

# SQL Para Leigos

Folha  
de Cola

Este lembrete consiste em diversas tabelas e listas úteis, contendo informações que surgem repetidamente ao trabalhar com o SQL. Em um único lugar, você pode obter uma resposta rápida a um número de diferentes questões que frequentemente surgem durante um esforço de desenvolvimento SQL.

## ***Critérios SQL para Formulários***

Para garantir que as tabelas de banco de dados sejam elaboradas de uma maneira que eles manterão seus dados de forma confiável, você precisa ter a certeza de que eles não estão sujeitos a anomalias de modificação. Ao normalizarmos seus bancos de dados, daremos a você esta garantia. Compare os critérios SQL na seguinte relação com as tabelas em seu banco de dados. Ao fazermos isso, alertaremos você sobre a possibilidade de anomalias, quando descobrir que seu banco de dados não está suficientemente normalizado.

### ✓ Formulário Normal Primário (1NF):

A tabela deve ser bidimensional, com linhas e colunas.

Cada linha deve conter dados que pertençam a um elemento ou uma parte de um elemento.

Cada coluna deve conter dados de um único atributo de um elemento que vem sendo descrito.

Cada célula (a intersecção de uma linha e uma coluna) da tabela deve possuir um único valor.

Todos os registros em uma coluna devem ser do mesmo tipo.

Cada coluna deve possuir um único nome.

Duas linhas não podem ser idênticas.

A ordem das colunas e das linhas não importa.

### ✓ Formulário Normal Secundário (2NF):

A tabela deve estar no formulário normal primário (1NF).

Todos os atributos não importantes (colunas) devem ser dependentes da tecla integral.

### ✓ Formulário Normal Terciário (3NF):

A tabela deve estar no formulário normal secundário (2NF).

A tabela não possui dependências transitivas.

### ✓ Formulário Normal de Domínio Principal (DK/NF):

Toda restrição na tabela, trata-se de uma consequência lógica da definição de teclas e domínios.

***Para Leigos®: A série de livros para iniciantes que mais vende no mundo.***

# SQL Para Leigos

Folha  
de Cola

## Funções de Valor SQL

Estas funções de valor SQL realizam as operações sobre os dados. Tratam-se de todos os tipos de operações que poderiam conceivelmente ser realizadas nos itens de dados, mas são algumas das necessárias com mais frequência.

### ✓ Funções de Valor da String

Função	Efeito
SUBSTRING	Extraí uma sublinha de um recurso de linha
SUBSTRING SIMILAR	Extraí uma sublinha de um recurso de linha, utilizando POSIX-com base em expressões regulares
SUBSTRING_REGEX	Extraí de uma linha a primeira ocorrência de uma expressão padrão regular XQuery e devolver uma ocorrência de uma combinação substring.
TRANSLATE_REGEX	Extraí de uma linha a primeira ou qualquer ocorrência de uma expressão padrão regular XQuery e a substitui ou as substituem por uma linha de reposição XQuery.
UPPER	Converte um caractere de um conjunto em todas as letras maiúsculas
LOWER	Converte um caractere de um conjunto em todas as letras minúsculas
TRIM	Exclui espaços em branco à direita ou a esquerda
TRANSLATE	Transforma um recurso de um conjunto de um caractere definindo-o em outro
CONVERT	Transforma um recurso de um conjunto de um caractere, definindo-o em outro

### ✓ Funções de Valor Numérico

Função	Efeito
POSITION	Retorna a posição inicial de um conjunto de metas dentro de um conjunto de recursos
CHARACTER_LENGTH	Retorna o número de caracteres em um conjunto
OCTET_LENGTH	Retorna o número de octetos (bytes) em um conjunto de caracteres
EXTRACT	Extraí um único domínio de um datetime ou intervalo

### ✓ Funções de Valor de Datetime

Função	Efeito
CURRENT_DATE	Retorna a data atual
CURRENT_TIME(p)	Retorna a hora atual; (p) é a precisão de segundos
CURRENT_TIMESTAMP(p)	Retorna a data atual e a hora atual; (p) é a precisão de segundos

## Funções de Configuração SQL

As funções de configuração SQL dão a você uma resposta rápida as questões que você possa ter sobre as características de seus dados como um todo. Quantas linhas uma tabela possui? Qual é o maior valor na tabela? Qual é o menor valor? Estes são os tipos de questões que as funções de configuração SQL podem responder a você.

COUNT	Retorna o número de linhas na tabela específica
MAX	Retorna o valor máximo que ocorre na tabela específica
MIN	Retorna o valor mínimo que ocorre na tabela específica
SUM	Adiciona valores em uma coluna específica
AVG	Retorna a média de todos os valores na coluna específica

# SQL

PARA

# LEIGOS®

*Tradução da 8ª Edição*

**Allen G. Taylor**



ALTA BOOKS  
EDITORA  
Rio de Janeiro, 2016

## *Sobre o autor*

**Allen G. Taylor** é um veterano de 30 anos da indústria de informática e autor de mais de 30 livros, incluindo *Crystal Reports 2008 For Dummies*, *Database Development For Dummies*, *Access Power Programming with VBA* e *SQL All-in-One For Dummies*. Ele dá palestras sobre bancos de dados, redes, inovação, astronomia e empreendedorismo no mundo todo. Ele também ensina o desenvolvimento de banco de dados por meio de um programa de educação on-line líder no setor. Para as últimas notícias sobre as atividades de Allen, visite [www.DatabaseCentral.Info](http://www.DatabaseCentral.Info). Você pode entrar em contato com Allen em [allen.taylor@ieee.org](mailto:allen.taylor@ieee.org).

## *Dedicatória*

Este livro é dedicado a Walker Taylor, que fará coisas incríveis quando crescer.

## *Agradecimentos do autor*

Primeiro e mais importante, quero agradecer a ajuda de Jim Melton, editor da especificação ISO/ANSI para SQL. Sem seus esforços incansáveis, este livro, e de fato o próprio SQL como um padrão internacional, teriam muito menos valor. Andrew Eisenberg também contribuiu para meu conhecimento do SQL por meio dos seus textos. Quero agradecer a Michael Durthaler pelas sugestões úteis sobre a cobertura dos cursores. Também quero agradecer à minha editora de projetos Pat O'Brien, ao meu editor técnico Mike Chapple e à minha editora de aquisições Kyle Looper pelas contribuições fundamentais para a produção deste livro. Agradeço também à minha agente, Carole McClendon da Waterside Productions, pelo apoio de minha carreira.

# Sumário Resumido

<b><i>Introdução.....</i></b>	<b><i>1</i></b>
<b><i>Parte I: Guia Rápido para o SQL</i></b>	<b><i>3</i></b>
Capítulo 1: Fundamentos de Banco de Dados Relacional.....	5
Capítulo 2: Fundamentos de SQL .....	21
Capítulo 3: Os Componentes do SQL.....	51
<b><i>Parte II: Utilizando SQL para Construir Bancos de Dados.....</i></b>	<b><i>79</i></b>
Capítulo 4: Construindo e Mantendo uma Estrutura Simples de Banco de Dados .....	81
Capítulo 5: Construindo um Banco de Dados Relacional Multitabela.....	105
<b><i>Parte III: Armazenando e Recuperando Dados.....</i></b>	<b><i>137</i></b>
Capítulo 6: Manipulando Dados .....	139
Capítulo 7: Tratando Dados Temporais.....	157
Capítulo 8: Especificando Valores .....	171
Capítulo 9: Utilizando Expressões de Valor SQL Avançadas .....	197
Capítulo 10: Obtendo os Dados Desejados.....	211
Capítulo 11: Usando Operadores Relacionais .....	243
Capítulo 12: Consultas Aninhadas .....	267
Capítulo 13: Consultas Recursivas.....	285
<b><i>Parte IV: Controlando Operações.....</i></b>	<b><i>295</i></b>
Capítulo 14: Segurança de Banco de Dados.....	297
Capítulo 15: Protegendo os Dados .....	313
Capítulo 16: Usando SQL dentro de Aplicativos .....	333
<b><i>Parte V: Levando o SQL para o Mundo Real.....</i></b>	<b><i>347</i></b>
Capítulo 17: Acessando Dados com ODBC e JDBC .....	349
Capítulo 18: Operando em Dados XML com SQL.....	359
<b><i>Parte VI: Tópicos Avançados.....</i></b>	<b><i>381</i></b>
Capítulo 19: Analisando um Conjunto de Dados com Cursores .....	383
Capítulo 20: Adicionando Capacidades Procedurais com Módulos Armazenamento Persistente.....	393
Capítulo 21: Tratando Erros.....	411
Capítulo 22: Gatilhos.....	423

<b><i>Parte VII: A Parte dos Dez.....</i></b>	<b><i>429</i></b>
Capítulo 23: Dez Erros Comuns .....	431
Capítulo 24: Dez Dicas de Recuperação .....	435
<b><i>Apêndice: Palavras reservadas do SQL:2011.....</i></b>	<b><i>439</i></b>
<b><i>Índice.....</i></b>	<b><i>443</i></b>

# Sumário

## ***Introdução* ..... 1**

Sobre este Livro.....	1
Quem Deve Ler este Livro? .....	2
Ícones Usados neste Livro.....	2
De Lá Pra Cá, Daqui Pra Lá .....	2

## ***Parte 1: Guia Rápido para o SQL* ..... 3**

### **Capítulo 1: Fundamentos de Banco de Dados Relacional ..... 5**

Monitorando Coisas .....	6
O Que É um Banco de Dados? .....	7
Tamanho e Complexidade do Banco de Dados.....	7
O Que É um Sistema de Gerenciamento de Banco de Dados? .....	8
Arquivos Simples .....	9
Modelo de Banco de Dados .....	11
Modelo relacional .....	11
Componentes de um banco de dados relacional .....	12
Lidando com relações.....	12
Aprecie a vista.....	14
Esquemas, domínios e restrições.....	16
O modelo de objetos desafiou o modelo relacional.....	18
O modelo objeto-relacional .....	18
Considerações de Projeto de Banco de Dados.....	19

### **Capítulo 2: Fundamentos de SQL ..... 21**

O Que o SQL É e o Que Não É .....	21
Uma História (muito) Pequena.....	23
Instruções SQL.....	24
Palavras Reservadas .....	26
Tipos de Dados.....	26
Numéricos exatos .....	27
Numéricos aproximados .....	29
Strings de caracteres.....	30
Strings binárias .....	32
Booleanos.....	33
Datas e horas .....	33
Intervalos .....	35
Tipo XML.....	35
Tipos ROW .....	38
Tipos de coleção.....	39

Tipos REF .....	41
Tipos definidos pelo usuário .....	41
Resumo dos tipos de dados .....	44
Valores Nulos .....	46
Restrições .....	46
Usando SQL em um Sistema Cliente/Servidor.....	47
O servidor .....	47
O cliente.....	48
Utilizando SQL na Internet ou Intranet .....	49

### **Capítulo 3: Os Componentes do SQL. .... 51**

Linguagem de Definição de Dados.....	52
Quando “apenas faça” não é um bom conselho.....	52
Criando tabelas .....	53
Um quarto com vista.....	55
Resumindo tabelas em esquemas .....	61
Ordenando por catálogo .....	61
Familiarizando-se com instruções DDL .....	62
Linguagem de Manipulação de Dados .....	64
Expressões de valor .....	64
Predicados .....	68
Conectivos lógicos .....	69
Funções de conjunto.....	69
Subconsultas .....	71
Linguagem de Controle de Dados .....	71
Transações.....	71
Usuários e privilégios.....	73
Restrições de integridade referenciais podem colocar seus dados em risco .....	75
Delegando a responsabilidade pela segurança.....	77

## ***Parte II: Utilizando SQL para Construir Bancos de Dados..... 79***

### **Capítulo 4: Construindo e Mantendo uma Estrutura Simples de Banco de Dados ..... 81**

Usando uma Ferramenta RAD para Construir um Banco de Dados Simples....	82
Decidindo o que monitorar.....	82
Criando uma tabela de banco de dados.....	83
Alterando a estrutura da tabela.....	90
Criando um índice .....	92
Excluindo uma tabela.....	94
Construindo POWER com a DDL do SQL .....	95
Utilizando o SQL com o Microsoft Access .....	95
Criando uma tabela .....	97
Criando um índice .....	101
Alterando a estrutura da tabela.....	102



Excluindo uma tabela.....	102
Excluindo um índice.....	103
Considerações de Portabilidade.....	103

## **Capítulo 5: Construindo um Banco de Dados**

### **Relacional Multitabela..... 105**

Projetando um Banco de Dados.....	105
Passo 1: Definindo objetos.....	106
Passo 2: Identificando tabelas e colunas.....	106
Passo 3: Definindo tabelas .....	107
Domínios, conjuntos de caracteres, agrupamentos e conversões ....	111
Acessando rapidamente seu banco de dados com chaves.....	112
Trabalhando com Índices.....	114
Afinal, o que é um índice? .....	115
Por que você deve querer um índice? .....	116
Mantendo um índice .....	117
Mantendo a Integridade dos Dados.....	118
Integridade de entidade.....	118
Integridade de domínio .....	119
Integridade referencial.....	120
Logo quando você achava que era seguro... ..	123
Potenciais áreas problemáticas .....	124
Restrições.....	126
Normalizando o Banco de Dados.....	129
Anomalias de modificação e formas normais .....	129
Primeira forma normal .....	132
Segunda forma normal.....	132
Terceira forma normal .....	134
Forma normal chave-domínio (DK/NF) .....	134
Forma anormal .....	135

## **Parte III: Armazenando e Recuperando Dados..... 137**

### **Capítulo 6: Manipulando Dados ..... 139**

Recuperando Dados .....	139
Criando Views.....	141
A partir de tabelas .....	142
Com uma condição de seleção .....	143
Com um atributo modificado .....	144
Atualizando Views.....	145
Adicionando Novos Dados .....	146
Adicionando dados uma linha de cada vez.....	146
Adicionando dados a colunas selecionadas.....	148
Adicionando um bloco de linhas a uma tabela.....	148
Atualizando os Dados Existentes .....	151
Transferindo Dados .....	154
Excluindo Dados Obsoletos.....	156

**Capítulo 7: Tratando Dados Temporais. . . . . 157**

Entendendo Tempos e Períodos no SQL:2011 .....	158
Trabalhando com Tabelas de Período de Tempo de Aplicativo .....	159
Designando chaves primárias em tabelas de período de tempo de aplicativo .....	162
Aplicando restrições referenciais a tabelas de período de tempo de aplicativo .....	163
Consultando tabelas de período de tempo de aplicativo .....	164
Trabalhando com Tabelas de Sistema Versionadas .....	165
Designando chaves primárias em tabelas de sistema versionadas ...	167
Aplicando restrições referenciais a tabelas de sistema versionadas ...	168
Consultando tabelas de sistema versionadas .....	168
Monitorando Ainda Mais Dados Temporais com Tabelas Bitemporais ...	169

**Capítulo 8: Especificando Valores . . . . . 171**

Valores .....	171
Valores de linha .....	172
Valores literais .....	172
Variáveis .....	174
Variáveis especiais .....	176
Referências de coluna .....	176
Expressões de Valor .....	177
Expressões de valor de string .....	178
Expressões de valor numérico .....	179
Expressões de valor de data e hora .....	179
Expressões de valor de intervalo .....	180
Expressões de valor condicionais .....	180
Funções .....	181
Resumindo: usando funções de agregação .....	181
Funções de valor .....	184

**Capítulo 9: Utilizando Expressões de Valor SQL Avançadas. . . . 197**

Expressões CASE Condicionais .....	197
Usando CASE com condições de pesquisa .....	198
Usando CASE com valores .....	200
Um CASE especial — NULLIF .....	202
Outro CASE especial — COALESCE .....	204
Conversões de Tipo de Dados CAST .....	205
Usando CAST dentro do SQL .....	206
Usando CAST entre o SQL e a linguagem host .....	206
Expressões de Valor de Linha .....	207

**Capítulo 10: Obtendo os Dados Desejados . . . . . 211**

Cláusulas Modificadoras .....	211
Cláusulas FROM .....	213
Cláusulas WHERE .....	213
Predicados de comparação .....	215

BETWEEN.....	215
IN e NOT IN .....	217
LIKE e NOT LIKE.....	218
SIMILAR .....	220
NULL.....	220
ALL, SOME, ANY .....	221
EXISTS.....	224
UNIQUE.....	225
DISTINCT .....	225
OVERLAPS.....	226
MATCH.....	226
Regras de integridade referencial e o predicado MATCH .....	228
Conectivos Lógicos.....	230
AND .....	230
OR.....	231
NOT .....	232
Cláusulas GROUP BY .....	232
Cláusulas HAVING .....	234
Cláusulas ORDER BY .....	235
FETCH Limitado.....	236
Olhando através de uma Janela para Criar um Conjunto	
de Resultados.....	238
Particionando uma janela em segmentos com NTILE .....	239
Navegando dentro de uma janela.....	239
Aninhando funções de janela .....	241
Avaliando grupos de linhas .....	242

## **Capítulo 11: Usando Operadores Relacionais ..... 243**

UNION.....	243
A operação UNION ALL .....	245
A operação CORRESPONDING .....	245
INTERSECT .....	246
EXCEPT .....	248
Operadores de Junção .....	249
Junção básica .....	249
Equijunção .....	251
Junção cruzada .....	253
Junção natural.....	253
Junção de condição.....	254
Junção por nome de coluna .....	254
Junção interna.....	255
Junção externa .....	256
Junção de união .....	259
ON Versus WHERE .....	266

## **Capítulo 12: Consultas Aninhadas ..... 267**

O Que Subconsultas Fazem.....	268
Consultas aninhadas que retornam conjuntos de linhas.....	269

Consultas aninhadas que retornam um valor único .....	272
Os quantificadores ALL, SOME e ANY .....	275
Consultas aninhadas que são um teste de existência .....	277
Outras subconsultas correlacionadas .....	278
UPDATE, DELETE e INSERT .....	282
Recuperando alterações com pipelined DML .....	284

## **Capítulo 13: Consultas Recursivas . . . . . 285**

O Que É Recursão?.....	285
Houston, temos um problema.....	287
Falha não é uma opção .....	287
O Que É uma Consulta Recursiva?.....	288
Onde Se Pode Usar uma Consulta Recursiva?.....	289
Consultando da maneira difícil .....	290
Economizando tempo com recursão.....	291
Onde Mais Se Pode Usar uma Consulta Recursiva? .....	293

## **Parte IV: Controlando Operações..... 295**

### **Capítulo 14: Segurança de Banco de Dados..... 297**

A Linguagem de Controle de Dados SQL.....	298
Níveis de Acesso do Usuário.....	298
O administrador de banco de dados .....	298
Proprietários de objetos de banco de dados .....	299
O público .....	300
Concedendo Privilégios a Usuários.....	300
Papéis.....	301
Inserindo dados .....	302
Visualizando dados .....	302
Modificando dados da tabela .....	303
Excluindo linhas obsoletas de uma tabela.....	304
Referenciando tabelas relacionadas .....	304
Usando domínios.....	305
Fazendo instruções SQL para serem executadas .....	307
Concedendo Privilégios entre Níveis .....	307
Concedendo o Poder de Atribuir Privilégios .....	309
Revogando Privilégios.....	310
Usando GRANT e REVOKE Juntos para Economizar Tempo e Esforço.....	311

### **Capítulo 15: Protegendo os Dados . . . . . 313**

Ameaças à Integridade dos Dados.....	313
Instabilidade de plataforma.....	314
Falha de equipamento .....	314
Acesso concorrente .....	315

Reduzindo a Vulnerabilidade à Corrupção de Dados .....	317
Usando transações SQL.....	318
A transação padrão .....	319
Níveis de isolamento.....	320
A instrução inicial de transação implícita.....	322
SET TRANSACTION .....	323
COMMIT .....	324
ROLLBACK .....	324
Bloqueando objetos de banco de dados.....	324
Fazendo o backup dos seus dados.....	325
Pontos de salvamento e subtransações.....	325
Restrições dentro de Transações .....	327

## **Capítulo 16: Usando SQL dentro de Aplicativos . . . . . 333**

SQL em um Aplicativo .....	333
Prestando atenção ao asterisco.....	334
Pontos fortes e fracos do SQL.....	334
Pontos fortes e fracos das linguagens procedurais .....	335
Problemas ao combinar SQL com uma linguagem procedural .....	335
Conectando SQL a Linguagens Procedurais .....	336
SQL embutido.....	336
Linguagem modular .....	339
Ferramentas RAD orientadas a objetos.....	342
Utilizando o SQL com o Microsoft Access .....	343

## ***Parte V: Levando o SQL para o Mundo Real . . . . . 347***

## **Capítulo 17: Acessando Dados com ODBC e JDBC . . . . . 349**

ODBC .....	350
A interface ODBC .....	350
Componentes do ODBC.....	351
ODBC em um Ambiente Cliente/Servidor .....	352
ODBC e a Internet.....	352
Extensões de servidor .....	353
Extensões de cliente.....	354
ODBC e uma Intranet.....	355
JDBC.....	355

## **Capítulo 18: Operando em Dados XML com SQL . . . . . 359**

Como a XML Se Relaciona com o SQL .....	359
O Tipo de Dados XML .....	360
Quando usar o tipo XML .....	361
Quando não usar o tipo XML.....	362

Mapeando SQL para XML e XML para SQL.....	362
Mapeando conjuntos de caracteres.....	362
Mapeando identificadores .....	363
Mapeando tipos de dados.....	364
Mapeando tabelas .....	364
Tratando valores nulos.....	365
Gerando o esquema XML.....	366
Funções SQL Que Operam sobre Dados XML .....	367
XMLDOCUMENT .....	367
XMLELEMENT .....	367
XMLFOREST.....	368
XMLCONCAT.....	368
XMLAGG .....	369
XMLCOMMENT .....	369
XMLPARSE.....	370
XMLPI.....	370
XMLQUERY .....	370
XMLCAST .....	371
Predicados.....	371
DOCUMENT .....	371
CONTENT.....	372
XMLEXISTS.....	372
VALID.....	372
Transformando Dados XML em Tabelas SQL .....	373
Mapeando Tipos Não Predefinidos de Dados para XML.....	375
Domínio .....	375
UDT Distinct Types.....	376
Linha.....	377
Array .....	378
Multiconjunto .....	379
O Casamento de SQL e XML.....	379

## ***Parte VI: Tópicos Avançados..... 381***

### **Capítulo 19: Analisando um Conjunto de Dados com Cursores . . 383**

Declarando um Cursor .....	384
Expressão de consulta.....	385
Cláusula ORDER BY.....	385
Cláusula de atualização .....	387
Sensibilidade.....	387
Rolagem.....	388
Abrindo um Cursor .....	388
Buscando Dados a partir de uma Única Linha .....	390
Sintaxe.....	390
Direção de um cursor rolável.....	391
Instruções DELETE e UPDATE posicionadas.....	391

Fechando um Cursor .....	392
--------------------------	-----

## **Capítulo 20: Adicionando Capacidades Procedurais com Módulos Armazenamento Persistente ..... 393**

Instruções Compostas .....	393
Atomicidade.....	394
Variáveis .....	395
Cursors .....	396
Condições.....	396
Tratando condições .....	397
Condições que não são tratadas.....	400
Atribuição.....	400
Fluxo das Instruções de Controle .....	400
IF ... THEN ... ELSE ... END IF .....	401
CASE ... END CASE .....	401
LOOP ... ENDLOOP .....	402
LEAVE.....	403
WHILE...DO...END WHILE .....	404
REPEAT...UNTIL...END REPEAT .....	404
FOR...DO...END FOR.....	405
ITERATE.....	405
Procedimentos Armazenados.....	406
Funções Armazenadas .....	407
Privilégios .....	408
Módulos Armazenados .....	409

## **Capítulo 21: Tratando Erros. .... 411**

SQLSTATE .....	411
Cláusula WHENEVER.....	413
Áreas de Diagnóstico.....	414
Área de título de diagnóstico de título .....	414
Área de detalhes de diagnóstico.....	416
Exemplo de violação de restrição .....	418
Adicionando restrições a uma tabela .....	419
Interpretando as informações retornadas por SQLSTATE.....	419
Tratando Exceções .....	420

## **Capítulo 22: Gatilhos ..... 423**

Examinando algumas Aplicações dos Gatilhos.....	423
Criando um Gatilho.....	424
Instrução e gatilhos de linha .....	425
Quando um gatilho é disparado .....	425
A instrução SQL disparada .....	425
Uma definição de exemplo de gatilho.....	426
Disparando uma Sucessão de Gatilhos.....	426

Referenciando Valores Antigos e Novos .....	427
Disparando Vários Gatilhos em uma Única Tabela.....	428

## ***Parte VII: A Parte dos Dez*..... 429**

### **Capítulo 23: Dez Erros Comuns ..... 431**

Supor Que seus Clientes Sabem do Que Eles Precisam.....	431
Ignorar o Escopo de Projeto .....	432
Considerar Apenas os Fatores Técnicos .....	432
Não Solicitar Feedback ao Cliente.....	432
Sempre Utilizar seu Ambiente de Desenvolvimento Favorito .....	433
Usar sua Arquitetura de Sistema Favorita de Maneira Exclusiva .....	433
Projetar Tabelas de Banco de Dados Isoladamente .....	433
Negligenciar Revisões de Projeto.....	434
Ignorar Testes Beta.....	434
Não Documentar seu Processo .....	434

### **Capítulo 24: Dez Dicas de Recuperação ..... 435**

Verifique a Estrutura do Banco de Dados .....	435
Use um Banco de Dados de Teste.....	436
Verifique as Consultas com Junções .....	436
Verifique as Consultas com Subseleções.....	436
Resuma os Dados com GROUP BY .....	436
Observe Restrições da Cláusula GROUP BY.....	437
Use Parênteses com AND, OR e NOT .....	437
Controle os Privilégios de Recuperação.....	437
Faça Backup de seus Bancos de Dados Regularmente.....	438
Trate Condições de Erro Elegantemente.....	438

## ***Apêndice: Palavras reservadas do SQL:2011*..... 439**

## ***Índice*..... 443**



# Introdução

.....

**B**em-vindo ao desenvolvimento de banco de dados usando o SQL, a linguagem de consulta de banco de dados padrão da indústria. Muitas ferramentas do sistema de gerenciamento de banco de dados (SGBD) são executadas em uma variedade de plataformas de hardware. As diferenças entre as ferramentas podem ser grandes, mas todos os produtos sérios têm uma coisa em comum: eles suportam manipulação e acesso a dados por meio do SQL. Se você conhece SQL, você pode construir bancos de dados relacionais e obter informações úteis a partir deles.

## Sobre este Livro

Sistemas de gerenciamento de banco de dados relacional são vitais para muitas organizações. As pessoas muitas vezes acham que criar e manter esses sistemas envolvem atividades extremamente complexas — o domínio dos gurus de banco de dados que têm esclarecimento que vai além daquele dos meros mortais. Este livro acaba com a mística do banco de dados. Neste livro, você irá:

- ✓ Conhecer os fundamentos dos bancos de dados.
- ✓ Entender como um SGBD é estruturado.
- ✓ Descobrir os principais componentes funcionais do SQL.
- ✓ Criar um banco de dados.
- ✓ Proteger um banco de dados contra danos.
- ✓ Operar sobre os dados do banco de dados.
- ✓ Determinar como obter a informação que você quer a partir de um banco de dados.

O objetivo deste livro é ajudá-lo a construir bancos de dados relacionais e obter informações valiosas a partir deles usando SQL. SQL é a linguagem padrão internacional usada para criar e manter bancos de dados relacionais. Esta edição inclui a versão mais recente do padrão, o SQL:2011.

Este livro não informa como projetar um banco de dados (faça isso em *Database Development For Dummies*). Aqui suponho que você ou alguém já criou um projeto válido. Então ilustro como implementar esse projeto usando o SQL. Se suspeitar que você não tem um bom projeto de banco de dados, então — sem dúvida — corrija seu projeto antes de tentar criar o banco de dados. Quanto mais cedo você detectar e corrigir os problemas em um projeto de desenvolvimento, mais baratas serão as correções.

## Quem Deve Ler este Livro?

Se você precisa armazenar ou recuperar dados a partir de um SGBD, você pode fazer um trabalho muito melhor se tiver conhecimento de SQL. Você não precisa ser programador para usar o SQL e você não precisa conhecer linguagens de programação, como Java, C ou BASIC. A sintaxe do SQL é como aquela do idioma inglês.

Se você é programador, você pode incorporar o SQL aos seus programas. O SQL adiciona capacidades poderosas de manipulação e recuperação de dados às linguagens convencionais. Este livro mostra o que você precisa saber para usar a rica variedade de ferramentas e recursos do SQL dentro de seus programas.

## Ícones Usados neste Livro



Dicas economizam uma grande quantidade de tempo e o mantêm longe de problemas.



Preste atenção às informações marcadas por este ícone — você pode precisar delas mais tarde.



Seguir o conselho que este ícone indica pode evitar grandes aflições. Ignore-o por sua conta e risco.



Este ícone alerta para a presença de detalhes técnicos que são interessantes, mas que não são absolutamente essenciais para compreender o tema sendo discutido.

## De Lá Pra Cá, Daqui Pra Lá

Agora vem a parte divertida! Bancos de dados são as melhores ferramentas já inventadas para monitorar as coisas importantes para você. Depois de entender bancos de dados e dominar o SQL para fazer o que precisa ser feito, você adquire um tremendo poder. Colegas de trabalho recorrem a você quando eles precisam de informações cruciais. Gerentes buscam seu conselho. Jovens pedem seu autógrafa. Mas mais importante, você conhece, em um nível muito profundo, como sua organização realmente funciona.

# Parte I



## ***Nesta parte...***

- ✓ Os fundamentos de bancos de dados relacionais
- ✓ Conceitos básicos de SQL
- ✓ Ferramentas fundamentais de banco de dados

## Capítulo 1

# Fundamentos de Banco de Dados Relacional

### *Neste capítulo*

- ▶ Organizando informações
- ▶ Definindo “banco de dados” em termos digitais
- ▶ Decifrando o SGBD
- ▶ Analisando a evolução dos modelos de banco de dados
- ▶ Definindo “banco de dados *relacional*” (você pode relacionar as coisas?)
- ▶ Considerando os desafios do projeto de banco de dados

**S**QL é uma linguagem especificamente projetada com bancos de dados em mente. O SQL permite às pessoas criar bancos de dados, adicionar e modificar dados a estes, e recuperar partes selecionadas dos dados. Desenvolvido na década de 1970 na IBM, o SQL cresceu e avançou ao longo dos anos para tornar-se o padrão da indústria. Ele é regido por um padrão formal mantido pela International Standards Organization (ISO).

Existem vários tipos de bancos de dados, cada um deles segue um modelo diferente de como os dados são organizados no banco de dados.

O SQL foi desenvolvido originalmente para operar em dados nos bancos de dados que seguem o *modelo relacional*. Recentemente, o padrão SQL internacional incorporou parte do *modelo de objetos*, resultando em estruturas híbridas chamadas banco de dados objeto-relacional. Neste capítulo, discutiremos o armazenamento de dados, dedicaremos uma seção à maneira como o modelo relacional se compara com outros principais modelos, e forneceremos uma análise dos recursos importantes dos bancos de dados relacionais.

Antes de falar do SQL, porém, vou definir precisamente o que quero dizer com o termo *banco de dados*. Seu significado mudou, assim como computadores mudaram a forma como as pessoas gravam e mantêm informações.

## Monitorando Coisas

Hoje as pessoas usam computadores para executar muitas tarefas anteriormente realizadas com outras ferramentas. Computadores substituíram máquinas de escrever para criar e modificar documentos. Eles superaram calculadoras eletromecânicas como a melhor maneira de fazer cálculos. Eles também substituíram milhões de pedaços de papel, pastas de arquivos e armários de arquivo como o principal meio de armazenar informações importantes. Comparados com essas ferramentas antigas, é claro, computadores fazem muito mais, muito mais rápido — e com maior precisão. Mas esses benefícios aprimorados têm um custo: usuários de computador não mais têm acesso físico direto aos dados.

Quando os computadores falham ocasionalmente, os funcionários do escritório costumam se perguntar se a informatização realmente melhorou alguma coisa. Antigamente, uma pasta de arquivo em papel só “travava” se você a derrubasse — então você simplesmente se ajoelharia, pegaria os papéis e os colocaria de volta na pasta. Com exceção de terremotos ou outros grandes desastres, armários de arquivo nunca “caíam” e eles nunca forneciam uma mensagem de erro. O travamento de um disco rígido é totalmente outra questão: você não pode “recuperar” bits e bytes perdidos. Falhas mecânicas, elétricas e humanas podem fazer seus dados desaparecer no Grande Além, sem nunca mais retornar.

Tomar as precauções necessárias para se proteger contra perda acidental de dados permite que você comece a tirar vantagem da maior velocidade e precisão que os computadores oferecem.

Se estiver armazenando dados importantes, você tem quatro preocupações principais:

- ✓ O armazenamento dos dados tem de ser rápido e fácil porque é provável que você faça isso frequentemente.
- ✓ O meio de armazenamento deve ser seguro. Você não quer voltar mais tarde e descobrir que alguns (ou todos) dos seus dados desapareceram.
- ✓ Recuperar os dados têm de ser rápido e fácil, independentemente do número de itens que você armazena.
- ✓ Você precisa de uma maneira fácil de separar as informações exatas que você quer *agora*, das toneladas de dados que você *não* quer neste exato momento.

Os bancos de dados mais modernos atendem esses quatro critérios. Se você armazena mais de uma dúzia de itens de dados, você provavelmente quer armazenar esses itens em um banco de dados.

## O Que É um Banco de Dados?

Ultimamente, o termo *banco de dados* é utilizado de maneira vaga, perdendo muito do seu significado original. Para algumas pessoas, um banco de dados é qualquer coleção de itens de dados (agendas de telefone, listas de lavanderia, rolos de pergaminho... qualquer coisa). Outras pessoas definem o termo de forma mais rigorosa.

Neste livro, defino um *banco de dados* como uma coleção autodescritiva dos registros integrados. E, sim, isso implica tecnologia de computador, incluindo linguagens de consulta como SQL.



Um *registro* é uma representação de algum objeto físico ou conceitual. Digamos, por exemplo, que você quer monitorar os clientes de uma empresa. Você atribui um registro a cada cliente. Cada registro tem vários *atributos*, como nome, endereço e número de telefone. Nomes, endereços individuais e etc. são os *dados*.

Um banco de dados consiste em dados e *metadados*. Metadados são os dados que descrevem a estrutura dos dados em um banco de dados. Se você sabe como seus dados são organizados, então você pode recuperá-los. Como o banco de dados contém uma descrição da sua própria estrutura, ele é *autodescritivo*. O banco de dados é *integrado* porque inclui não apenas os itens de dados, mas também as relações entre os itens de dados.

O banco de dados armazena os metadados em uma área chamada *dicionário de dados*, que descreve tabelas, colunas, índices, restrições e outros itens que compõem o banco de dados.

Uma vez que um sistema de arquivos simples (descrito mais adiante neste capítulo) não tem metadados, aplicativos escritos para funcionar com arquivos simples devem conter o equivalente dos metadados como parte do programa aplicativo.

## Tamanho e Complexidade do Banco de Dados

Há bancos de dados de todos os tamanhos, desde coleções simples de alguns registros até sistemas gigantescos contendo milhões de registros. A maioria dos bancos de dados enquadra-se em uma de três categorias, que se baseiam no tamanho do próprio banco de dados, tamanho do equipamento em que ele é executado e tamanho da organização que o mantém:

- ✓ Um **banco de dados pessoal** é projetado para uso por uma única pessoa em um único computador. Esse banco de dados normalmente tem uma estrutura bastante simples e um tamanho relativamente pequeno.

- ✓ Um **banco de dados departamental** ou **banco de dados de grupo de trabalho** é usado pelos membros de um único departamento ou grupo de trabalho em uma organização. Esse tipo de banco de dados costuma ser maior do que um banco de dados pessoal e é necessariamente mais complexo; esse banco de dados deve lidar com múltiplos usuários que tentam acessar os mesmos dados simultaneamente.
- ✓ Um **banco de dados corporativo** pode ser enorme. Bancos de dados corporativos podem modelar o fluxo de informações críticas de todas as grandes organizações.

## O Que É um Sistema de Gerenciamento de Banco de Dados?

Ainda bem que você perguntou. Um *sistema de gerenciamento de banco de dados* (*database management system*, SGBD) é um conjunto de programas utilizados para definir, administrar e processar bancos de dados e seus aplicativos associados. O banco de dados que é gerenciado é, em essência, uma estrutura que você constrói para armazenar dados valiosos. Um SGBD é a ferramenta utilizada para construir essa estrutura e operar nos dados contidos dentro do banco de dados.

Você pode encontrar muitos programas de SGBD no mercado hoje. Alguns são executados em máquinas grandes e poderosas, e outros em computadores pessoais, notebooks e tablets. Uma tendência forte, porém, é que esses produtos funcionem em múltiplas plataformas ou em redes que contêm diferentes classes de máquinas. Uma tendência ainda mais forte é armazenar dados em *data centers* ou até armazená-los em *nuvem*, que poderia ser uma nuvem pública gerenciada por uma grande empresa como a Amazon, Google ou Microsoft, via Internet ou pode ser uma nuvem privada operada pela mesma organização que armazena os dados em sua própria intranet.

Hoje em dia, *nuvem* é um jargão comentando incessantemente em círculos de aficionados por tecnologia. Como as coisas brancas pomposas no céu, ela tem bordas indistintas e parece flutuar em algum lugar lá fora. Na realidade, ela é uma coleção de recursos de computação que pode ser acessada por um navegador, por meio da Internet ou em uma intranet privada. A única coisa que distingue os recursos de computação na nuvem de recursos de computação semelhantes em um data center físico é o fato de que os recursos são acessíveis por meio de um navegador, em vez de um programa aplicativo que acessa diretamente esses recursos.

Um SGBD que é executado em plataformas de múltiplas classes, grandes e pequenas, é chamado *escalonável*.





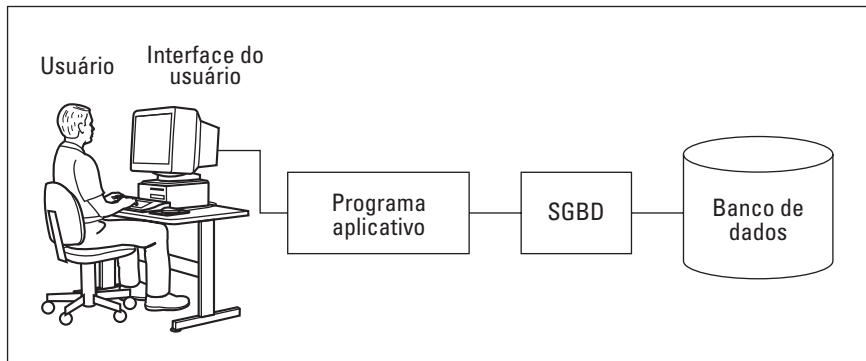
## O valor não está nos dados, mas na estrutura

Anos atrás, uma pessoa inteligente calculou que, se os seres humanos fossem reduzidos aos componentes de átomos de carbono, hidrogênio, oxigênio e nitrogênio (e vestígios de outros), só valeriam 97 centavos. Por mais que seja engraçada, ela é enganosa. As pessoas não são compostas de meras coleções isoladas de átomos. Nossos átomos transformam-se em enzimas, proteínas, hormônios e muitas outras substâncias que custariam

milhões de dólares por grama no mercado farmacêutico. A estrutura precisa dessas combinações de átomos é o que agrega a elas maior valor. Por analogia, a estrutura de banco de dados torna possível interpretar dados aparentemente sem sentido. A estrutura traz à tona padrões, vertentes e tendências nos dados. Dados não estruturados — como átomos não combinados — têm pouco ou nenhum valor.

Seja qual for o tamanho do computador que hospeda o banco de dados — e independentemente de a máquina estar conectada a uma rede — o fluxo de informações entre o banco de dados e o usuário sempre é o mesmo. A Figura 1-1 mostra que o usuário se comunica com o banco de dados por meio do SGBD. O SGBD mascara os detalhes físicos do armazenamento do banco de dados para que o aplicativo só se preocupe com as características lógicas dos dados, não com a forma como os dados são armazenados.

**Figura 1-1:**  
Um diagrama de blocos de um sistema de informação baseado em SGBD.



## Arquivos Simples

Quanto ao local onde os dados estruturados são armazenados, o arquivo básico é o mais simples possível. Não, um arquivo simples não é uma pasta que foi esmagada sob uma pilha de livros. *Arquivos simples* são assim chamados porque eles têm uma estrutura mínima. Se fossem edifícios, eles dificilmente se destacariam do solo. Um arquivo básico é simplesmente uma coleção dos registros de dados, um após outro, em um formato especificado

— os dados, todos os dados e nada mais além dos dados — de fato, uma lista. Em termos de computador, um arquivo de dados não estruturado é um arquivo simples. Como o arquivo não armazena informações estruturais (metadados), sua sobrecarga (coisas no arquivo que não são dados, mas ocupam espaço de armazenamento) é mínima.

Digamos que você quer monitorar os nomes e endereços dos clientes de sua empresa em um sistema de arquivos simples. O sistema pode ter uma estrutura semelhante a esta:

Harold Percival	26262 S. Howards Mill Rd	Westminster	CA92683
Jerry Appel	32323 S. River Lane Rd	Santa Ana	CA92705
Adrian Hansen	232 Glenwood Court	Anaheim	CA92640
John Baker	2222 Lafayette St	Garden Grove	CA92643
Michael Pens	77730 S. New Era Rd	Irvine	CA92715
Bob Michimoto	25252 S. Kelmsley Dr	Stanton	CA92610
Linda Smith	444 S.E. Seventh St	Costa Mesa	CA92635
Robert Funnell	2424 Sheri Court	Anaheim	CA92640
Bill Checkal	9595 Curry Dr	Stanton	CA92610
Jed Style	3535 Randall St	Santa Ana	CA92705

Como você pode ver, o arquivo não contém nada além de dados. Cada campo tem comprimento fixo (o campo nome, por exemplo, sempre tem exatamente 15 caracteres) e nenhuma estrutura separa um campo de outro. A pessoa que criou o banco de dados atribuiu tamanhos e posições aos campos. Qualquer programa usando esse arquivo deve “saber” como cada campo foi definido, porque essa informação não está contida no banco de dados.

Essa baixa sobrecarga significa que operar em arquivos simples pode ser muito rápido. Mas a desvantagem é que programas aplicativos devem incluir a lógica que manipula os dados do arquivo em um nível muito detalhado. O aplicativo deve saber exatamente onde e como o arquivo armazena os dados. Assim, para pequenos sistemas, arquivos simples funcionam bem. Mas quanto maior um sistema é, mais incômodo torna-se um sistema de arquivos simples.



Usar um banco de dados em vez de um sistema de arquivo simples elimina o retrabalho. Embora os próprios arquivos do banco de dados possam ter mais sobrecarga, os aplicativos podem ser mais portáteis entre as várias plataformas de hardware e sistemas operacionais. Um banco de dados também torna mais fácil escrever programas aplicativos porque o programador não precisa conhecer os detalhes físicos de onde e como os dados são armazenados.

Os bancos de dados eliminam o retrabalho porque o SGBD lida com os detalhes da manipulação dos dados. Aplicativos escritos para operar em arquivos simples devem incluir esses detalhes no código do aplicativo. Se múltiplos aplicativos acessam os mesmos dados nos arquivos simples, todos esses aplicativos devem (de maneira redundante) incluir esse código de manipulação de dados. Mas se você usar um SGBD, não será absolutamente necessário incluir esse código nos aplicativos.

Claramente, se um aplicativo baseado em arquivo simples incluir o código de manipulação de dados, que é executado somente em um sistema operacional (SO) específico, a migração do aplicativo para um sistema operacional diferente é uma dor de cabeça esperando acontecer. Você tem de alterar todo o código específico do sistema operacional — e isso é só o começo. Migrar um aplicativo semelhante baseado em SGBD para outro sistema operacional é muito mais simples — menos passos complicados, menos consumo de aspirina.

## Modelo de Banco de Dados

Os primeiros bancos de dados, no começo dos tempos (década de 1950), eram estruturados de acordo com um modelo hierárquico. Eles tinham problemas de redundância e sua rigidez estrutural dificultava a modificação do banco de dados. Logo depois vieram bancos de dados que seguiam o modelo de rede, que procurava eliminar as principais desvantagens do modelo hierárquico. Bancos de dados de rede têm redundância mínima, mas pagam por essa vantagem com complexidade estrutural.

Alguns anos mais tarde, o Dr. E. F. Codd na IBM desenvolveu o modelo *relacional*, que apresentava redundância mínima e uma estrutura fácil de compreender. A linguagem SQL foi desenvolvida para operar em bancos de dados relacionais. Com o tempo, bancos de dados relacionais colocaram bancos de dados hierárquicos e de rede na lata de lixo da história.



Um novo fenômeno é o surgimento dos chamados bancos de dados NoSQL, que não têm a estrutura dos bancos de dados relacionais e não utilizam a linguagem SQL. Não discutiremos bancos de dados NoSQL neste livro.

## Modelo relacional

O Dr. Codd formulou pela primeira vez o modelo de banco de dados relacional em 1970, e esse modelo começou a aparecer em produtos cerca de uma década mais tarde. Ironicamente, a IBM não forneceu o primeiro SGBD relacional. Essa distinção foi para uma pequena start-up, que batizou seu produto de Oracle.

Bancos de dados relacionais quase substituíram completamente os tipos iniciais dos bancos de dados. Isso se deve, em grande parte, ao fato de permitir alterar a estrutura de um banco de dados relacional, sem precisar alterar ou modificar os aplicativos que se baseavam estruturas antigas. Suponha, por exemplo, que você adicione uma ou mais novas colunas a uma tabela de banco de dados. Você não precisa alterar nenhum dos aplicativos escritos anteriormente que processam essa

tabela — a menos, é claro, que altere uma ou mais das colunas que esses aplicativos têm de usar.



Naturalmente, você terá problemas se remover uma coluna que um aplicativo existente precisa utilizar, independentemente do modelo de banco de dados usado. Uma das maneiras mais rápidas de fazer com que um aplicativo de banco de dados trave é solicitar que ele recupere algum dado que ele não contém.

## *Componentes de um banco de dados relacional*

Bancos de dados relacionais obtêm sua flexibilidade porque os dados residem em tabelas que são predominantemente independentes entre si. Você pode adicionar, excluir ou alterar dados em uma tabela sem afetar os dados nas outras mesas, desde que a tabela afetada não seja *pai* de qualquer uma das outras tabelas. (Relações de tabela pai-filho são explicadas no Capítulo 5 e, não, elas não envolvem discutir mesadas durante o jantar.) Nesta seção, veremos em que essas tabelas consistem e como elas se relacionam com as outras partes de um banco de dados relacional.

## *Lidando com relações*



Um banco de dados relacional é constituído por uma ou mais relações. Uma *relação* é uma matriz bidimensional de linhas e colunas, contendo entradas com valores exclusivos e nenhuma linha duplicada. Cada célula da matriz só pode ter um valor e nenhuma linha pode ser idêntica a outra. Se achar isso um pouco difícil de imaginar, eis um exemplo que colocará você em sintonia...

A maioria das pessoas conhece arrays *bidimensionais* de linhas e colunas, na forma de planilhas eletrônicas como o Microsoft Excel. Estatísticas de jogadores de beisebol, como aquelas impressas no verso de um *card* de beisebol, são um exemplo de uma matriz. No cartão de beisebol há colunas para o ano, time, jogos, vezes ao bastão, rebatidas válidas, pontos marcados, resultados das rebatidas, duplos, triplos, *home runs*, bases por bolas, roubadas e média de rebatidas. Uma linha abrange todos os anos em que o jogador jogou nas ligas principais. Você também pode armazenar esses dados em uma relação (uma tabela), que tem a mesma estrutura básica. A Figura 1-2 mostra uma tabela de banco de dados relacional que armazena as estatísticas de um jogador da liga principal. Na prática, essa tabela conteria as estatísticas para um time inteiro — ou talvez toda a liga.

Colunas na matriz são *autoconsistentes*: uma coluna tem o mesmo significado em todas as linhas. Se uma coluna contém o sobrenome de um jogador em uma linha, a coluna deve conter o sobrenome de um

jogador em todas as linhas. A ordem em que linhas e colunas aparecem na matriz não tem nenhuma importância. No que diz respeito ao SGBD, não importa qual coluna é a primeira, qual é a próxima e qual é a última. O mesmo é válido para as linhas. O SGBD processa a tabela da mesma maneira independentemente da organização.

**Figura 1-2:**

Uma tabela mostrando as estatísticas de desempenho de um jogador de beisebol.

Player	Year	Team	Game	At Bat	Hits	Runs	RBI	2B	3B	HR	Walk	Steals	Bat. Avg.
Roberts	1988	Padres	5	9	3	1	0	0	0	0	1	0	.333
Roberts	1989	Padres	117	329	99	81	25	15	8	3	49	21	.301
Roberts	1990	Padres	149	556	172	104	44	36	3	9	55	46	.309

Cada coluna em uma tabela de banco de dados incorpora um único atributo da tabela, quase como aquela no *card* de baseball. O significado da coluna é o mesmo para cada linha da tabela. A tabela pode, por exemplo, conter os nomes, endereços e números de telefone de todos os clientes de uma organização. Cada linha na tabela (também chamada *registro* ou *tupla*) armazena os dados para um cliente individual. Cada coluna contém um único *atributo* — como número, nome, rua, cidade, estado, código postal ou número de telefone do cliente. A Figura 1-3 mostra algumas das linhas e colunas desse tipo de tabela.



As *relações* nesse modelo de banco de dados correspondem a *tabelas* em qualquer banco de dados que se baseia no modelo. Tente dizer isso dez vezes rápido.

**Figura 1-3:**  
Cada linha de banco de dados contém um registro; cada coluna de banco de dados contém um único atributo.

Linha

Coluna

CUSTOMER	CustNo	Company	Addr1	Suite
1	1,221.00	Kauai Dive Shoppe	4-976 Sugarloaf Hwy	
2	1,231.00	Unisco	PO Box Z-547	
3	1,351.00	Sight Diver	1 Neptune Lane	
4	1,354.00	Cayman Divers World Unlimited	PO Box 541	
5	1,356.00	Tom Sawyer Diving Centre	632-1 Third Frydenhoj	
6	1,380.00	Blue Jack Aqua Center	23-738 Paddington Lane	Suite 3
7	1,384.00	VIP Divers Club	32 Main St.	
8	1,510.00	Ocean Paradise	PO Box 8745	
9	1,513.00	Fantastique Aquatica	Z32 999 #12A-77 A.A.	
10	1,551.00	Marmot Divers Club	872 Queen St.	
11	1,560.00	The Depth Charge	15243 Underwater Fwy.	
12	1,563.00	Blue Sports	203 12th Ave. Box 746	
13	1,624.00	Makai SCUBA Club	PO Box 8534	
14	1,645.00	Action Club	PO Box 5451-F	

Record 1 of 55

Field