IS5102
Database Management Systems

Lecture 16: Beyond SQL

Alexander Konovalov

alexander.konovalov@st-andrews.ac.uk

(with thanks to Susmit Sarkar)

2021

Objectives

- The rise of the term NoSQL
- Different non-relational data management systems
- Graph databases
- Document-oriented databases

Beyond Relational DBMS

- Object-oriented
- Semistructured (XML)
- Graph databases
- Key-Value databases
- Document-oriented databases

Relational (SQL) databases

Relational Databases have a long, successful history

Used in a wide variety of operational contexts

SQL is a standard (relational) query language

Not suitable in all situations

Useful when data is tabular

When data is non-tabular, it is less clear

This has **always** been true

New rebranding as **NoSQL** to emphasise additional capabilities

Object-Relational Databases and Mappings

Programming models are often **object-oriented**

Very structured data

- Encapsulation
- Inheritance

Not a good fit for SQL data models

Object-oriented databases (e.g. Versant, db4o) with Object Query Language

Object-relational mapping (e.g. Hibernate from Java)

Semistructured Data: XML

EXtended Markup Language (XML)

- Tree structured (nested)
- Data definition can be changed
- Common interchange format

Some databases can store XML

Several more can produce (and sometimes consume) XML

Useful for web services (and other service-oriented architectures)

Query languages can be defined (XPath, XQuery)

Need for Graph Databases

Much data is now in how things are connected

Social networks are prime example

Value in Relationships

Querying these are hard in relational DBMS

Graph Databases

Graph databases are structured as nodes (like entities) and relationships

Designed for fast querying of **relatedness**-information

Follow the links along

Neo4J most widely used

http://neo4j.com/

Graph Models

Nodes can have different types (like schemas)
So can relationships
And they can all have properties

```
(:Person) -[:LIVES_IN]-> (:City) -[:PART_OF]-> (:Country)
```

Useful in writing queries

```
MATCH (s:Person {name: 'Alexander Konovalov'}) -[:LIVES_IN]-> (e:City)
                     <-[:IS_IN] - (r:Restaurant)
RETURN r.name
```

Key-Value Databases

Often there is minimal formal structure

But huge amounts of data

Associations of **keys** to **values**

One approach: store these associations natively

Allow several indices, ad-hoc queries

Examples: Riak, Apache Cassandra

http://basho.com/products/#riak
http://cassandra.apache.org/

Document-oriented Databases

Store documents

. . . and their **metadata**

Metadata tends to be key-value associations

Examples: CouchDB, MongoDB

http://couchdb.apache.org/
https://www.mongodb.org/

Collections in MongoDB

Documents: MongoDB analogue for what we call **tuples**
Collections: MongoDB analogue for what we call **schema**

Can be (and usually is) nested. Also can be (and usually is) denormalised

Example:

```
{ 'project name': 'Starship',
  'project code': '1',
  'manager' : { 'name : 'Johnston',
    'staff id': '120',
    'phone': '42371' },
  'employees' : [
    { 'name': 'Brown', 'staff id': '108', 'hours': 12 },
    { 'name': 'Brown', 'staff id': '108', 'hours': 20 }
  ]
}
```

NoSQL

Umbrella term for graph, key-value, document (and other?) non-relational DBMS

Emphasise different query models

Analytics driving many of these models

Reading and Consolidation

Chapter 11, *Database System Concepts*, Silbershatz, Korth and Sudarshan

Chapter 33, *Database Systems*, Connolly and Begg