

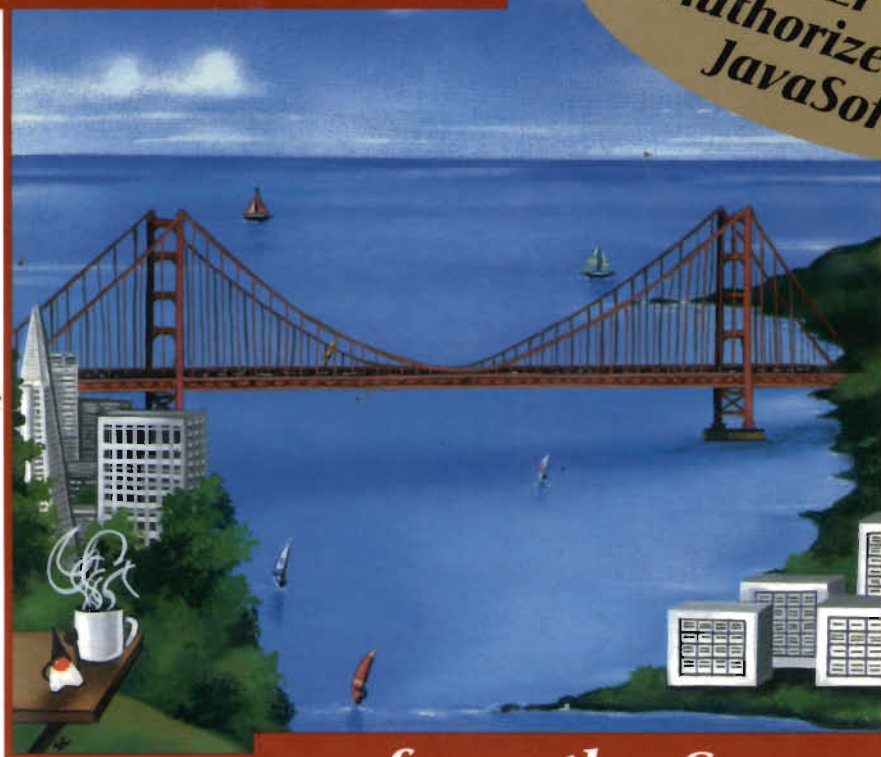
Graham Hamilton • Rick Cattell • Maydene Fisher

JDBC™ Database Access with Java™

A Tutorial and Annotated Reference

The Java Series

*The ONLY Books
Authorized by
JavaSoft*



... from the Source™



JDBC™ Database Access with Java™

A Tutorial and Annotated Reference

**Graham Hamilton
Rick Cattell
Maydene Fisher**



ADDISON-WESLEY

An imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Harlow, England • Menlo Park, California
Berkeley, California • Don Mills, Ontario • Sydney
Bonn • Amsterdam • Tokyo • Mexico City

Contents

Acknowledgments	v
-----------------------	---

Part One

1 Introduction	3
1.1 Contents of the Book	3
1.2 What Is JDBC ?	5
1.2.1 What Does JDBC Do?	6
1.2.2 A Low-level API and a Base for Higher-level APIs	6
1.2.3 JDBC versus ODBC and other APIs	7
1.2.4 Two-tier and Three-tier Models	8
1.2.5 SQL Conformance	10
1.2.6 JDBC Products	12
1.2.7 JavaSoft Framework	12
1.2.8 JDBC Driver Types	12
1.2.9 Obtaining JDBC Drivers	14
1.2.10 Other Products	15
1.3 Java Overview	15
1.3.1 Java Is Portable	15
1.3.2 Java Is Object-Oriented	16
1.3.3 Java Makes It Easy to Write Correct Code	20
1.3.4 Java Includes a Library of Classes and Interfaces	21
1.3.5 Java Is Extensible	22
1.3.6 Java Is Secure	23
1.3.7 Java Performs Well	23
1.3.8 Java Scales Well	24
1.3.9 Java Is Multithreaded	24
1.4 Relational Database Overview	24
1.4.1 Integrity Rules	25
1.4.2 SELECT Statements	26
1.4.3 WHERE Clauses	27

1.4.4	Joins	28
1.4.5	Common SQL Commands	29
1.4.6	Result Sets and Cursors	30
1.4.7	Transactions	30
1.4.8	Stored Procedures	31
1.4.9	Metadata	32
2	Basic Tutorial	33
2.1	Getting Started	33
2.2	Setting Up a Database	34
2.3	Establishing a Connection	35
2.3.1	Loading Drivers	35
2.3.2	Making the Connection	36
2.4	Setting Up Tables	37
2.4.1	Creating a Table	37
2.4.2	Creating JDBC Statements	40
2.4.3	Executing Statements	40
2.4.4	Entering Data into a Table	41
2.5	Getting Data from a Table	42
2.6	Retrieving Values from Result Sets	44
2.6.1	Using the Method next	44
2.6.2	Using the getXXX Methods	44
2.6.3	Using the Method getString	46
2.7	Updating Tables	47
2.8	Milestone: The Basics of JDBC	49
2.9	Using Prepared Statements	49
2.9.1	When to Use a PreparedStatement Object	49
2.9.2	Creating a PreparedStatement Object	50
2.9.3	Supplying Values for PreparedStatement Parameters	50
2.9.4	Using a Loop to Set Values	52
2.9.5	Return Values for the Method executeUpdate	53
2.10	Using Joins	53
2.11	Using Transactions	55
2.11.1	Disabling Auto-commit Mode	56
2.11.2	Committing a Transaction	56
2.11.3	Using Transactions to Preserve Data Integrity	57
2.11.4	When to Call the Method rollback	58

2.12	Stored Procedures	59
2.12.1	SQL Statements for Creating a Stored Procedure	59
2.12.2	Calling a Stored Procedure from JDBC	60
2.13	Creating Complete JDBC Applications	61
2.13.1	Putting Code in a Class Definition	62
2.13.2	Importing Classes to Make Them Visible	62
2.13.3	Using the main Method	63
2.13.4	Using try and catch Blocks	63
2.13.5	Retrieving Exceptions	63
2.13.6	Retrieving Warnings	65
2.14	Running the Sample Applications	67
2.15	Sample Code	68
2.15.1	Sample Code 1 and 2	69
2.15.2	Sample Code 3 and 4	73
2.15.3	Sample Code 5	77
2.15.4	Sample Code 6	78
2.16	Creating an Applet from an Application	81
2.16.1	Writing Applet Code	82
2.16.2	Running an Applet	84
2.16.3	Sample Code 7 and 8	85
3	Metadata Tutorial	89
3.1	Getting Information about a Result Set	89
3.1.1	Using the Method getColumnCount	90
3.1.2	Sample Code 9	90
3.1.3	Using Other ResultSetMetaData Methods	92
3.1.4	Getting Column Type Information	93
3.1.5	Sample Code 10 and 11	94
3.1.6	Sample Code 12	97
3.1.7	Getting Other Information	100
3.2	Getting Information about a Database or Database System	101
3.2.1	Categories of DatabaseMetaData Methods	101
3.2.2	Methods that Return a String	101
3.2.3	Methods that Return an int	103
3.2.4	Methods that Return a boolean	104
3.2.5	Methods that Return a ResultSet Object	104
3.2.6	Sample Code 13	104
3.2.7	Getting Information about DBMS Types	106

3.2.8	Sample Code 14.	106
3.2.9	Getting Information about Primary and Foreign Keys	110
3.2.10	Sample Code 15.	111
3.2.11	Sample Code 16.	114
3.3	A Generic Application	116
3.3.1	Sample Code 17 and 18.	117

Part Two

4	CallableStatement	129
4.1	CallableStatement Overview	129
4.1.1	Creating a CallableStatement Object	130
4.1.2	IN Parameters	130
4.1.3	OUT Parameters	131
4.1.4	Numbering of Parameters	132
4.1.5	INOUT Parameters	132
4.1.6	Retrieve OUT Parameters after Results	133
4.1.7	Retrieving NULL Values as OUT Parameters	133
4.2	CallableStatement Definition	134
4.3	CallableStatement Methods	134
5	Connection	143
5.1	Connection Overview	143
5.1.1	Opening a Connection.	143
5.1.2	URLs in General Use	144
5.1.3	JDBC URLs.	145
5.1.4	The <code>odbc</code> Subprotocol	146
5.1.5	Registering Subprotocols	147
5.1.6	Sending SQL Statements.	147
5.1.7	Transactions.	148
5.1.8	Transaction Isolation Levels	149
5.1.9	Using the Method <code>close</code> to Free DBMS Resources	150
5.2	Connection Interface Definition	150
5.3	Connection Methods	151
5.4	Connection Fields	161
6	DatabaseMetaData	163
6.1	DatabaseMetaData Overview	163
6.1.1	Creating a DatabaseMetaData Object	163
6.1.2	ResultSet Objects as Return Values	164

6.1.3	String Patterns as Arguments	164
6.1.4	Pseudo Columns	166
6.2	DatabaseMetaData Interface Definition	166
6.3	DatabaseMetaData Methods	173
6.4	DatabaseMetaData Fields	236
7	DataTruncation	247
7.1	DataTruncation Overview	247
7.1.1	Data Truncation with No Warning or Exception	247
7.1.2	Data Truncation on Reads	247
7.1.3	Data Truncation on Writes	248
7.1.4	What a DataTruncation Object Contains	248
7.1.5	Retrieving DataTruncation Information	248
7.2	DataTruncation Class Definition	249
7.3	DataTruncation Constructor	249
7.4	DataTruncation Methods	250
8	Date	253
8.1	Date Overview	253
8.1.1	Creating a Date Object	253
8.2	Date Class Definition	254
8.3	Date Constructors	254
8.4	Date Methods	255
9	Driver	257
9.1	Driver Overview	257
9.1.1	Loading and Registering a Driver	257
9.1.2	JDBC Implementation Alternatives	258
9.2	Driver Interface Definition	259
9.3	Driver Methods	259
10	DriverManager	263
10.1	DriverManager Overview	263
10.1.1	Keeping Track of Available Drivers	263
10.1.2	Establishing a Connection	264
10.1.3	DriverManager Methods Are Static	265
10.2	DriverManager Class Definition	266
10.3	DriverManager Methods	266

11	DriverPropertyInfo	273
11.1	DriverPropertyInfo Overview	273
11.1.1	Creating a DriverPropertyInfo Object	273
11.2	DriverPropertyInfo Class Definition	274
11.3	DriverPropertyInfo Constructor	274
11.4	DriverPropertyInfo Variables	275
12	PreparedStatement	277
12.1	PreparedStatement Overview	277
12.1.1	Creating PreparedStatement Objects	278
12.1.2	Passing IN Parameters	278
12.1.3	Data Type Conformance on IN Parameters	279
12.1.4	Using setObject	279
12.1.5	Sending JDBC NULL as an IN parameter	280
12.1.6	Sending Very Large IN Parameters	280
12.2	PreparedStatement Interface Definition	281
12.3	PreparedStatement Methods	282
13	ResultSet	297
13.1	ResultSet Overview	297
13.1.1	Rows and Cursors	298
13.1.2	Retrieving Column Values	298
13.1.3	Datatypes and Conversions	299
13.1.4	Positioned Updates	301
13.1.5	Using Streams for Very Large Row Values	301
13.1.6	NULL Result Values	302
13.1.7	Optional or Multiple Result Sets	303
13.1.8	Closing a ResultSet Object	303
13.2	ResultSet Interface Definition	303
13.3	ResultSet Methods	305
14	ResultSetMetaData	319
14.1	ResultSetMetaData Overview	319
14.1.1	Creating a ResultSetMetaData Object	319
14.1.2	Using ResultSetMetaData	319
14.2	ResultSetMetaData Interface Definition	321
14.3	ResultSetMetaData Methods	321
14.4	ResultSetMetaData Fields	329

15	SQLException	331
15.1	SQLException Overview	331
15.1.1	What an SQLException Object Contains	331
15.1.2	Retrieving SQLException Information	332
15.1.3	What an SQLException Means	332
15.2	SQLException Class Definition	333
15.3	SQLException Constructors	333
15.4	SQLException Methods	335
16	SQLWarning	337
16.1	SQLWarning Overview	337
16.1.1	What an SQLWarning Object Contains	337
16.1.2	Retrieving SQLWarning Information	338
16.2	SQLWarning Class Definition	339
16.3	SQLWarning Constructors	339
16.4	SQLWarning Methods	341
17	Statement	343
17.1	Statement Overview	343
17.1.1	Creating Statement objects	343
17.1.2	Executing Statements Using Statement Objects	344
17.1.3	Statement Completion	345
17.1.4	Closing Statement Objects	345
17.1.5	SQL Escape Syntax in Statement Objects	345
17.1.6	Using the Method execute	348
17.2	Statement Interface Definition	351
17.3	Statement Methods	352
18	Time	363
18.1	Time Overview	363
18.1.1	Creating a Time Object	363
18.2	Time Class Definition	364
18.3	Time Constructors	364
18.4	Time Methods	365
19	Timestamp	367
19.1	Timestamp Overview	367
19.1.1	Creating a Timestamp Object	367
19.2	Timestamp Class Definition	368

19.3	Timestamp Constructors	368
19.4	Timestamp Methods	370
20	Types	375
20.1	Overview of Class Types	375
20.1.1	Using the Constants in Class Types	375
20.1.2	Using the Constant OTHER	376
20.2	Types Class Definition	376
21	Mapping SQL and Java Types	379
21.1	Overview	379
21.2	Mapping SQL Datatypes into Java	379
21.3	JDBC Types	381
21.3.1	CHAR, VARCHAR, LONGVARCHAR	381
21.3.2	BINARY, VARBINARY, LONGVARBINARY	382
21.3.3	BIT	383
21.3.4	TINYINT	383
21.3.5	SMALLINT	384
21.3.6	INTEGER	384
21.3.7	BIGINT	384
21.3.8	REAL	384
21.3.9	DOUBLE	385
21.3.10	FLOAT	385
21.3.11	DECIMAL and NUMERIC	385
21.3.12	DATE, TIME, and TIMESTAMP	386
21.4	Examples of Mapping	388
21.4.1	Simple SQL Statement	388
21.4.2	SQL Statement with IN Parameters	388
21.4.3	SQL Statement with INOUT Parameters	388
21.5	Dynamic Data Access	391
21.6	Tables for Type Mapping	392
21.6.1	JDBC Types Mapped to Java Types	393
21.6.2	Java Types Mapped to JDBC Types	394
21.6.3	JDBC Types Mapped to Java Object Types	395
21.6.4	Java Object Types Mapped to JDBC Types	396
21.6.5	Conversions by setObject	397
21.6.6	Types Retrieved by ResultSet.getXXX Methods	398
21.6.7	Types Mapped to Database-specific SQL Types	399

Appendix A: For Driver Writers 401

A.1	Requirements for All Drivers	401
A.1.1	Implement All Methods in the Interfaces	401
A.1.2	Implement a Special Static Section	402
A.1.3	Support Extensions to SQL-92 Entry Level	403
A.1.4	Support Scalar Functions	404
A.1.5	Provide Locks for Positioned Updates and Deletes	408
A.1.6	Support Multithreading	408
A.1.7	Throw Exceptions for Truncated Input Parameters	409
A.2	Permitted Variants	409
A.2.1	When Functionality Is Not Supported	410
A.2.2	Variation in Fundamental Properties	410
A.2.3	Adding Functionality	410
A.3	Security Responsibilities of Drivers	410
A.3.1	Check Shared TCP Connections	411
A.3.2	Check All Local File Access	412
A.3.3	Assume the Worst	412
A.4	SQLExceptions	413
A.5	Suggested Implementations	413
A.5.1	Prefetch Rows	413
A.5.2	Provide “Finalize” Methods for Applets	413
A.5.3	Avoid Implementation-dependent States	414

Appendix B: JDBC Design 415

B.1	A Bit of History	415
B.1.1	How It Started	415
B.1.2	What Can Be Learned from Our Experience?	416
B.2	Evolution of JDBC	417
B.2.1	ResultSet.getXXX Methods	418
B.2.2	PreparedStatement.setXXX Methods	418
B.2.3	CallableStatement.registerOutParameter Method	418
B.2.4	Support for Large OUT Parameters	418
B.2.5	isNull versus wasNull	419
B.2.6	Java Type Names or JDBC Type Names	419
B.2.7	Scrollable Cursors	420
B.3	Post 1.0 Changes	420
B.3.1	Numeric to Bignum to BigDecimal	420
B.3.2	AutoClose Mode Dropped	421
B.4	The Future of JDBC	421

Glossary	423
Index	431

JDBC™ Database Access with Java™



JDBC (Java Database Connectivity), the powerful API that is a standard part of the Java Class Libraries (JDK 1.1), represents one of the most exciting and fertile areas for Java development. Offering SQL compatibility and sophisticated database development capabilities, JDBC allows Java programmers access to MIS and Legacy systems.

From the creators of the JDBC API at JavaSoft, here is the definitive guide to this vital new technology. Appropriate for Java programmers of all skill levels, this book provides explanations of JDBC and Java fundamentals, a step-by-step tutorial to bring you up to speed, and an extensively annotated reference to all JDBC classes and interfaces.

Overviews of Java and the JDBC API explain how JDBC fits into the world of SQL and relational databases. The tutorial walks you through JDBC, demonstrating how to execute common tasks with numerous examples. For those interested in writing applications that access various database systems, the JDBC Metadata API is also introduced.

The alphabetically arranged reference section is comprehensive and detailed, containing class and interface definitions along with a wealth of in-depth information on methods and fields. Also included is important information on mapping Java and SQL types, a look at the rationale behind JDBC's design, and a glimpse into JDBC's future.

Anyone interested in pursuing the many development opportunities made possible by JDBC will find this book an indispensable resource.

Dr. Graham Hamilton is a Distinguished Engineer at JavaSoft where he led the design of the JDBC API, the JavaOS™ standalone Java system, and the JavaBeans™ component architecture. He was previously the technical lead for the Spring distributed object-oriented operating system project at Sun, and he has written a variety of papers and patents on operating systems and distributed systems.

Dr. Rick Cattell, a Distinguished Engineer at JavaSoft, led the design of the JDBC API. He is currently Chief Architect for the Java Enterprise Technologies group. He was previously responsible for SunSoft's NEO CORBA database integration and the Sun Simplify database GUI. Widely recognized for his contributions to the database field, Dr. Cattell was a cofounder of SQL Access, founder and chair of the Object Database Management Group (ODMG), and author of the first monograph on object database systems as well as more than 40 papers and 3 books.

Maydene Fisher has extensive experience as a technical writer specializing in the documentation of object-oriented programming languages. Fisher began her technical writing career on Wall Street, where she documented complex computer models, written in C++, for simulating fixed income derivatives. Before joining the JDBC team at JavaSoft she wrote documentation for ScriptX, an object-oriented multimedia scripting language, at Kaleida Labs and at Apple Computer.

The Java™ Series



The Java™ Series is the official source of complete, expert, and definitive information on Java, from the creators of the Java technology at Sun Microsystems. These books, written by members of the JavaSoft team at Sun, provide the inside information you need to understand Java thoroughly and to work with it effectively to build flexible, robust, portable, and secure Java applications and Internet applets. The series is an indispensable resource for anyone programming in Java.

...from the Source™



Cover design by Simone R. Payment

Cover art by Sara Ciaffaroni

♻️ Text printed on recycled paper

♣️ ADDISON-WESLEY

Addison-Wesley Corporate & Professional is an imprint of Addison Wesley Longman, Inc.

