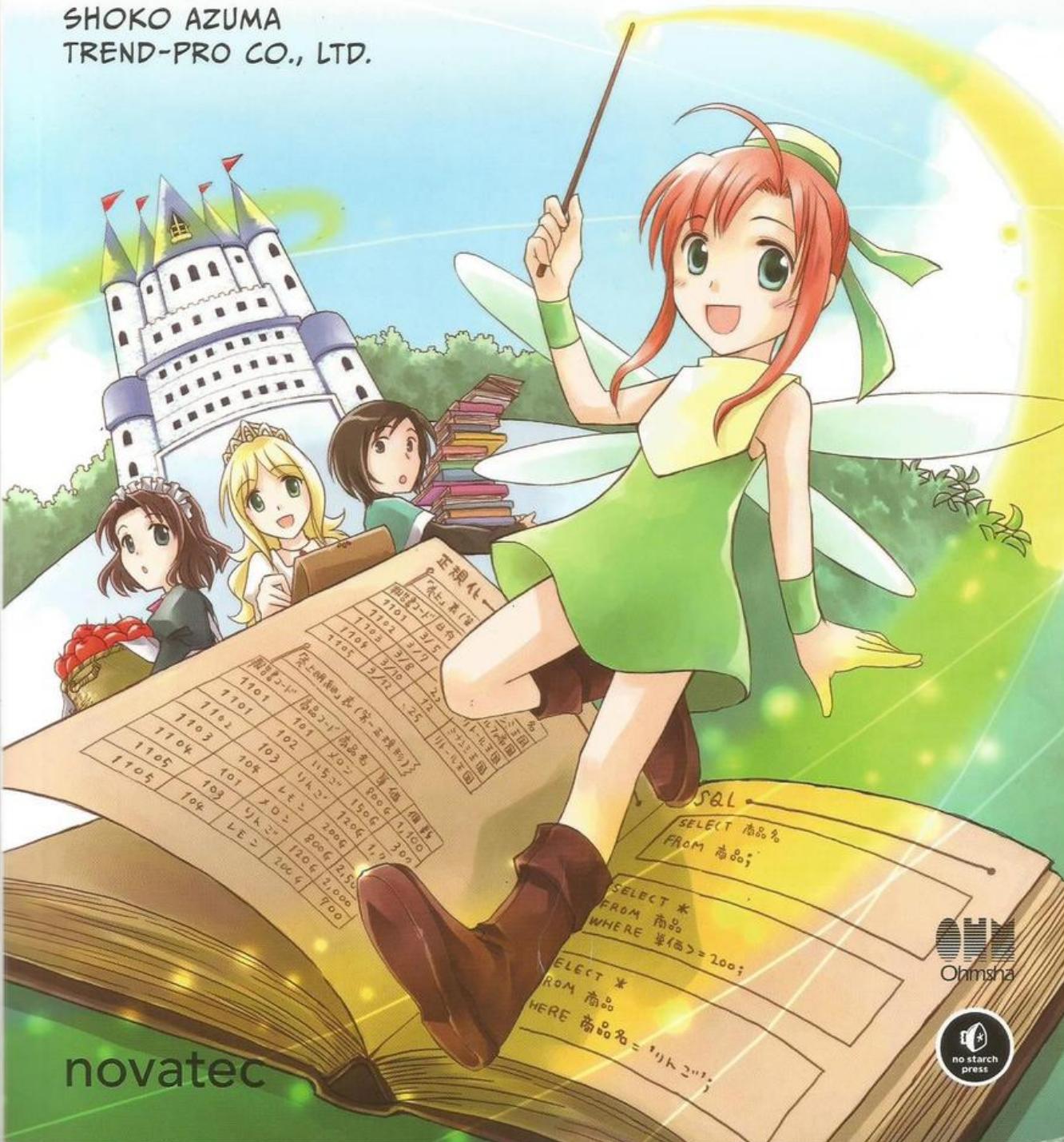


GUIA MANGÁ DE

BANCOS DE DADOS

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



novatec

OHM
Ohmsha

no starch
press

GUIA MANGÁ BANCOS DE DADOS



GUIA MANGÁ 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 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 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:

Maio/2015 Terceira reimpressão

Fevereiro/2013 Segunda reimpressão

Novembro/2010 Primeira reimpressão

Outubro/2009 Primeira edição

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

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

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
quadinhos 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
IG20150528

SUMÁRIO

PREFÁCIO	ix
1	1
O QUE É UM BANCO DE DADOS?	1
Por que precisamos de Bancos de Dados?	2
• 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!	
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	108
Padrões	108
Pesquisas	109
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!	
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
Respostas.....	205
APÊNDICE	
COMANDOS SQL USADOS COM FREQUÊNCIA	207
REFERÊNCIAS	209
ÍNDICE.....	211

PREFÁCIO

Bancos de dados são uma parte crucial de praticamente todos os sistemas de negócios baseados em computadores. Alguns leitores podem estar pensando em introduzir bancos de dados em sua rotina de trabalho. Outros podem ter que de fato desenvolver sistemas de negócios baseados em bancos de dados. Banco de dados é a tecnologia que dá suporte a esses sistemas nos bastidores, e sua verdadeira natureza é difícil de entender.

Este livro foi desenvolvido para que leitores sejam capazes de aprender o básico sobre bancos de dados por meio de uma história em mangá. No final de cada capítulo, exercícios práticos são fornecidos para confirmação e expansão do conhecimento já obtido. Cada capítulo é feito para que os leitores possam adquirir uma compreensão a respeito da tecnologia de bancos de dados enquanto confirmam o quanto eles entenderam sobre os conteúdos.

A estrutura deste livro é a seguinte.

O Capítulo 1 descreve o porquê do uso de banco de dados. Por que ele é necessário? Que tipo de dificuldades você terá se não usar um banco de dados? Você aprenderá as informações básicas que o uso de um banco de dados exige.

O Capítulo 2 apresenta a terminologia básica. Você aprenderá sobre vários modelos de bancos de dados e outros termos relacionados a banco de dados.

O Capítulo 3 explica como criar um banco de dados, especificamente, um banco de dados relacional, o tipo mais comum.

O Capítulo 4 trata de SQL, uma linguagem usada para gerenciar bancos de dados relacionais. O uso de SQL permite a você gerenciar facilmente seus dados.

O Capítulo 5 explica a estrutura do sistema de banco de dados. Já que um banco de dados é um sistema por meio do qual muitas pessoas compartilham dados, você aprenderá como fazer isso de forma segura.

O Capítulo 6 apresenta descrições de aplicações de bancos de dados. Você aprenderá como tipos de sistemas de banco de dados com base na Web e outros são usados.

Este livro foi publicado graças aos esforços conjuntos de muitas pessoas: Shoko Azuma nos desenhos, TREND-PRO na produção e Ohmsha no planejamento, edição, e marketing. Estendo minha profunda gratidão a todos eles.

Espero que este livro seja útil a todos os leitores.

MANA TAKAHASHI

1

O QUE É UM BANCO DE DADOS?



POR QUE PRECISAMOS DE BANCOS DE DADOS?

CLOC
CLOC

REINO DE KOD*

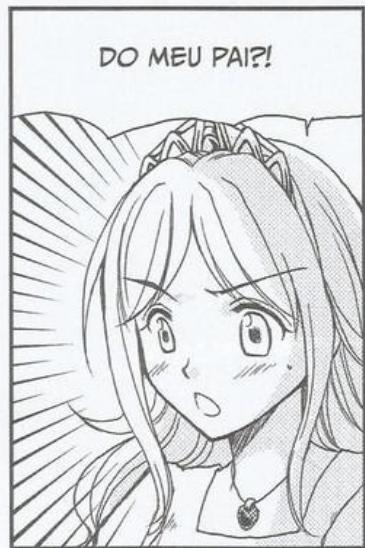
NÃO FAÇA ESSA CARA,

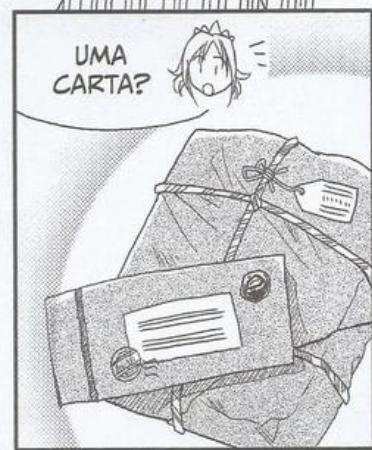
O QUE
VOCÊ
QUERIA...

PRINCESA RURUNA!

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









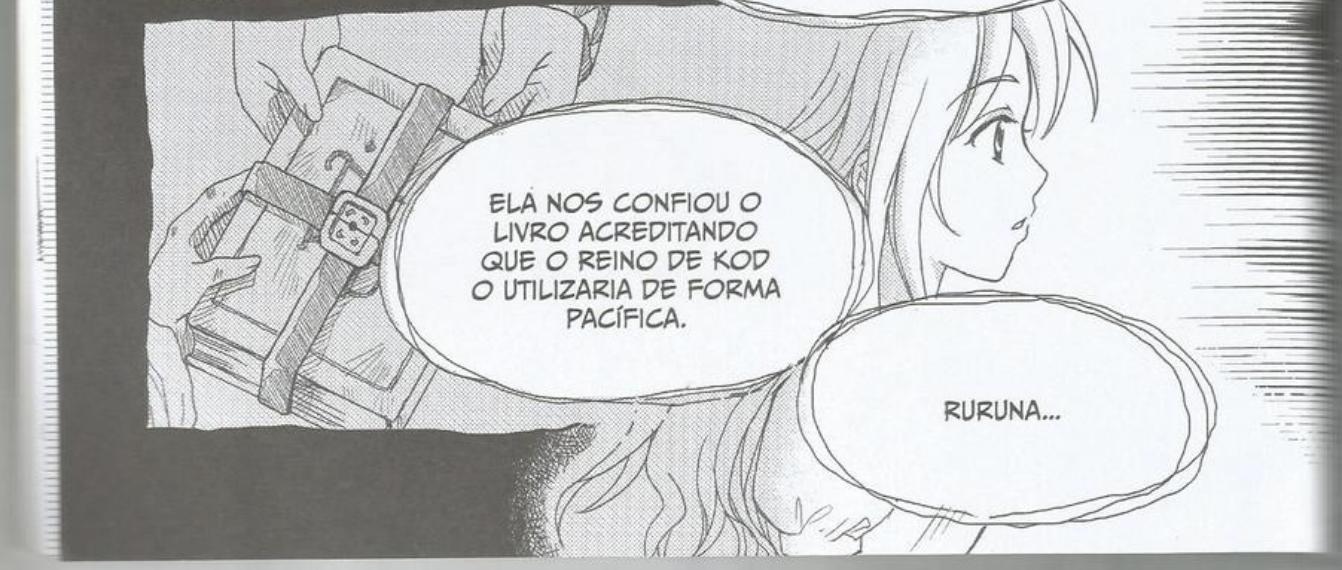
"ENCONTRAMOS UM LIVRO SOBRE UMA TECNOLOGIA INOVADORA EM UMA TERRA DISTANTE QUE VISITAMOS.

A PESSOA QUE NOS DEU ESTE LIVRO DISSE 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.

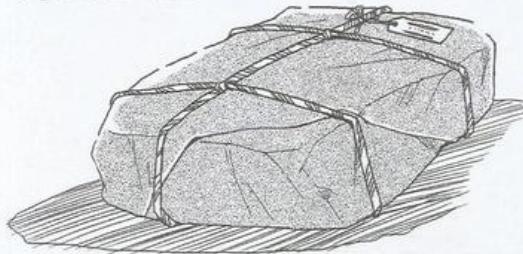
MAS COMO ELE SERÁ USADO DEPENDE DA ÍNDOLE DE QUIER LER O LIVRO.



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!"



QUE??

ESTÁ
TRANCADO.

NÃO CONSIGO ABRIR.

HMM...

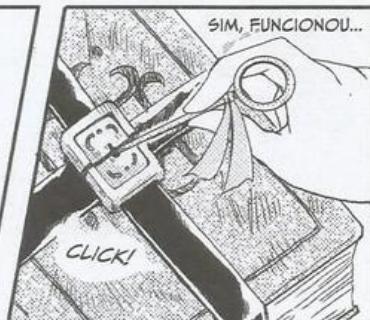
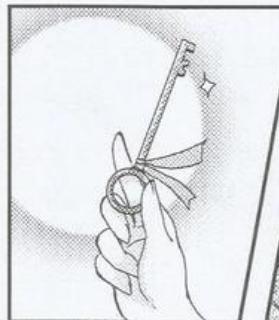
"ESSA É A CHAVE
PARA O LIVRO?"

ESTAVA NO ENVELOPE...



OH, É TÃO VELHO...

RASG



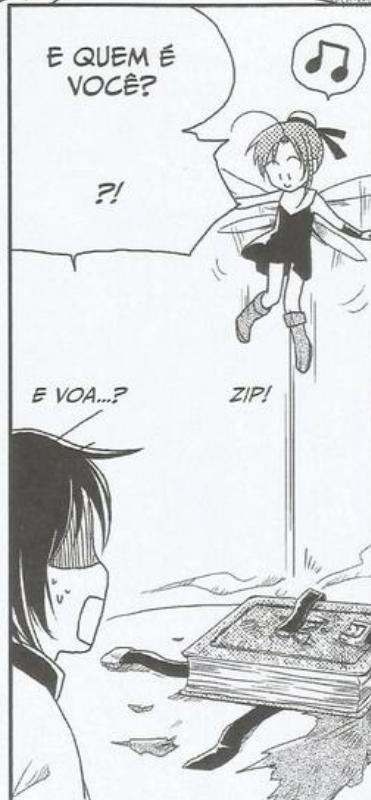
SIM, FUNCIONOU...



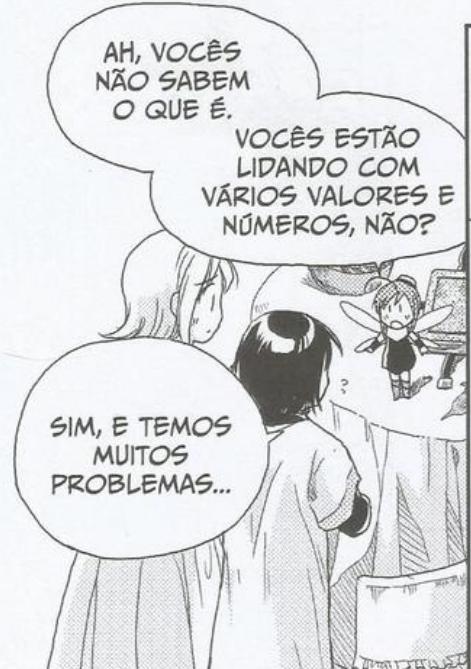


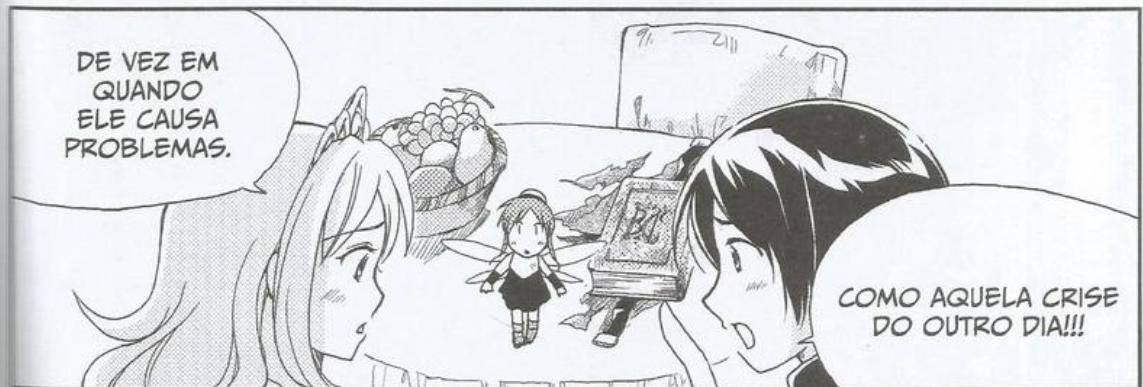
VOCÊ ESTÁ
NO CAS...CAS...
CASTELO DE
KOD.

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

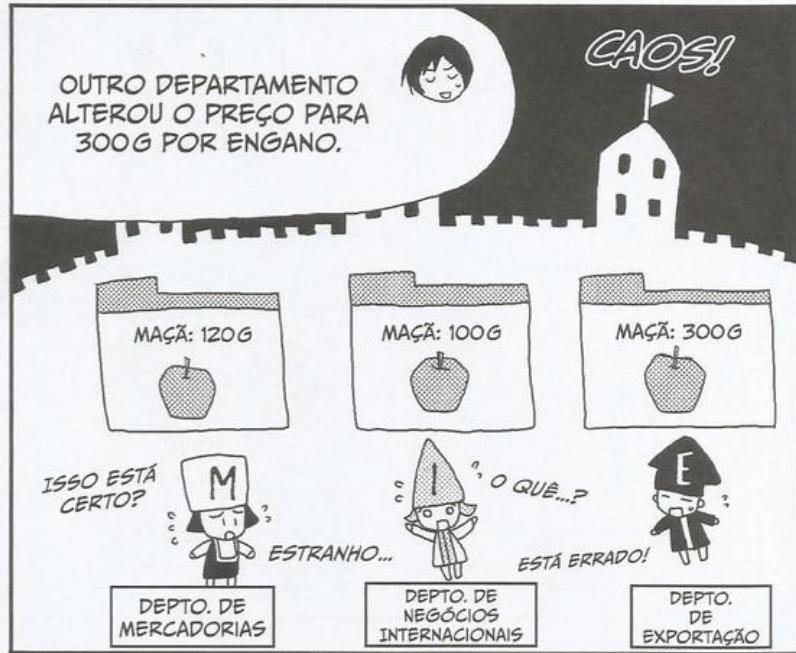












MEU PAI DISSE "VAMOS ABRIR OS POMARES PARA VISITAÇÃO EM BREVE!"

MAS EU SINTO QUE NÃO ESTAMOS PREPARADOS.

ELE É TÃO DESLIGADO COM AS COISAS...

BEM-VINDOS A KOD, O REINO DAS FRUTAS

VAMOS COMEÇAR UM NOVO NEGÓCIO USANDO O SISTEMA ATUAL, HA HA HA!

MESMO QUE COMECEMOS UM NOVO NEGÓCIO, ME PARECE IMPOSSÍVEL USAR OS DADOS DO SISTEMA ATUAL.

VIRARIA UMA BAGUNÇA...

ENTÃO... MINHAS OBRIGAÇÕES OFICIAIS NÃO DIMINUIRIAM EM NADA!

CALMA, POR FAVOR!

QUE RAIVA!

SE VOCÊS COMEÇASSEM UM NOVO NEGÓCIO, TERIAM QUE CRIAR NOVOS ARQUIVOS PARA CADA DEPARTAMENTO.

BOM, PARECE QUE VOCÊS TÊM QUE FAZER ALTERAÇÕES E CONFIRMÁ-LAS A CADA VEZ QUE ALGUMA COISA MUDA, E ISSO PARECE SER DIFÍCIL.

VOCÊS FICAM SOBRECARREGADOS SÓ DE ORGANIZAR OS DADOS

PILHA DE DOCUMENTOS

É AÍ QUE ENTRA O BANCO DE DADOS! UM SISTEMA EM QUE DADOS PODEM SER COMPARTILHADOS POR VÁRIAS PESSOAS AO MESMO TEMPO.

COM ELE, VOCÊS NÃO VÃO PRECISAR MAIS MANTER DADOS INÚTEIS.

GRANDE IDEIA!

SÉRIO?

ENTÃO PODEMOS TER UM SISTEMA MUITO MAIS EFICIENTE QUE O ATUAL?

PARECE ALGO QUE VALE A PENA ESTUDAR, NÃO?

CLARO!

MAS...

JÁ QUE VOCÊ SABE TANTO... NÃO PODERIA FAZER UM PARA MIM?

HEE
HEE

NÃO TENHO UM CORPO FÍSICO, ENTÃO NÃO POSSO USAR COMPUTADORES NO MUNDO REAL.

SINTO MUITO.

QUE PENA...



VOU ENSINAR TUDO PRA VOCÊS. MAS PRECISAM SER BONS ALUNOS!

POR MIM E PELO MEU PAÍS...

VAMOS LÁ!

OH,
PRINCESA!

O QUE ESTÁ ACONTECENDO NO REINO?



O Reino de Kod usa um sistema baseado em arquivos para gerenciar seus dados. Mas parece que o sistema tem alguns problemas. Quais são, exatamente? Vamos ver em detalhes.

O Reino tem três departamentos: o de mercadorias, o de negócios internacionais e o de exportação. O de mercadorias mantém registro de todas as frutas produzidas no país, o de negócios internacionais é responsável pelos países estrangeiros que são parceiros de negócios do reino e o de exportação registra as quantidades de frutas que o reino exporta.

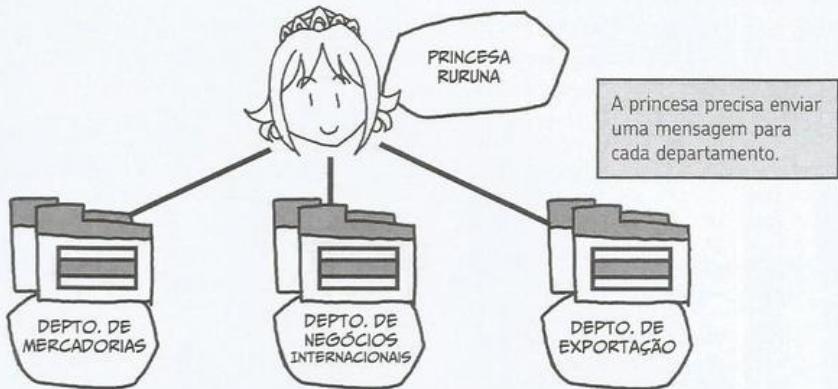


OS DADOS ESTÃO DUPLICADOS

A princesa Ruruna não está satisfeita com o sistema atual. Mas por quê? Cada departamento no reino gerencia seus dados independentemente. Por exemplo, o departamento de mercadorias e o de exportação criam cada um arquivos para gerenciar dados sobre frutas. Dessa forma, os dados são duplicados sem necessidade entre todos os departamentos. Cada um deles gasta tempo inserindo seus dados, e ocupa espaço desnecessário em seus discos rígidos, para só, depois imprimir formulários de checagem, e todo esse processo é um desperdício. Além disso, dados mantidos em um departamento em particular nunca são compartilhados eficientemente com os outros.



Mas isso não é tudo. Esse sistema também cria problemas quando é necessário alterar os dados. Por exemplo, vamos imaginar que o preço das maçãs mude. Para lidar com isso, a princesa Ruruna precisa comunicar esse fato a cada departamento separadamente. Isso não parece inconveniente?



PODEM OCORRER CONFLITOS NOS DADOS

Pode parecer fácil comunicar cada departamento a respeito da mudança de preço das maçãs, mas isso pode criar uma nova série de problemas. Vamos imaginar que a princesa Ruruna comunique aos três departamentos a alteração de preço. No entanto, o departamento de negócios internacionais pode esquecer de fazer a mudança de preço, ou o de exportações pode modificar o valor para 300 G em vez de 120 G. Esses tipos de erro resultam em dados conflitantes entre os departamentos, o que faz com que o conteúdo do sistema de arquivos seja diferente das condições no mundo real. Que problema!



Nome do produto	Preço Unitário
Melão	800 G
Morango	150 G
Maçã	120 G
Limão	200 G

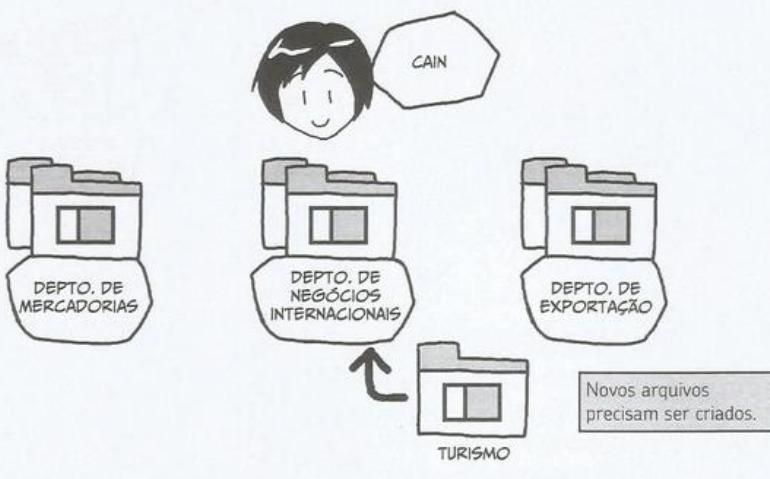
Nome do produto	Preço Unitário
Melão	800 G
Morango	150 G
Maçã	100 G
Limão	200 G

Nome do produto	Preço Unitário
Melão	800 G
Morango	150 G
Maçã	300 G
Limão	200 G

DADOS SÃO DIFÍCEIS DE ATUALIZAR

O sistema atual não apenas gera dados conflitantes, mas também dificulta responder à alterações nos negócios. Por exemplo, digamos que o Rei queira criar um novo empreendimento em turismo. Quando um guia turístico faz um tour pelos pomares e discute as vendas de frutas do Reino, ele vai querer usar os números de vendas mais atualizados.

Mas, infelizmente, o sistema atual não permite necessariamente que os departamentos acessem os dados uns dos outros, já que os arquivos são mantidos independentemente. Para administrar esse novo negócio de turismo, a Princesa Ruruna teria que abrir um novo arquivo só para isso!



ARQUIVO PARA DEPTO. DE MERCADORIAS

Nome do Produto	Preço Unitário
Melão	800 G
Morango	150 G
Maçã	120 G
Limão	200 G

ARQUIVO PARA EMPREENDIMENTOS DE TURISMO

Nome do Produto	Preço Unitário
Melão	800 G
Morango	150 G
Maçã	120 G
Limão	200 G

Ou seja, a cada novo empreendimento, mais dados duplicados acabariam sendo criados. Considerando essas limitações, o sistema atual não é eficiente. Ele dificulta o início de novos projetos e a resposta a alterações no ambiente.

UM BANCO DE DADOS - É A NOSSA SOLUÇÃO!



Bom, por que esse sistema é tão ineficiente? Os problemas todos se originam no gerenciamento isolado e independente dos dados. O que a Ruruna e o Cain devem fazer? Isso mesmo - criar um banco de dados! Eles devem unificar o gerenciamento de dados para o Reino inteiro. Vou mostrar como fazer isso no próximo capítulo.



Gerenciamento de dados centralizado garante que cada departamento obtenha as informações corretas, porque cada departamento efetua consultas a uma única fonte de dados. Que sistema eficiente! Ele impede dados conflitantes e também elimina duplicidade de dados, permitindo fácil introdução e integração de novos departamentos.

COMO USAR UM BANCO DE DADOS

Para criar e usar um banco de dados, deve-se entender seus desafios peculiares. Primeiro, o banco de dados será usado por muitas pessoas, então será necessária encontrar uma maneira para que elas possam acrescentar e extrair dados facilmente. Precisa ser um método que seja fácil para todo mundo usar.



O novo sistema também apresenta alguns riscos - por exemplo, ele pode deixar que usuários roubem ou alterem informações importantes como dados de salários, que são confidenciais e devem ser protegidos por uma restrição de acesso. Ou, por exemplo, apenas o departamento de exportação deve ter acesso a dados de vendas. Configurar segurança e permissões em um banco de dados é importante quando um sistema é criado.



O novo sistema pode ter outros problemas também. O banco de dados pode ser usado por muitas pessoas ao mesmo tempo. Suponha que alguém no departamento de negócios internacionais e alguém no departamento de exportação tentem alterar o nome de uma fruta ao mesmo tempo - o primeiro de *Maçã* para *M*, e o segundo de *Maçã* para *MA*. Se eles fizerem isso, o que acontecerá com o nome do produto? Para um banco de dados que será usado por muitas pessoas, esse tipo de problema deve ser considerado.



Você também deve ter cuidado para não perder nenhum dado. Além disso, o sistema pode ficar fora do ar ou um disco rígido pode falhar, fazendo com que dados sejam corrompidos. O banco de dados precisa ter mecanismos para recuperar-se desses tipos comuns de falhas.



Além disso, já que o banco de dados armazenará uma grande quantidade de dados, você deve ser capaz de fazer buscas com velocidade. O novo sistema precisa ter a capacidade de lidar com isso.

Vamos juntos com a Princesa Ruruna e o Cain começar a estudar bancos de dados para aprender como resolver esses problemas. Avance para o Capítulo 2!

RESUMO



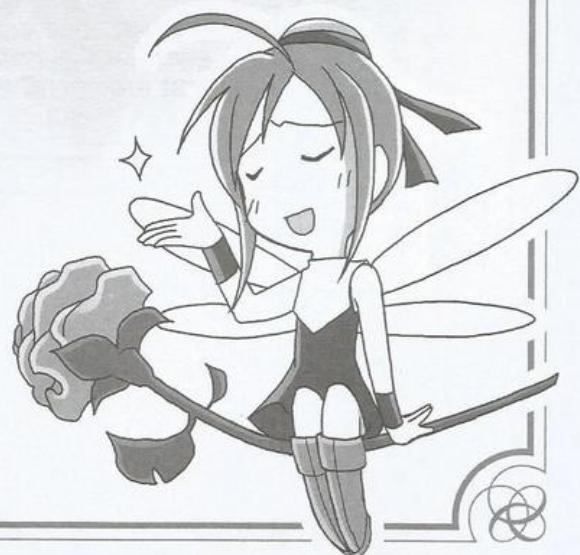
- Gerenciamento com base em arquivos pode criar duplicação e conflitos de dados.
- Um banco de dados permite compartilhar dados facilmente e impede dados conflitantes e duplicados.

USO DE SOFTWARE PARA GERENCIAR BANCOS DE DADOS

O banco de dados que vamos estudar é gerenciado por um software chamado de Sistema de Gerenciamento de Banco de Dados (SGBD). Um SGBD tem muitas funções úteis - ele permite fazer coisas como inserir dados em um banco de dados, impede dados conflitantes, e recupera rapidamente uma grande quantidade de dados. Graças a nosso SGBD, o banco de dados pode ser usado por muitas pessoas simultaneamente. Além disso, um SGBD pode proteger a segurança do banco de dados - por exemplo, ele permite que o banco de dados funcione apropriadamente mesmo se uma falha ocorrer. E ainda, oferece uma interface fácil de usar para seus usuários. Vamos estudar bancos de dados e as funções de um SGBD no próximo capítulo.

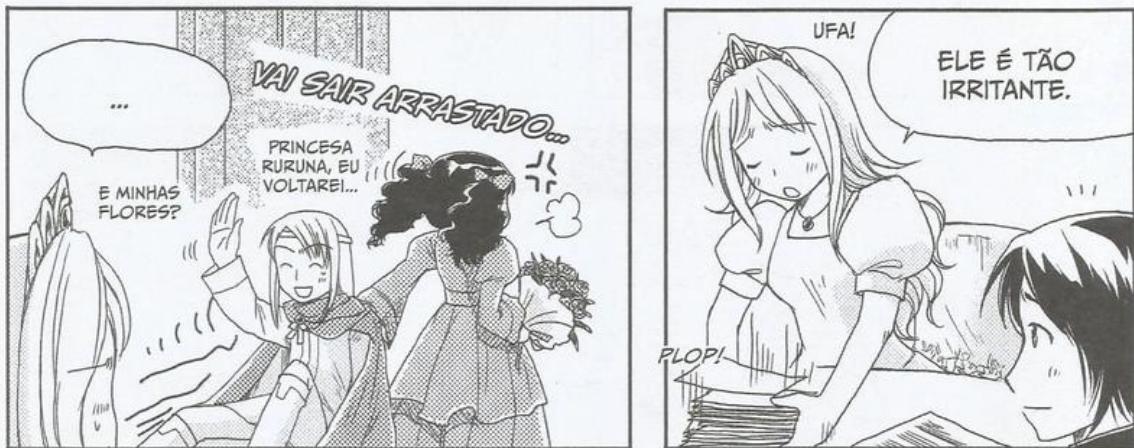
Z

O QUE É UM BANCO DE DADOS
RELACIONAL?



TERMINOLOGIA DE BANCO DE DADOS







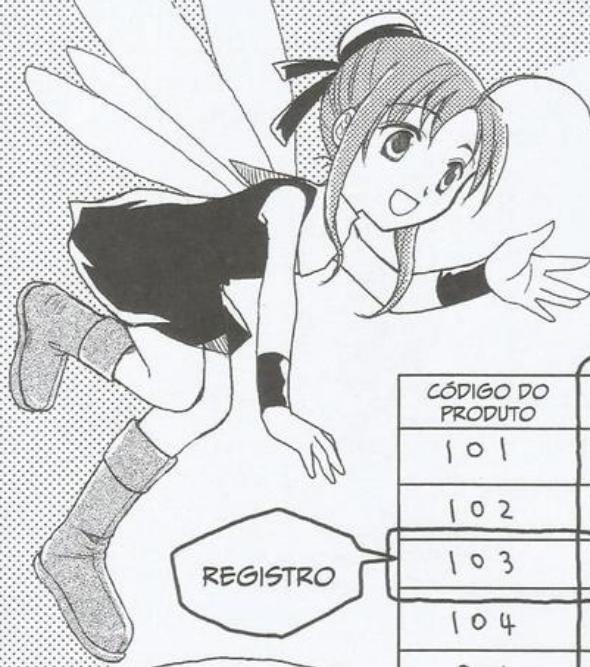


ESTE É UM DOS ARQUIVOS QUE VOCÊS ESTÃO USANDO, CERTO?

É.

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO	COMENTÁRIOS
101	MELÃO	80G	COM SEMENTES
102	MORANGO	150G	
103	MAGÃ	120G	
104	LIMÃO	200G	
201	PINHÃO	100G	
202	CAQUI	160G	





CADA REGISTRO CONTÉM
CAMPOS DO MESMO TIPO.

AH, ENTENDI.

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO	COMENTÁRIOS
101	MELÃO	800G	COM SEMENTES
102	MORANGO	150G	
103	MAÇÃ	120G	
104	LIMÃO	200G	AZEDO
201	PINHÃO	100G	COM CASCA
202	CAQUI	160G	
301	PESSEGO	130G	
302	KIWI	200G	ALTA QUALIDADE

CAMPO

POR EXEMPLO,
CÓDIGO DO
PRODUTO É UM
VALOR DE TRÊS
DÍGITOS...

E NOME DO
PRODUTO TEM DEZ
CARACTERES OU
MENOS.

CÓDIGO DO PRODUTO	NOME DO PRODUTO
101	MELÃO
102	MORA
103	MAÇÃ
104	LIMÃO
201	PINHÃO
202	CAQUI

ENTÃO, EM SEGUIDA,
VAMOS PENSAR
SOBRE O CÓDIGO
DO PRODUTO
UM POUCO MAIS,
DETALHADAMENTE.

AQUI!

HMM...
REGISTRO...
CAMPO...
TANTA
COISA PARA
DECORAR...
CAIN?



ENTÃO PODEMOS IDENTIFICAR DADOS PELOS CÓDIGOS DO PRODUTO, MAS NÃO PELO PREÇO UNITÁRIO.

EXATO.

NO MUNDO DOS BANCOS DE DADOS, UM CAMPO COMO O DE CÓDIGO DO PRODUTO, NO QUAL OS VALORES NÃO SE REPETEM...

É CHAMADO DE ÚNICO (UNIQUE).

ÚNICO?

O CÓDIGO DO PRODUTO É ÚNICO.

ALGUMAS PESSOAS COM FREQUÊNCIA CHAMAM MEU PAI ASSIM...

HUMMM... NÃO É BEM ISSO...

HA, HA, HA, REI KOD É ÚNICO

ÚNICO SIGNIFICA QUE SÓ EXISTE UM!

AH, TÁ. É NESSE SENTIDO...

EM SEGUITA, VAMOS FALAR SOBRE COMENTÁRIOS.

COMENTÁRIOS?

COMENTÁRIOS SÃO SIMPLES COMENTÁRIOS, NÃO?

VEJA DO PONTO DE VISTA DE UM BANCO DE DADOS.

ALGUNS VALORES ABAIXO DE COMENTÁRIOS ESTÃO VAZIOS, CERTO?

PESSOA	COMENTÁRIO
RURUNA	LOIRA ATIVA
CAIN	MORENO DESLIGADO

DESLIGADO, EU?

CARACTERÍSTICAS COMO ESSAS...

COMENTÁRIOS
G COM SEMENTES
OG
20 G AZEDO
200 G COM CASCA
100 G

É VERDADE...

COMENTÁRIOS
COM SEMENTES
AZEDO
COM CASCA
ALTA QUALIDADE

NÃO QUER DIZER QUE DIGITARAM ESPAÇOS...

ESTÃO REALMENTE VAZIOS.



SE É ASSIM, NÃO SE PODE IDENTIFICAR O PRODUTO OLHANDO OS COMENTÁRIOS.

NÃO.



NÃO PODE SER NULO.

CÓDIGO PRODUTO	NOME PRODUTO	PREÇO UNITÁRIO	COMENTÁRIO
101	MELÃO	800G	COM SEMENTES
102	MORANGO	150G	
103	MAÇÃ	120G	
104	LIMÃO	200G	AZEDO
201	PINHÃO	100G	COM CASCA
202	CAQUI	160G	
301	PÊSSEGO	130G	
302	KIWI	200G	

A AUSÊNCIA DE UM VALOR É CHAMADA DE NULO (NULL) NO MUNDO DOS BANCOS DE DADOS

UM NULO É ACEITÁVEL EM COMENTÁRIOS, MAS NÃO EM CÓDIGOS DE PRODUTOS, QUE IDENTIFICAM DADOS.



E ISTO É O BÁSICO SOBRE TERMOS DE BANCO DE DADOS.

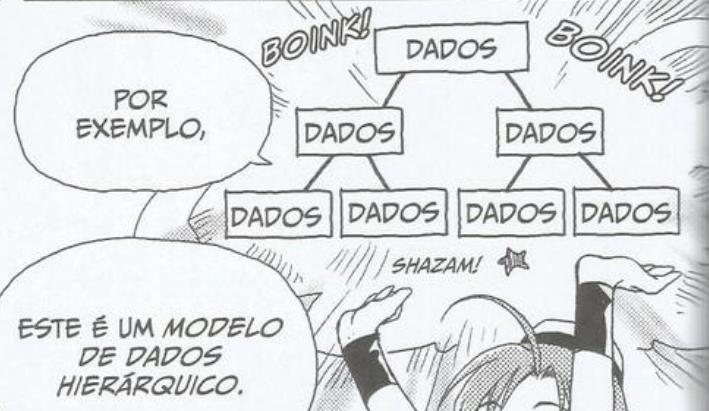
ENTENDERAM?

SIM....

MAIS OU MENOS, NÉ...

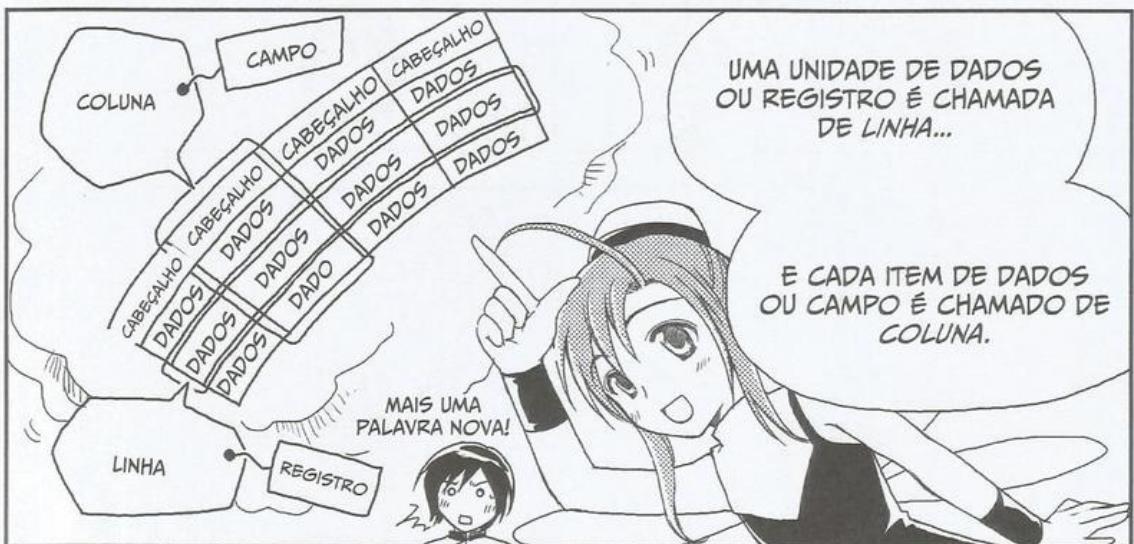
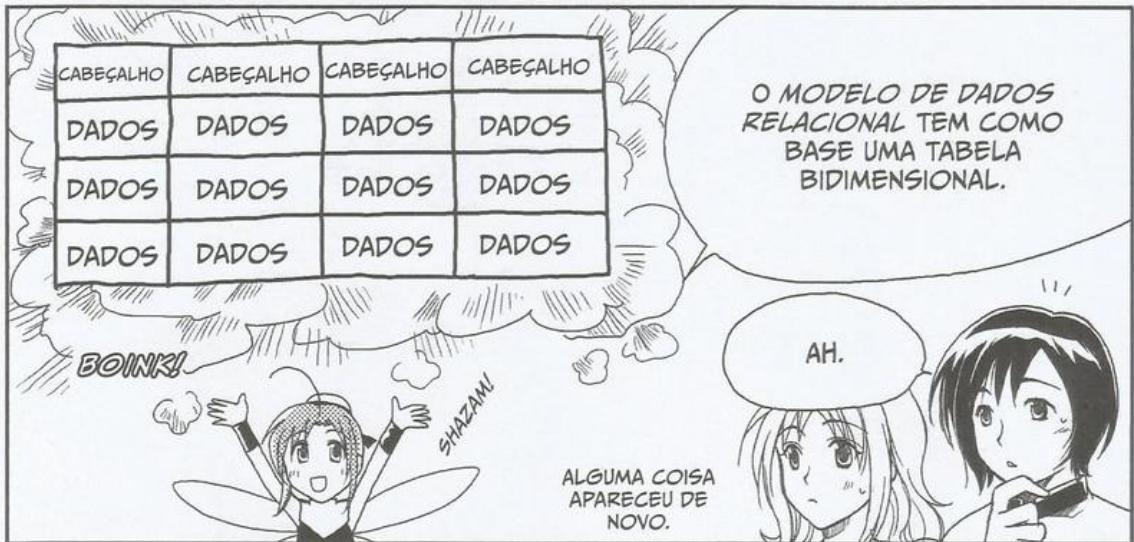
NULO?
VAZIO?
ÚNICO?

MAIS COISAS PARA DECORAR...





BANCOS DE DADOS RELACIONAIS



ALÉM DISSO,
ALGUMAS VEZES
UM CAMPO TEM UM
PAPEL IMPORTANTE
NO BANCO DE
DADOS.

ESSE CAMPO
ESPECIAL É
CHAMADO DE CHAVE
(KEY).

PAPEL
IMPORTANTE?

SIM. POR
EXEMPLO,

O CÓDIGO DO
PRODUTO NO
ARQUIVO QUE
VOCÊ VIU AGORA
HÁ POUCO.

ESSE CAMPO TEM UM
PAPEL IMPORTANTE:
IDENTIFICAR OS DADOS.

CHAVE
PRIMÁRIA

ESSE CÓDIGO
É CHAMADO DE
CHAVE PRIMÁRIA
(PRIMARY KEY).

NÃO SABIA QUE
EXISTIAM TANTOS
TERMOS.

BOM, EU SEI ALGUMA
COISA SOBRE
TABELAS.

É FÁCIL DE ENTENDER
SE VOCÊ JÁ LIDA COM
DADOS USANDO UMA
TABELA.

ESSE É UM DOS
MÉRITOS DO
MODELO DE DADOS
RELACIONAL.

MESMO PESSOAS
QUE NÃO SABEM
MUITO SOBRE
BANCOS DE DADOS
PODEM TRABALHAR
COM DADOS.



POR EXEMPLO,
VAMOS OLHAR DE
NOVO PARA A TABELA
DE PRODUTOS.

CÓDIGO PRODUTO	NOME PRODUTO	PREÇO UNITÁRIO	COMENTÁRIO
101	MELÃO	800G	COM SEMENTES
102	MORANGO	150G	
103	MAÇÃ	120G	
104	LIMÃO	200G	AZEDO
201	PINHÃO	100G	COM CASCA
202	CAQUI	160G	
301	PESSEGOGO	130G	
302	KIWI	200G	ALTA QUALIDADE

NOME PRODUTO
MELÃO
MORANGO
MAÇÃ
LIMÃO
PINHÃO
CAQUI
PESSEGOGO
KIWI

MÁGICA!

VOCÊ PODE
EXTRAIR O
NOME DOS
PRODUTOS?

UMA OPERAÇÃO
PARA EXTRAIR
UMA COLUNA
DESSA FORMA
É CHAMADA
PROJEÇÃO.

ENTÃO EXTRAIR O NOME DOS PRODUTOS É UMA OPERAÇÃO?

SIM, É SIMPLES.

EXISTEM MUITAS OUTRAS OPERAÇÕES. DE FATO, SÃO OITO!

UNIÃO

DIFERENÇA

PRODUTO CARTESIANO

DIVISÃO

PROJEÇÃO

SELEÇÃO

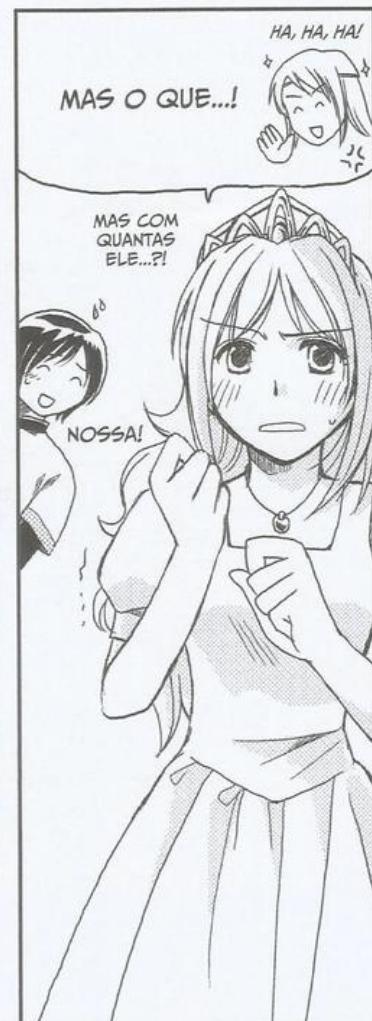
JUNÇÃO

NOSSA!

OUTRA VANTAGEM DO MODELO DE DADOS RELACIONAL É QUE SE PODE PROCESSAR OS DADOS...

COMBINANDO ESSAS OPERAÇÕES.

ENTENDI.

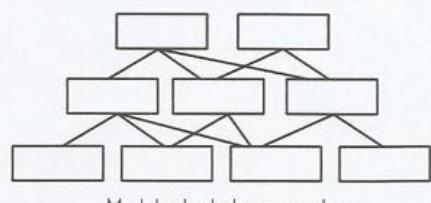
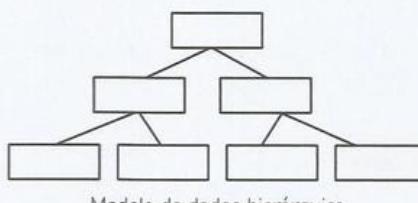


TIPOS DE MODELOS DE DADOS



Quando você usa o termo *banco de dados*, que tipo de banco de dados tem em mente? Existem muitos tipos possíveis para gerenciamento de dados. Os métodos de associação e operação de dados que um banco de dados usa é chamado de seu *modelo de dados*. Existem três modelos mais comuns utilizados.

Como descrevi para Ruruna e Cain, o primeiro tipo é o *modelo de dados hierárquico*. Nele, dados filhos têm apenas um único dado pai. O segundo tipo é o *modelo de dados em rede*. Diferente do modelo hierárquico, nesse modelo dados filhos podem ter múltiplos ancestrais.



Para usar qualquer um desses modelos, é preciso gerenciar os dados mantendo sua localização física e ordem em mente. Dessa forma, é difícil efetuar uma pesquisa flexível e de alta velocidade em seus dados se os modelos hierárquico ou de rede forem utilizados.

O terceiro tipo de modelo é o relacional. Um banco de dados relacional utiliza o conceito fácil de entender de tabelas. Vamos discutir esse modelo com maiores detalhes.

Modelo de dados relacional

OPERAÇÕES DE EXTRAÇÃO DE DADOS

Como os dados são extraídos em um banco de dados relacional? Pode-se processar e extrair os dados nesse modelo efetuando operações matemáticas estritamente definidas. Existem oito tipos de operações principais que podemos usar, e elas se dividem em duas categorias – de conjunto e relacionais.

OPERAÇÕES DE CONJUNTO

As operações de união, diferença, intersecção e produto cartesiano são chamadas de *operações de conjunto*. Elas atuam sobre um ou mais conjuntos de linhas para produzir um novo conjunto de linhas. Em resumo, elas determinam quantas linhas da entrada aparecem na saída. Vamos examinar alguns exemplos utilizando a Tabela de Produtos 1 e a Tabela de Produtos 2.

TABELA DE PRODUTO 1

Nome do Produto	Preço Unitário
Melão	800G
Morango	150G
Maçã	120G
Limão	200G

TABELA DE PRODUTO 2

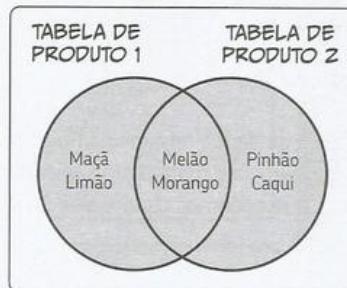
Nome do Produto	Preço Unitário
Melão	800G
Morango	150G
Pinhão	100G
Caqui	350G

UNIÃO

Efetuar a operação de *união* permite que se extraiam todos os produtos inclusos na Tabela de Produto 1 e na de Produto 2. O resultado encontra-se abaixo.

Nome do Produto	Preço Unitário
Melão	800G
Morango	150G
Maçã	120G
Limão	200G
Pinhão	100G
Caqui	350G

Executar uma operação de união extrai todas as linhas nas duas tabelas e as combina. A figura a seguir mostra como ficam os dados das duas tabelas depois que a operação foi executada. Todas as linhas na Tabela de Produto 1 e na 2 foram extraídas.

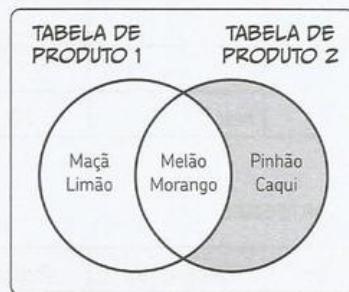
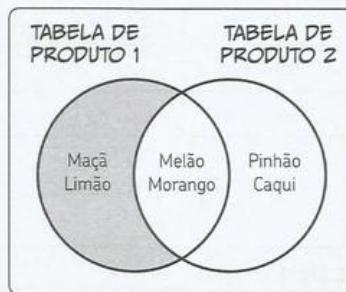


DIFERENÇA

Diferença é uma operação que extrai linhas de apenas uma das tabelas. Por exemplo, uma operação de diferença pode extrair todos os produtos da primeira tabela que não estão incluídos na segunda. Os resultados dependem de qual tabela contém as linhas para extrair, de qual tabela inclui as linhas para subtrair.

Nome do Produto	Preço Unitário
Maçã	120G
Limão	200G

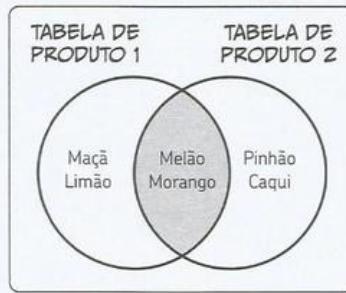
Nome do Produto	Preço Unitário
Pinhão	100G
Caqui	350G



INTERSECÇÃO

Você também pode extrair produtos que estão inclusos em *ambas* as Tabelas de Produto 1 e 2. Essa operação é chamada de operação de *intersecção*. Aqui vemos o resultado da intersecção das Tabelas de Produto 1 e 2.

Nome do Produto	Preço Unitário
Melão	800G
Morango	150G



PRODUTO CARTESIANO

A operação *produto cartesiano* é um método que combina todas as linhas nas duas tabelas. Vamos examinar as Tabelas de Produto e de Destino de Exportação abaixo.

A operação *produto Cartesiano* combina todas as linhas nas duas tabelas. Neste exemplo, ela resulta em $3 \times 3 = 9$ linhas. Repare que os nomes de coluna (ou campos) nessas duas tabelas não são os mesmos - diferente de nossos exemplos anteriores.

TABELA DE PRODUTOS

Código do Produto	Nome do Produto	Preço Unitário
101	Melão	800G
102	Morango	150G
103	Maçã	120G

TABELA DE DESTINOS DE EXPORTAÇÃO

Cód. Dest. Export.	Nome Dest. Export.
12	Reino de Minami
23	Império Alfa
25	Reino de Ritol

3
linhas

PRODUTO CARTESIANO

Código do Produto	Nome do Produto	Preço Unitário	Cód. Dest. Export.	Nome Dest. Export.
101	Melão	800G	12	Reino de Minami
101	Melão	800G	23	Império Alfa
101	Melão	800G	25	Reino de Ritol
102	Morango	150G	12	Reino de Minami
102	Morango	150G	23	Império Alfa
102	Morango	150G	25	Reino de Ritol
103	Maçã	120G	12	Reino de Minami
103	Maçã	120G	23	Império Alfa
103	Maçã	120G	25	Reino de Ritol

3 x 3 =
9 linhas



OPERAÇÕES RELACIONAIS

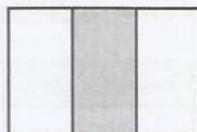
Um banco relacional é projetado para que os dados possam ser extraídos por operações de conjunto e operações relacionais. Vamos dar uma olhada em outras quatro operações específicas de um banco de dados relacional, chamadas *operações relacionais* – projeção, seleção, junção e divisão.

PROJEÇÃO

Projeção é uma operação que extrai colunas de uma tabela. No exemplo mostrado aqui, essa operação foi usada para extrair apenas os nomes de produtos incluídos na Tabela de Produtos.

Nome do Produto
Melão
Morango
Maçã
Limão

Pense na projeção como extração “vertical”, como mostrado abaixo.

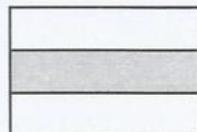


SELEÇÃO

A operação seleção extrai linhas de uma tabela.

Nome do Produto	Preço Unitário
Melão	800G
Morango	150G

Seleção é como projeção, mas extrai linhas em vez de colunas. A seleção extrai dados “horizontalmente”.



JUNÇÃO

A operação *junção* (*join*) é muito poderosa. Ela literalmente refere-se à tarefa de juntar tabelas. Vamos examinar as tabelas abaixo como exemplo.

TABELA DE PRODUTOS

Código do Produto	Nome do Produto	Preço Unitário
101	Melão	800G
102	Morango	150G
103	Maçã	120G
104	Limão	200G

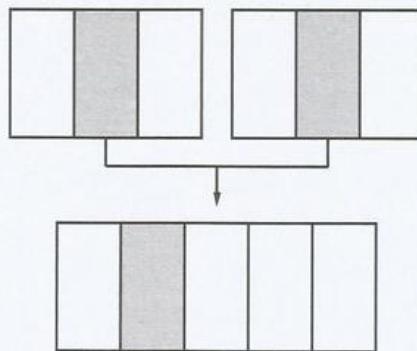
TABELA DE VENDAS

Data	Código do Produto	Quantidade
1/11	102	1100
1/11	101	300
5/11	103	1700
8/11	101	500

As colunas Código do Produto nessas duas tabelas representam a mesma informação. Em 1º de novembro, 1.100 morangos (código do produto 102) foram vendidos. A Tabela de Vendas não inclui o nome do produto, mas inclui seu código. Em outras palavras, a Tabela de Vendas permite entender qual produto foi vendido fazendo referência ao código, que é a *chave primária* (*primary key*) na Tabela de Produtos. O código do produto na Tabela de Vendas é uma *chave estrangeira* (*foreign key*). Juntar as duas tabelas para que a chave estrangeira corresponda à chave primária resulta na seguinte tabela.

Data	Código do Produto	Nome do Produto	Preço Unitário	Quantidade
1/11	102	Morango	800G	1100
1/11	101	Melão	800G	300
5/11	103	Maçã	120G	1700
8/11	101	Melão	800G	500

Isso cria uma nova tabela dinâmica de dados de venda, incluindo a data, o código do produto, o nome do produto, seu preço unitário e a quantidade. A figura abaixo mostra uma junção – a área sombreada representa uma coluna que aparece nas duas tabelas originais.



DIVISÃO

Finalmente, vamos examinar a divisão. *Divisão* é uma operação que extrai as linhas cujos valores de coluna correspondem àqueles na segunda tabela, mas apenas retorna colunas que não existem nessa última. Vamos ver um exemplo.

TABELA DE VENDAS

Cód. Dest. Exp.	Nome Dest. Exp.	Data
12	Reino de Minami	5/3
12	Reino de Minami	10/3
23	Império Alfa	5/3
25	Reino de Ritol	21/3
30	Reino de Sazanna	25/3

TABELA DE DESTINO DE EXPORTAÇÃO

Cód. Dest. Exp.	Nome Dest. Exp.
12	Reino de Minami
23	Império Alfa

Dividir a Tabela de Vendas pela de Destino da Exportação resulta na tabela a seguir. Isso permite encontrar as datas quando as frutas foram exportadas tanto para o Império Alfa quanto para o Reino de Minami.

Data
5/3

QUESTÕES



Agora, vamos responder a algumas questões para ver se você entendeu bem os bancos de dados relacionais. As respostas estão na página 48.

Q1

Como se chama a chave que se refere a uma coluna em uma tabela diferente no banco de dados relacional?

Q2

A tabela a seguir exibe informações sobre livros. Qual item você pode usar como uma chave primária? ISBN é Número Padrão Internacional de Livro (International Standard Book Number), um número de identificação único atribuído a qualquer livro publicado. Alguns livros podem ter o mesmo título.

ISBN	Nome do livro	Nome do autor	Data de publicação	Preço
------	---------------	---------------	--------------------	-------

Q3

Como se chama a operação utilizada aqui para extrair dados?

Cód. Dest. Exp.	Nome Dest. Exp.
12	Reino de Minami
23	Império Alfa
25	Reino de Ritol
30	Reino de Sazanna



Cód. Dest. Exp.	Nome Dest. Exp.
25	Reino de Ritol

Q4

Como se chama a operação utilizada aqui para extrair dados?

Cód. Dest. Exp.	Nome Dest. Exp.
12	Reino de Minami
23	Império Alfa
25	Reino de Ritol
30	Reino de Sazanna



Cód. Dest. Exp.	Nome Dest. Exp.
15	Reino de Paronu
22	Reino de Tokanta
31	Reino de Taharu
33	Reino de Mariyon

Cód. Dest. Exp.	Nome Dest. Exp.
12	Reino de Minami
15	Reino de Paronu
22	Reino de Tokanta
23	Império Alfa
25	Reino de Ritol
30	Reino de Sazanna
31	Reino de Taharu
33	Reino de Mariyon

Q5

Como se chamam as operações usadas aqui para extrair dados?

Cód. Dest. Exp.	Nome Dest. Exp.
12	Reino de Minami
23	Império Alfa
25	Reino de Ritol
30	Reino de Sazanna

Cód. Dest. Exp.	Data
12	1/3
23	1/3
12	3/3
30	5/3
12	6/3
25	10/3



Cód. Dest. Exp.	Data	Nome Dest. Exp.
12	1/3	Reino de Minami
23	1/3	Império Alfa
12	3/3	Reino de Minami
30	5/3	Reino de Sazanna
12	6/3	Reino de Minami
25	10/3	Reino de Ritol

O BANCO DE DADOS RELACIONAL VENCE!

Em um banco de dados relacional, pode-se utilizar oito diferentes operações para extrair dados. Os resultados obtidos também são tabelados. Se você combinar as operações explicadas nesta seção, poderá extraír dados para qualquer propósito. Por exemplo, pode-se usar o nome e o preço de um produto para criar dados totalizados de vendas brutas. Bancos de dados relacionais são populares porque são fáceis de entender e permitem processamento de dados flexível.

RESUMO



- Uma linha de dados é chamada de *registro* (*record*), e cada coluna, de *campo* (*field*).
- Uma coluna que pode ser usada para identificar dados é chamada de *chave primária*.
- Em um banco de dados relacional, pode-se processar dados utilizando o conceito de tabela.
- Em um banco de dados relacional, pode-se processar dados com base em operações matemáticas.

RESPOSTAS

Q1 Chave estrangeira

Q2 ISBN

Q3 Seleção

Q4 União

Q5 Junção (join)

3

VAMOS PROJETAR UM BANCO DE
DADOS!



O MODELO E-R

CAIN? ONDE VOCÊ ESTÁ?



...SOBRE BANCO DE DADOS OU ALGO ASSIM...

BOM DIA, MENINAS!

AI, CAIN...

PRINCESA RURUNA!

B... B...
BOM DIA!

AH, ENTENDI.

ENTÃO É ISSO...

CAIN!!





EM PRIMEIRO LUGAR,
PENSAMOS QUE SERIA
MAIS FÁCIL CRIAR UM
BANCO DE DADOS...

SE CONSEGUÍSSERAMOS
ENTENDER AS
CONDIÇÕES REAIS DO
REINO DE KOD...

A TICA DIZ QUE O
PROCEDIMENTO
CORRETO É CRIAR UM
MÓDELO E ANALISAR
O GERENCIAMENTO DE
EXPORTAÇÃO ATUAL
BASEANDO-SE NELE.

PO
FAVOR
SENTE-SE.
OBIGADA.

É UMA
LIÇÃO MEIO
LONGA...

ENTÃO TEMOS QUE
ESTUDAR ALGO NOVO.

BOM, ESTÃO
PRONTOS?

USAREMOS UM
MÓDELO DE
ANALISE CHAMADO...

MÓDELO E-R!

E DE ENTIDADE E R DE
RELACIONAMENTO.

E =
ENTIDADE

R =
RELACIONAMENTO

E-R...?

ENTIDADE E
RELACIONAMENTO...

NEM EU.

NÃO CONHEÇO
ESSES NOMES...



TÁ, VOCÊ ANALISOU FRUTAS E DESTINOS DE EXPORTAÇÃO.

ENTÃO, O QUE É UM RELACIONAMENTO?

TEM A VER COM COMO AS ENTIDADES SE RELACIONAM.

POR EXEMPLO...

FRUTAS E DESTINOS DE EXPORTAÇÃO ESTÃO RELACIONADOS PORQUE VENDEM-SE FRUTAS PARA ESSES DESTINOS.

FRUTA

DESTINO DE EXPORTAÇÃO

ENTÃO PODEMOS CONSIDERAR VENDAS COMO UM RELACIONAMENTO.

FRUTA

VENDAS

R!!!!SC!!

FRUTAS SÃO EXPORTADAS PARA MUITOS PAÍSES, E CADA DESTINO DE EXPORTAÇÃO COMpra MUITOS TIPOS DE FRUTAS.

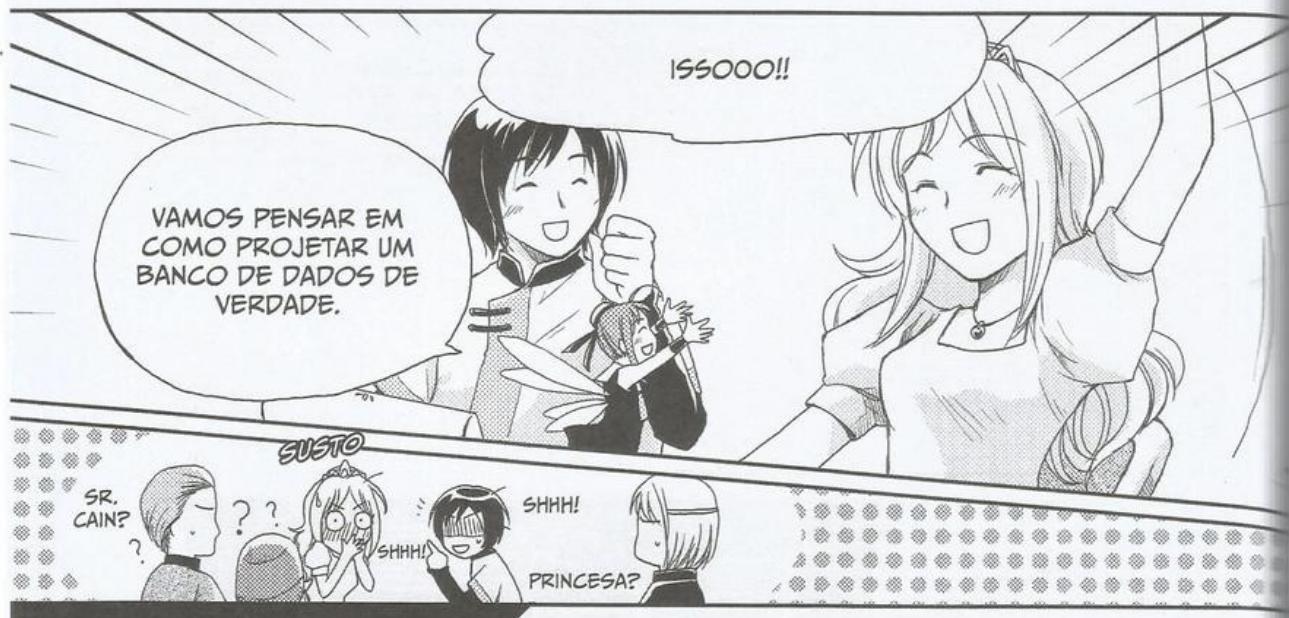


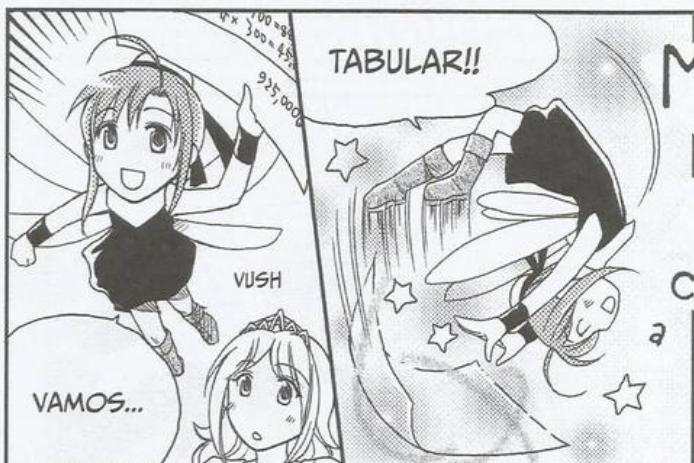


NORMALIZAÇÃO DE UMA TABELA



ISSO MESMO! A PRIMEIRA COISA A FAZER É ANALISAR A SITUAÇÃO REAL. ISSO É MUITO IMPORTANTE.

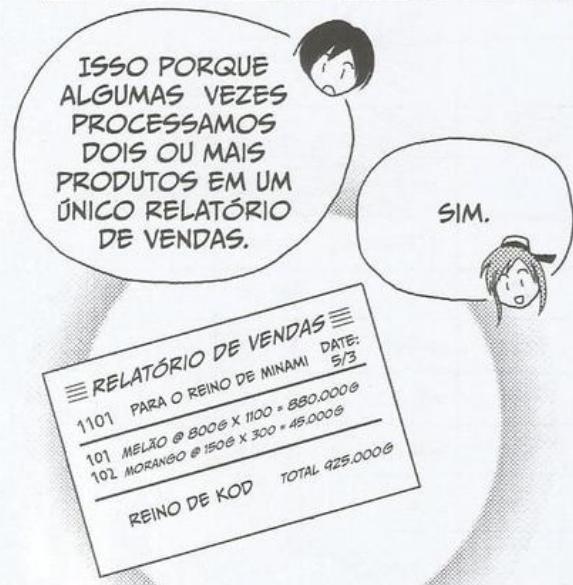
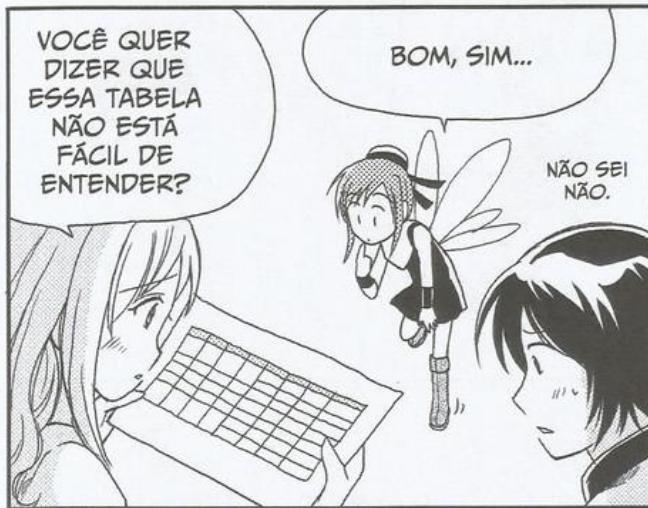




CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.	CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO	QUANTIDADE
1101	5/3	12	REINO DE MINAMI	101	MELÃO	800G	1100
				102	MORANGO	150G	300
1102	7/3	23	IMPÉRIO ALFA	103	MAÇÃ	120G	1700
1103	8/3	25	REINO DE RITOL	104	LIMÃO	200G	500
1104	10/3	12	REINO DE MINAMI	101	MELÃO	800G	2500
1105	12/3	25	REINO DE RITOL	103	MAÇÃ	120G	2000
				104	LIMÃO	200G	700

TABELA CRIADA A PARTIR DOS RELATÓRIOS DE VENDAS

Mági
ca



PARA ISSO

EM VEZ DE USAR UMA
ÚNICA TABELA PARA
TODOS OS DADOS...

BLAM!!

NÓS OS DIVIDIREMOS EM
VÁRIAS TABELAS!

SHAZAM!!

ASSIM!!

PODE-SE FAZER
ISSO?!

MAS ISSO PARECE
MUITO MAIS DIFÍCIL
DO QUE USAR UMA
ÚNICA TABELA...

PODE PARECER
COMPLICADO,

MAS É IMPORTANTE PARA
GERENCIAR OS DADOS
CORRETAMENTE E SEM
DISCREPÂNCIAS.

E ESSE É
O NOSSO
TRABALHO.

DESCOBRIR AS
CONDIÇÕES
REais...

USANDO UM
MÓDULO
E-R...

TÁ BOM...

O QUÊ?! EU
ESTAVA SÓ
PENSANDO...

TEM
ALGUMA
COISA
ERRADA,
CAIN?

VOÇÊ
RESMUNGA EM
VOZ ALTA.

SUPONHA QUE
QUEIRAMOS
AUMENTAR 20G NO
PREÇO UNITÁRIO DO
MELÃO.



20G
SOBE



SE VOCÊ USAR
ESSA TABELA DO
JEITO QUE ESTÁ,

TERÁ QUE
ENCONTRAR TODAS
AS LINHAS COM
MELÃO E CORRIGIR
O PREÇO UNITÁRIO.

820G

800G

150G

AQUI
TAMBÉM...

820G

10G

MAS, SE VOCÊ TEM
UM TABELA RELATIVA
SOMENTE AOS
PRODUTOS,

TABELA DE
PRODUTOS

MELÃO	800G
MORANGO	150G
MAÇÃ	120G
LIMÃO	200G

PODE CORRIGIR O
PREÇO EM APENAS UM
LUGAR - UMA LINHA NA
TABELA DE PRODUTOS.

SÓ AQUI!!!

FÁCIL!

ASSIM, NÃO HAVERÁ
CONFLITOS POR VOCÊ
ESQUECER DE CORRIGIR
ALGUMA LINHA!! ISSO NÃO É
ÓTIMO?

É, TODO MUNDO
PODE ESQUECER
UMA OU DUAS
COISAS...

ENTENDI. VISTO
POR ESSE
ÂNGULO, É BEM
CONVENIENTE.

RECEBE O NOME DE
NORMALIZAÇÃO.

ENTÃO, COMO
DEVEMOS
PROCEDER?

DIVIDIR A TABELA
PARA EVITAR QUE
CONFLITOS DE DADOS
ACONTEÇAM...

HUM...

NORMALIZAÇÃO,
NORMALIZAÇÃO

X/II
SIM, ISSO É
IMPORTANTE!

RESMUNGANDO
DE NOVO?

EM
PRIMEIRO
LUGAR...

VAMOS TENTAR
ALTERÁ-LA PARA
QUE CADA LINHA
TENHA UM ÚNICO
VALOR.

DADOS REPETIDOS SÃO
UMA PISTA DE QUE LINHAS
DEVEM SER DIVIDIDAS.

ENTÃO, VOU DIVIDIR EM...

UMA TABELA COM DATAS,
CÓDIGO DE DESTINO DA
EXPORTAÇÃO E NOME DO
DESTINO DA EXPORTAÇÃO...

E OUTRA COM CÓDIGO
DO PRODUTO, NOME DO
PRODUTO, PREÇO UNITÁRIO
E QUANTIDADE.

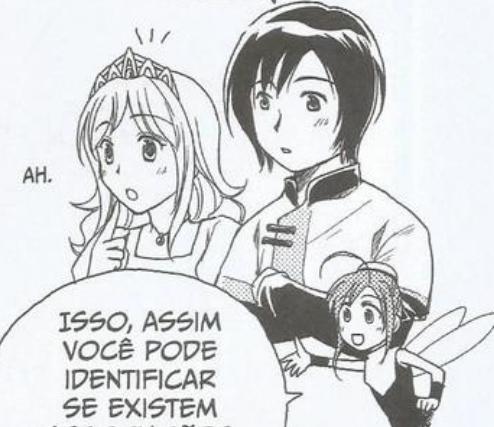
TABELA DE VENDAS
(PRIMEIRA FORMA NORMAL (1))

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
1101	5/3	12	REINO DE MINAMI
1102	7/3	23	IMPÉRIO ALFA
1103	8/3	25	REINO DE RITOL
1104	10/3	12	REINO DE MINAMI
1105	12/3	25	REINO DE RITOL

MAS O CÓDIGO DE
RELATÓRIO EXISTE
EM AMBAS AS
TABELAS, NÃO?

TABELA DE DETALHES DE VENDAS
(PRIMEIRA FORMA NORMAL (2))

CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO	QUANTIDADE
1101	101	MELÃO	800G	1100
1101	102	MORANGO	150G	300
1102	103	MAÇÃ	120G	1700
1103	104	LIMÃO	200G	500
1104	101	MELÃO	800G	2500
1105	103	MAÇÃ	120G	2000
1105	104	LIMÃO	200G	700

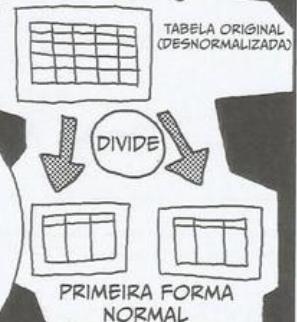


UMA TABELA QUE RESULTA DE UMA DIVISÃO COMO ESSA ESTÁ NA PRIMEIRA FORMA NORMAL.



A TABELA QUE TINHA LINHAS COM DOIS OU MAIS VALORES ANTES DE SER DIVIDIDA ESTÁ NA FORMA DESNORMALIZADA.

ISSO SIGNIFICA QUE A PRIMEIRA FORMA NORMAL É CRIADA DIVIDIENDO-SE A FORMA DESNORMALIZADA.



VAMOS VER...
ESPERE UM POUCO.

ESTAS ESTÃO NA "PRIMEIRA FORMA NORMAL". ISSO QUER DIZER QUE EXISTEM UMA "SEGUNDA" E UMA "TERCEIRA" FORMAS?

BINGO!

A PRIMEIRA FORMA NORMAL NÃO PODE SER USADA COMO UMA TABELA DE BANCO RELACIONAL ASSIM COMO ESTÁ.

AGUENTEM FIRME!

AH, ENTENDO...

VAMOS LÁ!!

É UM LONGO CAMINHO!!

MONTE BANCO DE DADOS RELACIONAL

TUDO BEM ENTÃO,
VAMOS OLHAR A
PRIMEIRA FORMA
NORMAL (2).

AQUI ESTÁ.

É A TABELA DE
DETALHES DE
VENDAS.

CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO	QUANTIDADE
1101	101	MELÃO	800G	1100
1101	102	MORANGO	150G	300

TABELA DE DETALHES DE VENDAS
(PRIMEIRA FORMA NORMAL (1))

VOCÊ AINDA NÃO PODE
GERENCIAR PRODUTOS
COM ESSA TABELA.

EE!!! POR
QUÊ?

PORQUE, SE VOCÊS
RECEBESSEM LARANJAS
MANDARIM,

NÃO PODERIAM
ADICIONÁ-LAS
A ESSA TABELA
POR ELAS AINDA
NÃO TEREM
SIDO VENDIDAS.



COMO ASSIM?

ENTENDI!

JÁ QUE NENHUMA VENDA FOI
INFORMADA, O CÓDIGO DE
RELATÓRIO E A QUANTIDADE
SÃO DESCONHECIDOS.

MAÇÃS COM VENDAS



E NA TABELA (2), OS
DADOS RELACIONADOS
A PRODUTOS E VENDAS
ESTÃO MISTURADOS.

LARANJAS SEM VENDAS

FALTA DE
DADOS

CÓDIGO
DO
PRODUTO

NOME DO
PRODUTO

PREÇO
UNITÁRIO

MUITO BEM!

VOCÊ NÃO PODE GERENCIAR PRODUTOS INDEPENDENTEMENTE USANDO A TABELA (2).

HUMM.

TABELA (2)

ISSO MESMO!! ENTÃO,

DIVIDA A TABELA (2)

EM DUAS!!

ESSAS SÃO AS TABELAS QUE RESULTAM DA DIVISÃO DA PRIMEIRA FORMA NORMAL (1) EM DUAS.

TABELA DE PRODUTOS
(SEGUNDA FORMA NORMAL (1))

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
102	MORANGO	150G
103	MAÇÃ	120G
104	LIMÃO	200G

TABELA DE DETALHES DE VENDAS
(SEGUNDA FORMA NORMAL (2))

CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO	QUANTIDADE
1101	101	1100
1101	102	300
1102	103	1700
1103	104	500
1104	101	2500
1105	103	2000
1105	104	700

A TABELA (1) CONTÉM DADOS RELACIONADOS AOS PRODUTOS.

SE O VALOR NA COLUNA DE CÓDIGO DO PRODUTO FOR DETERMINADO, PODEMOS ENCONTRAR DADOS NAS COLUNAS DO NOME DO PRODUTO E DO PREÇO UNITÁRIO.

ENTÃO ISSO SIGNIFICA QUE O CÓDIGO DO PRODUTO, COMO CHAVE PRIMÁRIA, DETERMINA OS VALORES EM OUTRAS COLUNAS.

AH, NOSSA!!

EXATO.

SABEM OS DADOS RELACIONADOS AOS ITENS DE DETALHES DE VENDAS NA TABELA (2)?

O VALOR DE SUAS COLUNAS TAMBÉM É DETERMINADO PELA CHAVE PRIMÁRIA

CONTUDO...

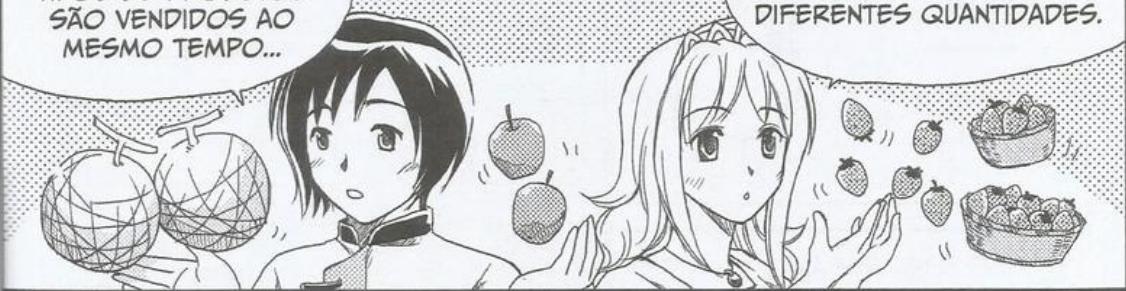
PARA A TABELA (2), CONSIDEREM A COMBINAÇÃO DO CÓDIGO DO RELATÓRIO E DO PRODUTO COMO UMA CHAVE PRIMÁRIA.

CHAVE PRIMÁRIA	
CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO



EM ALGUNS CASOS, DOIS TIPOS DE PRODUTOS SÃO VENDIDOS AO MESMO TEMPO...

EM OUTROS CASOS, UM TIPO DE PRODUTO É VENDIDO EM DIFERENTES QUANTIDADES.



ISSO QUER DIZER...

DEVE-SE DIVIDIR A TABELA PARA QUE QUANDO UMA CHAVE PRIMÁRIA FOR DETERMINADA, VALORES EM OUTRAS COLUNAS SEJAM DETERMINADOS.

ENTENDEU?



ENTENDI.



A TABELA QUE
RESULTA DA DIVISÃO
DE ACORDO COM
ESSA REGRAS ESTÁ NA

SEGUNDA FORMA
NORMAL.

PODEMOS ACRESCENTAR
AS LARANJAS MANDARIM
DE QUE FALAMOS ANTES
NA SEGUNDA FORMA
NORMAL (1).

TAMBÉM PODEMOS
ACRESCENTAR
KIWIS E UVAS

QUE AINDA NÃO
FORAM VENDIDOS!

MESMO SE O PREÇO
DO MELÃO MUDAR,
BASTA CORRIGIRMOS
OS DADOS EM UMA
ÚNICA LINHA, CERTO?

820G

...ALIÁS, VOCÊ DIVIDIU
A PRIMEIRA FORMA
NORMAL (2),

HÄ?

ENTÃO NÃO É NECESSÁRIO
DIVIDIR A TABELA DE VENDAS
NA PRIMEIRA FORMA
NORMAL (1)?

OH, VOCÊ
ESTÁ
USANDO
ÓCULOS
AGORA.

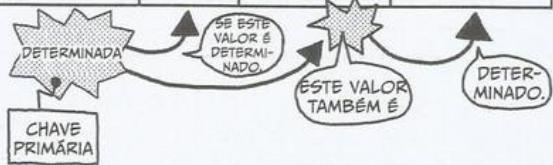
VUPT

VAMOS VER!!

BEM PENSADO!

TABELA DE VENDAS (PRIMEIRA
FORMA NORMAL (1))

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
1101	5/3	12	REINO DE MINAMI
1102	7/3	23	IMPÉRIO ALFA
1103	8/3	25	REINO DE RITOL
1104	10/3	12	REINO DE MINAMI
1105	12/3	25	REINO DE RITOL



PARA ESSA TABELA, SE O VALOR DO
CÓDIGO DE RELATÓRIO FOR DETERMINADO,
TODOS OS OUTROS VALORES, DATA,
CÓDIGO DE DESTINO DE EXPORTAÇÃO E
NOME DO DESTINO DE EXPORTAÇÃO FICAM
DETERMINADOS.

É VERDADE!!

COMO SE CLASSIFICA UMA TABELA EM QUE OS VALORES EM OUTRAS COLUNAS SÃO DETERMINADOS QUANDO A CHAVE PRIMÁRIA TAMBÉM É?

NA SEGUNDA FORMA NORMAL!!

ENTÃO ESTA TABELA ESTÁ NA SEGUNDA FORMA NORMAL, CERTO?

TABELA DE VENDAS
(PRIMEIRA FORMA NORMAL (1))

CÓDIGO DO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
			REINO DE MINAMI
			IMPÉRIO ALFA
			REINO DE RITOL

CERTO. VOCÊ PODE CONSIDERAR A PRIMEIRA FORMA NORMAL (1)...

TABELA DE VENDAS
(SEGUNDA FORMA NORMAL (3))

CÓDIGO DO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
			REINO DE MINAMI
			IMPÉRIO ALFA
			REINO DE RITOL

COMO A SEGUNDA FORMA NORMAL (3)!

COMPLETAMOS A PRIMEIRA E A SEGUNDA FORMAS NORMAIS.

AGORA, NOSSO BANCO DE DADOS ESTÁ COMPLETO?



! C H O Q U E !



OLHE A SEGUNDA FORMA
NORMAL (3) DE NOVO.



NÃO SE CONSEGUE
GERENCIAR DESTINOS
DE EXPORTAÇÃO COM
ESSA TABELA.

PENSANDO,
PENSANDO,
PENSANDO...
AH!

TABELA DE VENDAS
(SEGUNDA FORMA NORMAL (3))

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
1101	5/3	12	REINO DE MINAMI
	7/3	23	IMPÉRIO ALFA
	8/3	25	REINO DE RITOL

O REINO DE SAZANNA, QUE
NÃO IMPORTOU FRUTA ALGUMA,
NÃO PODE SER GERENCIADO
ACRESCENTANDO-O NESSA
TABELA.



NA TABELA (3), OS
DADOS QUE RELACIONAM
DESTINOS DE
EXPORTAÇÃO E VENDAS
ESTÃO MISTURADOS.

HUMM...

COMO PODEMOS
GERENCIAR
DESTINOS DE
EXPORTAÇÃO
DE FORMA
INDEPENDENTE?

DE NOVO, DIVIDA-A!

TABELA DE VENDAS
(TERCEIRA FORMA NORMAL (1))

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.
1101	5/3	12
1102	7/3	23
1103	8/3	25
1104	10/3	12
1105	12/3	25

ISSO
MESMO...

TABELA DE DEST. DE EXPORTAÇÃO
(TERCEIRA FORMA NORMAL (2))

CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
12	REINO DE MINAMI
23	IMPÉRIO ALFA
25	REINO DE RITOL

SHAZAM!

NA SEGUNDA FORMA NORMAL (3), O NOME DOS DESTINOS DE EXPORTAÇÃO É DETERMINADO DE ACORDO COM O CÓDIGO DE RELATÓRIO.

SIM.

MAS, DE FATO, DETERMINADO CÓDIGO DE RELATÓRIO CORRESPONDE A UM VALOR DE CÓDIGO DE DESTINO DE EXPORTAÇÃO,

CÓDIGO DO RELATÓRIO



CÓDIGO DEST. DE EXP.



NOME DEST. DE EXP.

DESSA FORMA DETERMINANDO O NOME DO DESTINO DE EXPORTAÇÃO INDIRETAMENTE.

PARA LIDAR COM TAI QUESTÕES,

VOCÊ DIVIDE A TABELA PARA QUE NENHUM ITEM SEJA DETERMINADO INDIRETAMENTE.

DETERMINAÇÃO

DETERMINAÇÃO

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.

CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.

CORRETO. UMA TABELA QUE NÃO PERMITE QUE OUTRA CHAVE, EXCETO A PRIMÁRIA, DETERME VALORES EM OUTRAS COLUNAS...

ESTÁ NA TERCEIRA FORMA NORMAL!!

FINALMENTE CHEGAMOS NA TERCEIRA FORMA NORMAL.

AGORA, VOCÊS PODEM GERENCIAR INCLUSIVE O REINO DE SAZANNA.

SEM FOLEGO...



TABELA DE VENDAS

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.
1101	5/3	12
1102	7/3	23
1103	8/3	25
1104	10/3	12
1105	12/3	25

TABELA DE DESTINOS DE EXPORTAÇÃO

CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
12	REINO DE MINAMI
23	IMPÉRIO ALFA
25	REINO DE RITOL

TABELA DE DETALHES DE VENDAS

CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO	QUANTIDADE
1101	101	1100
1101	102	300
1102	103	1700
1103	104	500
1104	101	2500
1105	103	2000
1105	104	700

TABELA DE PRODUTOS

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
102	MORANGO	150G
103	MAÇÃ	120G
104	LIMÃO	200G

UM BANCO DE DADOS RELACIONAL NORMALMENTE USA TABELAS DIVIDIDAS ATÉ A TERCEIRA FORMA NORMAL.



ESTES SÃO OS RESULTADOS DA DIVISÃO DE UMA TABELA ATÉ A TERCEIRA FORMA NORMAL.

AGORA NOSSAS TABELAS ESTÃO COMPLETAS!

TOCA AQUI!!

VIVA! ❤



AGORA, VOCÊS PODEM GERENCIAR PRODUTOS, DESTINOS DE EXPORTAÇÃO, E VENDAS USANDO TABELAS,

ENTÃO PODEM GERENCIAR TUDO ISSO SEM PROBLEMAS.

PRODUTO

DESTINO DE EXPORTAÇÃO

VENDAS

SIM.

NENHUM CONFLITO VAI OCORRER MESMO QUE MAIS DADOS SEJAM ADICIONADOS.

QUE ALÍVIO...

EMBORA TENHAMOS DIVIDIDO NOSSA TABELA ORIGINAL EM MUITAS OUTRAS TABELAS,

OS DADOS ORIGINAIS ESTÃO DISTRIBUÍDOS POR TODAS ELAS.

TABELA DE DETALHES DE VENDAS

TABELA DE DEST. DE EXPORTAÇÃO

TABELA DE VENDAS

TABELA DE PRODUTOS

TABELA CRIADA A PARTIR DOS RELATÓRIOS DE VENDAS

ESTA TABELA MOSTRA RELACIONAMENTOS ENTRE OS DADOS.

EXATAMENTE! ELA É RELACIONAL.

TABELA DE VENDAS

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.
1101		

O CÓDIGO DE DESTINO DE EXPORTAÇÃO NA TABELA DE VENDAS SE REFERE AO CÓDIGO CORRESPONDENTE NA TABELA DE DESTINOS DE EXPORTAÇÃO.

TABELA DE DETALHES DE VENDAS

CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO	QUANTIDADE
1101	101	1000
1101	102	1000
1102	103	1700
1103	104	500
1104	101	2500
1105	103	?
1105	104	?

SE REFERE A...

TABELA DE DESTINOS DE EXPORTAÇÃO

CÓDIGO DEST. DE EXP.
12
23

NOME DEST. DE EXP.

REINO DE MINAMI

IMPÉRIO ALFA

REINO DE RITOL

DA MESMA FORMA, O CÓDIGO DO PRODUTO NA TABELA DE DETALHAMENTO DE VENDAS SE REFERE AO CÓDIGO EQUIVALENTE NA TABELA DE PRODUTOS,

E O CÓDIGO DE RELATÓRIO NA TABELA DE DETALHAMENTO SE REFERE AO CÓDIGO EQUIVALENTE NA TABELA DE VENDAS.

SE REFERE A...

TABELA DE PRODUTOS

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
102	MORANGO	150G
		120G
		200G

AS COLUNAS QUE SE REFEREM A OUTRAS SÃO CHAMADAS DE CHAVES ESTRANGEIRAS (FOREIGN KEY).

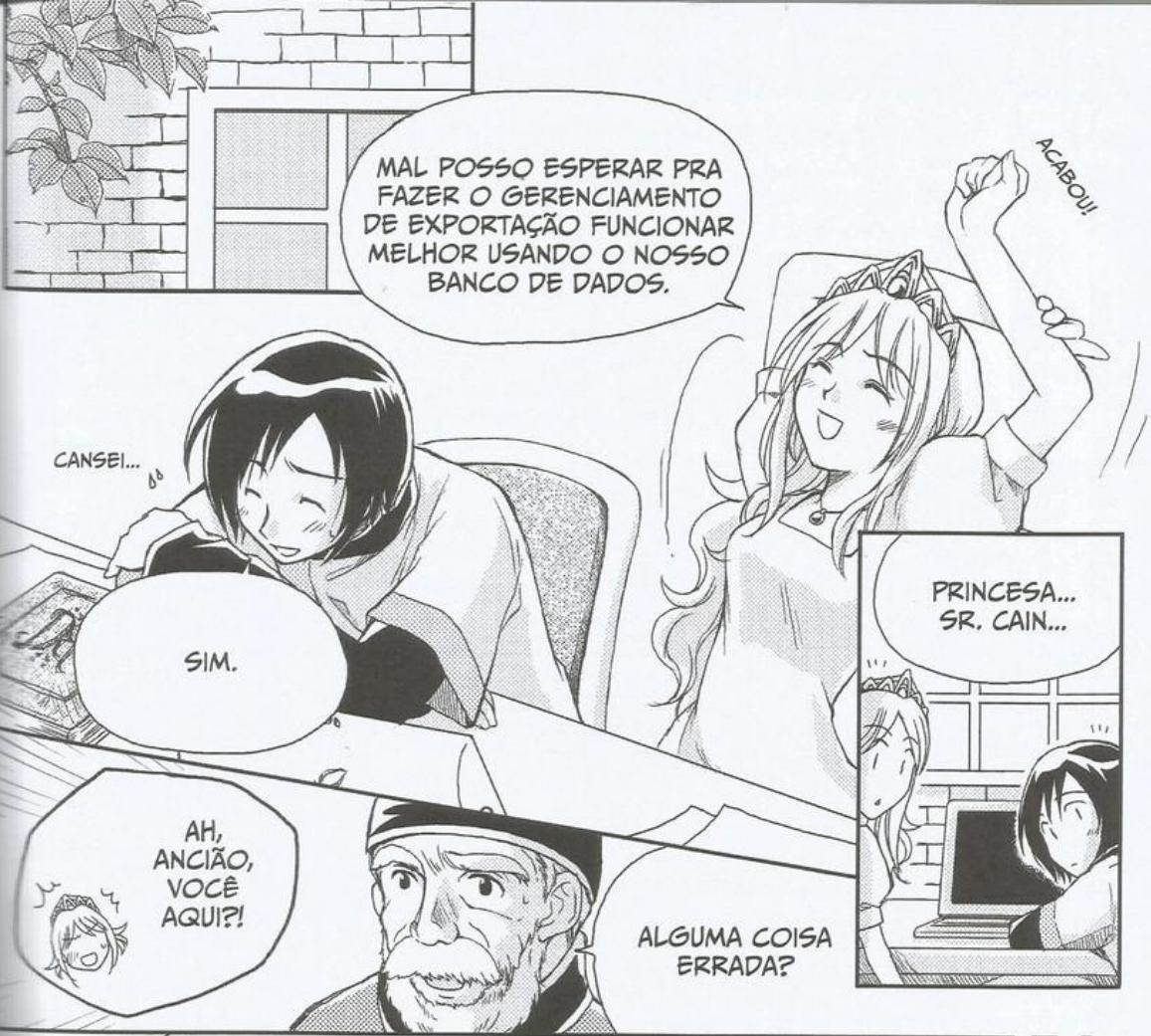
AS CHAVES ESTRANGEIRAS SE REFEREM ÀS CHAVES PRIMÁRIAS DE OUTRAS TABELAS.



MONTE BANCO DE DADOS RELACIONAL

ELAS PARECEM TABELAS SEPARADAS E INDEPENDENTES, MAS ESTÃO PROFUNDAMENTE ASSOCIADAS POR CHAVES ESTRANGEIRAS.

ESTAMOS... QUASE... LÁ... !!



O QUE É O MODELO E-R?



A princesa Ruruna e o Cain descobriram as condições reais do Reino de Kod utilizando um modelo E-R (Entidade-Relacionamento). Quando você tentar criar um banco de dados por si mesmo, o primeiro passo é determinar as condições dos dados que se está tentando modelar.

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*.

Além disso, um modelo E-R mostra relacionamentos entre entidades. A princesa Ruruna e o Cain fizeram sua análise pressupondo que existisse um relacionamento chamado *vendas* entre frutas e destinos de exportação. Frutas são exportadas para múltiplos destinos de exportação, e cada um destes importa diversos tipos de frutas. Por esse motivo, uma análise para o modelo E-R foi feita presumindo-se que existia um relacionamento chamado de *muitos-para-muitos* entre frutas e destinos de exportação. O número de associações entre entidades é chamado de *cardinalidade*.

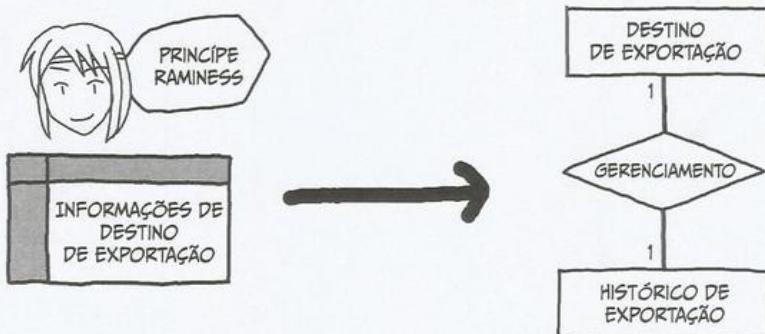


COMO ANALISAR UM MODELO E-R

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

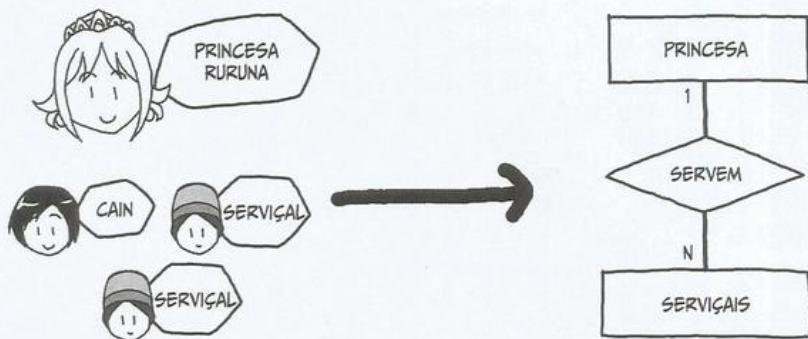
1º CASO: RELACIONAMENTO UM-PARA-UM

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

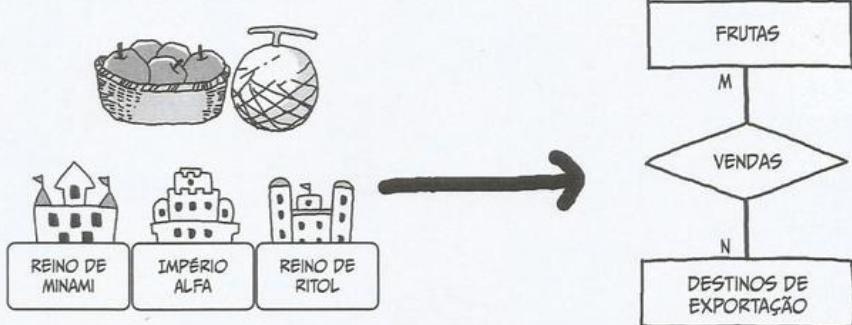
Vários serviços servem uma princesa. Os serviços não servem outra princesa, nem mesmo o rei.



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*.

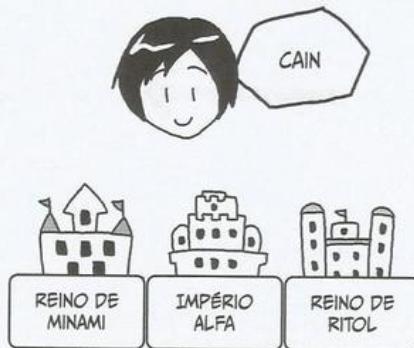


QUESTÕES

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.

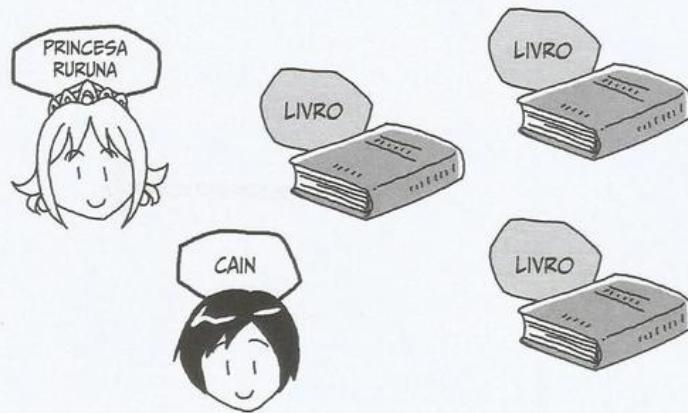
Q1

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



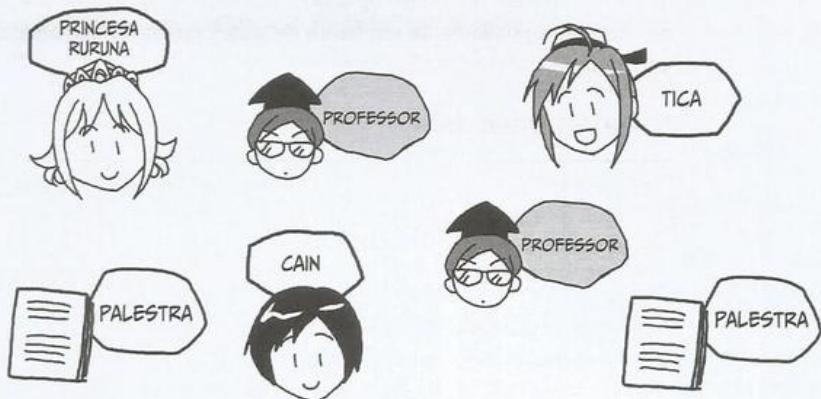
Q2

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. Vemos 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 Emprestimo	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 Emprestimo	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.



ESQUEMA INTERNO

ESQUEMA CONCEITUAL

ESQUEMA EXTERNO

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.

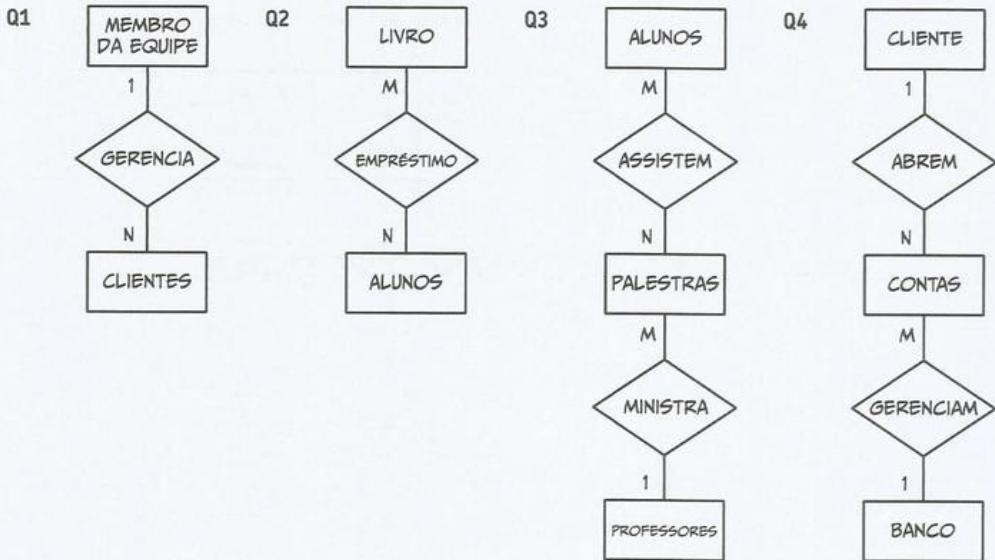
Agora que você completou o projeto básico de um banco de dados, vamos examinar seu uso no próximo capítulo.

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



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 tabela. 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.

4

VAMOS APRENDER SOBRE SQL!



USO DE SQL

ANDAR PELA
CIDADE ME FAZ
LEMBRAR MINHA
INFÂNCIA.

HA, HA, HA!

VOCÊ MUITAS VEZES
ESCAPAVA DA AULA E
FUGIA DO CASTELO.

MUITOS ANOS ATRÁS...

PRINCESA!!
PRINCESA
RURUNA!!

SÉRIO?

VOCÊ NÃO PODE SAIR
ASSIM DO CASTELO
QUANDO QUISER!

PUF
PUF
PUF

SABE DE UMA COISA,
CAIN? VOCÊ PODE
FICAR NO CASTELO
SE SÓ VAI GRITAR
COMIGO!

PUF

HUNF!

MA...MAS...

É FUNÇÃO DO
AJUDANTE SEGUIR
A PRINCESA!

SENSO DE RESPONSABILIDADE!

VOCÊ ME SEGUE
PORQUE É SEU
DEVER?

ACHEI QUE ME
SEGRIA PORQUE
ERA MEU AMIGO.

BOM...

DEIXA EU
PENSAR...

SEU BOBO!

AH, NÃO, EU NÃO
QUIS DIZER
ISSO...!

VUP!

PRINCESA...

AI!

ENCONTREI VOCÊ,
PRINCESA RURUNA!!

CHEFE DA GUARDA!
I...IGOR....!

BUM!



O CAIN TEM CUIDADO DE MIM...

DESENDE QUE EU ERA CRIANÇA.





PRÍNCIPE RAMINESS...

EEEEEE!!

HA, HA, HA

QUANTA FORMOSURA!!

RAMINESS!!

COMO VOCÊS ESTÃO, ADORÁVEIS SENHORITAS?

AH NÃO, ELE ESTÁ VINDO PARA CÁ.

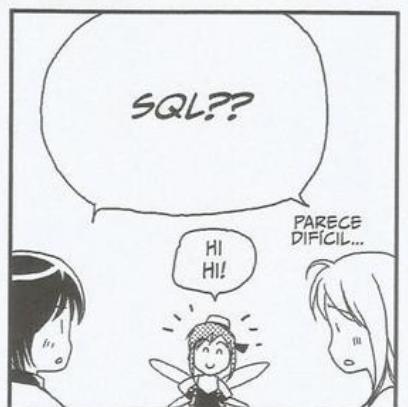
CAIN!! VAMOS ENTRAR NESTE CAFÉ.

VAMOS ESTUDAR UM POUCO DE BANCOS DE DADOS.

QUE?

AAAAA!!!!
HA, HA, HA





POR EXEMPLO,
QUANDO VOCÊ
CONVERSA COM
ALGUÉM NA REGIÃO
QUEDAPÉ NO ALÉM-
MAR,

VOCÊ PRECISA
FALAR A LÍNGUA
QUEDAPÉ.

REGIÃO
QUEDAPÉ

OH, VOCÊ
FALA O
IDIOMA
QUEDAPÉ.

SPLISH,
SPLASH....

DA MESMA FORMA,
UTILIZA-SE UMA
LINGUAGEM CHAMADA
LINGUAGEM ESTRUTURADA
DE CONSULTA (SQL,
STRUCTURED QUERY
LANGUAGE)

PARA CONVERSAR
COM UM BANCO
DE DADOS.

ISSO É ÓTIMO!!

JURA?

ESPERE,
O QUÊ?

O BANCO DE
DADOS NÃO FALA
DE VERDADE, VOCÊ
ENTENDE NÉ?

CLARO, EU SEI
DISSO!

TABELA DE VENDAS

CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.
1101	5/3	12
1102	7/3	23
1103	8/3	25
1104	10/3	12
1105	12/3	25

TABELA DE DESTINOS DE EXPORTAÇÃO

CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.
12	REINO DE MINAMI
23	IMPÉRIO ALFA
25	REINO DE RITOL

TABELA DE DETALHES DE VENDAS

CÓDIGO DO RELATÓRIO	CÓDIGO DO PRODUTO	QUANTIDADE
1101	101	1100
1101	102	300
1102	103	1700
1103	104	500
1104	101	2500
1105	103	2000
1105	104	700

TABELA DE PRODUTOS

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
102	MORANGO	150G
103	MAÇÃ	120G
104	LIMÃO	200G

VOCÊS PROJETARAM
ESSAS TABELAS NO
OUTRO DIA, LEMBRAM?





PESQUISA DE DADOS USANDO UM COMANDO SELECT



ESSES GRUPOS DE PALAVRAS SÃO CHAMADOS DE CLÁUSULAS.

NA SQL, VOCÊ ESPECIFICA OS NOMES DAS COLUNAS QUE QUER EXTRAIR COM A CLÁUSULA SELECT E O NOME DA TABELA DA QUAL VOCÊ QUER EXTRAÍ-LA COM A CLÁUSULA FROM.

FROM

TABELA DE PRODUTOS

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
102	MORANGO	150G
103	MAÇÃ	120G
104	LIMÃO	200G

SELECT

AQUI ESTÃO OS DADOS EXTRAÍDOS.

ISSO PERMITE QUE VOCÊS EXTRAÍAM TODOS OS NOMES DO PRODUTO DA TABELA RESPECTIVA.

AQUI ESTÁ! ▶

NOME DO PRODUTO
MELÃO
MORANGO
MAÇÃ
LIMÃO

ESTAMOS CONVERSANDO COM UM BANCO DE DADOS USANDO SQL.

ISSO MESMO. VOCÊ PODE EXTRAIR OS DADOS NECESSÁRIOS USANDO VÁRIOS TIPOS DE CLÁUSULAS.

VÁRIOS TIPOS... HUMM.

TUDO BEM ENTÃO, POR EXEMPLO,

QUE TAL PEDIR UMA LISTA DE PRODUTOS CUJO PREÇO UNITÁRIO É MAIOR OU IGUAL A 200G?

MAIOR OU IGUAL A 200G



NESSE CASO, VOCÊ NÃO VAI QUERER TODOS OS DADOS DE PRODUTO.

VOCÊ PRECISA APENAS RECUPERAR PRODUTOS COM PREÇO UNITÁRIO MAIOR OU IGUAL A 200G.

SIM, CLARO.

NESSES CASOS, VOCÊ ESPECIFICA CONDIÇÕES COM UMA CLÁUSULA WHERE. POR EXEMPLO,

WHERE

WHERE preco_unitario >= 200

VOCÊ ESCREVE ASSIM.

ENTENDO... MAS...

É INCONVENIENTE ESPECIFICAR O NOME DAS COLUNAS TODAS AS VEZES, NÃO?

SEM PROBLEMAS! PARA ESPECIFICAR TODAS AS COLUNAS,

É CHATO!

VOCÊ PODE USAR * (ASTERISCO)! PODEMOS RESUMIR TUDO ASSIM:

BANG!!

ESSE COMANDO RECUPERA TODOS OS DADOS DA TABELA DE PRODUTOS...

AÍ ESTÁ!

```
SELECT *
FROM produtos
WHERE preco_unitario >= 200;
```

ENTÃO,

PRODUTOS QUE CUSTAM 200G OU MAIS

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
104	LIMÃO	200G

QUE TENHAM PREÇO UNITÁRIO MAIOR OU IGUAL A 200G.

ENTÃO, SE VOCÊ MUDA OS CRITÉRIOS, PODE RECUPERAR PRODUTOS COM PREÇO UNITÁRIO MENOR DO QUE 200G.

WHERE preco_unitario < 200

ISSO MESMO - ASSIM!

AGORA PRECISAMOS APRENDER COMO CRIAR ESSES CRITÉRIOS.

DE FATO...

ENTÃO COMO EU FAÇO PARA LOCALIZAR MAÇÃS, POR EXEMPLO?

```
SELECT *  
FROM produtos  
WHERE nome_produto = 'maçã';
```

ESCREVA DESSA FORMA. QUANDO SE UTILIZAM SEQUÊNCIAS DE CARACTERES COMO CONDIÇÃO, ELAS PRECISAM ESTAR ENTRE APÓSTROFES (').

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
103	MAÇÃ	120G

SE VOCÊ FIZER ISSO, PODE EXTRAIR EXATAMENTE DADOS COM A PALAVRA "MAÇÃ", CERTO?

EXATAMENTE.

E SE VOCÊ NÃO TIVER
CERTEZA SOBRE
O NOME DE UM
PRODUTO?



O QUE FAZER
NESSE CASO?

VOCÊ COMBINA A
PALAVRA LIKE COM UM
TERMO DE BUSCA.

E ESCRVE A PARTE
DESCONHECIDA USANDO %,
ASSIM...

```
SELECT *  
FROM produtos  
WHERE nome_produto LIKE '%ão';
```

MELÃO LIMÃO

ISSO VAI RECUPERAR
NOMES DE PRODUTOS
QUE TERMINAM COM
"ÃO".

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G
104	LIMÃO	200G



E ASSIM EXTRAÍMOS
MELÃO E LIMÃO!

BASTANTE PRÁTICO.

NÃO É?



USO DE FUNÇÕES DE TOTALIZAÇÃO

VOCÊS TAMBÉM PODEM ORDENAR OS RESULTADOS EXTRAÍDOS UTILIZANDO UMA CLÁUSULA ORDER BY.

PARA ORDENAR PRODUTOS EM ORDEM ASCENDENTE DE PREÇO, ADICIONE UM COMANDO COMO ORDER BY PRECO_UNITARIO.

ASSIM VOCÊ PODE DEDUZIR INFORMAÇÕES SOBRE OS PRODUTOS.



```
SELECT *  
FROM produtos  
WHERE nome_produto LIKE '%ão'  
ORDER BY preco_unitario;
```

ISSO É
ÓTIMO!!

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
104	LIMÃO	200G
101	MELÃO	800G

QUERO APRENDER MAIS SOBRE SQL, TICA!

SÉRIO?

FICO FELIZ.

ENTÃO VEJA O QUE VOCÊ ACHA DISSO.

NA CLÁUSULA DE SELEÇÃO, USE AVG(NOME DA COLUNA) PARA OBTER A MÉDIA DE VALORES DE CADA LINHA.

```
SELECT AVG(preco_unitario)  
FROM produtos;
```

É INCRÍVEL.

ZUP!!!

AVG(PRECO_UNITARIO)

317,5

AGORA SABEMOS O PREÇO UNITÁRIO MÉDIO DOS PRODUTOS.

NÃO SABIA QUE O
PREÇO MÉDIO ERA
ASSIM TÃO ALTO...

HÁ MUITAS COISAS
SOBRE MEU PRÓPRIO
PAÍS QUE EU NÃO SEI.

SELECT COUNT(*)
FROM COUNTRY_AVG
EXATO.

A SQL TAMBÉM
TEM UMA FUNÇÃO
QUE TOTALIZA OS
VALORES DOS DADOS
RECUPERADOS..

— NÃO É ÓTIMO?

ENTÃO PODEMOS
OBTER OUTROS DADOS
ALÉM DA MÉDIA?

CLARO. POR EXEMPLO...

A QUANTIDADE DE ITENS,
SUA SOMA, MÉDIA,
VALORES MÁXIMOS
E MÍNIMOS, TODOS
PODEM SER OBTIDOS
ESPECIFICANDO UMA
FUNÇÃO DE TOTALIZAÇÃO.

FUNÇÕES DE TOTALIZAÇÃO EM SQL

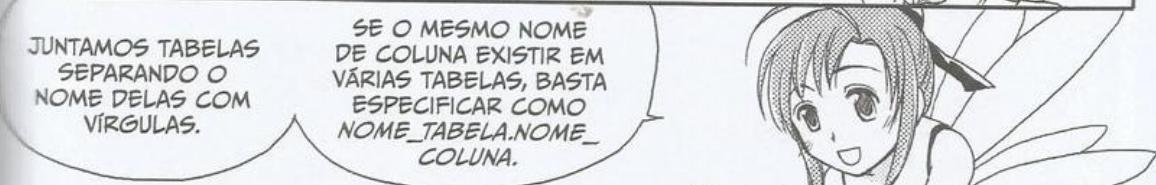
Função	Descrição
COUNT(*)	Resulta na quantidade de linhas
COUNT(nome_coluna)	Resulta na quantidade de itens que não forem nulos na coluna
COUNT(DISTINCT nome_coluna)	Resulta na quantidade de itens distintos na coluna
SUM(nome_coluna)	Resulta na soma dos valores da coluna em todas as linhas
AVG(nome_coluna)	Resulta na média dos valores da coluna por todas as linhas
MAX(nome_coluna)	Resulta no valor máximo na coluna
MIN(nome_coluna)	Resulta no valor mínimo na coluna

ASSIM...

UAU!



JUNÇÃO DE TABELAS



```
SELECT vendas.codigo_relatorio, data, vendas.cod_destino_exportacao, nome_destino_exportacao, detalhes_vendas.codigo_produto, nome_produto, preco_unitario, destino_exportacao  
  
FROM vendas, detalhes_vendas, produtos, destino_exportacao  
  
WHERE vendas.codigo_relatorio = detalhes_vendas.codigo_relatorio  
AND  
detalhes_vendas.codigo_produto = produtos.codigo_produto  
AND  
destinos_exportacao.cod_destino_exportacao =  
vendas.cod_destino_exportacao
```

DEPOIS DE JUNTAR AS TABELAS, FILTRAMOS OS RESULTADOS USANDO WHERE.



ASSIM, VOCÊ PODE RECUPERAR DADOS DE RELATÓRIO DE VENDAS DAS TABELAS, MESMO QUE ELAS ESTEJAM DIVIDIDAS.

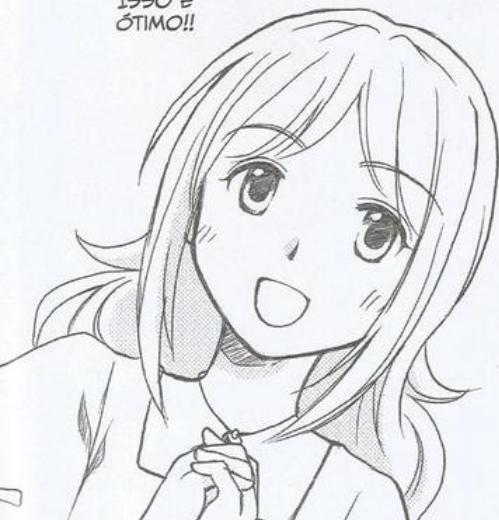
CÓDIGO DO RELATÓRIO	DATA	CÓDIGO DEST. DE EXP.	NOME DEST. DE EXP.	CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO	QUANTIDADE
1101	5/3	12	REINO DE MINAMI	101	MELÃO	800G	1100
1101	5/3	12	REINO DE MINAMI	102	MORANGO	150G	300
1102	7/3	23	IMPÉRIO ALFA	103	MAÇÃ	120G	1700
1103	8/3	25	REINO DE RITOL	104	LIMÃO	200G	500
1104	10/3	12	REINO DE MINAMI	101	MELÃO	800G	2500
1105	12/3	25	REINO DE RITOL	103	MAÇÃ	120G	2000
1105	12/3	25	REINO DE RITOL	104	LIMÃO	200G	700

ESSA É A MESMA TABELA ORIGINAL. NÓS A RECREAMOS!

ISSO É ÓTIMO!!

PODE-SE RECUPERAR DADOS RELATIVOS A RELATÓRIOS DE VENDAS MESMO QUE SE GERENCIE OS PRODUTOS, DESTINOS DE EXPORTAÇÃO E VENDAS INDEPENDENTEMENTE.

UAU!



CRIAÇÃO DE UMA TABELA



```
CREATE TABLE produtos
(
    codigo_produto int NOT NULL,
    nome_produto varchar(255),
    preco_unitario int,
    PRIMARY KEY(codigo_produto)
);
```

CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO

VOÇÊ DEVE ESPECIFICAR TAMBÉM UMA CHAVE PRIMÁRIA. EU USEI O CÓDIGO DO PRODUTO PARA ISSO.*

TAMBÉM DETERMINAMOS O TIPO DE DADOS DE CADA COLUNA. VOCÊS PODEM VER QUE O CÓDIGO É O PREÇO UNITÁRIO SÃO INTEIROS (INT). VARCHAR QUER DIZER QUE O BANCO DE DADOS VAI ESPERAR TEXTO, E (255) LIMITA NOME_PRODUTO A 255 CARACTERES.

ASSIM...

ISSO IMPIDE QUE VOCÊ INCLUA VALORES INCORRETOS.

AGORA PODEMOS
INSERIR OS DADOS NA
TABELA QUE CRIAMOS,
CERTO?

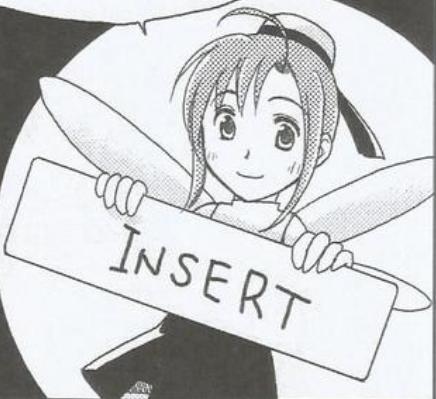


UTILIZA-SE O COMANDO
INSERT PARA INSERIR
DADOS.



CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO

CERTO.



```
INSERT INTO produtos (codigo_produto, nome_produto,  
preco_unitario)  
VALUES (101, 'melão', 800);
```

PODE-SE TAMBÉM
APAGAR (COMANDO
DELETE) E ATUALIZAR
(COMANDO UPDATE) OS
DADOS.



CÓDIGO DO PRODUTO	NOME DO PRODUTO	PREÇO UNITÁRIO
101	MELÃO	800G

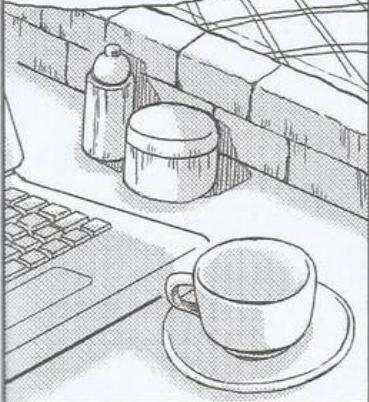


ASSIM, MELÃO FOI
INSERIDO FACILMENTE
NA TABELA DE
PRODUTOS.



E O PREÇO
UNITÁRIO DE UM
PRODUTO PODE
SER CORRIGIDO
USANDO SQL.

ASSIM VOCÊS
PODEM GERENCIAR
EXPORTAÇÃO DE
FRUTAS USANDO UM
BANCO DE DADOS.



MAS O COMANDO
SELECT É A PARTE
MAIS IMPORTANTE DA
SQL. ENTÃO ESTUDEM
BASTANTE.



OH, CÉUS, VEJA
A HORA!

VAMOS VOLTAR
PARA O CASTELO
ANTES QUE
ESCUREÇA.

SENÃO O CAPITÃO
DA GUARDA IGOR VAI
GRITAR COMIGO DE
NOVO.



É, JÁ LEVAMOS
MUITA BRONCA
DELE QUANDO
CRIANÇAS...



CORRA, EU JÁ
ESTOU INDO.

CERTO,
VAMOS.

CORRE
CORRE



VISÃO GERAL SOBRE SQL



Neste capítulo, a princesa Ruruna e o Cain aprenderam a respeito de *SQL*, ou *Linguagem Estruturada de Consulta*, uma linguagem utilizada para operar bancos de dados relacionais. Os comandos SQL podem ser divididos em três tipos distintos:

Linguagem de Definição de Dados (DDL, Data Definition Language) Para criar tabelas.

Linguagem de Manipulação de Dados (DML, Data Manipulation Language) Entrada e recuperação de dados.

Linguagem de Controle de Dados (DCL, Data Control Language) Gerencia acesso de usuários.

A SQL tem comandos que criam a estrutura geral do banco de dados, e comandos que criam tabelas dentro do banco de dados. Pode-se usar essa linguagem para modificar e apagar tabelas da mesma forma. A parte da linguagem que tem essa função é chamada de *Linguagem de Definição de Dados (DDL)*.

A SQL também tem comandos que manipulam os dados no banco de dados, seja para inserção, exclusão ou modificação. Há também comandos que permitem a pesquisa de dados. A parte da linguagem com essa função é chamada de *Linguagem de Manipulação de Dados (DML)*.

Além disso, a SQL oferece a capacidade de controlar o banco de dados, de forma que conflitos de dados não ocorrerão mesmo que várias pessoas o utilizem ao mesmo tempo. A parte da linguagem com essa função é chamada de *Linguagem de Controle de Dados (DCL)*.

PESQUISA NOS DADOS USANDO UM COMANDO SELECT

A princesa Ruruna e o Cain começaram a aprender SQL pela sua função de pesquisa básica. A SQL busca dados quando um *comando* (uma combinação de cláusulas) é apresentado. Para pesquisar se certo produto com preço unitário de 200G existe, por exemplo, utilizaria-se o seguinte comando SQL.

```
SELECT *
FROM produtos
WHERE preco_unitario=200
```

Cria-se um comando SQL combinando-se cláusulas.

O comando SELECT é o mais básico deles. Ele retorna *determinadas colunas de determinadas tabelas (FROM)*, *atendendo certas condições (WHERE)*. Pode-se combinar essas cláusulas para criar comandos de consulta intuitivos em SQL - mesmo um usuário que não esteja familiarizado com bancos de dados pode fazer pesquisas.

CRIAÇÃO DE CRITÉRIOS

Cain disse anteriormente "Agora precisamos aprender como criar critérios". Vamos examinar algumas formas de criá-los utilizando SQL.

OPERADORES DE COMPARAÇÃO

Uma forma de criar critérios é utilizando *operadores de comparação* como \geq e $=$. Por exemplo, a condição "A é maior ou igual a B" é expressa utilizando \geq , e o critério "A é igual a B" é escrito utilizando $=$. Mais exemplos de operadores de comparação são mostrados na tabela abaixo.

OPERADORES DE COMPARAÇÃO

Operador de Comparação	Descrição	Exemplo	Descrição do Exemplo
A = B	A é igual a B.	preco_unitario=200	O preço unitário é 200G.
A > B	A é maior que B.	preco_unitario>200	O preço unitário é maior que 200G.
A \geq B	A é maior ou igual a B.	preco_unitario \geq 200	O preço unitário é maior ou igual a 200G.
A < B	A é menor do que B.	preco_unitario<200	O preço unitário é menor que 200G.
A \leq B	A é menor ou igual a B.	preco_unitario \leq 200	O preço unitário é menor ou igual a 200G.
A \neq B	A é diferente de B.	preco_unitario \neq 200	O preço unitário não pode ser 200G.

OPERADORES LÓGICOS

Em alguns casos, é preciso expressar condições que são mais complexas do que uma simples comparação. Pode-se utilizar os operadores lógicos (*AND*, "e", *OR*, "ou" e *NOT*, "não") para combinar condições baseadas em operadores de comparação e criar critérios mais complexos, como mostra a tabela abaixo.

OPERADORES LÓGICOS

Operador Lógico	Descrição	Exemplo	Descrição do Exemplo
AND	A e B	codigo_produto \geq 200 AND preco_unitario = 100	O código do produto é maior ou igual a 200 e o preço unitário é 100G.
OR	A ou B	codigo_produto \geq 200 OR preco_unitario = 100	O código do produto é maior ou igual a 200 ou o preço unitário é 100G.
NOT	Não A	NOT preco_unitario = 100	O preço unitário não pode ser 100G.

PADRÕES

Quando não se sabe exatamente o que se está procurando, pode-se usar pesquisa de padrões (pattern matching) em condições utilizando caracteres curinga. Quando se faz isso, utilizam-se caracteres como % ou _ em um comando LIKE (semelhante a), e isso vai pesquisar por sequências de caracteres que correspondem aos padrões especificados. Pode-se pesquisar valores que correspondam parcialmente a uma sequência informada usando %, que indica uma sequência de caracteres de qualquer tamanho, e _, que indica apenas um único caractere. Um exemplo de consulta utilizando caracteres curinga é apresentado a seguir. O comando no exemplo procura por uma sequência de caracteres com "ão" no final, no nome do produto.

SELECT *
FROM produtos
WHERE nome_produto LIKE '%ão';

Este comando procura padrões utilizando um caractere curinga.

Código do produto	Nome do produto	Preço unitário
101	Melão	800 G
104	Limão	200 G

Os caracteres curinga que podem ser usados em SQL são mostrados abaixo.

CARACTERES CURINGA

Caractere curinga	Descrição	Exemplo de padrão	Seqüência correspondente de exemplo
%	Corresponde a qualquer quantidade de caracteres	%ão m%	Limão Melão Maçã Morango
_	Corresponde a um único caractere	_o o_	aquilo outra

PESQUISAS

Existe ainda muitos outros métodos de busca. Por exemplo, pode-se especificar BETWEEN (entre) X e Y para obter um intervalo de valores. Se um intervalo for indicado como mostrado abaixo, pode-se extrair produtos com preços unitários maiores ou iguais a 150 G e menores que 200 G.

SELECT *
FROM produtos
WHERE preco_unitario
BETWEEN 150 AND 200;

Especifica um intervalo de busca

Adicionalmente, pode-se especificar IS NULL (é nulo) quando se efetuam pesquisas em linhas. Se o comando mostrado abaixo for usado, pode-se recuperar produtos com preço unitário nulo.

SELECT *
FROM produtos
WHERE preco_unitario is NULL;

Procura por nulos



QUESTÕES

Agora vamos criar comandos SQL utilizando diversos tipos de condições. Usaremos a Tabela de Destinos de Exportação a seguir (que utiliza 10.000 como unidade de população). Responda as perguntas abaixo utilizando comandos SQL. As respostas estão na página 119.

TABELA DE DESTINOS DE EXPORTAÇÃO

Cód. de destino de exportação	Nome de destino de exportação	População
12	Reino de Minami	100
23	Império Alfa	120
25	Reino de Ritol	150
30	Reino de Sazanna	80

Q1

Para encontrar países em que a população seja maior ou igual a 1 milhão, extraia a tabela abaixo.

Cód. de destino de exportação	Nome de destino de exportação	População
12	Reino de Minami	100
23	Império Alfa	120
25	Reino de Ritol	150

Q2

Para encontrar países em que a população seja menor que 1 milhão, extraia a tabela abaixo.

Cód. de destino de exportação	Nome de destino de exportação	População
30	Reino de Sazanna	80

Q3

Determine os países em que o código de destino de exportação seja menor que 20 e a população maior ou igual a 1 milhão.

Q4

Encontre os países em que o código de destino de exportação seja maior ou igual a 30 e a população maior que 1 milhão.

Q5

Qual a população do Reino de Ritol?

Q6

Encontre países com nomes contendo a letra n.

FUNÇÕES DE AGREGAÇÃO NUMÉRICAS



A princesa Ruruna e o Cain aprenderam sobre várias funções de agregação. Elas são também conhecidas como *funções de conjunto*. Pode-se utilizá-las para totalizar informações, para obter, por exemplo, valores máximos e mínimos, quantidade de itens, e somas.

Se uma cláusula WHERE for especificada juntamente com uma função de conjunto, pode-se obter a totalização apenas das linhas especificadas. Se uma cláusula como a mostrada abaixo for usada, pode-se determinar a quantidade de produtos com preços unitários maiores ou iguais a 200 G.

```
SELECT COUNT(*)  
FROM produtos  
WHERE preco_unitario >= 200;
```



COUNT(*)
2

TOTALIZAÇÃO DE DADOS COM AGRUPAMENTO

Se os dados forem agrupados, pode-se obter valores totalizados facilmente. Por exemplo, se deseja-se obter a quantidade de produtos e preços unitários médios baseando-se em distritos, é possível utilizar funções de agrupamento.

Para agrupar dados, combinam-se as funções de agregação com a cláusula GROUP BY (agrupar por). Vamos utilizar a tabela de produtos mostrada abaixo.

TABELA DE PRODUTOS

Código do produto	Nome do produto	Preço Unitário	Distrito
101	Melão	800G	Mar do Sul
102	Morango	150G	Central
103	Maçã	120G	Mar do Norte
104	Limão	200G	Mar do Sul
201	Pinhão	100G	Mar do Norte
202	Caqui	160G	Central
301	Pêssego	130G	Mar do Sul
302	Kiwi	200G	Mar do Sul

Para obter o preço unitário médio por distrito nessa Tabela de Produtos, especifique a coluna Distrito e a função AVG, bem como a cláusula GROUP BY. Isso agrupará os dados baseando-se em distritos e informará o valor unitário médio de produtos em cada distrito.

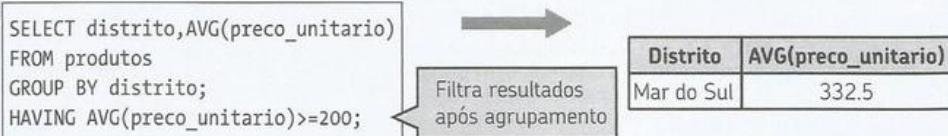
```
SELECT distrito, AVG(preco_unitario)  
FROM produtos  
GROUP BY distrito;
```

Ativa o agrupamento



Distrito	AVG(preco_unitario)
Mar do Sul	332.5
Mar do Norte	110
Central	155

E se você quiser restringir mais seus resultados, baseando-se em uma propriedade particular dos dados? Imagine que você queira encontrar médias regionais de preços de produtos maiores ou iguais a 200G. Nesse caso, não especifique uma condição numa cláusula WHERE, mas sim utilize uma cláusula HAVING. Isso permite que se extraia apenas distritos em que o preço unitário médio seja maior ou igual a 200G.



QUESTÕES



Responda às questões abaixo utilizando a seguinte Tabela de Destinos de Exportação (presumindo que a unidade de população seja 10.000). As respostas estão na página 120.

TABELA DE DESTINOS DE EXPORTAÇÃO

Código de destino de exportação	Nome de destino de exportação	População	Distrito
12	Reino de Minami	100	Mar do Sul
15	Reino de Paronu	200	Central
22	Reino de Tokanta	160	Mar do Norte
23	Império Alfa	120	Mar do Norte
25	Reino de Ritol	150	Mar do Sul
30	Reino de Sazanna	80	Mar do Sul
31	Reino de Taharu	240	Mar do Norte
33	Reino de Mariyon	300	Central

Q7

Qual a menor população?

Q8

Qual a maior população?

Q9

Qual a população total de todos os países na Tabela de Destinos de Exportação?

Q10

Qual a população total de todos os países com código de destino de exportação maior que 20?

Q11

Quantos países com população maior ou igual a 1 milhão constam na tabela?

Q12

Quantos países pertencem ao distrito do Mar do Norte?

Q13

Qual país do distrito do Mar do Norte tem a maior população?

Q14

Qual a população total de todos os países exceto o Reino de Ritol?

Q15

Encontre os distritos com população média maior ou igual a 2 milhões.

Q16

Encontre os distritos que contenham pelo menos três países.

PESQUISA DE DADOS



Existem métodos mais complexos de consulta disponíveis na SQL, além daqueles que já discutimos.

USO DE UMA SUBCONSULTA

Por exemplo, pode-se embutir uma consulta dentro de outra. Consultas embutidas são conhecidas como *subconsultas*. Vamos examinar as tabelas abaixo.

TABELA DE PRODUTOS

Código do produto	Nome do produto	Preço Unitário
101	Melão	800G
102	Morango	150G
103	Maçã	120G
104	Limão	200G

TABELA DE DETALHES DE VENDAS

Código do relatório	Código do produto	Quantidade
1101	101	1100
1101	102	300
1102	103	1700
1103	104	500
1104	101	2500
1105	103	2000
1105	104	700

Pode-se utilizar essas duas tabelas para pesquisar por nomes de produtos cujo volume de vendas seja maior ou igual a 1000. O seguinte comando SQL vai efetuar essa pesquisa.

```
SELECT * FROM produtos
WHERE codigo_produto IN
(SELECT codigo_produto
FROM detalhes_vendas
WHERE quantidade>=1000);
```

Este comando contém uma subconsulta.

Em um comando SQL, o comando SELECT entre parênteses é efetuado primeiro: o código do produto na Tabela de Detalhes de Vendas é pesquisado em primeiro lugar, e os códigos 101 e 103 são encontrados (já que são esses os relatórios com volumes de venda maior do que 1000). Esses códigos são então usados como parte da condição do comando SELECT fora dos parênteses. Para o termo IN, a condição é satisfeita quando uma linha corresponde a qualquer valor constante dentro dos parênteses. Dessa forma, produtos que correspondam aos códigos 101 e 103 serão retornados.

Em outras palavras, no caso de uma subconsulta, o resultado do comando SELECT dentro dos parênteses será enviado para o outro comando SELECT para pesquisa. As seguintes informações serão o resultado da consulta completa.

Código do produto	Nome do produto	Preço Unitário
101	Melão	800G
103	Maçã	120G

USO DE UMA SUBCONSULTA CORRELADA

Vamos considerar uma subconsulta que esteja *contida* dentro de outra consulta. Estas são conhecidas como *subconsultas correlatas*. No comando abaixo, a tabela detalhes_vendas na consulta externa recebe temporariamente o nome U para que a subconsulta possa se referir a ela sem ambiguidade. A sintaxe U.codigo_produto indica com qual coluna codigo_produto estamos trabalhando, uma vez que existem duas fontes para essa coluna dentro da consulta.

Já que a subconsulta se refere a dados da consulta externa, elas não são independentes, como no exemplo anterior. Essa dependência é chamada de *correlação*.

① `SELECT *
FROM detalhes_vendas U

② WHERE quantidade>

③ (SELECT AVG(quantidade)
FROM detalhes_vendas
WHERE codigo_produto=U.codigo_produto);`

Código do relatório	Código do produto	Quantidade
1104	101	2500
1105	103	2000
1105	104	700

Esta consulta extraí comandos com volume de vendas maior do que a média para o produto.

Vamos examinar o processamento da subconsulta correlata. Numa delas, a consulta externa é efetuada primeiro.

① `SELECT *
FROM detalhes_venda U`

Esses resultados são enviados para a consulta interna para serem avaliados linha por linha. Vamos analisar a avaliação da primeira linha, com código do produto 101.

③ (SELECT AVG(quantidade)
FROM detalhes_vendas
WHERE codigo_produto=101)

O código do produto da primeira linha é 101, ou seja, melões - a quantidade média de vendas para eles é 1800. Esse resultado é enviado então para a condição da consulta externa.

② WHERE quantidade>(1800)

Esse processo continua para todas as linhas de detalhes de vendas - os passos ② e ③ são executados para todos os códigos de produtos possíveis. Em outras palavras, esta consulta extrai relatórios onde o volume de vendas da fruta seja maior do que a média de vendas daquela fruta em particular. Consequentemente, apenas a 5a, 6a e 7a linhas de ① são extraídas.



QUESTÕES

Agora, responda às seguintes questões com base na Tabela de Produtos e na de Detalhes de Vendas. As respostas estão na página 122.

Q17

Encontre o detalhe de venda para frutas com preços unitários maiores ou iguais a 300G, e extraia a tabela abaixo.

Código do relatório	Código do produto	Quantidade
1101	101	1100
1104	101	2500

Q18

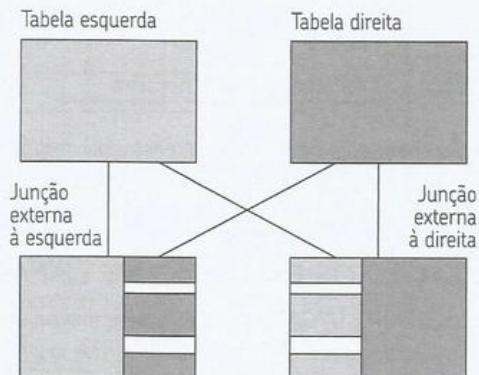
Obtenha o volume médio de vendas por produto e encontre itens que tenham vendas abaixo da média.

JUNTANDO TABELAS

Depois efetuar uma pesquisa baseada em SQL, a princesa Ruruna e o Cain criaram um relatório de vendas combinando tabelas. Juntar tabelas combinando colunas com os mesmos nomes é chamado de *equijunção*. Nesse tipo de junção, linhas com os mesmos valores são utilizadas como condição de junção para fundir as tabelas. Juntar colunas com o mesmo nome numa única é conhecido como *junção natural*.

O método de junção no qual apenas linhas que tenham um valor comum como equijunção são selecionadas é chamado *junção interna (inner join)*.

Em contraste, o método de junção que mantém todas as linhas de uma tabela e especifica um nulo para linhas não incluídas na outra é chamado de *junção externa (outer join)*. Se você colocar uma tabela criada de uma junção externa à direita ou à esquerda de um comando SQL, ela é chamada de *junta externa à esquerda* ou uma *junta externa à direita*, dependendo de quais linhas sejam mantidas.



CRIAÇÃO DE UMA TABELA



Finalmente a Princesa Ruruna e o Cain aprenderam sobre a sintaxe de comando que cria uma tabela, `CREATE TABLE`. A sintaxe desse comando frequentemente depende do tipo específico de banco de dados que você usa. Um exemplo é mostrado abaixo.

```
CREATE TABLE produtos
(
    codigo_produto int NOT NULL,
    nome_produto varchar(255),
    preco_unitario int,
    PRIMARY KEY(codigo_produto)
);
```

Esse comando cria uma tabela.

Quando você cria uma tabela, deve especificar seus nomes de coluna. Adicionalmente, pode informar uma chave primária e uma chave estrangeira para cada coluna. Neste exemplo, o código do produto é especificado como uma `PRIMARY KEY` (chave primária) e o código do produto não pode ser nulo. Quando se cria uma tabela, pode ser necessário incluir as seguintes especificações.

Restrição	Descrição
PRIMARY KEY	Cria uma chave primária
UNIQUE	Deve ser única
NOT NULL	Não aceita um valor NULL
CHECK	Confere um intervalo
DEFAULT	Cria um valor padrão
FOREIGN KEY / REFERENCES	Cria uma chave estrangeira

Essas especificações são chamadas de *restrições* (*constraint*). Incluir restrições quando se cria uma tabela ajuda a evitar conflitos de dados posteriormente e permite que se gerencie de forma correta o banco de dados.

INSERÇÃO, ATUALIZAÇÃO OU EXCLUSÃO DE LINHAS

Você pode usar os comandos INSERT, UPDATE e DELETE para inserir, atualizar ou excluir dados de uma tabela criada pelo comando CREATE TABLE. Vamos inserir, atualizar e excluir alguns dados usando SQL.

```
INSERT INTO produtos
(codigo_produto, nome_produto, preco_unitario)
VALUES (200, 'cereja', 200);
```

Esse comando acrescenta cereja.

```
UPDATE produtos
SET nome_produto='cantalupo'
WHERE nome_produto='melão';
```

Esse comando atualiza melão para cantalupo.

```
DELETE FROM produtos
WHERE nome_produto='maçã';
```

Esse comando apaga maçã.

Cód. do produto	Nome do produto	Preço unitário
101	Cantalupo	800 G
102	Morango	150 G
103	Maçã	120 G
104	Limão	200 G
200	Cereja	200 G

Atualizado para cantalupo.

Maçã é excluída.

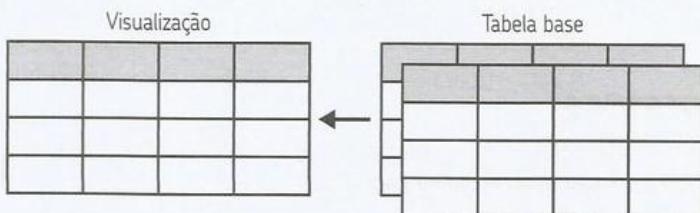
Cereja é acrescentada.

Quando inserir, atualizar ou excluir uma linha, você não pode violar as restrições estabelecidas pelo comando CREATE TABLE. Se um produto com código 200 já existisse, não se poderia acrescentar cereja, já que você não pode adicionar dados duplicados numa chave primária. Ao inserir, atualizar ou excluir dados em um banco de dados, deve-se considerar as restrições da mesma.



CRIAÇÃO DE UMA VISUALIZAÇÃO (VIEW)

Com base na tabela que criou com o comando CREATE TABLE, você também pode criar uma tabela virtual que existe apenas quando é visualizada por um usuário. Elas se chamam *visualizações*. A tabela da qual uma visualização é derivada chama-se *tabela base*.



Utilize o comando SQL mostrado abaixo para criar uma visualização.

```
CREATE VIEW produto_caro
(codigo_produto, nome_produto, preço_unitario)
AS SELECT *
FROM produtos
WHERE preco_unitario>=200;
```

Esse comando cria uma visualização.

A Tabela de Produto Caro é uma visualização baseada na Tabela de Produtos, que é uma tabela base. Ela foi criada extraíndo os dados com preços unitários maiores ou iguais a 200G da Tabela de Produtos.

TABELA DE PRODUTO CARO

Código do produto	Nome do produto	Preço Unitário
101	Melão	800G
104	Limão	200G
202	Caqui	200G

Depois que você criar a visualização Produto Caro, pode procurar por dados da mesma forma que faria em uma tabela base.

```
SELECT *
FROM produto_caro
WHERE preco_unitario>=500;
```

Permite que a visualização seja usada da mesma forma que uma tabela base

É prático criar uma visualização quando se quer manter pública parte dos dados em uma tabela.

Há também os comandos SQL para excluir uma tabela base ou visualização. O comando usado para excluir uma tabela base ou visualização é mostrado abaixo.

```
DROP VIEW produto_caros;
```

```
DROP TABLE produto;
```



QUESTÕES

Crie comandos SQL para as questões a seguir (supondo que a unidade para população é 10.000). As respostas estão na página 123.

Q19

A Tabela de Destino de Exportação a seguir foi criada usando o comando CREATE TABLE. Acrescente os dados abaixo.

TABELA DE DESTINO DE EXPORTAÇÃO

Código de destino de exportação	Nome de destino de exportação	População	Distrito
12	Reino de Minami	100	Mar do Sul
15	Reino de Paronu	200	Central
22	Reino de Tokanta	160	Mar do Norte
23	Império Alfa	120	Mar do Norte

Q20

Usando a Tabela de Destino de Exportação na Q19, crie uma visualização chamada *Região do Mar do Norte* que mostre países pertencentes ao distrito do Mar do Norte.

TABELA DE DESTINO DE EXPORTAÇÃO

Código de destino de exportação	Nome de destino de exportação	População
22	Reino de Tokanta	160
23	Império Alfa	120

Q21

Altere a população do Reino de Tokanta na Tabela de Destino de Exportação para 1,5 milhão.

Q22

Na Tabela de Destino de Exportação, apague todos os dados do Reino de Paronu.

RESUMO



- Você pode usar funções SQL para definir, operar e controlar dados.
- Para procurar por dados, use um comando SELECT.
- Para especificar uma condição, use uma cláusula WHERE.
- Para inserir, atualizar e excluir dados, use os comandos INSERT, UPDATE e DELETE.
- Para criar uma tabela, use um comando CREATE TABLE.

RESPOSTAS

Q1

```
SELECT *
FROM destinos_exportacao
WHERE populacao>=100;
```

Q2

```
SELECT *
FROM destinos_exportacao
WHERE populacao<100;
```

Q3

```
SELECT *
FROM destinos_exportacao
WHERE codigo_destino_exportacao<20
AND populacao>=100;
```

Código de destino de exportação	Nome de destino de exportação	População
12	Reino de Minami	100

Q4

```
SELECT *
FROM destinos_exportacao
WHERE codigo_destino_exportacao>=30
AND populacao>100;
```

Código de destino de exportação	Nome de destino de exportação	População
23	Império Alfa	120
25	Reino de Ritol	150
30	Reino de Sazanna	80

Q5

```
SELECT populacao  
FROM destinos_exportacao  
WHERE nome_destino_exportacao='Reino de Ritol';
```

População
150

Q6

```
SELECT *  
FROM destinos_exportacao  
WHERE nome_destino_exportacao LIKE '%n%';
```

Código de destino de exportação	Nome de destino de exportação	População
12	Reino de Minami	100
30	Reino de Sazanna	80

Q7

```
SELECT MIN(populacao)  
FROM destinos_exportacao;
```

MIN(populacao)
80

Q8

```
SELECT MAX(populacao)  
FROM destinos_exportacao;
```

MAX(populacao)
300

Q9

```
SELECT SUM(populacao)  
FROM destinos_exportacao;
```

SUM(populacao)
1350

Q10

```
SELECT SUM(populacao)
FROM destinos_exportacao
WHERE codigo_destino_exportacao>20;
```

SUM(populacao)

1050

Q11

```
SELECT COUNT(*)
FROM destinos_exportacao
WHERE populacao>=100;
```

COUNT(*)

7

Q12

```
SELECT COUNT(*)
FROM destinos_exportacao
WHERE distrito='mar do norte';
```

COUNT(*)

3

Q13

```
SELECT MAX(populacao)
FROM destinos_exportacao
WHERE distrito='mar do norte';
```

MAX(populacao)

240

Q14

```
SELECT SUM(populacao)
FROM destinos_exportacao
WHERE NOT(nome_destino_exportacao = 'Reino de Ritol');
```

SUM(populacao)
1200

Q15

```
SELECT distrito, AVG(populacao)
FROM destinos_exportacao
GROUP BY distrito
HAVING AVG(populacao) > 200;
```

Distrito	AVG(populacao)
Central	250

Q16

```
SELECT distrito, COUNT(*)
FROM destinos_exportacao
GROUP BY distrito
HAVING COUNT(*)>=3;
```

Distrito	COUNT(*)
Mar do Norte	3
Mar do Sul	3

Q17

```
SELECT *
FROM detalhes_vendas
WHERE codigo_produto IN
(SELECT codigo_produto
FROM produto
WHERE preco_unitario>=300);
```

Q18

```
SELECT *
FROM detalhes_vendas U-
WHERE quantidade<
(SELECT AVG(quantidade)
FROM detalhes_vendas
WHERE codigo_produto = U.codigo_produto);
```

Código do relatório	Código do produto	Quantidade
1101	101	1100
1102	103	1700
1103	104	500

Q19

```
INSERT INTO destino_exportacao(codigo_destino_exportacao,
nome_destino_exportacao, populacao, distrito)
VALUES(12, 'Reino de Minami', 100, 'mar do sul');
INSERT INTO destino_exportacao(codigo_destino_exportacao,
nome_destino_exportacao, populacao, distrito)
VALUES(15, 'Reino de Paronu', 200, 'central');
INSERT INTO destino_exportacao(codigo_destino_exportacao,
nome_destino_exportacao, populacao, distrito)
VALUES(23, 'Império Alfa', 120, 'mar do norte');
INSERT INTO destino_exportacao(codigo_destino_exportacao,
nome_destino_exportacao, populacao, distrito)
VALUES(22, 'Reino de Tokanta', 160, 'mar do norte');
```

Q20

```
CREATE VIEW regiao_mar_norte(codigo_destino_exportacao,
nome_destino_exportacao, populacao)
AS SELECT codigo_destino_exportacao, nome_destino_exportacao, populacao
FROM destinos_exportacao
WHERE distrito = 'mar do norte';
```

Q21

```
UPDATE destinos_exportacao
SET populacao = 150
WHERE nome_destino_exportacao = 'Reino de Tokanta';
```

Q22

```
DELETE FROM destinos_exportacao
WHERE nome_destino_exportacao = 'Reino de Paronu';
```

PADRONIZAÇÃO DE SQL

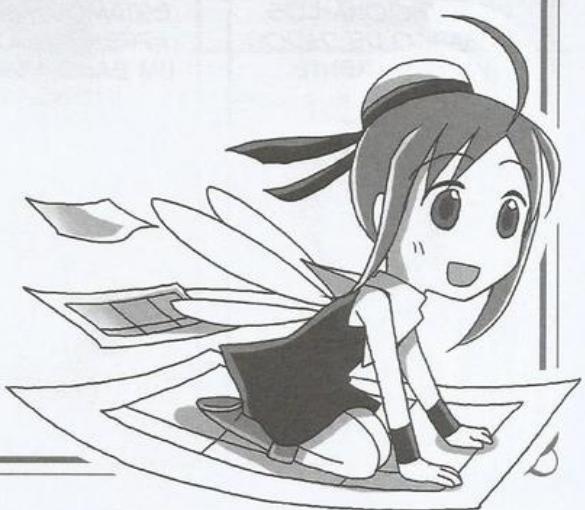
SQL é padronizado pela Organização Internacional para Padronização (ISO, International Organization for Standardization). No Japão, ela é padronizada pelo JIS (Padrões Industriais Japoneses).

Padrões SQL incluem o SQL92, criado em 1992, e o SQL99, criado em 1999. Produtos de bancos de dados relacionais são feitos para que consultas possam ser feitas de acordo com esses padrões.

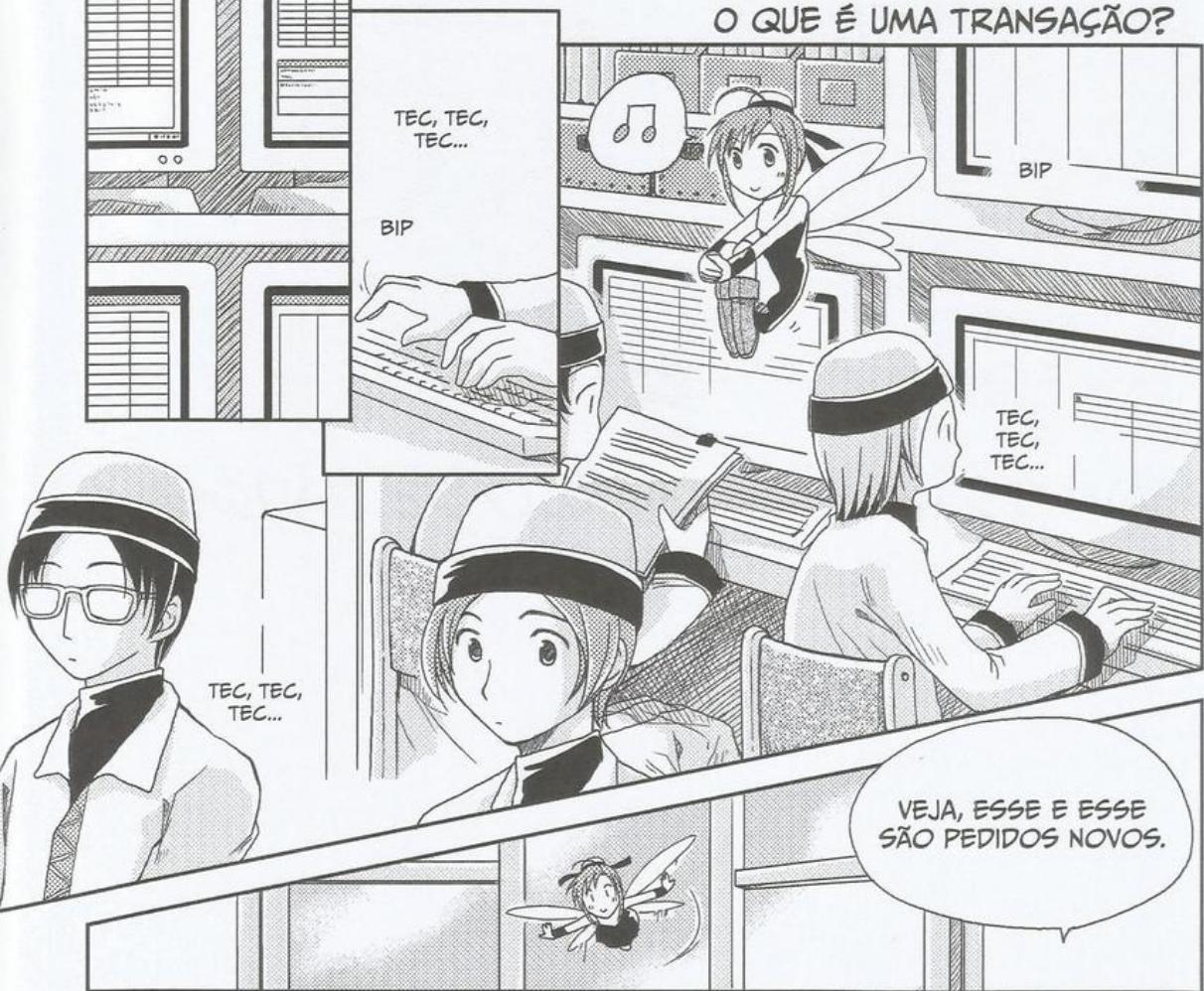
Alguns produtos de bancos de dados relacionais possuem suas próprias especificações. Consulte o manual de operação de seu produto de banco de dados para mais informações.

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?

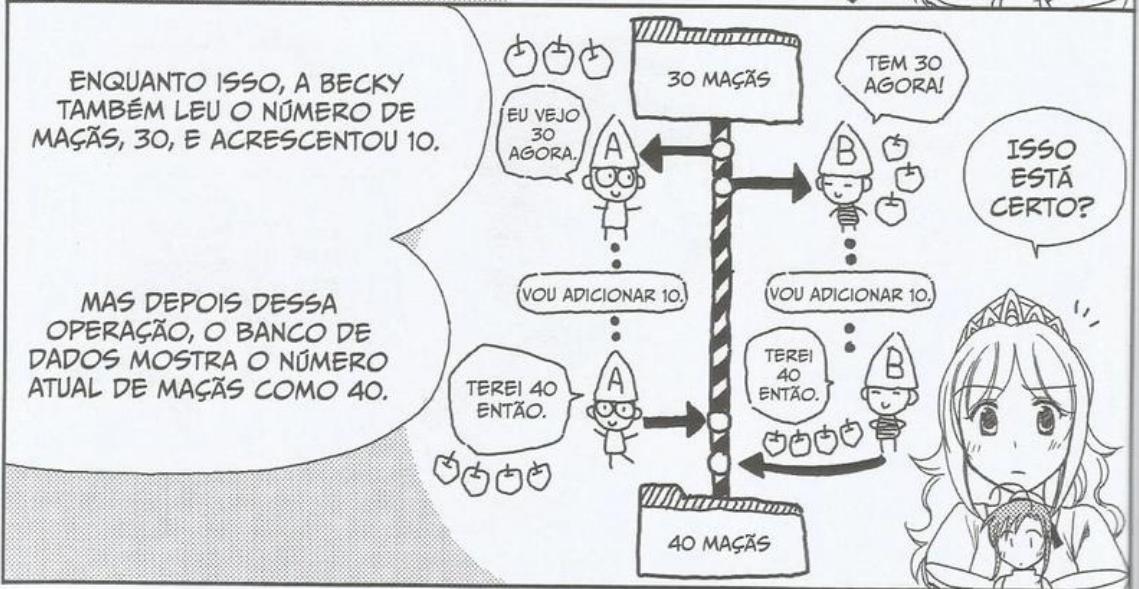
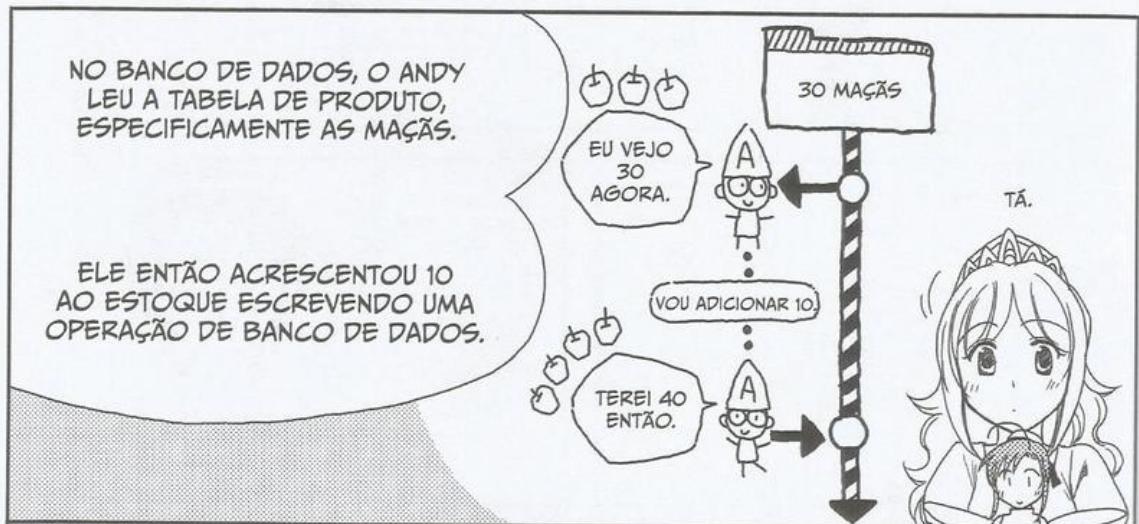
EU ATÉ PREPAREI
UMAS ILUSTRAÇÕES
PARA AJUDAR EM
SEU ENTENDIMENTO!

UAU, QUE
ÓTIMO.

TEATRO DE
BANCO DE
DADOS

ADORO UM BOM
SHOW!

OH!





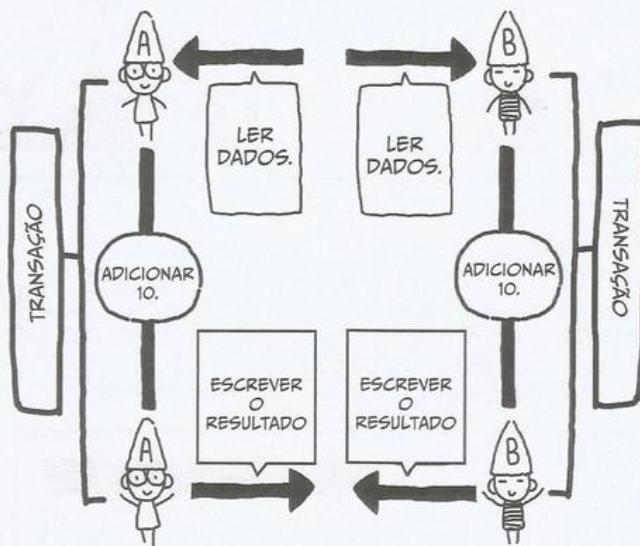
EM PRIMEIRO LUGAR, UM BANCO DE DADOS É FEITO PARA PROCESSAR OPERAÇÕES DE DADOS EM GRANDES BLOCOS.



UMA UNIDADE DE OPERAÇÕES DE DADOS É CHAMADA DE TRANSAÇÃO.



NESTE EXEMPLO, UMA OPERAÇÃO DE LEITURA, UMA DE ADIÇÃO E UMA DE ESCRITA SÃO PROCESSADAS COMO UMA ÚNICA TRANSAÇÃO.



ENTÃO, AS OPERAÇÕES DO ANDY FORMAM UMA TRANSAÇÃO,

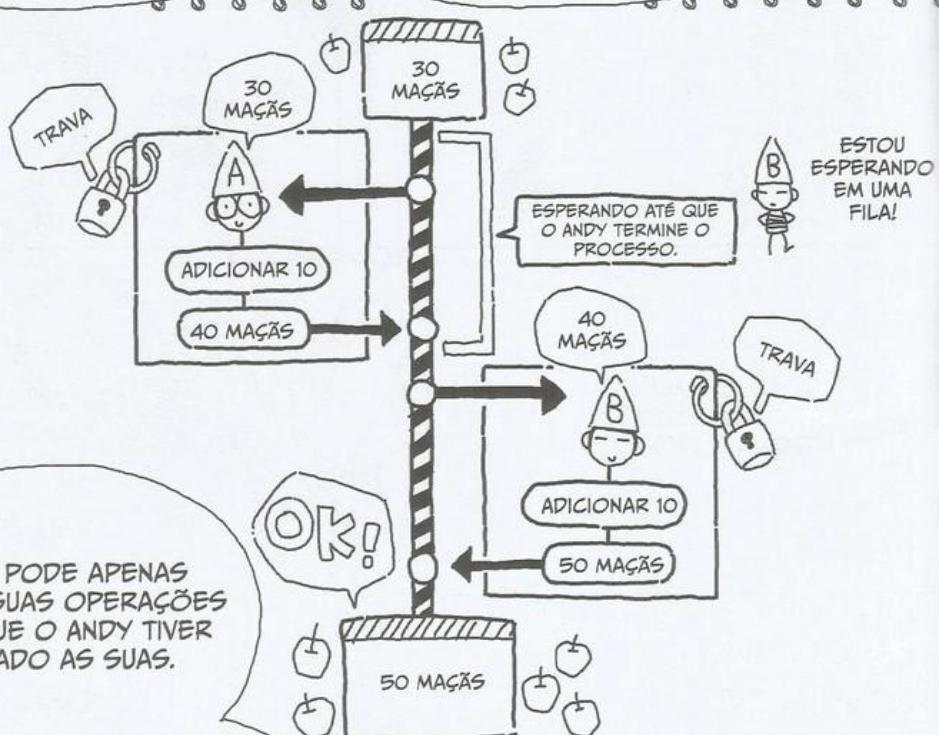
E AS DA BECKY FORMAM OUTRA.

O QUE É UMA TRAVA (BLOQUEIO)?



O ANDY TRAVA OS DADOS ANTES DE EXECUTAR UMA SÉRIE DE OPERAÇÕES.

QUANDO A BECKY TENTA EXECUTAR AS OPERAÇÕES DELA, ELA DEVE ESPERAR ATÉ QUE O ANDY TERMINE.



A BECKY PODE APENAS EXECUTAR SUAS OPERAÇÕES DEPOIS QUE O ANDY TIVER TERMINADO AS SUAS.

COMO RESULTADO, O BANCO DE DADOS APRESENTA O VALOR 50, COMO DEVERIA.

NOSSA! CAIN, ESTOU IMPRESSIONADA.

ISSO MESMO.

ENTÃO, ENTENDO QUE AS OPERAÇÕES NO BANCO DE DADOS SEJAM FINALIZADAS QUANDO CADA TRANSAÇÃO É PROCESSADA CORRETAMENTE.

BANCO DE DADOS



ESSA FINALIZAÇÃO É CHAMADA DE OPERAÇÃO DE EFETIVAÇÃO (COMMIT).



EMBORA UMA TRAVA TENHA SEU PAPEL EM UM BANCO DE DADOS, ESSA OPERAÇÃO NÃO DEVE SER USADA ABUSIVAMENTE. ELA PODE IR CONTRA O PROPÓSITO DE UM BANCO DE DADOS: COMPARTILHAR DADOS COM MUITAS PESSOAS.

ENTÃO UTILIZAMOS TIPOS DIFERENTES DE TRAVAS DEPENDENDO DA SITUAÇÃO.



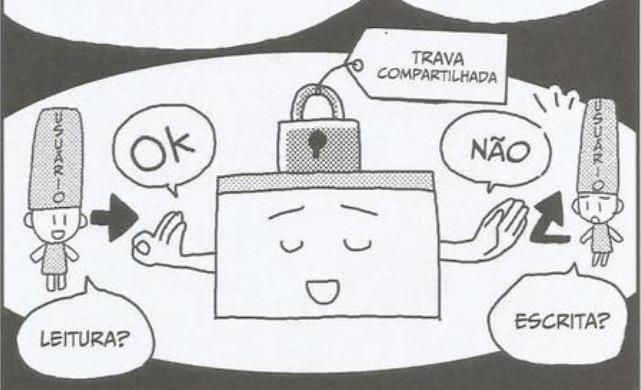
COMO FAZEMOS ISSO?

POR EXEMPLO, VOCÊ PODE USAR UMA TRAVA COMPARTILHADA PARA UMA OPERAÇÃO DE LEITURA QUANDO ELA FOR A ÚNICA OPERAÇÃO NECESSÁRIA.



ENQUANTO UMA TRAVA COMPARTILHADA ESTIVER APlicada, OUTROS USUÁRIOS PODEM LER OS DADOS...

MAS NÃO PODEM EXECUTAR UMA OPERAÇÃO DE ESCRITA NELES.



QUANDO EXECUTA UMA OPERAÇÃO DE ESCRITA, UM USUÁRIO APlica UMA TRAVA EXCLUSIVA.

QUANDO UMA TRAVA EXCLUSIVA É APLICADA, OUTROS USUÁRIOS NÃO PODEM LER OU ESCREVER DADOS.



TRAVA EXCLUSIVA



TRAVA EXCLUSIVA

NAO!!

LEITURA?

NAO!!

ESCRITA?

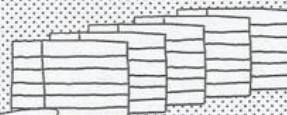
VEJO QUE HÁ TIPOS DIFERENTES DE TRAVAS.

ISSO FAZ SENTIDO.

QUANDO UMA TRAVA É USADA PARA CONTROLAR DUAS OU MAIS TRANSAÇÕES, ISSO É CHAMADO DE CONTROLE DE SIMULTANEIDADE.

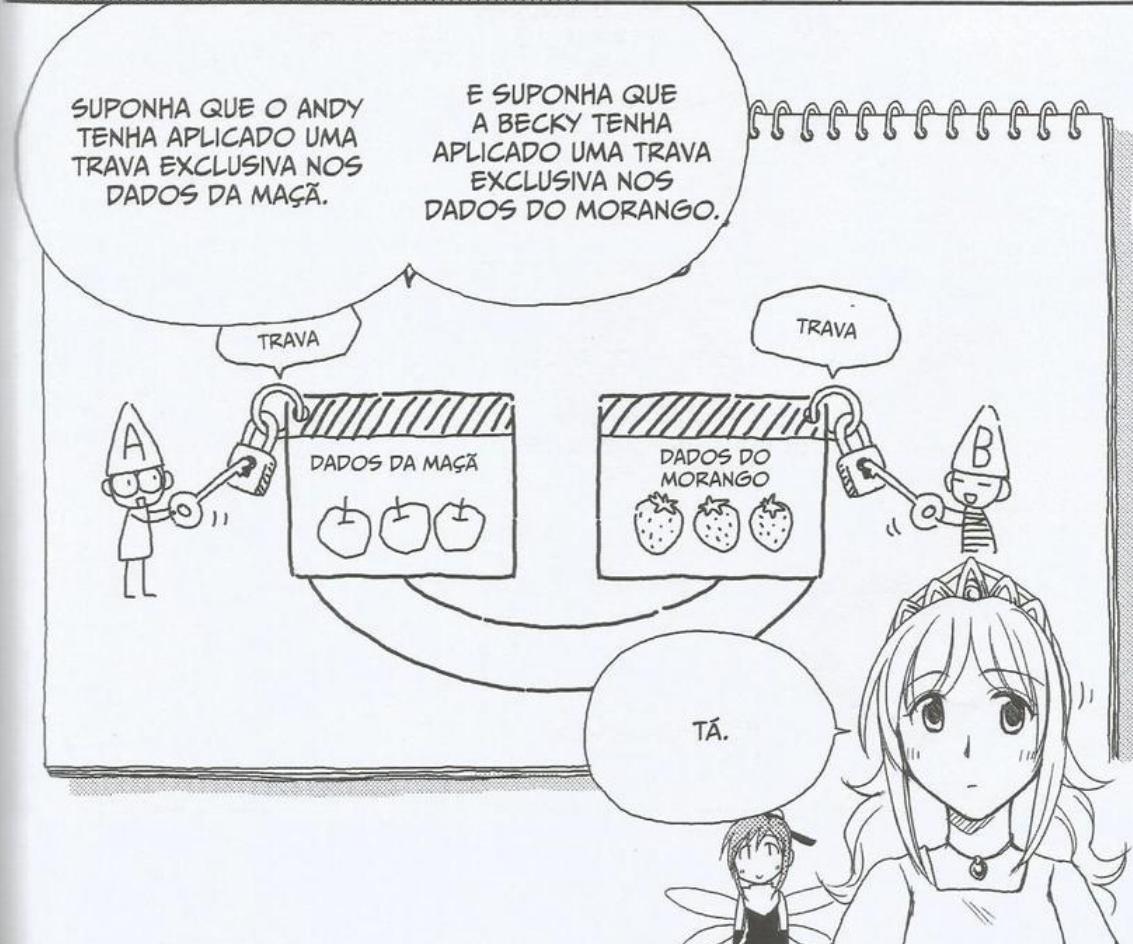
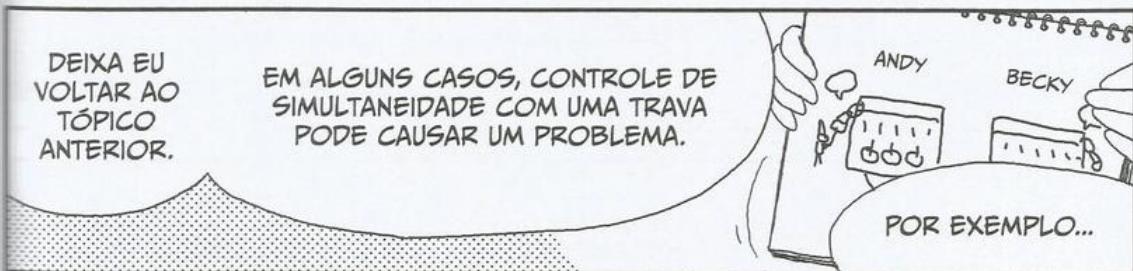
EM UM BANCO DE DADOS, CONTROLE DE SIMULTANEIDADE PERMITE

O MAIOR NÚMERO POSSÍVEL DE USUÁRIOS USANDO UM BANCO DE DADOS DE UMA VEZ ENQUANTO IMPIDE QUE CONFLITOS DE DADOS OCORRAM.



ALIÁS, CAIN!

SIM, TICA?



A SEGUIR, O ANDY PODE TENTAR APLICAR UMA TRAVA EXCLUSIVA NOS DADOS DO MORANGO,

E A BECKY PODE TENTAR APLICAR UMA TRAVA EXCLUSIVA NOS DADOS DA MAÇÃ.

O QUE ACONTECERIA ENTÃO?



JÁ QUE CADA UM DELES DEVE ESPERAR A TRAVA APLICADA PELO OUTRO USUÁRIO SER LIBERADA,

NENHUM DOS DOIS PODE PROCEDER COM QUALQUER OPERAÇÃO. É ISSO?

DEIXA EU PENSAR...

EXATAMENTE!

ELES NÃO PODEM FAZER NADA.

ESSA SITUAÇÃO, QUE É CHAMADA DE DEADLOCK, NÃO PODE SER RESOLVIDA A NÃO SER QUE UMA DAS TRAVAS SEJA LIBERADA.

POR EXEMPLO, VOCÊ PODE OBSERVAR TRANSAÇÕES QUE FORAM ENFILEIRADAS POR UM CERTO TEMPO...



E CANCELÁ-LAS!

QUANDO VOCÊ CANCELA UMA TRANSAÇÃO, ISSO É CHAMADO DE REVERSÃO (ROLLBACK).

REVERSÃO?

VOCÊ QUER DIZER
QUE PODE CANCELAR
TODAS AS OPERAÇÕES
EM UMA TRANSAÇÃO
DE UMA VEZ?

SIM! POR EXEMPLO, SE UMA
TRANSAÇÃO PARA "FRUTAS COM
DESCONTO COM PREÇOS MENORES
OU IGUAIS A 150 G" FALHOU,

OPERAÇÕES PARA MAÇÃS E
MORANGOS DEVEM SER TODAS
CANCELADAS, CERTO?

ENTENDI.

ENTÃO O BANCO
DE DADOS SE
COMPORTA COMO
SE A OPERAÇÃO
NUNCA TIVESSE SIDO
EXECUTADA?

É, MAIS OU
MENOS ISSO.

SE ALGO QUE
IMPOSSIBILITE A
FINALIZAÇÃO OCORREU
DURANTE UMA
TRANSAÇÃO,

ENTÃO UMA
REVERSÃO É
EXECUTADA EM VEZ
DE UMA OPERAÇÃO
DE EFETIVAÇÃO.



ISSO MESMO. UMA
TRANSAÇÃO SEMPRE
TERMINA COM UMA
OPERAÇÃO DE EFETIVAÇÃO
(COMMIT) OU DE
REVERSÃO (ROLLBACK).

NÃO HÁ MEIAS MEDIDAS, EM
OUTRAS PALAVRAS.

TUDO BEM!
AGORA, TICA,
O PRÓXIMO
TÓPICO É...

SEGURANÇA DE BANCOS DE DADOS



NOSSA TABELA
DE PRODUTOS.



CÓDIGO PRODUTO	NOME PRODUTO	PREÇO UNITÁRIO
101	MELÃO	10000G
102	MORANGO	12500G
103	MAÇÃ	8000G
104	LIMÃO	6000G
201	PINHÃO	9000G
202	CAQUI	12400G
301	PESSEGO	5000G
302	KIWI	6000G

O QUE TEM DE
ERRADO COM
ELA?



OS PREÇOS.
OS PREÇOS!



?, OS PREÇOS?



AH, NÃO!!!



OS VALORES
NA COLUNA DE
PREÇO UNITÁRIO
ESTÃO TODOS
BAGUNÇADOS!

ISSO É
UM MELÃO
QUE
CARO!!

COMO PODE?!



COM TODA ESSA CONFUSÃO
EM SUAS NOTAS FISCAIS,
MEU PAÍS, UM IMPORTADOR,
ESTÁ UM CAOS.

UM BANCO DE DADOS
É UMA COISA MUITO
PERIGOSA.

COMO PARTE DA
COMPENSAÇÃO POR
ESSE PROBLEMA...

ALGUÉM COM INTENÇÃO
MALDOSA DEVE TER
EXECUTADO UMA
SOBREPOSIÇÃO DE DADOS
NÃO-AUTORIZADA.

QUE
HORROR!

VOCÊS
ESTÃO SE
COMPORTANDO
COMO SE EU
NÃO ESTIVESSE
AQUI.

PRÍNCIPE
RAMINESS,

PEDIMOS
DESCULPA.

POR QUE VOCÊ NÃO ACEITA
MEU PEDIDO, PRINCESA
RURUNA? VENHA AO MEU PAÍS
E SEJA MINHA NOIVA.

IGNORANDO



PROMETEMOS TOMAR MEDIDAS PARA PROTEGER O BANCO DE DADOS E IMPEDIR QUE ESSE TIPO DE COISA ACONTEÇA DE NOVO.

PERDOE-NOS POR ISSO, POR FAVOR.

VOCÊ DIZ QUE VAI CONSERTAR, MAS... NÃO ESTOU TÃO CERTO... VOCÊ PODE SER MAIS ESPECÍFICO?

A CAUSA DESSE PROBLEMA É QUE QUALQUER UM NO REINO DE KOD TEM LIVRE ACESSO AO BANCO DE DADOS.

EM PRIMEIRO LUGAR, TEMOS QUE CRIAR UM CONTROLE DE ACESSO PARA LIMITAR OS USUÁRIOS DO BANCO DE DADOS.

E ISSO QUER DIZER...?

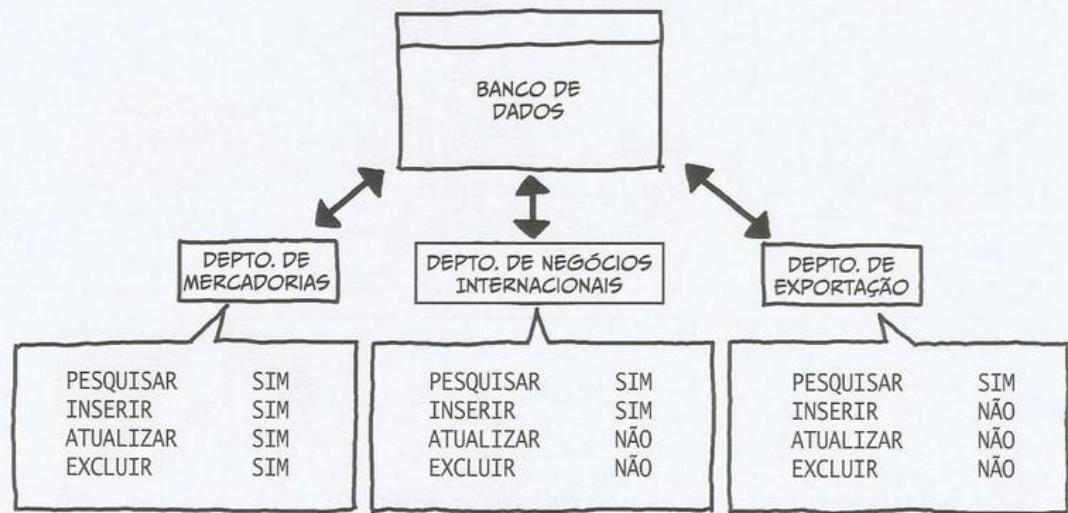
UMA BOA SOLUÇÃO PODE SER EXIGIR NOMES DE USUÁRIOS E SENHAS PARA ACESSAR O BANCO DE DADOS, PARA ASSEGURAR QUE CADA USUÁRIO É DIGNO DE CONFIANÇA PARA TER DIREITO DE ACESSO.

PARECE UMA SOLUÇÃO INTELIGENTE!

SEGUNDO, IREMOS CONFIGURAR OPÇÕES PARA DAR PERMISSÃO A CERTAS OPERAÇÕES APENAS PARA USUÁRIOS AUTORIZADOS.

B
L
A
M!
• PERMISSÃO PARA PESQUISAR (SELECT),
INSERIR, ATUALIZAR, E EXCLUIR DADOS DO
PRODUTO.
• PERMISSÃO PARA PESQUISAR E INSERIR
DADOS DO PRODUTO, SEM PERMISSÃO
PARA ATUALIZAR/EXCLUIR
• PERMISSÃO PARA BUSCAR PRODUTO...

- O PESSOAL DO DEPARTAMENTO DE MERCADORIAS PODE PESQUISAR, INSERIR, ATUALIZAR E EXCLUIR DADOS DO PRODUTO.
- O PESSOAL DO DEPARTAMENTO DE NEGÓCIOS INTERNACIONAIS PODE PESQUISAR E INSERIR DADOS DO PRODUTO, MAS NÃO PODE ATUALIZÁ-LOS OU EXCLUI-LOS.
- O PESSOAL DO DEPARTAMENTO DE EXPORTAÇÃO PODE PESQUISAR DADOS DO PRODUTO, MAS NÃO PODE INSERIR, ATUALIZAR OU EXCLUIR.



TCHARAM!



ACELERANDO AS COISAS COM INDEXAÇÃO



SUPONHA QUE VAMOS PROCURAR O SIGNIFICADO DO TERMO TRANSAÇÃO CONSULTANDO ESTE LIVRO SOBRE BANCOS DE DADOS.

UMA PESQUISA ÀS CEGAS PELO TERMO PODERIA SER TRABALHOSA, ENTÃO CHECAMOS O ÍNDICE EM VEZ DISSO.



NO TRECHO PARA TRANSAÇÃO, AS PÁGINAS QUE FALAM SOBRE ELA ESTÃO LISTADAS.



USANDO UM ÍNDICE, PODEMOS RAPIDAMENTE ENCONTRAR A PÁGINA QUE PROCURAMOS.

ISSO MESMO.

E É A MESMA COISA PARA UM ÍNDICE DE BANCO DE DADOS. POR EXEMPLO...



SE VOCÊ CRIAR ÍNDICES
PARA CÓDIGOS DO
PRODUTO

PODE INSTANTANEAMENTE
DESCOBRIR ONDE OS
DADOS DO PRODUTO
ESTÃO ARMAZENADOS
PARA UM PRODUTO
COM CÓDIGO 101, POR
EXEMPLO.

ELE INFORMA EM QUE
LUGAR NO DISCO OS
DADOS DO PRODUTO ESTÃO
LOCALIZADOS.



INDEXAÇÃO AJUDA A
ACELERAR A PESQUISA.

CERTO.

BOM, NÃO ACHO
TÃO FÁCIL DE
ACOMPANHAR...
MAS...

EXAMINAR TODAS AS LINHAS
QUANDO PROCURAMOS
CERTO TIPO DE DADO
CONSOME MUITO TEMPO.

DEVO
REPETIR
MINHA
EXPLICAÇÃO
DESPDE O
INÍCIO? ♪
NÃO,
NÃO...

USANDO ÍNDICES,
PODEMOS REDUZIR A
TAXA DE ACESSO AO
DISCO.

REDUZA A TAXA DE
ACESSO A DISCO, E
NOSSA BUSCA SERÁ
MUITO MAIS VELOZ!

O QUE
É ISSO?

GLUP!

EI! COM
QUEM
VOCÊ ESTÁ
FALANDO?

TEM ALGUÉM
AÍ?

PRINCESA,
TOME
CUIDADO!

XI!





RECUPERAÇÃO DE DESASTRES

DENTRO DE UM BANCO DE DADOS, REGISTROS CHAMADOS DE LOGS SÃO CRIADOS SEMPRE QUE UMA OPERAÇÃO É EXECUTADA, NÃO?



EXATO.



É ASSIM QUE AS MUDANÇAS NO CONTEÚDO DO BANCO DE DADOS SÃO REGISTRADAS.

LOGS, É...?

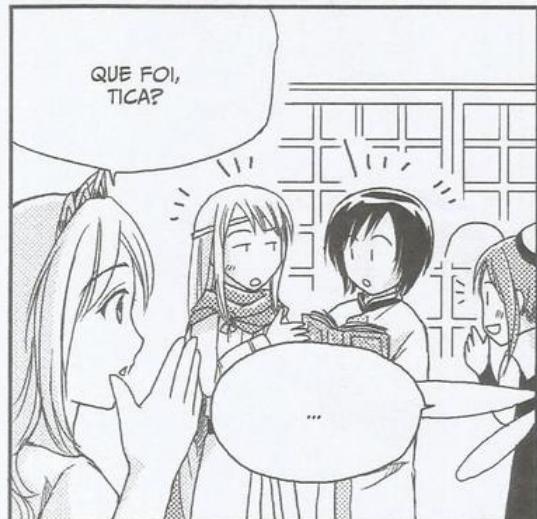


O MAIS IMPORTANTE SÃO REGISTROS DE VALORES ANTES E DEPOIS DE UMA ATUALIZAÇÃO NO BANCO DE DADOS.

HUMM...

QUE FOI,
TICA?

EI, ELE TAMBÉM
ESTÁ SE
INTERESSANDO POR
BANCOS DE DADOS.



VOCÊ QUER DIZER
O RAMINESS?
TEM CERTEZA?

QUANDO UM PROBLEMA OCORRER, PRIMEIRO SE REINICIA O SISTEMA.

O MÉTODO DE RECUPERAÇÃO VARIA DEPENDENDO DE SE A TRANSAÇÃO FOI EFETIVADA OU NÃO.

DEPOIS UTILIZAM-SE OS LOGS PARA RECUPERAR O BANCO DE DADOS.

TRANS -
O QUÉ?

APARENTEMENTE, ELE NÃO ESTÁ ENTENDENDO NADA... É UMA PENA.

SIM.

VOU ILUSTRAR...

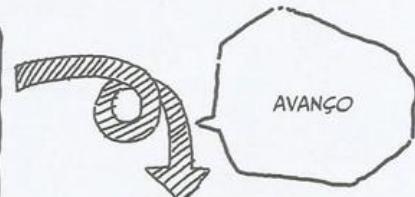
NÃO É NECESSÁRIO...

RISC,
RISC

SE O PROBLEMA OCORREU DEPOIS QUE UMA TRANSAÇÃO JÁ TINHA SIDO EFETIVADA, ISSO QUER DIZER QUE AS OPERAÇÕES JÁ ESTAVAM FINALIZADAS PARA AQUELA TRANSAÇÃO.

ENTÃO, PODE-SE RECUPERAR OS DADOS REAPLICANDO-SE AS OPERAÇÕES NO BANCO DE DADOS.

NESSE TIPO DE RECUPERAÇÃO, O BANCO DE DADOS CONFIRMA OS VALORES POSTERIORES À ATUALIZAÇÃO.



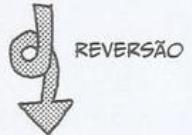
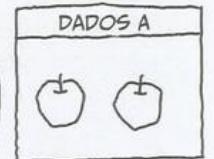
ESSE MÉTODO DE RECUPERAÇÃO É CHAMADO DE AVANÇO (ROLLING FORWARD).

O QUE ACONTECE
SE A TRANSAÇÃO
NÃO TIVER SIDO
EFETIVADA AINDA
QUANDO O
PROBLEMA OCORRE?



NUMA OPERAÇÃO
DE REVERSÃO (OU
ROLLBACK), O VALOR
ANTES DA ATUALIZAÇÃO
É CONFIRMADO,
PARA CANCELAR A
TRANSAÇÃO.

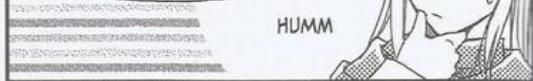
EM OUTRAS
PALAVRAS, O
ESTADO DO BANCO
DE DADOS ANTES
QUE A TRANSAÇÃO
TENHA INICIADO É
RESTAURADO.



O SISTEMA RECUPERA OS
DADOS E GARANTE QUE
ELES ESTEJAM LIVRES DE
INCONSISTÊNCIAS.



NÃO ESTOU
FAMILIARIZADO COM
TERMOS COMO
EFETIVAÇÃO (COMMIT) E
TRANSAÇÃO.

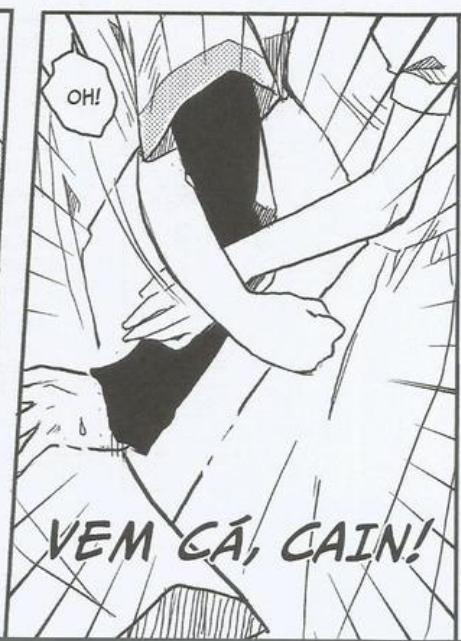


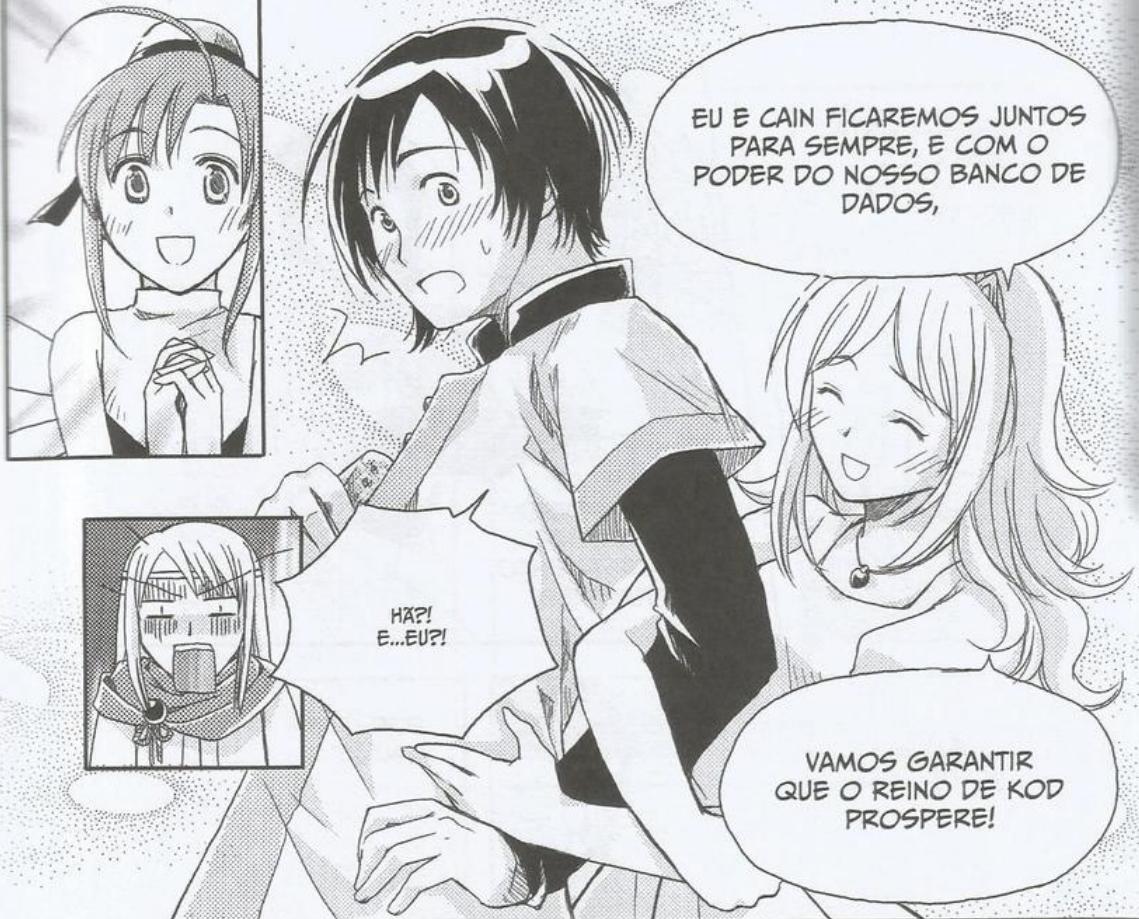
NO ENTANTO, PARECE
QUE AS MEDIDAS DE
SEGURANÇA DO SEU
BANCO DE DADOS
SÃO BOAS.



SABE, BANCOS
DE DADOS SÃO
ROBUSTOS! MESMO
QUANDO DESASTRES
OCORREM!



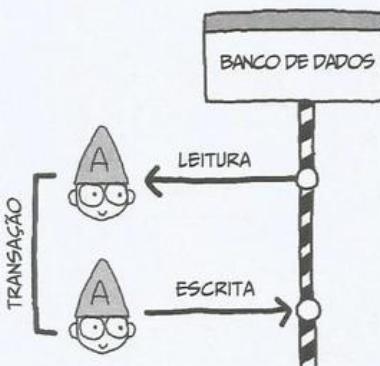




PROPRIEDADES DAS TRANSAÇÕES



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.

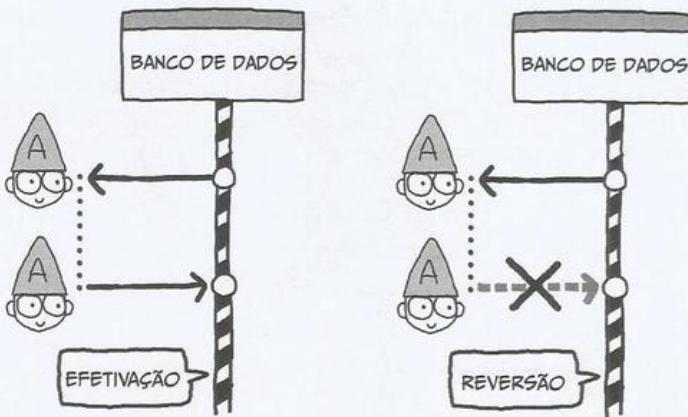
PROPRIEDADES OBRIGATÓRIAS PARA UMA TRANSAÇÃO

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 quanto 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.

`COMMIT;`

Use este comando para efetivar uma transação

`ROLLBACK;`

Use este comando para reverter uma transação.



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.

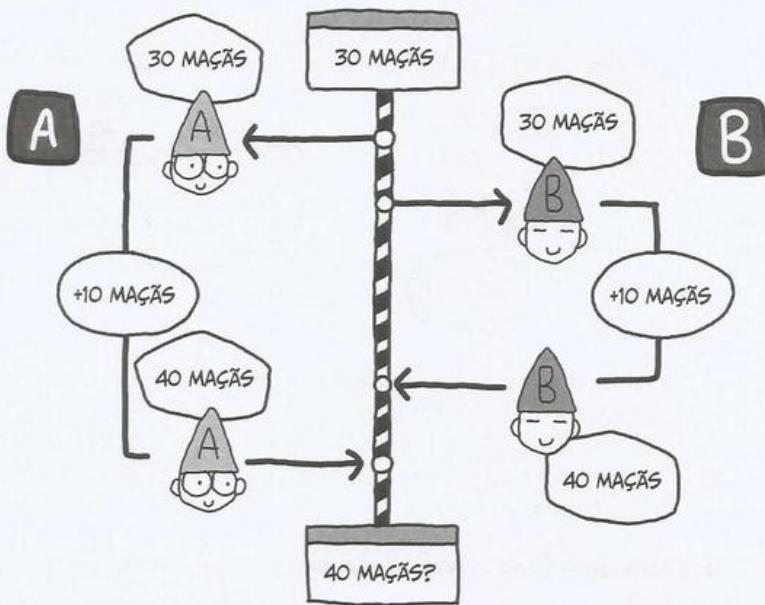
Q2

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

CONSISTÊNCIA

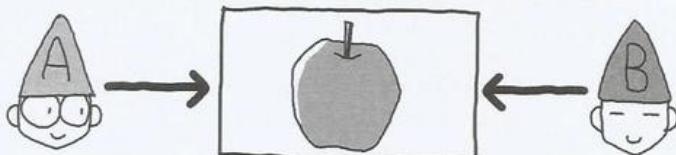
Uma transação não deve gerar erros. Se o banco de dados estava consistente antes do processamento de uma transação, ele precisa continuar consistente depois que a transação se encerra.

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?

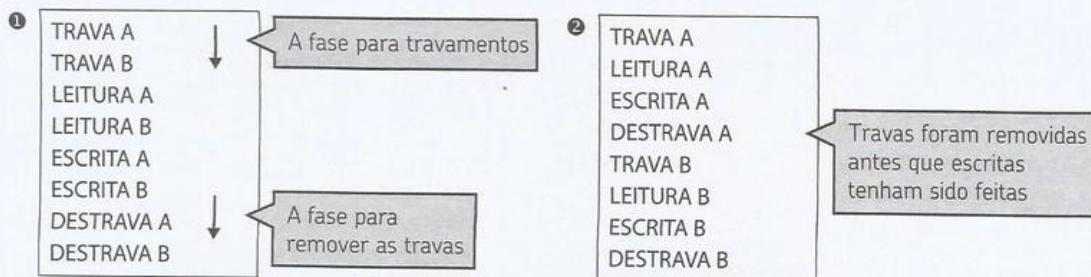
Q6

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

TRAVAMENTO EM DUAS FASES

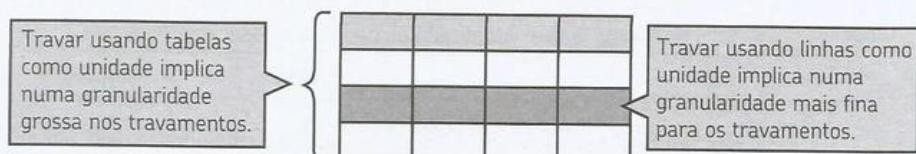
Para ter certeza de que as transações são serializáveis, precisamos obedecer certas regras para aplicar e remover travas. Uma dessas regras é o *travamento em duas fases* - para cada transação, duas fases devem ser implementadas: uma para aplicar travas, e outra para removê-las.

Por exemplo, suponha que existam os recursos A e B, ambos sujeitos a travamento. A transação ① observa a regra do travamento em duas fases, mas a ② 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 necessário 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.

NIVEIS 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;
```

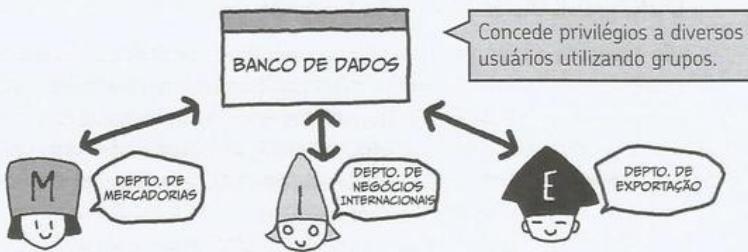
O usuário que recebe os privilégios pode concedê-los a outros.

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

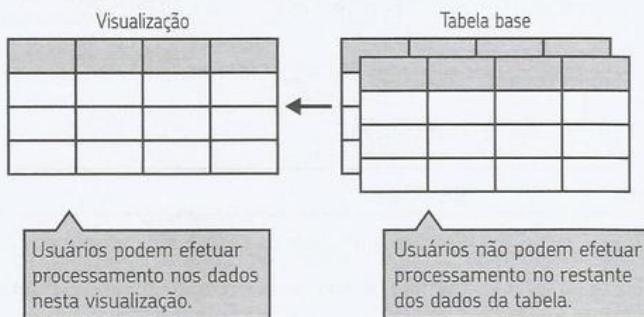
```
REVOKE SELECT, UPDATE ON produtos  
FROM Departamento_Negocios_Internacionais;
```

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				

QUANDO UM DESASTRE OCORRE



Um banco de dados precisa ter um mecanismo que possa proteger dados no sistema em caso de falha. Para garantir a durabilidade das transações, é obrigatório que nenhuma falha possa criar dados incorretos. Para proteger a si mesmo de falhas, um banco de dados executa várias operações, o que inclui a criação de cópias de segurança e logs de transação.

TIPOS DE FALHAS

Uma falha no banco de dados pode ocorrer sob várias circunstâncias. Tipos possíveis de falhas incluem os seguintes:

- Falhas na transação
- Falhas no sistema
- Falhas de mídia

Falha de transação ocorre quando uma transação não pode ser concluída devido a um erro nela própria. Ela é revertida quando a falha ocorre.

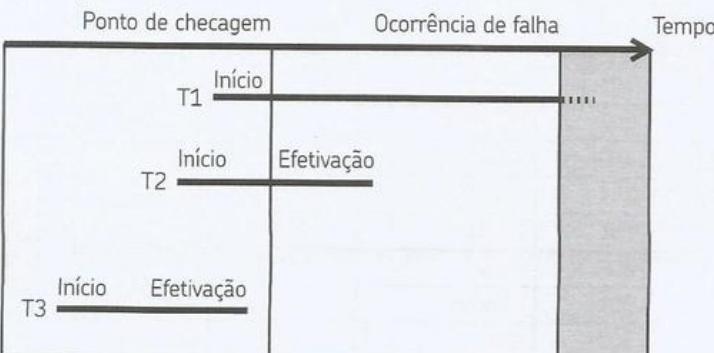
Falha de sistema ocorre quando o sistema cai devido à queda de energia ou outro problema do tipo. No caso de uma falha de sistema, a recuperação de desastre acontece após a reinicialização do sistema. Geralmente, transações que ainda não foram efetivadas quando acontece a falha são revertidas, e aquelas que já foram são avançadas (reaplicadas).

Falha de mídia ocorre quando o disco rígido que contém o banco de dados é danificado. No caso de uma falha dessas, a recuperação de desastre é feita com a utilização de cópias de backup. As transações efetuadas após a criação dos arquivos de backup são avançadas.

PONTOS DE CHECAGEM (CHECKPOINTS)

Para melhorar a eficiência das operações de escrita em um banco de dados, um buffer (um segmento de memória usado para armazenar dados temporariamente) é usado com frequência para gravar dados a curto prazo. Os conteúdos do buffer e o banco de dados são sincronizados, e então um ponto de checagem é criado. Quando o banco de dados escreve um ponto de checagem, ela não tem que executar qualquer recuperação contra falhas para transações que foram efetuadas antes dele. As transações que não foram efetivadas antes do ponto de checagem devem ser recuperadas.

Agora, suponha que as transações mostradas abaixo estejam sendo executadas no momento em que ocorra uma falha de sistema. Quais transações devem ser revertidas? E quais devem ser reaplicadas?





QUESTÕES

Tente responder às questões seguintes na tabela da página anterior. As respostas estão na página 168.

Q12

Como T1 deve ser processado?

Q13

Como T2 deve ser processado?

Q14

Como T3 deve ser processado?

Nos casos de falha no banco de dados, os mecanismos de recuperação descritos acima protegerão o banco contra inconsistências. É por isso que você pode estar seguro a respeito da integridade do banco quando a utiliza.

ÍNDICES

Um banco de dados gerencia grandes quantidades de dados, de modo que buscar por valores específicos pode tomar muito tempo. Mas você pode usar índices para acelerar as buscas!

Cód. do produto	Nome do produto	Preço unitário	Distrito
101	Melão	800 G	Mar do Sul
102	Morango	150 G	Central
103	Maçã	120 G	Mar do Norte
104	Limão	200 G	Mar do Sul
201	Pinhão	100 G	Mar do Norte
202	Caqui	160 G	Central
301	Pêssego	130 G	Mar do Sul
302	Kiwi	200 G	Mar do Sul

Leva muito tempo para procurar por cada item linha por linha.

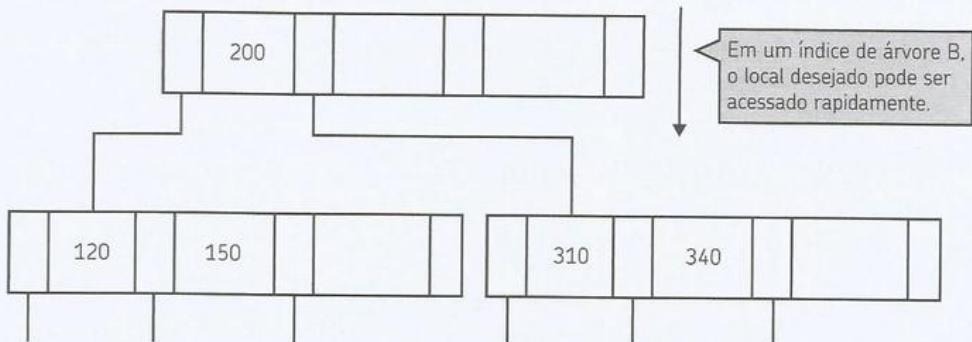
Um índice é uma ferramenta que permite acessar rapidamente o local do dado procurado. Quando procura-se por algum dado em um grande banco de dados, a busca com índices promete resultados rápidos.

Cód. do produto	Nome do produto	Preço unitário	Distrito
101	Melão	800 G	Mar do Sul
102	Morango	150 G	Central
103	Maçã	120 G	Mar do Norte
104	Limão	200 G	Mar do Sul
201	Pinhão	100 G	Mar do Norte
202	Caqui	160 G	Central
301	Pêssego	130 G	Mar do Sul
302	Kiwi	200 G	Mar do Sul

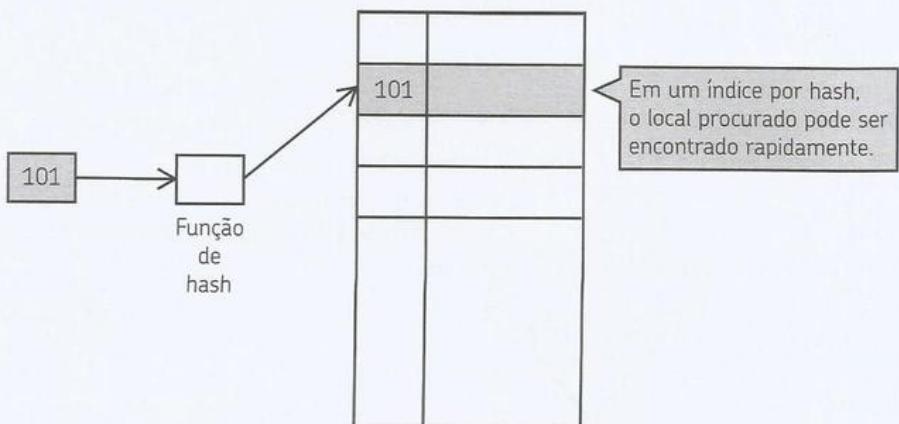
A localização do dado procurado pode ser acessada rapidamente usando seu índice.

Índice

Métodos de indexação incluem árvore binária (B-tree) e métodos de hash. Um índice de árvore-B é composto de nós ancestrais e filhos, que podem ter mais nós filhos. Os nós são dispostos de forma ordenada. Cada ancestral contém informações sobre os valores mínimo e máximo contidos em todos os seus nós filhos. Isso permite que o banco de dados navegue rapidamente até o local desejado, pulando seções inteiras da árvore que não podem conter o valor que se busca.



O método de índice por *hash* encontra o local dos dados procurados aplicando uma função de hash ao valor chave dos dados. O hash age como uma impressão digital única para um valor. O método de índice por hash pode executar consultas específicas de correspondência total (full-match searches), tal como uma busca pelo código do produto 101. No entanto, ele não serve para buscar com eficiência por condições comparativas como códigos do produto maiores que 101 ou por referências parciais como produtos cujos nomes terminam em ão.



Em alguns casos, usar um índice pode não acelerar a busca – usar um índice não poupa tempo a menos que você esteja procurando apenas por uma pequena parte dos dados. Além disso, há casos onde os índices são recriados sempre que os dados são atualizados, o que resulta em um processamento mais lento de uma operação de atualização.



QUESTÕES

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

Q15

Qual índice seria mais produtivo em uma busca com um sinal de igual, um de árvore-B ou um de hash?

Q16

Qual índice seria mais produtivo em uma busca com um sinal de diferente, um de árvore-B ou um de hash?

OTIMIZAÇÃO DE UMA CONSULTA

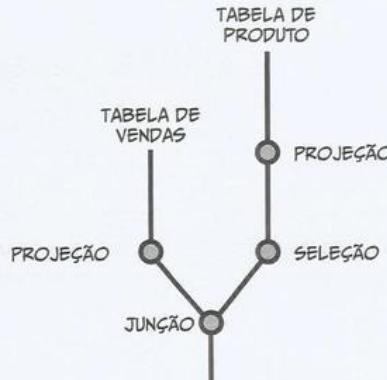
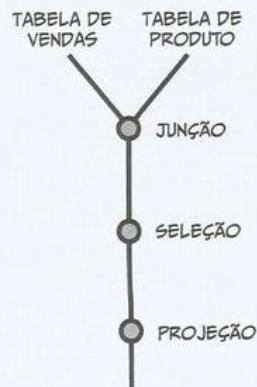
Quando você consulta um banco de dados, ele analisa a consulta SQL e considera se deve utilizar um índice para que possa processar a consulta mais rapidamente. Vamos examinar o procedimento para processar uma consulta.

O banco de dados pode decidir sobre uma ordem ideal para processar uma consulta. A maioria das consultas pode ser realizada em diversas ordens com os mesmos resultados, mas com velocidades diferentes possíveis. Por exemplo, suponha que haja uma consulta para extrair datas de venda e nomes para produtos com um preço unitário maior que 200G. Essa consulta pode ser dividida nos seguintes passos.

```
SELECT data, nome_produto  
FROM produto, vendas  
WHERE preco_unitario>=200  
AND produtos.codigo_produto = vendas.codigo_produto;
```

1. Junte as Tabelas de Produto e de Vendas
2. Selecione os produtos cujos preços unitários sejam maiores que 200G.
3. Extraia as colunas de datas e de nomes do produto.

Por exemplo, a figura à esquerda mostra a consulta processada na ordem de 1 a 3. A figura à direita mostra a consulta processada na ordem de 3 a 1. De qualquer forma, as consultas são equivalentes.



No entanto, quando processada de 1 a 3, a mesma consulta geralmente exigiria um tempo de processamento maior, porque quando a primeira junção é feita, uma tabela intermediária com muitas linhas pode ser criada. Por outro lado, o procedimento de 3 para 1 exige um tempo de processamento menor, já que a seleção e a projeção acontecem primeiro, eliminando dados indesejados o mais rapidamente possível. Assim, a mesma consulta pode exigir um tempo de processamento diferente, dependendo da ordem na qual a projeção, seleção e junção são executadas.

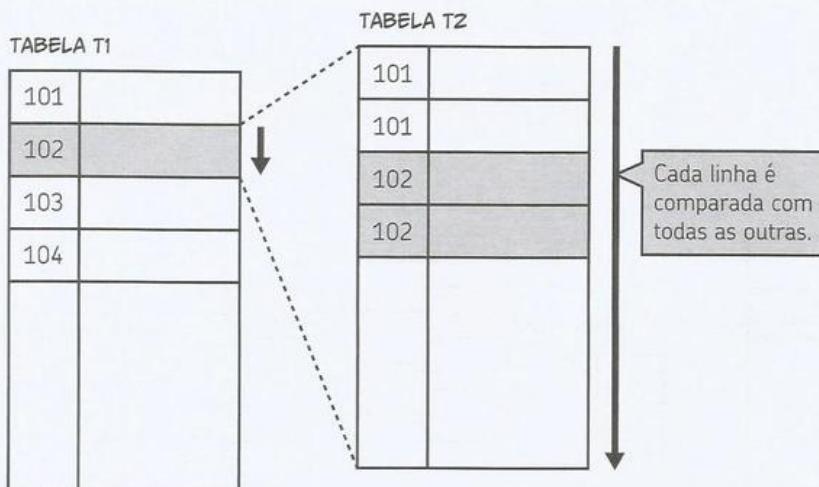
Geralmente, o banco de dados deve usar as seguintes regras para encontrar a melhor ordem para consultas:

- Executar seleção primeiro para reduzir o número de linhas.
- Executar projeção primeiro para reduzir o número de colunas irrelevantes ao resultado.
- Executar junção (*join*) mais tarde.

Há diferentes técnicas para executar projeção, seleção e junção, respectivamente. Para seleção, pode-se usar tanto uma busca de correspondência total quanto uma baseada em índice. Para a junção, os métodos a seguir estão disponíveis.

LAÇOS ANINHADOS

O método de laços aninhados compara uma linha em uma tabela a várias linhas em outra (veja a figura abaixo). Por exemplo, um dos valores em uma linha na Tabela T1 é usado para encontrar linhas correspondentes na Tabela T2. Se os valores forem os mesmos, então uma linha de junção é criada.



FUSÃO ORDENADA (SORT MERGE)

Esse método ordena (sort) e então funde (merge) linhas em várias tabelas (veja a figura abaixo). Primeiro, as Tabelas T1 e T2 são ordenadas parcial ou inteiramente. Então são comparadas começando pela linha de cima, e uma linha de junção é criada sempre que o mesmo valor é encontrado. Já que as linhas já foram ordenadas, o processamento só precisa ser feito em uma direção, de modo que isso levará menos tempo. Você deve ter em mente, no entanto, o tempo necessário para a ordenação inicial.

TABELA T1

101	
102	
103	
104	

TABELA T2

101	
101	
102	
102	

Primeiro ordena,
depois compara.

HASH

Um hash divide uma das tabelas usando uma função hash e então a funde com uma linha em outra tabela que tenha o mesmo valor. Esse método seleciona efetivamente a linha para a junção.

TABELA T1

Função
Hash



TABELA T2

Função
Hash

OTIMIZADOR

Quando uma consulta é processada, decide-se entre essas técnicas diferentes para obter um ótimo desempenho. Em um banco de dados, a função responsável pela otimização de consultas é chamada de otimizador. Há dois tipos comuns.

PROCESSAMENTO COM BASE EM REGRAS

Certas regras são estabelecidas antes que qualquer operação seja executada. Por exemplo, algumas operações podem ser combinadas ou reordenadas quase da mesma forma que uma equação algébrica pode ser manipulada e ainda querer dizer a mesma coisa. O otimizador tenta encontrar a forma mais eficiente de processar a consulta que forneça os mesmos resultados.

PROCESSAMENTO COM BASE EM CUSTO

Esse método tenta estimar o custo do processamento da consulta, baseado em estatísticas que o banco de dados mantém. Esse tipo de processamento às vezes é mais flexível do que o baseado em regras, mas exige atualizações periódicas das estatísticas do banco de dados. Gerenciar e analisar essas estatísticas leva muito tempo.

RESUMO



- Você pode estabelecer privilégios de usuário em um banco de dados.
- O uso de travas (bloqueios) garante consistência quando um banco de dados tem muitos usuários.
- A indexação permite buscas rápidas.
- Um banco de dados contém funções para recuperação contra desastre.

RESPOSTAS

Q1

COMMIT;

Q2

ROLLBACK;

Q3 Sim

Q4 Não

Q5 Não

Q6 Não

Q7 Vai aumentar

Q8 Vai diminuir

Q9

```
GRANT SELECT ON produtos TO Departamento_Exportacao;
```

Q10

```
REVOKE DELETE ON produtos FROM Departamento_Negocios_Internacionais;
```

Q11

	Pesquisar	Inserir	Atualizar	Excluir
Dept. de negócios internacionais	SIM	SIM	SIM	SIM
Dept. de mercadorias	NÃO	SIM	NÃO	SIM
Dept. de exportação	NÃO	NÃO	SIM	SIM

Q12

Uma reversão é executada, uma vez que não ocorreu efetivação no momento da ocorrência de falha.

Q13

Um avanço (reaplicação) é executado, uma vez que ocorreu efetivação no momento da ocorrência da falha.

Q14

Nenhuma operação de recuperação é necessária, uma vez que a efetivação já tinha ocorrido no momento do ponto de checagem.

Q15

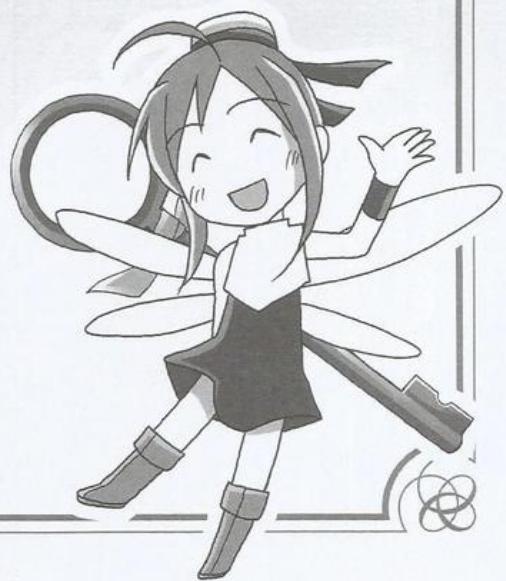
Hash

Q16

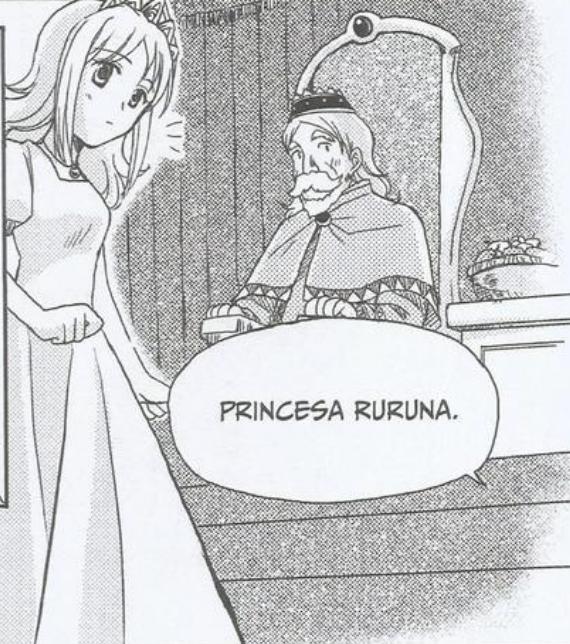
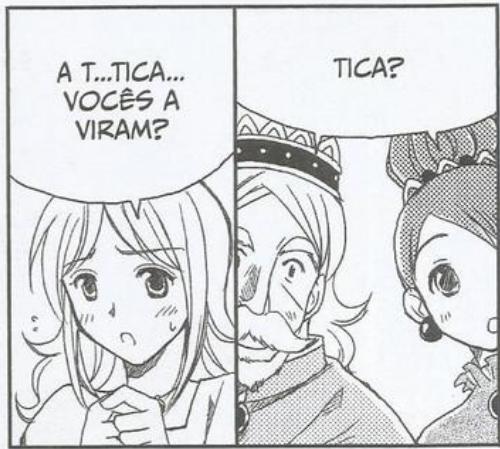
Árvore-B

6

OS BANCOS DE DADOS ESTÃO EM
TODOS OS LUGARES!





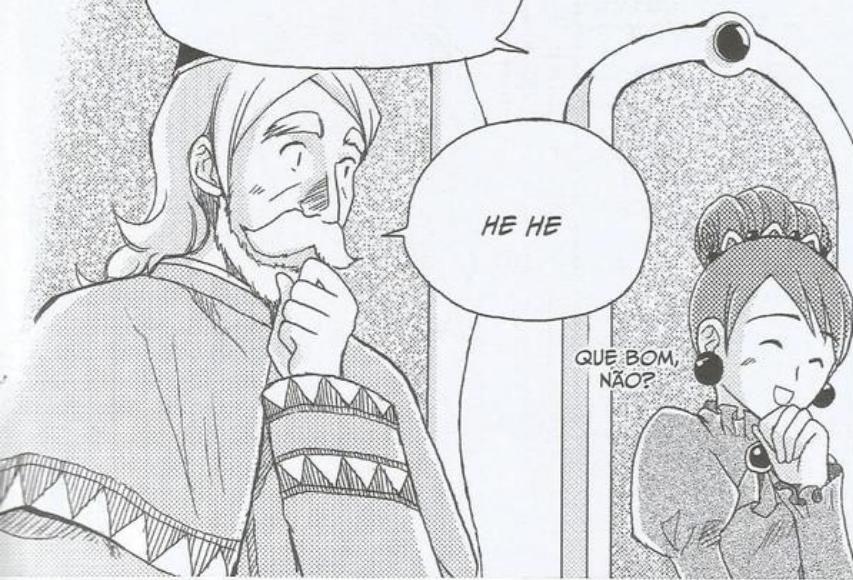




VOSSA ALTEZA,
ESSES DOIS ESTÃO,
REALMENTE...

SE DANDO BEM, NÃO?

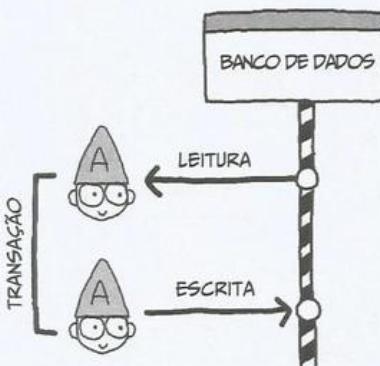
ELA SE FOI!



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.

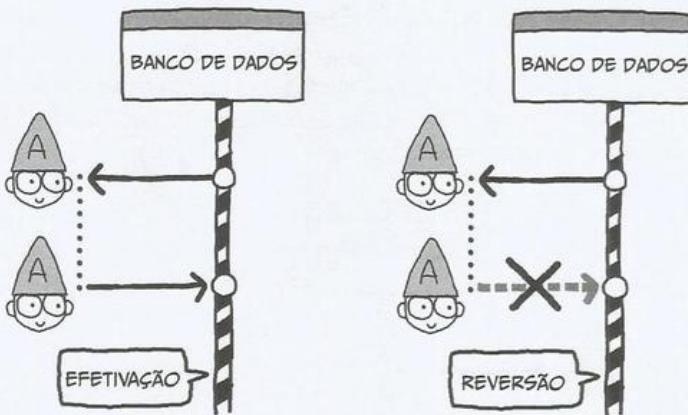
PROPRIEDADES OBRIGATÓRIAS PARA UMA TRANSAÇÃO

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 quanto 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.

`COMMIT;`

Use este comando para efetivar uma transação

`ROLLBACK;`

Use este comando para reverter uma transação.



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.

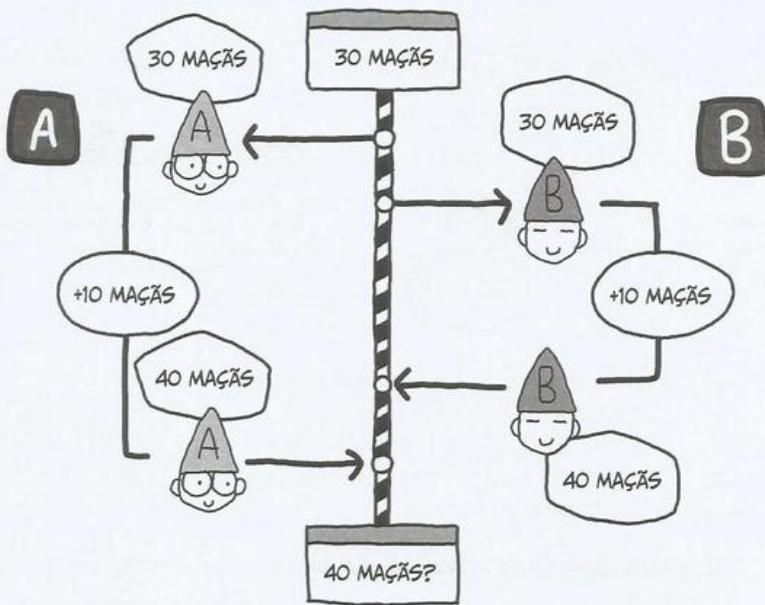
Q2

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

CONSISTÊNCIA

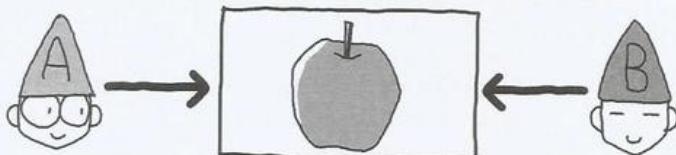
Uma transação não deve gerar erros. Se o banco de dados estava consistente antes do processamento de uma transação, ele precisa continuar consistente depois que a transação se encerra.

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?

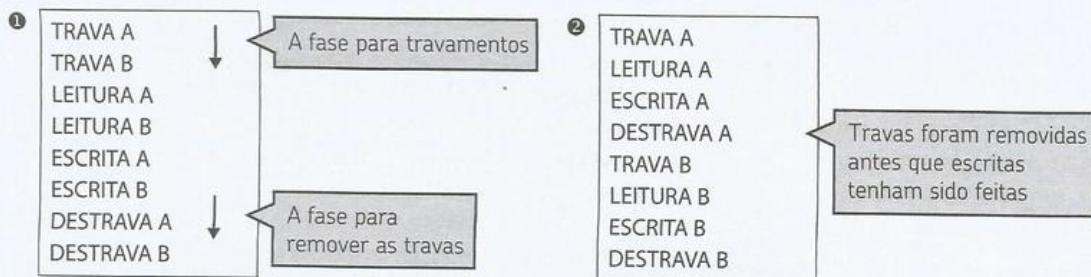
Q6

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

TRAVAMENTO EM DUAS FASES

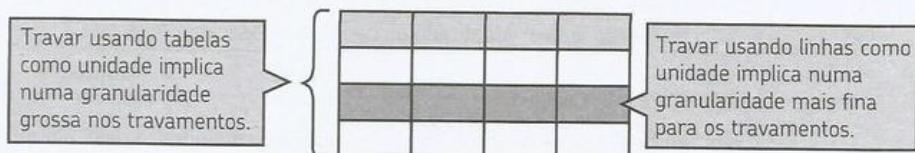
Para ter certeza de que as transações são serializáveis, precisamos obedecer certas regras para aplicar e remover travas. Uma dessas regras é o *travamento em duas fases* - para cada transação, duas fases devem ser implementadas: uma para aplicar travas, e outra para removê-las.

Por exemplo, suponha que existam os recursos A e B, ambos sujeitos a travamento. A transação ① observa a regra do travamento em duas fases, mas a ② 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 necessário 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.

NIVEIS 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;
```

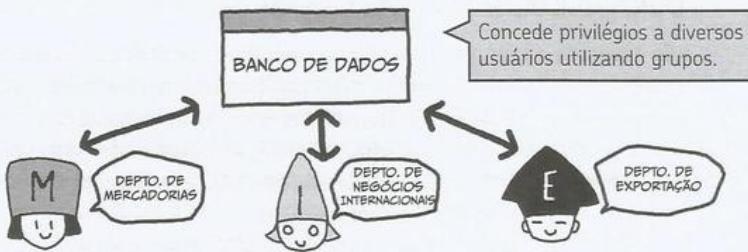
O usuário que recebe os privilégios pode concedê-los a outros.

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

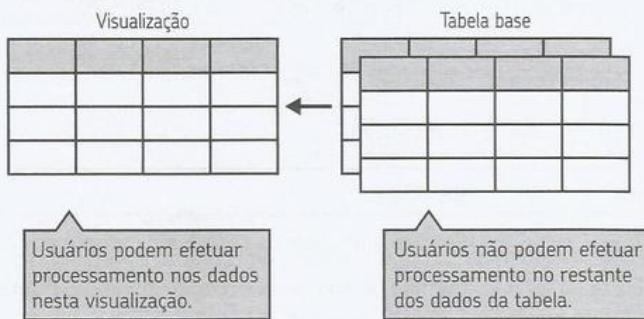
```
REVOKE SELECT, UPDATE ON produtos  
FROM Departamento_Negocios_Internacionais;
```

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				

QUANDO UM DESASTRE OCORRE



Um banco de dados precisa ter um mecanismo que possa proteger dados no sistema em caso de falha. Para garantir a durabilidade das transações, é obrigatório que nenhuma falha possa criar dados incorretos. Para proteger a si mesmo de falhas, um banco de dados executa várias operações, o que inclui a criação de cópias de segurança e logs de transação.

TIPOS DE FALHAS

Uma falha no banco de dados pode ocorrer sob várias circunstâncias. Tipos possíveis de falhas incluem os seguintes:

- Falhas na transação
- Falhas no sistema
- Falhas de mídia

Falha de transação ocorre quando uma transação não pode ser concluída devido a um erro nela própria. Ela é revertida quando a falha ocorre.

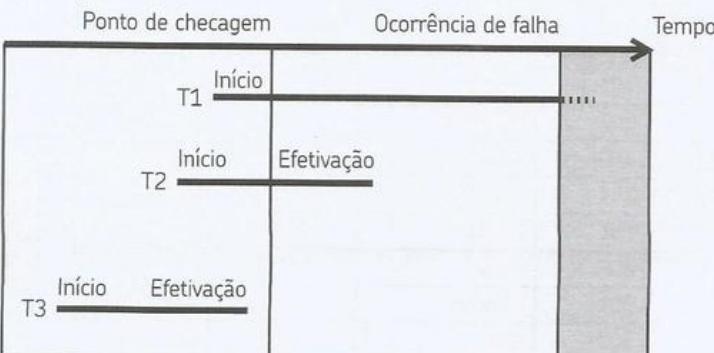
Falha de sistema ocorre quando o sistema cai devido à queda de energia ou outro problema do tipo. No caso de uma falha de sistema, a recuperação de desastre acontece após a reinicialização do sistema. Geralmente, transações que ainda não foram efetivadas quando acontece a falha são revertidas, e aquelas que já foram são avançadas (reaplicadas).

Falha de mídia ocorre quando o disco rígido que contém o banco de dados é danificado. No caso de uma falha dessas, a recuperação de desastre é feita com a utilização de cópias de backup. As transações efetuadas após a criação dos arquivos de backup são avançadas.

PONTOS DE CHECAGEM (CHECKPOINTS)

Para melhorar a eficiência das operações de escrita em um banco de dados, um buffer (um segmento de memória usado para armazenar dados temporariamente) é usado com frequência para gravar dados a curto prazo. Os conteúdos do buffer e o banco de dados são sincronizados, e então um ponto de checagem é criado. Quando o banco de dados escreve um ponto de checagem, ela não tem que executar qualquer recuperação contra falhas para transações que foram efetuadas antes dele. As transações que não foram efetivadas antes do ponto de checagem devem ser recuperadas.

Agora, suponha que as transações mostradas abaixo estejam sendo executadas no momento em que ocorra uma falha de sistema. Quais transações devem ser revertidas? E quais devem ser reaplicadas?





QUESTÕES

Tente responder às questões seguintes na tabela da página anterior. As respostas estão na página 168.

Q12

Como T1 deve ser processado?

Q13

Como T2 deve ser processado?

Q14

Como T3 deve ser processado?

Nos casos de falha no banco de dados, os mecanismos de recuperação descritos acima protegerão o banco contra inconsistências. É por isso que você pode estar seguro a respeito da integridade do banco quando a utiliza.

ÍNDICES

Um banco de dados gerencia grandes quantidades de dados, de modo que buscar por valores específicos pode tomar muito tempo. Mas você pode usar índices para acelerar as buscas!

Cód. do produto	Nome do produto	Preço unitário	Distrito
101	Melão	800 G	Mar do Sul
102	Morango	150 G	Central
103	Maçã	120 G	Mar do Norte
104	Limão	200 G	Mar do Sul
201	Pinhão	100 G	Mar do Norte
202	Caqui	160 G	Central
301	Pêssego	130 G	Mar do Sul
302	Kiwi	200 G	Mar do Sul

Leva muito tempo para procurar por cada item linha por linha.

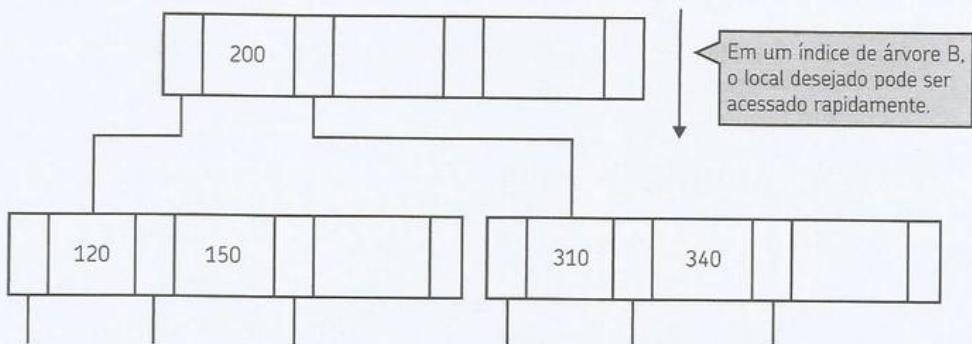
Um índice é uma ferramenta que permite acessar rapidamente o local do dado procurado. Quando procura-se por algum dado em um grande banco de dados, a busca com índices promete resultados rápidos.

Cód. do produto	Nome do produto	Preço unitário	Distrito
101	Melão	800 G	Mar do Sul
102	Morango	150 G	Central
103	Maçã	120 G	Mar do Norte
104	Limão	200 G	Mar do Sul
201	Pinhão	100 G	Mar do Norte
202	Caqui	160 G	Central
301	Pêssego	130 G	Mar do Sul
302	Kiwi	200 G	Mar do Sul

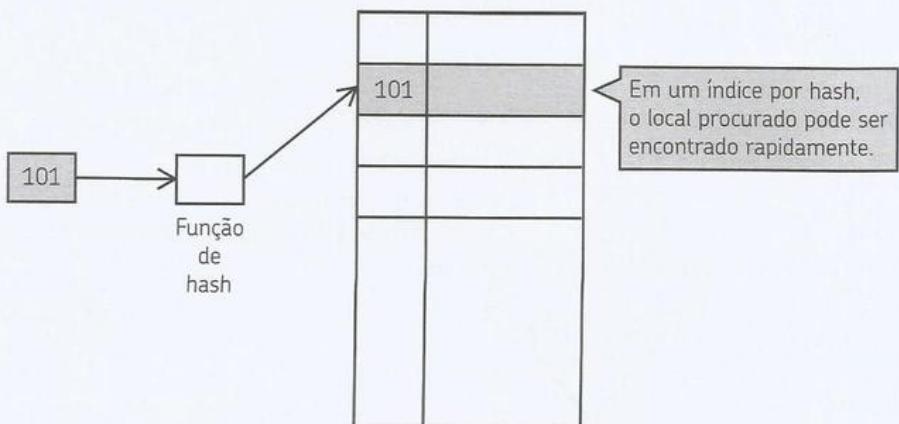
A localização do dado procurado pode ser acessada rapidamente usando seu índice.

Índice

Métodos de indexação incluem árvore binária (B-tree) e métodos de hash. Um índice de árvore-B é composto de nós ancestrais e filhos, que podem ter mais nós filhos. Os nós são dispostos de forma ordenada. Cada ancestral contém informações sobre os valores mínimo e máximo contidos em todos os seus nós filhos. Isso permite que o banco de dados navegue rapidamente até o local desejado, pulando seções inteiras da árvore que não podem conter o valor que se busca.



O método de índice por *hash* encontra o local dos dados procurados aplicando uma função de hash ao valor chave dos dados. O hash age como uma impressão digital única para um valor. O método de índice por hash pode executar consultas específicas de correspondência total (full-match searches), tal como uma busca pelo código do produto 101. No entanto, ele não serve para buscar com eficiência por condições comparativas como códigos do produto maiores que 101 ou por referências parciais como produtos cujos nomes terminam em ão.



Em alguns casos, usar um índice pode não acelerar a busca – usar um índice não poupa tempo a menos que você esteja procurando apenas por uma pequena parte dos dados. Além disso, há casos onde os índices são recriados sempre que os dados são atualizados, o que resulta em um processamento mais lento de uma operação de atualização.



QUESTÕES

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

Q15

Qual índice seria mais produtivo em uma busca com um sinal de igual, um de árvore-B ou um de hash?

Q16

Qual índice seria mais produtivo em uma busca com um sinal de diferente, um de árvore-B ou um de hash?

OTIMIZAÇÃO DE UMA CONSULTA

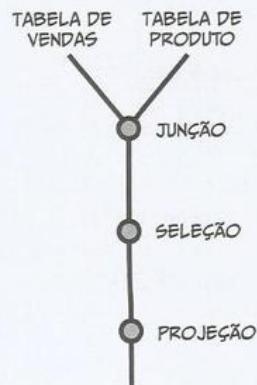
Quando você consulta um banco de dados, ele analisa a consulta SQL e considera se deve utilizar um índice para que possa processar a consulta mais rapidamente. Vamos examinar o procedimento para processar uma consulta.

O banco de dados pode decidir sobre uma ordem ideal para processar uma consulta. A maioria das consultas pode ser realizada em diversas ordens com os mesmos resultados, mas com velocidades diferentes possíveis. Por exemplo, suponha que haja uma consulta para extrair datas de venda e nomes para produtos com um preço unitário maior que 200G. Essa consulta pode ser dividida nos seguintes passos.

```
SELECT data, nome_produto  
FROM produto, vendas  
WHERE preco_unitario>=200  
AND produtos.codigo_produto = vendas.codigo_produto;
```

1. Junte as Tabelas de Produto e de Vendas
2. Selecione os produtos cujos preços unitários sejam maiores que 200G.
3. Extraia as colunas de datas e de nomes do produto.

Por exemplo, a figura à esquerda mostra a consulta processada na ordem de 1 a 3. A figura à direita mostra a consulta processada na ordem de 3 a 1. De qualquer forma, as consultas são equivalentes.



No entanto, quando processada de 1 a 3, a mesma consulta geralmente exigiria um tempo de processamento maior, porque quando a primeira junção é feita, uma tabela intermediária com muitas linhas pode ser criada. Por outro lado, o procedimento de 3 para 1 exige um tempo de processamento menor, já que a seleção e a projeção acontecem primeiro, eliminando dados indesejados o mais rapidamente possível. Assim, a mesma consulta pode exigir um tempo de processamento diferente, dependendo da ordem na qual a projeção, seleção e junção são executadas.

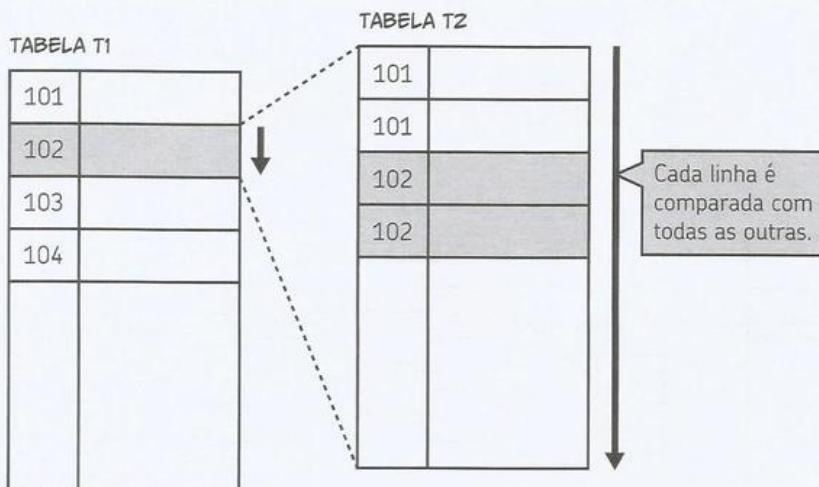
Geralmente, o banco de dados deve usar as seguintes regras para encontrar a melhor ordem para consultas:

- Executar seleção primeiro para reduzir o número de linhas.
- Executar projeção primeiro para reduzir o número de colunas irrelevantes ao resultado.
- Executar junção (*join*) mais tarde.

Há diferentes técnicas para executar projeção, seleção e junção, respectivamente. Para seleção, pode-se usar tanto uma busca de correspondência total quanto uma baseada em índice. Para a junção, os métodos a seguir estão disponíveis.

LAÇOS ANINHADOS

O método de laços aninhados compara uma linha em uma tabela a várias linhas em outra (veja a figura abaixo). Por exemplo, um dos valores em uma linha na Tabela T1 é usado para encontrar linhas correspondentes na Tabela T2. Se os valores forem os mesmos, então uma linha de junção é criada.



FUSÃO ORDENADA (SORT MERGE)

Esse método ordena (sort) e então funde (merge) linhas em várias tabelas (veja a figura abaixo). Primeiro, as Tabelas T1 e T2 são ordenadas parcial ou inteiramente. Então são comparadas começando pela linha de cima, e uma linha de junção é criada sempre que o mesmo valor é encontrado. Já que as linhas já foram ordenadas, o processamento só precisa ser feito em uma direção, de modo que isso levará menos tempo. Você deve ter em mente, no entanto, o tempo necessário para a ordenação inicial.

TABELA T1

101	
102	
103	
104	

TABELA T2

101	
101	
102	
102	

Primeiro ordena,
depois compara.

HASH

Um hash divide uma das tabelas usando uma função hash e então a funde com uma linha em outra tabela que tenha o mesmo valor. Esse método seleciona efetivamente a linha para a junção.

TABELA T1

Função
Hash



TABELA T2

Função
Hash

OTIMIZADOR

Quando uma consulta é processada, decide-se entre essas técnicas diferentes para obter um ótimo desempenho. Em um banco de dados, a função responsável pela otimização de consultas é chamada de otimizador. Há dois tipos comuns.

PROCESSAMENTO COM BASE EM REGRAS

Certas regras são estabelecidas antes que qualquer operação seja executada. Por exemplo, algumas operações podem ser combinadas ou reordenadas quase da mesma forma que uma equação algébrica pode ser manipulada e ainda querer dizer a mesma coisa. O otimizador tenta encontrar a forma mais eficiente de processar a consulta que forneça os mesmos resultados.

PROCESSAMENTO COM BASE EM CUSTO

Esse método tenta estimar o custo do processamento da consulta, baseado em estatísticas que o banco de dados mantém. Esse tipo de processamento às vezes é mais flexível do que o baseado em regras, mas exige atualizações periódicas das estatísticas do banco de dados. Gerenciar e analisar essas estatísticas leva muito tempo.

RESUMO



- Você pode estabelecer privilégios de usuário em um banco de dados.
- O uso de travas (bloqueios) garante consistência quando um banco de dados tem muitos usuários.
- A indexação permite buscas rápidas.
- Um banco de dados contém funções para recuperação contra desastre.

RESPOSTAS

Q1

COMMIT;

Q2

ROLLBACK;

Q3 Sim

Q4 Não

Q5 Não

Q6 Não

Q7 Vai aumentar

Q8 Vai diminuir

Q9

```
GRANT SELECT ON produtos TO Departamento_Exportacao;
```

Q10

```
REVOKE DELETE ON produtos FROM Departamento_Negocios_Internacionais;
```

Q11

	Pesquisar	Inserir	Atualizar	Excluir
Dept. de negócios internacionais	SIM	SIM	SIM	SIM
Dept. de mercadorias	NÃO	SIM	NÃO	SIM
Dept. de exportação	NÃO	NÃO	SIM	SIM

Q12

Uma reversão é executada, uma vez que não ocorreu efetivação no momento da ocorrência de falha.

Q13

Um avanço (reaplicação) é executado, uma vez que ocorreu efetivação no momento da ocorrência da falha.

Q14

Nenhuma operação de recuperação é necessária, uma vez que a efetivação já tinha ocorrido no momento do ponto de checagem.

Q15

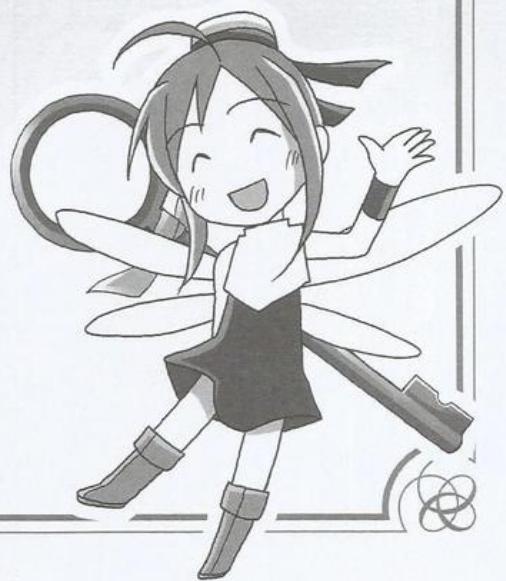
Hash

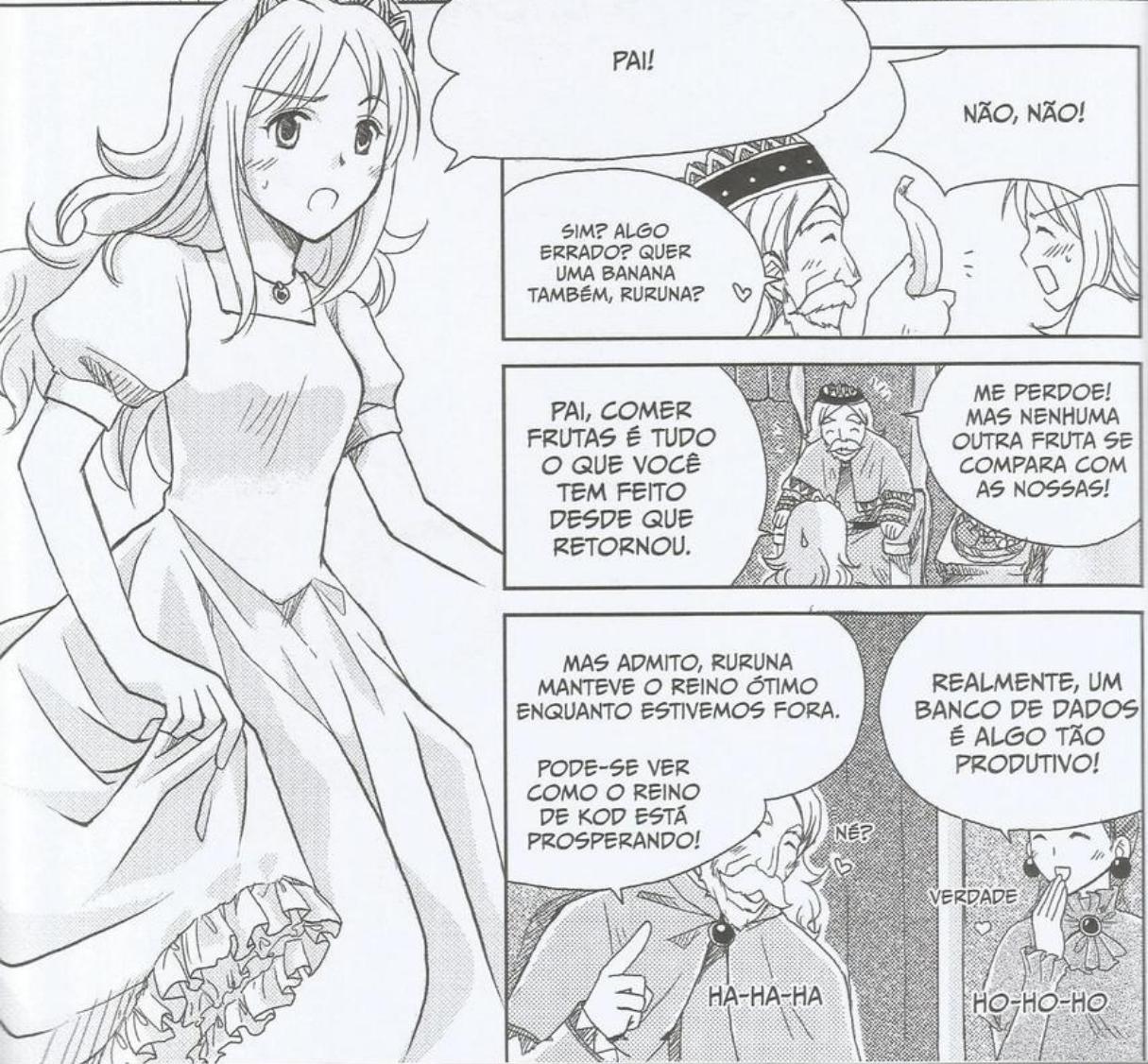
Q16

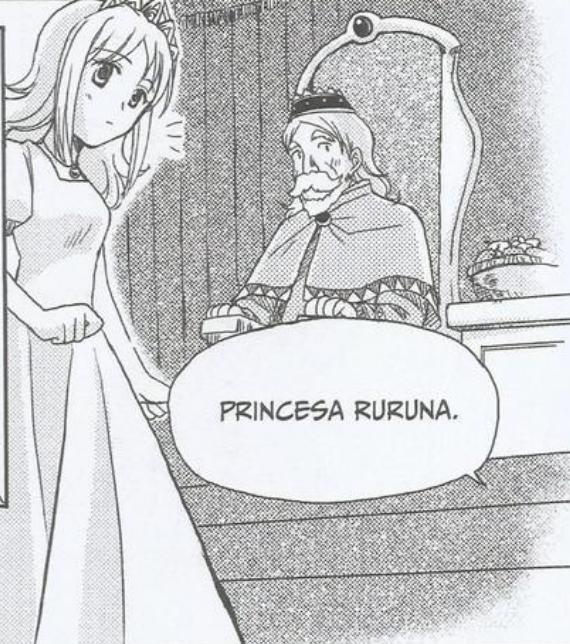
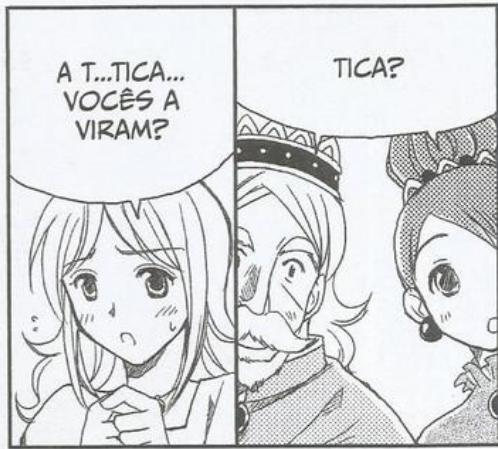
Árvore-B

6

OS BANCOS DE DADOS ESTÃO EM
TODOS OS LUGARES!





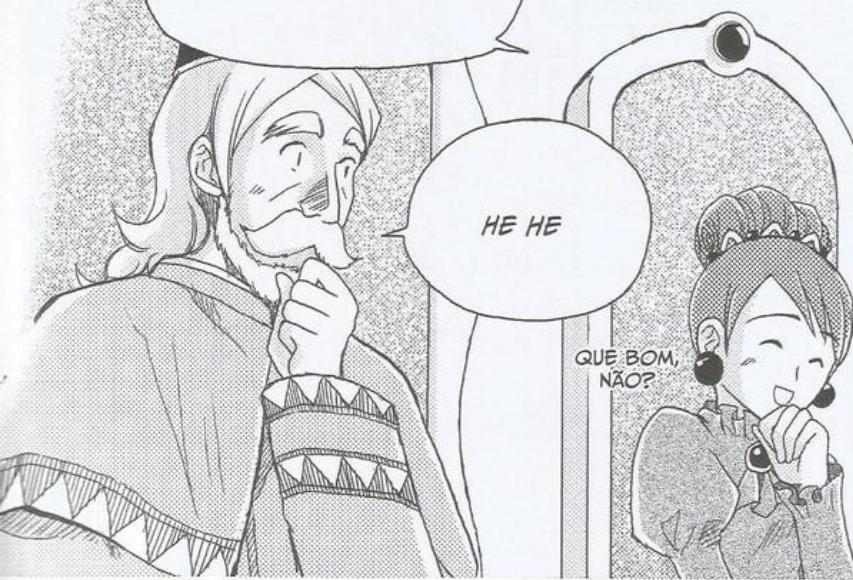




VOSSA ALTEZA,
ESSES DOIS ESTÃO,
REALMENTE...

SE DANDO BEM, NÃO?

ELA SE FOI!



ELA NÃO ESTÁ EM
LUGAR NENHUM!

TICA, POR QUE VOCÊ
DESAPEARECEU ASSIM
DE REPENTE?

EU QUERIA A
CHANCE DE
DIZER A ELA...

PELO
MENOS UMA
DESPEDIDA.

EU, TAMBÉM.

QUEM VOCÊS
ESTÃO
PROCURANDO?

!!!

O QUÉ?

ESTOU AQUI!

OH!

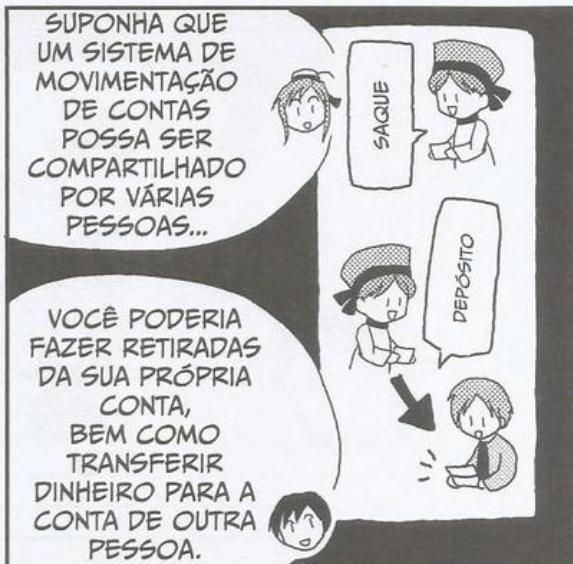
TICA!

TCHARAM!

HI HI HI

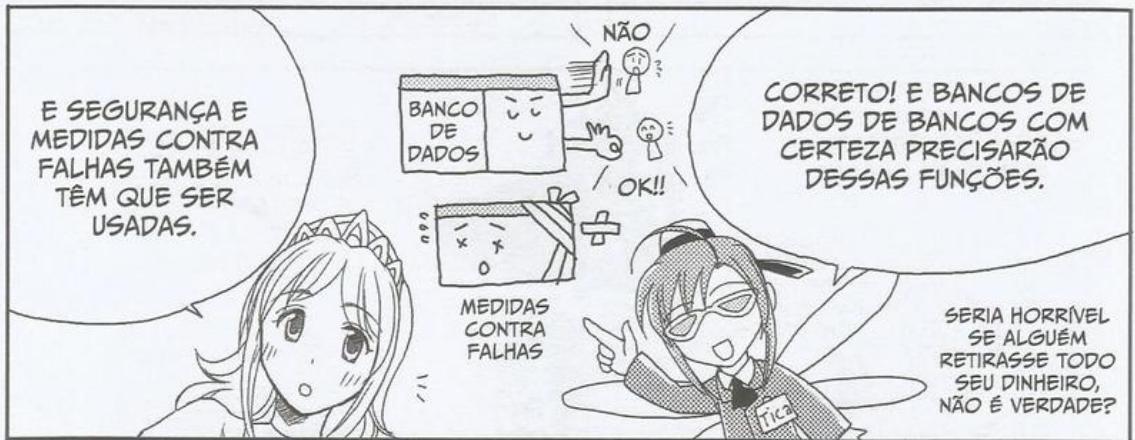


BANCO DE DADOS EM USO

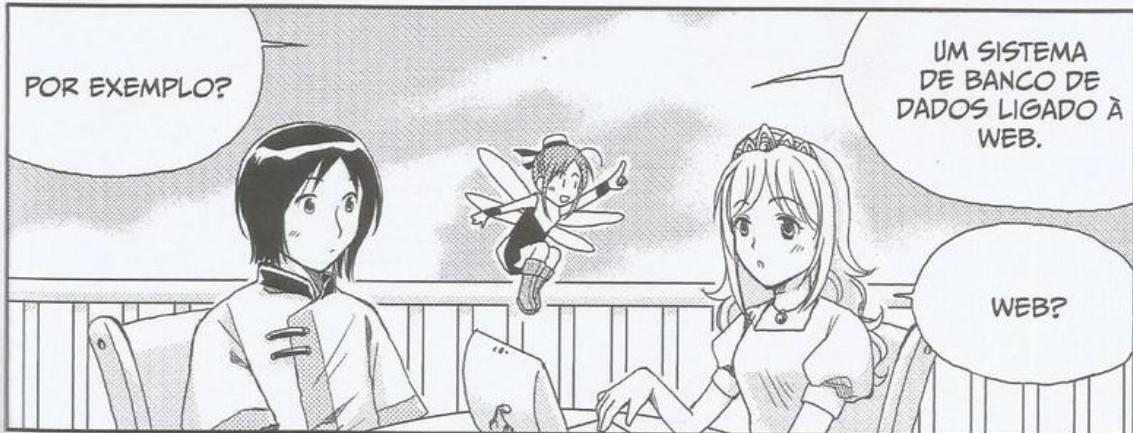


ALGUMAS VEZES, SISTEMAS DE RESERVAS DE PASSAGENS DE TREM UTILIZAM BANCOS DE DADOS.





BANCOS DE DADOS E A WEB



QUANDO FOR
PROCURAR
POR UM CERTO
LIVRO QUE
VOCÊ TENHA EM
MENTE,

PLOF!

BASTA DIGITAR UMA
PALAVRA-CHAVE
NUM NAVEGADOR DA
INTERNET.

PESQUISA DE
LIVROS >>>
PESQUISA DA TICA

CATEGORIAS ▾

PALAVRA-CHAVE



PESQUISA
DA TICA

DIGA UMA
CATEGORIA DE
ALGUM LIVRO QUE
VOCÊ PROCURA.

BOM, DEIXE-ME VER... ACHO
QUE SERIA "FRUTA", IMAGINO.

ENTÃO DIGITE
FRUTA NESTE
CAMPO DE
PALAVRA-CHAVE.

PALAVRA-CHAVE

Fruta

ESSA PALAVRA-CHAVE
É ENVIADA NUMA
REQUISIÇÃO HTTP.

UM COMPUTADOR QUE
RECEBE UMA REQUISIÇÃO
E A PROCESSA É UM
SERVIDOR.

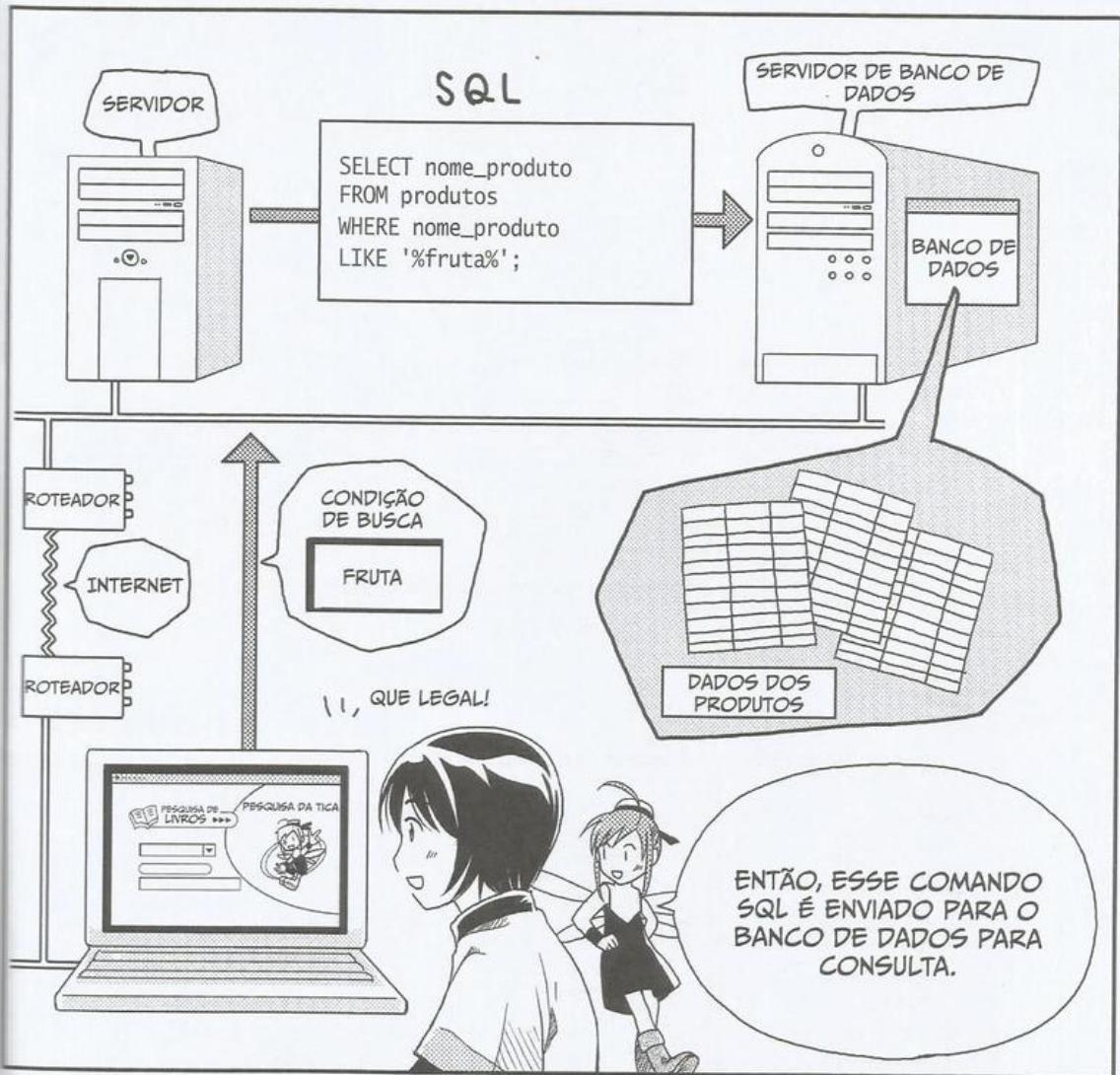
UM COMANDO SQL É
GERADO NO SERVIDOR,
EU SUPONHO.

PESQUISA DE
LIVROS >>>

CATEGORIAS ▾

PALAVRA-CHAVE

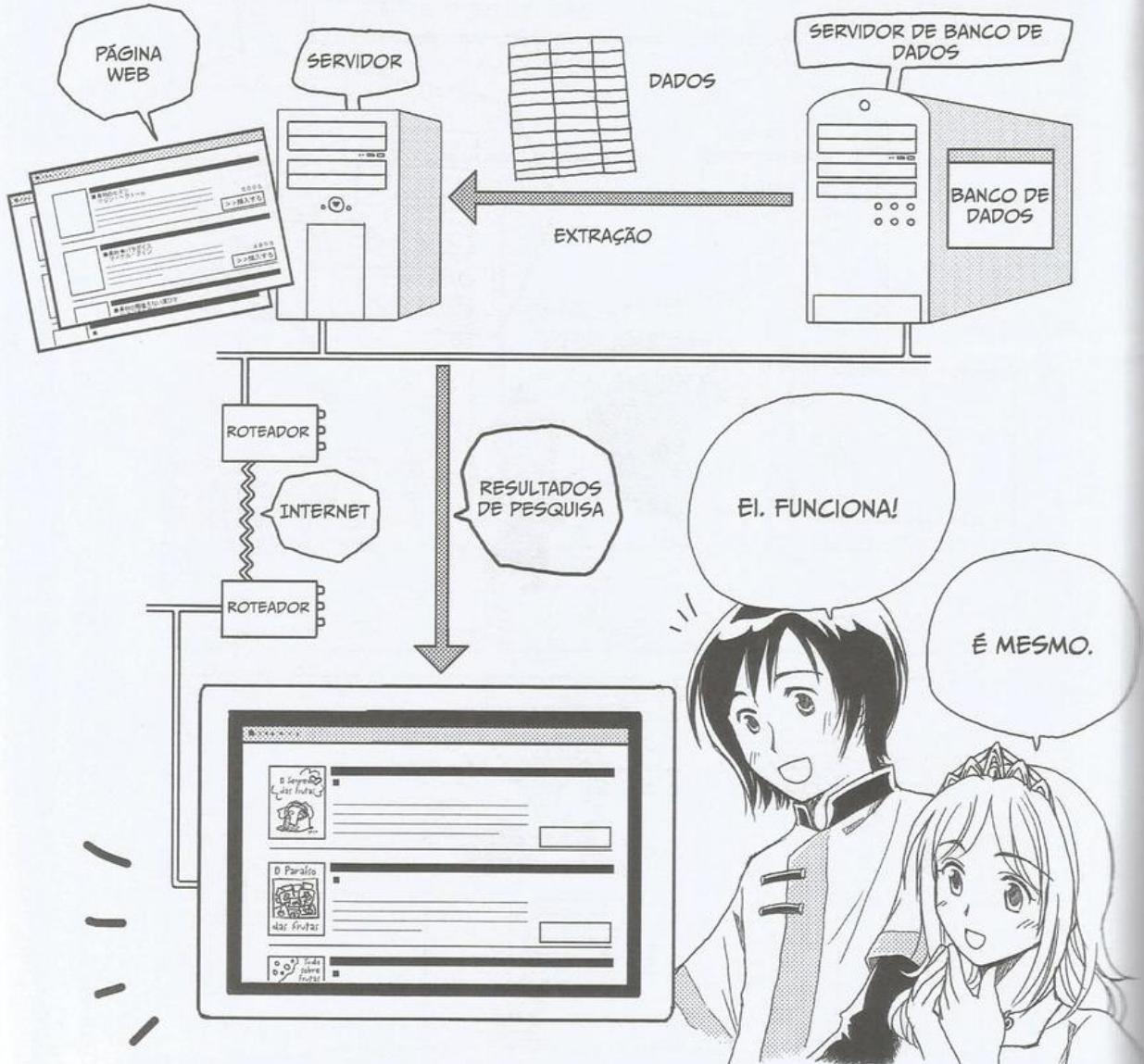
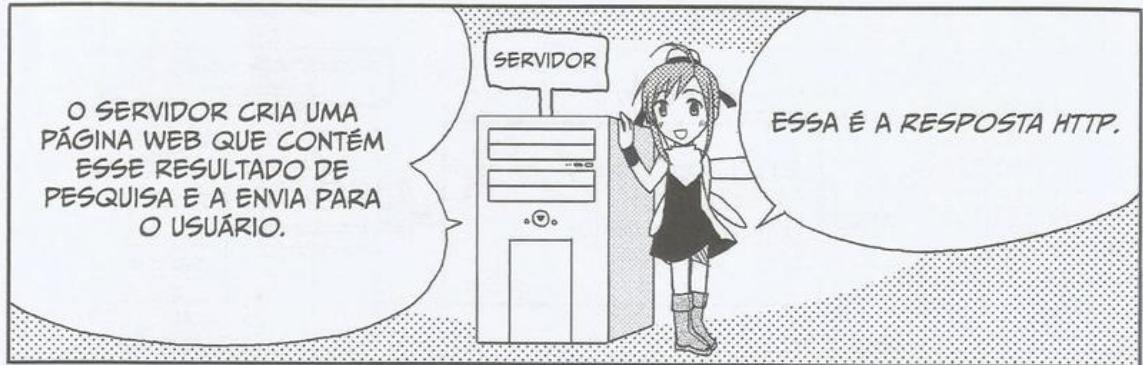
Fruta



POR SUA VEZ, O BANCO DE DADOS RETORNA OS DADOS DOS PRODUTOS COMO RESULTADO.

TÁ.

SERVIDOR DE BANCO DE DADOS





IMAGINE MUITOS CLIENTES ACESSANDO UMA LIVRARIA NA WEB AO MESMO TEMPO.

MESMO QUE TRAVAS E FUNÇÕES DE SEGURANÇA ESTEJAM PROTEGENDO TOTALMENTE O BANCO DE DADOS,



NESSE CASO, A CARGA DE PROCESSAMENTO É DIVIDIDA ENTRE UMA SÉRIE DE SERVIDORES

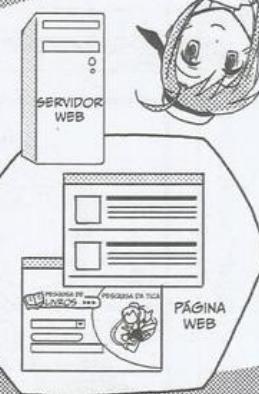
VOCÊ QUER DIZER QUE SE USA MAIS DE UM SERVIDOR?

SIM, A CARGA É DISTRIBUÍDA ENTRE DIVERSOS SERVIDORES, POR EXEMPLO, UM SERVIDOR WEB E UM SERVIDOR DE APLICAÇÕES.

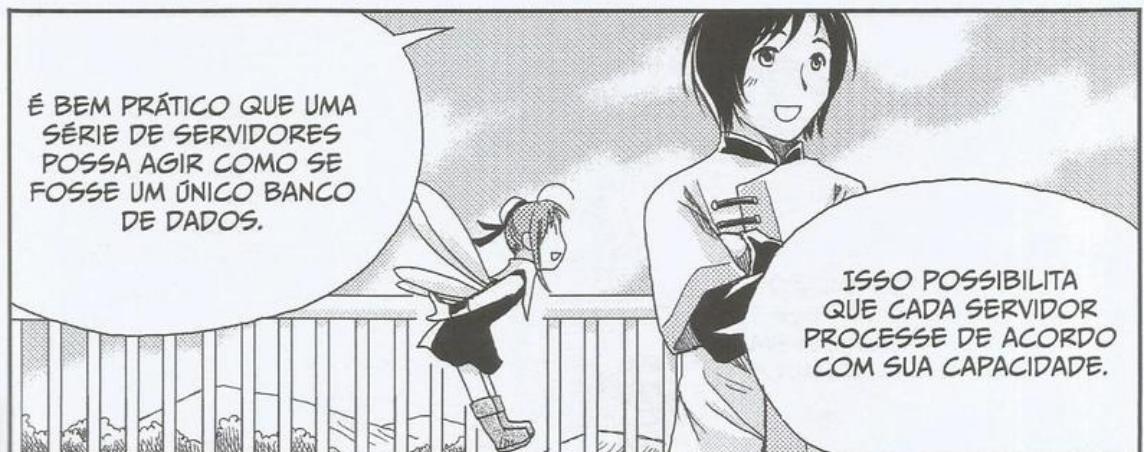
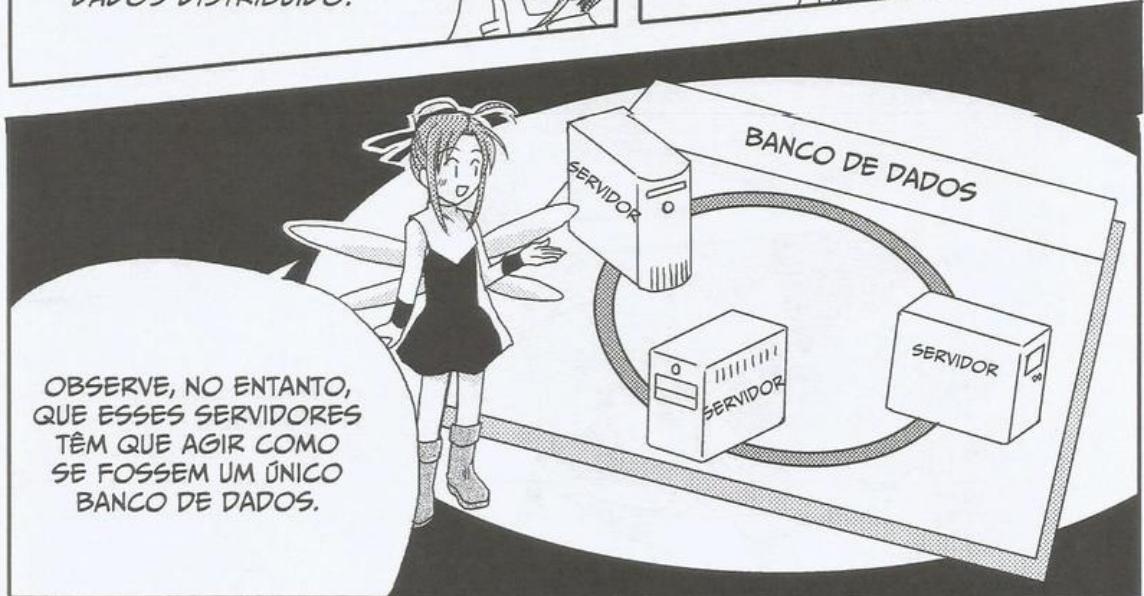


UM SERVIDOR WEB É AQUELE RESPONSÁVEL POR CRIAR AS PÁGINAS WEB, CERTO?

ISSO! E UM SERVIDOR DE APLICAÇÕES FICA ENCARREGADO DE COMPOR OS COMANDOS SQL, ENTRE OUTRAS COISAS.



BANCOS DE DADOS DISTRIBUÍDOS



MUITOS SERVIDORES OFERECEM PROTEÇÃO EXTRA CONTRA FALHAS, TAMBÉM!

FALHA

Nocaute !!

TI-CA!
TI-CA!

ISSO SIGNIFICA QUE O SISTEMA DO BANCO DE DADOS COMO UM TODO NÃO VAI CAIR, MESMO SE A FALHA OCORRER EM ALGUNS SERVIDORES NO SISTEMA.

NO CHÃO

FALHA

NOSSA!

MAS LEMBREM-SE: É NECESSÁRIO CERTO CUIDADO PARA LIDAR COM O SEU BANCO DE DADOS DESSA FORMA.

POR EXEMPLO?

QUANDO UMA TRANSAÇÃO É EFETIVADA, VOCÊ DEVE GARANTIR CONSISTÊNCIA EM TODO SEU BANCO DE DADOS DISTRIBuíDO.

ALÉM DISSO, TODOS OS SERVIDORES DEVEM SER ATUALIZADOS APROPRIADAMENTE CASO OCORRA QUALQUER PROBLEMA NA REDE.

PROCEDIMENTOS ARMAZENADOS E GATILHOS

UMA REDE É OBRIGATÓRIA EM QUALQUER AMBIENTE ONDE UM CONJUNTO DE SERVIDORES É USADO.

ISSO! É AÍ QUE OS PROCEDIMENTOS ARMAZENADOS (STORED PROCEDURES) SÃO ÚTEIS;

ELES SÃO CRIADOS ÀS VEZES PARA AJUDAR A REDUZIR A SOBRECARGA NA REDE.

ARMAZENADOS?

AHA!

EM OUTRAS PALAVRAS,
ARMAZENAR NÃO SIGNIFICA
COLOCAR NA MEMÓRIA?

SIM!

PARA REDUZIR A SOBRECARGA NA REDE, OPERAÇÕES FREQUENTEMENTE USADAS PODEM SER ARMAZENADAS EM BANCO DE DADOS.

OPERAÇÕES FREQUENTEMENTE USADAS, VOCÊ DISSE...QUE TIPO DE OPERAÇÕES SÃO ESSAS, EU ME PERGUNTO.

HUMMM?

BOM, JÁ QUE ESTAMOS FALANDO SOBRE OPERAÇÕES PARA COMPRAR UM LIVRO, SUBTRAIR DA CONTA DE ESTOQUE NA TABELA DE ESTOQUE E ADICIONAR DADOS NA TABELA DE REMESSA,

DEIXA EU PENSAR...

ELAS NÃO SÃO OPERAÇÕES COMUNS?

SIM, DE FATO.

OPERAÇÕES QUE TENHAM PROBABILIDADE DE SER USADAS FREQUENTEMENTE COMO PROCEDIMENTOS

PODEM SER ARMAZENADAS NO BANCO DE DADOS, DE ANTEMÃO!

SE PREPARARMOS UM PROCEDIMENTO ARMAZENADO, NÃO TEREMOS QUE CRIAR UM COMANDO SQL

CADA VEZ QUE QUISERMOUS REDUZIR O ESTOQUE E PROCESSAR UMA OPERAÇÃO DE ENVIO.

DESSA FORMA, A CARGA OPERACIONAL NA REDE É REDUZIDA.

ENTENDI.

E O NOSSO TRABALHO TAMBÉM.

AH, SIM, VOCÊ TEM RAZÃO!

ALÉM DISSO, VOCÊ SABE, HÁ PROCEDIMENTOS ARMAZENADOS QUE SÃO AUTOMATICAMENTE INICIADOS.

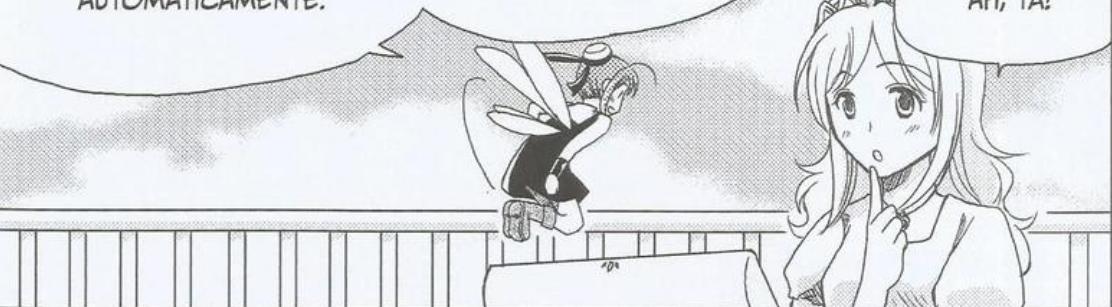
AUTOMATICAMENTE?

QUANDO OS DADOS SÃO ATUALIZADOS, POR EXEMPLO, UM PROCEDIMENTO ARMAZENADO PODE INICIAR AUTOMATICAMENTE.

ISSO É CHAMADO DE GATILHO (TRIGGER).

GATILHO...

AH, TÁ!

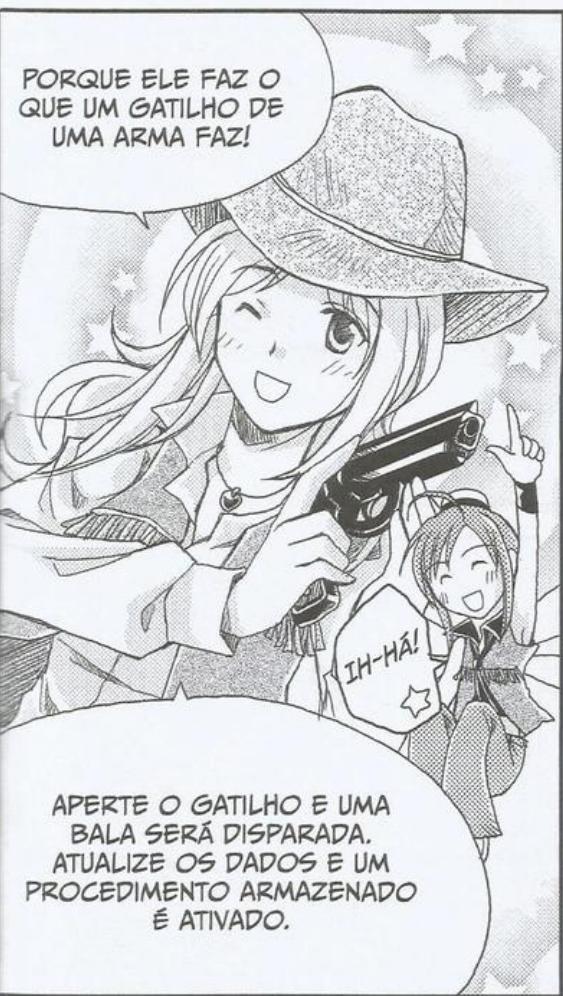


PORQUE ELE FAZ O QUE UM GATILHO DE UMA ARMA FAZ!

SERIA ÓTIMO SE, DE FATO, AO FAZER UM PEDIDO E ATUALIZAR O BANCO DE DADOS

POR QUE EU TAMBÉM ESTOU COM ESTA ROUPA?

AUTOMATICAMENTE LANÇASSEM UMA OPERAÇÃO PARA REDUZIR O ESTOQUE E PREPARAR O ENVIO.



APERTE O GATILHO E UMA BALA SERÁ DISPARADA. ATUALIZE OS DADOS E UM PROCEDIMENTO ARMAZENADO É ATIVADO.

O SIMPLES FATO DE COMPRAR UM LIVRO CRIA UM MONTE DE TRABALHO NOS BASTIDORES, NÃO É?



EXATAMENTE.

VERDADE.

EMBORA, NA MAIORIA DOS CASOS, O BANCO DE DADOS POSSA NÃO SER VISTO

QUANDO VOCÊ COMPRA UM LIVRO NA INTERNET.

DE NOSSA PARTE, PRECISAMOS APRENDER MAIS SOBRE ISSO, PASSO A PASSO.

VOCÊ NÃO ACHA, CAIN?

SIM, CONCORDO, PRINCESA.

CERTO!

A TICA NOS AJUDOU A APRENDER COMO USAR UM BANCO DE DADOS, E PODEMOS IR EM FRENTE A PARTIR DAQUI.

NÃO ESQUEÇAM QUE BANCOS DE DADOS SÃO A MANEIRA PRINCIPAL DE ORGANIZAR CONVENIENTEMENTE SEUS DADOS.

PARECE ÓTIMO! EU PROMETO. USANDO BANCOS DE DADOS,

VOU CONSTRUIR UM PAÍS MARAVILHOSO ONDE TODOS POSSAM GOZAR DE UMA FORMA DE VIDA PRODUTIVA.

VAMOS FAZER O MELHOR!

SI-I-I-M!!

VOCÊS DOIS SÓ
COMEÇARAM
RECENTEMENTE A ENTENDER
OS MECANISMOS DE UM
SISTEMA DE BANCO DE
DADOS.

MAS TENHO CERTEZA
DE QUE FICARÃO BEM
SOZINHOS.

VOCÊS
PROGREDIRAM
MUITO...
COM
CERTEZA...

MAS HÁ TANTA
COISA QUE
AINDA NÃO
APRENDEMOS!

DO QUE VOCÊ
ESTÁ FALANDO?

VOCÊ VAI FICAR
CONOSCO, NÃO
VAI, TICA?

DESCULPEM,

NÃO POSSO.

MEU TRABALHO
AQUI ESTÁ FEITO.

AH, NÃO...

TENHO MAIS
PESSOAS PARA
VISITAR.

SIM!

AFINAL, EU SOU...

VOCÊS SABEM, HÁ MUITAS
PESSOAS QUE QUEREM
APRENDER SOBRE BANCOS
DE DADOS!

ENTÃO VOCÊ VAI VISITAR
MAIS ALGUÉM QUE
ABRIU UM LIVRO SOBRE
BANCOS DE DADOS?

A FADA DOS
BANCOS DE
DADOS!

TADAM!



EU SÓ VIM ME
DESPEDIR HOJE.

HE-HE-HE

MAS NÃO PODE
EVITAR FAZER MAIS
QUE ISSO.

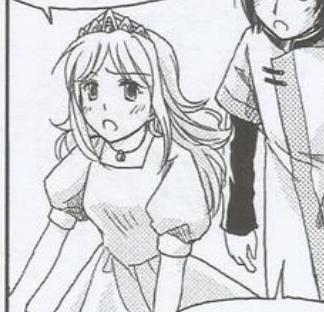
ESSE TEMPO QUE EU
PASSEI COM VOCÊS
FOI CURTO, MAS FELIZ!



TICA...

TICA, ESPERE!

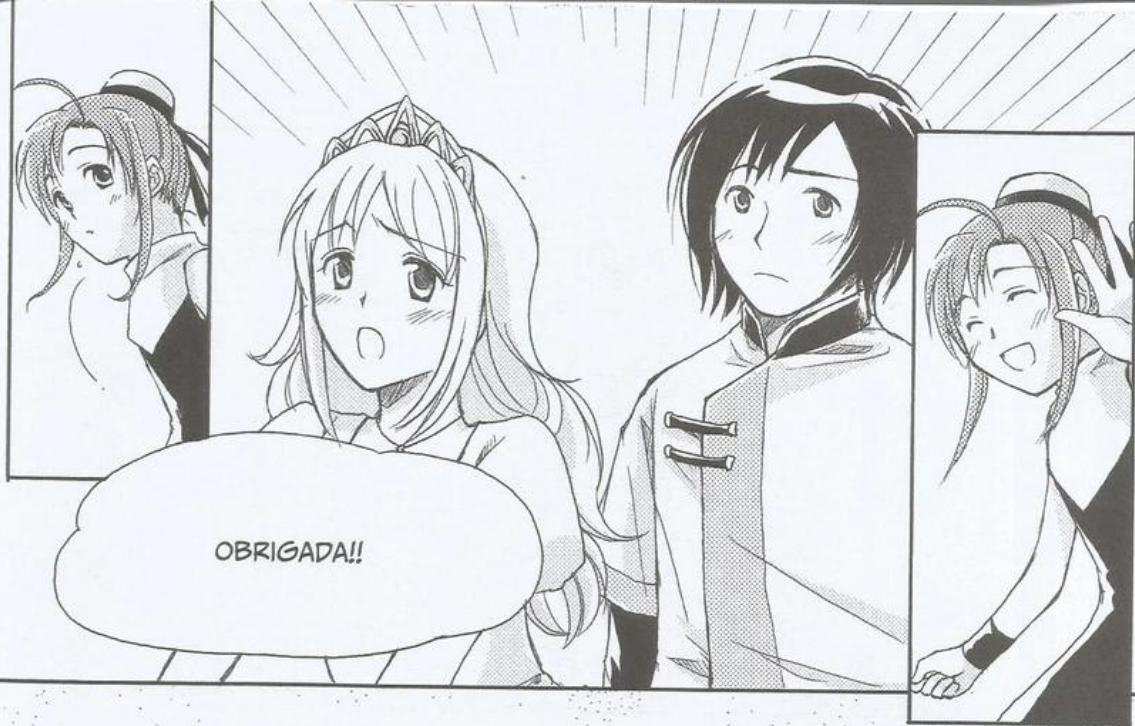
EU TENHO
QUE...



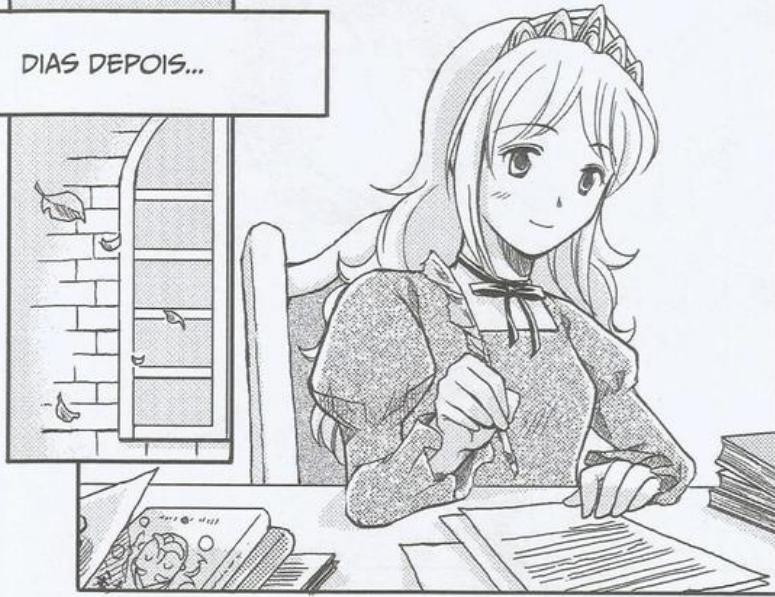
TICA FIQUE!



TICA!!



DIAS DEPOIS...





BANCOS DE DADOS
NO REINO DE KOD

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



SURGIU UMA MINÚSCULA MENINA CHAMADA...

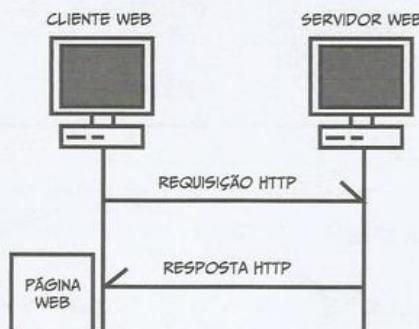
FIM

BANCOS DE DADOS NA INTERNET

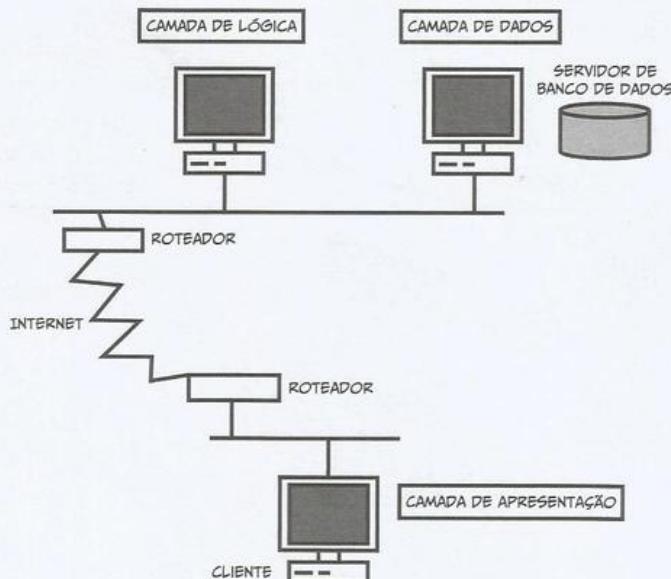


Bancos de dados são usados para muitos propósitos diferentes, como para sistemas de reserva de poltronas em trens e de depósitos bancários. Elas são indispensáveis na vida cotidiana e nas operações comerciais. Como eu mostrei para a Ruruna e o Cain, sistemas de bancos de dados com base na web são comuns também. Em um sistema com base na web, o protocolo de comunicações usado é o Protocolo de Transferência de Hipertexto (HTTP). O software servidor sendo executado em um servidor web espera por uma requisição de um usuário. Quando uma requisição de usuário (requisição HTTP) é enviada, o software responde a ela e retorna uma página correspondente (resposta HTTP).

Uma página web consiste em arquivos de texto em formato HTML. Outros arquivos especificados por URLs (Uniform Resource Locators – Localizadores Uniformizados de Recursos) são embutidos na página para apresentar informações, como imagens.



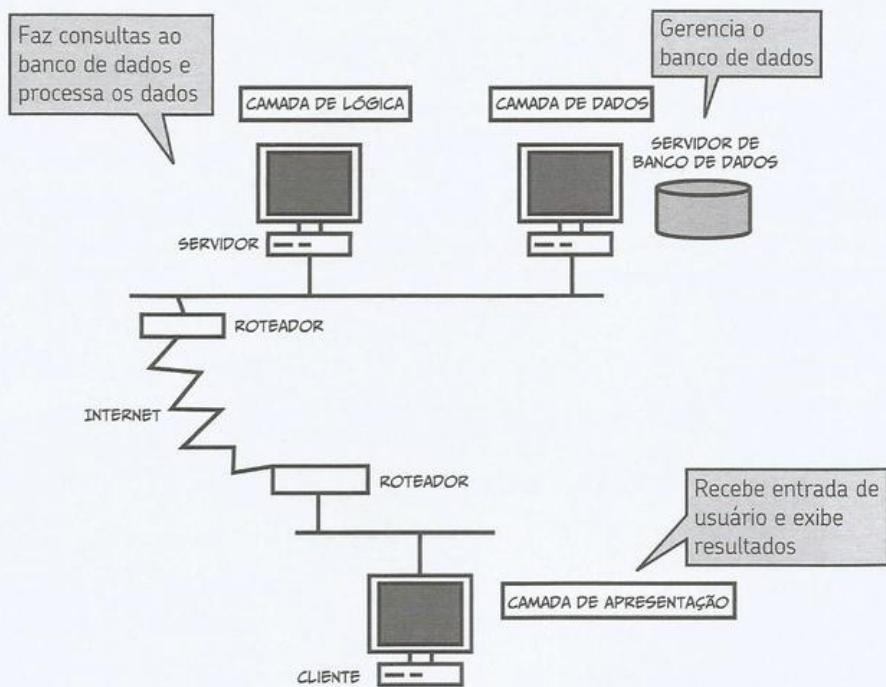
Quando um banco de dados é usado em conjunto com uma página web, um servidor de banco de dados é adicionado ao sistema mostrado acima. Esse sistema pode ser configurado em três camadas e é chamado de *sistema cliente/servidor de três camadas*. Esse sistema consiste em uma camada de apresentação, uma de lógica e uma de dados.



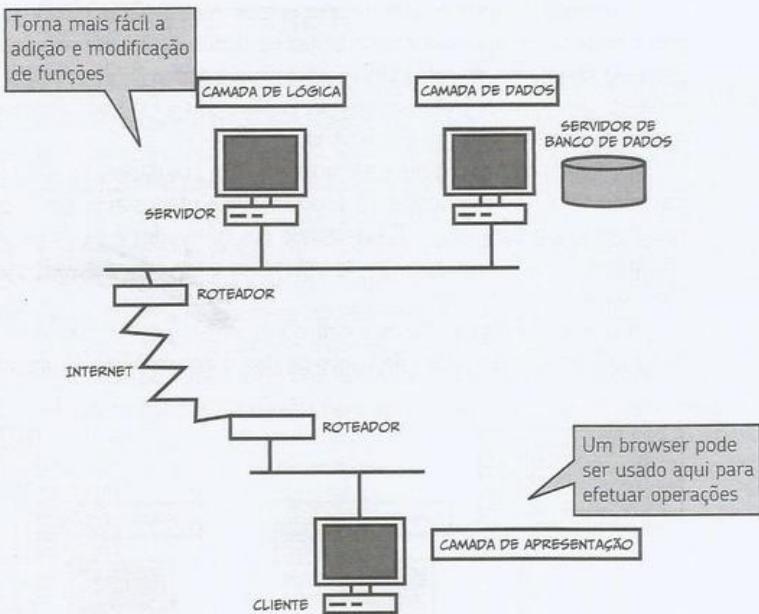
A camada de apresentação recebe entradas de usuário, como condições de consulta, que precisam ser passadas para o banco de dados. A camada de apresentação também processa resultados de consultas recebidas pelo banco de dados para que eles possam ser exibidos. Um browser (como o Internet Explorer ou o Firefox) funciona como uma ferramenta de apresentação para o usuário.

A camada de lógica efetua o processamento dos dados. Essa camada é onde os comandos SQL são compostos. Os processos executados aqui são escritos em uma ou mais linguagens de programação. Dependendo dos conteúdos e da carga dos processos, diversos servidores, tal como um servidor de aplicações e um servidor web, podem ser usados para lidar com o processamento.

A camada de dados processa dados em um servidor de banco de dados. Os resultados de consultas são retornados do banco de dados em resposta a consultas SQL.



A configuração de cliente/servidor de três camadas é um sistema flexível e simples. Por exemplo, quando modificar ou acrescentar algo em uma aplicação, você pode separar a parte que deseja editar como uma camada de lógica. Na camada de apresentação, você pode usar um browser, eliminando a necessidade de instalar um programa separado.



USO DE PROCEDIMENTOS ARMAZENADOS

Em um sistema com base na web, muito tráfego na rede pode ser um problema. Por sorte, pode-se manter lógica de programação dentro do próprio servidor de banco de dados como um procedimento armazenado (stored procedure).

Procedimentos armazenados nos servidores de bancos de dados ajudam a reduzir a carga na rede, porque eliminam a necessidade de transferências frequentes de consultas SQL. Além disso, armazenar procedimentos também torna mais fácil desenvolver aplicativos, uma vez que processos padronizados podem ser encapsulados em procedimentos fáceis de usar. Na realidade, os procedimentos armazenados são apenas um tipo especial de uma categoria mais abrangente chamada de *programas armazenados*. Os outros dois tipos de programas armazenados são as *funções armazenadas* e os *gatilhos* (triggers).

TIPOS DE PROGRAMAS ARMAZENADOS

Programa	Definição
Procedimento armazenado	Programa que não retorna valores após o processamento
Função armazenada	Programa que retorna valores após o processamento
Gatilho	Programa que é executado automaticamente antes e depois de operações no banco de dados



QUESTÕES

Você consegue responder às próximas questões? As respostas corretas estão na página 205.

Q1

Num sistema cliente/servidor de três camadas, em qual delas o banco de dados opera?

Q2

Em um sistema cliente/servidor de três camadas, em qual delas ocorrem as interações com o usuário e onde os resultados são apresentados?

O QUE É UM BANCO DE DADOS DISTRIBUÍDO?



Em um sistema com base na web, o processamento é distribuído entre um servidor de banco de dados, um servidor de Internet e um programa navegador, cada um responsável por diferentes tarefas. Esse tipo de sistema distribuído permite flexibilidade e diminui a capacidade de processamento necessária em cada servidor.

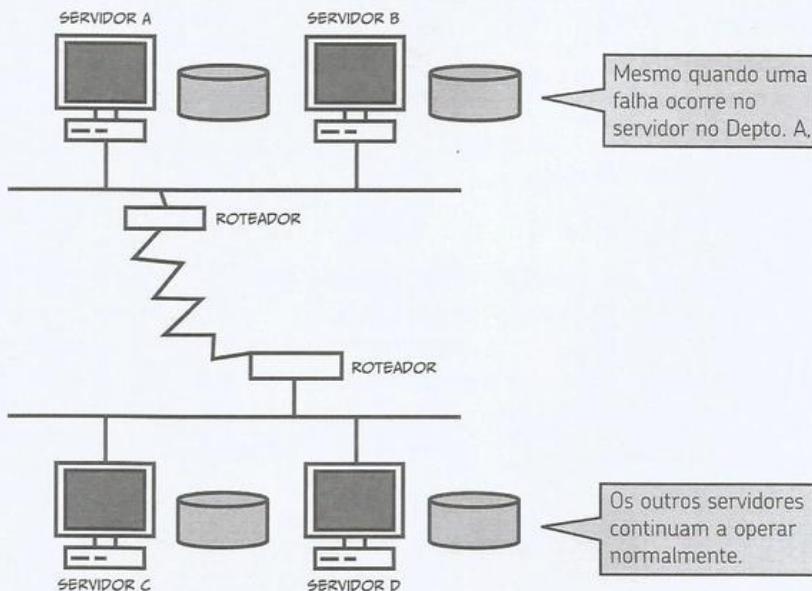
Mas um servidor de banco de dados por si só pode ser distribuído entre diversos servidores. Servidores de bancos de dados distribuídos podem estar em locais diferentes ou na mesma rede. Observe, no entanto, que um banco de dados distribuído pode ser tratado como se fosse um banco de dados único. Se o banco distribuído parece ser um servidor único, o usuário não precisa se preocupar com a localização dos dados ou com transferências.

Um banco de dados pode ser distribuído horizontalmente e verticalmente.

DISTRIBUIÇÃO HORIZONTAL

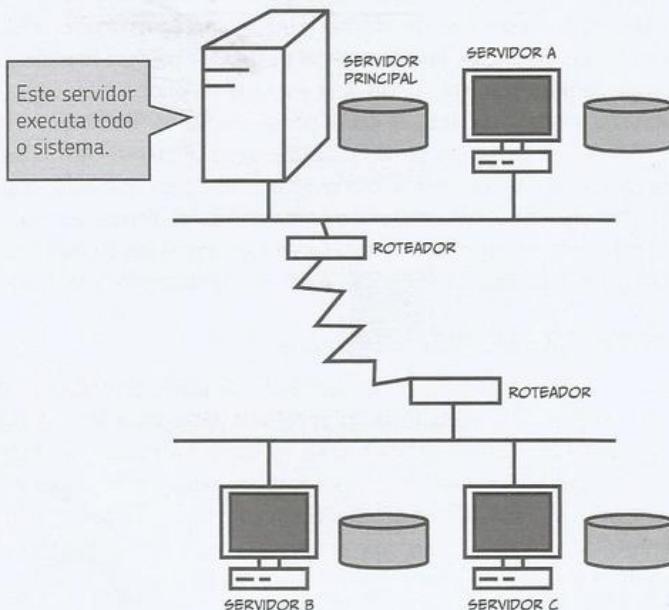
Distribuição horizontal utiliza diversos servidores de banco de dados em colaboração igualitária. Cada servidor de banco de dados pode usar dados dos outros e, por sua vez, cada um está disponível para todos os demais. Essa estrutura é utilizada para sistemas de bancos de dados extendidas que operam separadamente em departamentos isolados.

Um banco de dados distribuído horizontalmente é um sistema projetado para ser resistente a falhas, uma vez que problemas num servidor não afetarão a operação do banco como um todo.



DISTRIBUIÇÃO VERTICAL

Distribuição vertical atribui diferentes funções para diferentes servidores de banco de dados. Um deles funciona como servidor principal e executa um papel central, enquanto outros estão encarregados de tarefas de processamento enviadas a eles. Um banco de dados distribuído verticalmente facilita o gerenciamento do servidor principal, embora ele possa receber uma grande carga. Um exemplo de distribuição vertical incluiria um servidor único atendendo toda uma empresa e servidores individuais operando em cada um de seus *departamentos*.

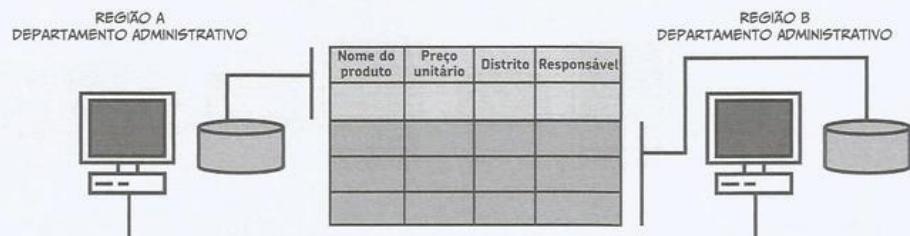


PARTICIONAMENTO DE DADOS

Em um banco de dados distribuído, os dados ficam espalhados entre diversos servidores de armazenamento. Deve-se considerar cuidadosamente como dividir os dados. Eles podem ser separados das seguintes formas.

PARTICIONAMENTO HORIZONTAL

Um *particionamento horizontal* é uma divisão tendo linhas como unidade. Linhas resultantes dessa separação são distribuídas por vários servidores. Este tipo de particionamento é muito utilizado quando dados que podem ser ordenados em grupos relacionados, frequentemente acessados ao mesmo tempo, são armazenados em um mesmo servidor.



PARTICIONAMENTO VERTICAL

Um *particionamento vertical* é o que divide dados utilizando colunas como unidade. Colunas resultantes dessa divisão são distribuídas entre servidores. Por exemplo, uma partição vertical pode ser utilizada para gerenciar e unir bancos de dados independentes que pertençam a departamentos diversos como o de mercadorias, o de negócios internacionais e o de exportação.

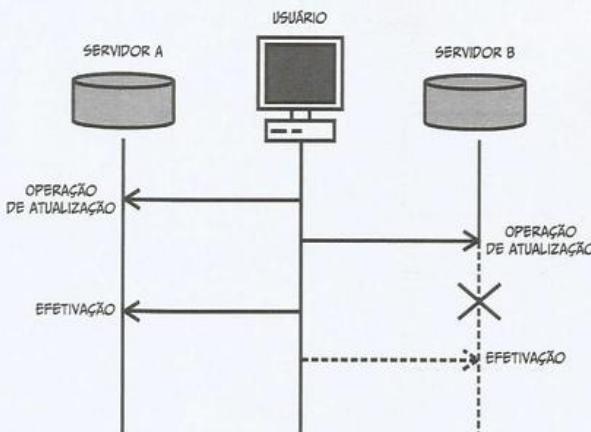


EVITANDO INCONSISTÊNCIAS COM EFETIVAÇÃO EM DUAS FASES

Bancos de dados em servidores diferentes num sistema distribuído podem ser configurados para agir como se fossem um único banco no ponto de vista dos usuários. Para conseguir isso, vários passos têm que ser seguidos para lidar com o fato de que os dados estão na realidade espalhados entre diversos servidores.

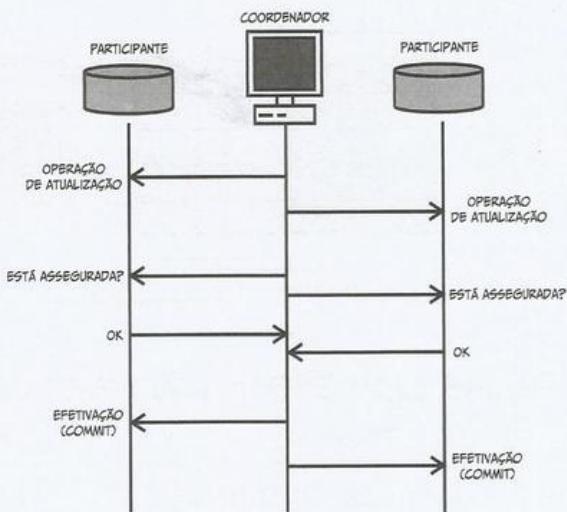
Primeiro, sempre que ocorrer efetivação, os dados em todos os servidores precisam ser atualizados de forma consistente.

Em um sistema de banco de dados distribuído, o método-padrão de efetivação pode fazer com que um dos servidores seja atualizado e outros não, como mostrado abaixo. Isso é uma violação da propriedade de atomicidade das transações, uma vez que elas estariam sendo encerradas não por uma efetivação (commit) nem por uma reversão (rollback). Isso também faria com que o sistema de banco de dados como um todo se tornasse inconsistente.

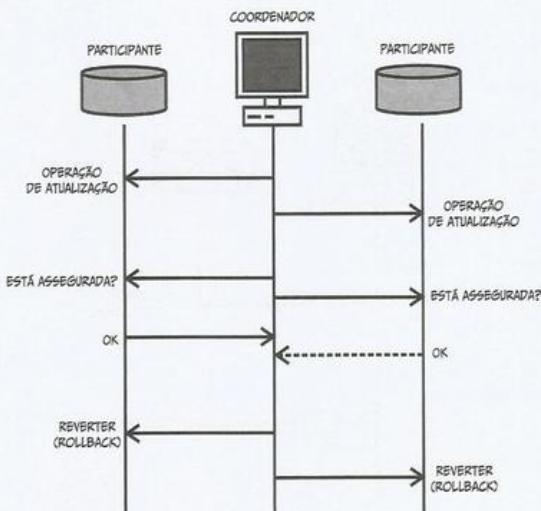


Dessa forma, uma efetivação em duas fases é adotada em um sistema de banco de dados distribuído. Cria uma única operação de efetivação a partir de duas ações diversas.

Uma operação de efetivação em duas fases envolve um coordenador e outros participantes. Na primeira fase de uma operação desse tipo, o coordenador pergunta aos participantes se a operação de efetivação é possível. Os participantes enviam uma mensagem OK se for o caso. Esse passo é conhecido como preparação. Na segunda fase, o coordenador envia instruções para uma efetivação, e todos os participantes procedem de acordo.



Se qualquer ponto falhar em assegurar o procedimento de efetivação nessa operação em duas fases, todos os participante recebem comandos para reversão. É por isso que os bancos de dados em todos os servidores permanecem consistentes entre si.





QUESTÕES

Tente responder às seguintes questões sobre efetivação em duas fases. As respostas estão na página 205.

Q3

Em um esquema de efetivação em duas fases, quais instruções o coordenador dá durante a primeira fase?

Q4

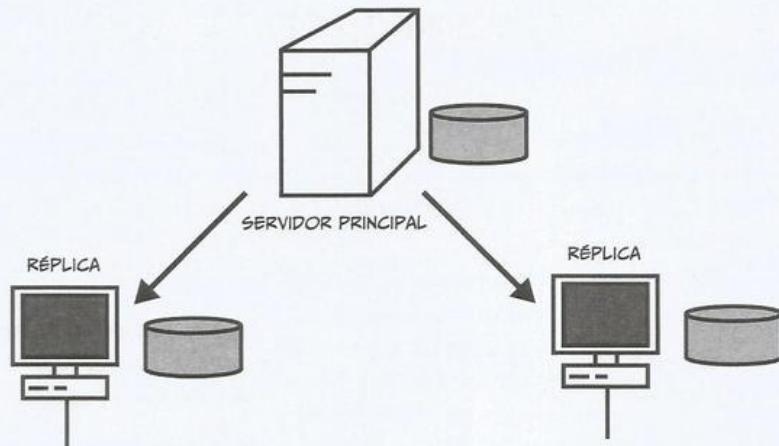
Em um esquema de efetivação em duas fases, quais instruções o coordenador dá durante a segunda fase?

REPLICAÇÃO DE BANCO DE DADOS

Alguns bancos de dados distribuídos contêm um banco de dados em duplicidade, ou réplica, que reduz a carga na rede. Essa prática é chamada de replicação. O banco de dados primário é chamado de banco de dados principal, e a cópia é chamada de réplica. Há diversos tipos de replicação.

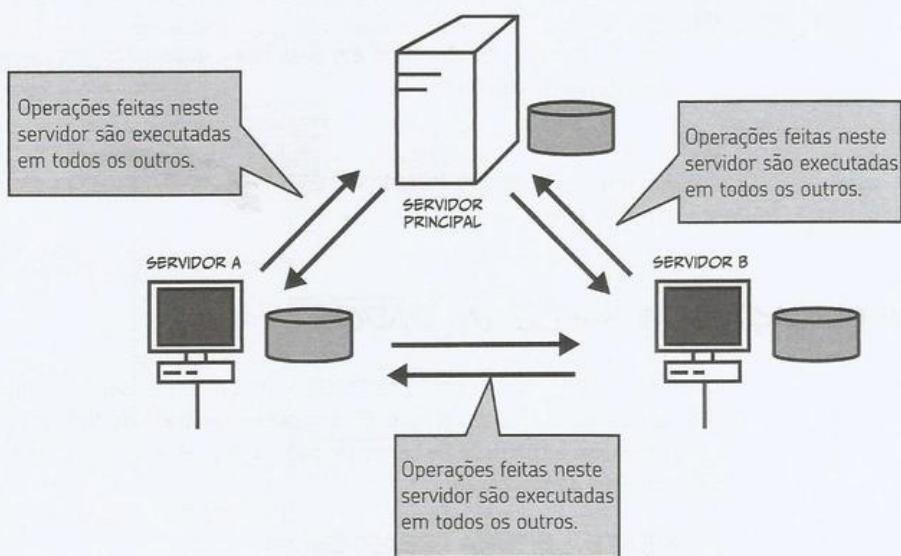
SOMENTE LEITURA (READ-ONLY)

Uma réplica somente para leitura é criada e baixada do banco de dados principal no servidor principal. Para alterar os dados, os usuários devem conectar-se ao servidor principal.



REPLICAÇÃO ATIVA EM TODOS OS SERVIDORES

Nesse método, o mesmo banco de dados principal é compartilhado por todos os servidores. Atualizações em qualquer um dos servidores são feitas em todos os outros.



OUTRAS APLICAÇÕES DE BANCOS DE DADOS



Essa seção final introduz tecnologias aplicadas relacionadas aos bancos de dados.

XML

A Linguagem Extensível de Marcação (XML) é cada vez mais popular como um método de armazenamento de dados. O XML representa dados delimitando-os com tags. Já que essas tags podem trazer informações sobre os dados que contêm, essa linguagem é útil para armazenamento e recuperação de dados.

O XML é útil porque sua sintaxe estritamente estruturada facilita processos programados. Além do mais, o XML aparece em arquivos de texto (que são fáceis de editar) e pode se comunicar com outros sistemas. Por essas razões, o XML é às vezes usado como um método de representação de dados no lugar de um banco de dados.

```
<?xml version="1.0" ?>
<produtos>
  <fruta>
    <codigo_produto>101</codigo_produto>
    <nome_produto>Melão</nome_produto>
    <preco_unitario>800</preco_unitario>
  </fruta>
  <fruta>
    <codigo_produto>102</codigo_produto>
    <nome_produto>Morango</nome_produto>
    <preco_unitario>150</preco_unitario>
  </fruta>
  <fruta>
    <codigo_produto>103</codigo_produto>
    <nome_produto>Maçã</nome_produto>
    <preco_unitario>120</preco_unitario>
  </fruta>
</produtos>
```

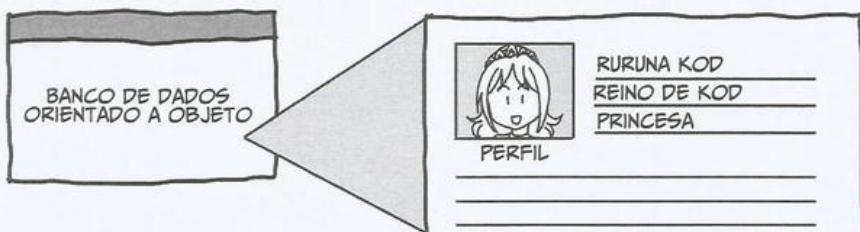
BANCOS DE DADOS ORIENTADO A OBJETO

Um banco de dados relacional armazena dados em formato de texto em tabelas. No entanto, ele pode ser inadequado quando se lida com certos tipos de dados. É aí que entra um banco de dados orientado a objeto (BDOO).

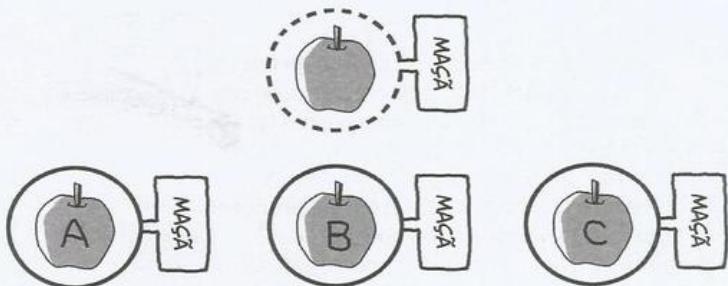
O método orientado a objeto utiliza objetos - conjuntos de dados e instruções sobre como esses dados devem ser usados. Você pode ocultar os dados e expor apenas as operações sobre os dados para lidar com o objeto como um componente independente. Essa técnica é chamada de encapsulamento.

Em um banco de dados orientado a objeto, cada objeto é representado com um identificador. Às vezes, um objeto é também chamado de instância.

Em um banco de dados orientado, você pode também gerenciar objetos compostos - um objeto definido dentro de outro. Isso significa, por exemplo, que você pode armazenar dados que consistem em uma imagem com texto como um único objeto. O banco de dados orientado a objeto permite gerenciamento flexível de dados complexos.

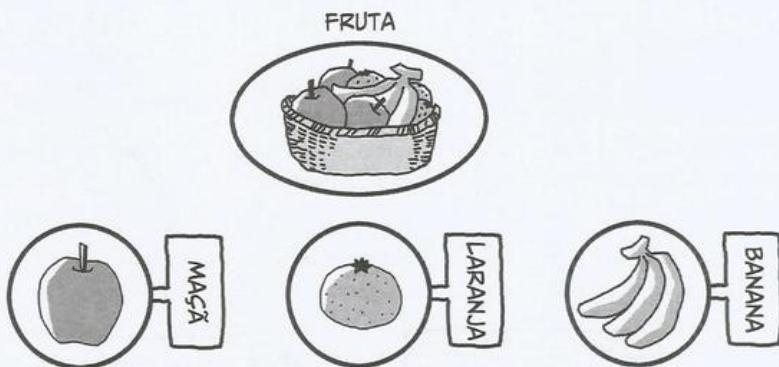


Em um banco de dados orientado a objeto, vários conceitos podem facilitar o desenvolvimento orientado a objeto. O modelo para objetos é chamado de classe. Por exemplo, suponha que você tenha criado uma classe Maçã. Os objetos (instâncias) dessa classe podem ser Maçã A, Maçã B e assim por diante. A classe Maçã permite a criação desses objetos.



Em um esquema orientado a objeto, uma classe pode também ter relacionamentos hierárquicos. Você pode criar uma classe filha que tenha os mesmos dados e funções de uma classe base. Esse relacionamento é chamado de herança. Você também pode dar funções exclusivas à classe filha.

Por exemplo, a classe Maçã e a classe Laranja podem herdar os dados e funções da classe Fruta, mas cada um deles tem seus próprios dados e funções únicas. Em um esquema orientado a objeto, você pode usar relacionamentos hierárquicos para permitir um desenvolvimento eficiente.



RESUMO



- Um sistema de cliente/servidor de três camadas é um método de configuração de sistema baseado na Internet.
- Um banco de dados age como uma camada de dados.
- Um sistema de banco de dados distribuído lida com bancos de dados que estão dispersos.
- Um método de efetivação em duas fases é usado em um banco de dados distribuído.

REPOSTAS

Q1 Camada de dados

Q2 Camada de apresentação

Q3 Preparação

Q4 Efetivar (commit) ou reverter (rollback)

COMENTÁRIOS FINAIS

Gostou de estudar bancos de dados? Você vai precisar aprender ainda mais antes que possa dominar todos os aspectos para operar um banco de dados, mas o básico sobre eles sempre é o mesmo. Entendendo profundamente o básico, você pode identificar dados significativos no mundo real e assim criar e operar bancos de dados. Você pode adquirir habilidades avançadas sobre bancos de dados trabalhando sobre seu conhecimento fundamental. Boa sorte!

COMANDOS SQL USADOS COM FREQUÊNCIA

CONSULTA BÁSICA

```
SELECT nome_coluna, ...
FROM nome_tabela;
```

CONSULTA CONDICIONAL

```
SELECT nome_coluna, ...
FROM nome_tabela
WHERE condição;
```

CONSULTA POR PADRÃO

```
SELECT nome_coluna, ...
FROM nome_tabela
WHERE nome_coluna LIKE 'padrão';
```

CONSULTA ORDENADA

```
SELECT nome_coluna, ...
FROM nome_tabela
WHERE condição
ORDER BY nome_coluna;
```

TOTALIZAÇÃO E AGRUPAMENTO

```
SELECT nome_coluna, ...
FROM nome_tabela
WHERE condição
GROUP BY nomes_colunas_para_agrupamento
HAVING condição_para_linhas_agrupadas
```

JUNÇÃO DE TABELAS

```
SELECT tabela_nome1.nome_coluna, ...
FROM nome1_tabela, nome2_tabela, ...
WHERE tabela_nome1.nome_coluna = tabela_nome2.nome_coluna
```

CRIAÇÃO DE UMA TABELA

```
CREATE TABLE nome_tabela(  
    nome1_coluna tipo_de_dado,  
    nome2_coluna tipo_de_dado,  
    ...  
)
```

CRIAÇÃO DE UMA VISUALIZAÇÃO (VIEW)

```
CREATE VIEW nome_visualizacao  
AS SELECT comando
```

EXCLUSÃO DE UMA TABELA REAL

```
DROP TABLE nome_tabela;
```

EXCLUSÃO DE UMA VISUALIZAÇÃO (VIEW)

```
DROP VIEW nome_visualizacao;
```

INSERÇÃO DE UMA LINHA

```
INSERT INTO nome_tabela(nome_coluna1, ...)  
VALUES (valor1, ...)
```

ATUALIZAÇÃO DE UMA LINHA

```
UPDATE nome_tabela  
SET nome_coluna = valor1, ...  
WHERE condição;
```

EXCLUSÃO DE UMA LINHA

```
DELETE FROM nome_tabela  
WHERE condição;
```

REFERÊNCIAS

- Chen, P. P. 1976. "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems* 1 (1): 9-36.
- Codd, E. F. 1970. "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, 13 (6): 377-387.
- Date, C.J. and Hugh Darwen. 1997. *A Guide to the SQL Standard*, 4th ed. Reading, MA: Addison-Wesley.
- Masunaga, Yoshifumi. 1990. *Basics of Relational Database*. Tokyo: Ohmsha.
- Database Language: SQL, JIS X3005-1-4, 2002.
- ISO/IEC 9075, Information Technology—Database Languages—SQL, 1992.
- ISO/IEC 9075, Information Technology—Database Languages—SQL, 1995.
- ISO/IEC 9075, 1, 2, 3, 4, Information Technology—Database Languages—SQL, 1999.
- IT Engineers' Skill Standards—Technical Engineers (Database), Information-Technology Promotion Agency, Japan.

ÍNDICE

A

agrupamento, 110, 159
ALL (comando), 159
atomicidade, 153-154
AVG (coluna) função, 98, 99, 110

B

B-tree indexação, 163
backup, 161
banco de dados
definição, 6, 10, 15, 187
eficiência de, 3-4, 15, 19, 146, 174
tipos de, 32-39
uso de, 19-21, 175-182
banco de dados da Internet. Veja sistemas de banco de dados com base na web
banco de dados orientado a objeto (OODB), 203-205
banco de dados primário, 201-202
buffers, 161

C

camada de apresentação, 205
camada de dados, 194-196, 205
camada de lógica, 194-196
campos, 27-28, 30, 34, 35, 48
cardinalidade, 74
chaves
estrangeiras, 44, 48, 72, 101
primárias, 35, 44, 48, 65, 67, 72, 78-79, 101, 103, 115
checkpoints, 161-162
colunas, 34, 84
comandos SQL
ALL, 159
COMMIT, 133, 137, 150, 154, 205
CREATE TABLE, 103, 115-119
CREATE VIEW, 117
DELETE, 104, 116, 118, 119, 159
DROP TABLE/DROP VIEW, 118
GRANT, 159, 168
INSERT, 104, 116, 119, 159
LIKE, 97, 108
ORDER BY, 98
REVOKE, 159, 160, 168
ROLLBACK, 136-137, 150, 153-154, 154, 205

SELECT, 93-97, 98, 105, 106, 113, 119, 159
SET TRANSACTION, 158, 160
UPDATE, 104, 116, 119, 159
comentários, 30-31
COMMIT (comando), 133, 137, 150, 154, 205
compartilhamento de dados, problemas com, 12, 20, 21, 129, 175
condição real, 52, 55-56, 59, 74
consistência, 153, 154-155, 184
consultas. Veja SQL: comandos SQL
consultas full-match, 163
controle de acesso de usuários, 19, 106, 126-129, 141-142, 159-160, 167
controles de execução simultânea
controle com base em trava, 131-137, 155-157, 167
controle otimista, 158
controle timestamp, 158
níveis de isolamento, 158
controles com base em trava, 131-137, 155-157, 167, 175-176, 182
controle otimista, 158
controle timestamp, 158
COUNT funções, 99-100, 110
CREATE TABLE (comando), 103, 115-119
CREATE VIEW (comando), 117
criação de views, 117, 160
curinga, 97, 108

D

dados conflitantes, 13, 17-18, 21, 60, 71, 116, 153, 158
dados corrompidos, 20, 154
dados duplicados, 11, 16, 18, 19, 21, 29
dados inconsistentes, 153, 154, 159, 199-201
dados perdidos, 20, 154
Data Control Language (DCL), 106
Data Definition Language (DDL), 106
Data Manipulation Language (DML), 106
Database Management System (DBMS), 21
DBMS (Database Management System), 21
DCL (Data Control Language), 106
DDL (Data Definition Language), 106
deadlocks, 136, 158

DELETE (comando), 104, 116, 118, 119, 159
direitos de acesso, 19, 106, 126-129, 141-142, 159-160, 167
distribuição horizontal, 197
distribuição vertical, 198
divisão de tabelas. Veja normalização
DML (Data Manipulation Language), 106
DROP TABLE/DROP VIEW (comando), 118
durabilidade, 153, 159-160

E

encapsulamento, 203
entidades, 52-54, 74
entrada de dados, 21, 90-92, 103-104, 106, 116
esquema conceitual, 81
esquema externo, 81
esquema interno, 81
esquemas, 81
Extensible Markup Language (XML), 202
extração de dados, 36-37, 39-47, 90-92, 95-99, 101-102, 106, 180, 202

F

falhas de mídia, 161
falhas no banco de dados, 161
falhas no sistema, 161
forma desnormalizada, 62, 78-79
formas, 62-70, 81-82
funções armazenadas, 196
funções de conjunto, 98, 110
funções de totalização, 98-100, 110-111

G

gatilhos (triggers), 187, 196
gerenciamento de dados de forma não coordenada, 10
gerenciamento de dados uniforme, 19
GRANT (comando), 159, 168
granularidade, 157
granularidade de trava, 157
granularidade fina, 157
granularidade grossa, 157
GROUP BY (cláusula), 110

H

hash função, 167
HAVING (cláusula), 111

HyperText Markup Language (HTML), 194
HyperText Transfer Protocol (HTTP), 178,
180, 194

I

indexação por hash, 163
índices/indexação, 143-147, 162-164,
167
inner join, 115
INSERT (comando), 104, 116, 119, 159
instância, 203
International Organization for Standardization (ISO), 124
International Standard Book Number (ISBN), 45
ISBN (International Standard Book Number), 45
ISO (International Organization for Standardization), 124
isolamento, 153, 155-158

J

Japanese Industrial Standards (JIS), 124
junção de tabelas, 44, 101-102, 114-115

L

leitura fantasma, 158
leitura não reproduzível, 158
leitura suja, 158
LIKE (comando), 97, 108
linguagens de programação, 178, 180,
194, 202
linhas, 34, 84, 116
logs, 148-149

M

MAX (coluna) função, 99-100, 110
mecanismos de recuperação de desastres,
20, 147-150, 161-164, 167
memória. Veja procedimentos armazena-
dos
método de laços aninhados, 165
método rolling forward, 149
método sort merge, 166
métodos de consulta, 93-97, 106, 108,
112-115, 163. Veja também SQL
MIN (coluna) função, 99, 110
modelos de dados, 32-39
modelo de dados hierárquico, 32, 33,
39, 204
modelo de dados relacional, 33-34, 35,
39, 47, 48

modelo entidade-relacionamento (E-R),
50-55, 74-77, 81

N

níveis de isolamento, 158
nomes de usuários, 141
normalização, 60-72, 78-81
null, 30, 108

O

objetos compostos, 203
OODB (object-oriented databases),
203-205
operações
conjunto, 39-42
diferença, 37, 39, 41
divisão, 37, 43, 45
extração de dados, 39-47
intersecção, 37, 39, 41
junção (join), 37, 43, 44, 48, 165
produto cartesiano, 37, 39, 42
projeção, 36, 37, 43, 165
relacionais, 43-47
seleção, 37, 39, 43, 47, 48, 165
união, 37, 39, 40, 48
operações de conjunto, 39-42
operações de diferença, 37, 39, 41
operações de divisão, 37, 43, 45
operações de efetivação (commit) em
duas fases, 199-201, 205
operações de escrita, 130, 133-134, 159
operações de extração de dados, 36-37,
39-47
operações de intersecção, 37, 39, 41
operações de junção (join), 37, 43, 44,
48, 165
operações de leitura, 130, 133, 134, 159
operações de projeção, 36, 37, 43, 165
operações de seleção, 37, 39, 43, 47,
48, 165
operações de união, 37, 39, 40, 48
operações matemáticas. Veja operações
operações produto cartesiano, 37, 39, 42
operações relacionais, 43-47
operadores, 107
operadores de comparação, 107
operadores lógicos, 107
ordenação. Veja funções de totalização;
índices/indexação
ORDER BY (comando), 98
otimização de consulta, 164-167
otimizadores, 167
outer join, 115

P

particionamento de dados, 198-199
partição horizontal, 198
partição vertical, 199
permissões, 19, 141-142, 159-160, 167
permissões de usuários, 19, 106,
126-129, 141-142, 159-160, 167
pesquisa de padrões, 108
primeira forma normal, 62-64, 66, 78-79
problemas no gerenciamento de dados
dados conflitantes, 13, 17-18, 21, 60,
71, 116, 153, 158
dados corrompidos/perdidos, 20, 154
falhas em banco de dados, 161
dificuldade para alterar dados, 13, 14,
17, 18
duplicidade de dados, 11, 16, 18, 19,
21, 29
dados inconsistentes, 153, 154, 159,
199-201
dados compartilhados, 12, 20, 21,
129, 175
processamento com base em custo, 167
processamento com base em regras, 167
processamento de dados, 35-37, 47-48,
130, 159, 167, 182, 195-198
projeto de banco de dados, 19, 26
determinar as condições dos dados, 74
modelo E-R (entidade-relacionamen-
to), 50-55, 74-77, 81
normalização, 60-72, 78-81
passos para, 81, 84

R

READ COMMITTED (transação), 158
READ UNCOMMITTED (transação), 158
recuperação de dados, 20, 147-152,
161-164, 167
recuperação de desastre, 20, 147-150,
161-164, 167
recursos em transações, 155
registros, 27-28, 34, 48, 148-149
relacionamento
conceito de, 54, 74
modelo E-R (entidade-relacionamen-
to), 50-55, 74-77, 81
relacionamento hierárquico, 32, 33,
39, 204
relacionamento muitos-para-muitos,
55, 74, 75, 81
relacionamento um-para-muitos, 55,
75, 81
relacionamento um-para-um, 74, 81
REPEATABLE READ (transação), 158

replicação do banco de dados, 201-202
réplicas, 201-202
réplica somente leitura (read-only), 201
REVOKE (comando), 159, 160, 168
right outer join, 115
ROLLBACK (comando), 136-137, 150,
153-154, 205

S

segunda forma normal, 62, 64, 66-69,
78-79, 82
segurança de dados, 19, 127, 138-142,
159-160, 161-164, 167, 176, 182,
184
SELECT (comando), 93-97, 98, 105, 106,
113, 119, 159
senhas, 141
SERIALIZABLE (transação), 155, 156,
158
servidores, 178-185, 194-197, 205
servidores de aplicações, 182, 195
SET TRANSACTION (comando), 158, 160
sistemas cliente/servidor em três camadas, 194-196, 197, 205
sistemas com base em arquivos, 3, 10,
16, 21, 32
sistemas de bancos de dados com base
na web, 177-182, 194-197
sistemas de bancos de dados distribuídos
distribuição horizontal, 197
distribuição vertical, 198
operações de efetivação (commit) em
duas fases em, 199-201, 205
particionamento de dados, 198-199
replicação em, 201-202
visão geral de, 183-184, 197-199,
205
sistemas resistentes a falhas, 184, 197
sobreposição de dados não-autorizada,
140
SQL (Structured Query Language)
bancos de dados com base na web e,
178-179, 195-196
cláusulas usadas em, 94-97, 106,
110-111, 119
condições, 95-96, 101, 107-109
criação de tabelas, 91-92, 103-105,
106, 115-119
criação de views, 117, 160
curinga, 97, 108
funções de totalização, 98-100,
110-111
GROUP BY (cláusula), 110
HAVING (cláusula), 111

junção de tabelas, 44, 101-102,
114-115
manipulação de dados, 90-92, 100,
106, 116
métodos de consulta, 93-97, 106,
108, 112-115, 163
operadores de comparação, 107
operadores lógicos, 107
otimização de consulta, 164-167
padronização de, 124
pesquisa de padrões, 108
subconsultas, 112-114
visão geral de, 90-92, 106, 116, 124
WHERE (cláusula), 94-97, 106, 110,
119
SQL92/SQL99, 124
stored procedures, 185-188, 196
strings caractere, 84, 108
Structured Query Language. Veja SQL
(Structured Query Language)
subconsultas, 112-114
subconsulta correlata, 113-114
SUM função, 99, 110

T

tabela base, 160
tabelas
base, 160
bidimensionais, 34, 79
conceito de, 34, 39, 48
criação, 57-59, 91-92, 103-105, 106,
115-119
formas para, 62-70, 81-82
junção, 44, 101-102, 114-115
múltiplas, 59
normalização de, 60-72, 78-81
normalizadas, 72, 91
restrições em, 116
views, 117, 160
tabelas normalizadas, 72, 91
tabulação, 34, 47, 57. Veja também
normalização
tags de dados, 202
transações
definição, 126-130
falhas em, 161
operações de leitura/escrita, 130, 133,
134, 159
propriedades de, 153-160
recuperação de desastre e, 149-150
trava compartilhada, 133, 155-156
trava exclusiva, 134-136, 155-156
travamento em duas fases, 156-157
taxa de acesso ao disco, 145

terceira forma normal, 62, 68, 69-70,

78-79, 81, 82

termos de banco de dados, 26-31

U

unique, 30
UPDATE (comando), 104, 116, 119, 159
usuários autorizados, 141, 159

V

valores funcionalmente dependentes, 79
valores transitivamente dependentes, 79

W

WHERE (cláusula), 94-97, 106, 110, 119

X

XML (Extensible Markup Language), 202

SOBRE O AUTOR

Mana Takahashi é formada pela Faculdade de Economia da Universidade de Tóquio, Japão. Ela é uma escritora técnica ativa e publicou alguns livros sobre tópicos como Java, C, XML, Engenharia da Informação e Administração de Sistemas.

ATUALIZAÇÕES

Visite <http://www.novatec.com.br> para atualizações, erratas e outras informações.

CONHEÇA MAIS GUIAS MANGÁ

A série *The Manga Guide* em português é uma co-publicação da Novatec Editora com a editora americana No Starch Press e a editora japonesa Ohmsha, uma das mais antigas e mais respeitadas editoras de livros técnicos e científicos do Japão. Cada título desta coleção de sucesso é produto do trabalho conjunto de um ilustrador, um roteirista e um acadêmico ou um profissional especializado. O resultado é a versão que você tem em suas mãos.

Encontre mais *Guias Mangá* em sua livraria favorita, e saiba mais sobre a série em <http://www.novatec.com.br>.



LIBERE O PODER DOS BANCOS DE DADOS!



A PRINCESA RURUNA E CAIN TÊM UM PROBLEMA: SEU IMPÉRIO DE VENDA DE FRUTAS É UM AMONTOADO DE DADOS CONFLITANTES E DUPLICADOS, E SEPARAR OS MELÕES DAS MAÇÃS E MORANGOS ESTÁ CAUSANDO MUITA DOR DE CABEÇA. MAS O QUE ELES PODEM FAZER?

ORA, CRIAR UM BANCO DE DADOS RELACIONAL, É CLARO, COM A AJUDA DE TICA, A FADA MÁGICA DOS BANCOS DE DADOS. ACOMPANHE NO *GUIA MANGÁ DE BANCOS DE DADOS* COMO TICA ENSINA RURUNA E CAIN A CRIAR UM BANCO DE DADOS PARA GERENCIAR AS VENDAS, MERCADORIAS E EXPORTAÇÕES DE SEU REINO. VOCÊ APRENDERÁ COMO OS BANCOS DE DADOS FUNCIONAM E O SIGNIFICADO DE TERMOS COMO ESQUEMAS, CHAVES, NORMALIZAÇÃO E TRANSAÇÕES.

JUNTO COM RURUNA E CAIN VOCÊ VAI APRENDER COMO:

- EXTRAIRED DADOS DE UM BANCO RELACIONAL UTILIZANDO OPERAÇÕES RELACIONAIS E DE CONJUNTO

- APPLICAR O MODELO ENTIDADE-RELACIONAMENTO PARA REPRESENTAR COM PRECISÃO SEUS PRÓPRIOS DADOS
- CONTROLAR PERMISSÕES DE USUÁRIOS E UTILIZAR BLOQUEIOS PARA EVITAR CONFLITOS E DUPLICIDADE DE DADOS
- UTILIZAR SQL PARA ATUALIZAR OU CONSULTAR DADOS E CRIAR RELATÓRIOS

E VOCÊ AINDA VAI APRENDER OS FUNDAMENTOS DE INDEXAÇÃO, SEGURANÇA, RECUPERAÇÃO DE DESASTRES, REPLICACÃO E MUITO MAIS.

SE A SUA CABEÇA DÁ VOLTAS SÓ DE OUVIR OUTRAS PESSOAS DIZENDO "BANCO DE DADOS", OU SE VOCÊ SIMPLESMENTE SE SENTE PERDIDO NUMA CONFUSÃO DE NÚMEROS E DADOS QUE PARECEM ESTAR FORA DE CONTROLE, APRENDA JUNTO COM RURUNA E CAIN TUDO O QUE ELES PRECISAM SABER NAS PÁGINAS DO *GUIA MANGÁ DE BANCOS DE DADOS*.

