

25

Acesso a bancos de dados com o JDBC



OBJETIVOS

- Neste capítulo, você aprenderá:
- Conceitos de banco de dados relacional.
- Como utilizar Structured Query Language (SQL) para recuperar dados de um banco de dados, e manipular dados em um banco de dados.
- Como utilizar a API do JDBC™ do pacote `java.sql` para acessar bancos de dados.



25.1 Introdução

- **Banco de dados:**
 - Coleção de dados.
- **DBMS (*database management system*):**
 - Sistema de gerenciamento de banco de dados.
 - Armazenando e organizando dados.
- **SQL:**
 - Linguagem-padrão internacional utilizada com bancos de dados relacionais.
 - Structured Query Language.



25.1 Introdução (*Continuação*)

- **RDBMS (Relational Database Management Systems):**
 - Sistema de gerenciamento de banco de dados relacional.
 - MySQL:
 - Código-fonte aberto.
 - Disponível tanto para Windows como para Linux.
 - `dev.mysql.com/downloads`
- **JDBC:**
 - Estrutura de um banco de dados relacional.
 - Driver JDBC:
 - Permite que aplicações Java se conectem a bancos de dados.
 - Permite que programadores manipulem bancos de dados utilizando o JDBC.



Observação de engenharia de software 25.1

A separação entre API do JDBC e drivers de banco de dados particulares permite aos desenvolvedores alterar o banco de dados subjacente sem modificar o código Java que acessa o banco de dados.



25.2 Bancos de dados relacionais

- **Banco de dados relacional:**
 - **Tabela:**
 - Linhas, colunas.
 - **Chave primária:**
 - Dados únicos.
- **Consultas SQL:**
 - Especificam quais dados selecionar em uma tabela.



	Number	Name	Department	Salary	Location
Linha {	23603	Jones	413	1100	New Jersey
	24568	Kerwin	413	2000	New Jersey
	34589	Larson	642	1800	Los Angeles
	35761	Myers	611	1400	Orlando
	47132	Neumann	413	9000	New Jersey
	78321	Stephens	611	8500	Orlando
	Chave primária		Coluna		

Figura 25.1 | Dados de exemplo da tabela Empl oyee.



Slide 7

EFz1

OK

Edson Furmankiewicz; 21/10/2005

Department	Location
413	New Jersey
611	Orlando
642	Los Angeles

Figura 25.2 | Resultado de selecionar dados distintos de Department e Location a partir da tabela Employee.



25.3 Visão geral de banco de dados relacional: O banco de dados books

- Banco de dados books de exemplo:
 - Quatro tabelas:
 - authors
 - AuthorID, firstName, lastName
 - publishers
 - publisherID, publisherName
 - titles
 - isbn, title, editionNumber, copyright, publisherID, imageFile, price
 - authorISBN
 - authorID, isbn



Coluna	Descrição
authorID	Número de ID do autor no banco de dados. No banco de dados books, essa coluna de inteiros é definida como auto-incrementada . Para cada linha inserida nessa tabela, o valor authorID é automaticamente aumentado por 1 para assegurar que cada linha tenha um authorID único. Essa coluna representa a chave primária da tabela.
firstName	Nome do autor (uma string).
lastName	Sobrenome do autor (uma string).

Figura 25.3 | A tabela authors de books.



authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry

Figura 25.4 | Dados de exemplo da tabela authors.



Coluna	Descrição
publ i sherID	O número de ID do editor no banco de dados. Esse inteiro auto-incrementado é a chave primária da tabela.
publ i sherName	O nome do editor (uma string).

Figura 25.5 | A tabela publ i shers de books.



publ i sherI D	publ i sherName
1	Prentice Hall
2	Prentice Hall PTG

Figura 25.6 | Dados da tabela publ i shers.



Coluna	Descrição
i s b n	ISBN do livro (uma string). A chave primária da tabela. O ISBN é a abreviação de 'International Standard Book Number' — um sistema internacional de numeração padronizado que os editores utilizam para dar a todos os livros um número de identificação único.
t i t l e	Título do livro (uma string).
e d i t i o n N u m b e r	Número de edição do livro (um inteiro).
c o p y r i g h t	Ano de direitos autorais do livro (uma string).
p u b l i s h e r I D	Número de ID do editor (um inteiro). Uma chave estrangeira que relaciona essa tabela com a tabela p u b l i s h e r s.
i m a g e F i l e	Nome do arquivo que contém a imagem da capa do livro (uma string).
p r i c e	Preço de varejo sugerido do livro (um número real). [Nota: Os preços exibidos na Figura 25.8 são apenas para fins de exemplo.]

Figura 25.7 | A tabela t i t l e s de books.



25.3 Visão geral de banco de dados relacional: Banco de dados books (Cont.)

- **Chave estrangeira:**
 - Uma coluna:
 - coincide com a coluna da chave primária em uma outra tabela.
 - Ajuda a manter a *Regra da Integridade Referencial*:
 - todo valor de chave estrangeira deve aparecer como o valor de chave primária da outra tabela.
- **Diagrama de relacionamento de entidade (ER — *entity-relationship*):**
 - Tabelas no banco de dados.
 - Relacionamentos entre as tabelas.



25.3 Visão geral de banco de dados relacional: Banco de dados books (Cont.)

- **Regra de Integridade de Entidade:**
 - A chave primária identifica unicamente cada linha.
 - Cada linha deve ter um valor para cada coluna da chave primária.
 - O valor da chave primária deve ser único na tabela.



Erro comum de programação 25.1

Não fornecer um valor para cada coluna em uma chave primária quebra a Regra de Integridade de Entidade e faz com que o DBMS informe um erro.



Erro comum de programação 25.2

Fornecer o mesmo valor para a chave primária em múltiplas linhas faz com que o DBMS informe um erro.



isbn	title	editionNumber	copy-right	publisherID	Image File	price
0131426443	C How to Program	4	2004	1	chtp4.jpg	85.00
0130384747	C++ How to Program	4	2003	1	cpphtp4.jpg	85.00
0130461342	Java Web Services for Experienced Programmers	1	2003	1	jwsfep1.jpg	54.99
0131483986	Java How to Program	6	2005	1	jhtp6.jpg	85.00
013100252X	The Complete C++ Training Course	4	2003	2	cppctc4.jpg	109.99
0130895601	Advanced Java 2 Platform How to Program	1	2002	1	advjhtp1.jpg	69.95

Figura 25.8 | Dados de exemplo da tabela `titles` de books.



Coluna	Descrição
authorID	O número de ID do autor, uma chave estrangeira para a tabela authors.
i sbn	O ISBN de um livro, uma chave estrangeira para a tabela ti t l es.

Figura 25.9 | A tabela authorISBN de books.



authorID	i_sbn	authorID	i_sbn
1	0130895725	2	0139163050
2	0130895725	3	0130829293
2	0132261197	3	0130284173
2	0130895717	3	0130284181
2	0135289106	4	0130895601

Figura 25.10 | Dados de exemplo da tabela authorI SBN de books.



Erro comum de programação 25.3

Fornecer um valor de chave estrangeira que não aparece como um valor de chave primária em outra tabela quebra a Regra de Integridade Referencial e faz com que o DBMS informe um erro.



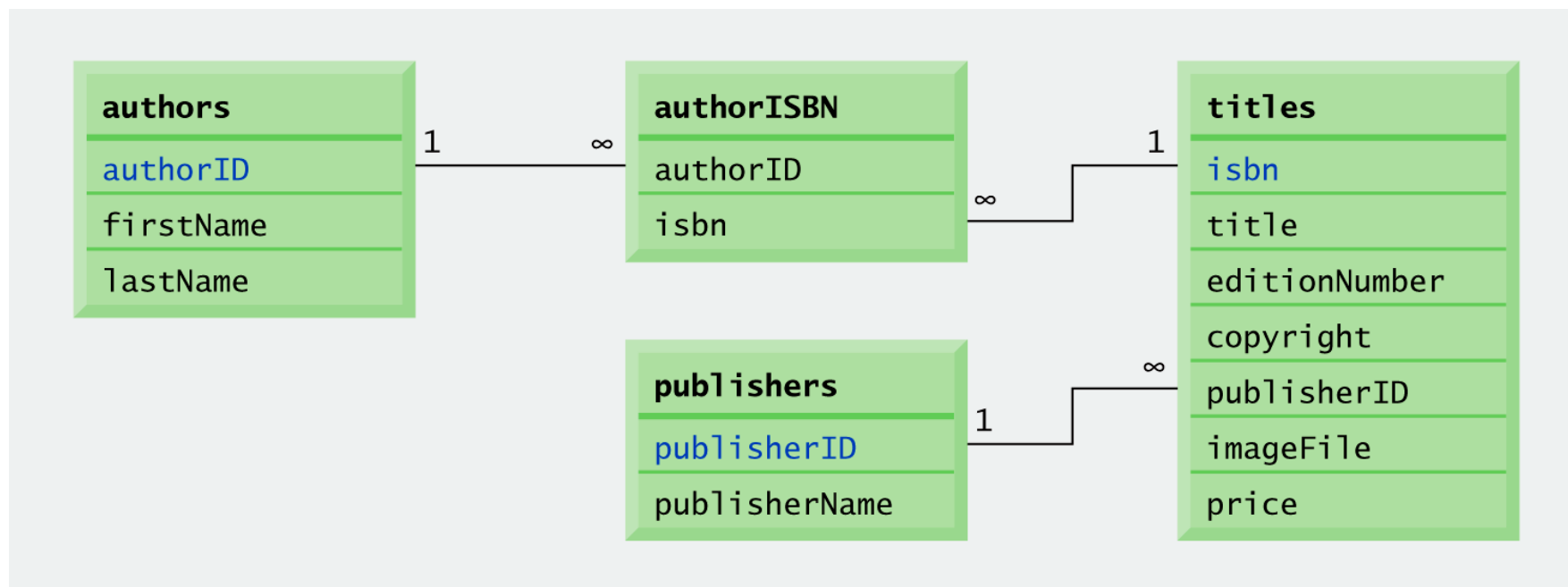


Figura 25.11 | Os relacionamentos de tabela em books.



25.4 SQL

- **Palavras-chave de SQL.**
 - **Consultas e instruções SQL.**



Palavra-chave de SQL	Descrição
SELECT	Recupera dados de uma ou mais tabelas.
FROM	Tabelas envolvidas na consulta. Requeridas em cada SELECT.
WHERE	Critérios de seleção que determinam as linhas a ser recuperadas, excluídas ou atualizadas. Opcional em uma consulta ou uma instrução SQL.
GROUP BY	Critérios para agrupar linhas. Opcional em uma consulta SELECT.
ORDER BY	Critérios para ordenar linhas. Opcional em uma consulta SELECT.
INNER JOIN	Mescla linhas de múltiplas tabelas.
INSERT	Insere linhas em uma tabela especificada.
UPDATE	Atualiza linhas em uma tabela especificada.
DELETE	Exclui linhas de uma tabela especificada.

Figura 25.12 | Palavras-chave de consulta do SQL.



25.4.1 Consulta SELECT básica

- **Formato básico de uma consulta SELECT:**
 - **SELECT** * **FROM** *nomeDaTabela*
 - **SELECT** * **FROM** authors
- **Para selecionar campos específicos de uma tabela, como authorID e lastName:**
 - **SELECT** authorID, lastName **FROM** authors



authorID	lastName
1	Deitel
2	Deitel
3	Nieto
4	Santry

Figura 25.13 | Dados authorID e lastName de exemplo da tabela authors.



Observação de engenharia de software 25.2

Para a maioria das consultas, o asterisco (*) não deve ser utilizado para especificar nomes de coluna. Em geral, os programadores processam resultados sabendo antecipadamente a ordem das colunas no resultado — por exemplo, selecionar authorID e lastName da tabela authors assegura que as colunas aparecerão no resultado com authorID como a primeira coluna e lastName como a segunda coluna. Em geral, os programas processam colunas de resultados especificando o número de coluna no resultado (iniciando do número 1 da primeira coluna). Selecionar colunas por nome também evita retornar colunas desnecessárias e protege contra alterações na ordem real das colunas na(s) tabela(s).



Erro comum de programação 25.4

Se um programador assume que as colunas são sempre retornadas na mesma ordem de uma consulta que utiliza o asterisco (*), o programa pode processar o resultado incorretamente. Se a ordem de coluna na(s) tabela(s) mudar ou se colunas adicionais forem adicionadas posteriormente, a ordem das colunas no resultado mudaria de maneira correspondente.



25.4.2 Cláusula WHERE

- **Especifica os critérios de seleção:**
 - **SELECT** *nomeDaColuna1, nomeDaColuna2, ...* **FROM** *nomeDaTabela* **WHERE** *critérios*
 - **SELECT** title, editionNumber, copyright
 - **FROM** titles
 - **WHERE** copyright > 2002



ti t l e	edi ti onNumber	copyri ght
The Complete C++ Training Course	4	2003
Java How to Program	5	2003
C How to Program	4	2004
Internet and World Wide Web How to Program	3	2004
Java How to Program	6	2005
C# How to Program	1	2003

Figura 25.14 | Amostragem de títulos com direitos autorais posteriores a 2002 da tabela ti t l es.



25.4.3 Cláusula ORDER BY

- Cláusula ORDER BY opcional:
 - **SELECT** *nomeDaColuna1, nomeDaColuna2, ...* **FROM** *nomeDaTabela* **ORDER BY** *coluna* **ASC**
 - **SELECT** authorID, firstName, lastName
 - **FROM** authors
 - **ORDER BY** lastName **ASC**
 - **SELECT** *nomeDaColuna1, nomeDaColuna2, ...* **FROM** *nomeDaTabela* **ORDER BY** *coluna* **DESC**
 - **SELECT** authorID, firstName, lastName
 - **FROM** authors
 - **ORDER BY** lastName **DESC**



authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry

Figura 25.17 | Dados de exemplo da tabela authors em ordem crescente por lastName.



authorID	fi rstName	l astName
4	Sean	Santry
3	Tem	Nieto
1	Harvey	Deitel
2	Paul	Deitel

Figura 25.18 | Dados de exemplo da tabela authors em ordem decrescente por l astName.



25.4.5 Instrução INSERT

- **Inserir uma linha em uma tabela:**
 - **INSERT INTO** *nomeDaTabela* (*nomeDaColuna1*, ... , *nomeDaColunaN*)
 - **VALUES** (*valor1*, ... , *valorN*)
 - **INSERT INTO** authors (first_name, last_name)
 - **VALUES** (' Sue' , ' Smith')



authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry
5	Sue	Smith

Figura 25.22 | Dados de exemplo da tabela Authors depois de uma operação INSERT.



Erro comum de programação 25.6

É um erro especificar um valor para uma coluna auto-incrementada.



Erro comum de programação 25.7

O SQL utiliza o caractere aspas simples (') como um delimitador de strings. Para especificar uma string que contém aspas simples (por exemplo, O'Malley) em uma instrução SQL, a string deve ter duas aspas simples na posição em que o caractere aspas simples aparece na string (por exemplo, ' O' ' Mal l ey'). O primeiro dos dois caracteres de aspas simples atua como um caractere de escape para o segundo. Não 'escapar' caracteres aspas simples em uma string que seja parte de uma instrução SQL é um erro de sintaxe de SQL.



25.4.6 Instrução UPDATE

- **Modifica os dados em uma tabela:**
 - **UPDATE** *nomeDaTabela*
 - **SET** *nomeDaColuna1 = valor1, ... , nomeDaColunaN = valorN*
 - **WHERE** *critérios*
 - **UPDATE** authors
 - **SET** lastName = ' Jones'
 - **WHERE** lastName = ' Smith' **AND** firstName = ' Sue'



authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry
5	Sue	Jones

Figura 25.23 | Dados de exemplo da tabela authors depois de uma operação UPDATE.



25.4.7 Instrução DELETE

- **Remove dados de uma tabela:**
 - **DELETE FROM** *nomeDaTabela* **WHERE** *critérios*
 - **DELETE FROM** authors
 - **WHERE** *lastName* = ' Jones' **AND** *firstName* = ' Sue'



authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tem	Nieto
4	Sean	Santry

Figura 25.24 | Dados de exemplo da tabela authors depois de uma operação DELETE.



25.5 Instruções para instalar o MySQL e o MySQL Conector/J

- **Instalar o MySQL:**

- Acesse o site dev.mysql.com/downloads e adquira a versão 5.1 para a plataforma que desejar
- Instale o programa
- Siga as instruções.

- **Instalar o MySQL Connector/J:**

- Acesse <http://dev.mysql.com/downloads/connector/j/5.1.html> e adquira o arquivo para sua plataforma
- Abra o arquivo e extraia seu conteúdo



25.6 Instruções para a configuração da conta de usuário do MySQL

- **Configure uma conta de usuário:**

- **Inicie o servidor do banco de dados executando o script:** `mysql d`

- **Inicie o monitor MySQL executando o comando:**
`~bin>mysql -h localhost -u root`

- **Crie uma conta:**

```
mysql > USE mysql ;
```

```
mysql > INSERT INTO user SET Host='localhost' ,
```

```
User='jhtp6' , Password=PASSWORD('jhtp6') ,
```

```
Select_priv='Y' , Insert_priv='Y' , Update_priv='Y' ,
```

```
Delete_priv='Y' , Create_priv='Y' , Drop_priv='Y' ,
```

```
References_priv='Y' , Execute_priv='Y' ,
```

```
ssl_cipher='', x509_issuer='', x509_subject=''
```

```
mysql > FLUSH PRIVILEGES;
```



25.7 Criando banco de dados books no MySQL

- **Crie o banco de dados books:**
 - Abra o prompt de comando e mude para o diretório de instalação do MySQL e entre na pasta bin
 - Inicie o banco de dados executando o comando `mysql d`
 - Copie o script de SQL `books.sql` para o diretório `bin`
 - Abra um outro prompt de comando e mude para o diretório `bin`
 - Crie o banco de dados `books` executando o comando `mysql -h localhost -u jhttp6 -p < books.sql`



25.8 Manipulando bancos de dados com o JDBC

- **Conecte-se a um banco de dados.**
- **Consulte o banco de dados.**



25.8.1 Consultando e conectando-se a um banco de dados

- **DisplayAuthors:**
 - **Recupera toda a tabela authors.**
 - **Exibe os dados no fluxo de saída-padrão.**
 - **O exemplo ilustra como:**
 - **conectar-se ao banco de dados;**
 - **consultar o banco de dados; e**
 - **processar o resultado.**



Resumo

```

1 // Fig. 25.25: DisplayAuthors.java
2 // Exibindo o conteúdo da tabela authors.
3 import java.sql.Connection;
4 import java.sql.Statement;
5 import java.sql.DriverManager;
6 import java.sql.ResultSet;
7 import java.sql.ResultSetMetaData;
8 import java.sql.SQLException;
9

```

Importa as classes e interfaces JDBC do pacote `java.sql`

```

10 public class DisplayAuthors
11 {

```

```

12 // nome do driver JDBC e URL do banco de dados

```

```

13 static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";

```

```

14 static final String DATABASE_URL = "jdbc:mysql://localhost/books";

```

```

15

```

```

16 // carrega o aplicativo

```

```

17 public static void main( String args[] )

```

```

18 {

```

```

19     Connection connection = null; // gerencia a conexão

```

```

20     Statement statement = null; // instrução de consulta

```

```

21

```

```

22 // conecta-se ao banco de dados books e o consulta

```

```

23 try

```

```

24 {

```

```

25     Class.forName( JDBC_DRIVER ); // carrega classe de driver do banco de dados

```

```

26

```

```

27 // estabelece conexão com o banco de dados

```

```

28     connection =

```

```

29         DriverManager.getConnection( DATABASE_URL, "jhttp6", "jhttp6" );

```

```

30

```

Declara uma constante **String** que especifica o nome da classe do driver JDBC

Declara uma constante **String** que especifica o URL do banco de dados

Carrega a definição de classe para o driver do banco de dados.

Declara e inicializa uma referência **Connection** chamada **connection**.

Resumo

```

31 // cria Statement para consultar banco de dados
32 statement = connection.createStatement();
33
34 // consulta banco de dados
35 ResultSet resultSet = statement.executeQuery(
36     "SELECT authorID, firstName, lastName FROM authors" );
37
38 // processa resultados da consulta
39 ResultSetMetaData metaData = resultSet.getMetaData();
40 int numberOfColumns = metaData.getColumnCount();
41 System.out.println( "Authors Table of Books Database: " );
42
43 for ( int i = 1; i <= numberOfColumns; i++ )
44     System.out.printf( "%-8s\t", metaData.columnName( i ) );
45 System.out.println();
46
47 while ( resultSet.next() )
48 {
49     for ( int i = 1; i <= numberOfColumns; i++ )
50         System.out.printf( "%-8s\t", resultSet.getObject( i ) );
51     System.out.println();
52 } // fim do while
53 } // fim do try
54 catch ( SQLException sqlException )
55 {
56     sqlException.printStackTrace();
57     System.exit( 1 );
58 } // fim do catch

```

Utiliza o método **executeQuery** do objeto **Statement** para executar uma consulta que seleciona todas as informações sobre autores na tabela **authors**

Obtém o nome de coluna utilizando o método **columnName**

Extraí o conteúdo de uma das colunas na linha atual

Captura a **SQLException**, que é lançada se a execução da consulta ou o processo do **ResultSet** falhar

s. jav



```

59 catch ( ClassNotFoundException classNotFound )
60 {
61     classNotFound.printStackTrace();
62     System.exit( 1 );
63 } // fim do catch
64 finally // assegura que a instrução e conexão são fechadas
65 {
66     try
67     {
68         statement.close();
69         connection.close();
70     } // fim do try
71     catch ( Exception exception )
72     {
73         exception.printStackTrace();
74         System.exit( 1 );
75     } // fim do catch
76 } // fim do finally
77 } // fim do main
78 } // fim da classe DisplayAuthors

```

A
ClassNotFoundException
é lançada se o carregador de
classe não puder localizar a
classe driver

Fecha a **Statement** e a
Connection de banco
de dados.

(3 de 3)

Linha 69

Linhas 68-69

Saída do programa

```

Authors Table of Books Database:
authorID      firstName      lastName
1             Harvey        Deitel
2             Paul         Deitel
3             Tom         Nieto
4             Sean        Santry

```



Observação de engenharia de software 25.4

A maioria dos fornecedores de banco de dados importantes fornece seus próprios drivers de banco de dados de JDBC e muitos fornecedores independentes também fornecem drivers JDBC. Para obter informações adicionais sobre drivers JDBC, visite o site Web JDBC da Sun Microsystems, `servl et. j ava. sun. com/products/j dbc/dri vers`.



RDBMS	Nome de driver JDBC	Formato de URL de banco de dados
MySQL	com.mysql.jdbc.Driver	jdbc:mysql:// <i>nomeDoHost</i> / <i>nomeDoBancoDeDados</i>
ORACLE	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@ <i>nomeDoHost</i> : <i>númeroDaPorta</i> : <i>nomeDoBancoDeDados</i>
DB2	COM.ibm.db2.jdbc.net.DB2Driver	jdbc:db2: <i>nomeoHost</i> : <i>númeroDaPorta</i> / <i>nomeDoBancoDeDados</i>
Sybase	com.sybase.jdbc.SybDriver	jdbc:sybase:Tds: <i>nomeDoHost</i> : <i>númeroDaPorta</i> / <i>nomeDoBancoDeDados</i>

Figura 25.27 | Nomes de driver JDBC e URL de bancos de dados populares.



Observação de engenharia de software 25.6

A maioria dos sistemas de gerenciamento de bancos de dados requer que o usuário efetue logon antes de acessar o conteúdo do banco de dados. O método DriverManager.getConnection é sobrecarregado com as versões que permitem ao programa fornecer o nome de usuário e a senha para ganhar acesso.



Observação de engenharia de software 25.7

Os metadados permitem aos programas processar o conteúdo de ResultSet dinamicamente quando as informações detalhadas sobre o ResultSet não forem conhecidas com antecedência.



Erro comum de programação 25.8

Inicialmente, um cursor ResultSet é posicionado antes da primeira linha. Tentar acessar o conteúdo de um ResultSet antes de posicionar o cursor ResultSet na primeira linha com o método next produz uma SQLException.



Dica de desempenho 25.1

Se uma consulta especificar as colunas exatas a selecionar do banco de dados, o ResultSet conterá as colunas na ordem especificada. Nesse caso, utilizar o número de coluna para obter o valor da coluna é mais eficiente que utilizar o nome de coluna. O número de coluna fornece acesso direto à coluna especificada. Utilizar o nome de coluna requer uma pesquisa linear dos nomes de coluna para localizar a coluna apropriada.



Erro comum de programação 25.9

Especificar número de coluna 0 ao obter valores de um ResultSet produz uma SQLException.



Erro comum de programação 25.10

Tentar manipular um ResultSet depois de fechar Statement que criou o ResultSet causa uma SQLException. O programa descarta ResultSet quando Statement correspondente é fechado.



Observação de engenharia de software 25.8

Todo objeto Statement pode abrir apenas um objeto ResultSet por vez. Quando um Statement retorna um novo ResultSet, Statement fecha o ResultSet anterior. Para utilizar múltiplos ResultSets paralelamente, objetos Statement separados devem retornar ResultSets.



15.10 Interface RowSet

- **Interface RowSet:**
 - **Configura a conexão de banco de dados automaticamente.**
 - **Prepara as instruções de consulta automaticamente.**
 - **Fornece os métodos *set* para especificar as propriedades necessárias para estabelecer uma conexão.**
 - **Parte do pacote `javax.sql`.**
- **Dois tipos de RowSet:**
 - **RowSet conectado: conecta a banco de dados uma vez e permanece conectado.**
 - **RowSet desconectado: conecta a banco de dados, executa uma consulta e, então, fecha a conexão.**



15.10 Interface RowSet (Continuação)

- Pacote `javax.sql.rowset`:
 - `JdbcRowSet`:
 - `RowSet` conectado.
 - Empacotador de um `ResultSet`.
 - Rolável e atualizável por padrão.
 - `CachedRowSet`:
 - `RowSet` desconectado.
 - Armazena os dados em cache do `ResultSet` na memória.
 - Rolável e atualizável por padrão.
 - Serializável:
 - Pode ser passada entre aplicações Java.
 - Limitação:
 - A quantidade de dados que pode ser armazenada na memória é limitada.



Resumo

JdbcRowSetTest.java

(1 de 3)

Linha 27

Linha 17

Linha 28

Linha 29

Linha 30

Linha 31

```

1 // Fig. 25.32: JdbcRowSetTest.java
2 // Exibindo o conteúdo da tabela authors com JdbcRowSet.
3 import java.sql.ResultSetMetaData;
4 import java.sql.SQLException;
5 import javax.sql.rowset.JdbcRowSet;
6 import com.sun.rowset.JdbcRowSetImpl; // Implementação JdbcRowSet da Sun
7
8 public class JdbcRowSetTest
9 {
10     // nome do driver JDBC e URL do banco de dados
11     static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
12     static final String DATABASE_URL = "jdbc:mysql://localhost/books";
13     static final String USERNAME = "jhtp6";
14     static final String PASSWORD = "jhtp6";
15
16     // construtor conecta-se ao banco de dados, consulta banco de dados, processa
17     // resultados e os exibe na janela
18     public JdbcRowSetTest()
19     {
20         // conecta-se ao banco de dados books e o consulta
21         try
22         {
23             Class.forName( JDBC_DRIVER ); // carrega classe de driver do banco de dados
24
25             // especifica propriedades de JdbcRowSet
26             JdbcRowSet rowSet = new JdbcRowSetImpl();
27             rowSet.setUrl( DATABASE_URL ); // configura o URL de banco de dados
28             rowSet.setUsername( USERNAME ); // configura o nome de usuário
29             rowSet.setPassword( PASSWORD ); // configura a senha
30             rowSet.setCommand( "SELECT * FROM authors" ); // configura a consulta
31             rowSet.execute(); // executa a consulta
32

```



Resumo

JdbcRowSetTest.java

(2 de 3)

```
33      // processa resultados da consulta
34      ResultSet metaData = rowSet.getMetaData();
35      int numberOfColumns = metaData.getColumnCount();
36      System.out.println( "Authors Table of Books Database: " );
37
38      // exibe o cabeçalho rowset
39      for ( int i = 1; i <= numberOfColumns; i++ )
40          System.out.printf( "%-8s\t", metaData.getColumnName( i ) );
41      System.out.println();
42
43      // exibe cada linha
44      while ( rowSet.next() )
45      {
46          for ( int i = 1; i <= numberOfColumns; i++ )
47              System.out.printf( "%-8s\t", rowSet.getObject( i ) );
48          System.out.println();
49      } // fim do while
50  } // fim do try
51  catch ( SQLException sqlException )
52  {
53      sqlException.printStackTrace();
54      System.exit( 1 );
55  } // fim do catch
56  catch ( ClassNotFoundException classNotFound )
57  {
58      classNotFound.printStackTrace();
59      System.exit( 1 );
60  } // fim do catch
61  } // fim do construtor DisplayAuthors
62
```



Resumo

JdbcRowSetTest.java

(3 de 3)

Saída do
programa

```
63 // carrega o aplicativo
64 public static void main( String args[] )
65 {
66     JdbcRowSetTest window = new JdbcRowSetTest();
67 } // fim do main
68 } // fim da classe JdbcRowSetTest
```

Authors Table of Books Database:

authorID	firstName	lastName
1	Harvey	Deitel
2	Paul	Deitel
3	Tim	Nieto
4	Sean	Santry

