contents

foreword to the revised edition xix
foreword to the first edition xxi
preface to the revised edition xxiii
preface to the first edition xxv
acknowledgments xxviii
about this book xxix
about the cover illustration xxxiii

PART 1 GETTING STARTED WITH HIBERNATE AND EJB 3.0 1

1 ***Understanding object/relational persistence 3***

1.1 What is persistence? 5

Relational databases 5 ▪ Understanding SQL 6 ▪ Using SQL in Java 7 ▪ Persistence in object-oriented applications 8

1.2 The paradigm mismatch 10

The problem of granularity 12 ▪ The problem of subtypes 13
The problem of identity 14 ▪ Problems relating to associations 16 ▪ The problem of data navigation 18
The cost of the mismatch 19

1.3 Persistence layers and alternatives 20

Layered architecture 20 ▪ Hand-coding a persistence layer with SQL/JDBC 22 ▪ Using serialization 23 Object-oriented database systems 23 ▪ Other options 24

1.4 Object/relational mapping 24

What is ORM? 25 ▪ Generic ORM problems 27 Why ORM? 28 ▪ Introducing Hibernate, EJB3, and JPA 31

1.5 Summary 35

2 ***Starting a project 37***

2.1 Starting a Hibernate project 38

Selecting a development process 39 ▪ Setting up the project 41 ▪ Hibernate configuration and startup 49 ▪ Running and testing the application 60

2.2 Starting a Java Persistence project 68

Using Hibernate Annotations 68 ▪ Using Hibernate EntityManager 72 ▪ Introducing EJB components 79 Switching to Hibernate interfaces 86

2.3 Reverse engineering a legacy database 88

Creating a database configuration 89 ▪ Customizing reverse engineering 90 ▪ Generating Java source code 92

2.4 Integration with Java EE services 96

Integration with JTA 97 ▪ JNDI-bound SessionFactory 101 JMX service deployment 103

2.5 Summary 104

3 ***Domain models and metadata 105***

3.1 The CaveatEmptor application 106

Analyzing the business domain 107 ▪ The CaveatEmptor domain model 108

3.2 Implementing the domain model 110

Addressing leakage of concerns 111 ▪ Transparent and automated persistence 112 ▪ Writing POJOs and persistent entity classes 113 ▪ Implementing POJO associations 116 Adding logic to accessor methods 120

3.3 Object/relational mapping metadata 123

Metadata in XML 123 ▪ Annotation-based metadata 125 Using XDoclet 131 ▪ Handling global metadata 133 Manipulating metadata at runtime 138

3.4 Alternative entity representation 140

Creating dynamic applications 141 ▪ Representing data in XML 148

3.5 Summary 152

PART 2 MAPPING CONCEPTS AND STRATEGIES 155

4 *Mapping persistent classes 157*

4.1 Understanding entities and value types 158

Fine-grained domain models 158 ▪ Defining the concept 159 Identifying entities and value types 160

4.2 Mapping entities with identity 161

Understanding Java identity and equality 162 ▪ Handling database identity 162 ▪ Database primary keys 166

4.3 Class mapping options 171

Dynamic SQL generation 172 ▪ Making an entity immutable 173 ▪ Naming entities for querying 173 Declaring a package name 174 ▪ Quoting SQL identifiers 175 Implementing naming conventions 175

4.4 Fine-grained models and mappings 177

Mapping basic properties 177 ▪ Mapping components 184

4.5 Summary 189

5 *Inheritance and custom types 191*

5.1 Mapping class inheritance 192

Table per concrete class with implicit polymorphism 192
Table per concrete class with unions 195 ▪ Table per class hierarchy 199 ▪ Table per subclass 203
Mixing inheritance strategies 207 ▪ Choosing a strategy 210

5.2 The Hibernate type system 212

Recapitulating entity and value types 212
Built-in mapping types 214 ▪ Using mapping types 219

5.3 Creating custom mapping types 220

Considering custom mapping types 221 ▪ The extension points 222 ▪ The case for custom mapping types 223 ▪ Creating a UserType 224
Creating a CompositeUserType 228 ▪ Parameterizing custom types 230 ▪ Mapping enumerations 233

5.4 Summary 239

6 *Mapping collections and entity associations 240*

6.1 Sets, bags, lists, and maps of value types 241

Selecting a collection interface 241 ▪ Mapping a set 243 ▪ Mapping an identifier bag 244
Mapping a list 246 ▪ Mapping a map 247
Sorted and ordered collections 248

6.2 Collections of components 251

Writing the component class 252 ▪ Mapping the collection 252 ▪ Enabling bidirectional navigation 253
Avoiding not-null columns 254

6.3 Mapping collections with annotations 256

Basic collection mapping 256 ▪ Sorted and ordered collections 257 ▪ Mapping a collection of embedded objects 258

6.4 Mapping a parent/children relationship 260

Multiplicity 261 ▪ The simplest possible association 261 Making the association bidirectional 264 ▪ Cascading object state 267

6.5 Summary 275

7 ***Advanced entity association mappings 277***

7.1 Single-valued entity associations 278

Shared primary key associations 279 ▪ One-to-one foreign key associations 282 ▪ Mapping with a join table 285

7.2 Many-valued entity associations 290

One-to-many associations 290 ▪ Many-to-many associations 297 ▪ Adding columns to join tables 303 Mapping maps 310

7.3 Polymorphic associations 313

Polymorphic many-to-one associations 313 ▪ Polymorphic collections 315 ▪ Polymorphic associations to unions 316 Polymorphic table per concrete class 319

7.4 Summary 321

8 ***Legacy databases and custom SQL 322***

8.1 Integrating legacy databases 323

Handling primary keys 324 ▪ Arbitrary join conditions with formulas 337 ▪ Joining arbitrary tables 342 ▪ Working with triggers 346

8.2 Customizing SQL 350

Writing custom CRUD statements 351 Integrating stored procedures and functions 356

8.3 Improving schema DDL 364

Custom SQL names and datatypes 365 ▪ Ensuring data consistency 367 ▪ Adding domains and column

constraints 369 ▪ Table-level constraints 370
Database constraints 373 ▪ Creating indexes 375
Adding auxiliary DDL 376

8.4 Summary 378

PART 3 CONVERSATIONAL OBJECT PROCESSING 381

9 ***Working with objects 383***

9.1 The persistence lifecycle 384

Object states 385 ▪ The persistence context 388

9.2 Object identity and equality 391

Introducing conversations 391 ▪ The scope of object identity 393 ▪ The identity of detached objects 394
Extending a persistence context 400

9.3 The Hibernate interfaces 401

Storing and loading objects 402 ▪ Working with detached objects 408 ▪ Managing the persistence context 414

9.4 The Java Persistence API 417

Storing and loading objects 417 ▪ Working with detached entity instances 423

9.5 Using Java Persistence in EJB components 426

Injecting an EntityManager 426 ▪ Looking up an EntityManager 429 ▪ Accessing an EntityManagerFactory 429

9.6 Summary 431

10 ***Transactions and concurrency 433***

10.1 Transaction essentials 434

Database and system transactions 435 ▪ Transactions in a Hibernate application 437 ▪ Transactions with Java Persistence 449

10.2 Controlling concurrent access 453

Understanding database-level concurrency 453 ▪ Optimistic concurrency control 458 ▪ Obtaining additional isolation guarantees 465

10.3 Nontransactional data access 469

Debunking autocommit myths 470 ▪ Working nontransactionally with Hibernate 471 ▪ Optional transactions with JTA 473

10.4 Summary 474

11 *Implementing conversations 476*

11.1 Propagating the Hibernate Session 477

The use case for Session propagation 478 ▪ Propagation through thread-local 480 ▪ Propagation with JTA 482 ▪ Propagation with EJBs 483

11.2 Conversations with Hibernate 485

Providing conversational guarantees 485 ▪ Conversations with detached objects 486 ▪ Extending a Session for a conversation 489

11.3 Conversations with JPA 497

Persistence context propagation in Java SE 498
Merging detached objects in conversations 499
Extending the persistence context in Java SE 501

11.4 Conversations with EJB 3.0 506

Context propagation with EJBs 506
Extended persistence contexts with EJBs 510

11.5 Summary 515

12 *Modifying objects efficiently 517*

12.1 Transitive persistence 518

Persistence by reachability 519 ▪ Applying cascading to associations 520 ▪ Working with transitive state 524
Transitive associations with JPA 531

12.2 Bulk and batch operations 532

Bulk statements with HQL and JPA QL 533 ▪ Processing with batches 537 ▪ Using a stateless Session 539

12.3 Data filtering and interception 540

Dynamic data filters 541 ▪ Intercepting Hibernate events 546 The core event system 553 ▪ Entity listeners and callbacks 556

12.4 Summary 558

13 ***Optimizing fetching and caching 559***

13.1 Defining the global fetch plan 560

The object-retrieval options 560 ▪ The lazy default fetch plan 564 ▪ Understanding proxies 564 ▪ Disabling proxy generation 567 ▪ Eager loading of associations and collections 568 ▪ Lazy loading with interception 571

13.2 Selecting a fetch strategy 573

Prefetching data in batches 574 ▪ Prefetching collections with subselects 577 ▪ Eager fetching with joins 578 ▪ Optimizing fetching for secondary tables 581 ▪ Optimization guidelines 584

13.3 Caching fundamentals 592

Caching strategies and scopes 593 ▪ The Hibernate cache architecture 597

13.4 Caching in practice 602

Selecting a concurrency control strategy 602 ▪ Understanding cache regions 604 ▪ Setting up a local cache provider 605 Setting up a replicated cache 606 ▪ Controlling the second-level cache 611

13.5 Summary 612

14 ***Querying with HQL and JPA QL 614***

14.1 Creating and running queries 615

Preparing a query 616 ▪ Executing a query 625 Using named queries 629

14.2 Basic HQL and JPA QL queries 633

Selection 633 ▪ Restriction 635 ▪ Projection 641

14.3 Joins, reporting queries, and subselects 643

Joining relations and associations 643 ▪ Reporting queries 655 ▪ Using subselects 659

14.4 Summary 662

15 ***Advanced query options 663***

15.1 Querying with criteria and example 664

Basic criteria queries 665 ▪ Joins and dynamic fetching 670 ▪ Projection and report queries 676 Query by example 680

15.2 Using native SQL queries 683

Automatic resultset handling 683 ▪ Retrieving scalar values 684 ▪ Native SQL in Java Persistence 686

15.3 Filtering collections 688

15.4 Caching query results 691

Enabling the query result cache 691 ▪ Understanding the query cache 692 ▪ When to use the query cache 693 Natural identifier cache lookups 693

15.5 Summary 695

16 ***Creating and testing layered applications 697***

16.1 Hibernate in a web application 698

Introducing the use case 698 ▪ Writing a controller 699 The Open Session in View pattern 701 ▪ Designing smart domain models 705

16.2 Creating a persistence layer 708

A generic data-access object pattern 709 ▪ Implementing the generic CRUD interface 711 ▪ Implementing entity DAOs 713 Using data-access objects 715

16.3 Introducing the Command pattern 718

The basic interfaces 719 ▪ Executing command objects 721 Variations of the Command pattern 723

16.4 Designing applications with EJB 3.0 725

Implementing a conversation with stateful beans 725 ▪ Writing DAOs with EJBs 727 ▪ Utilizing dependency injection 728

16.5 Testing 730

Understanding different kinds of tests 731 ▪ Introducing TestNG 732 ▪ Testing the persistence layer 736 Considering performance benchmarks 744

16.6 Summary 746

17 ***Introducing JBoss Seam 747***

17.1 The Java EE 5.0 programming model 748

Considering JavaServer Faces 749 ▪ Considering EJB 3.0 751 Writing a web application with JSF and EJB 3.0 752 Analyzing the application 762

17.2 Improving the application with Seam 765

Configuring Seam 766 ▪ Binding pages to stateful Seam components 767 ▪ Analyzing the Seam application 773

17.3 Understanding contextual components 779

Writing the login page 779 ▪ Creating the components 781 Aliasing contextual variables 784 ▪ Completing the login/logout feature 786

17.4 Validating user input 789

Introducing Hibernate Validator 790 ▪ Creating the registration page 791 ▪ Internationalization with Seam 799

17.5 Simplifying persistence with Seam 803

Implementing a conversation 804 ▪ Letting Seam manage the persistence context 811

17.6 Summary 816

appendix A SQL fundamentals 818

appendix B Mapping quick reference 822

references 824

index 825