

GUIA MANGÁ DE BANCOS DE DADOS

MANA TAKAHASHI
SHOKO AZUMA
TREND-PRO CO., LTD.



novatec

Original Japanese-language edition Manga de Wakaru Database ISBN 4-274-06631-2 © 2004 by Mana Takahashi and TREND-PRO Co., Ltd., published by Ohmsha, Ltd.

English-language edition The Manga Guide to Databases ISBN 978-1-59327-190-9 © 2009 by Mana Takahashi and TREND-PRO Co., Ltd., co-published by No Starch Press, Inc. and Ohmsha, Ltd.

Portuguese-language rights arranged with Ohmsha, Ltd. and No Starch Press, Inc. for Guia Mangá de Bancos de Dados ISBN 978-85-7522-163-1 © 2009 by Mana Takahashi and TREND-PRO Co., Ltd., published by Novatec Editora Ltda.

Edição original em Japonês Manga de Wakaru Database ISBN 4-274-06631-2 © 2004 por Mana Takahashi e TREND-PRO Co., Ltd., publicado pela Ohmsha, Ltd.

Edição em Inglês The Manga Guide to Databases ISBN 978-1-59327-190-9 © 2009 por Mana Takahashi e TREND-PRO Co., Ltd., co-publicação da No Starch Press, Inc. e Ohmsha, Ltd.

Direitos para a edição em Português acordados com a Ohmsha, Ltd. e No Starch Press, Inc. para Guia Mangá de Bancos de Dados ISBN 978-85-7522-163-1 © 2009 por Mana Takahashi e TREND-PRO Co., Ltd., publicado pela Novatec Editora Ltda.

Copyright © 2009 da Novatec Editora Ltda.

Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Ilustração: Shoko Azuma

Tradução: Thaís Cristina Casson

Revisão gramatical: Lia Gabriele Regius

Editoração eletrônica: Camila Kuwabata e Carolina Kuwabata

ISBN: 978-85-7522-163-1

Histórico de impressões:

Novembro/2010 Primeira reimpressão

Outubro/2009 Primeira edição

NOVATEC EDITORA LTDA.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Fax: +55 11 2950-8869

E-mail: novatec@novatec.com.br

Site: www.novatec.com.br

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Takahashi, Mana

Guia mangá de bancos de dados / Mana Takahashi, Shoko Azuma, Trend-pro Co ; [ilustração] Shoko Azuma ; [tradução Thaís Cristina Casson]. -- São Paulo : Novatec Editora ; Tokyo : Ohmsha ; São Francisco : No Starch Press, 2009.

Título original: The manga guide to databases
ISBN 978-85-7522-163-1

1. Banco de dados - História em quadrinhos
2. Banco de dados - Gerência - História em quadrinhos
3. SQL (Linguagem de programação para computadores) - História em quadrinhos
I. Azuma, Shoko. II. Trend-pro Co.. III. Título.

09-09515

CDD-005.7565

Índices para catálogo sistemático:

1. Banco de dados : Computadores : Processamento de dados : História em quadrinhos 005.7565
VC20101112

SUMÁRIO

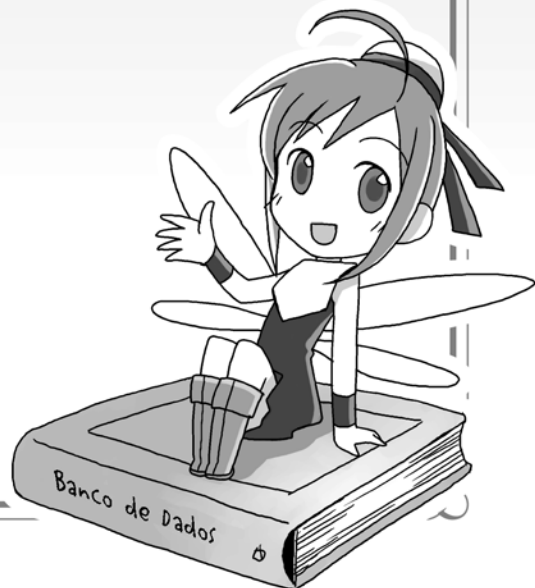
Prefácio	ix
1.....	1
O que é um Banco de Dados?	1
Por que precisamos de Bancos de Dados?	2
O que está acontecendo no reino?	16
Os dados estão duplicados.	16
Podem ocorrer conflitos nos dados.	17
Dados são difíceis de atualizar.	18
Um Banco de Dados - é a nossa solução!	19
Como usar um Banco de Dados	19
Resumo	21
2.....	23
O que é um banco de dados relacional?	23
Terminologia de Banco de Dados	24
Bancos de Dados relacionais	34
Tipos de modelos de dados	39
Operações de extração de dados	39
Operações de conjunto	39
Operações relacionais	43
Questões	45
O Banco de Dados relacional vence!	47
Resumo	48
Respostas	48
3.....	49
Vamos projetar um banco de dados!	49
O modelo E-R	50
Normalização de uma tabela	56
O que é o modelo E-R?	74
Como analisar um modelo E-R	74
1º Caso: Relacionamento um-para-um	74
2º Caso: Relacionamento um-para-muitos	75
3º Caso: Relacionamento muitos-para-muitos.	75
Questões	76
Normalizando uma tabela	78
Questões	79
Passos para projetar um Banco de Dados	81
Resumo	81
Respostas	82

4	85
Vamos aprender sobre SQL!	85
Uso de SQL	86
Pesquisa de dados usando um comando SELECT	93
Uso de funções de totalização	98
Junção de tabelas	101
Criação de uma tabela	103
Visão geral sobre SQL	106
Pesquisa nos dados usando um comando SELECT	106
Criação de critérios	107
Operadores de comparação	107
Operadores lógicos	107
Padrões	108
Pesquisas	108
Questões	109
Funções de agregação numéricas	110
Totalização de dados com agrupamento	110
Questões	111
Pesquisa de dados	112
Uso de uma subconsulta	112
Uso de uma subconsulta correlata	113
Questões	114
Juntando tabelas	114
Criação de uma tabela	115
Inserção, atualização ou exclusão de linhas	116
Criação de uma visualização (View)	117
Questões	118
Resumo	119
Respostas	119
5	125
Vamos usar um Banco de Dados!	125
O que é uma transação?	126
O que é uma trava (bloqueio)?	131
Segurança de Bancos de Dados	138
Acelerando as coisas com indexação	143
Recuperação de desastres	148
Propriedades das transações	153
Atomicidade	153
Consistência	154
Isolamento	155
Durabilidade	159
Quando um desastre ocorre	161
Tipos de falhas	161
Pontos de checagem (checkpoints)	161
Questões	162

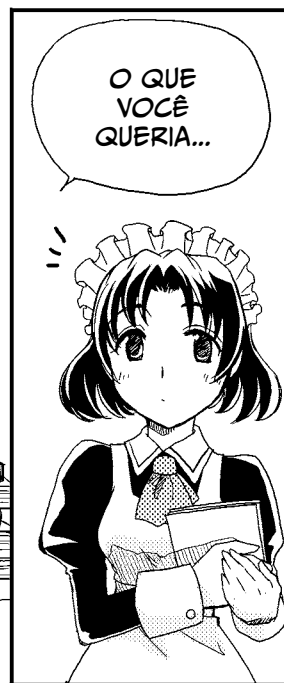
Índices	162
Questões	164
Otimização de uma consulta	164
Laços aninhados	165
Fusão ordenada (sort merge)	166
Hash	166
Otimizador	167
Resumo	167
Respostas	167
6	169
os Bancos de Dados estão em todos os lugares!	169
Banco de Dados em uso	175
Bancos de Dados e a web	177
Bancos de Dados distribuídos	183
Procedimentos armazenados e gatilhos	185
Bancos de Dados na Internet	194
Uso de procedimentos armazenados	196
Questões	196
O que é um Banco de Dados distribuído?	197
Distribuição horizontal	197
Distribuição vertical	198
Particionamento de dados	198
Particionamento horizontal	198
Particionamento vertical	199
Evitando inconsistências com efetivação em duas fases	199
Questões	201
Replicação de Banco de Dados	201
Somente leitura (Read-Only)	201
Replicação ativa em todos os servidores	202
Outras aplicações de Bancos de Dados	202
XML	202
Bancos de dados orientado a objeto	203
Resumo	205
Repostas	205
Apêndice	
Comandos SQL Usados com Frequência	207
Referências	209
Índice	211

1

O QUE É UM BANCO DE DADOS?

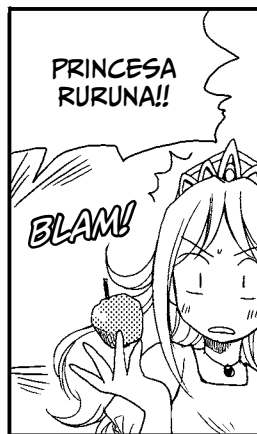
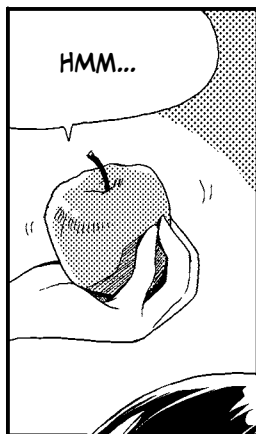


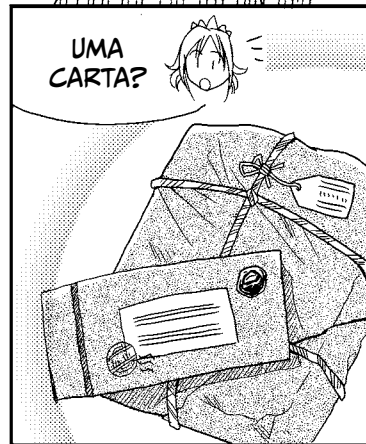
Por que precisamos de Bancos de Dados?

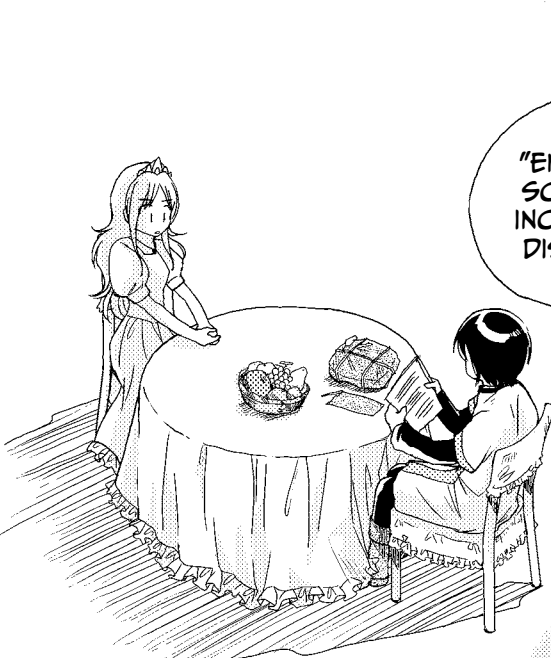


* POSSÍVEL REFERÊNCIA A EDGAR FRANK CODD (1924-2003),
MATEMÁTICO E CRIADOR DO MODELO DE BANCO DE DADOS RELACIONAL.











"ENCONTRAMOS UM LIVRO
SOBRE UMA TECNOLOGIA
INOVADORA EM UMA TERRA
DISTANTE QUE VISITAMOS.


A PESSOA QUE NOS DEU
ESTE LIVRO DISSSE QUE
ELE DESCREVE UMA MAGIA
SECRETA CHAMADA DE
BANCO DE DADOS.



SEGUNDO A PESSOA NOS
CONTOU, O BANCO DE
DADOS É UM SISTEMA
QUE PERMITE A TODO
MUNDO COMPARTILHAR,
GERENCIAR, E USAR DADOS.



MAS COMO ELE SERÁ
USADO DEPENDE DA ÍNDOLE
DE QUEM LER O LIVRO.

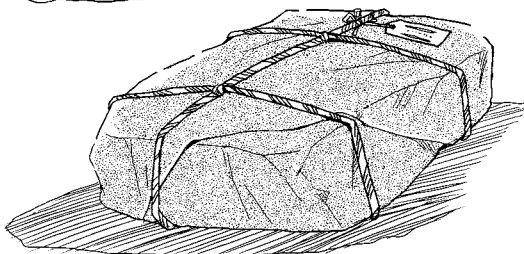


ELA NOS CONFIOU O
LIVRO ACREDITANDO
QUE O REINO DE KOD
O UTILIZARIA DE FORMA
PACÍFICA.



RURUNA...

ABRA O LIVRO,
E USE-O PARA
MELHORAR NOSSO
PAÍS."



QUÊ??

OH, POR FAVOR! VOCÊS
NÃO SABEM O QUANTO
ESTOU ESTRESSADA!



OH, É TÃO VELHO...



ESTÁ
TRANCADO.

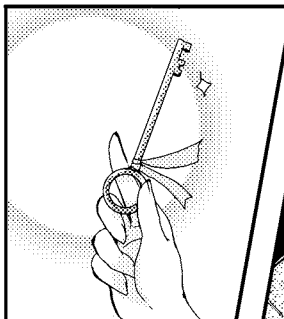
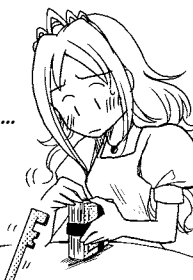
NÃO CONSIGO ABRIR.

HMM...

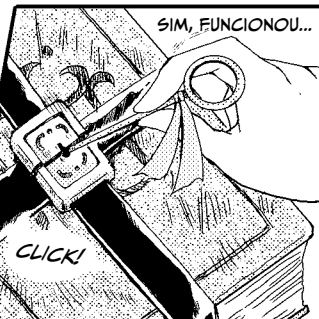


ESSA É A CHAVE
PARA O LIVRO?

ESTAVA NO ENVELOPE...



SIM, FUNCIONOU...



KA-BOOM!

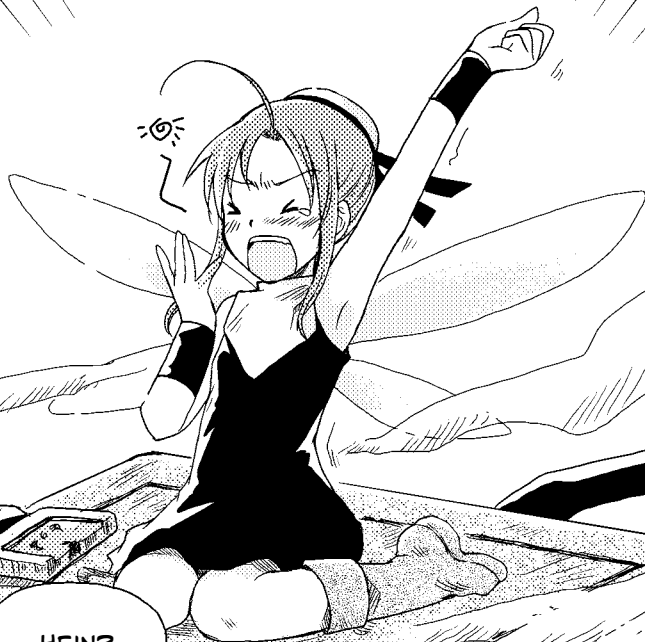
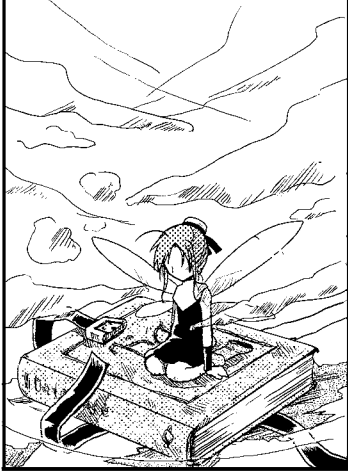
AIIEEE!!

?!

HAM!

COF!
COF!





HEIN?

ONDE ESTOU?

E QUEM SÃO
VOCÊS?

VOCÊ ESTÁ
NO CAS...CAS...
CASTELO DE
KOD.

ISSO
FALA!?

EU SOU CAIN!
- O AJUDANTE
PRINCIPAL
DA PRINCESA
RURUNA.

E QUEM É
VOCÊ?

?!

E VOA...?

ZIP!

UM FANTASMA?!

NÃO!

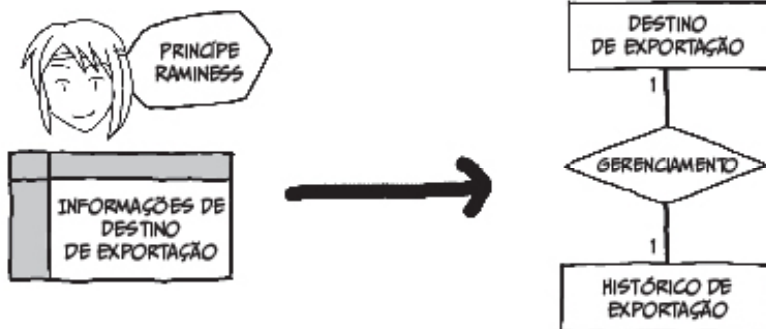
Utilizando um modelo E-R, tente definir as entidades em seus dados. Uma entidade é um objeto ou “coisa” no mundo real, como frutas ou destino de exportação.

```

graph TD
    FRUTA[FRUTA] --> VENDAS{VENDAS}
    VENDAS --> DESTINO[DESTINO DE EXPORTAÇÃO]
    FRUTA --- Codigo[CÓDIGO DO PRODUTO]
    FRUTA --- Nome[NOME DO PRODUTO]
    FRUTA --- Preço[PREÇO UNITÁRIO]
    VENDAS --- Quantidade[QUANTIDADE]
    DESTINO --- CodDest[COD. DE DESTINO DE EXPORTAÇÃO]
    DESTINO --- NomeDest[NOME DE DESTINO DE EXPORTAÇÃO]
  
```

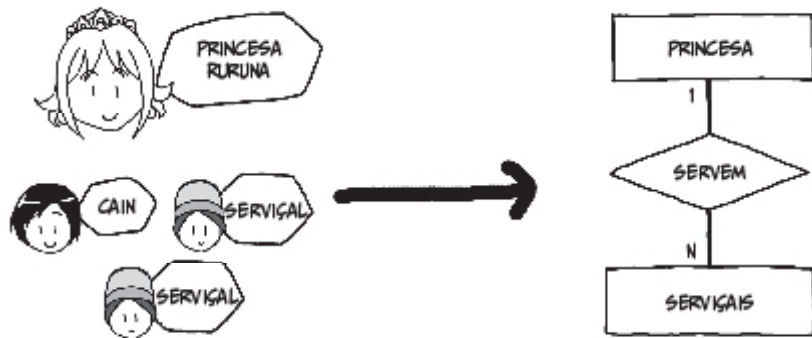
Como se fariam análises nos casos abaixo? Pense a respeito.

Um destino de exportação gerencia uma unidade de informação do histórico de exportação. Esse tipo de relacionamento é chamado de um-para-um.



2º CASO: RELACIONAMENTO UM-PARA-MUITOS

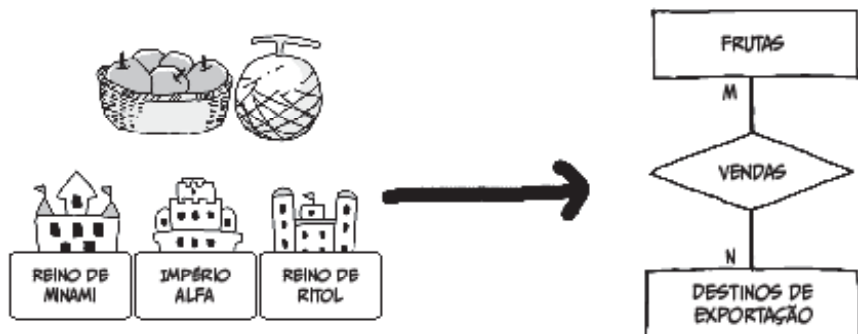
Vários serviçais servem uma princesa. Os serviçais não servem outra princesa, nem mesmo o rei.



Esse tipo de relacionamento é chamado de um-para-muitos.

3º CASO: RELACIONAMENTO MUITOS-PARA-MUITOS

Frutas são exportadas para vários destinos. Os destinos de exportação importam diversos tipos de frutas.



Este tipo de relacionamento é chamado de muitos-para-muitos.

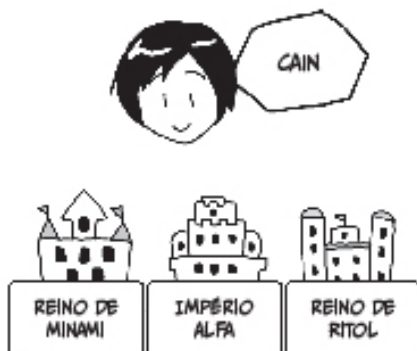


QUESTÕES

Quanto você entendeu do modelo E-R? Analise e desenhe um modelo E-R para cada um dos casos abaixo. As respostas estão na página 82.

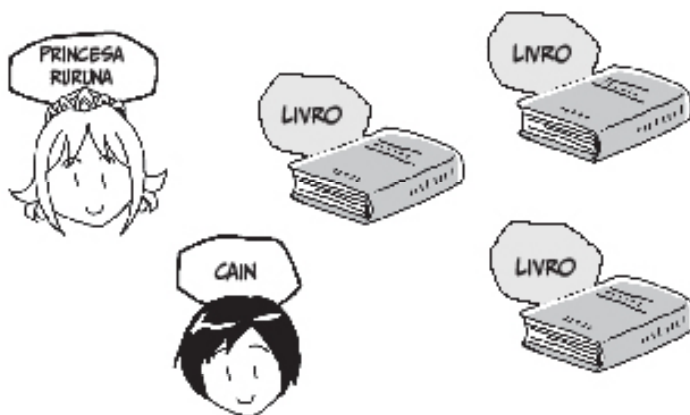
Q1

Um membro da equipe gerencia vários clientes. Um cliente nunca será contactado por mais de um membro da equipe.



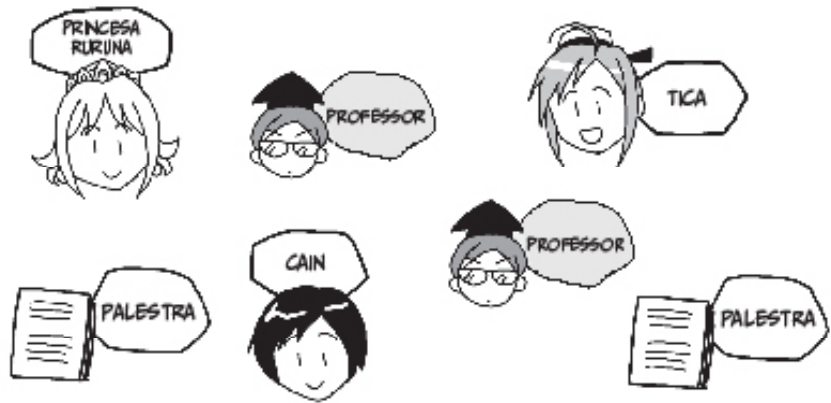
Q2

Uma pessoa pode consultar diversos livros. Estes podem ser lidos por vários alunos em momentos diferentes.



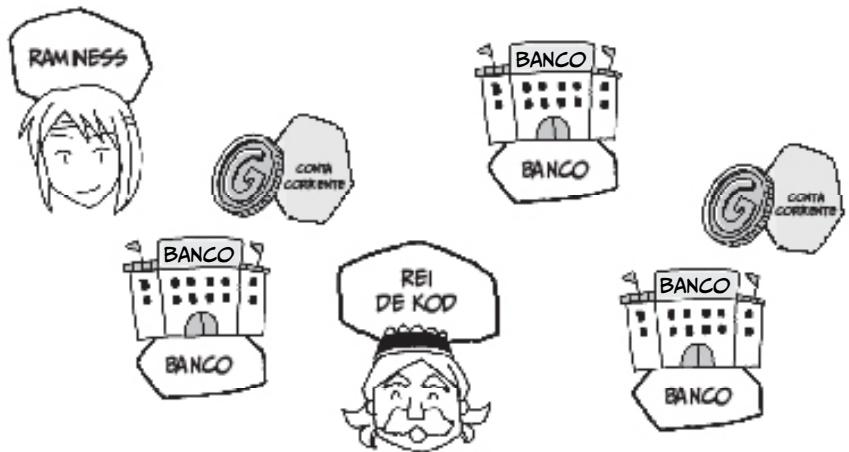
Q3

Cada aluno participa de diversas palestras. Cada palestra é assistida por diversos alunos. Um professor dá diversas palestras. Cada palestra é dada por um professor.



Q4

Cada cliente pode abrir várias contas correntes. Cada conta é aberta por um cliente. Cada banco gerencia diversas contas. Cada conta é gerenciada por um banco.



Lembre-se de que a análise baseada no modelo E-R não necessariamente apresenta um resultado “correto”. Pode haver muitas formas de organizar logicamente os dados para refletir condições reais.

Normalizando uma tabela



A princesa Ruruna e o Cain aprenderam sobre normalização, o processo de tabular dados do mundo real em um banco de dados relacional. É necessário normalizar dados para gerenciar apropriadamente um banco de dados relacional. Resumimos normalização aqui (os campos sombreados são chaves primárias).

FORMA DESNORMALIZADA

Código do Relatório	Data	Código de destino de exportação	Nome de destino de exportação	Código do Produto	Nome do Produto	Preço Unitário	Quantidade
---------------------	------	---------------------------------	-------------------------------	-------------------	-----------------	----------------	------------

PRIMEIRA FORMA NORMAL

Código do Relatório	Data	Código de destino de exportação	Nome de destino de exportação
---------------------	------	---------------------------------	-------------------------------

Código do Relatório	Código do Produto	Nome do Produto	Preço Unitário	Quantidade
---------------------	-------------------	-----------------	----------------	------------

SEGUNDA FORMA NORMAL

Código do Relatório	Data	Código de destino de exportação	Nome de destino de exportação
---------------------	------	---------------------------------	-------------------------------

Código do Relatório	Código do Produto	Quantidade
---------------------	-------------------	------------

Código do Produto	Nome do Produto	Preço Unitário
-------------------	-----------------	----------------

TERCEIRA FORMA NORMAL

Código do Relatório	Data	Código de destino de exportação
---------------------	------	---------------------------------

Código de destino de exportação	Nome de destino de exportação
---------------------------------	-------------------------------

Código do Relatório	Código do Produto	Quantidade
---------------------	-------------------	------------

Código do Produto	Nome do Produto	Preço Unitário
-------------------	-----------------	----------------

A forma desnormalizada é uma tabela na qual itens que aparecem mais de uma vez não foram removidos. Vimos que não se pode gerenciar bem dados usando este tipo de tabela em um banco de dados relacional. Consequentemente, é preciso dividir a tabela.

A primeira forma normal refere-se a uma tabela simples, bidimensional, resultante da divisão da original desnormalizada. Pode ser considerada como uma tabela com um item em cada célula. A tabela é dividida para que nenhum item apareça mais de uma vez.

A segunda forma normal refere-se a uma tabela na qual uma chave que pode identificar dados determina os valores de outras colunas. Aqui, é a chave primária que determina valores em outras colunas.

Em um banco de dados relacional, um valor é chamado de funcionalmente dependente se ele determinar valores em outras colunas. Na segunda forma normal, a tabela é dividida para que valores em outras colunas sejam funcionalmente dependentes da chave primária.

Na terceira forma normal, uma tabela é dividida para que um valor não seja determinado por nenhuma chave não-primária. Em um banco de dados relacional, um valor é chamado de transitivamente dependente se ele determinar valores em outras colunas indiretamente, o que é parte de uma operação funcionalmente dependente. Na terceira forma normal, a tabela é dividida para que valores transitivamente dependentes sejam removidos.



QUESTÕES

É importante ser capaz de criar uma tabela de banco de dados relacional para diversas situações, então vamos examinar alguns exemplos de tabelas normalizadas. Determine como a tabela foi normalizada em cada caso abaixo. As respostas estão na página 82.

Q5

A tabela a seguir gerencia empréstimos de livros como o exemplo em Q2. Para qual estágio ela foi normalizada?

Código de Empréstimo	Data	Código do Aluno	Nome do Aluno	Endereço de Aluno	Curso	Ano de Ingresso
----------------------	------	-----------------	---------------	-------------------	-------	-----------------

ISBN	Nome do Livro	Nome do Autor	Data de Publicação	Número de Páginas
------	---------------	---------------	--------------------	-------------------

Código de Empréstimo	ISBN	Quantidade
----------------------	------	------------

Q6

A tabela a seguir também mostra uma situação de empréstimo de livros. Para qual estágio ela foi normalizada?

Código de Empréstimo	Data	Código de Aluno
----------------------	------	-----------------

Código de Aluno	Nome de Aluno	Endereço de Aluno	Curso	Ano de Ingresso
-----------------	---------------	-------------------	-------	-----------------

ISBN	Nome do Livro	Nome do Autor	Data de Publicação	Número de Páginas
------	---------------	---------------	--------------------	-------------------

Código de Empréstimo	ISBN	Quantidade
----------------------	------	------------

Q7

A tabela a seguir mostra as vendas mensais para cada membro da equipe. Cada departamento contém vários membros. Um membro da equipe pode fazer parte de apenas um departamento. Normalize essa tabela na terceira forma normal.

Código de Membro da Equipe	Nome de Membro da Equipe	Mês	Vendas do Membro	Código de Departamento	Nome do Departamento
----------------------------	--------------------------	-----	------------------	------------------------	----------------------



Q8

A tabela a seguir representa um sistema de recebimento de pedidos. Normalize-a na terceira forma normal. No entanto, processe um cliente por código de pedido. Você pode processar diversos produtos baseado em um código de pedido. Além disso, um código de pedido deve corresponder a apenas um representante.

Código de Pedido	Data	Código de Cliente	Nome do Cliente	Código do Produto	Nome do Produto	Preço Unitário	Código do Representante	Nome do Representante	Quantidade
------------------	------	-------------------	-----------------	-------------------	-----------------	----------------	-------------------------	-----------------------	------------

Q9

A tabela a seguir representa um sistema de recebimento de pedidos. Normalize-a na terceira forma normal. Suponha que os produtos sejam classificados por código.

Código de Pedido	Data	Código do Cliente	Nome do Cliente	Código do Produto	Nome do Produto	Preço Unitário	Código de Classificação do produto	Nome de Classificação do Produto	Quantidade
------------------	------	-------------------	-----------------	-------------------	-----------------	----------------	------------------------------------	----------------------------------	------------

Passos para projetar um Banco de Dados

Você aprendeu como projetar um banco de dados! No entanto, você tem que fazer mais do que isso. É preciso projetar uma estrutura de arquivos detalhada dentro do banco de dados e desenvolver métodos para importar e exportar dados. De forma geral, pode-se dividir todo o projeto do banco de dados em três partes: esquemas conceitual, interno e externo.

O esquema conceitual refere-se a um método que retrata o mundo real. Em outras palavras, é uma forma de determinar a estrutura lógica de um banco de dados. O esquema conceitual é realizado levando-se em consideração um entendimento do mundo real baseado no modelo E-R e normalização de tabelas.

O esquema interno refere-se ao banco de dados visto de dentro de um computador. Isso quer dizer que é uma forma de determinar a estrutura física de um banco de dados. O esquema interno é projetado após a criação de um método para fazer pesquisas no banco de dados rapidamente.

O esquema externo refere-se a um banco de dados como ele é visto pelos usuários ou aplicativos. O esquema externo é projetado após a criação de dados necessários para os aplicativos.



A Princesa Ruruna e o Cain criaram um banco de dados com foco no esquema conceitual neste capítulo. Eles estão no processo de melhoria do banco de dados.

Agora que você completou o projeto básico de um banco de dados, vamos examinar seu uso no próximo capítulo.

Resumo



- Um modelo E-R é usado para analisar entidades e relacionamentos.
- Relacionamentos entre entidades podem ser um-para-um, um-para-muitos e muitos-para-muitos.
- Os dados em uma tabela devem ser normalizados antes que possam ser usados para criar um banco de dados relacional.
- O projeto de um banco de dados pode ser dividido em três tipos: esquemas conceitual, interno e externo.

Respostas

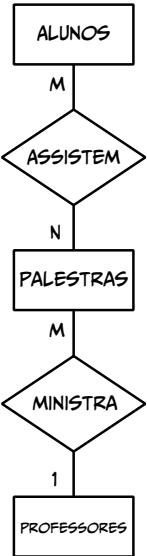
Q1



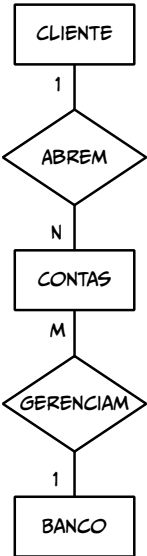
Q2



Q3



Q4



Q5 Segunda forma normal

Q6 Terceira forma normal

Q7

Código de membro de equipe	Mês	Vendas do membro
----------------------------	-----	------------------

Código de membro de equipe	Nome de membro de equipe	Código de departamento
----------------------------	--------------------------	------------------------

Código de departamento	Nome de departamento
------------------------	----------------------

Q8

Código de pedido	Data	Código do cliente	Código do representante
------------------	------	-------------------	-------------------------

Código do cliente	Nome do cliente
-------------------	-----------------

Código de pedido	Código do produto	Quantidade
------------------	-------------------	------------

Código do produto	Nome do produto	Preço unitário
-------------------	-----------------	----------------

Código do representante	Nome do representante
-------------------------	-----------------------

Q9

Código de pedido	Data	Código de cliente
------------------	------	-------------------

Código de cliente	Nome de cliente
-------------------	-----------------

Código de pedido	Código do produto	Quantidade
------------------	-------------------	------------

Código do produto	Código de classificação do produto	Nome do produto	Preço unitário
-------------------	------------------------------------	-----------------	----------------

Código de classificação do produto	Nome de classificação do produto
------------------------------------	----------------------------------

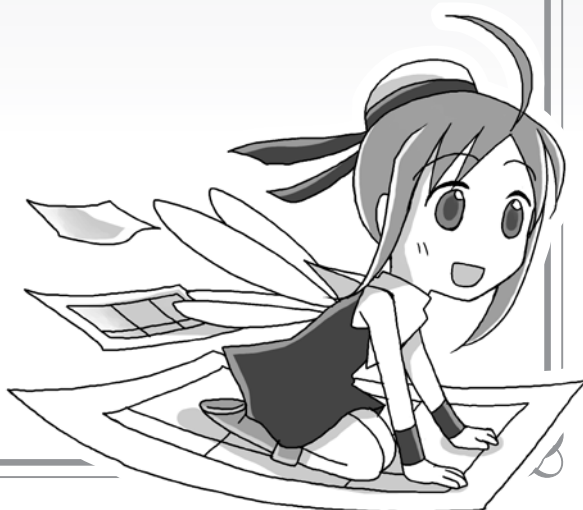
PROJETANDO UM BANCO DE DADOS

Neste capítulo, você aprendeu como criar um banco de dados relacional. No entanto, existem outros métodos de projeto. Usabilidade e eficiência de um banco de dados dependem de um método de análise e de projeto. Portanto, é importante criar um banco de dados apropriado no estágio de projeto.

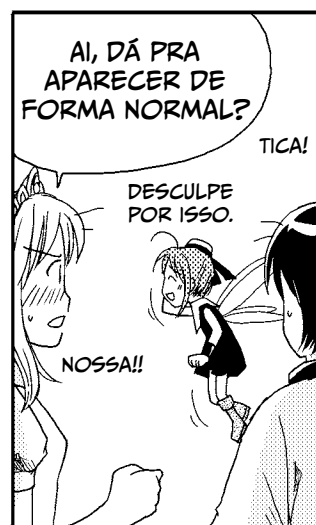
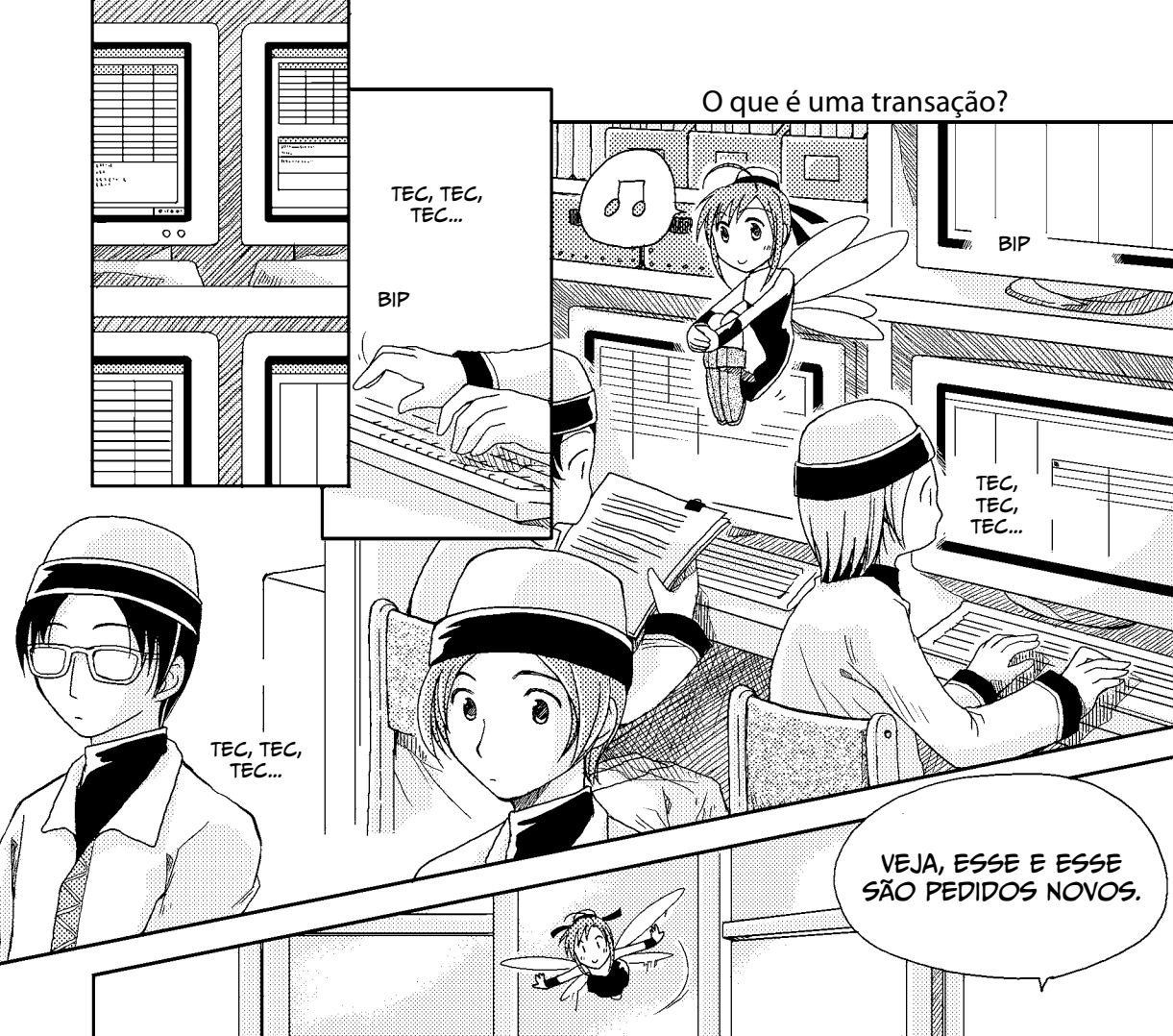
No estágio de projeto do banco de dados, é preciso executar várias tarefas além do design. Por exemplo, você precisa considerar os tipos de dados para usar nas tabelas. Você também pode precisar especificar colunas utilizando valores numéricos, moedas e sequências de caracteres. Além disso, é preciso desenvolver um método de pesquisa para possibilitar pesquisas rápidas. Às vezes, você deve criar um design enquanto pensa na organização física de arquivos. E você tem que controlar quais usuários podem acessar o banco de dados para garantir a segurança. Há muitos fatores que devem ser considerados ao projetar um banco de dados. Vamos examinar alguns desses fatores nos capítulos seguintes.

5

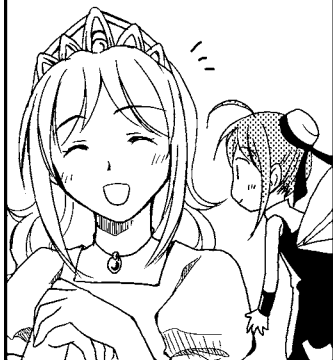
Vamos usar um Banco de Dados!



O que é uma transação?



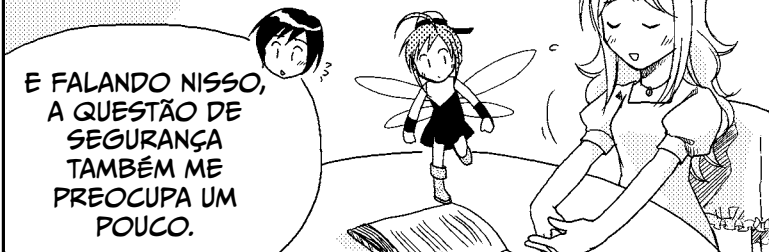
NA VERDADE,
EU TENHO QUE
AGRADECER...



MAS AINDA
TEMOS
MUITO PARA
APRENDER.

POR EXEMPLO, ME PERGUNTO
POR QUE UM BANCO DE
DADOS AINDA PODE OPERAR
QUANDO TANTOS USUÁRIOS O
ACESSAM AO MESMO TEMPO.

E FALANDO NISSO,
A QUESTÃO DE
SEGURANÇA
TAMBÉM ME
PREOCUPA UM
POUCO.



APARENTEMENTE,
VOCÊ TEM ALGUMAS
PREOCUPAÇÕES EM
RELAÇÃO A SEU BANCO
DE DADOS.



ACHO QUE
SIM.

BOM, PARA
ENTENDER
MELHOR ESSAS
QUESTÕES,

EU FIZ UMA
PEQUENA
PESQUISA.



AHAM!

AH É?

O TÍTULO DE MINHA
APRESENTAÇÃO É:

COMO UM BANCO DE
DADOS PODE DEIXAR
UM GRANDE NÚMERO DE
USUÁRIOS ACESSÁ-LO
SIMULTANEAMENTE?



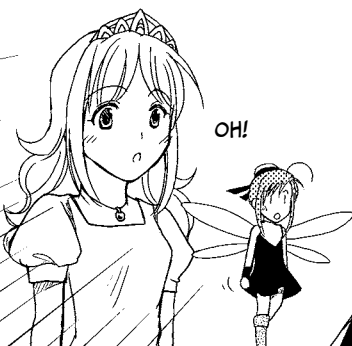
EU ATÉ PREPAREI
UMAS ILUSTRAÇÕES
PARA AJUDAR EM
SEU ENTENDIMENTO!

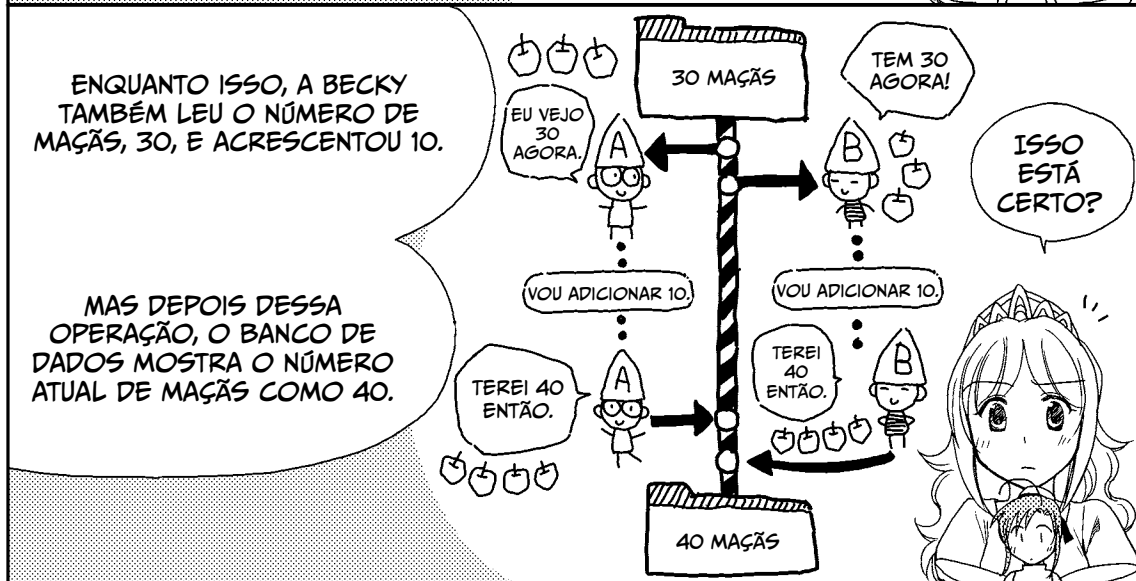
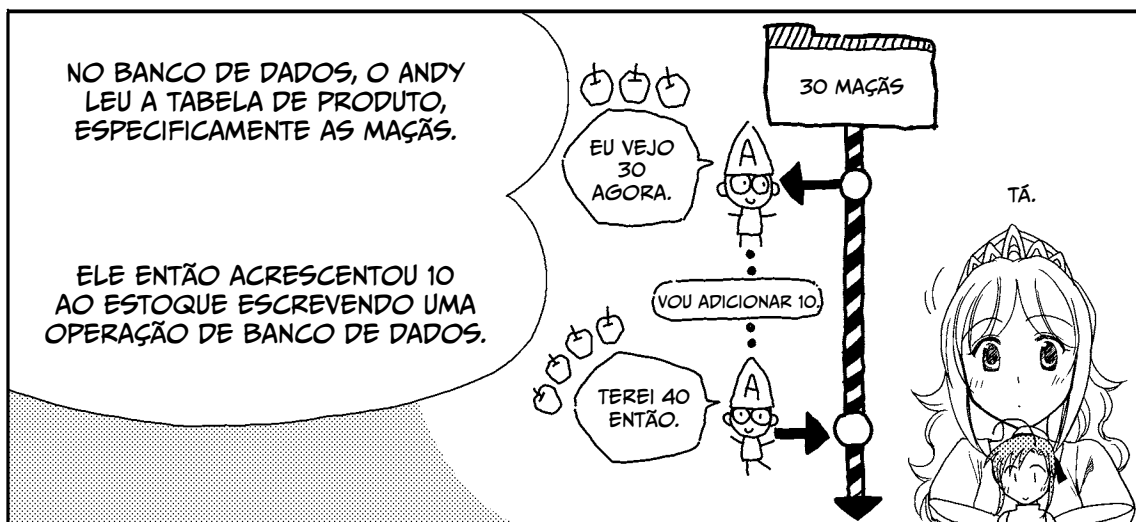
UAU, QUE
ÓTIMO.

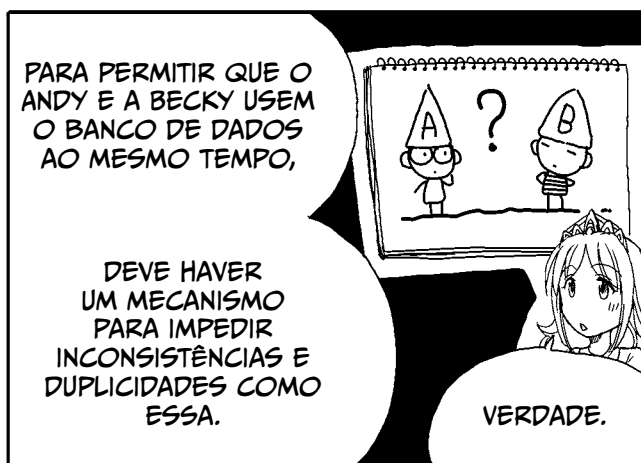


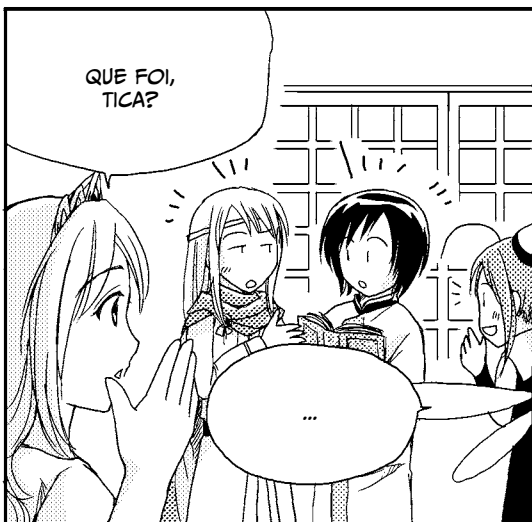
ADORO UM BOM
SHOW!

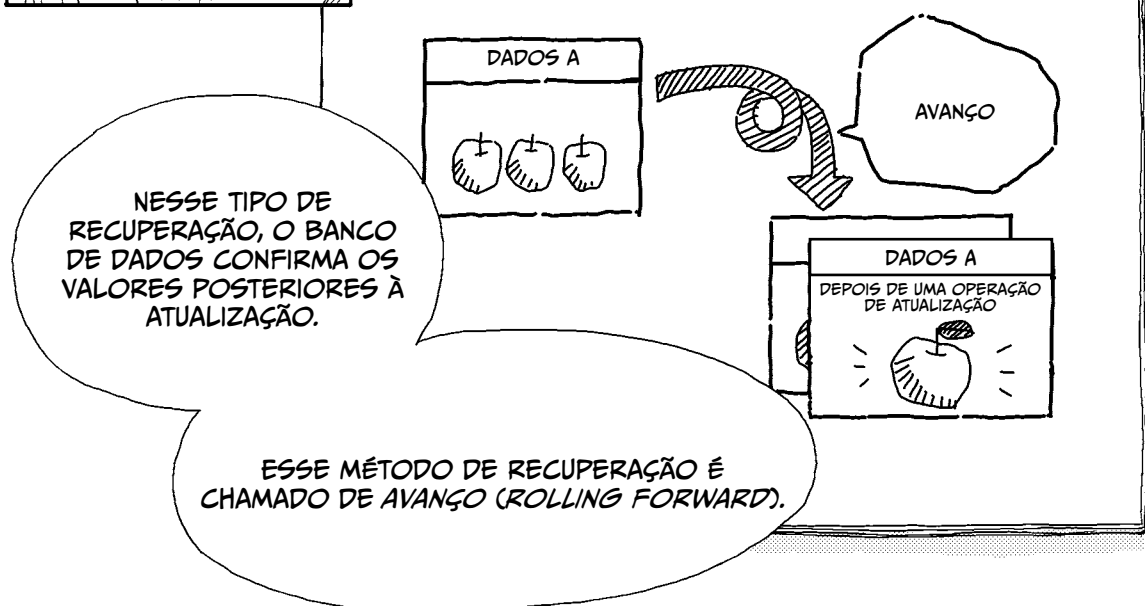
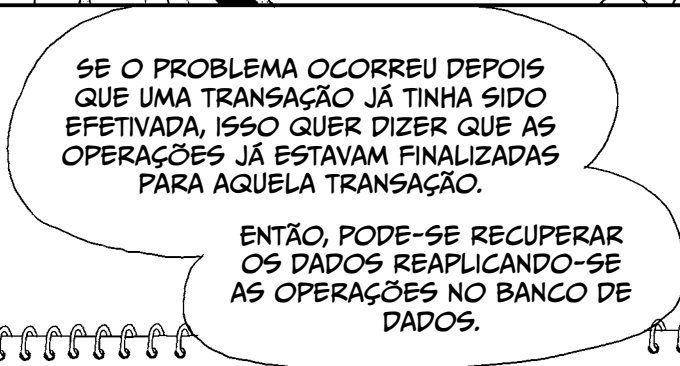
OH!

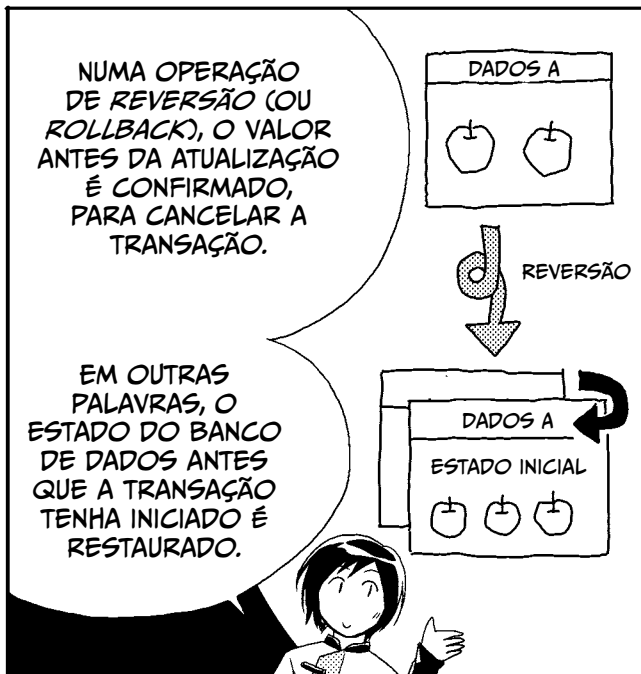


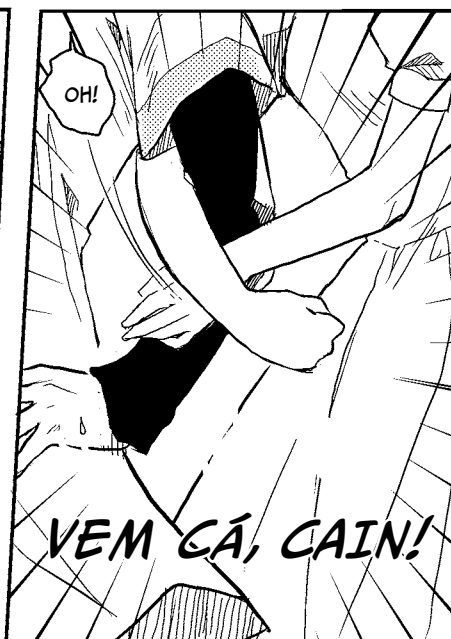
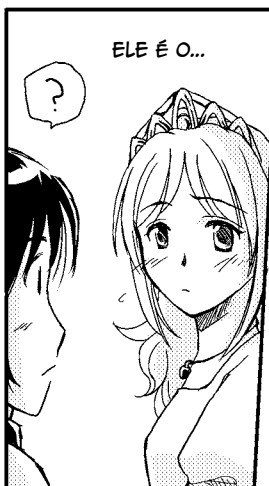














EU E CAIN FICAREMOS JUNTOS
PARA SEMPRE, E COM O
PODER DO NOSSO BANCO DE
DADOS,



HÃ?!
E...EU?!

VAMOS GARANTIR
QUE O REINO DE KOD
PROSPERE!



EM NOME DOS
CÉUS... NÃO...
NÃO FAÇA ISSO
COMIGO...

SINTO
MUITO.

QUE
DECEPÇÃO!

VOCE PREFERE
ALGUÉM
HUMILDE COMO
O CAIN?

BEM, SIM,
QUER DIZER,
PERDOE-ME.

POR QUE
VOCE ESTÁ SE
DESCULPANDO,
CAIN?

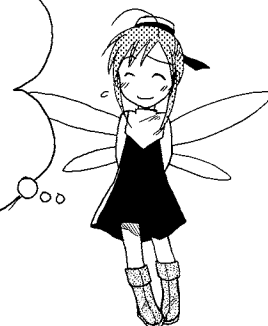
AH, NÃO,
EU NÃO
DEVIA, ME
DESCULPE...

FIQUE
COMIGO
PARA
SEMPRE,
CAIN.

SIM, SIM,
VOSSA
ALTEZA!

POR QUE, OH, POR
QUÊ?

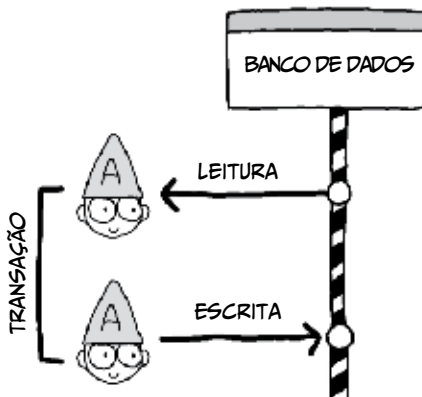
ELES FORMAM
UM ÓTIMO
CASAL.



Propriedades das transações



A pesquisa de Cain mostrou que usuários de um banco de dados podem pesquisar, inserir, atualizar e excluir dados. Um conjunto de operações bem-sucedidas efetuadas por um único usuário é chamado de transação.



Quando se compartilha um banco de dados, é importante garantir que múltiplas transações possam ser processadas sem causar conflitos nos dados. É também importante proteger os dados para evitar inconsistências na eventualidade de falhas durante o processamento de uma transação. Nesse sentido, listamos na tabela seguinte as propriedades que uma transação precisa apresentar, que de forma memorável formam a palavra ACID.

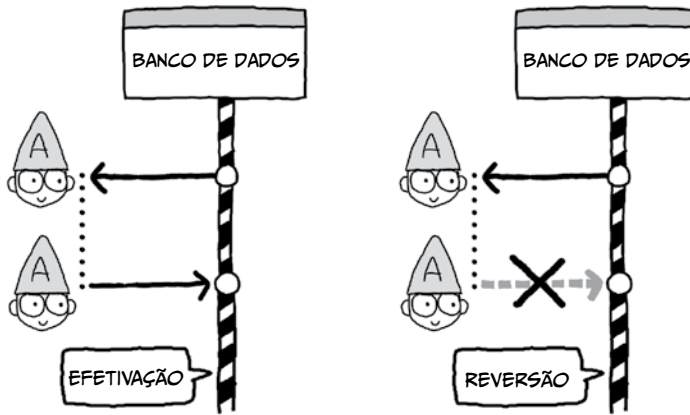
PROPRIEDADES OBRIGATÓRIAS PARA UMA TRANSAÇÃO

Propriedade	Significado	Descrição
A	Atomicidade	Uma transação precisa sempre terminar com uma operação de efetivação ou de reversão.
C	Consistência	Processar uma transação nunca pode resultar em perda de consistência no banco de dados.
I	Isolamento	Mesmo quando transações são processadas simultaneamente, os resultados precisam ser os mesmos de um processamento sequencial.
D	Durabilidade	O conteúdo de uma transação completa não pode ser afetado por falhas.

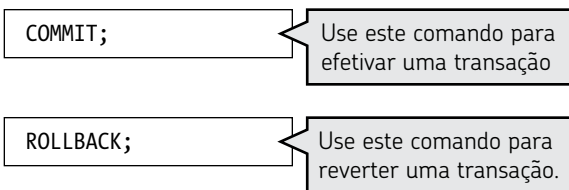
Vamos examinar cada uma dessas propriedades em detalhes.

ATOMICIDADE

A primeira propriedade obrigatória para uma transação, atomicidade, significa que uma transação precisa terminar com uma efetivação ou com uma reversão para manter o banco de dados livre de inconsistências. Em resumo, ou todas as ações da transação são concluídas, ou todas são canceladas. Uma efetivação finaliza as operações de uma transação. Uma reversão as cancela.



Em alguns casos, uma efetivação ou uma reversão são efetuadas automaticamente. Pode-se também especificar qual das duas deve ser utilizada. Por exemplo, pode-se ordenar uma reversão se um erro ocorrer. Para efetuar essas operações, utilizam-se os comandos COMMIT (efetivar) ou ROLLBACK (reverter) da SQL.



QUESTÕES

Responda a estas questões para ver quão bem entendeu a atomicidade. As respostas estão na página 167.

Q1

Escreva um comando SQL que possa ser usado para finalizar uma transação.

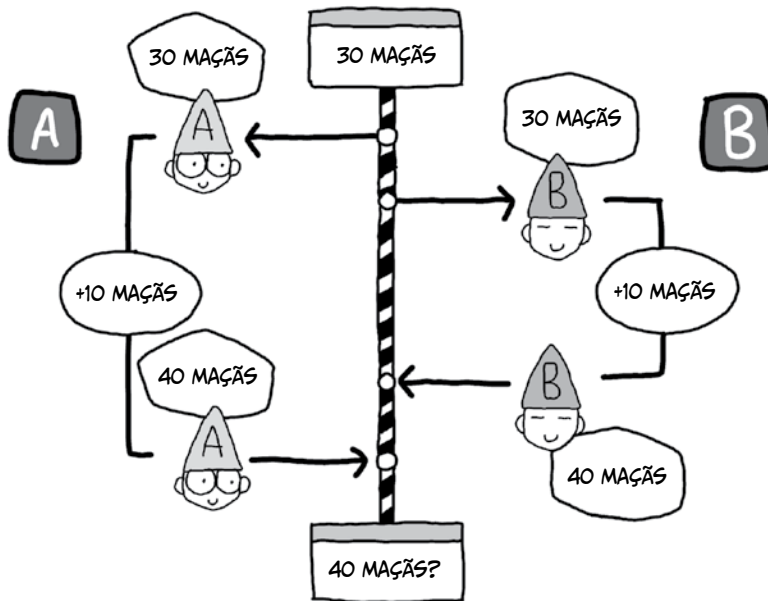
Q2

Escreva um comando SQL que possa ser usado para cancelar uma transação.

CONSISTÊNCIA

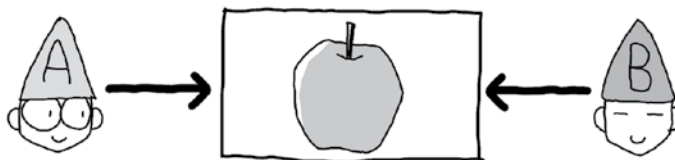
Uma transação não deve gerar erros. Se o banco de dados estava consistente antes do processamento de uma transação, ele precisa continuar consistente depois que a transação se encerra.

O Cain deu como exemplo Andy e Becky, cada um tentando adicionar 10 maçãs em um total de 30 maçãs. Em vez de obter o valor correto de 50 maçãs, o banco de dados mostra um total de 40. Este tipo de erro é chamado de atualização perdida.



Quando transações são processadas simultaneamente, mais de uma pode acessar a mesma linha da mesma tabela ao mesmo tempo, e conflitos nos dados podem ocorrer.

Tabelas e linhas sujeitas a operações em uma transação são chamadas de recursos. Em um banco de dados, transações devem ser capazes de acessar os mesmos recursos simultaneamente sem criar inconsistências.



ISOLAMENTO

Quando duas ou mais transações simultâneas produzem os mesmos resultados que seriam obtidos se elas tivessem sido processadas em momentos diferentes, diz-se que elas são serializáveis. A propriedade de isolamento exige que as transações sejam serializáveis, o que serve como proteção contra erros.

Para conseguir que as transações sejam serializáveis, é preciso ter controle sobre aquelas que ocorrem simultaneamente. O método mais comum utilizado para esse propósito é o controle baseado em travas (bloqueios). Uma trava compartilhada é usada quando se leem os dados, enquanto uma trava exclusiva é usada quando se gravam dados.



Quando uma trava compartilhada está em uso, outro usuário pode aplicar uma trava compartilhada em outras transações, mas não uma trava exclusiva. Quando uma trava exclusiva está aplicada, outro usuário não pode aplicar nem uma trava compartilhada nem uma exclusiva em outras transações. A tabela a seguir resume o relacionamento entre travas compartilhadas e exclusivas.

RELACIONAMENTO DE COEXISTÊNCIA ENTRE TIPOS DE TRAVA (BLOQUEIO)

	Trava compartilhada	Trava exclusiva
Trava compartilhada	SIM	NÃO
Trava exclusiva	NÃO	NÃO



QUESTÕES

Você entendeu as travas? Responda às questões e verifique as respostas na página 167.

Q3

Quando Andy aplica uma trava compartilhada, Becky pode fazer o mesmo?

Q4

Quando Andy aplica uma trava exclusiva, Becky pode aplicar uma compartilhada?

Q5

Quando Andy aplica uma trava compartilhada, Becky pode aplicar uma exclusiva?

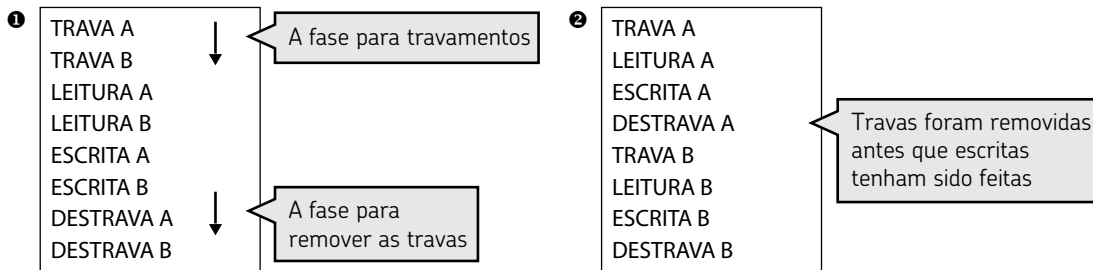
Q6

Quando Andy aplica uma trava exclusiva, Becky pode fazer o mesmo?

TRAVAMENTO EM DUAS FASES

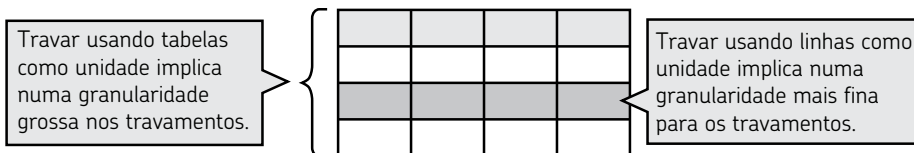
Para ter certeza de que as transações são serializáveis, precisamos obedecer certas regras para aplicar e remover travas. Uma dessas regras é o travamento em duas fases - para cada transação, duas fases devem ser implementadas: uma para aplicar travas, e outra para removê-las.

Por exemplo, suponha que existam os recursos A e B, ambos sujeitos a travamento. A transação ❶ observa a regra do travamento em duas fases, mas a ❷ não. Serialização só pode ser obtida se todas as transações atendem a regra do travamento em duas fases.



GRANULARIDADE DE TRAVA

Existem diversos recursos que podem ser travados (bloqueados). Por exemplo, pode-se travar dados usando tabelas ou linhas como unidade. A extensão do travamento do recurso é conhecida como granularidade. Granularidade grossa ocorre quando muitos recursos são travados ao mesmo tempo, enquanto que granularidade fina ocorre quando menos são.



Quando a granularidade é grossa (ou alta), o número de travas necessárias por transação é reduzido, o que torna gerenciar a granularidade mais fácil. Por um lado, isso reduz a quantidade de processamento necessária para a CPU que executa o banco de dados. Por outro lado, conforme mais recursos vão sendo travados, o tempo de espera de transações aguardando remoção de travas tende a aumentar. Dessa forma, a quantidade de transações que se pode executar tende a cair quando a granularidade é alta.

Em contraste, quando a granularidade é fina (ou baixa), uma quantidade maior de travas é usada por transação, resultando em mais operações para gerenciar o travamento. Isso resulta em mais processamento sendo imposto à CPU. No entanto, já que menos recursos estão travados, menos tempo será empregado esperando que outras transações removam travas. Assim, o número de transações que se pode efetuar tende a aumentar.



QUESTÕES

Responda às questões, e confira as respostas corretas na página 168.

Q7

A unidade de travamento para recursos foi alterada de tabelas para linhas. O que acontece com a quantidade de transações que se pode efetuar simultaneamente?

Q8

A unidade de travamento para recursos mudou de linha para tabela. O que acontecerá com o número de transações que se pode executar ao mesmo tempo?

OUTROS CONTROLES DE EXECUÇÃO SIMULTÂNEA

Pode-se utilizar travamento para executar com sucesso duas ou mais transações ao mesmo tempo. No entanto, utilizar travas implica no fardo do gerenciamento de travamento, uma vez que deadlocks - momentos em que ações entram em conflito - podem ocorrer. Métodos mais simples de controle de execução simultânea podem ser usados quando se tem uma quantidade menor de transações ou um grande número de operações de leitura. Nesses casos, os seguintes métodos podem ser utilizados:

Controle timestamp

Um rótulo contendo o momento do acesso, conhecido como estampa de tempo (timestamp), é atribuído aos dados acessados durante uma transação. Se outra com uma estampa mais adiantada no tempo já atualizou os dados, a operação não será permitida. Quando uma operação de leitura ou escrita é bloqueada, a transação é revertida.

Controle otimista

Este método permite operações de leitura. Quando se tenta uma operação de escrita, os dados são verificados para ver se qualquer outra transação está ocorrendo. Se outra transação já tiver atualizado os dados, a transação é revertida.

NÍVEIS DE ISOLAMENTO

Em um banco de dados no mundo real, pode-se configurar o nível em que transações serão processadas simultaneamente. Isso é chamado de nível de isolamento.

Na SQL, o comando SET TRANSACTION (configurar transação) pode ser usado para especificar os seguintes níveis de isolamento:

- READ UNCOMMITTED
- READ COMMITTED
- REPEATABLE READ
- SERIALIZABLE

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

Dependendo do nível de isolamento configurado, os seguintes problemas podem ocorrer.

	Leitura suja	Leitura não reproduzível	Leitura fantasma
READ UNCOMMITTED	Possível	Possível	Possível
READ COMMITTED	Não ocorre	Possível	Possível
REPEATABLE READ	Não ocorre	Não ocorre	Possível
SERIALIZABLE	Não ocorre	Não ocorre	Não ocorre

- Uma leitura suja ocorre quando uma segunda transação lê uma linha antes que uma primeira a efetive.
- Uma leitura não reproduzível ocorre quando uma transação lê duas vezes os mesmos dados e obtém valores diferentes.
- Uma leitura fantasma ocorre quando uma transação procura linhas que obedecem um certo critério, mas encontra as linhas erradas devido a alterações feitas por outra transação.

DURABILIDADE

Um banco de dados gerencia dados importantes, de modo que garantir a segurança e durabilidade no caso de falhas é essencial. Segurança é também importante para evitar que usuários não autorizados substituam dados e criem inconsistências.

Em um banco de dados, pode-se configurar permissões relativas a quem pode acessar todo o banco de dados ou as tabelas contidas. O Cain evitou problemas para o banco de dados do reino melhorando a segurança.

Em um banco relacional, o comando GRANT (conceder) é utilizado para conceder permissões para que outros usuários possam efetuar processamento nas tabelas que você cria. Configurar permissões é uma tarefa importante na operação de um banco de dados.

```
GRANT SELECT, UPDATE ON produtos  
TO Departamento_Negocios_Internacionais;
```

Este comando concede permissões para processar os dados.

Pode-se conceder os seguintes privilégios (permissões) com comandos SQL.

PRIVILÉGIOS EM BANCOS DE DADOS

Comando	Resultado
SELECT	Permite que usuários pesquisem linhas numa tabela.
INSERT	Permite que usuários acrescentem linhas numa tabela.
UPDATE	Permite que usuários atualizem linhas numa tabela.
DELETE	Permite que usuários apaguem linhas numa tabela.
ALL	Concede todos os privilégios.

Conceder privilégios com WITH GRANT OPTION permite que o usuário conceda privilégios que recebeu a outros. Como o comando mostrado abaixo, o departamento de negócios internacionais pode permitir a outros usuários a atualização do banco de dados.

```
GRANT SELECT, UPDATE ON produtos  
TO Departamento_Negocios_Internacionais WITH GRANT OPTION;
```

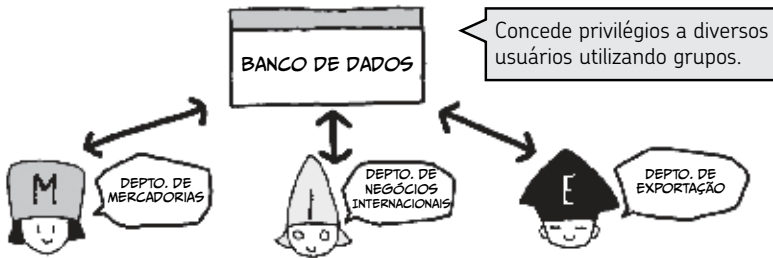
O usuário que recebe os privilégios pode concedê-los a outros.

Pode-se também retirar privilégios. Para fazer isso, utiliza-se o comando REVOKE.

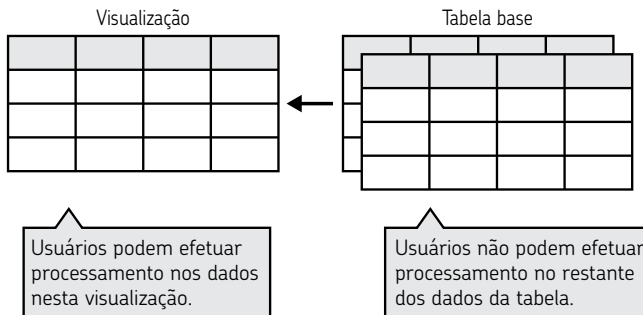
```
REVOKE SELECT, UPDATE ON produtos  
FROM Departamento_Negocios_Internacionais;
```

Este comando revoga os privilégios concedidos ao usuário.

Alguns produtos de banco de dados podem agrupar uma série de privilégios e concedê-los a diversos usuários ao mesmo tempo. O agrupamento pode facilitar o gerenciamento de permissões.



Utilizar visualizações, como descrito na página 117, permite um gerenciamento ainda mais controlado para melhorar a segurança. Primeiro, extraia parte de um banco de dados para criar uma visualização. Configurar privilégios para ela significa que esses também foram aplicados na parte dos dados que foi selecionada para a visualização.



QUESTÕES

Tente responder às seguintes questões sobre durabilidade. As respostas estão na página 168.

Q9

Escreva um comando SQL que permita ao departamento de exportações a pesquisa nos dados da tabela de produtos.

Q10

Crie um comando SQL para revogar o privilégio dado ao departamento de negócios internacionais de excluir dados da tabela de produtos.

Q11

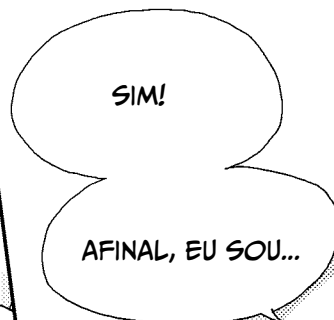
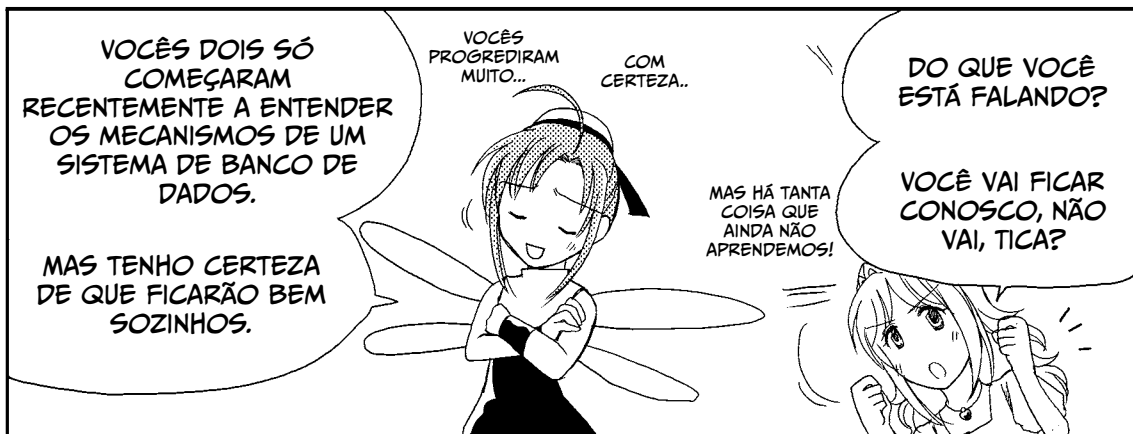
Privilégios foram configurados como segue na tabela de produtos criada pelo administrador. Responda SIM ou NÃO em cada célula da tabela abaixo para indicar a presença ou ausência de um privilégio para cada departamento, respectivamente.

```

GRANT ALL produtos TO Departamento_Negocios_Internacionais;
GRANT INSERT, DELETE ON produtos TO Departamento_Mercadorias;
GRANT UPDATE, DELETE ON produtos TO Departamento_Exportacoes;

```

	Pesquisar	Inserir	Atualizar	Excluir
Depto. negócios internacionais				
Depto. mercadorias				
Depto. exportação				



A FADA DOS
BANCOS DE
DADOS!

TADAM!

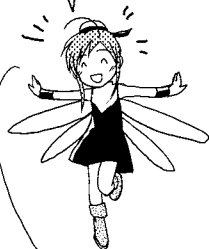


EU SÓ VIM ME
DESPEDIR HOJE.

HE-HE-HE

MAS NÃO PUDE
EVITAR FAZER MAIS
QUE ISSO.

ESSE TEMPO QUE EU
PASSEI COM VOCÊS
FOI CURTO, MAS FELIZ!



VUP!

TICA...

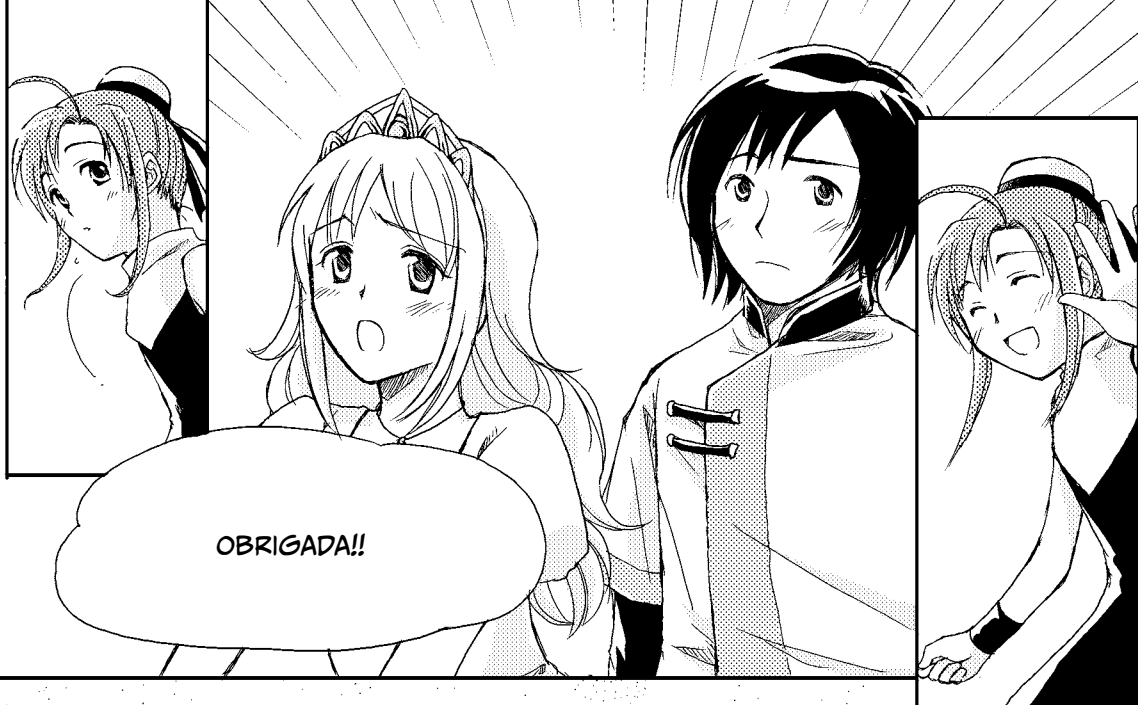
TICA, ESPERE!

EU TENHO
QUE...

TICA FIQUE!



TICA!!



OBRIGADA!!

ELA SE FOI.

NÃO GOSTO
DE VER VOCÊ
TÃO TRISTE,
PRINCESA.

TEMOS QUE
IMPLEMENTAR O
CONHECIMENTO
QUE A TICA NOS
OFERECEU

EM UM SISTEMA
REAL.

AH, SIM. VOCÊ
ESTÁ CERTO.

Dias depois...

ESTÁ indo BEM SEU
LIVRO SOBRE BANCOS
DE DADOS, PRINCESA?

SIM!

EU ESTOU
FACILITANDO AS
COISAS PARA TODO
MUNDO ENTENDER.

VOCÊ QUER
DAR UMA
OLHADA?

CLARO!

TÃO FOFINHO... ♥

É UMA BOA
IDEIA FAZÊ-LO
EM FORMA DE
QUADRINHOS.

E OS
DESENHOS DO
CAIN ESTÃO
EXCELENTE.

E OLHE!

AQUI! ESTA É A
CAPA.

FABULOSO!



FALANDO DO CAIN, O
REI ESTÁ ESPERANDO
POR VOCÊ PARA
CONVERSAR SOBRE
OS PREPARATIVOS
DO CASAMENTO.



MEU PAI ESTÁ
ESPERANDO?

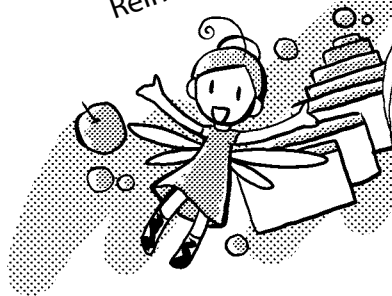


ERA UMA VEZ,

UM PEQUENINO PAÍS
CHAMADO REINO DE
KOD.

UM DIA, DE DENTRO
DE UM LIVRO ANTIGO
SOBRE BANCOS DE
DADOS,

Bancos de Dados no
Reino de Kod



SURTIU UMA
MINÚSCULA MENINA
CHAMADA...

FIM