

Table of Contents

Preface	xiii
1. Jumping Right In: Some Data and Some Queries	1
The Data to Query	2
Querying the Data	3
More Realistic Data and Matching on Multiple Triples	8
Searching for Strings	12
What Could Go Wrong?	13
Querying a Public Data Source	14
Summary	17
2. The Semantic Web, RDF, and Linked Data (and SPARQL)	19
What Exactly Is the “Semantic Web”?	19
URLs, URIs, IRIs, and Namespaces	21
The Resource Description Framework (RDF)	24
Storing RDF in Files	24
Storing RDF in Databases	29
Data Typing	30
Making RDF More Readable with Language Tags and Labels	31
Blank Nodes and Why They’re Useful	33
Named Graphs	35
Reusing and Creating Vocabularies: RDF Schema and OWL	36
Linked Data	41
SPARQL’s Past, Present, and Future	43
The SPARQL Specifications	44
Summary	45
3. SPARQL Queries: A Deeper Dive	47
More Readable Query Results	48
Using the Labels Provided by DBpedia	50
Getting Labels from Schemas and Ontologies	53

Data That Might Not Be There	55
Finding Data That Doesn't Meet Certain Conditions	59
Searching Further in the Data	61
Searching with Blank Nodes	68
Eliminating Redundant Output	69
Combining Different Search Conditions	72
FILTERing Data Based on Conditions	75
Retrieving a Specific Number of Results	78
Querying Named Graphs	80
Queries in Your Queries	87
Combining Values and Assigning Values to Variables	88
Creating Tables of Values in Your Queries	91
Sorting, Aggregating, Finding the Biggest and Smallest and...	95
Sorting Data	96
Finding the Smallest, the Biggest, the Count, the Average...	98
Grouping Data and Finding Aggregate Values within Groups	100
Querying a Remote SPARQL Service	102
Federated Queries: Searching Multiple Datasets with One Query	105
Summary	107
4. Copying, Creating, and Converting Data (and Finding Bad Data)	109
Query Forms: SELECT, DESCRIBE, ASK, and CONSTRUCT	110
Copying Data	111
Creating New Data	115
Converting Data	120
Finding Bad Data	123
Defining Rules with SPARQL	124
Generating Data About Broken Rules	127
Using Existing SPARQL Rules Vocabularies	131
Asking for a Description of a Resource	133
Summary	134
5. Datatypes and Functions	135
Datatypes and Queries	135
Representing Strings	141
Comparing Values and Doing Arithmetic	142
Functions	145
Program Logic Functions	146
Node Type and Datatype Checking Functions	150
Node Type Conversion Functions	153
Datatype Conversion	158
Checking, Adding, and Removing Spoken Language Tags	164
String Functions	171

Numeric Functions	175
Date and Time Functions	177
Hash Functions	179
Extension Functions	182
Summary	183
6. Updating Data with SPARQL	185
Getting Started with Fuseki	186
Adding Data to a Dataset	188
Deleting Data	194
Changing Existing Data	196
Named Graphs	201
Dropping Graphs	204
Named Graph Syntax Shortcuts: WITH and USING	206
Copying and Moving Entire Graphs	209
Deleting and Replacing Triples in Named Graphs	210
Summary	215
7. Query Efficiency and Debugging	217
Efficiency Inside the WHERE Clause	217
Reduce the Search Space	218
OPTIONAL Is Very Optional	219
Triple Pattern Order Matters	220
FILTERs: Where and What	221
Property Paths Can Be Expensive	225
Efficiency Outside the WHERE Clause	226
Debugging	227
Manual Debugging	227
SPARQL Algebra	229
Debugging Tools	231
Summary	232
8. Working with SPARQL Query Result Formats	235
SPARQL Query Results XML Format	238
Processing XML Query Results	241
SPARQL Query Results JSON Format	244
Processing JSON Query Results	247
SPARQL Query Results CSV and TSV Formats	249
Using CSV Query Results	250
TSV Query Results	251
Summary	252

9. RDF Schema, OWL, and Inferencing	253
What Is Inferencing?	254
Inferred Triples and Your Query	256
More than RDFS, Less than Full OWL	257
SPARQL and RDFS Inferencing	258
SPARQL and OWL Inferencing	263
Using SPARQL to Do Your Inferencing	269
Querying Schemas	271
Summary	273
10. Building Applications with SPARQL	275
Applications and Triples	277
Property Functions	277
Model-Driven Development	279
SPARQL and Web Application Development	282
SPARQL Processors	291
Standalone Processors	292
Triplestore SPARQL Support	292
Middleware SPARQL Support	293
Public Endpoints, Private Endpoints	294
SPARQL and HTTP	295
GET a Graph of Triples	298
PUT a Graph of Triples	300
POST a Graph of Triples	300
DELETE a Graph of Triples	301
Summary	301
11. A SPARQL Cookbook	303
Themes and Variations	303
Exploring the Data	306
How Do I Look at All the Data at Once?	306
What Classes Are Declared?	308
What Properties Are Declared?	310
Which Classes Have Instances?	313
What Properties Are Used?	314
Which Classes Use a Particular Property?	316
How Much Was a Given Property Used?	317
How Much Was a Given Class Used?	320
A Given Class Has Lots of Instances. What Are These Things?	321
What Data Is Stored About a Class's Instances?	324
What Values Does a Given Property Have?	326
A Certain Property's Values Are Resources. What Data Do We Have About Them?	328

How Do I Find Undeclared Properties?	330
How Do I Treat a URI as a String?	333
Which Data or Property Name Includes a Certain Substring?	334
How Do I Convert a String to a URI?	336
How Do I Query a Remote Endpoint?	338
How Do I Retrieve Triples from a Remote Endpoint?	339
Creating and Updating Data	341
How Do I Delete All the Data?	341
How Do I Globally Replace a Property Value?	342
How Do I Replace One Property with Another?	343
How Do I Change the Datatype of a Certain Property's Values?	345
How Do I Turn Resources into Instances of Declared Classes?	347
Summary	349
Glossary	351
Index	357

Why Learn SPARQL?

More and more people are using the query language SPARQL (pronounced "sparkle") to pull data from a growing collection of public and private data. Whether this data is part of a semantic web project or an integration of two inventory databases on different platforms behind the same firewall, SPARQL is making it easier to access it. In the words of W3C Director and web inventor Tim Berners-Lee, "Trying to use the Semantic Web without SPARQL is like trying to use a relational database without SQL."

SPARQL was not designed to query relational data, but to query data conforming to the RDF data model. RDF-based data formats have not yet achieved the mainstream status that XML and relational databases have, but an increasing number of IT professionals are discovering that tools that use this data model make it possible to expose diverse sets of data (including, as we'll see, relational databases) with a common, standardized interface. Accessing this data doesn't require learning new APIs because both open source and commercial software (including Oracle 11g and IBM's DB2) are available with SPARQL support that lets you take advantage of these data sources. Because of this data and tool availability, SPARQL has let people access a wide variety of public data and has provided easier integration of data silos within many enterprises. Although this book's table of contents, glossary, and index let it serve as a reference guide when you want to look up the syntax of common SPARQL tasks, it's not a complete reference guide—if it covered every corner case that might happen when you use strange combinations of different keywords, it would be a much longer book.