BANCO DE DADOS
MODELAGEM DE DADOS

Autor: Me. Otacílio José Pereira

INICIAR

introdução

Introdução

Bom, vamos começar mais uma trajetória, e se você é daqueles que gostam de analisar cenários, avaliar e representar soluções com esquemas, esta unidade, que trata de Modelagem de Dados, é do que precisa.

No início, daremos uma visão geral mais abrangente acerca de modelagem, e faremos uma introdução sobre o assunto.

Depois, abordaremos a modelagem em si, com entidades e relacionamentos, cuja ênfase será mais próxima do mundo real, com um cenário a resolver. No Modelo Relacional, o foco é materializar a representação do mundo real em um banco de dados relacional, comumente encontrado em nosso dia a dia.

No final, um assunto mais técnico de normalização dos dados será discutido para melhorar a qualidade no trato das informações.

É isso, relaxe a mente, porque é hora de observar o mundo, modelar seus elementos, e transformar nossa análise em um banco de dados!

Introdução à Modelagem de Dados

A implementação de um banco de dados começa com a sua modelagem pelo analista ou projetista de dados, ou, simplesmente, analista. Nesta etapa, com base nos requisitos do sistema a serem tratados, o analista identifica o modo de estruturar entidades, atributos ou campos, e relações, e assim especificar o banco de dados que será implantado para prover as informações aos sistemas e seus usuários.

Uma pergunta que podemos nos fazer é: por que modelar os dados? Para compreender as vantagens de se criar um bom modelo de dados, podemos nos inspirar em outras áreas, por exemplo, na engenharia civil ou na engenharia elétrica/eletrônica. Um engenheiro civil, diante do objetivo de construir um prédio, antes, precisa criar as denominadas plantas, por exemplo, a planta baixa, que apresenta a posição de paredes, portas e etc., e que serve de suporte para as outras plantas, como a elétrica e a hidráulica. Na engenharia elétrica ou eletrônica, antes de os dispositivos eletrônicos serem construídos, precisam ser diagramados. As Figuras 2.1 e 2.2 ilustram alguns desses “modelos”: uma planta baixa e um circuito eletrônico:

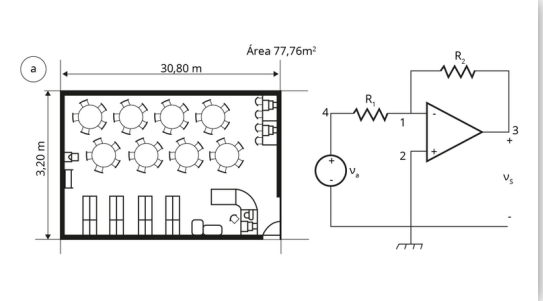


Figura 2.1 – Exemplos de modelos de outras áreas
Fonte: Kowaltovisky (2011, p. 98) e Yaro (2006, p. 35).

Equivalente aos diagramas anteriores para suporte às suas respectivas áreas, no caso da modelagem de dados, o nosso "protagonista" é um modelo relacional. Um exemplar é representado na Figura 2.2. O foco, nesta unidade, será sabermos como construir um desses diagramas, com as suas várias decisões de projeto.

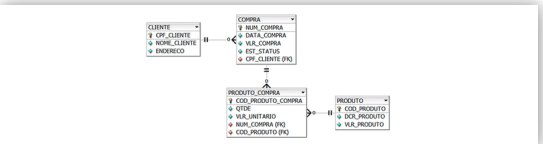


Figura 2.2 – Modelo Relacional de um Sistema de Compras
Fontes: Elaborada pelo autor.

O ato de modelar, nas diversas áreas, permite uma visão preliminar do que será construído e apresenta algumas vantagens no processo de sua construção. Para *software* , em específico, Booch (2012) apresenta uma visão bem interessante e abrangente. As vantagens de uma boa modelagem com os seus diagramas, em em suas diversas áreas e contextos, englobam:

1. **Melhor compreensão do que está sendo concebido** : observe que o diagrama desenhado, no ato de modelar, é uma representação, isto é, expressa o produto final a ser construído. Portanto, durante a sua elaboração, o projetista enfrentará algumas dúvidas, precisará entender melhor situações e tomará decisões para expressar no diagrama. Essas atividades aprofundam a compreensão do projetista quanto ao mundo real que está sendo retratado.
2. **Comunicação entre partes interessadas** : pare e pense um pouco, olhando para um dos esquemas da Figura 2.1: o que é mais fácil de compreender? Aquela Figura 2.1 em si ou um texto que descreveria o

projeto desta maneira: a sala é composta por 8 mesas redondas à direita, 4 escrivaninhas à esquerda e um balcão de atendimento próximo à porta de entrada”? É fácil percebermos que o diagrama apresentará melhor essas informações, e ainda com uma série de outros dados, como medidas da mesa, distância da janela, largura da porta etc. No nosso caso, de trato com dados, o diagrama já expressa suas tabelas, os campos e relacionamentos mais facilmente do que se tratados em texto.

3. **Especificação para construção** : complementando a comunicação, o diagrama da modelagem acaba ajudando a quem depois vai precisar de construir o artefato em questão. Um mestre de obras saberá a largura da janela e a posição na parede, sem a necessidade de o engenheiro estar todo o tempo disponível para esclarecer as dúvidas. Para o nosso caso de dados, o analista não precisa, a todo o momento, dizer quais os campos e tipos de dados da tabela. O desenvolvedor pode consultar o diagrama ou modelo para entender de que modo os dados estão estruturados.
4. **Automatização do processo** : um diagrama bem especificado pode servir para gerar, automaticamente, via ferramentas, outros produtos para a construção do sistema. Por exemplo, diante de um modelo relacional, uma ferramenta já pode gerar todos os comandos de SQL para a criação das tabelas e, ainda, outras ferramentas podem ajudar a gerar outros códigos para a programação do *software* .

reflita

Reflita

Existe um dito popular que diz que “uma figura vale mais do que 1000 palavras!”. Pare e pense em ocasiões, sejam do cotidiano ou em áreas como Engenharia e Tecnologia de

Informação, enumere situações em que um esquema torna muito mais simples e intuitiva a comunicação.

Cenário de Exemplo

Uma forma de desenvolvermos nossa habilidade de modelar é por meio de um exemplo da exploração de um cenário prático. Imagine, portanto, que nós fazemos parte de uma empresa de desenvolvimento de software, e uma loja gostaria de abrir a sua versão virtual para venda on-line na Internet. Em uma conversa com o cliente, surgiu a seguinte descrição do sistema:

“Quando um cliente deseja um produto, ele acessa o site da loja, na internet. Na primeira tela, pode consultar produtos e analisar características como Descrição e Preço. Conforme interesse, adiciona os produtos em seu ‘carrinho de compras’. Ao final do processo de escolha dos produtos, o usuário pode, então, efetivar a compra. Acessa uma seção para conferir o carrinho, depois faz o login no sistema para se autenticar, e daí efetiva o pagamento da compra”.

Diante desse cenário, quais são os produtos finais da modelagem? Uma primeira visão está na versão de Modelo de Entidade-Relacionamento, da Figura 2.3, e o Modelo Relacional do sistema é o diagrama que já foi apresentado na Figura 2.2. Algumas concepções estão abstraídas, e a ideia, neste momento, é termos uma visão dos produtos que construiremos no decorrer do aprendizado.

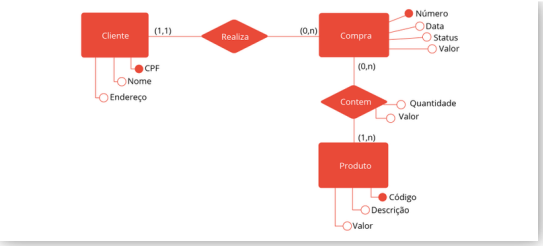


Figura 2.3 – Modelo de Entidade e Relacionamento do Sistema de Compras
Fonte: Elaborada pelo autor.

Modelo Conceitual, Modelo Lógico e Modelo Físico

Nas fases para a implementação de um banco de dados, o nível de abstração ou detalhe varia desde uma visão mais próxima do mundo real e da sua representação conceitual até a forma como os dados serão concretizados no armazenamento em disco no SGBD. Esse caminho, com variação na profundidade, origina a distinção entre Modelos Conceitual, Lógico e Físico, um conteúdo muito comum sobre o tema banco de dados.

Conforme Amadeu (2019), o modelo conceitual de dados representa as informações que existem no contexto do negócio, com maior foco nos processos. Esse modelo utiliza termos e linguagem próprios do negócio mais adequados ao dia a dia do segmento envolvido no projeto. Perceba que, dessa forma, o modelo conceitual é o mais próximo do mundo real, não há ainda preocupações sobre como os dados serão representados por tipos de dados, como inteiro ou varchar. Para exemplificar, em um consultório poderia ser mapeada a existência de alguns médicos disponíveis, compostos de informações como nome e CRM. Esses médicos possuem uma agenda com os dias da semana e horários em que atende e os clientes podem realizar as consultas. Não é necessário, nesse nível de abstração, perceber que nome não pode ter mais do que 50 caracteres. Concentra-se em um nível mais alto de representação dos dados.

No nível lógico, começa a preocupação com a tecnologia com que o modelo conceitual vai “ganhar vida” para a solução. Por exemplo, o tipo de banco de

dados deve ser identificado. É muito comum usarmos os bancos de dados relacionais comerciais. Mas existem alguns outros, como os bancos de dados hierárquicos, em rede e ainda os mais recentes, como os bancos orientados a objetos ou os bancos NOSQL (*not only* SQL). Além do tipo de banco, questões como quais os nomes das entidades e dos campos, agora em uma linguagem mais próxima da programação e menos do mundo real.

Descendo ainda mais o nível de abstração, o projetista precisa definir questões do armazenamento físico dos dados, de como os dados serão implementados em uma solução computacional. De acordo com o tipo de banco de dados, as especificidades, como nomes de tabelas, campos, quais tipos de dados de cada campo e, ainda, restrições de integridade serão codificadas.

A Figura 2.4 permite compreender essa transição entre os modelos com seus respectivos níveis de detalhes. Perceba que o modelo conceitual está mais próximo dos processos de negócio. Na parte lógica e física, os modelos interagem com a aplicação e o modelo externo (parte baixa à direita). Por exemplo, nesse ponto estão os *softwares* clientes, que acessam os dados.

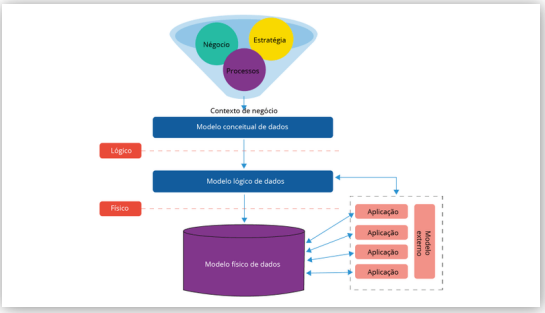


Figura 2.4 – Contextualização dos Modelos Conceitual, Lógico e Físico
Fonte: Amadeu (2019, p. 34).

Em um escritório de uma empresa de desenvolvimento de *software* , um analista de requisitos faz uma reunião com o projetista de dados, depois de uma visita de levantamento inicial a uma indústria para a qual será desenvolvido um *software* grande e complexo. O foco é ter uma visão inicial de como os dados do sistema serão tratados. Situações sobre como a indústria funciona, sobretudo da parte a ser informatizada, foram discutidas por ambos nessa primeira compreensão do sistema. Nesse caso, qual nível de modelo e qual tipo de modelo poderiam oferecer melhor suporte para a reunião?

- ○ **a)** Modelo Conceitual com Modelo de Entidade e Relacionamento.
- ○ **b)** Modelo Físico com Modelo de Entidade e Relacionamento.
- ○ **c)** Modelo Lógico com Modelo Relacional.
- ○ **d)** Modelo Físico com Modelo Relacional.
- ○ **e)** Nenhuma das alternativas está correta.

Modelagem de Entidade e Relacionamento

Em cada um dos níveis de abstração explorados no tópico anterior, um determinado modelo pode ser aplicado, e dois deles são muito comuns: o Modelo de Entidade e Relacionamento, que é muito usado para os níveis de modelo conceitual, e o Modelo Relacional, popularmente empregado para o nível lógico e físico.

Conforme Puga (2013), o Modelo de Entidade-Relacionamento (MER) foi proposto por Peter Chen, nos anos 1970, e visa apresentar uma perspectiva do mundo real como constituído de um conjunto de objetos, entidades que se relacionam entre si. Ainda em Puga (2013), uma entidade representa uma categoria de elementos relevantes para um negócio, podendo ser, por exemplo, os clientes, fornecedores, vendas, contratos, pagamentos, registro de funcionários, entre outros. Sobre esses objetos são realizadas as regras do negócio, como “Cadastrar um cliente” ou “Efetivar uma venda”. Essas entidades, portanto, representam um conjunto de dados que precisam ser armazenados e que são manipulados pelas aplicações que informatizam o negócio.

Uma entidade é formada por atributos que apresentam alguma característica ou informação da entidade. Por exemplo, um cliente pode ser formado pelos atributos como nome, endereço, telefone, data de nascimento e outros. Uma transação financeira em um banco pode ter os atributos de data e hora da ocorrência, o valor e o tipo de transação, de crédito ou débito na conta. Um relacionamento indica uma relação que existe entre as entidades, por exemplo, um cliente realiza uma compra no site ou uma transação financeira é realizada em uma determinada conta corrente. Assim, no primeiro caso, existe um relacionamento entre as entidades cliente e compra e, no segundo caso, transação e conta corrente estão relacionados.

Diagrama de Entidade e Relacionamento

Modelagem e diagrama são dois conceitos intimamente relacionados: a modelagem é o ato de refletir sobre como representar, e o diagrama é o esquema ou desenho que especifica o que está sendo modelado. Muitas vezes, o termo modelo e diagrama são usados indistintamente. Os conceitos de entidades, atributos e relacionamentos são o "coração" do diagrama de entidade e relacionamento (DER), e a Figura 2.5 ilustra uma parte de um diagrama para visualização desses conceitos:

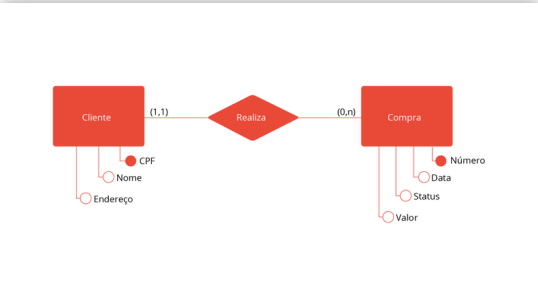


Figura 2.5 – Exemplo de parte de um diagrama de entidade e relacionamento
Fonte: Elaborada pelo autor.

Note que, em texto, o que estamos dizendo é: "um cliente realiza uma compra e é formado pelos atributos CPF, Nome e Endereço, e a entidade compra é composta de Número, Data, Status e Valor". Quando estamos modelando, o que fazemos é o oposto: em geral, temos uma compreensão do mundo real muitas vezes em texto e daí traduzimos para o formato de diagrama. Verificamos também que entidade é desenhada como um retângulo; o

relacionamento, como um losango e os atributos, como círculos, todos eles ligados por linhas. Essa maneira de desenhar os componentes de um diagrama é denominada notação, uma “linguagem” para expressar o que estamos representando, e é útil, pois como todo diagrama DER, ou mesmo um diagrama de outra natureza é desenhado da mesma forma. Qualquer projetista de dados conhecerá a “linguagem” usada e, portanto, o que está sendo representado.

Um primeiro exemplo

A melhor forma para aprender a modelar e a diagramar algo é praticando. Depois de passados alguns conceitos, vamos refinar o cenário de exemplo descrito nesta unidade para ser mais específico com o trato de informações, e assim nos auxiliar nesta modelagem. O texto é o seguinte:

“Para o site da Loja Virtual funcionar, é necessário que os produtos sejam cadastrados com suas informações de Código de Barras, Descrição e Valor. Quando um Cliente acessa o site, ele se cadastra com seus dados de CPF, Nome, Endereço e Telefone. O Endereço, na verdade, é um conjunto de campos ou atributos, dentre eles, Rua e Número, Complemento, Bairro, Cidade, UF e CEP. Uma vez cadastrado, o cliente pode realizar as suas compras, que, enquanto transação, precisam do registro de seu Número, Data, Valor e Status. O Status indica se a compra já foi entregue, se está em andamento etc. E, para finalizar nossa compreensão, as Compras contêm os produtos que o Cliente escolheu para comprar”.

Para construir uma primeira versão do diagrama DER com base no texto, basicamente, três ações serão necessárias: identificar as entidades, mapear os seus atributos e estabelecer os relacionamentos entre as entidades. Nessa diagramação, ferramentas são empregadas e, no nosso caso, está sendo usado o BrModelo, uma ferramenta muito simples e usada, sobretudo, para o contexto didático de modelagem ER.

Saiba mais

Para a modelagem de entidade e relacionamento, existe uma ferramenta bastante usada, que é o BrModelo. O link, disponibilizado a seguir, fala sobre esse projeto, como ele surgiu e como a ferramenta pode ser utilizada.

ACESSAR

A visão de como representar o cenário pode ter algumas variações de projetista para a projetista. Uma primeira versão para expressar o cenário descrito no texto está na Figura 2.6. Perceba que as entidades identificadas foram Produtos, Clientes e Compras, e dentre os atributos de Produtos, foram diagramados Código, Descrição e Valor, e existem dois relacionamentos entre Clientes-Compras e entre Compras-Produtos. Outras especificações podem ser vistas no próprio diagrama. Para efeito didático, algumas questões do mundo real foram abstraídas, por exemplo, um produto pode ter outras informações, como fotos, um Cliente pode ter muitos outros dados associados, e assim por diante.

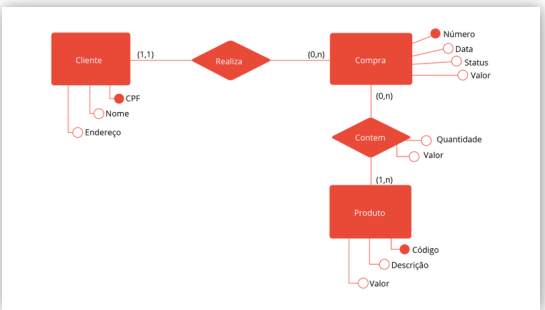


Figura 2.6 – Versão inicial de um DER para o cenário de exemplo (Loja Virtual)
Fonte: Elaborada pelo autor.

Reflita

Pense em três cenários de sistemas comuns em nosso dia a dia: uma papelaria, uma padaria ou uma agenda no celular. Identifique as entidades nesses sistemas, como se relacionam, e, caso seja possível, esquematize de que modo você desenharia um modelo de entidade e relacionamento.

Outras Características: Atributos e Cardinalidade

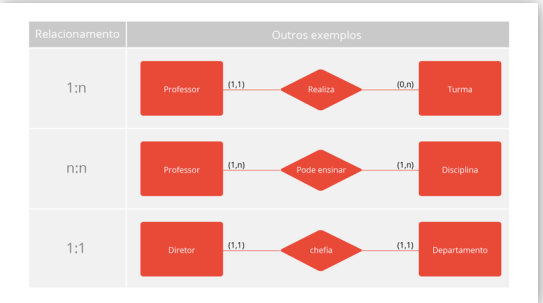
Perceba que, no diagrama, existem alguns atributos que estão marcados ou pintados. São os atributos identificadores, isto é, podem identificar, exclusivamente, a entidade. Dessa forma, o CPF é um atributo identificador de Cliente, pois não existem dois clientes que possuem o mesmo CPF, ou seja, um número de CPF é único, identifica unicamente o Cliente. O mesmo ocorre para Produtos, que são identificados pelo atributo Código de Barras, e a entidade Compras, que é identificada pelo atributo Número. Esses atributos serão, depois, tratados como chaves primárias no banco de dados.

Ainda, é possível visualizar, no modelo, alguns números (0,1), (0,n) ou (1,n) junto aos relacionamentos nas entidades, representando a cardinalidade. A cardinalidade indica quantos objetos de uma entidade podem se relacionar com objetos de outra entidade. No nosso exemplo, um Cliente pode se relacionar com (0,n) Compras, o que indica que um Cliente pode fazer nenhuma compra ou até n compras. Por outro lado, uma Compra tem uma relação (1,1) com Cliente, isto é, ela precisa estar associada a um Cliente, e somente a um Cliente. Não é possível, por exemplo, ter uma compra sem um cliente associado.

Do mesmo modo, uma compra pode conter N produtos e deve ter, pelo menos, um produto (relação (1,n)), e um produto pode estar registrado em N compras, mas pode também não ser associado a uma compra, por exemplo, um tipo de produto novo que não foi vendido ainda. Dessa forma, com os vários tipos de cardinalidade são estabelecidas as regras sobre a quantidade de itens envolvidos nos relacionamentos.

Além das citadas, existem situações em que relações 1:1 (1 para 1) que não foram representadas em nosso exemplo. Nesse caso, um objeto se relaciona com apenas um, de outra entidade. Por exemplo, em um sistema, imagine que um Funcionário só pode ocupar um computador em uma estação de trabalho; assim, existe uma relação 1:1 entre eles. Ou ainda, imagine que, em uma empresa, um só funcionário pode chefiar um departamento, que só é chefiado por um funcionário.

O quadro a seguir, resume alguns dos tipos de relacionamentos, conforme a cardinalidade. Os valores de 0 no lugar de 1 indicariam que um objeto pode não se relacionar com o outro objeto da outra entidade:



Quadro 2.1 – Exemplos de relacionamentos em um MER
Fonte: Elaborada pelo autor.

Outras Situações Importantes

Além das situações e elementos citados até agora, mais comuns em modelos ER, existem outras situações importantes, que podem, por vezes, ocorrer: autorrelacionamento, generalização e o conceito de entidades fortes e fracas.

No autorrelacionamento, há um relacionamento de uma entidade para ela mesma. Por exemplo, em um sistema, uma Pessoa pode ser cônjuge de outro elemento que é também da entidade Pessoa, ou um Funcionário pode ter como supervisor um outro elemento, que também é Funcionário. A Figura 2.7 representa um desses autorrelacionamentos:

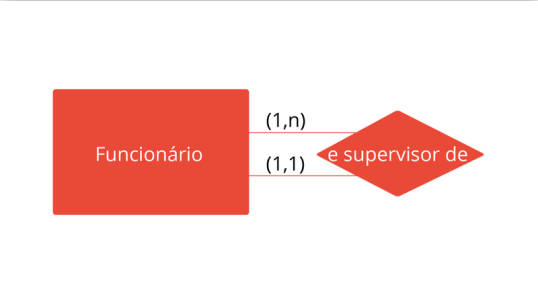


Figura 2.7 – Exemplo de autorrelacionamento
Fonte: Elaborada pelo autor.

A generalização ocorre quando uma entidade engloba características de outras entidades subjacentes. A entidade mais "alta" generaliza os conceitos subjacentes. Um exemplo típico de uma entidade Pessoa, em um sistema, pode ser uma Pessoa Física ou uma Pessoa Jurídica, ou uma determinada entidade Usuário, que pode ser um Funcionário, um Professor ou um Estudante, em uma biblioteca da universidade. No caso, o conceito Pessoa é mais geral, generaliza os conceitos de Pessoa Física e Pessoa Jurídica. Da mesma forma todo Funcionário, Professor e Estudante são um Usuário da biblioteca da universidade. Portanto, Usuário generaliza os outros elementos. Um exemplo de generalização pode ser visto na Figura 2.8:

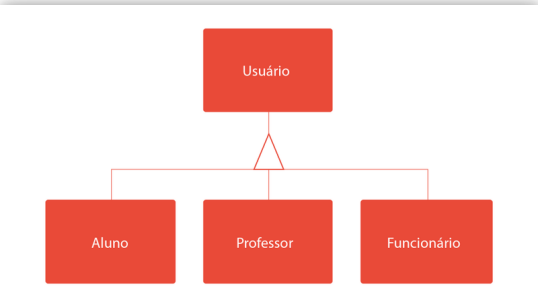


Figura 2.8 – Exemplo de relacionamento de generalização
Fonte: Elaborada pelo autor.

Uma entidade fraca é uma entidade cuja existência depende da existência de outra entidade, no caso, uma entidade forte. Para exemplificar, imagine que um sistema de Recursos Humanos precisa ter uma entidade de Dependentes.

A existência de um dependente só faz sentido se estiver vinculada a um elemento da entidade Funcionário. Em geral, a identificação da entidade fraca depende de um atributo identificador da entidade forte. Em dependente, para ser identificado, ele poderia ter um número, juntamente com a identificação do funcionário, isto é, a tupla Identificação do Funcionário e número do dependente identifica o Dependente. A Figura 2.9 ilustra outro exemplo de relacionamento entre uma entidade fraca e uma entidade forte. Uma Agência só faz sentido de existir relacionada a um Banco, e sua identificação será pelo número da agência em conjunto com a identificação do banco.

Figura 2.9 – Exemplo de relacionamento entre entidade forte e entidade fraca
Fonte: Elaborada pelo autor.

praticar

Vamos Praticar

Em Sistema de Escolas, um novo módulo foi pensado para organizar as atividades extracurriculares, como aulas de música, esportes e outros. As atividades serão organizadas em TURMAS, e em cada turma serão inscritos os alunos. Um aluno pode, por exemplo, fazer a inscrição numa turma de Natação e em outra turma de Violão.

Na Modelagem de Entidade e Relacionamento, qual a cardinalidade na relação Turmas e Alunos do novo módulo?

- ○ **a)** Relação 1:N
- ○ **b)** Relação 0:N

- ○ **c)** Relação N:N
- ○ **d)** Relação 1:1
- ○ **e)** Relação de Generalização

Modelagem Relacional

Vamos relembrar aqui o que foi abordado: no modelo de entidade e relacionamento, o foco é “estar mais próximo” do mundo real, representando-o em uma perspectiva conceitual com independência quanto à tecnologia que seria empregada. Por exemplo, o mesmo modelo vale para implementar nos diversos tipos de bancos de dados (hierárquico, rede e outros). Para tratar a forma como as informações serão implementadas, efetivamente, em um banco de dados, com tipos de dados, codificações, arquivos e outras questões mais tecnológicas de implementação (e menos conceituais), é necessário utilizar uma modelagem do nível Lógico e Físico, e para bancos de dados relacionais utiliza-se o Modelo Relacional.

Consoante Elmasri (2011), existe uma boa e histórica introdução ao modelo relacional, que diz:

> *O modelo de dados relacional foi introduzido inicialmente por Ted Codd, da IBM Research, em 1970, em um artigo clássico (Codd, 1970), que atraiu atenção imediata devido à simplificidade e base matemática. O modelo usa o conceito de relação matemática – que se parece com uma tabela de valores – como seu bloco de montagem básico, e sua base teórica reside em uma teoria de*

> *conjuntos e lógica de predicado de primeira ordem. (ELMASRI, 2011, p. 38).*

saiba mais
Saiba mais

Além dos bancos de dados relacionais, existem outros, tais como os históricos em rede e os hierárquicos. Existem os Bancos Orientados a Objetos e, ultimamente, muito tem se falado sobre os bancos NoSQL. Para saber mais, acesse o link.

ACESSAR

Em termos práticos, neste momento, é feita uma conversão da modelagem ER em uma modelagem relacional. Na verdade, será muito comum encontrar equipes de desenvolvimento, que inicia o trabalho com banco de dados com base no modelo relacional, já que existe forte equivalência entre eles. E o que vai mesmo determinar como as aplicações vão trabalhar com dados são as tabelas (ou relações) no banco de dados. Para nós, que tivemos a compreensão do MER antes, aprenderemos o Modelo Relacional justamente fazendo uma equivalência do que foi representado no MER e sua conversão para esse novo modelo. E, além disso, como nesse nível lógico e físico, novos detalhes tecnológicos são tratados. Aprenderemos também o que é necessário acrescentar no modelo.

Relações ou Tabelas

Conforme Elmasri (2001), o bloco básico de montagem do modelo relacional e, por conseguinte, do banco de dados relacional, é o conceito de relação. Os termos relação e tabela são equivalentes: em geral, relação é usado com mais frequência no campo teórico, e tabela será o jargão adotado no dia a dia.

Mas, por ora, entenderemos as definições de relação, tuplas, atributos e outros, e, para isso, a Figura 2.10 pode ser bem útil. Uma relação é um conjunto de tuplas (ou linhas de uma tabela), que, por sua vez, é um conjunto de valores com o respectivo valor referente a um atributo. Esses valores pertencem a um domínio, ou são de um tipo de dados (inteiro, conjunto de caracteres, real e outros), e podem, ainda, ter um valor NULL, que indica sem valor.

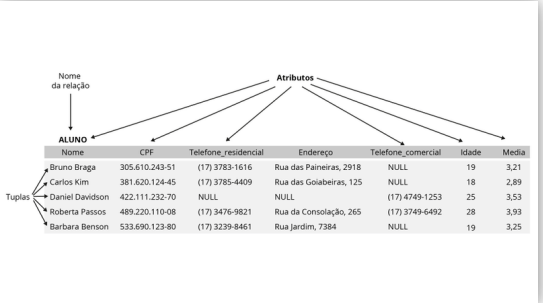


Figura 2.10 – Exemplo de relação
Fonte: Elmasri (2011, p. 40).

Com essa compreensão, percebe-se que uma primeira equivalência, e conversão a ser feita, é o conceito de relação com base em uma entidade do modelo MER. As entidades acabarão se constituindo em relações ou tabelas, no modelo relacional.

Um Pouco de Prática e como Aprender

Na prática, veremos que nosso modelo será o apresentado a seguir (Figura 2.11). Note que é fácil ver algumas entidades convertidas aqui, nas relações, e também a ocorrência de outros elementos. Para criar modelos relacionais, existem diversas ferramentas, e duas foram base para criação dos modelos aqui: o DBDesigner e o MySQL Workbench.

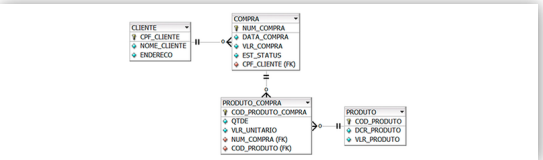


Figura 2.11 – Modelo Relacional do Cenário de Exemplo
Fonte: Elaborada pelo autor.

saiba mais

Saiba mais

Conheça algumas ferramentas que podem ser usadas para fazer modelagem relacional, na prática. Duas que eventualmente serão usadas no texto são o DBDesigner 4 e o MySQL Workbench. Recomendamos este último, pois é bastante popular na comunidade de MySQL. Mas vale perceber que o mais relevante é o modelo em si, como os dados foram representados. Para saber mais, acesse os links disponíveis a seguir.

- Artigo sobre ferramentas de modelagem:

< https://becode.com.br/diagramas-er-ferramentas/ >. Acesso em: 26 abr. 2019.

- Sobre o MySQL Workbench:

ACESSAR

Atributos, Tipos, Obrigatoriedade e Chaves Primárias

Cada um dos atributos equivalentes aos atributos no MER – precisa ser associado a um tipo de dados, que, na teoria, é associado ao conceito de domínio, pois indica qual domínio de valores o atributo pode assumir, domínio dos números inteiros, dos números reais ou outros. O Quadro a seguir mostra alguns dos tipos de dados mais usados.

Tipo de Dados	Descrição	Exemplos
INTEGER	Números inteiros	23, 341, 899
CHAR (N)	“Palavras” ou sequências de caracteres que são armazenados com um tamanho fixo de N caracteres	“Joana”, “Marcos”
VARCHAR(N)	Nesse caso, o banco de dados pode otimizar, variando o armazenamento conforme necessidade até o máximo de N caracteres	Idem anterior, a mudança é no armazenamento interno
NUMERIC(P,D)	Muito usado para números reais. A Precisão P indica quando dígitos podem ter o número e a quantidade de decimais; D indica quantas casas decimais. Por exemplo, NUMERIC(5,2) seria um número como 432,21	347,20 480,21
DATE	Para campos do tipo data	22/03/2019
BLOB	Usado para dados binários, por exemplo, caso seja necessário armazenar um arquivo de imagem ou de vídeo no banco de dados	-

Quadro 2.2 – Tipos de Dados
Fonte: Elaborada pelo autor.

Algumas restrições podem ser associadas a um atributo, por exemplo, a obrigatoriedade. Um atributo obrigatório é aquele que precisa, necessariamente, ter um valor, que não pode ser NULL. Por exemplo, se um novo aluno está sendo cadastrado, não faz sentido ele não ter o seu nome preenchido.

Alguns atributos podem identificar uma tupla, linha da tabela ou entidade. Esses são aqueles atributos identificadores discutidos na modelagem de entidade e relacionamento. Aqui no modelo relacional, esses atributos são tratados como chaves. O atributo que identifica a tupla é considerada chave primária, e outros eventuais atributos, que também identificam a tupla, podem ser considerados chaves candidatas. Uma chave pode ser composta, isto é, pode conter mais de um atributo. Assim, em uma entidade ALUNO, a sua matrícula pode ser considerada a chave primária, o CPF, uma chave candidata, e os campos CI e UF (carteira de identidade e estado), também chave candidata, porém chave composta.

A Figura 2.12 a seguir mostra como nossa entidade PRODUTO, do modelo de entidade e relacionamento de nosso modelo de loja, foi convertido para sua tabela equivalente no modelo relacional, com seus atributos, tipos de dados e sinalização da chave primária:

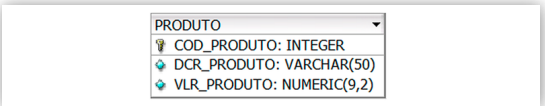


Figura 2.12 – Exemplo de relação ou tabela no Modelo Relacional
Fonte: Elaborada pelo autor.

O quadro a seguir ilustra com mais detalhes as definições dos atributos da tabela:

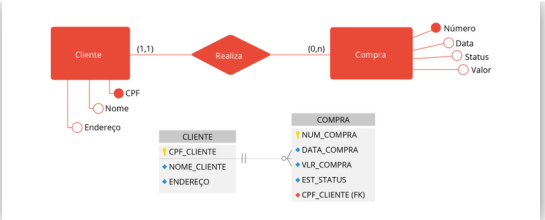
Atributo	Descrição	Tipo	Obrigatório	Chave
COD_PRODUTO	Código do produto	INTEGER	SIM	SIM
DCR_PRODUTO	Descrição do produto	VARCHAR(50)	SIM	-
VLR_PRODUTO	Valor do Produto	NUMERIC(9,2)	NÃO	-

Quadro 2.3 – Definição dos campos na tabela Produtos
Fonte: Elaborada pelo autor.

Relacionamentos e Chaves Estrangeiras

Um conhecimento típico, nessa transição do modelo de entidade e relacionamento, é o trato com os relacionamentos, pois a forma de fisicamente representá-los em um modelo relacional é transpondo a chave primária de uma tabela para a outra tabela, que passa a ser chamada chave estrangeira, nome dado justamente para referenciar uma chave em outra tabela.

Um exemplo pode nos ajudar a compreender melhor essa conversão, e vamos relembrar de nosso cenário. Em nosso modelo, um CLIENTE realiza COMPRA com cardinalidade, em que um cliente pode estar associado a diversas compras, e uma compra só pode estar associada um CLIENTE. Nesse caso, a chave primária de CLIENTE passa a incorporar a tabela COMPRA, e na tabela de compra é denominada chave estrangeira por referenciar um elemento da tabela de CLIENTE. A Figura 2.13 ilustra essa conversão muito comum em relacionamentos 1:N:



Quadro 2.13 – Transposição de chave estrangeira em relacionamento 1:N
Fonte: Elaborada pelo autor.

Assim como há esse tratamento no relacionamento 1:N, os outros tipos precisam também de ser tratados. Outra situação muito comum é a do relacionamento N:N, por exemplo, no nosso exemplo, a relação que existe entre COMPRA e PRODUTO, pois uma COMPRA pode conter diversos produtos, e um PRODUTO pode estar em diversas compras. Nesse caso, surge uma tabela associativa, isto é, uma nova tabela, que faz a associação com as outras duas. A Figura 2.14 representa melhor essa situação. Na esquerda, o relacionamento N:N, e na direita, a tabela associativa equivalente, PRODUTO_COMPRA.

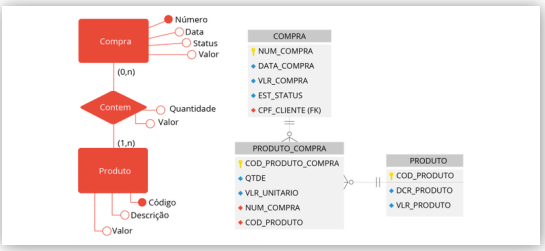
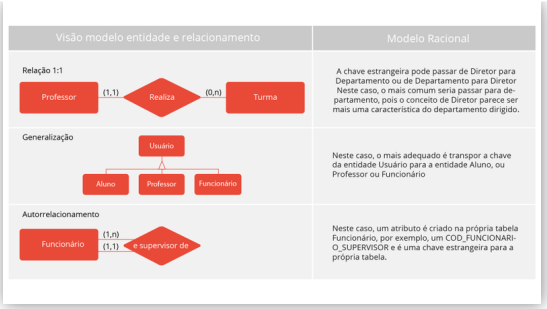


Figura 2.14 – Conversão de relação N:N em tabela associativa
Fonte: Elaborada pelo autor.

Os outros relacionamentos também apresentam uma forma de conversão para sua versão equivalente no modelo relacional. Note que a forma como as chaves primárias são transpostas para as tabelas relacionadas é que determina essa forma de conversão. O Quadro 3.X, a seguir, mostra a conversão que resta, do relacionamento 1:1 e do relacionamento de generalização:



Quadro 2.4 – Descrição da forma de determinar as chaves estrangeiras nos relacionamentos
Fonte: Elaborada pelo autor.

praticar

Vamos Praticar

Em um Sistema Escolar, um professor pode ministrar diversas disciplinas, e uma disciplina possui vários professores aptos para ministrá-la. Para converter essa representação de uma modelagem de entidade e relacionamento, deve-se:

- ○ **a)** transpor a chave primária de Disciplina para Professor;
- ○ **b)** transpor a chave primária de Professor para Disciplina;
- ○ **c)** criar uma chave estrangeira em Disciplina;
- ○ **d)** criar uma chave estrangeira em Professor;
- ○ **e)** criar uma tabela associativa com chaves estrangeiras, uma para cada tabela Disciplina e Professor.E.

Normalização

Ao analisar o que vimos anteriormente, sobre modelagem, usando como base o modelo relacional, inferimos que cada informação possui a sua tabela, o seu local de ser armazenado, evitando, desse modo, redundâncias e algumas inconsistências ao manipular os dados. De maneira informal, essa pode ser a explicação para o que, em banco de dados, chamamos normalização.

Elmasri trata normalização da seguinte maneira:

> *A normalização de dados pode ser considerada um processo de analisar os esquemas de relação dados com base em suas dependências funcionais e chaves primárias para conseguir as propriedades desejadas de (1) minimização de redundância e (2) minimização de anomalias de inserção, exclusão e atualização. Ele pode ser considerado um processo de filtragem ou purificação que faz com que o modelo tenha qualidade cada vez melhor. (ELMASRI, 2011, p. 348).*

A dependência funcional referida, ao referir à normalização, diz respeito à dependência entre dois atributos. Diz-se que um atributo B depende, funcionalmente, de A quando duas tuplas com mesmo valor para A,

necessariamente, devem apresentar o mesmo valor para B. Veja o exemplo de relação a seguir. Verifique que misturamos os valores de clientes com compras, e existem algumas diferenças em relação ao que fizemos até agora (CPF mudou para o campo COD). Obviamente, essa “mistura” não é interessante, apenas será útil para nossa compreensão de normalização.

Voltando ao conceito de dependência funcional, note que, para cada duas tuplas que possuem o mesmo código, têm, necessariamente, o mesmo Nome. As tuplas com valor 1 possuem nome “João”; as tuplas com valor 2, “Maria”, e as tuplas com COD igual a 4, “Joana”. Portanto, podemos dizer que o atributo Nome depende funcionalmente de COD.

COD	NOME	NUM_COMPRA	DATA	VALOR
1	João	10	07/04/2019	35,60
1	João	11	12/04/2019	35,60
2	Maria	12	25/03/2019	42,20
2	Maria	14	02/04/2019	28,50
4	Joana	19	19/03/2019	56,80
4	Joana	21	10/04/2019	62,90

Tabela 2.1 – Uma tabela “desnormalizada” para exemplificar a normalização
Fonte: Elaborada pelo autor.

Na normalização, algumas dessas dependências serão checadas ou testadas de forma que se encontre uma melhor forma de separar as informações. Para cada tipo de teste, de acordo com uma situação, podemos dizer que o conjunto de dados foi normalizado conforme a primeira, segunda ou terceira forma normal, as mais comuns.

Primeira Forma Normal (1FN)

Na primeira forma normal (1FN), os atributos não podem, para uma mesma entidade ter valores diferentes ou atributos possuindo mais de um valor, o que significa que não é permitido ter, na mesma relação, atributos multivalorados. No nosso exemplo anterior, observe o campo Telefone. Para uma mesma pessoa existem valores diferentes, e na segunda linha o campo possui dois valores. Portanto, Telefone é um atributo multivalorado.

Para tratar essa situação, da primeira forma normal, deve-se identificar a chave primária e o atributo multivalorado e separá-los (o atributo multivalorado), em uma outra relação ou tabela, e utilizar a chave primária da entidade como chave estrangeira na nova relação. A decomposição da relação ficaria conforme figura, a seguir:

COD (FK)	TELEFONE
1	996542331
1	997218765
1	998831245
2	322284129
2	344567881
4	987221015
4	987221015

COD	NOME	NUM_COMPRA	DATA	VALOR
1	João	10	07/04/2019	35,60
1	João	11	12/04/2019	35,60
2	Maria	12	25/03/2019	42,20
2	Maria	14	02/04/2019	28,50
4	Joana	19	19/03/2019	56,80
4	Joana	21	10/04/2019	62,90

Tabela 2.2 – Tratamento da primeira forma normal (1FN)
Fonte: Elaborada pelo autor.

Segunda Forma Normal (2FN)

Na segunda forma normal (2FN), os atributos que não compõem a chave devem depender unicamente da chave primária da tabela, e, além disso, os atributos que não são dependentes dessa chave devem ser realocados em uma nova tabela, utilizando esses dados.

No nosso caso, perceba que o atributo Nome depende unicamente de COD (no caso Código do Cliente) os atributos de NUM_COMPRA, DATA e VALOR não possuem essa dependência funcional, pois duas tuplas com valores diferentes estão associadas ao mesmo valor de código. Isso ocorre porque misturamos os valores de Compras junto aos valores de Clientes.

Antes de seguir para a decomposição, vale comentar que as formas normais devem ser acumulativas. Portanto, para uma relação obedecer à segunda forma normal, (2FN), deve, previamente, ter atendido à primeira forma normal (1FN) dos atributos multivalorados.

No caso da decomposição da segunda forma normal, a tarefa é identificar a chave primária, os atributos dependentes e os atributos não dependentes. Desse modo, separam-se os atributos não dependentes em uma outra relação e faz-se o relacionamento através da chave estrangeira.

COD	NOME
1	João
2	Maria
4	Joana

COD (FK)	NUM_COMPRA	DATA	VALOR
1	10	07/04/2019	35,60
1	11	12/04/2019	35,60
2	12	25/03/2019	42,20
2	14	02/04/2019	28,50
4	19	19/03/2019	56,80
4	21	10/04/2019	62,90

Tabela 2.3 – Tratamento da segunda forma normal (2FN)
Fonte: Elaborada pelo autor.

Terceira Forma Normal (3FN)

Para estar na terceira forma normal (3FN), a relação, primeiramente, deve estar na segunda forma normal. Lembre-se de que as formas normais são acumulativas, e, para estar em uma forma normal “maior”, devem-se ter

atendido as formas normais “menores”. Para a 3FN, em específico, é preciso eliminar os campos que podem ser obtidos a partir de um cálculo, ou de alguma lógica aplicada a outro campo.

Imagine que, na nossa relação de compras, tivesse um atributo chamado VLR_IMPOSTO, que diz respeito a um cálculo do imposto, calculado sobre o valor da compra. Portanto, o valor do imposto não precisa, necessariamente, ser armazenado. Nesse caso, o atributo, ou campo, pode ser retirado da relação, e os valores são calculados via regras de negócio da aplicação, isto é, o programa, quando manipulasse os dados de valor, faria os cálculos de imposto via alguma parte do programa.

Normalização, Desnormalização e Performance

A normalização, como vimos, permite que os dados fiquem melhor estruturados, e isso tem uma motivação: consegue-se melhor integridade e qualidade da informação. Por exemplo, imagine que a nossa relação desnormalizada, em que existem várias tuplas com os mesmos nomes e códigos, concentra na primeira e segunda linha, que possui o valor de COD igual a 1, e NOME igual a “João”. Suponha que gostaríamos de alterar o nome do cliente de “João” para “João Manoel”, e fizéssemos isso apenas na primeira linha. Teríamos, dessa forma, uma inconsistência ou anomalia nos dados, uma tupla de cliente com código 1 e nome “João”, e outra também com código 1, porém com nome “João Manoel”. Portanto, a normalização dos dados permite encontrar um esquema, ou estruturação, que garante melhor a qualidade das informações armazenadas, que ficam mais imunes a essas distorções.

Por outro lado, um banco muito normalizado pode causar alguns impactos de performance. Retomemos o caso da 3FN, do cálculo de imposto. A todo o momento que a compra for manipulada, um processamento deve ser feito para realizar o cálculo do imposto. Se formos ainda mais a fundo em nosso modelo, desde o início colocamos um campo de Valor da Compra, porém esse

valor pode ser processado a partir da soma dos valores dos itens presentes na Compra. Mas note que, toda vez que uma compra for recuperada, a tabela de itens precisa ser processada para se calcular o total da compra para se ter o total da compra. Isso requer maior processamento para tratar dos dados.

Dessa forma, em algumas situações o projetista de dados pode tomar decisões quanto à normalização ou não dos dados, tendo a devida cautela para especificar bem, no caso de algumas desnormalizações, as regras de negócios a fim de que os dados sejam mantidos com boa consistência.

saiba mais

Saiba mais

As consistências das informações em um banco de dados dependem das regras estabelecidas no próprio banco de dados e das regras de negócios que são implementadas de outra forma, via alguma lógica ou procedimentos no próprio banco de dados ou, ainda, outras camadas na arquitetura do software. O link disponibilizado a seguir permite entender mais as regras de negócios e decisões de projetos sobre esse assunto.

ACESSAR

praticar

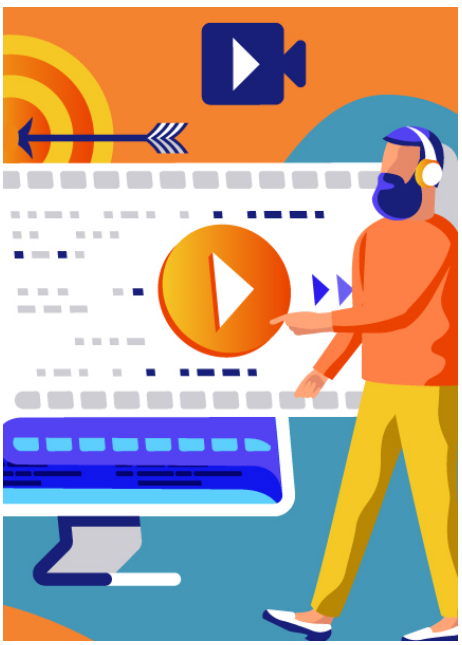
Vamos Praticar

Numa relação NOTAS_ALUNOS, que armazena as notas de alunos em determinada disciplina, existem as colunas NOTA1, NOTA2 e NOTA_FINAL. A coluna NOTA_FINAL é calculada com base na média aritmética da colunas NOTA1 e NOTA2. Analisando as regras de normalização, qual modificação deveria ser feita nesse caso?

- ○ **a)** Identificar a chave da tabela e criar outra tabela com média, para atender à primeira forma normal.
- ○ **b)** A relação poderia permanecer assim, pois já está normalizada.
- ○ **c)** O campo média poderia ser abstraído, pois é um cálculo com base em outros dois campos, situação tratada pela 3FN.
- ○ **d)** Identificar a chave da tabela e criar outra tabela com os campos NOTA1, NOTA2 e NOTA_FINAL.
- ○ **e)** Identificar a chave da tabela e criar outra tabela com os campos NOTA1 e NOTA2, apenas.

indicações

Material Complementar



FILME

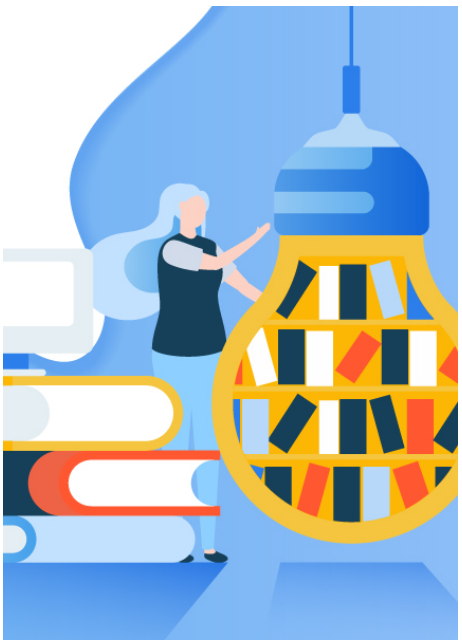
A Rede Social

Ano: 2010

Comentário: O filme trata de como o Facebook, uma das redes sociais mais populares hoje, foi concebida por Mark Zuckemberg. Na relação com este capítulo, o longa é um exemplo de concepção de um sistema e de toda a história relacionada à criação de um novo negócio com base em uma ideia de um estudante. A história pode servir para demonstrar como novos negócios surgem atualmente, ao observar e promover mudanças no mundo real. Criar e modelar *software* pode estar no contexto de grandes empresas que contratam um desenvolvimento em que os dados

precisam ser modelados, mas pode também ocorrer quando uma *startup* resolve empreender uma nova revolução.

TRAILER



LIVRO

UML: Guia do Usuário

Grady Booch, James Rumbaugh e Ivar Jacobson.

Editora: Elsevier

ISBN: 9788535217841

Comentário: Esse é um livro de modelagem, o que tem forte relação com o foco deste capítulo, de modelagem de dados. Porém, a modelagem a que se refere o livro é Modelagem de Software, em geral, e em UML. A leitura pode ser útil, pois abrirá a mente com relação a aspectos de modelagem, como representar coisas do software, dentre elas, as representações de dados que, no caso da UML, são feitas com modelagem de classes.

conclusão

Conclusão

A modelagem de dados é uma atividade de suma relevância na construção de bancos de dados, sobretudo para “domar a complexidade” de sistemas maiores. Analisar os cenários numa visão conceitual e construir uma lógica de representação de um banco de dados será um pilar para outras decisões na engenharia de *software* .

Por isso, os assuntos estudados agregam bastante valor para a área, e serão essenciais para o estudante que pretende internalizar o papel de um Projetista de Dados.

Seja em uma modelagem ER, ou na modelagem relacional, habilidades de identificar elementos, ter visão sistêmica, organizar e esquematizar um diagrama, tomar decisões de projeto, com domínio no trato com tabelas e elementos de um banco relacional, são competências que podem ser refinadas, quando aprofundados os assuntos abordados nesta unidade.

referências

Referências Bibliográficas

AMADEU, C. V. **Banco de Dados** . São Paulo: Pearson Education do Brasil, 2014.

BOOCH, G.; RUMBAUGH, J; JACOBSON, I. **UML** : guia do usuário. Rio de Janeiro: Elsevier, 2012.

ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados** . São Paulo: Pearson Addison Wesley, 2011.

KOWALTOVISKY, D. C. C. K. **Arquitetura escolar** : o projeto do ambiente de ensino. São Paulo: Oficina de Textos, 2011. p. 98.

PUGA, S. **Banco de Dados** : implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson Education do Brasil, 2013.

YARO JR, B. LYRA, A. C. C. **Circuitos Elétricos** . São Paulo: Pearson Prentice Hall, 2006. p. 35.