisição, de análise e de comunicação dos dados para que eles sirvam de suporte nos processos de tomada de decisão (DAMA INTERNATIONAL, 2017).

## 1.1.1 Definição de informação

De acordo com Rob e Coronel (2014), o processamento dos dados brutos resulta em uma informação.

Uma informação é um conjunto de dados colocados dentro de um contexto, ou seja, é uma sequência de dados organizados que devem ser interpretados para que tenham um significado. Ao se integrar os dados a um contexto, eles atingem um nível mais complexo e tornam-se informações. Podemos tomar como exemplo que 32 é a idade de uma funcionária que se chama Meg. Logo, temos uma informação de determinada funcionária.



Figura 2

Exemplo de informação

Funcionário	Cargo	Idade	Salário
Bob	Engenheiro de dados	42	R\$ 12.500
Meg	Analista de dados	32	R\$ 9.800




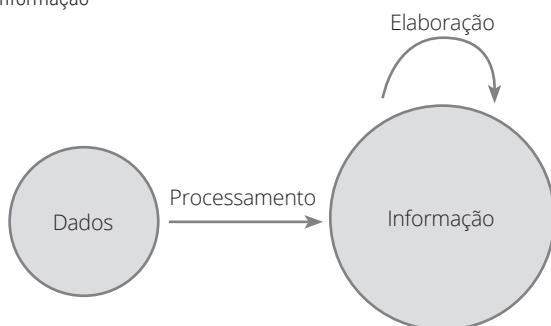
Informação

32 anos é a idade da funcionária Meg

Fonte: Elaborada pelo autor.

A confusão entre dados e informações geralmente surge porque as informações são geradas a partir de um conjunto de dados. Além disso, os dados muitas vezes são interpretados como fatos no contexto do significado coloquial e, portanto, são considerados informações.

 **Figura 3**  
Elaboração da informação




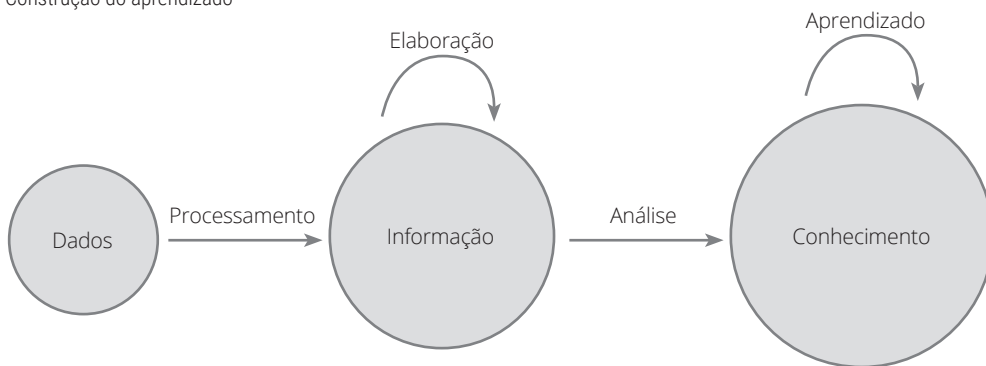
Fonte: Elaborada pelo autor.

Podemos notar que os computadores são muito bons em processar dados e só agora eles estão aprendendo a dar um sentido a isso para derivar informações com a ajuda do aprendizado de máquina.

## 1.1.2 Geração do conhecimento

O conhecimento descreve as informações coletadas sobre determinado fato ou pessoa. O conhecimento de determinada situação permite a tomada de decisões e a resolução de problemas. Assim, o conhecimento influencia o pensamento e as ações das pessoas.

 **Figura 4**  
Construção do aprendizado



Fonte: Elaborada pelo autor.

As máquinas também podem tomar decisões com base em novos conhecimentos a partir da análise das informações. Para adquirir conhecimento, é necessário aplicar tais informações. “O termo conhecimento é interpretado de forma livre com algo que envolve algum grau de inteligência” (ELMASRI; NAVATHE, 2011, p. 700).

O conceito de inteligência ou sabedoria está acima do conceito de conhecimento. O estágio mais complexo é o da construção do conhecimento ou da sabedoria, pois precisamos elaborar as informações para que elas auxiliem a responder a questões mais complexas, conforme um conjunto de leis, normas, boas práticas e princípios.

Por exemplo: imagine um grupo de pessoas em uma sala fechada onde nada entra ou sai. Essas pessoas irão receber uma tarefa de levantar a cadeira em que estão sentadas. Vamos considerar que todas as pessoas possuem a mesma força física e foram criadas em famílias e educadas em escolas de níveis similares, ou seja, possuem um nível semelhante de conhecimento obtido em escolas, leituras, viagens, experiências etc. Algumas dessas pessoas conseguirão resolver o problema e outras não. Então, se todas possuem a mesma força física e os mesmos conhecimentos, não deveriam todas resolver o problema?

A resposta é não, pois a diferença está na forma como cada uma dessas pessoas irá utilizar o conhecimento.

A inteligência é a forma de se resolver problemas por meio da utilização do conhecimento que cada um possui, em que esse conhecimento se aplica na resolução de novos problemas, usando adaptações, analogias etc.

Portanto, podemos entender que as principais diferenças entre dados, informações e conhecimento são:

1. Os dados são pedaços fragmentados de símbolos e caracteres unidos, enquanto a informação é um dado refinado e o conhecimento é uma informação útil. Além disso, os dados podem carecer de contexto quando vistos individualmente, de maneira que as informações fornecem contexto aos dados e o conhecimento traz profundidade no entendimento dessas informações.
2. Vale ressaltar que os dados são incompreensíveis de maneira independente, mas o resultado da informação é a sua compreensão, enquanto o resultado do conhecimento é a

utilização da informação para a resolução de problemas. Os dados não têm sentido sem serem compilados em uma estrutura sensível, enquanto a informação melhora a representação e o conhecimento amplifica a consciência.

3. Dados e informações por si só não são suficientes para fazer quaisquer previsões. Já no conhecimento a previsão é possível se alguém possuir a experiência necessária.
4. Não podemos utilizar os dados para fazer declarações, enquanto as informações são dados agrupados, formando, portanto, declarações. O conhecimento traz a habilidade de se ter uma conclusão deduzida utilizando pedaços de informações que serão juntadas.
5. Os dados não podem ser independentes de uma base para a formação de perguntas. Informação é um texto que responde às questões sobre quem, quando, o quê ou onde, enquanto o conhecimento é um texto que responde às questões sobre por quê e como.
6. A última diferença que podemos levar em consideração é que dados e informações são facilmente transferíveis. Já para transferir conhecimento, é necessário aprender.

## 1.2 Banco de dados - definição

---

 Vídeo



Todas as organizações têm a necessidade de informações. Por exemplo, uma biblioteca mantém uma lista de associados, livros, datas de entrega e multas. Uma empresa mantém informações sobre seus funcionários, departamentos e salários.

Banco de dados é um conjunto de informações relacionadas entre si, referentes a um mesmo assunto e organizadas de maneira útil, com o propósito de servir de base para que possamos recuperar informações, tirar conclusões e tomar decisões.

Antes de 1970 as estruturas de dados hierárquicas e de redes eram os modelos mais populares de armazenamento de informações.

Esses modelos dependem da estrutura dos dados e da maneira como as informações estão vinculadas na hierarquia ou na rede.

Há cinco tipos principais de banco de dados:

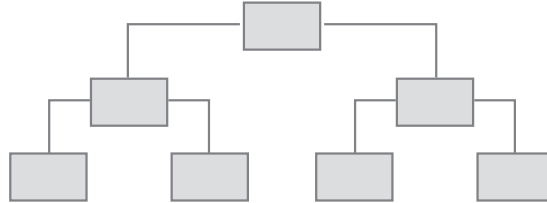
## 1. Hierárquico

Em bancos de dados hierárquicos os registros são organizados em uma hierarquia, como um organograma. Os nós são conectados por meio de links que representam uma relação. Cada tipo de registro nesse modelo é chamado de *nó* ou *segmento*. Um nó representa uma entidade específica.



Figura 5

Esquema de um modelo de banco de dados hierárquico



Fonte: Elaborada pelo autor.

O nó superior é chamado de *raiz*. Cada nó é um subordinado do nó em um nível superior.

O modelo de banco de dados hierárquico segue um relacionamento pai-filho, representando assim uma estrutura em árvore.

Um nó de nível superior é chamado de *pai* e um nó de nível inferior é chamado de *filho*. Um pai pode ter um ou mais nós filhos, mas um nó filho pode ter apenas um pai.

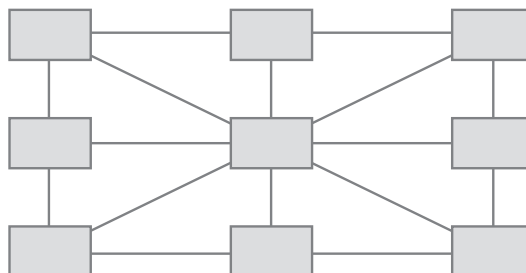
## 2. De rede

O modelo de dados de rede é uma evolução do modelo de dados hierárquico. Na verdade, ele foi projetado para lidar com as desvantagens do hierárquico. O modelo de dados de rede permite implementar a abordagem de muitos para muitos, que não era suportada no hierárquico.



Figura 6

Esquema de um modelo de banco de dados em rede



Fonte: Elaborada pelo autor.



O modelo de dados de rede produz uma estrutura semelhante à da web, interligando vários nós.

### 3. Relacional

No modelo de dados relacional os dados são armazenados ou representados na forma de tabelas, ou seja, de relações. Podemos dizer, portanto, que os dados estão estruturados em forma de tabelas.



Figura 7

Esquema de um modelo de banco de dados relacional



Fonte: Elaborada pelo autor.

O modelo de dados relacional foi proposto por Edgar Frank Codd em 1970 em um artigo chamado *A relational model of data for large shared data banks*.

Conforme Korth, Silverschatz e Sudarshan (2012), um banco de dados relacional é composto de um conjunto de tabelas que se relacionam entre si tendo por base o modelo relacional.

As tabelas são compostas de várias colunas, sendo que cada coluna deve ter um nome exclusivo. As tabelas a seguir ilustram o armazenamento de dados de funcionários e de departamentos.



Tabela 1

Tabela FUNCIONARIOS

↕ID_FUNCIONARIO	↕PRIMEIRO_NOME	↕SALARIO	↕ID_DEPARTAMENTO
100	Steven	24000	90
101	Neena	17000	90
102	Lex	17000	90
103	Alexander	9000	60
104	Bruce	6000	60
105	David	4800	60
106	Valli	4800	60
107	Diana	4200	60
108	Nancy	12000	100
109	Daniel	9000	100

Fonte: Elaborada pelo autor.

**Tabela 2**  
Tabela DEPARTAMENTOS

↕ ID_DEPARTAMENTO	↕ NOME
10	Administration
20	Marketing
30	Purchasing
40	Human Resources
50	Shipping
60	IT
70	Public Relations
80	Sales
90	Executive
100	Finance

Fonte: Elaborada pelo autor.

Note que não utilizamos caracteres especiais quando nomeamos as tabelas e as colunas.

A tabela de funcionários mostra que um funcionário chamado Alexander tem o código 103 e o salário de R\$ 9.000. Vemos que o código do departamento em que ele está alocado é 60.

Ao relacionar a tabela de funcionários com a tabela de departamentos, podemos concluir que o nome do departamento em que o funcionário Alexander trabalha se chama *IT*.

#### 4. Relacional de objetos

Em bancos de dados orientados a objetos, os dados são armazenados ou representados na forma de objetos e classes baseando-se nos princípios da programação orientada a objetos.

**Figura 8**  
Esquema de um modelo de banco de dados relacional de objetos



Fonte: Elaborada pelo autor.

Conceitos como herança, polimorfismo e encapsulamento e propriedades como integridade, concorrência e processamento de

consulta de modelos relacionais constituem os bancos de dados orientados a objetos.

Nesse modelo de dados, todas as entidades são representadas como objetos, e esses objetos possuem suas próprias propriedades (estado) e métodos (comportamento).

## 5. NoSQL

Os bancos de dados NoSQL (*Not Only SQL*) surgiram para superar algumas limitações dos bancos de dados relacionais, principalmente em relação à alta velocidade na geração dos dados e na flexibilidade no armazenamento de dados. Esses bancos de dados foram financiados principalmente por grandes empresas, como Amazon, Google e Facebook, que, em primeiro lugar, sentiram que deveria haver um banco de dados que oferecesse alta velocidade transacional e flexibilidade na manipulação dados.

Os tipos de bancos de dados NoSQL que podemos listar atualmente são:

- documento;
- chave-valor;
- colunar;
- grafo;
- in-memory.

# 1.3 Conceito de SGBD e BD

📺 Vídeo



Para gerenciar um banco de dados com eficiência, é preciso que haja um DBMS (*Database Management System* – Sistema de Gerenciamento de Banco de Dados). Um DBMS é um sistema que armazena, modifica e recupera dados do banco de dados quando solicitado.

Um sistema gerenciador de banco de dados “é um software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados” (HEUSER, 2009, p. 5).

O SGBD é composto das seguintes estruturas:

### 1. Instância (*Instance*)


São as estruturas alocadas na memória RAM do servidor de banco de dados.

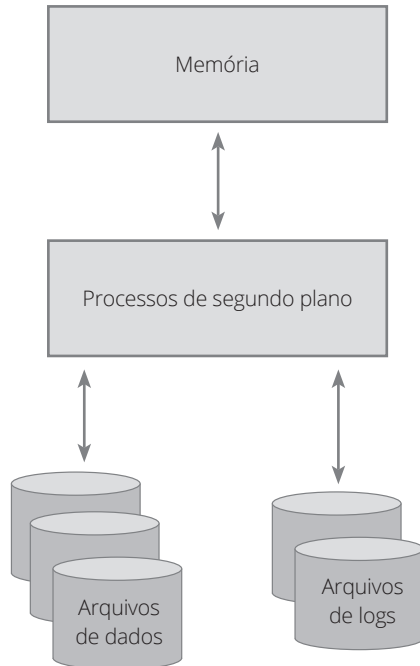
## 2. Banco de dados (*database*)

São os arquivos físicos que armazenam informações de logs transacionais e arquivos de dados que armazenam as tabelas.

## 3. Processos de segundo plano (*background process*)

Esses processos são responsáveis por permitirem a comunicação entre a estrutura lógica (*instance*) e as estruturas físicas (*database*).

 **Figura 9**  
Arquitetura de um SGBDR



Fonte: Elaborada pelo autor.

Para determinar se um software pode ser classificado como um Sistema Gerenciador de Banco de Dados Relacional, devemos verificar se os requisitos a seguir são atendidos.

### 1. Independência dos dados

Um Sistema Gerenciador de Banco de Dados (SGBD) é um software de controle posicionado entre o banco de dados e as aplicações. Ele é responsável por atender e gerenciar as solicitações de acesso aos dados feitas por diversas aplicações.



Figura 10

Dados independentes das aplicações



Fonte: Elaborada pelo autor.

Tipicamente, encontramos uma arquitetura em que usuários e aplicações realizam milhares de requisições para um servidor no qual temos instalado o SGBDR. Ele manipula o banco de dados que geralmente está armazenado em uma estrutura chamada de *storage*, que são equipamentos compostos de dezenas de discos que permitem o armazenamento de petabytes de dados.

## 2. Controle de redundância dos dados

Em um SGBDR ninguém na verdade quer abolir as ocorrências de dados duplicados. Sempre haverá redundância, mas ela será controlada. Exemplo: chave estrangeira.



Tabela 3

Relacionamento entre as tabelas ALUNOS e CURSOS

ALUNOS				Foreign key – chave estrangeira
codigo_aluno	nome	sobrenome	codigo_curso	
L0002345	Jim	Black	C002	Relacionamento
L0001254	James	Harradine	A004	
L0002349	Amando	Holland	C002	
L0001198	Simon	Stuart	S042	

CURSOS	
codigo_curso	nome
A004	Modelagem de Dados
C002	Administração de BD
P301	SQL Básico
S042	SQL Avançado

Primary key – chave primária →

Fonte: Elaborada pelo autor.

### 3. Garantia da integridade dos dados

Mecanismos de controle de bloqueio (*lock*) são o método mais comum utilizado pelos SGBDR. Esses mecanismos garantem que o dado não será alterado ao mesmo tempo por processos diferentes, garantido assim o isolamento. Apesar de haver diversos modos de bloquear um dado que está sendo alterado, os modos mais utilizados de bloqueio são o compartilhado e o exclusivo.

### 4. Compartilhamento dos dados

Se existe um banco de dados, todos os usuários devem acessar todos os dados, pois o banco não é construído apenas para uma pessoa, mas sim para a empresa.

### 5. Privacidade dos dados

Somente usuários devidamente autorizados poderão acessar os seus respectivos dados.

### 6. Segurança dos dados

Envolve todos os conceitos anteriores mais outros recursos técnicos. Vária desde a segurança lógica (software) até a segurança física (instalações).

Para Korth, Silverschatz e Sudarshan (2012), as grandes quantidades de dados devem ser gerenciadas por um SGBD, pois a gestão dos dados engloba a definição das estruturas que irão armazenar os dados e quais mecanismos devem ser utilizados para a manipulação dos dados.

Os SGBD devem possibilitar a recuperação das informações, caso ocorra uma falha no sistema, e impedir a manipulação dessas informações por pessoas que não tenham permissão. Eles devem permitir ainda a realização de auditoria fina com o objetivo de possibilitar a identificação, por exemplo, de quem foi o responsável por determinada manipulação dos dados.

## 1.3.1 Conceito de SGBD e BD

As empresas dependem dos bancos de dados para melhorar o gerenciamento dos seus processos organizacionais, bem como necessitam dos dados organizados e disponíveis para que as tomadas de

decisões sejam baseadas em informações, ou seja, as decisões devem ser tomadas com base em evidências, e não em suposições.

Os bancos de dados contribuem para o gerenciamento de uma grande quantidade de dados e ajudam os usuários a realizar diversas tarefas. As operações de negócios tendem a se tornar mais eficientes, proporcionando assim redução de custos. Os tipos de dados que são facilmente gerenciados são: folha de pagamento, registros de funcionários, informações de alunos, gerenciamento de projetos, processos contábeis, livros de uma biblioteca, inventário, gestão da cadeia de suprimentos, gestão do relacionamento com o cliente etc.

Na visão de Korth, Silverschatz e Sudarshan (2012), os principais segmentos que utilizam o SGBD são:

- bancos e finanças;
- companhias aéreas;
- telecomunicações;
- universidades;
- informações empresariais sobre as áreas de contabilidade, manufatura, recursos humanos, varejo e vendas.

Contudo, de acordo com Rob e Coronel (2014), a implementação de um SGBD pode trazer algumas desvantagens, como:

- complexidade de gerenciamento;
- manutenção;
- dependência do fornecedor;
- necessidade de frequentes atualizações/substituições.



## CONCLUSÃO

Com relação a dado, informação, conhecimento e sabedoria, podemos concluir que:

- dado é o que será armazenado;
- informação é a compreensão dos dados;
- conhecimento é o que se extrai de útil das informações com base no cruzamento de diversas informações;
- sabedoria é a utilização do conhecimento para a resolução de problemas de diversas naturezas.

Um banco de dados é um conjunto de informações organizadas para que possamos recuperá-las com a finalidade de que nos auxiliem nas tomadas decisões com base em evidências, e não em suposições.

O objetivo da construção de um banco de dados deve ser a busca da integração das atividades gerenciais e operacionais na empresa.

As organizações competem pelo domínio do conhecimento científico e tecnológico.

Um SGBD é um software que atende às requisições feitas pelas aplicações para selecionar, inserir, atualizar e deletar dados de um BD.



## ATIVIDADES



### Atividade 1



Discuta cada uma das definições a seguir.

- a) Dado. Cite três exemplos.
- b) Informações. Cite três exemplos.
- c) Conhecimento.
- d) Sabedoria.



### Atividade 2



O que é um SGBD e quais são as suas principais funções?



### Atividade 3



Cite três vantagens e três desvantagens de se utilizar um SGBD.



## REFERÊNCIAS

DAMA INTERNATIONAL. *DAMA-DMBOK: data management body of knowledge*. 2. ed. Basking Ridge: Technics Publication, 2017.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de bancos de dados*. 6. ed. São Paulo: Pearson, 2011.

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2009.

KORTH, H. F.; SILVERSCHATZ, A.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Campus-Elsevier, 2012.

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e gerenciamento*. São Paulo: Cengage Learning, 2014.



## As fases do projeto de banco de dados



### Objetivos de aprendizagem

Com o estudo deste capítulo, você será capaz de:

- entender as razões que levam à criação do modelo de banco de dados;
- explicar as diferenças entre os projetos conceitual, lógico e físico.

Para que qualquer ideia se transforme em um produto, seja, por exemplo, a construção de uma casa, seja a fabricação de um carro, a montagem de um avião, a criação de um banco de dados etc., é primordial que exista um projeto composto por diversas fases bem definidas e segregadas, o que, por consequência, irá exigir a participação de muitos profissionais especializados.

De acordo com Rob e Coronel (2014), a construção do projeto de banco de dados é fundamental. Esse projeto tem por objetivo definir e construir as estruturas que serão utilizadas no armazenamento e no gerenciamento dos dados que o usuário final irá manipular.

Antes de abordarmos aspectos relativos à construção de um banco de dados, vamos fazer uma comparação com um projeto resumido de construção de uma casa.

Imagine que alguém deseja construir uma casa. Inicialmente, ela só existe na mente dos futuros proprietários, como ideias. Às vezes, os futuros habitantes não sabem explicar em detalhes tudo aquilo que querem que a casa tenha ou sequer sabem se o que almejam é viável. Os sonhos podem estar cheios de contradições e impossibilidades e, sem dúvidas, isso não é um problema no mundo dos sonhos, porém, na realidade, quaisquer inconsistências e obstáculos devem ser resolvidos antes que alguém possa iniciar a construção da casa.

Um empreiteiro precisa de um plano sólido, composto por um conjunto de plantas e por uma descrição dos materiais a serem utilizados, como o tamanho das vigas do telhado, a capacidade do encanamento e muitas outras coisas. Esse profissional irá seguir o plano, pois tem o conhecimento técnico necessário para construir o que está descrito no projeto. Mas como as ideias do proprietário da casa serão transmitidas para o empreiteiro? É aqui que o arquiteto se envolve.

O arquiteto é o intermediário entre o proprietário e o empreiteiro, uma vez que possui as habilidades para traduzir as ideias dos futuros habitantes para o projeto. Portanto, o arquiteto irá ouvir a descrição das ideias do proprietário e fazer diversas perguntas. As habilidades desse profissional em extrair as ideias, colocá-las em um formato que permita a discussão e análise, dar sugestões, descrever opções sensatas, documentá-las e confirmá-las com o proprietário da casa são os pilares para fornecer ao futuro morador um plano da casa que ele quer.

A compreensão dos sonhos feita pelo arquiteto é transformada em esboços da nova casa – apenas esboços! Estes consistem em plantas baixas e várias impressões que mostram os requisitos funcionais da casa, e não os detalhes da construção. Isso é o que chamamos de *modelo conceitual*, ou seja, a primeira versão.

Se alguma das partes do modelo não forem satisfatórias ou forem mal interpretadas, ele pode ser facilmente alterado. Essa mudança precisaria apenas de um pouco de tempo e de uma borracha ou uma folha de papel nova. Lembre-se, é apenas mudar um modelo. O custo das mudanças nessa fase é muito baixo – certamente, é mais barato fazer alterações na planta baixa ou nas dimensões do telhado antes do início da construção. O modelo da casa é, então, revisado novamente e outras alterações são feitas. O arquiteto continua a explorar e esclarecer os sonhos e faz diversas sugestões alternativas, até que todas as questões sejam resolvidas e o modelo esteja estável e pronto para a aprovação final do proprietário da casa.

Na fase seguinte, o arquiteto converte o modelo em um projeto técnico, isto é, um plano que o empreiteiro pode utilizar para construir a casa. Os cálculos são feitos para determinar, por exemplo, o número de portas, a espessura das paredes, a espessura do contrapiso, as dimensões do encanamento e a quantidade de vigas necessária para a construção do telhado. Essas são questões técnicas que não necessitam do envolvimento do cliente.

Vemos, portanto, que, enquanto o modelo conceitual aborda a questão “o que deve ser feito no projeto?”, o projeto lógico traz o questionamento “como o projeto deve ser construído?”.

A modelagem conceitual de dados é semelhante ao trabalho de um arquiteto que irá transformar “coisas” que só existem como ideias em um projeto viável, para que seja criado fisicamente.

Logo, a modelagem entidade-relacionamento trata da modelagem de um negócio. Para ser mais preciso, ela modela os requisitos de dados para determinado negócio, com base nas funcionalidades atuais ou desejadas em um futuro sistema. Durante essa fase, devemos estudar e analisar as necessidades da empresa. Para isso, entrevistamos os usuários e os gerentes, a fim de identificarmos as informações necessárias e as futuras especificações do banco de dados.

Para modelar uma solução para certo problema de negócio, é necessário entendermos, com um grau razoável de detalhes, do que se trata esse negócio. Dessa forma, a modelagem entidade-relacionamento, que veremos neste capítulo, é uma técnica utilizada para descrever o entendimento das necessidades de informações referentes a um negócio.

## 2.1 Objetivos da modelagem entidade-relacionamento

▣ Vídeo



A modelagem conceitual de dados tem por objetivo garantir que todas as informações necessárias para gerenciar determinado negócio sejam reconhecidas adequadamente e, conseqüentemente, armazenadas.

Os modelos devem ser completos, os requisitos devem ser conhecidos antes do início da implementação e as dependências entre as informações devem ser claras.

Um dos objetivos do modelo é garantir que cada informação seja armazenada apenas uma vez; em outras palavras, podemos afirmar que as informações não podem ser redundantes.

Com a utilização da modelagem de dados, é possível estudar e analisar as informações que pertencem a determinado contexto. As informações estudadas são estruturadas e apresentadas na forma de um modelo lógico de dados.

Portanto, para construirmos um modelo de dados, devemos utilizar uma linguagem de modelagem de dados.

Para Elsmari e Navathe (2011), existem diversas linguagens gráficas e textuais que permitem descrever os modelos em diferentes níveis de abstração e com objetivos variados. Cada descrição é chamada de *esquema de banco de dados*.

## 2.2 Modelo conceitual

---

▶ Vídeo



Essa é a primeira etapa do projeto. Representa a realidade (contexto de negócio) por meio da visão global e genérica dos dados e de seus relacionamentos.

Nessa fase, devemos capturar formalmente os requisitos de informação de um banco de dados. Seu objetivo é coletar todas as informações dentro do contexto de negócio, as quais serão armazenadas no banco de dados, sem que se retratem aspectos relativos ao sistema gerenciador de banco de dados que será utilizado.

Para Heuser (2009), a descrição do banco de dados é realizada com a utilização do modelo conceitual, que tem por objetivo descrever o banco de dados de modo independente do SGBDR (Sistema de Gerenciamento de Bancos de Dados Relacionais), em que ele será implementado fisicamente. Dessa forma, nós sabemos quais dados podem ser armazenados, porém ainda não determinamos a maneira como eles serão armazenados no SGBD (Sistema de Gerenciamento de Bancos de Dados).

São funções de um modelo conceitual de dados:

- a. entender os processos e as regras de negócio;
- b. expressar as necessidades de informação da organização;
- c. apoiar a definição da abrangência do sistema, delimitando seu escopo;
- d. definir as necessidades, possibilitando uma melhor definição da estrutura de armazenamento, de modo que seja apresentada uma estrutura flexível, facilitando sua manutenção.

O modelo conceitual de dados apresenta as principais necessidades de armazenamento, dentro de um contexto de negócio, por meio de uma visão global dessas necessidades mais importantes – ou seja, sem o detalhamento e os seus relacionamentos.

Segundo Heuser (2009), a forma mais utilizada para criar o modelo conceitual é a abordagem entidade-relacionamento, com a construção do diagrama chamado de *diagrama entidade-relacionamento* (DER).

Essa é uma macrodefinição ou descrição de alto nível, que retrata a realidade de uma organização, o processo de negócio, o setor, a repartição e o departamento. Seu foco é o entendimento do contexto e a representação de uma realidade de uma forma simples e de fácil compreensão pelo usuário final.

Geralmente, o modelo conceitual é uma descrição textual sobre o entendimento dos requisitos de negócio capturados. Muitos profissionais mais experientes com a modelagem de dados utilizam, nessa fase, as representações do modelo lógico, pois não há uma regra formal de como os atributos devem ser capturados.

## 2.3 Modelo lógico

---

▣ Vídeo



No modelo lógico são definidas as estruturas de dados que irão implementar os requisitos identificados na primeira fase, ou seja, no projeto conceitual. Nesse momento serão construídas, de maneira gráfica, as estruturas em que os dados serão armazenados. Nessa fase também serão definidos os nomes para os componentes, bem como as ações que um componente irá exercer sobre outro componente. Ainda, essa fase não determina nenhuma característica específica de um SGBD (HEUSER, 2009).

O modelo lógico de dados permite a descrição dos elementos que serão interpretados pelo SGBD, como o detalhamento dos atributos, as chaves de identificação, a integridade referencial e as entidades.

A Figura 1 apresenta um exemplo do modelo lógico de dados, por meio da representação gráfica das estruturas de armazenamento, de acordo com um contexto de negócio, possuindo um nível de detalhamento maior. Nela, podemos identificar as entidades, os atributos, os relacionamentos e as restrições básicas, como chave primária, chave estrangeira e atributos obrigatórios.

O exemplo mostra a entidade empregados com os atributos `cod_emp`, `nome`, `sobrenome`, `data_contratacao`, `salario` e `pct_comissao`, e a entidade departamentos com os atributos `codigo_departamento`

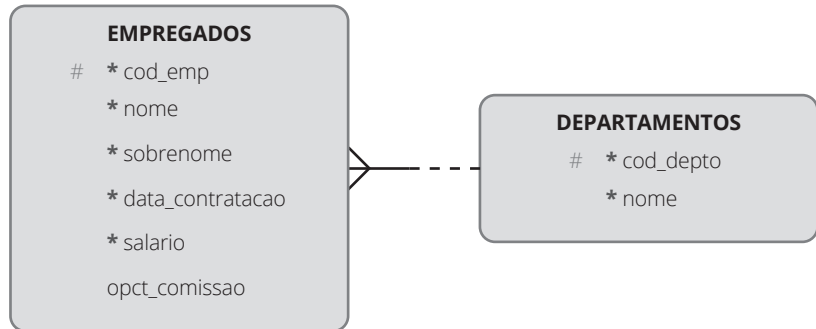
e nome. O relacionamento entre essas entidades implementa as seguintes regras:

- cada departamento pode ser composto por muitos empregados;
- cada empregado deve ser alocado para um e somente um departamento.



Figura 1

Modelo lógico de dados: relação entre empregados e departamentos



Fonte: Elaborada pelo autor.

Atenção: não utilizamos caracteres especiais (acentuação, espaço, parênteses, aspas, apóstrofo etc.) nos nomes das entidades e dos atributos.

## 2.4 Modelo físico

📺 Vídeo



Segundo Korth, Silverschatz e Sudarshan (2012), nessa última etapa, é descrito como os dados devem ser realmente armazenados, pois deverão ser definidos os parâmetros físicos de acesso ao banco de dados, procurando otimizar a performance do ambiente como um todo. O modelo relacional ou físico de dados é derivado do modelo lógico, no qual se encontram detalhados os componentes de estrutura física do banco de dados, como tabelas, colunas, tipos de dados, índices e restrições.

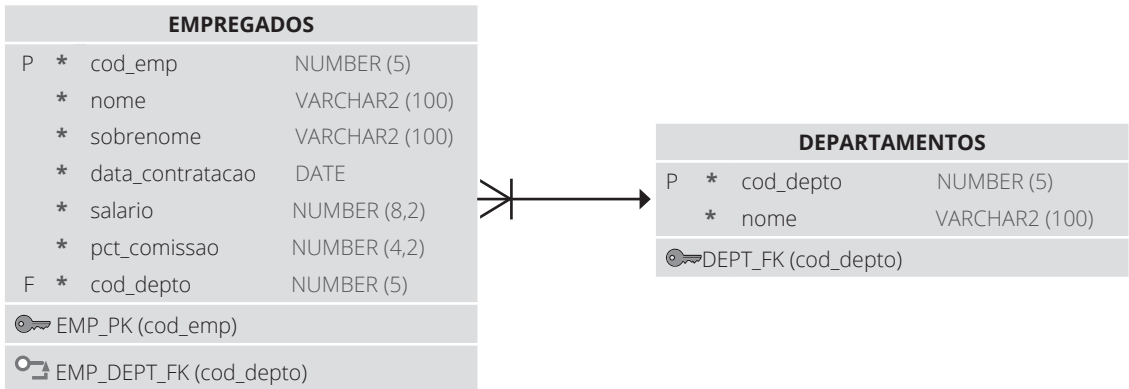
Portanto, nesse ponto, podemos realizar a criação do banco de dados, utilizando um SGBDR. No modelo físico ou relacional, devemos aplicar os elementos físicos mencionados, considerando o SGBD que será utilizado para implementação. Exemplos de SGBD relacionais: Oracle, SQL Server, DB2, Informix, MySQL e PostGree.

Essa é a última etapa do projeto de banco de dados. Após essa fase, utilizaremos uma linguagem de definição de dados reconhecida pelo SGBD, a DDL (Data Definition Language ou, em português, Linguagem de Definição de Dados, uma das subdivisões da linguagem SQL). O conjunto de instruções DDL é normalmente chamado de *script de criação* de banco de dados e será executado no SGBD específico.

A Figura 2 apresenta o exemplo do modelo relacional ou físico de dados, por meio da representação gráfica da estrutura de armazenamento, levando em consideração os componentes físicos necessários para a criação do banco de dados e utilizando um sistema gerenciador de banco de dados.

O exemplo mostra as características das tabelas empregados e departamentos. Foram definidos os tipos e as precisões dos dados que cada uma das colunas poderá armazenar, conforme descrito no modelo lógico de dados, além das chaves primárias, da chave estrangeira, das restrições NOT NULL, entre outros detalhes físicos.

 **Figura 2**  
Modelo físico de dados: relação entre empregados e departamentos



Fonte: Elaborada pelo autor.

Atenção: também não utilizamos caracteres especiais (acentuação, espaço, parênteses, aspas, apóstrofo etc.) nos nomes das tabelas e das colunas.

Dessa forma, o conjunto de dados que será armazenado no banco de dados em determinado instante de tempo recebe o nome de *instância*.

Nessa fase, é recomendável construir o dicionário de dados. Trata-se do registro das características físicas do banco de dados, como os nomes das tabelas, os nomes das colunas, os tipos e o tamanho dos dados, a chave primária, a chave estrangeira, a chave única, a restrição de verificação, o nome das restrições e os comentários sobre as tabelas e colunas.

O Quadro 1 mostra as informações sobre a tabela empregados.



**Quadro 1**

Dicionário de dados: tabela empregados

Tabela		Empregados			
Descrição		Tabela que armazena as informações relativas a todos os empregados contratados da empresa. Relaciona-se com a tabela departamento.			
Coluna	Tipo de dados	Tamanho	Restrição	Nome restrição	Descrição
cod_emp	Númerico	5	PK	EMP_PK	Identificador único do empregado.
nome	Caracter	100	NN	EMP_PNOME>NN	Primeiro nome do empregado.
sobrenome	Caracter	100	NN	EMP_SNOME>NN	Último nome do empregado.
data_contratacao	Data		NN	-	Data em que o funcionário foi contratado. Deve ser igual ao registro na carteira de trabalho.
salario	Númerico	8,2	NN,CK	EMP_SALMIN	Salário mensal do empregado. Deve ser maior do que zero.
pct_comissao	Númerico	4,2	-	-	Coluna opcional. Somente os empregados do departamento de vendas serão comissionados.
cod_depto	Númerico	5	FK	EMP_DEPT_FK	Código do departamento em que o empregado é alocado. Chave estrangeira que referencia a coluna cod_departamento da tabela departamentos.

Fonte: Elaborado pelo autor.

Da mesma forma, o Quadro 2 mostra as informações sobre a tabela departamentos.





## Quadro 2

Dicionário de dados: tabela departamentos

Tabela	Departamentos				
Descrição	Tabela que armazena as informações sobre todos os departamentos da empresa.				
Coluna	Tipo de dados	Tamanho	Restrição	Nome restrição	Descrição
cod_depto	Numérico	5	PK	DEPT_FK	Identificador único do departamento.
nome	Caracter	100	NN	DEP_NOME_NN	Nome do departamento.

Fonte: Elaborado pelo autor.

Não existe um modelo formal para a criação de um dicionário de dados-DD, e este pode ser construído utilizando um programa, por exemplo, uma planilha eletrônica, um editor de textos etc.



## CONCLUSÃO

Como vimos, para construirmos um banco de dados, precisamos elaborar o projeto ou modelo de banco de dados. Este é composto de: projeto/modelo conceitual, que descreve de maneira textual ou gráfica os requisitos que determinam quais dados devem ser armazenados no banco de dados; projeto/modelo lógico, em que construímos o diagrama entidade-relacionamento (DER); e projeto/modelo físico ou relacional, etapa na qual será criado o modelo entidade-relacionamento (MER), com base no DER. Nesse diagrama são registrados aspectos físicos de armazenamento dos dados, como tipo e precisão.



## ATIVIDADES



### Atividade 1



Qual é a importância da criação de um projeto de banco de dados?



### Atividade 2



Cite as principais características dos seguintes projetos:

- Conceitual
- Lógico
- Físico



#### Atividade 3



Qual é o nome do diagrama que deve ser criado no projeto lógico?



#### Atividade 4



Qual é o nome do diagrama que deve ser criado no projeto físico?



## REFERÊNCIAS

- ELSMARI, R.; NAVATHE, S. B. *Sistemas de bancos de dados*. 6. ed. São Paulo: Pearson, 2011.
- HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Sagra Luzzatto, 2009. (Série Livros Didáticos da UFRGS).
- KORTH, H. F.; SILVERSCHATZ, A.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Elsevier, 2012.
- ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e gerenciamento*. São Paulo: Cengage Learning, 2014.

# 3

## Representando o Entendimento do Negócio – diagrama entidade relacionamento (DER)



### Objetivos de aprendizagem

Ao final do estudo deste capítulo, você será capaz de:

- construir o diagrama entidade relacionamento (DER).
- identificar os componentes do DER.
- entender as convenções utilizadas na construção do DER.
- realizar a leitura do DER.

O diagrama entidade relacionamento tem como objetivo identificar quais dados devem ser armazenados e o que eles representam para um determinado contexto de negócio. Em outras palavras, esse diagrama é utilizado para modelar os requisitos de dados para um determinado negócio tendo como base a funcionalidade atual ou desejada de um futuro sistema.

Um diagrama E-R pode expressar graficamente a estrutura lógica geral de um banco de dados. Os diagramas E-R são simples e claros – qualidades que podem ter motivado o amplo uso do modelo E-R. (SILBERSCHATZ; KORTH; SUDARSHAN, 2012, p. 167).

Para modelar um determinado negócio, você precisa entender com um grau razoável de detalhes do que se trata esse negócio.

A modelagem entidade relacionamento é uma técnica usada para descrever o entendimento compartilhado das necessidades de informação de um negócio. É uma técnica bem estabelecida que resulta em diagramas bastante fáceis de ler e, portanto, fáceis de verificar.

## 3.1 Definições das convenções

📺 Vídeo



A abordagem E-R foi definida pelo Dr. Peter Chen em 1976. Em seu artigo *The entity-relationship model: toward a unified view of data*. *ACM Transactions on Database Systems*, Chen propôs uma forma de representar graficamente como os dados se relacionam, tendo como base a teoria proposta pelo Dr. Codd em 1970. A notação de Chen pode ser considerada como uma base para outros padrões que surgiram posteriormente (HEUSER, 2008).

## 3.2 Componentes do diagrama entidade relacionamento (DER)

📺 Vídeo



A notação do DER adotada nesta obra para representar graficamente o modelo de dados é chamada de *diagrama de Barker*. Ao analisarmos diferentes tipos de notações E-R, é difícil não encontrarmos a notação de Barker, que é comumente utilizada para descrever dados para o SGBDR Oracle.

O DER é composto de:

- entidades;
- atributos;
- relações.

### + Saiba mais

Para ler o artigo *Um Modelo de Dados Relacional para Grandes Bancos de Dados Compartilhados* escrito pelo Dr. Edgar Frank Codd, acesse: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>.

O artigo intitulado *Modelo entidade relacionamento: uma visão unificada de dados*, escrito pelo Dr. Peter Chen, pode ser encontrado em: <https://dspace.mit.edu/bitstream/handle/1721.1/47432/entityrelationshx00chen.pdf>.

### 3.2.1 Entidade

De acordo com Rob e Coronel (2014), uma entidade é um objeto que irá armazenar os dados significantes para um determinado negócio.

Para definirmos uma entidade, devemos considerar que, ao fazer a identificação dos requisitos de dados durante a construção do modelo conceitual, foi gerada uma série de textos, anotações e que você recebeu uma série de *prints* de algumas telas.

Geralmente os substantivos que você encontra no material gerado anteriormente, que têm relação com o negócio, dão um excelente indicio de que ele representa uma entidade.

Em seguida, tente encontrar um nome único e claro para a entidade. Isso nem sempre é fácil, pois há muito mais conceitos do que nomes claros. Use sua imaginação, um dicionário ou uma combinação de palavras, mas não deixe que a falta de um bom nome o impeça de modelar.

Verifique os nomes que você identificou. A definição implícita de uma entidade pode mudar durante a análise, por exemplo, como resultado da adição de um atributo ou da alteração da opcionalidade de um relacionamento.

Crie uma descrição formal da entidade. Geralmente isso não é difícil, e a escrita ajuda a esclarecer o seu pensamento em relação ao que você está falando. Verifique essa descrição regularmente, pois, às vezes, os conceitos evoluem durante o processo de modelagem. As definições devem seguir essa evolução.

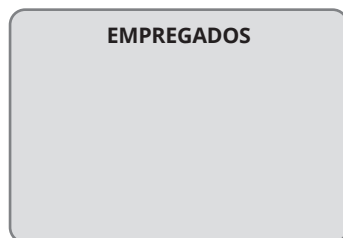
Em muitos contextos de negócios, o mesmo conceito é conhecido por nomes diferentes. Selecione um, e mencione os sinônimos na descrição, por exemplo: “professores também conhecidos como instrutores”.

Como exemplo de entidades podemos citar:

- PESSOAS;
- PEDIDOS;
- ITENS;
- ALUNOS;
- EMPREGADOS;
- PACIENTES.

No diagrama de Barker, uma entidade é representada por uma caixa retangular, conforme o exemplo a seguir:

 **Figura 1**  
Entidade “EMPREGADOS”



Fonte: Elaborada pelo autor.

#### ? Curiosidade

Os projetistas Richard Barker, Ian Palmer e Harry Ellis desenvolveram a notação E-R na época em que eles trabalhavam na empresa de consultoria britânica CACI por volta do ano de 1981. Quando Barker se juntou à Oracle, a sua notação foi adotada.

#### ! Atenção

Embora você seja livre para usar qualquer nome que desejar para uma entidade é recomendável evitar nomes que contenham caracteres especiais como acentuação, espaço etc., pois quando você for implementar o modelo físico no SGBDR isso poderá acarretar problemas.

Uma entidade pode ser classificada como:

- **Forte:** é entidade que envia a chave primária para a entidade fraca/dependente.
- **Fraca/dependente:** é a entidade que recebe a chave primária da entidade forte. Essa chave é chamada de chave estrangeira (*foreign key*)
- **Associativa/composta:** é uma entidade utilizada para resolver uma relação com o grau de muitos para muitos.


## 3.2.2 Atributo

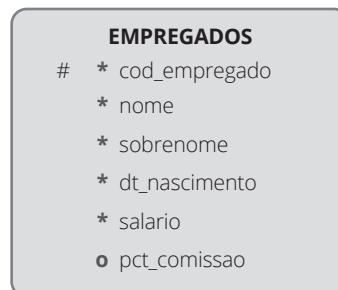
Os atributos descrevem, quantificam, qualificam, classificam, especificam ou fornecem um status da entidade a que pertencem. Definimos um atributo como uma propriedade de uma entidade.

Para cada atributo, há um conjunto de valores chamado de *domínio* ou *conjunto de valores de determinado atributo* (SILBERSCHATZ; KORTH; SUDARSHAN, 2012, p. 162).

De acordo com Heuser (2009), os atributos permitem a caracterização das propriedades de uma entidade. Normalmente, você pode reconhecer facilmente os atributos fazendo as perguntas como “de quê?” e “de qual formato?”.

Na notação de Baker, os atributos aparecem dentro da caixa retangular que representa a entidade. Vejamos o exemplo a seguir de uma entidade e seus respectivos atributos.

 **Figura 2**  
Entidade e atributos EMPREGADOS



Fonte: Elaborada pelo autor.

Podemos interpretar que os qualificadores de um empregado, nesse caso, são: **cod\_empregado**, **nome**, **sobrenome**, **dt\_nascimento**, **sa-**

lario e pct\_comissao. A representação dos símbolos (#), (\*) e (o) será detalhada nos próximos tópicos.

### 3.2.2.1 Chave primária (*primary key*)/identificador único (*unique identifier*)

Imagine uma situação em que podemos ter empregados com o mesmo nome ou sobrenome. Como identificar o empregado corretamente?



Tabela 1

Tabela EMPREGADOS sem chave primária

EMPREGADOS			
nome	sobrenome	data_nascimento	salario
Maria	Paula	27/02/2001	9500
Sérgio	Mário	12/03/1982	7500
Pedro	Henrique	12/03/1990	3500
Mário	Sérgio	12/04/1982	7500
Mário	Paula	01/03/1997	9000
Ana	Paula	27/02/2004	9500

Fonte: Elaborada pelo autor.

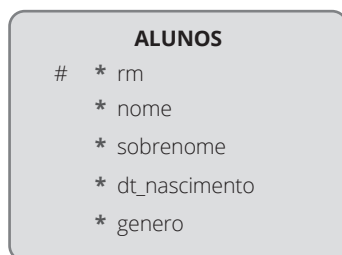
Para garantirmos que cada uma das linhas da tabela de empregados seja única/atômica/exclusiva, nós devemos criar um atributo que seja único (*unique*) e obrigatório (*not null*). Esse atributo, com as duas características, é conhecido formalmente no projeto lógico como *identificador único* (UID – *unique identifier*). Ele pode ser chamado também de *chave primária* (PK – *primary key*) no projeto lógico.

O exemplo a seguir apresenta a entidade ALUNOS e seus respectivos atributos:



Figura 3

Entidade ALUNOS



Fonte: Elaborada pelo autor.

A chave primária da entidade ALUNOS é o atributo RM, representado por (#\*), pois ele permite a identificação exclusiva de cada ocorrência.



#### Atenção

Assim como as entidades, recomenda-se não utilizar caracteres especiais, como acentuação, espaço etc., para nomear os atributos de uma entidade.



#### Dica

Como sugestão, nomeie os atributos no singular.

cia da entidade alunos. Chamamos de *ocorrências* o conjunto de dados que uma entidade representa que irá armazenar.

Em alguns casos, devemos ter uma chave primária composta. Vejamos o exemplo a seguir que demonstra essa necessidade:



**Tabela 2**

Tabela CLIENTES com chave primária composta

CLIENTES				
rg	oe	nome	sobrenome	data_nascimento
17299384	SSP-SP	Henrique	Maia	27/02/2001
17299384	SSP-PR	Patrícia	Cunha	12/03/1982
28304838	SSP-SP	Matheus	Pires	12/03/1990
32848493	SSP-RJ	Fernanda	Sampaio	12/04/1982
39503030	SSP-MA	Mária	Rita	01/03/1997
39395030	SSP-DF	André	Cruz	27/02/2004

Fonte: Elaborada pelo autor.

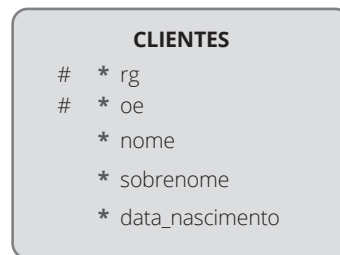
Como o sistema que emite RG (Registro Geral) não é unificado podem ocorrer casos em que o número do RG irá repetir em estados diferentes. Nesse caso, a chave primária é composta pelos atributos RG e OE (órgão emissor).

Na representação a seguir, da entidade CLIENTES, a chave primária composta ficará da seguinte forma:



**Figura 4**

Entidade CLIENTES



Fonte: Elaborada pelo autor.

### ! Atenção

Uma entidade só pode ter uma chave primária, portanto a chave primária pode ser simples ou composta.

Portanto, o SGBDR irá considerar, para fins de validação, o dado composto 17299384SSP-SP e 17299384SSP-PR, garantido dessa forma a atomicidade de cada uma das ocorrências.



### 3.2.2.2 Atributo obrigatório/opcional/único

Um atributo pode assumir diversas características. Todo atributo que não é um UID pode ser:

- único (*unique*);
- obrigatório (*not null*);
- opcional (*null*).

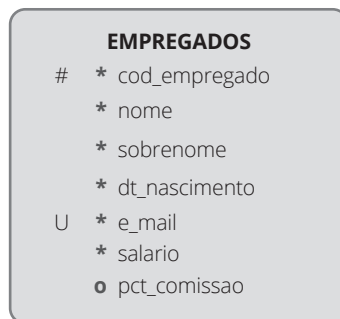
O que determina a característica de um atributo é a regra de negócio que estamos modelando. Vale salientar que um atributo único pode ser definido como obrigatório ou opcional.

Vejamos o exemplo a seguir. A entidade EMPREGADOS apresenta atributos com várias características:



Figura 5

Entidade EMPREGADOS – atributos único/obrigatório/opcional



Fonte: Elaborada pelo autor.

O atributo e-mail deve ser único (**U**) e obrigatório (**\***), o que significa que todo empregado deve ter um e-mail e ele não pode se repetir.

Os atributos nome, sobrenome, dt\_nascimento e salario, identificados por (**\***), devem ser conhecidos, portanto eles são obrigatórios.

E, por fim, o atributo pct\_comissao, identificado por (**o**), pode ser omitido, o que significa que nem todo empregado é comissionado.

## 3.2.3 Relacionamentos

Um relacionamento é uma associação entre entidades, que indica opcionalidade, grau ou cardinalidade.

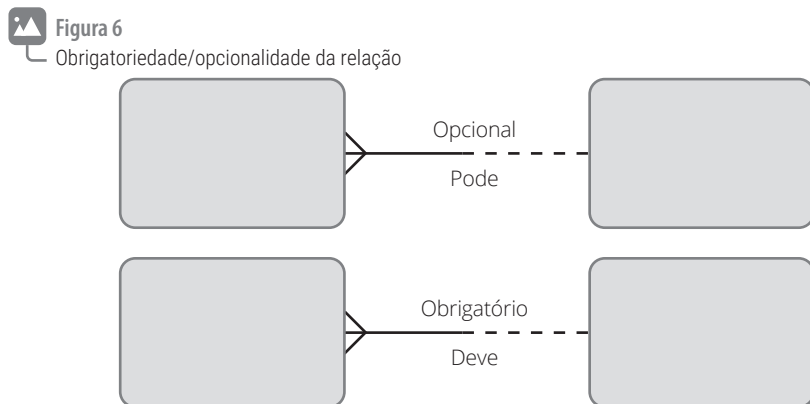
As entidades se relacionam por meio do compartilhamento de um atributo (valor em uma coluna) (ROB; CORONEL, 2014, p. 41).

Para determinarmos se existe uma relação entre as entidades, devemos perguntar, para cada uma de suas entidades, se ela está de alguma forma relacionada a uma ou mais entidades e, em caso afirmativo, devemos ligar (associar) as entidades por meio de uma linha em que o estilo da linha irá determinar a opcionalidade.

Uma linha sólida indica que o relacionamento entre as entidades é obrigatório. Já uma linha pontilhada indica que o relacionamento entre as entidades é opcional.

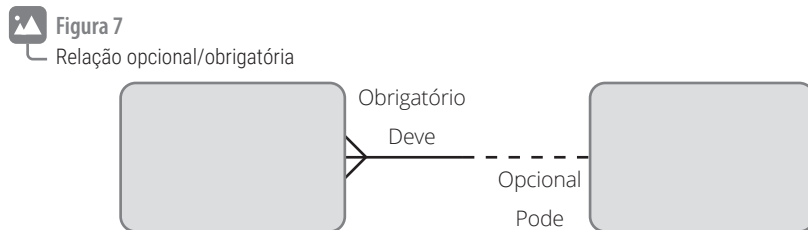
A opcionalidade é diferenciada da obrigatoriedade por meio da utilização das palavras *deve* e *pode*. A palavra *deve* indica que um relacionamento é obrigatório, e a palavra “*pode*” indica opcionalidade.

Vejamos o exemplo a seguir, que apresenta os dois tipos de associação:



Fonte: Elaborada pelo autor.

Em alguns casos, metade da linha pode ser sólida, e a outra metade pontilhada. Isso indica que um relacionamento é obrigatório para uma entidade (A) e opcional para a entidade (B). Vejamos o exemplo a seguir, que apresenta a relação obrigatória em um dos lados e opcional para o outro:



Fonte: Elaborada pelo autor.

Normalmente todas as entidades em um modelo estão relacionadas a pelo menos uma outra entidade. As exceções são raras, mas existem.

Duas entidades podem se relacionar mais de uma vez. Por exemplo, no sistema de correio eletrônico existem duas relações entre as entidades MENSAGENS e USUARIOS, em que uma relação é sobre quem envia MENSAGENS, e a outra é sobre quem recebe MENSAGENS.

Uma entidade pode se relacionar com ela mesma. Esse tipo de relação é chamada de *relacionamento recursivo*. Por exemplo, uma MENSAGEM pode ser uma resposta a outra MENSAGEM.

Em um modelo ER, o estilo do final da linha do relacionamento indica grau. Um relacionamento pode apresentar um grau ou cardinalidade de:

- um-para-um (1:1);
- um-para-muitos (1:N);
- muitos-para-muitos (N:N).

### 3.2.3.1 Relação de um para um – 1:1

Quando, entre duas entidades, temos uma linha única que as associa isso significa que só pode haver uma e somente uma ocorrência de cada uma das entidades.

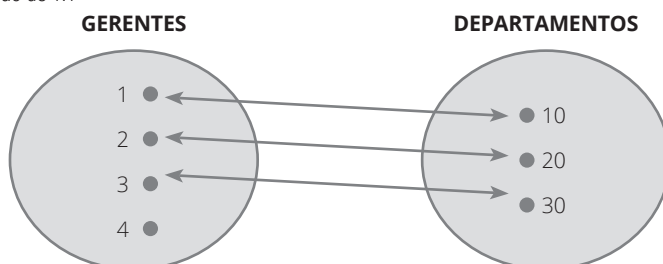
Por exemplo, considere que temos uma série de departamentos na empresa e que esses departamentos devem ser gerenciados por um, e somente um, gerente. Por outro lado, um gerente só pode gerenciar um departamento. Portanto, vamos considerar para essa relação as entidades GERENTES e DEPARTAMENTOS.

Vejamos como essa relação é representada conforme a ilustração a seguir:



Figura 8

Relação de 1:1



(Continua)



Dica

Como sugestão nomeie os atributos no singular.

GERENTES		
cod_ger	nome	sobrenome
1	Marcos	Pena
2	Rodrigo	Alvarez
3	Cinthia	Padro
4	Heleno	Magalhães

DEPARTAMENTOS		
cod_depto	nome	cod_ger
10	TI	1
20	RH	2
30	ENG	3

Fonte: Elaborada pelo autor.

Podemos concluir que:

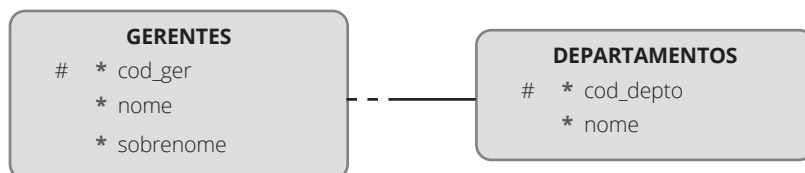
- O gerente Marcos é gerente do departamento de TI.
- O gerente Rodrigo é gerente do departamento de RH.
- A gerente Cinthia é gerente do departamento de ENG.
- O gerente Heleno não gerencia nenhum departamento.

Vejamos como ficará a representação das entidades relacionadas, conforme a seguir:



Figura 9

Relação de 1:1 entre as entidades GERENTES e DEPARTAMENTOS



Fonte: Elaborada pelo autor.

Portanto, conforme o problema de negócio analisado, ao lado da entidade gerentes a linha é tracejada, pois um gerente pode gerenciar um departamento, que é o caso do funcionário chamado *Heleno*.

Já um departamento deve ser gerenciado por um somente, logo a linha deve ser sólida ao lado da entidade departamentos.

### 3.2.3.2 Relação de um para muitos – 1:N

Nesse tipo de relação, iremos notar a existência de um símbolo representado por três linhas que se estendem da linha de relacionamento em direção à entidade. Essa situação indica um relacionamento de grau de muitos para um. Esse símbolo é chamado de *pé-de-galinha*.

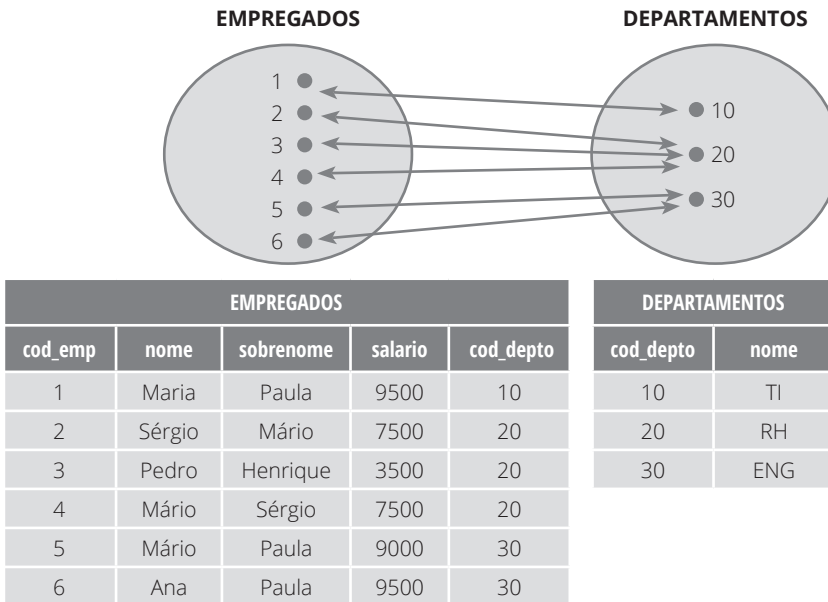
Vamos assumir, como exemplo, que cada empregado deve ser alocado para um, e somente um, departamento e que cada departamento pode ter muitos empregados alocados.

Vejamos na ilustração a seguir a representação desse tipo de relação:



Figura 10

Relação entre EMPREGADOS e DEPARTAMENTOS



Fonte: Elaborada pelo autor.

Podemos concluir que nesse caso:

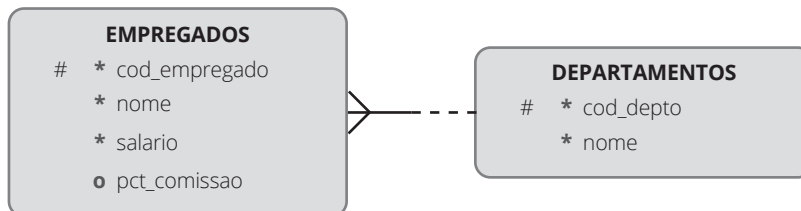
- A funcionária Maria trabalha no departamento de TI.
- Os funcionários Sérgio, Pedro e Mário Sérgio trabalham no departamento de RH.
- Os funcionários Mário Paula e Ana Paula trabalham no departamento de ENG.

Para essa regra de negócio, a relação entre as entidades EMPREGADOS e DEPARTAMENTOS ficará da seguinte forma:



Figura 11

Relação de 1:N entre as entidades EMPREGADOS e DEPARTAMENTOS



Fonte: Elaborada pelo autor.

### ! Atenção

No modelo lógico, a chave estrangeira não é representada no DER. Esse conceito bem como o conceito de relação identificada serão abordados no próximo capítulo, que irá tratar sobre a normalização.

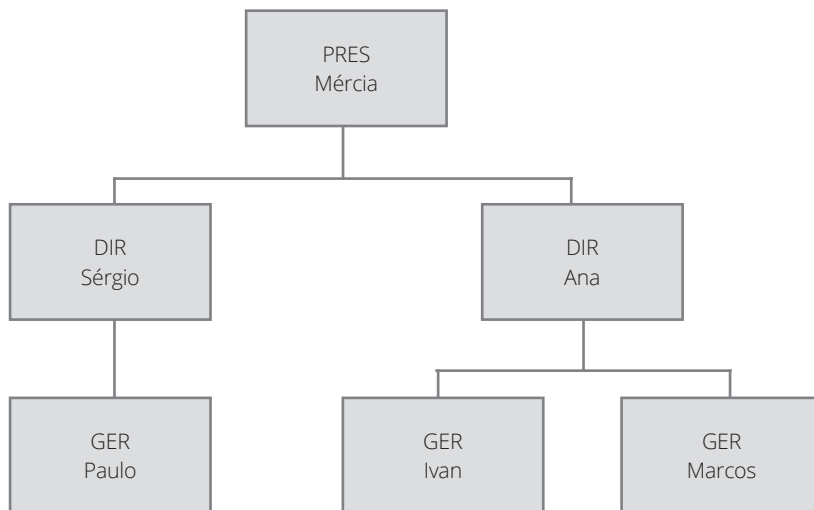
Portanto, para esse problema de negócio, o pé-de-galinha fica ao lado da entidade EMPREGADOS, e a linha é sólida, pois cada empregado deve ser alocado para um departamento. Nesse caso, podemos ter departamentos vazios.

Uma relação de 1:N recursiva pode ser implementada para resolver questões relacionadas à hierarquia de cargos. Vamos considerar que temos na nossa empresa os cargos de presidente, diretores e gerentes, conforme o diagrama a seguir:



Figura 12

Diagrama hierárquico



Fonte: Elaborada pelo autor.

Se construirmos uma tabela com alguns dados e implementarmos essa hierarquia, ela terá a seguinte estrutura:



Tabela 3

Tabela de empregados que se relacionam

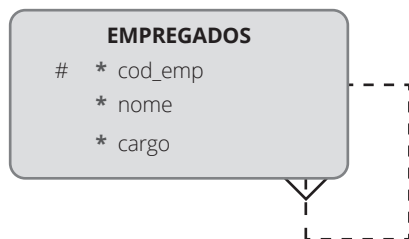
EMPREGADOS			
cod_emp	nome	cargo	cod_sup
10	Mércia	PRES	
20	Sérgio	DIR	10
30	Ana	DIR	10
40	Paulo	GER	20
50	Ivan	GER	30
60	Marcos	GER	30

Fonte: Elaborada pelo autor.

Ao realizar a leitura da tabela anterior, podemos notar que a funcionária Mércia não se relaciona com nenhum outro funcionário, pois ela ocupa o maior cargo dentro da hierarquia e, portanto, não há valor na coluna `cod_sup`. Se observamos a linha 2, em que temos o diretor Sérgio, vemos que o código do superior dele é igual a 10. Portanto, se a presidente tem o código de empregado 10, o funcionário 20 é subordinado a ela. As outras linhas são interpretadas de maneira análoga. A coluna `cod_sup` é a chave estrangeira que permite a implementação da recursividade. Lembre-se de que o conceito de chave estrangeira será estudado posteriormente.

A entidade EMPREGADOS, com a implementação da recursividade, ficará conforme a figura a seguir:

 **Figura 13**  
Relação hierárquica de empregados



Fonte: Elaborada pelo autor.

Uma relação recursiva que implementa a hierarquia deve ser sempre opcional em ambas as direções, lembre-se de que toda hierarquia deve ser finita.

### 3.2.3.3 Relação de muitos para muitos – N:N

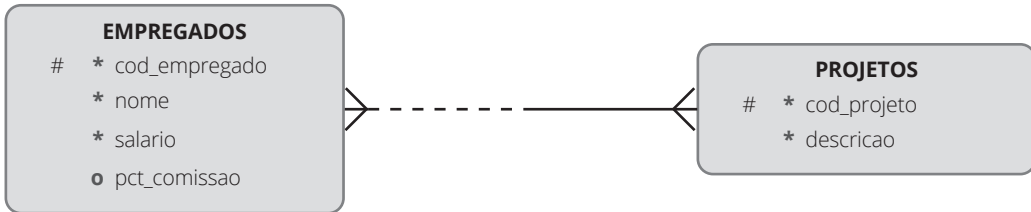
Nesse tipo de relação, teremos uma linha com um pé-de-galinha em cada extremidade da linha que associa as entidades. Isso significa que uma ou mais entidades podem estar relacionadas a várias entidades.

Uma relação N:N não pode ser implementada fisicamente, pois ela só representa a regra de negócio. A resolução desse tipo de relação será estudada no Capítulo 4.

Para representarmos, no modelo lógico, a relação de N:N, vamos tomar como exemplo a relação em que cada empregado pode participar de muitos projetos e que cada projeto deve ter a participação de um ou mais (muitos) empregados.

Portanto, o diagrama lógico que representa a associação de muitos para muitos entre duas entidades ficará da seguinte forma:

 **Figura 14**  
Relação de N:N entre as entidades EMPREGADOS e PROJETOS




Fonte: Elaborada pelo autor.

## 3.3 Realizando a leitura do diagrama entidade relacionamento

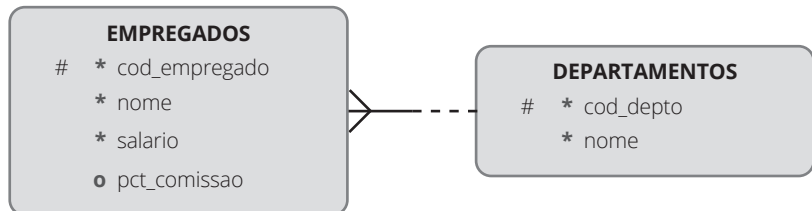
 Vídeo



Para realizar a leitura de um modelo E-R, utilizamos a seguinte sintaxe, conforme a ilustração a seguir:

 **Figura 15**  
Sintaxe para a leitura do diagrama entidade relacionamento

Cada entidade de origem {pode ser | deve ser} nome de relacionamento  
{um e somente um | uma ou mais} entidade de destino.



Cada empregado deve ser alocado em um, e somente um, departamento.  
Cada departamento pode alocar um ou mais empregados.

Fonte: Elaborada pelo autor.

No exemplo anterior, o modelo indica que cada funcionário deve ser designado a apenas um departamento. Por outro lado, todo departamento deve ser composto de um ou mais funcionários.

Uma relação deve ser sempre lida em duas direções.

Na sintaxe, o símbolo | indica uma propriedade mutuamente exclusiva, ou seja, um ou outro, nunca ambos.



### Case 1 – Modelando um banco de dados para o módulo de RH de uma empresa

Baseando-se na nota a seguir devemos construir o diagrama entidade relacionamento – DER:

Em uma grande empresa, o departamento de recursos humanos é responsável por controlar diversas informações sobre os empregados e sobre as instalações da empresa.

Para cada empregado, o RH armazena dados como o número de identificação, nome completo, e-mail, código de identificação do cargo, salário, data de contratação e telefone. Alguns empregados, dependendo do cargo, podem ser comissionados.

Cada um dos cargos possui um código de identificação, título do cargo e faixa salarial mínima e máxima.

A nossa empresa atua em diversas regiões, como na América do Norte, América Latina, Europa e Ásia, sendo, portanto, necessário armazenar as informações dos diversos departamentos e armazéns existentes nas diversas localidades em que atuamos.

Todo empregado pode ser alocado a um departamento. Os departamentos são identificados por um número e por um nome. Cada departamento está associado a um local e cada um dos locais possui um endereço completo que inclui o nome da rua, CEP, cidade e estado.

Para cada departamento, podemos ter um empregado (gerente) responsável por ele.

Nos locais onde os departamentos e depósitos estão localizados, a empresa registra detalhes como o nome do país.

Para cada um dos nossos funcionários, devemos conhecer pelo menos um contato de emergência obrigatoriamente. Para cada contato, devemos conhecer o nome completo, grau de parentesco e os telefones residencial, celular e comercial. Quando um funcionário é deletado da nossa base, os seus contatos são automaticamente deletados.

Devemos identificar a quem um determinado funcionário se subordina e, consequentemente, devemos saber quantos subordinados um funcionário que exerce a função de chefia tem.

Para isso, devemos seguir os seguintes passos:

1. Identificar as entidades (substantivos).
2. Identificar os atributos das entidades (qualificar as entidades).
3. Relacionar as entidades (verbos).
4. Construir o DER.

(Continua)

### ! Atenção

Para construir o DER será utilizada a ferramenta Oracle Developer Data Modeler.

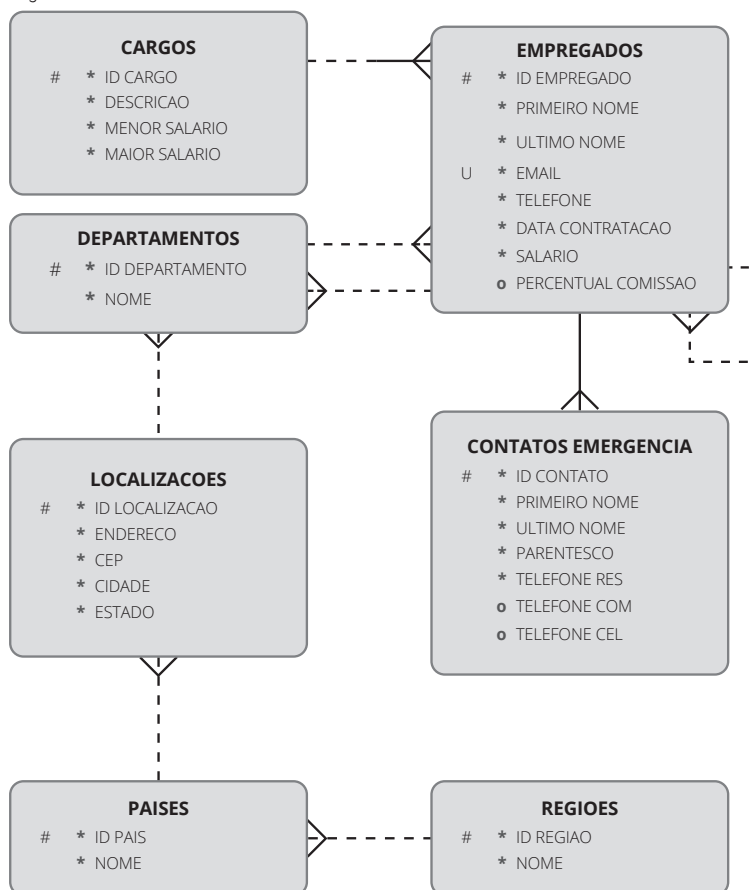
Para fazer o download e para consultar a documentação, acesse o link: <https://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html>.

A ferramenta é gratuita e está disponível para os sistemas operacionais Windows, Linux e MacOS.



Figura 16

Diagrama entidade relacionamento do módulo de RH



Fonte: Elaborada pelo autor.



## CONCLUSÃO

Nesse capítulo, vimos que antes de construirmos o modelo lógico é necessário entender os detalhes dos requisitos de dados por meio da interpretação do modelo conceitual.

O projeto lógico é composto por entidades, atributos e relações. Um atributo pode apresentar as seguintes características:

- Ser uma chave primária.
- Ser único.
- Ser obrigatório ou opcional.

Para relacionarmos as entidades, precisamos associá-las. Uma relação pode ser obrigatória ou opcional, e toda relação apresenta um grau ou cardinalidade que pode ser:

- Grau de um para um (1:1).
- Grau de um para muitos (1:N).
- Grau de muitos para muitos (N:N).

Chamamos de recursividade uma tabela que se relaciona com ela mesma.



## ATIVIDADES



### Atividade 1



Classifique os itens a seguir com (E) para entidade (A) para atributo e (R) para relacionamento:

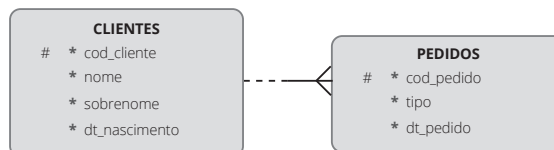
- ( ) PACIENTES
- ( ) VALOR\_UNITARIO
- ( ) COMPRAR
- ( ) ENVIAR
- ( ) TELEFONE
- ( ) CPF
- ( ) ITENS
- ( ) ALUNOS
- ( ) PRODUTOS
- ( ) ITEM\_PEDIDO
- ( ) DATA\_NASCIMENTO
- ( ) CARGOS
- ( ) PROFISSOES
- ( ) NOME\_COMPLETO
- ( ) CEP



### Atividade 2



Escreva as sentenças que representam a leitura do diagrama entidade relacionamento apresentado a seguir utilizando para isso a sintaxe apropriada para a sua leitura:





### Atividade 3



Considerando que: “cada cliente pode realizar muitos pagamentos e que cada pagamento deve ser realizado por um e somente um cliente” construa do DER utilizando o modelo de Barker.

Como atributos considere:

Entidade: CLIENTES – Atributos: (cod\_cli, nome, sobrenome, dt\_nascimento).

Entidade: PAGAMENTOS – Atributos: (cod\_pagto, forma, dt\_pagto).

Todos os atributos são obrigatórios. Lembre-se de definir o atributo que será a chave primária.



## REFERÊNCIAS

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Sagra Luzzatto, 2009. (Série Livros Didáticos da UFRGS).

SILBERSCHATZ, A.; KORTH, H F.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Elsevier, 2012.

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e gerenciamento*. São Paulo: Cengage Learning, 2014.

# 4

## Organizando o projeto de banco de dados – normalização



### Objetivos de aprendizagem

Com o estudo deste capítulo, você será capaz de:

- definir e utilizar a relação identificada;
- normalizar o diagrama entidade-relacionamento (DER);
- resolver relações de muitos para muitos;
- utilizar a recursividade para resolver problemas de hierarquia;
- criar o modelo entidade-relacionamento (MER).

A construção de um diagrama entidade-relacionamento objetiva atender às necessidades de negócio identificadas no projeto conceitual de banco de dados.

Para Korth, Silverschatz e Sudarshan (2012), não é necessário aplicar a normalização quando o DER é construído corretamente. No entanto, é possível que haja em uma entidade atributos que dependam funcionalmente de outros atributos. Por exemplo, considere que temos uma entidade empregados com os atributos `cod_emp`, `nome`, `data_nascimento`, `codigo_departamento` e `nome_departamento`. Como há uma dependência funcional entre os atributos `codigo_departamento` e `nome_departamento`, essa entidade deve ser normalizada.

Dizemos que um atributo é dependente funcionalmente de outro quando um atributo B é dependente, de modo totalmente funcional, do atributo A se cada valor de A determina um, e somente um, valor de B (Rob; Coronel, 2014, p. 168). Por exemplo, se considerarmos a entidade CARGOS com os atributos `cod_cargo` e `descricao_cargo` podemos dizer que, nesse caso, o atributo `cod_cargo` é chamado de

*atributo determinante* e o atributo *descricao\_cargo* é denominado *atributo dependente*, pois é preciso definir um valor para o código do cargo para que ele seja conhecido.

De acordo com Rob e Coronel (2014), é denominado *normalização* o processo utilizado para avaliar e corrigir as estruturas e tabelas que visam eliminar as redundâncias e reduzir as prováveis anomalias. Portanto, para normalizarmos um diagrama lógico, devemos aplicar as chamadas *regras de formas normais* (FN), em que cada uma das formas deve ser aplicada em todas as entidades do modelo.

## 4.1 Relação identificada

▶ Vídeo

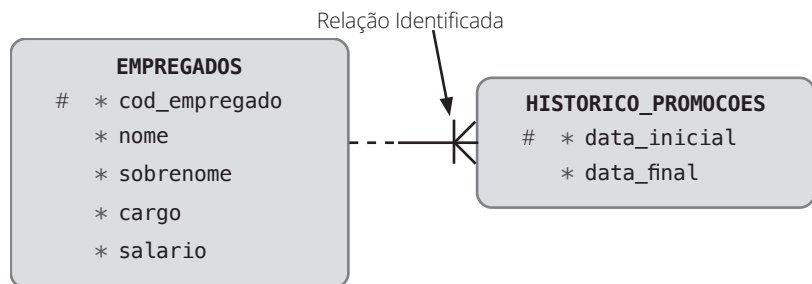


Uma relação identificada é aquela que transforma uma chave estrangeira em uma chave primária. No diagrama de Barker, esse tipo de relação é representada por uma barra vertical no fim do relacionamento, conforme podemos verificar na Figura 1 a identificação do relacionamento da entidade HISTÓRICO\_PROMOÇÕES.



Figura 1

Relação identificada – chave primária composta



Fonte: Elaborada pelo autor.

### ! Atenção

O atributo que é chave estrangeira não é exibido no diagrama lógico. Contudo, devemos entender que **sempre** a chave estrangeira **está na entidade que contém o pé de galinha**; logo, na **entidade fraca** da relação.

No diagrama relacional, que estudaremos posteriormente, todo atributo que é chave estrangeira será exibido.

A chave primária da entidade HISTORICO\_PROMOCOES é composta pelos atributos *cod\_empregado* e *data\_inicial*. Nesse caso, se a relação não fosse identificada, só haveria a possibilidade de promover um funcionário por dia, visto que não poderíamos duplicar a data.

## 4.2 Formas normais – definições



Formas normais é um conjunto de regras – proposto inicialmente por E. F. Codd em seu artigo *Further Normalization of the Data Base Relational Model* – aplicadas em todas as entidades do modelo com o objetivo de testar cada uma delas.

Na visão de Heuser (2009), a decomposição até a 3ª FN é suficiente para obter o esquema de um banco de dados correspondente ao projeto conceitual. Na literatura, aparecem outras formas normais, como a forma normal de Boyce-Codd (BCNF), a 4ª FN e a 5ª FN, também conhecida como forma normal de junção de projeção (FNJP).

Vale destacarmos que vamos discutir aqui somente a 1ª FN, a 2ª FN e a 3ª FN.

O quadro a seguir apresenta um resumo sobre as formas normais.



Quadro 1

Resumo das formas normais

Regras de formas normais	Descrição
Primeira forma normal	Toda ocorrência de uma entidade deve ser identificada de maneira exclusiva (UID).
Segunda forma normal	Um atributo não UID deve depender integralmente de um UID composto.
Terceira forma normal	Todo atributo que não é UID deve depender sempre do UID.

Fonte: Elaborado pelo autor.



### Leitura

Você pode encontrar o artigo *Further Normalization of the Data Base Relational Model* no link a seguir.

Disponível em: <https://forum.thethirdmanifesto.com/wp-content/uploads/asgarosforum/987737/00-efc-further-normalization.pdf>. Acesso em: 23 nov. 2021.

Existe uma classe de restrições ainda mais gerais que leva a uma forma normal chamada de *forma normal de chave de domínio* (FNCD) (KORTH; SILVERSCHATZ; SUDARSHAN, 2012, p. 221).

### 4.2.1 Primeira forma normal – 1ª FN

Na primeira forma normal, todos os atributos devem ter um valor único. Para isso, cada um deles deve ser validado para, assim, identificarmos se apresenta um único valor para cada ocorrência da entidade.

Caso o atributo tenha vários valores, deverá ser criada uma entidade adicional, que deve se relacionar com entidade original por meio de um relacionamento N:1.

Vamos tomar como exemplo uma entidade chamada **ALUNOS** com os atributos **rm**, **nome**, **sobrenome**, **data\_nascimento**, **nome\_pai** e **nome\_mãe**.

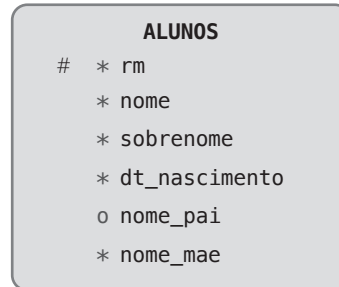
1

Em um modelo relacional, nós nunca eliminaremos os dados repetidos, eles sempre existirão, mas serão controlados (chave estrangeira).



Figura 2

Entidade ALUNOS com grupos de repetição – Anomalia de 1ª FN



Fonte: Elaborada pelo autor.

Para fins de entendimento, vamos gerar algumas ocorrências para a entidade **ALUNOS** conforme o quadro a seguir.



Quadro 2

Ocorrências da entidade **ALUNOS**

Alunos					
rm	nome	sobrenome	dt_nascimento	nome_pai	nome_mae
1	Ivan	Cunha	24/03/1997	Sandro	Patrícia
2	Pedro	Pires	12/09/1999	Alexandre	Maria
3	André	Alvarez	21/03/2002	Lúcio	Marília
4	Patrícia	Andrade	29/02/2004	Sandro	Patrícia
5	Ivone	Pádua	31/01/2000	Sandro	Carla
6	Emílio	Feitosa	13/07/1996	Newton	Carla

Fonte: Elaborado pelo autor.

Perceba que, há grupos de repetição nos nomes do pai e da mãe dos alunos. Se considerarmos, por exemplo, que devemos identificar cada um dos pais dos respectivos alunos, podemos perguntar se Sandro e Patrícia são pais do aluno Ivan Cunha ou da aluna Patrícia Andrade?

Ainda, podemos perguntar se os alunos Patrícia e Ivan são irmãos e podemos ter a necessidade de saber se Carla é a mãe da aluna Ivone ou do aluno Emílio?

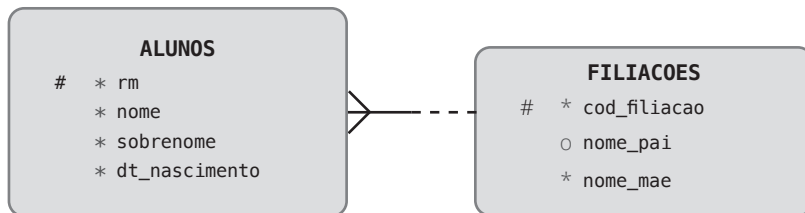


Nesse caso, a normalização ficará da seguinte forma:



Figura 3

Criação da entidade FILIACOES para a eliminação dos grupos de repetição



Fonte: Elaborada pelo autor.

Desse modo, para eliminarmos o grupo de repetição, devemos criar para esse caso a entidade FILIACOES, que se relaciona com a entidade ALUNOS no grau de 1:N, pois cada aluno deve ter uma filiação; e cada um dos PAIS pode ter um ou mais filhos.

Vejamos agora como ficaram as ocorrências em entidades separadas após a normalização:



Quadro 3

Ocorrências da entidade ALUNOS normalizada

Alunos				
rm	nome	sobrenome	dt_nascimento	cod_filiacao
1	Ivan	Cunha	24/03/1997	1
2	Pedro	Pires	12/09/1999	2
3	André	Alvarez	21/03/2002	3
4	Patrícia	Andrade	29/02/2004	1
5	Ivone	Pádua	31/01/2000	4
6	Emílio	Feitosa	13/07/1996	5

Fonte: Elaborado pelo autor.

As ocorrências da antiga entidade ALUNOS geraram ocorrências na entidade FILIACOES.



Quadro 4

Ocorrências da entidade FILIACOES normalizada

FILIACOES		
cod_filiacao	nome_pai	nome_mae
1	Sandro	Patrícia
2	Alexandre	Maria
3	Lúcio	Marília
4	Sandro	Carla
5	Newton	Carla

Fonte: Elaborado pelo autor.

Podemos concluir após a normalização que:

- o casal de pais cujo o código é 1 é Sandro e Patrícia e que eles são os pais de Ivan e Patrícia;
- a mãe que se chama Carla e o pai que se chama Sandro, cujo código é 4, são os pais da aluna Ivone;
- a mãe que se chama Carla, que é casada com Newton, é mãe do aluno Emílio.

Logo, podemos afirmar que as entidades ALUNOS e FILIAÇÕES estão normalizadas porque atendem a 1ª FN.

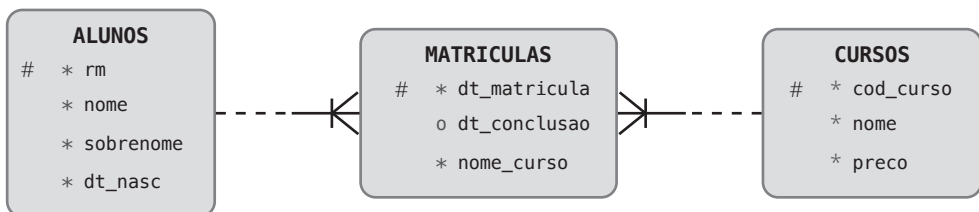
## 4.2.2 Segunda forma normal – 2ª FN

Uma entidade está na segunda forma normal se o atributo não UID depender integralmente do UID composto. Assim, essa regra determina que o atributo não chave não pode depender parcialmente do UID composto.

Para isso, devemos validar se um atributo não depende apenas de parte (dependência parcial) do UID da entidade. Se isso acontecer, então o atributo dependente deve ser movido para outra entidade ou, em alguns casos, o UID não deve ser composto.

A Figura 4 apresenta uma anomalia de 2º FN na entidade MATRICULAS. Vejamos o exemplo:

 **Figura 4**  
Entidade associativa (MATRICULAS) com anomalia de 2ª FN



Fonte: Elaborada pelo autor.

Nesse exemplo, temos um UID composto formado pela combinação dos atributos rm e cod\_curso. Os atributos dt\_matricula e dt\_formacao dependem tanto do atributo cod\_curso quanto do atributo rm, portanto são dependentes integralmente do UID composto.

O atributo nome\_curso apresenta dependência parcial da chave composta. Ou seja, dado o código do curso, podemos identificar o nome do curso, independentemente de conhecermos o atributo rm.

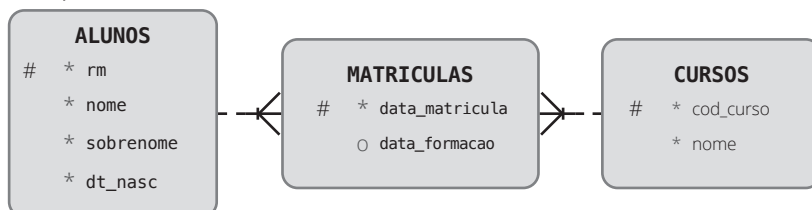
Dessa forma, temos um atributo com dependência parcial do UID composto e, por isso, dizemos que a entidade MATRICULAS atende à segunda forma normal.

Nesse caso, basta retirar o atributo nome\_curso da entidade MATRICULAS; o resultado da normalização é o apresentado na figura a seguir.



Figura 5

Resolução da anomalia de 2ª FN na entidade MATRICULAS



Fonte: Elaborada pelo autor.

Por isso, após a remoção do atributo nome\_curso da entidade MATRICULAS, podemos afirmar que o modelo está normalizado, uma vez que as entidades ALUNOS e CURSOS também estão normalizadas, pois ambas atendem a 1ª FN.

## 4.2.3 Terceira forma normal – 3ª FN

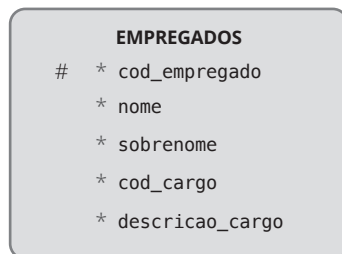
Para que uma entidade esteja na terceira forma normal, um atributo que não é UID deve depender sempre do atributo UID da sua entidade. Um atributo que não é UID nunca deve depender de outro atributo que não é UID. Quando esse evento ocorre, dizemos que a entidade não atende à terceira forma normal.

Considere como exemplo a entidade EMPREGADOS, que apresenta os seguintes atributos:



Figura 6

Entidade EMPREGADOS com anomalia de 3ª FN



Fonte: Elaborada pelo autor.

2

A distinção entre a segunda e a terceira forma normal, que veremos logo em seguida, muitas vezes é confusa.

A segunda forma normal está ligada à ocorrência de UID compostos.

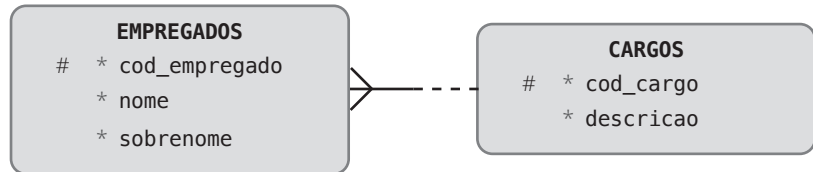
Ao verificar o atributo `descricao_cargo`, podemos notar que ele não tem dependência do UID e, consequentemente, podemos afirmar que a entidade não atende à terceira forma normal.

Para normalizar essa entidade, devemos dividi-la de acordo com a figura apresentada a seguir.



Figura 7

Resolução da anomalia de 3ª FN da entidade EMPREGADOS



Fonte: Elaborada pelo autor.

## 4.3 Resolvendo a relação de muitos para muitos – N:N

Vídeo



Toda relação de grau muitos para muitos deve ser resolvida, visto que esse tipo de relação, no modelo lógico, objetiva expressar o verdadeiro sentido do negócio. Para resolvermos esse tipo de relação, devemos criar uma entidade chamada *entidade interseção* ou *entidade associativa*, que substituirá o relacionamento N:N. Se a relação N:N não for resolvida, será impossível de implementá-la fisicamente.

Para isso, devemos seguir os seguintes passos:

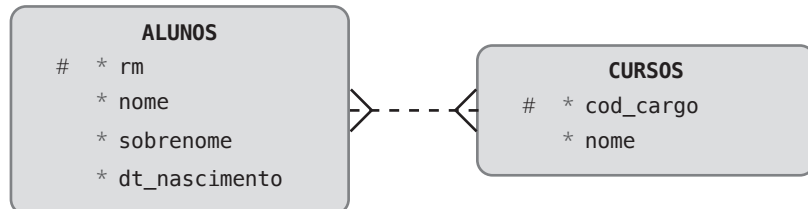
1. Primeiro, criar uma entidade associativa. Você perceberá que, às vezes, não há uma palavra adequada disponível para nomear essa entidade de acordo com o conceito que está modelando. A nova entidade sempre pode ser nomeada com o neologismo **COMBINAÇÃO A/B**, ou um nome que é de alguma forma derivado do nome da relação N:N original.
2. Na sequência, criar dois relacionamentos identificados de grau N:1 da **COMBINAÇÃO A/B** da entidade, um para A e um para B. Caso as relações não sejam identificadas, deveremos criar um atributo UID na entidade associativa.
3. Na próxima etapa, devemos remover o relacionamento N:N.

Vamos considerar a seguinte situação: cada aluno pode se matricular em muitos cursos e podemos ter matriculados muitos alunos em cada curso, portanto o DER ficará da seguinte forma:



**Figura 8**

Relação N:N entre ALUNOS e CURSOS



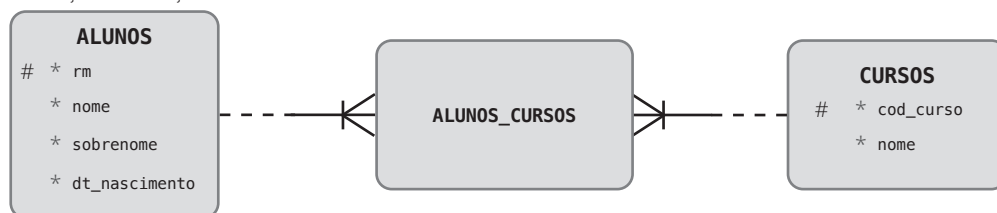
Fonte: Elaborada pelo autor.

A resolução dessa relação N:N será resolvida da seguinte forma:



**Figura 9**

Resolução 1 da relação N:N entre ALUNOS e CURSOS



Fonte: Elaborada pelo autor.

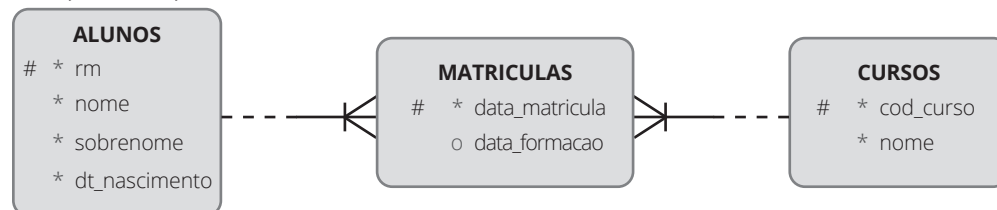
Note que, como as duas relações são identificadas, podemos concluir que a chave primária da entidade **ALUNOS\_CURSOS** é composta pelos atributos **RM** e **COD\_CURSO**.

Se precisarmos determinar a data de matrícula e a data de formação dos nossos alunos, uma segunda solução ficaria da seguinte forma:



**Figura 10**

Resolução 2 da relação N:N entre ALUNOS e CURSOS

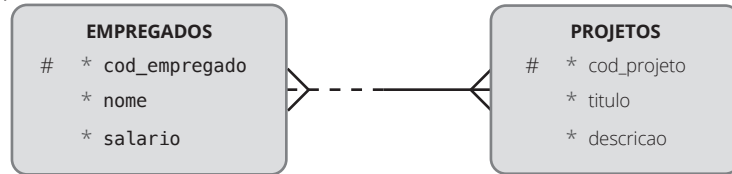


Fonte: Elaborada pelo autor.

Note que, para que haja melhor entendimento do negócio a entidade **ALUNOS\_CURSOS** foi renomeada para **MATRICULAS**.

Vejamos mais um exemplo em que consideraremos a seguinte situação: cada empregado pode ser alocado em muitos projetos, e cada projeto deve ter a alocação de um ou mais empregados.

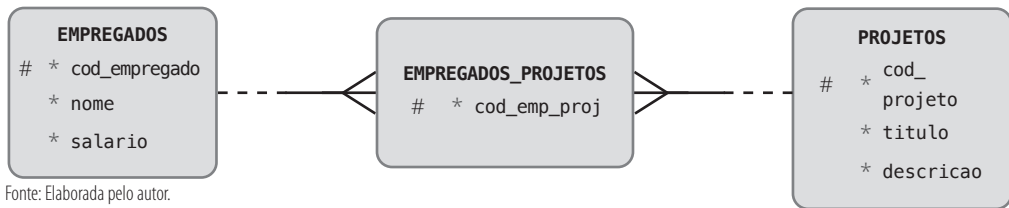
**Figura 11**  
Relação N:N entre EMPREGADOS e PROJETOS



Fonte: Elaborada pelo autor.

A resolução desse caso resultará na criação da entidade associativa EMPREGADOS\_PROJETOS e ficará da seguinte forma:

**Figura 12**  
Resolução da relação N:N entre EMPREGADOS e PROJETOS



Fonte: Elaborada pelo autor.

Note que, na solução apresentada anteriormente, as relações não foram identificadas, portanto tivemos que criar um atributo UID na entidade associativa (EMPREGADOS\_PROJETOS). O que determinará ou não a criação do UID é a regra de negócio.

## 4.4 Criação do modelo entidade-relacionamento (MER)

📺 Vídeo



A criação do modelo entidade-relacionamento pode ser entendida como a transformação do modelo lógico para o modelo relacional. Nesse momento, o modelo objetiva apresentar as informações técnicas, ou seja, será possível implementar e visualizar no modelo as restrições (constraints):

- chave primária – primary key (PK);
- obrigatórias – not null (NN);

- chave única – unique key (UQ);
- Chave estrangeira: foreign key (FK);
- Verificação – check (CK).

Também serão definidos os tipos de dados (datatype) e a precisão de cada uma das colunas. Nesse diagrama, as entidades serão transformadas em tabelas e os atributos serão transformados em colunas.

Portanto, podemos afirmar que o diagrama relacional é resultado do diagrama lógico.

### Estudo de caso

#### **Case 2: modelagem de banco de dados de um estabelecimento de ensino – realização de matrículas e controle de uma biblioteca.**

Com base na descrição a seguir, devemos construir:

o diagrama entidade-relacionamento (modelo lógico);

o modelo entidade-relacionamento (modelo físico/relacional).

Nossa instituição de ensino oferece diversos cursos de informática nos períodos matutino, vespertino e noturno. Para isso, precisamos realizar o controle de matrícula dos nossos alunos.

Desse modo, devemos identificar em qual curso e período o aluno se matriculou, mais especificamente precisamos das datas de matrícula e término do curso.

Cada um dos nossos alunos pode realizar diversos cursos, desde que sejam realizados em períodos distintos. Logo, devemos conhecer o nome completo do aluno e sua data de nascimento. Nossos cursos são identificados por um código, nome, descrição e duração.

Nós temos um acervo de material didático que está disponível para empréstimo para os alunos. Um aluno só pode retirar uma obra com o mesmo título por vez. Realizamos o controle dos nossos empréstimos, pois cada aluno só pode ficar com um livro por até dez dias. Ao término desse prazo, entramos em contato com o aluno.

Mantemos um histórico dos nossos empréstimos e temos um cadastro de autores, isso porque nossos alunos costumam pesquisar os livros de determinado autor.

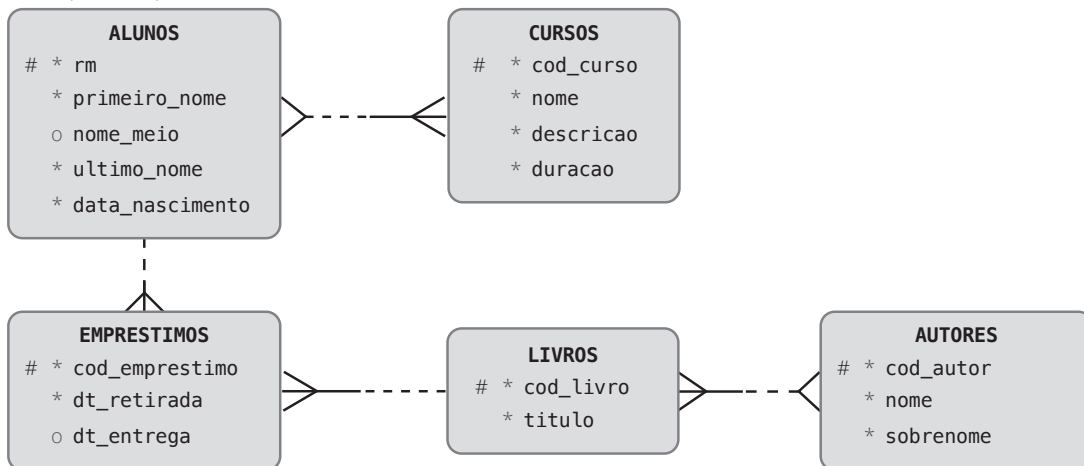
Não controlaremos nesse momento os professores que ministram diversas disciplinas em nossos cursos.

Resolução:

#### **Diagrama entidade-relacionamento**

(Continua)

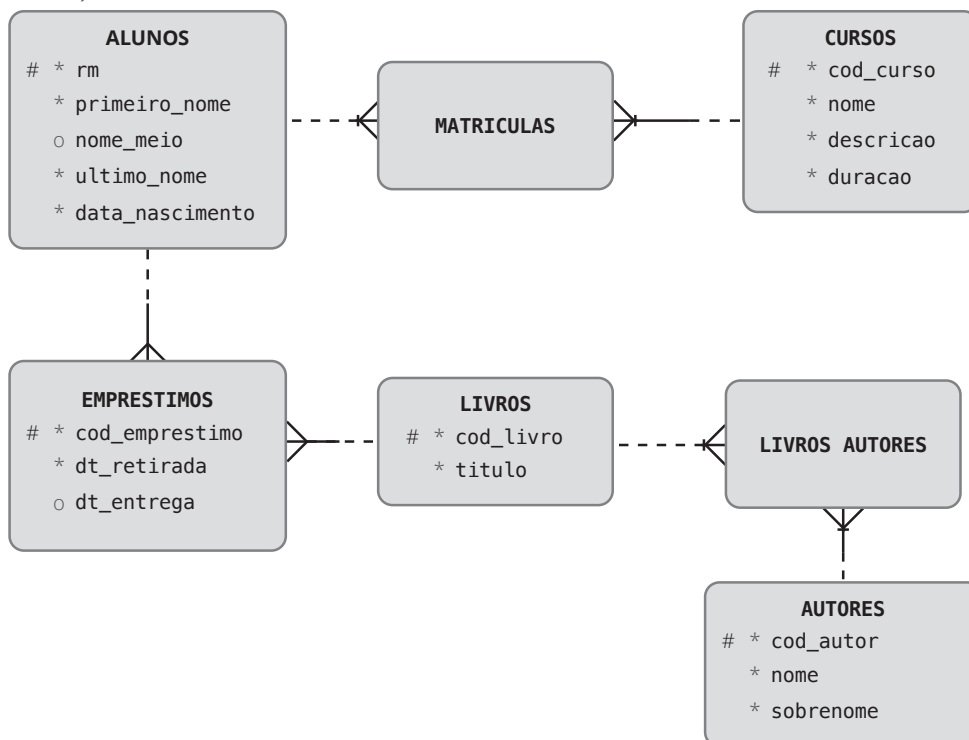
**Figura 13**  
Resolução da relação N:N entre EMPREGADOS e PROJETOS



Fonte: Elaborada pelo autor.

## Resolução das relações de N:N

**Figura 14**  
Relação N:N entre EMPREGADOS e PROJETOS



Fonte: Elaborada pelo autor.

(Continua)

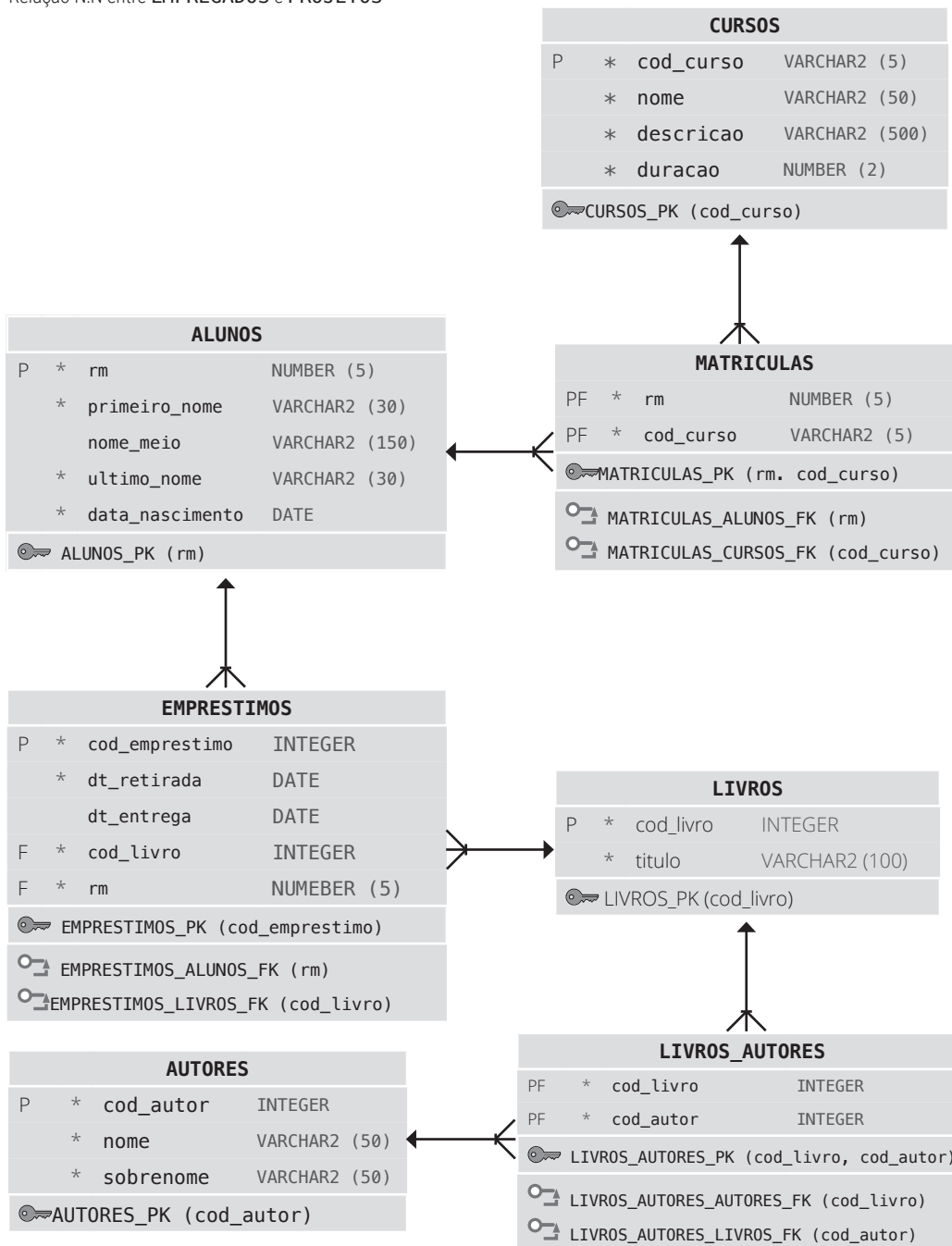


## Modelo entidade-relacionamento



Figura 15

Relação N:N entre EMPREGADOS e PROJETOS



Fonte: Elaborada pelo autor.



## CONCLUSÃO

Neste capítulo, estudamos que normalização é um conceito de banco de dados relacional que objetiva eliminar anomalias do projeto lógico. Se o diagrama entidade-relacionamento for criado corretamente, as tabelas que serão criadas no projeto relacional estarão em conformidade com as regras de normalização.

Toda relação identificada faz com que a chave estrangeira seja uma chave primária ou componha uma chave primária caso ela tenha sido definida na entidade dependente.

As formas normais são as regras que devem ser aplicadas em todas as entidades do modelo com a finalidade de eliminarmos possíveis anomalias.

Analisamos que a primeira, a segunda e a terceira formas normais determinam o seguinte:

- 1ª FN: toda entidade deve ter um UID.
- 2ª FN: quando há um UID composto, todos os atributos não UID devem depender integralmente do UID.
- 3ª FN: um atributo não UID deve sempre depender do atributo UID.

Para resolvermos uma relação de muitos para muitos, devemos criar a entidade associativa. Essa entidade terá sempre uma relação de grau obrigatória e, geralmente, o seu UID é composto pelos UID das entidades de origem.

Por fim, vimos que para construirmos o modelo relacional (modelo entidade-relacionamento), devemos transformar:

- as entidades em tabelas;
- os atributos em colunas;
- os UID em PK;
- as relações em constraint FK;
- definir os tipos de dados e a precisão de todas as colunas.



## ATIVIDADES



### Atividade 1

Defina o que é normalização.



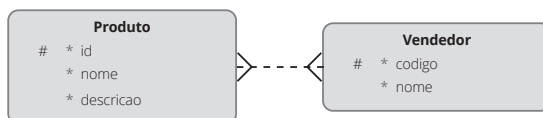
### Atividade 2

Quais são as três primeiras formas normais? Explique cada uma delas de maneira objetiva.



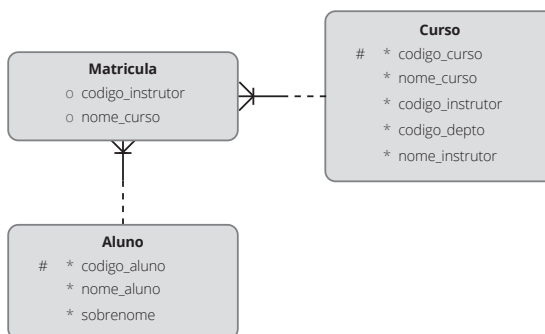
### Atividade 3

Resolva a relação de N:N a seguir.



### Atividade 4

O DER apresentado a seguir não está normalizado. Redesenhe-o produzindo um novo DER normalizado.



## REFERÊNCIAS

HEUSER, C. A. *Projeto de banco de dados*. Porto Alegre: Série Livros Didáticos da UFRGS. 6. ed. Sagra Luzzatto, 2009.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Elsevier, 2012.

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e gerenciamento*. São Paulo: Cengage Learning, 2014.

# 5

## Do projeto físico ao conceito – engenharia reversa



### Objetivos de aprendizagem

Ao final do estudo deste capítulo, você será capaz de:

- identificar cada uma das tabelas no processo de engenharia reversa;
- determinar o tipo de relação (1:N e 1:1);
- identificar as colunas e os tipos de dados;
- definir os identificadores únicos das tabelas (*PK-Primary Key*).

A engenharia reversa de um banco de dados relacional é o processo que cria um modelo de dados conceitual. Ela se baseia em um modelo de implementação (modelo físico) no qual a base pode ser um arquivo *Data Definition Language* (DDL) existente ou podem ser realizadas consultas ao dicionário de dados, que irão permitir identificar os requisitos de informação que foram formalmente capturados na fase conceitual.

Com a engenharia reversa é possível utilizar um produto implementado para que seja possível entender a sua especificação.

### 5.1 Engenharia reversa

▶ Vídeo



Para Rob e Coronel (2014), o processo de engenharia reversa é útil quando não existe um modelo conceitual para um banco de dados que já está implementado. Isso pode ocorrer caso não tenha sido utilizada uma metodologia para a implementação de um projeto de banco de dados.

De acordo com Silverschatz, Korth e Henry (2012), um modelo relacional ou físico de dados é derivado do modelo lógico. Com a utilização da engenharia reversa, o modelo físico será transformado em um modelo lógico.

É importante salientar que é possível realizar a engenharia reversa de qualquer banco de dados relacional, independentemente do fabricante, pois as consultas feitas por meio da utilização da linguagem SQL irão utilizar as tabelas do dicionário de dados.

Uma das partes mais importantes de um banco de dados é seu dicionário de dados. Ele é um conjunto de tabelas, somente de leitura, que fornece informações sobre o banco de dados. Um dicionário de dados armazena basicamente as seguintes informações:

- As definições de todos os objetos de esquema (coleção de estruturas lógicas de dados) no banco de dados (tabelas, visões, índices, clusters, sinônimos, sequências, procedimentos, funções, gatilhos etc.)
- Quanto espaço foi alocado e quanto dele é utilizado atualmente pelos objetos de esquema.
- Valores padrão para colunas.
- Informação de restrição de integridade.
- Os nomes dos usuários.
- Privilégios e funções que cada usuário recebeu.
- Informações de auditoria, como quem acessou ou atualizou vários objetos de esquema.
- Outras informações gerais do banco de dados.

Há uma série de ferramentas utilizadas para fazer engenharia reversa de um banco de dados, como Oracle Data Modelo, ER/Studio, Erwin Data Modeler, IBM Rational Rose Data Modeler, entre outras. Neste capítulo, o exemplo feito será o RDBMS Oracle, e a ferramenta utilizada para a engenharia reversa será o Oracle Data *Modeler*.

Será utilizado como exemplo um banco de dados de um sistema de controle acadêmico, e serão apresentadas as descrições e as restrições (*constraints*) de cada uma das tabelas.



Figura 1

Descrição da tabela PREDIOS

↕ NOME_COLUNA	↕ TIPO_DADOS
COD PREDIO	NUMBER (5,0)
ENDEREÇO	VARCHAR2 (200)

Fonte: Elaborada pelo autor.

 **Figura 2**  
Chave primária da Tabela PREDIOS


↕ CHAVE_PRIMARIA	↕ COLUNAS
PREDIOS PK	COD PREDIO

Fonte: Elaborada pelo autor.

 **Figura 3**  
Restrições da Tabela PREDIOS


↕ NOME_RESTRICAO	↕ TIPO_RESTRICAO
PREDIOS END NN	Not Null

Fonte: Elaborada pelo autor.

 **Figura 4**  
Descrição da Tabela SALAS

↕ NOME_COLUNA	↕ TIPO_DADOS
COD SALA	NUMBER (5,0)
CAPACIDADE	NUMBER (38,0)
COD PREDIO	NUMBER (5,0)

Fonte: Elaborada pelo autor.

 **Figura 5**  
Chave Primária da Tabela SALAS

↕ CHAVE_PRIMARIA	↕ COLUNAS
SALAS PK	COD SALA
SALAS PK	COD PREDIO

Fonte: Elaborada pelo autor.

 **Figura 6**  
Restrições da Tabela SALAS

↕ NOME_RESTRICAO	↕ TIPO_RESTRICAO
SALAS COD SALA NN	Not Null
SALAS CAPACIDADE NN	Not Null
SALAS COD PREDIO NN	Not Null
SALAS PREDIOS FK	Foreign Key

Fonte: Elaborada pelo autor.

 **Figura 7**  
Descrição da Tabela LABORATORIOS

↕ NOME_COLUNA	↕ TIPO_DADOS
EQUIPAMENTO	VARCHAR2 (100)
COD SALA	NUMBER (5,0)
COD PREDIO	NUMBER (5,0)

Fonte: Elaborada pelo autor.

**Figura 8**

Chave Primária da Tabela LABORATORIOS

↕ CHAVE PRIMARIA	↕ COLUNAS
LABORATORIOS PK	COD SALA
LABORATORIOS PK	COD PREDIO

Fonte: Elaborada pelo autor.

**Figura 9**

Restrições da Tabela LABORATORIOS

↕ NOME_RESTRICAO	↕ TIPO_RESTRICAO
LAB EQUIPAMENTO NN	Not Null
LAB COD SALA NN	Not Null
LABS COD PREDIO NN	Not Null
LABORATORIOS SALAS FK	Foreign Key

Fonte: Elaborada pelo autor.

**Figura 10**

Descrição da Tabela TURMAS

↕ NOME_COLUNA	↕ TIPO_DADOS
SIGLA	VARCHAR2 (5)
ANO SEM	NUMBER (38,0)
CAPACIDADE	NUMBER (38,0)
COD SALA	NUMBER (5,0)
COD PREDIO	NUMBER (5,0)
COD DISCIPLINA	NUMBER (5,0)

Fonte: Elaborada pelo autor.

**Figura 11**

Chave Primária da Tabela TURMAS

↕ CHAVE PRIMARIA	↕ COLUNAS
TURMA PK	ANO SEM
TURMA PK	SIGLA

Fonte: Elaborada pelo autor.

**Figura 12**

Restrições da Tabela TURMAS

↕ NOME_RESTRICAO	↕ TIPO_RESTRICAO
TURMAS SIGA NN	Not Null
TURMAS ANO SEM NN	Not Null
TURMAS CAPACIDADE NN	Not Null
TURMAS COD SALA NN	Not Null
TURMAS COD PREDIO NN	Not Null
TURMAS COD DISCIPLINA NN	Not Null
TURMA DISCIPLINAS FK	Foreign Key
TURMA SALAS FK	Foreign Key

Fonte: Elaborada pelo autor.

 **Figura 13**  
Descrição da Tabela DISCIPLINAS

↕ NOME_COLUNA	↕ TIPO_DADOS
COD DISCIPLINA	NUMBER (5,0)
NOME	VARCHAR2 (100)

Fonte: Elaborada pelo autor.

 **Figura 14**  
Chave Primária da Tabela DISCIPLINAS

↕ CHAVE_PRIMARIA	↕ COLUNAS
DISCIPLINAS PK	COD DISCIPLINA

Fonte: Elaborada pelo autor.

 **Figura 15**  
Restrições da Tabela DISCIPLINAS


↕ NOME_RESTRICAO	↕ TIPO_RESTRICAO
DISCIPLINAS NOME NN	Not Null

Fonte: Elaborada pelo autor.

 **Figura 16**  
Descrição da Tabela CURSOS

↕ NOME_COLUNA	↕ TIPO_DADOS
COD CURSO	NUMBER (5,0)
NOME	VARCHAR2 (100)

Fonte: Elaborada pelo autor.

 **Figura 17**  
Chave Primária da Tabela CURSOS

↕ CHAVE_PRIMARIA	↕ COLUNAS
CURSOS PK	COD CURSO

Fonte: Elaborada pelo autor.

 **Figura 18**  
Restrições da Tabela CURSOS

↕ NOME_RESTRICAO	↕ TIPO_RESTRICAO
CURSOS NOME NN	Not Null

Fonte: Elaborada pelo autor.

As informações sobre a descrição, restrições e chaves podem ser obtidas por meio das consultas ao dicionário de dados.

O exemplo a seguir irá consultar as informações sobre a tabela PREDIOS:

```
SELECT
table_name,column_name,data_type,data_length,data_precision
FROM user_tab_columns
```



WHERE table\_name = 'PREDIOS';



Figura 19

Resultado da consulta na tabela user\_tab\_columns

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION
PREDIOS	COD PREDIO	NUMBER	22	5
PREDIOS	ENDERECO	VARCHAR2	200	(null)

Fonte: Elaborada pelo autor.

```
SELECT table_name,index_name,column_name
FROM user_ind_columns
WHERE table_name = 'PREDIOS';
```



Figura 20

Resultado da consulta na tabela user\_ind\_columns

TABLE_NAME	INDEX_NAME	COLUMN_NAME
PREDIOS	PREDIOS_PK	COD_PREDIO

Fonte: Elaborada pelo autor.

```
SELECT constraint_name,constraint_type
FROM user_constraints
WHERE table_name = 'PREDIOS';
```



Figura 21

Resultado da consulta na tabela user\_constraints

CONSTRAINT_NAME	CONSTRAINT_TYPE
PREDIOS END NN	C
PREDIOS PK	P

Fonte: Elaborada pelo autor.

Na próxima seção, iremos detalhar o processo de identificação das tabelas, bem como suas respectivas relações.



#### Importante

Para consultar outras tabelas, basta alterar a condição (WHERE).

## 5.2 Identificando as tabelas



Vídeo



Nessa fase é realizada a correspondência entre as tabelas e o modelo relacional, na qual uma tabela pode corresponder a uma:

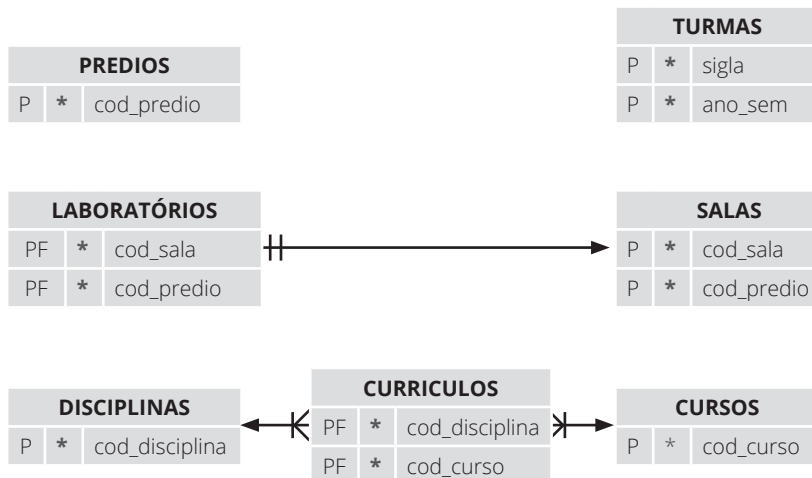
- entidade;
- relacionamento N:N (entidade associativa);
- entidade especializada.

Podemos classificar as tabelas conforme a sua chave primária:

1. Toda tabela que possuir uma chave primária composta de múltiplas chaves estrangeiras resolve uma relação de muitos para muitos. Esse tipo de tabela é chamada de *entidade associativa*. A tabela no modelo com essas características é a tabela CURRICULOS.
2. A chave primária da tabela que é uma chave estrangeira implementa uma especialização da entidade que corresponde à tabela que faz referência à chave estrangeira. No exemplo, a chave primária da tabela LABORATORIO é composta das colunas cod\_sala e cod\_predio, que são chaves estrangeiras.
3. Se não houver uma chave primária composta de múltiplos atributos e uma chave estrangeira, a tabela irá representar uma entidade. As tabelas CURSOS, DISCIPLINAS, PREDIOS E TURMAS. A tabela SALA tem uma chave primária composta apenas de uma chave estrangeira, portanto não se aplica à regra de múltiplas chaves estrangeiras e à regra de uma chave primária da tabela que é uma chave estrangeira.

Após a identificação das tabelas, das chaves e das colunas que não são chaves, o diagrama inicial é apresentado conforme a Figura 22.

 **Figura 22**  
Diagrama inicial



Fonte: Adaptada de Heuser, 2008.

Conforme Heuser (2008), para que ocorra uma relação entre duas tabelas ou com ela mesma (autorrelação), é necessário que a tabela tenha uma chave primária e que cada chave estrangeira seja consistente

com sua chave primária correspondente, ou seja, se existe uma relação entre as tabelas A e B, toda chave estrangeira em B é obrigatoriamente chave primária em A.

- Tabela (*Table*): uma tabela é uma estrutura de armazenamento na qual os dados são organizados e armazenados. Tabelas possuem colunas e linhas. Cada coluna é utilizada para armazenar um tipo específico de valor (Dado). No exemplo da figura 26, a tabela EMPLOYEES é a estrutura utilizada para armazenar informações sobre os funcionários.
- Linha (*Row*): cada linha descreve uma ocorrência de um empregado (registro ou informação). No exemplo da Figura 26, cada linha descreve integralmente todas as propriedades exigidas pelo sistema para que um funcionário seja conhecido.
- Coluna (*Column*): cada coluna contém dados de um tipo específico, como código do funcionário, sobrenome, e-mail e código do departamento no qual o funcionário está alocado.
- Chave Primária (*Primary Key*): a coluna EMPLOYEE\_ID é uma chave primária, ou seja, cada funcionário possui um número de identificação exclusivo nessa tabela, o que distingue cada linha individualmente.
- Chave Exclusiva (*Unique Key*): a coluna e-mail está associada a uma chave exclusiva, ou seja, nesse caso o e-mail é único, pois cada funcionário deve ter o seu e-mail. Portanto, essa restrição não permite que dois funcionários tenham o mesmo e-mail.
- Chave Estrangeira (*Foreign Key*): o valor da coluna de chave estrangeira permite identificar em qual departamento um funcionário específico está alocado.

**Quadro 1**  
Tabela e restrições (PF-FK-UK)

Colunas

	EMPLOYEE_ID	LAST_NAME	E-MAIL	DEPARTMENT_ID
1	100	King	SKING	90
2	101	Kochhar	NKOCHHAR	90
3	102	De Haan	LDEHAAN	90
4	103	Hunold	AHUNOLD	60
5	104	Ernst	BERNST	60
6	105	Austin	DAUSTIN	60
7	106	Pataballa	VPATABAL	60
8	107	Lorentz	DLORENTZ	60
9	108	Greenberg	NGREENBE	100
10	109	Faviet	DFAVIET	100
11	110	Chen	JCHEN	100
12	111	Sciarra	ISCIARRA	100
13	112	Urman	JMURMAN	100
14	113	Popp	LPOPP	100
15	114	Raphaely	DRAPHEAL	30

Linhas

Coluna chave primária  
Primary Key (PK)

Coluna chave única  
Unique Key (UK)

Coluna chave estrangeira  
Foreign Key (FK)

Fonte: Elaborado pelo autor.

Para realizar o mapeamento do modelo físico para o modelo lógico, é necessário mudar os termos que utilizamos em cada um dos modelos. A terminologia que deve ser adotada é apresentada no Quadro 2.

**Quadro 2**  
Mapeamento das terminologias

Projeto físico	Modelo E-R
Tabela	Entidade
Coluna	Atributo
Chave primária	Identificador único primário
Chave única	Identificador único secundário
Chave estrangeira	Relacionamento
Restrições	Regras de negócio

Fonte: Elaborado pelo autor.

- Uma tabela leva a uma entidade.
- Uma coluna se torna um atributo.
- Uma chave primária é um identificador exclusivo.
- Uma chave exclusiva é um identificador exclusivo secundário.
- A chave estrangeira é transformada em uma relação.
- Restrições (*constraints*) são as regras com as quais o banco de dados deve lidar para ser consistente. Algumas das regras de negócios são traduzidas em restrições de verificação (*check constraint*).

Um relacionamento conecta duas entidades. Um relacionamento representa uma dependência significativa de duas entidades.

## 5.3 Identificando as relações 1:N, 1:1, N:N

📺 Vídeo



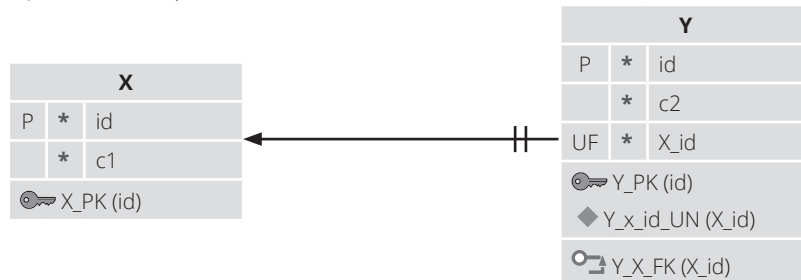
A construção de um diagrama entidade relacionamento tem como objetivo atender às necessidades de negócio identificadas no projeto conceitual de banco de dados.

Ao transformar um relacionamento um-para-um, deverão ser criadas uma chave estrangeira (FK – *Foreign Key*) e uma chave única (UK – *Unique Key*). Todas as colunas dessa chave estrangeira também fazem parte de uma chave única.



Figura 23

Mapeamento de relação 1:1



Fonte: Elaborada pelo autor.

Portanto, conforme apresentado na Figura 23, a relação é obrigatória de um lado e a chave estrangeira é criada na tabela correspondente. Se a relação é obrigatória em ambos os lados ou opcional em ambos os lados, você deve escolher em qual tabela você deseja criar a chave estrangeira.

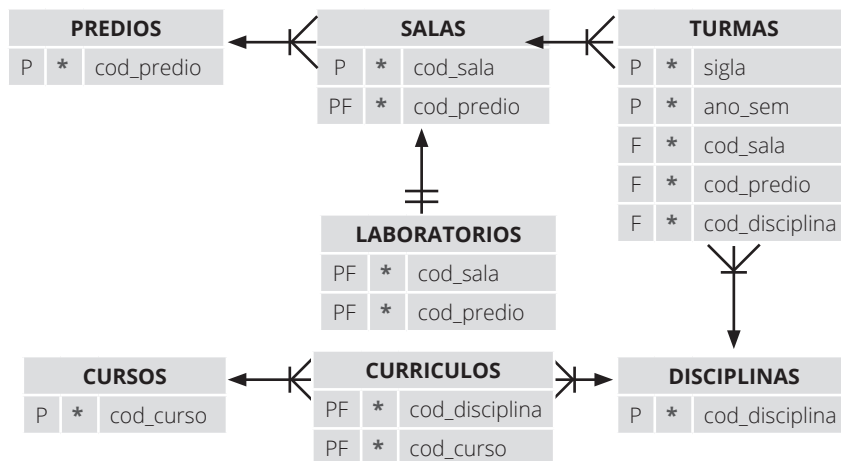
## 5.3.1 Identificando relacionamentos (N:N)

Toda chave estrangeira que não corresponde a uma relação de N:N e que não representa uma especialização é uma relação 1:N ou 1:1. É necessário verificar o conteúdo do banco de dados para determinar quais dos dois tipos de relacionamento foram definidos, caso não ocorra uma relação similar à apresentada na Figura 24.



Figura 24

Identificação das relações de 1:N e 1:1



Fonte: Adaptada de Heuser, 2008.

Como resultado, conforme apresentado na Figura 24, as tabelas que representam relacionamentos 1:N ou 1:1 são as tabelas SALAS, que fazem referência às tabelas PREDIOS e TURMAS, que fazem referência às tabelas DISCIPLINAS e SALAS.

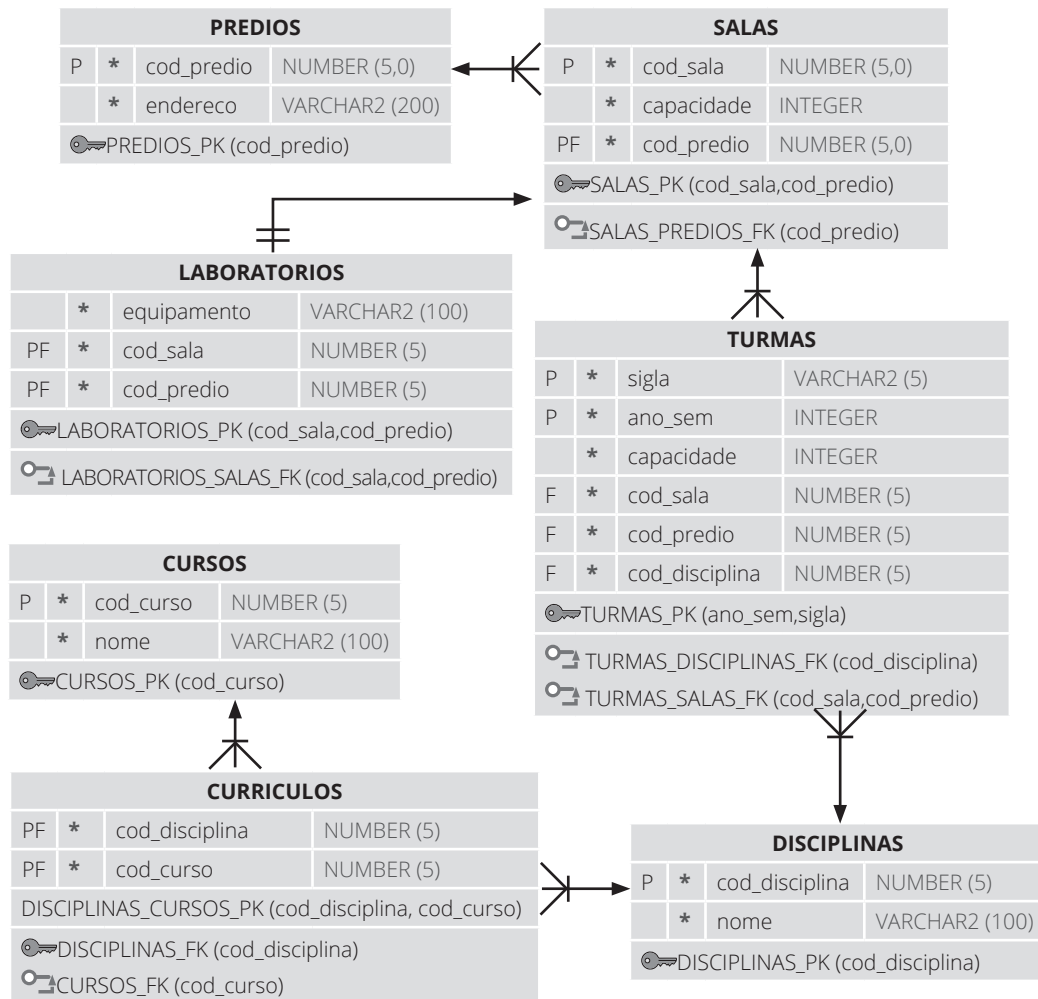
## 5.4 Identificando os atributos e os tipos de dados

Vídeo



Nessa fase serão definidas as colunas que não são chaves, bem como os tipos de dados de cada uma delas. O diagrama final é o apresentado na Figura 25.

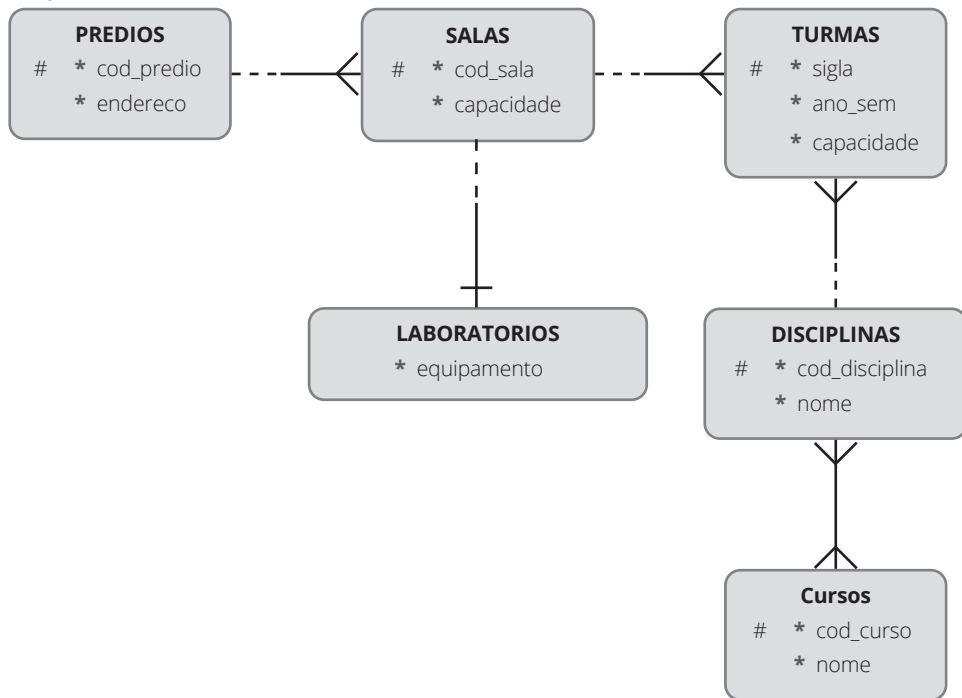
**Figura 25**  
Identificação dos atributos e dos tipos de dados



Fonte: Adaptada de Heuser, 2008.

Com base no diagrama físico, é possível construir o modelo lógico de um banco de dados para um sistema acadêmico, conforme apresentado na Figura 26:

 **Figura 26**  
Modelo lógico



Fonte: Elaborada pelo autor.

Portanto, temos como resultado da engenharia reversa o diagrama entidade relacionamento ou o modelo lógico de banco de dados.



## CONCLUSÃO

Vimos neste capítulo que a engenharia reversa de um banco de dados relacional é o processo que cria um modelo de dados conceitual baseando-se em um modelo de implementação (modelo físico), no qual a base pode ser um arquivo DDL existente ou podem ser realizadas consultas ao dicionário de dados, as quais irão permitir identificar os requisitos de informação que foram formalmente capturados na fase conceitual.

Com a engenharia reversa, é possível utilizar um produto implementado para que seja possível entender a sua especificação.

O processo de engenharia reversa é útil quando não existe um modelo conceitual para um banco de dados que já está implementado. Isso pode ocorrer caso não tenha sido utilizada uma metodologia para a implementação de um projeto de banco de dados.





## ATIVIDADES



### Atividade 1



Defina o que é engenharia reversa.



### Atividade 2



Complete a tabela a seguir utilizando a terminologia correta do projeto físico e do projeto lógico.

Projeto físico	Projeto lógico
Tabela	
	Atributo
	Identificador único primário
Chave estrangeira	
	Regras de negócio



### Atividade 3



O que representa um relacionamento?



## REFERÊNCIAS

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Sagra Luzzatto, 2009. (Série Livros Didáticos da UFRGS).

ROB, P.; CORONEL, C. *Sistemas de banco de dados: projeto, implementação e gerenciamento*. São Paulo: Cengage Learning, 2014.

SILVERSCCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Elsevier; Campus, 2012.

# Resolução das atividades

---

## 1 Introdução ao universo dos dados

### 1. Discuta cada uma das definições a seguir.

#### a) Dado. Cite três exemplos.

Dados não apresentam significado se não forem colocados dentro de um contexto. Exemplos: 33; vendas; presidente.

#### b) Informações. Cite três exemplos.

A informação é elaborada por meio do processamento dos dados. Exemplos: 33 é a idade de Mário; Ivan tem o cargo de presidente; Abel trabalha no departamento de vendas.

#### c) Conhecimento.

Conhecimento é o processo de análise da informação por meio do aprendizado, o qual resulta em conhecimento.

#### d) Sabedoria.

A sabedoria é a utilização do conhecimento para resolver problemas, inovar, melhorar e desvendar.

### 2. O que é um SGBD e quais são as suas principais funções?

SGBD é um software composto de um conjunto de programas responsáveis por manipular os dados quando estes são solicitados pelas aplicações. O SGBD também é responsável por garantir a privacidade, o compartilhamento, a integridade e a segurança dos dados.

### 3. Cite três vantagens e três desvantagens de se utilizar um SGBD.

As vantagens são:

- auxilia no gerenciamento de grandes volumes de dados;
- integra as atividades operacionais e gerenciais;
- permite a execução de tarefas em paralelo.

As desvantagens são:

- necessidade de atualizações e migrações;
- dependência do fornecedor;
- complexidade no gerenciamento e, por consequência, necessidade de mão de obra qualificada para a sua administração.

## 2 As fases do projeto de banco de dados

### 1. Qual é a importância da criação de um projeto de banco de dados?

Se o projeto de banco de dados não for construído, não há como definir e, posteriormente, construir as estruturas que irão armazenar os dados, bem como não há como garantir a consistência dos dados e uma boa performance quando os dados forem manipulados.

### 2. Cite as principais características dos seguintes projetos:

#### a) Conceitual

Nessa fase, levantamos os requisitos de dados, de acordo com o problema de negócio a ser estudado.

#### b) Lógico

Após o entendimento dos requisitos de negócio, iremos representar graficamente as relações entre os dados.

#### c) Físico

Nessa fase, iremos definir os tipos de dados e a precisão de cada um, para que seja possível, por meio da linguagem SQL, criar o banco de dados em um SGBDR.

### 3. Qual é o nome do diagrama que deve ser criado no projeto lógico?

Diagrama entidade-relacionamento (DER).

### 4. Qual é o nome do diagrama que deve ser criado no projeto físico?

Modelo entidade-relacionamento (MER).

## 3 Representando o Entendimento do Negócio - Diagrama Entidade Relacionamento - DER

### 1. Classifique os itens a seguir com (E) para entidade (A) para atributo e (R) para relacionamento:

(E) PACIENTES

(A) VALOR\_UNITARIO

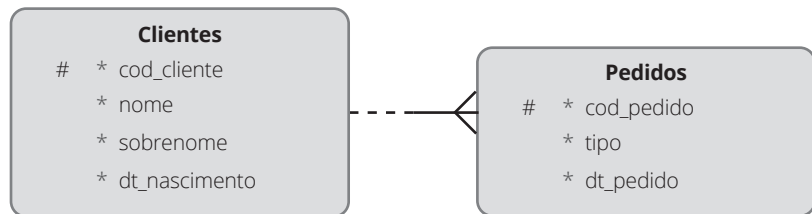
(R) COMPRAR

(R) ENVIAR

(A) TELEFONE

- (A) CPF
- (E) ITENS
- (E) ALUNOS
- (E) PRODUTOS
- (E) ITENS\_PEDIDOS
- (A) DATA\_NASCIMENTO
- (E) CARGOS
- (E) PROFISSOES
- (A) NOME\_COMPLETO
- (A) CEP

2. Escreva as sentenças que representam a leitura do diagrama entidade relacionamento apresentado a seguir utilizando para isso a sintaxe apropriada para a sua leitura:



Cada cliente pode fazer muitos pedidos.

Cada pedido deve ser feito por um e somente um cliente.

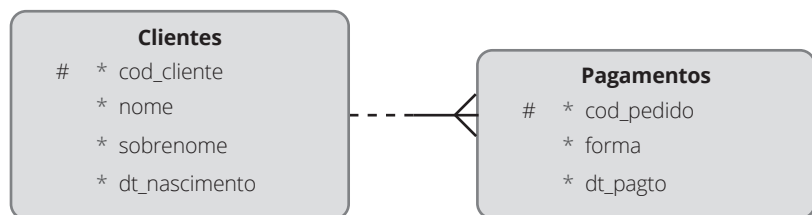
3. Considerando que: “Cada cliente pode realizar muitos pagamentos e que cada pagamento deve ser realizado por um e somente um cliente” construa o DER utilizando o modelo de Barker.

Como atributos considere:

Entidade: CLIENTES – Atributos: (cod\_cli, nome, sobrenome, dt\_nascimento)

Entidade: PAGAMENTOS – Atributos: (cod\_pagto, forma, dt\_pagto)

Todos os atributos são obrigatórios. Lembre-se de definir o atributo que será a chave primária.



## 4 Organizando o projeto de banco de dados – normalização

### 1. Defina o que é normalização.

É um conceito de banco de dados relacional que visa eliminar anomalias do modelo lógico.

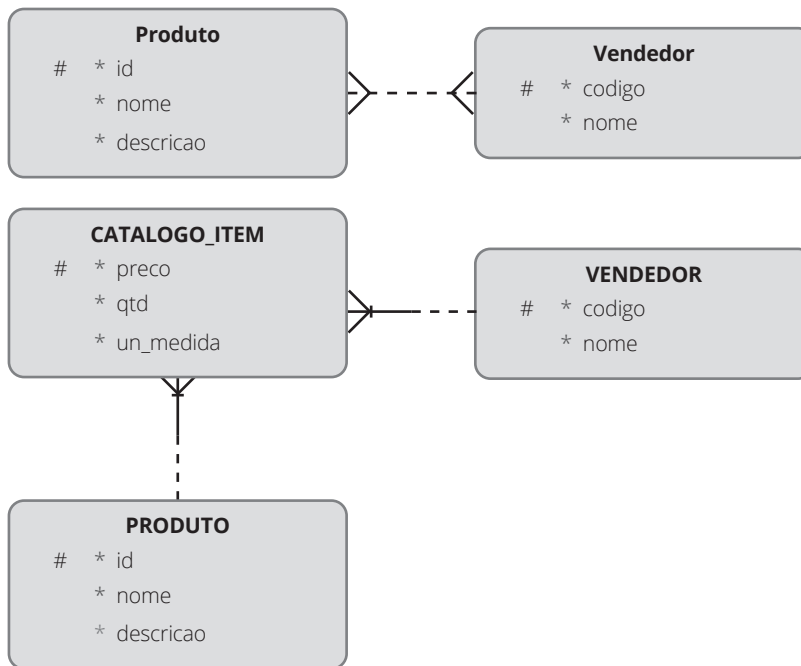
### 2. Quais são as três primeiras formas normais. Explique cada uma delas de maneira objetiva.

1ª FN: definir PK.

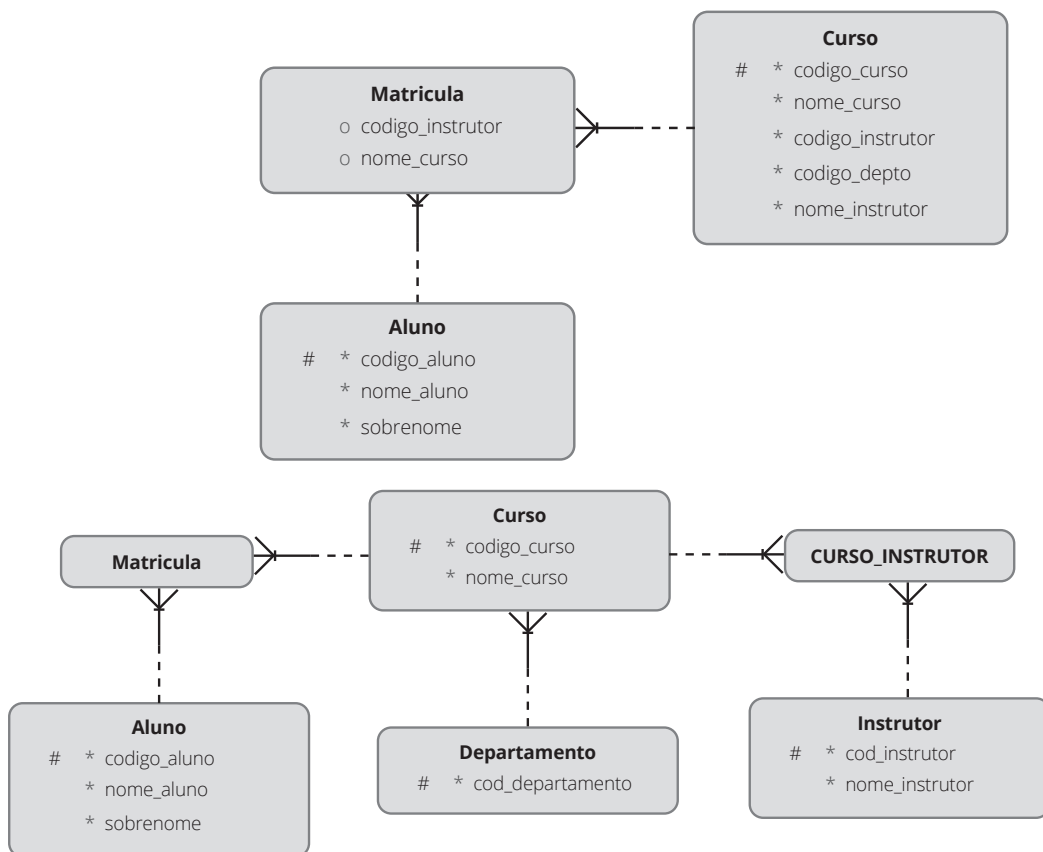
2ª FN: dependência integral da chave composta.

3ª FN: todo atributo não chave depende da chave.

### 3. Resolva a relação de N:N a seguir.



### 4. O DER apresentado a seguir não está normalizado. Redesenhe-o produzindo um novo DER normalizado.



## 5 Do projeto físico ao conceito – engenharia reversa

### 1. Defina o que é engenharia reversa.

É processo de transformação do modelo físico em modelo lógico para que a sua especificação (projeto conceitual) seja entendido.

### 2. Complete a tabela a seguir utilizando a terminologia correta do projeto físico de do projeto lógico.

Projeto Físico	Projeto Lógico
Tabela	ENTIDADE
COLUNA	Atributo
CHAVE PRIMÁRIA	Identificador Único Primário
Chave Estrangeira	RELACIONAMENTO
RESTRIÇÕES	Regras de Negócio

### 3. O que representa um relacionamento?

Ele representa a dependência funcional entre as entidades ou entre ela mesma no caso da relação recursiva.





**IESDE**  
*A Melhor Sempre!*

Código Logístico



1000294

ISBN 978-85-387-6444-1



9 788538 764441