contents

foreword *ix*
preface *xi*
acknowledgments *xii*
about this book *xiii*
about the authors *xviii*
about the cover illustration *xix*

PART 1 GETTING STARTED WITH GRAPH DATABASES 1

1 ***Introduction to graphs* 3**

1.1 What is a graph? 4

What is a graph database? 6 ▪ Comparison with other types of databases 7 ▪ Why can't I use SQL? 9

1.2 Is my problem a graph problem? 16

Explore the questions 16 ▪ I'm still confused. . . . Is this a graph problem? 19

2 ***Graph data modeling* 24**

2.1 The data modeling process 25

Data modeling terms 25 ▪ Four-step process for data modeling 27

2.2 Understand the problem 29

Domain and scope questions 30 ▪ Business entity questions 31 Functionality questions 32

2.3 Developing the whiteboard model 33

Identifying and grouping entities 33 ▪ Identifying relationships between entities 34

2.4 Constructing the logical data model 37

Translating entities to vertices 38 ▪ Translating relationships to edges 41 ▪ Finding and assigning properties 49

2.5 Checking our model 51

3 ***Running basic and recursive traversals 54***

3.1 Setting up your environment 55

Starting the Gremlin Server 55 ▪ Starting the Gremlin Console, connecting to the Gremlin Server, and loading the data 56

3.2 Traversing a graph 57

Using a logical data model (schema) to plan traversals 58 Planning the steps through the graph data 59 ▪ Fundamental concepts of traversing a graph 61 ▪ Writing traversals in Gremlin 63 ▪ Retrieving properties with values steps 68

3.3 Recursive traversals 68

Using recursive logic 68 ▪ Writing recursive traversals in Gremlin 72

4 ***Pathfinding traversals and mutating graphs 80***

4.1 Mutating a graph 81

Creating vertices and edges 82 ▪ Removing data from our graph 86 ▪ Updating a graph 88 ▪ Extending our graph 95

4.2 Paths 98

Cycles in graphs 100 ▪ Finding the simple path 101

4.3 Traversing and filtering edges 103

Introducing the E and V steps for traversing edges 103 Filtering with edge properties 108 ▪ Include edges in path results 109 ▪ Performant edge counts and denormalization 110

5 ***Formatting results 112***

5.1 Review of values steps 113

5.2 Constructing our result payload 118

Applying aliases in Gremlin 120 ▪ Projecting results instead of aliasing 127

5.3 Organizing our results 131

Ordering results returned from a graph traversal 131
Grouping results returned from a graph traversal 133
Limiting results 135

5.4 Combining steps into complex traversals 137

6 ***Developing an application 142***

6.1 Starting the project 143

Selecting our tools 144 ▪ Setting up the project 145
Obtaining a driver 145 ▪ Preparing the database server Instance 146

6.2 Connecting to our database 147

Building the cluster configuration 147 ▪ Setting up the GraphTraversalSource 148

6.3 Retrieving data 151

Retrieving a vertex 151 ▪ Using Gremlin language variants (GLVs) 153 ▪ Adding terminal steps 154 ▪ Creating the Java method in our application 155

6.4 Adding, modifying, and deleting data 156

Adding vertices 156 ▪ Adding edges 158 ▪ Updating properties 160 ▪ Deleting elements 160

6.5 Translating our list and path traversals 163

Getting a list of results 163 ▪ Implementing recursive traversals 164 ▪ Implementing paths 166

PART 2 BUILDING ON GRAPH DATABASES 169

7 ***Advanced data modeling techniques 171***

7.1 Reviewing our current data models 172

7.2 Extending our logical data model 173

7.3 Translating entities to vertices 176

Using generic labels 177 ▪ Denormalizing graph data 183
Translating relationships to edges 188 ▪ Finding and assigning

properties 190 ▪ Moving properties to edges 191 ▪ Checking our model 194

7.4 Extending our data model for personalization 194

7.5 Comparing the results 196

8 ***Building traversals using known walks 198***

8.1 Preparing to develop our traversals 199

Identifying the required elements 199 ▪ Selecting a starting place 204 ▪ Setting up test data 205

8.2 Writing our first traversal 206

Designing our traversal 206 ▪ Developing the traversal code 210

8.3 Pagination and graph databases 214

8.4 Recommending the highest-rated restaurants 218

Designing our traversal 218 ▪ Developing the traversal code 220

8.5 Writing the last recommendation engine traversal 230

Designing our traversal 232 ▪ Adding this traversal to our application 234

9 ***Working with subgraphs 237***

9.1 Working with subgraphs 238

Extracting a subgraph 238 ▪ Traversing a subgraph 244

9.2 Building a subgraph for personalization 246

9.3 Building the traversal 252

Reversing the traversing direction 253 ▪ Evaluating the individualized results of the subgraph 255

9.4 Implementing a subgraph with a remote connection 256

Connecting with TinkerPop's Client class 257 ▪ Adding this traversal to our application 258

PART 3 MOVING BEYOND THE BASICS 261

10 ***Performance, pitfalls, and anti-patterns 263***

10.1 Slow-performing traversals 264

Explaining our traversal 264 ▪ Profiling our traversal 266 Indexes 268

10.2 Dealing with supernodes 269

It's about instance data 270 ▪ It's about the database 270
What makes a supernode? 270 ▪ Monitoring for supernodes 271
What to do if you have a supernode 273

10.3 Application anti-patterns 277

Using graphs for non-graph use cases 277 ▪ Dirty data 278
Lack of adequate testing 280

10.4 Traversal anti-patterns 280

Not using parameterized traversals 280 ▪ Using unlabeled filtering steps 283

11 *What's next: Graph analytics, machine learning, and resources 286*

11.1 Graph analytics 287

Pathfinding 287 ▪ Centrality 290 ▪ Community detection 292 ▪ Graphs and machine learning 296
Additional resources 297

11.2 Final thoughts 299

appendix Apache TinkerPop installation and overview 301

index 311