

---

# *Hibernate in Action*

CHRISTIAN BAUER  
GAVIN KING



MANNING

Greenwich  
(74° w. long.)

---

# contents

---

<i>foreword</i>	<i>xi</i>
<i>preface</i>	<i>xiii</i>
<i>acknowledgments</i>	<i>xv</i>
<i>about this book</i>	<i>xvi</i>
<i>about Hibernate3 and EJB 3</i>	<i>xx</i>
<i>author online</i>	<i>xxi</i>
<i>about the title and cover</i>	<i>xxii</i>

## **I Understanding object/relational persistence 1**

1.1	What is persistence? 3
	<i>Relational databases 3</i> ▪ <i>Understanding SQL 4</i> ▪ <i>Using SQL in Java 5</i> ▪ <i>Persistence in object-oriented applications 5</i>
1.2	The paradigm mismatch 7
	<i>The problem of granularity 9</i> ▪ <i>The problem of subtypes 10</i> <i>The problem of identity 11</i> ▪ <i>Problems relating to associations 13</i> <i>The problem of object graph navigation 14</i> ▪ <i>The cost of the mismatch 15</i>
1.3	Persistence layers and alternatives 16
	<i>Layered architecture 17</i> ▪ <i>Hand-coding a persistence layer with SQL/JDBC 18</i> ▪ <i>Using serialization 19</i> ▪ <i>Considering EJB entity beans 20</i> ▪ <i>Object-oriented database systems 21</i> <i>Other options 22</i>
1.4	Object/relational mapping 22
	<i>What is ORM? 23</i> ▪ <i>Generic ORM problems 25</i> <i>Why ORM? 26</i>
1.5	Summary 29

<b>2</b>	<b>Introducing and integrating Hibernate</b>	<b>30</b>
2.1	“Hello World” with Hibernate	31
2.2	Understanding the architecture	36
	<i>The core interfaces</i>	38
	<i>Callback interfaces</i>	40
	<i>Types</i>	40
	<i>Extension interfaces</i>	41
2.3	Basic configuration	41
	<i>Creating a SessionFactory</i>	42
	<i>Configuration in non-managed environments</i>	45
	<i>Configuration in managed environments</i>	48
2.4	Advanced configuration settings	51
	<i>Using XML-based configuration</i>	51
	<i>JNDI-bound SessionFactory</i>	53
	<i>Logging</i>	54
	<i>Java Management Extensions (JMX)</i>	55
2.5	Summary	58
<b>3</b>	<b>Mapping persistent classes</b>	<b>59</b>
3.1	The CaveatEmptor application	60
	<i>Analyzing the business domain</i>	61
	<i>The CaveatEmptor domain model</i>	61
3.2	Implementing the domain model	64
	<i>Addressing leakage of concerns</i>	64
	<i>Transparent and automated persistence</i>	65
	<i>Writing POJOs</i>	67
	<i>Implementing POJO associations</i>	69
	<i>Adding logic to accessor methods</i>	73
3.3	Defining the mapping metadata	75
	<i>Metadata in XML</i>	75
	<i>Basic property and class mappings</i>	78
	<i>Attribute-oriented programming</i>	84
	<i>Manipulating metadata at runtime</i>	86
3.4	Understanding object identity	87
	<i>Identity versus equality</i>	87
	<i>Database identity with Hibernate</i>	88
	<i>Choosing primary keys</i>	90
3.5	Fine-grained object models	92
	<i>Entity and value types</i>	93
	<i>Using components</i>	93
3.6	Mapping class inheritance	97
	<i>Table per concrete class</i>	97
	<i>Table per class hierarchy</i>	99
	<i>Table per subclass</i>	101
	<i>Choosing a strategy</i>	104

- 3.7 Introducing associations 105
  - Managed associations?* 106
  - Multiplicity* 106
  - The simplest possible association* 107
  - Making the association bidirectional* 108
  - A parent/child relationship* 111
- 3.8 Summary 112

## 4 **Working with persistent objects 114**

- 4.1 The persistence lifecycle 115
  - Transient objects* 116
  - Persistent objects* 117
  - Detached objects* 118
  - The scope of object identity* 119
  - Outside the identity scope* 121
  - Implementing equals() and hashCode()* 122
- 4.2 The persistence manager 126
  - Making an object persistent* 126
  - Updating the persistent state of a detached instance* 127
  - Retrieving a persistent object* 129
  - Updating a persistent object* 129
  - Making a persistent object transient* 129
  - Making a detached object transient* 130
- 4.3 Using transitive persistence in Hibernate 131
  - Persistence by reachability* 131
  - Cascading persistence with Hibernate* 133
  - Managing auction categories* 134
  - Distinguishing between transient and detached instances* 138
- 4.4 Retrieving objects 139
  - Retrieving objects by identifier* 140
  - Introducing HQL* 141
  - Query by criteria* 142
  - Query by example* 143
  - Fetching strategies* 143
  - Selecting a fetching strategy in mappings* 146
  - Tuning object retrieval* 151
- 4.5 Summary 152

## 5 **Transactions, concurrency, and caching 154**

- 5.1 Transactions, concurrency, and caching 154
- 5.2 Understanding database transactions 156
  - JDBC and JTA transactions* 157
  - The Hibernate Transaction API* 158
  - Flushing the Session* 160
  - Understanding isolation levels* 161
  - Choosing an isolation level* 163
  - Setting an isolation level* 165
  - Using pessimistic locking* 165
- 5.3 Working with application transactions 168
  - Using managed versioning* 169
  - Granularity of a Session* 172
  - Other ways to implement optimistic locking* 174

- 5.4 Caching theory and practice 175
  - Caching strategies and scopes* 176 ▪ *The Hibernate cache architecture* 179 ▪ *Caching in practice* 185
- 5.5 Summary 194

## 6 **Advanced mapping concepts** 195

- 6.1 Understanding the Hibernate type system 196
  - Built-in mapping types* 198 ▪ *Using mapping types* 200
- 6.2 Mapping collections of value types 211
  - Sets, bags, lists, and maps* 211
- 6.3 Mapping entity associations 220
  - One-to-one associations* 220 ▪ *Many-to-many associations* 225
- 6.4 Mapping polymorphic associations 234
  - Polymorphic many-to-one associations* 234 ▪ *Polymorphic collections* 236 ▪ *Polymorphic associations and table-per-concrete-class* 237
- 6.5 Summary 239

## 7 **Retrieving objects efficiently** 241

- 7.1 Executing queries 243
  - The query interfaces* 243 ▪ *Binding parameters* 245
  - Using named queries* 249
- 7.2 Basic queries for objects 250
  - The simplest query* 250 ▪ *Using aliases* 251 ▪ *Polymorphic queries* 251 ▪ *Restriction* 252 ▪ *Comparison operators* 253
  - String matching* 255 ▪ *Logical operators* 256 ▪ *Ordering query results* 257
- 7.3 Joining associations 258
  - Hibernate join options* 259 ▪ *Fetching associations* 260
  - Using aliases with joins* 262 ▪ *Using implicit joins* 265
  - Theta-style joins* 267 ▪ *Comparing identifiers* 268
- 7.4 Writing report queries 269
  - Projection* 270 ▪ *Using aggregation* 272 ▪ *Grouping* 273
  - Restricting groups with having* 274 ▪ *Improving performance with report queries* 275

- 7.5 Advanced query techniques 276
  - Dynamic queries* 276 ▪ *Collection filters* 279
  - Subqueries* 281 ▪ *Native SQL queries* 283
- 7.6 Optimizing object retrieval 286
  - Solving the n+1 selects problem* 286 ▪ *Using iterate()* queries 289 ▪ *Caching queries* 290
- 7.7 Summary 292

## 8 Writing Hibernate applications 294

- 8.1 Designing layered applications 295
  - Using Hibernate in a servlet engine* 296
  - Using Hibernate in an EJB container* 311
- 8.2 Implementing application transactions 320
  - Approving a new auction* 321 ▪ *Doing it the hard way* 322
  - Using detached persistent objects* 324 ▪ *Using a long session* 325
  - Choosing an approach to application transactions* 329
- 8.3 Handling special kinds of data 330
  - Legacy schemas and composite keys* 330 ▪ *Audit logging* 340
- 8.4 Summary 347

## 9 Using the toolset 348

- 9.1 Development processes 349
  - Top down* 350 ▪ *Bottom up* 350 ▪ *Middle out (metadata oriented)* 350 ▪ *Meet in the middle* 350
  - Roundtripping* 351
- 9.2 Automatic schema generation 351
  - Preparing the mapping metadata* 352 ▪ *Creating the schema* 355 ▪ *Updating the schema* 357
- 9.3 Generating POJO code 358
  - Adding meta-attributes* 358 ▪ *Generating finders* 360
  - Configuring hbm2java* 362 ▪ *Running hbm2java* 363
- 9.4 Existing schemas and Middlegen 364
  - Starting Middlegen* 364 ▪ *Restricting tables and relationships* 366 ▪ *Customizing the metadata generation* 368
  - Generating hbm2java and XDoclet metadata* 370

9.5	XDoclet	372
	<i>Setting value type attributes</i>	372
	<i>Mapping entity associations</i>	374
	<i>Running XDoclet</i>	375
9.6	Summary	376
<b><i>appendix A: SQL fundamentals</i></b>		<b>378</b>
<b><i>appendix B: ORM implementation strategies</i></b>		<b>382</b>
B.1	Properties or fields?	383
B.2	Dirty-checking strategies	384
<b><i>appendix C: Back in the real world</i></b>		<b>388</b>
C.1	The strange copy	389
C.2	The more the better	390
C.3	We don't need primary keys	390
C.4	Time isn't linear	391
C.5	Dynamically unsafe	391
C.6	To synchronize or not?	392
C.7	Really fat client	393
C.8	Resuming Hibernate	394
<b><i>references</i></b>		<b>395</b>
<b><i>index</i></b>		<b>397</b>