

Gunter Saake • Kai-Uwe Sattler

Datenbanken & Java

JDBC, SQU, ODMG und JDO

2., überarbeitete und aktualisierte Auflage



dpunkt.verlag

Inhaltsverzeichnis

1	Einführung	1
1.1	Inhalt des Buches.	2
1.2	Architekturen für Datenbankanwendungen.	4
1.3	Beispielanwendung.	7
1.4	Technische Hinweise.	8
2	Java im Überblick	11
2.1	Einführung.	11
2.2	Datentypen.	13
2.3	Kontrollstrukturen und Operatoren.	13
2.4	Klassen, Objekte und Methoden.	14
2.5	Pakete.	18
2.6	Ausnahmen.	19
2.7	Übersetzung und Ausführung.	20
2.8	Dateiarbeit mit Java.	22
2.8.1	Ströme für Datenquellen und -senken.	23
2.8.2	Filterströme.	24
2.8.3	Datenströme.	25
2.8.4	Objektströme.	26
3	Grundlagen von relationalen Datenbanken	31
3.1	Einführung.	32
3.1.1	Datenunabhängigkeit.	33
3.1.2	Transaktionen.	34
3.2	Das Relationenmodell.	36
3.3	Relationaler Datenbankentwurf und Normalisierung.	38
3.4	SQL - ein Überblick.	40
3.4.1	Tabellendeklaration in SQL.	40
3.4.2	Anfragen in SQL.	42
3.4.3	Änderungsoperationen in SQL.	44
3.4.4	Weitere Aspekte von SQL.	45
3.5	Anwendungsentwicklung für RDBMS.	49

4	JDBC - das Datenbank-API für Java	55
4.1	Überblick	55
4.2	JDBC an einem Beispiel	59
4.3	JDBC im Detail	63
4.3.1	Treiber und Datenbankverbindung	64
4.3.2	Ausführung von SQL-Anfragen	78
4.3.3	ResultSet: Ergebnisse von Anfragen	92
4.3.4	Metadaten	106
4.3.5	Fehlerbehandlung	111
4.3.6	Abbildung von SQL-Typen in Java	114
4.4	Erweiterte Konzepte	116
4.4.1	SQL:1999-Datentypen	116
4.4.2	Batch-Updates	129
4.4.3	Connection Pooling	131
4.4.4	Verteilte Transaktionen	133
4.4.5	RowSets	134
4.4.6	Abschließende Bemerkungen	138
4.5	Beispielanwendung mit JDBC	138
5	SQLJ - die Verbindung von Java und SQL	145
5.1	Embedded SQL für Java	145
5.1.1	Überblick	145
5.1.2	Embedded SQL an einem Beispiel	147
5.1.3	SQLJ-Klauseln und Hostvariablen	150
5.1.4	Iteratoren	155
5.1.5	Kontexte	161
5.1.6	Der SQLJ-Translator und Profile	165
5.1.7	Interoperabilität mit JDBC	170
5.1.8	Dynamisches SQL mit Oracle-SQLJ	173
5.2	Gespeicherte Prozeduren	175
5.2.1	Überblick	176
5.2.2	Implementierung von gespeicherten Prozeduren	178
5.2.3	Installation und Registrierung	184
5.2.4	Aufruf und Anwendung	190
5.3	Java-Klassen als SQL-Datentypen	190
5.3.1	Implementierung von SQL-Typen	191
5.3.2	Installation und Registrierung	191
6	Java und Objektdatenbanken	195
6.1	Einführung	195
6.2	Das Objektmodell der ODMG	200
6.3	Das Java-Binding der ODMG	208
6.3.1	Prinzip	208

6.3.2	Java-ODL	210
6.3.3	Java-OML	217
6.4	Anfragen in OQL	233
6.5	Beispielanwendung mit der ODMG-Anbindung	239
7	Von Objekten zu Relationen.	247
7.1	Grundlagen der objektrelationalen Abbildung	247
7.2	Aspekte der Laufzeitunterstützung	254
7.3	Java Data Objects	257
7.3.1	Persistenzmodell	259
7.3.2	Mapping-Definition	268
7.3.3	Das JDO-API	276
7.3.4	Anfragen mit JDOQL	286
7.4	Zusammenfassung	292
8	Java und Transaktionen	295
8.1	Einführung in Transaktionsbegriffe	295
8.1.1	Der Scheduler	295
8.1.2	Synchronisation und Serialisierbarkeit	297
8.1.3	Operationen für die Transaktionssteuerung	298
8.1.4	Transaktionsmonitore	299
8.2	Verteilte Transaktionen und Transaktionsmanager	300
8.3	Java Transaction API	303
8.3.1	Schnittstelle JTA	303
8.3.2	Beispielprogramm	305
9	Orthogonal persistentes Java — eine Alternative?	309
9.1	Persistente Programmiersprachen	309
9.2	PJama - ein orthogonal persistentes Java	311
9.3	PJama: Programmierschnittstellen	312
9.4	PJama: Ein Beispiel	314
9.5	Anwendungsentwicklung mit PJama	317
10	Ausblick	319
A	Schemadefinition für die Beispielanwendung	323
A.1	SQL-Skript zur Definition des Beispielschemas	323
A.2	Relationale Repräsentation der Beispieldaten	325
A.3	Java-Klassen für die ODMG-Datenbank	327
A.4	XML-Datei für das JDO-Mapping mit Kodo	336
	Literaturverzeichnis	337
	Tabellenverzeichnis	341

Abbildungsverzeichnis	343
Verzeichnis der Beispiele	345
Index	347