NoSQL DBs and MongoDB

Terminology

DBMS: Database management system

- Software which controls the storage, retrieval, deletion, security, and integrity of data within the database
- Examples: MySQL, mongoDB

RDBMS: Relational database management system

- Relational database stores data in tables
- Organized in columns
- Each column stores one type of data

Terminology

CRUD: basic DB functionality

Create, read, update, delete

Schema:

A method of data modeling; a framework that describes the relationships in your data, how they are stored in tables, and how tables relate to each other

Principles of Relational Databases

Schemas are planned in advance and are relatively static.

- Changes require tacking on new tables and joins, or complete schema overhauls

Data for a single entity can be split among many tables

- Reassembled using link tables and joins

Issues with relational databases

Slow or expensive to reassemble fragmented data quickly

- One machine is best – sometimes must be one extremely large system
- Multiple machines require difficult technical overhead, expertise, and maintenance, vulnerable to downtime in any one piece of the system

Enter: Non-relational databases

NoSQL = “Not Only SQL”

Some examples of NoSQL databases:

- Document databases: mongoDB, couchDB
- Key-value stores: Riak, Voldemort, Redis
- Graph databases: Neo4j, HyperGraph
- Wide-column stores: Cassandra, HBase

mongoDB

Mongo is the most popular NRDBMS / NoSQL database

Source: http://db-engines.com/en/ranking

Mongo concepts

Stores information in *documents* rather than in rows

- Documents are data structures like objects, dictionaries, hashes, maps, associative arrays

MongoDB documents are BSON documents

- JSON = javascript serial object notation
- BSON = binary (javascript) serial object notation

mongoDB document

```
{
    one_field: one_value,
    another_field: [an,
                    array,
                    of,
                    values]
}
```

mongoDB document

```
{
    name: “Sue”,
    age: 20,
    status: “A”,
    groups: [“news”, “sports”]
}
```

SQL vs.mongoDB

SQL table of books:

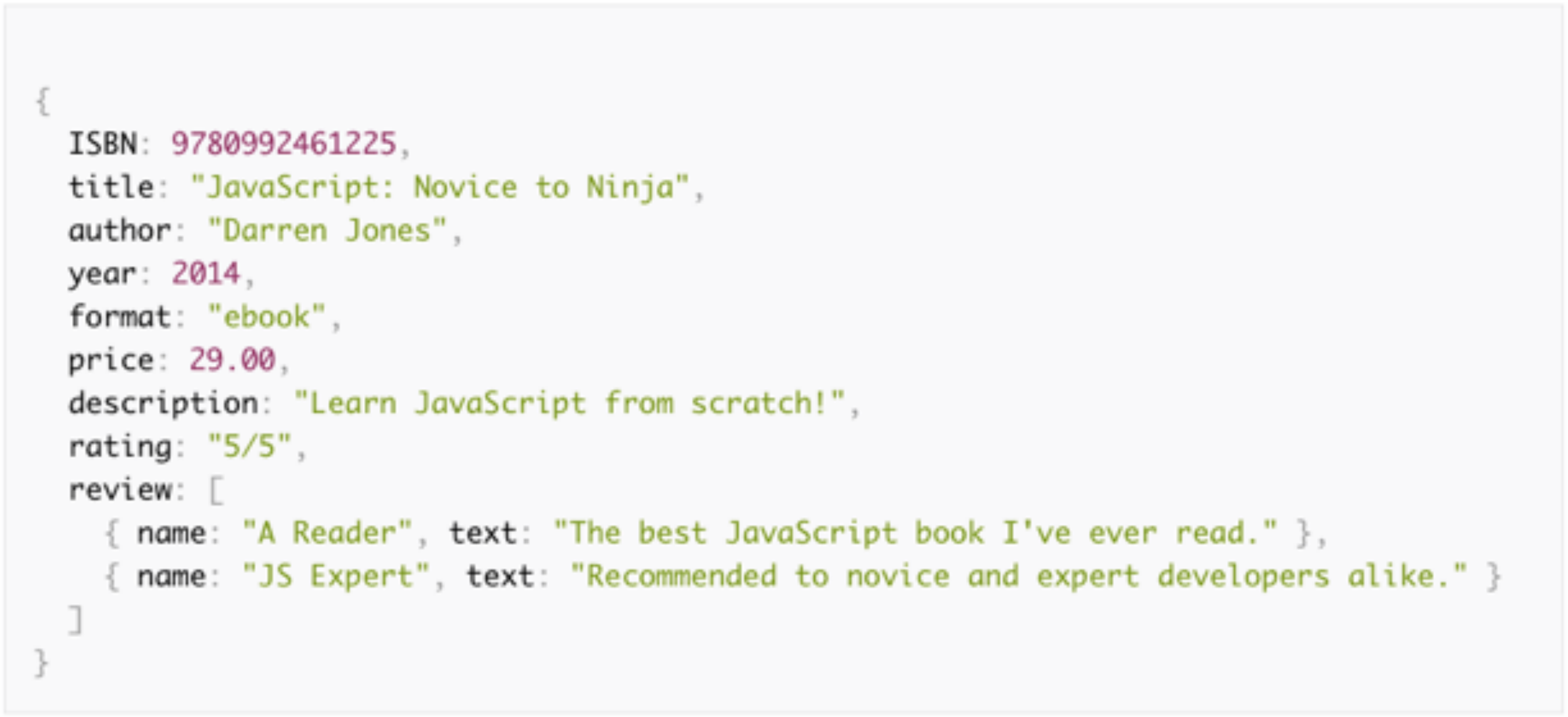
ISBN	title	author	format	price
9780992461225	JavaScript: Novice to Ninja	Darren Jones	ebook	29.00
9780994182654	Jump Start Git	Shaumik Daityari	ebook	29.00

mongoDB book document:

```
{
  ISBN: 9780992461225,
  title: "JavaScript: Novice to Ninja",
  author: "Darren Jones",
  format: "ebook",
  price: 29.00
}
```

SQL vs.mongoDB

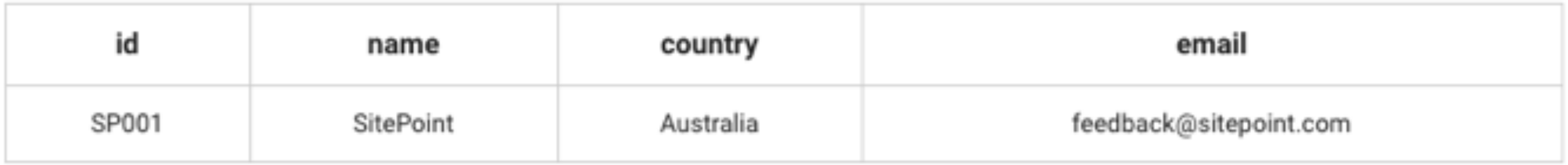
mongoDB more flexible
add fields on the go, mongoDB won’t complain



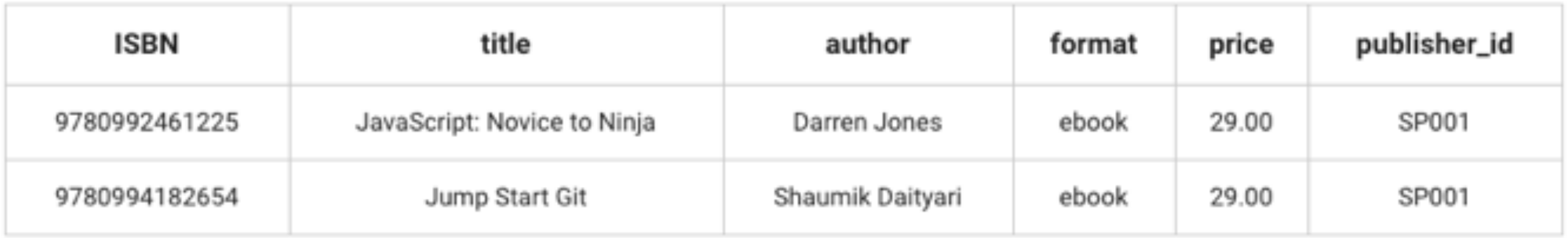
```
{
  ISBN: 9780992461225,
  title: "JavaScript: Novice to Ninja",
  author: "Darren Jones",
  year: 2014,
  format: "ebook",
  price: 29.00,
  description: "Learn JavaScript from scratch!",
  rating: "5/5",
  review: [
    { name: "A Reader", text: "The best JavaScript book I've ever read." },
    { name: "JS Expert", text: "Recommended to novice and expert developers alike." }
  ]
}
```

SQL vs.mongoDB

Add table publisher in SQL:

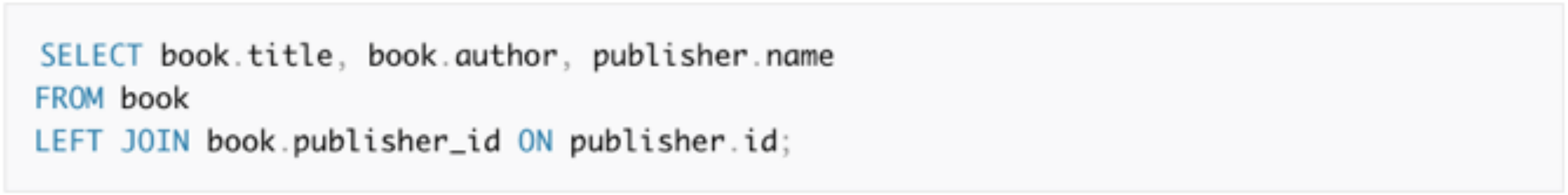


id	name	country	email
SP001	SitePoint	Australia	feedback@sitepoint.com



ISBN	title	author	format	price	publisher_id
9780992461225	JavaScript: Novice to Ninja	Darren Jones	ebook	29.00	SP001
9780994182654	Jump Start Git	Shaumik Daityari	ebook	29.00	SP001

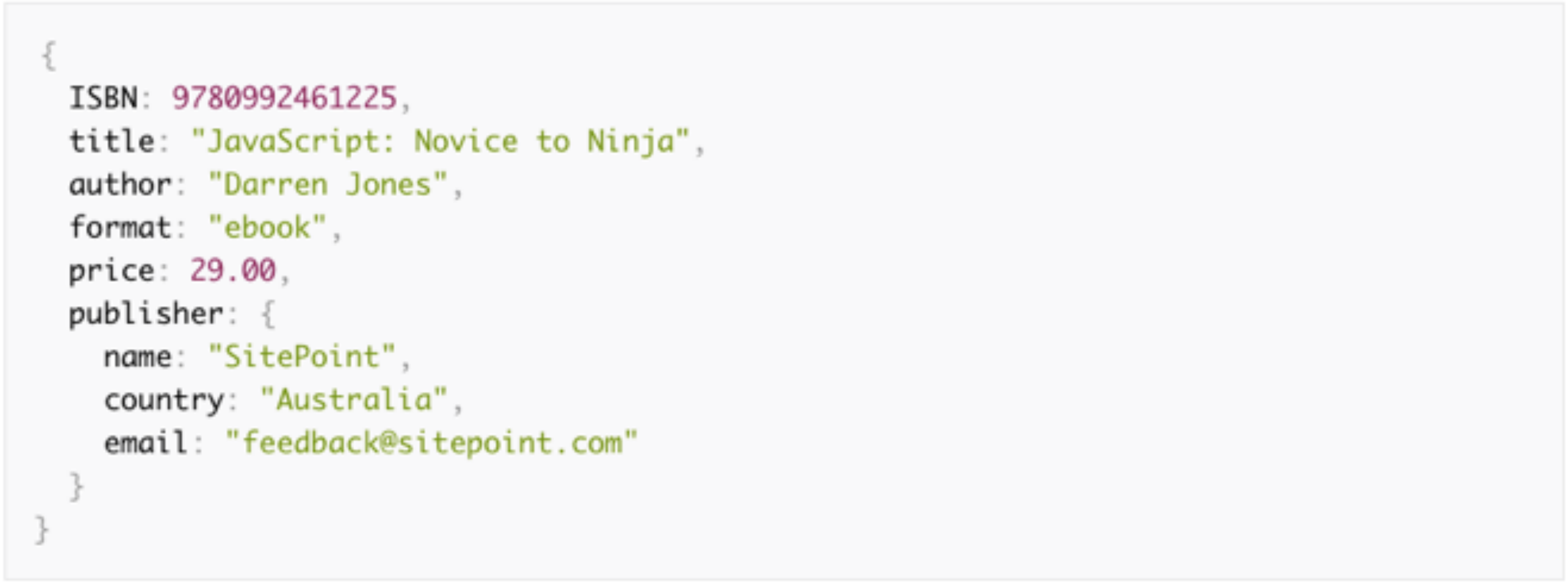
Retrieve publisher for book:



```
SELECT book.title, book.author, publisher.name
FROM book
LEFT JOIN book.publisher_id ON publisher.id;
```

SQL vs.mongoDB

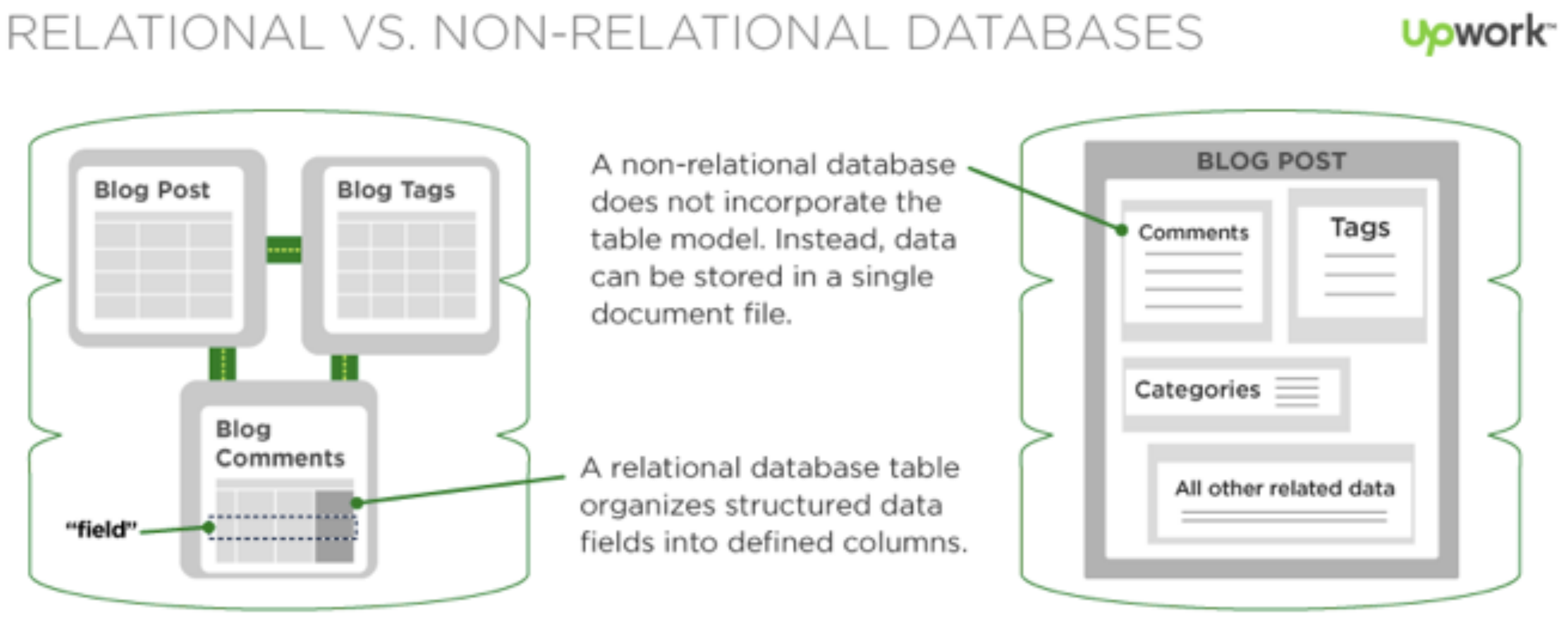
mongoDB document with publisher:



```
{
  ISBN: 9780992461225,
  title: "JavaScript: Novice to Ninja",
  author: "Darren Jones",
  format: "ebook",
  price: 29.00,
  publisher: {
    name: "SitePoint",
    country: "Australia",
    email: "feedback@sitepoint.com"
  }
}
```

SQL vs.mongoDB

SQL: blog post stored across multiple tables
mongoDB: each blog post is one document


RELATIONAL VS. NON-RELATIONAL DATABASES
upwork
Blog Post
Blog Tags
Blog Comments
"field"
A non-relational database does not incorporate the table model. Instead, data can be stored in a single document file.
A relational database table organizes structured data fields into defined columns.
BLOG POST
Comments
Tags
Categories
All other related data

Mongo concepts

Dynamic schemas:

- New fields can be entered on-the-fly
- No enforcement of pre-defined columns

“Horizontal scalability”

- “Sharding”: data may be spread across multiple machines
- Replication and fault tolerance

Mongo concepts

Unstructured data

- Well-suited for holding sloppy information like text, web pages, etc.
- CRUD operations also allow for storage now, structure later

Semi-structured data

Fields in document databases can be:

- added on the fly
- present or absent
- lists, subdocuments (hierarchical), links, etc.

SQL-to-mongo phrasebook

SQL	**Mongo**
database	database
table	collection
row	document
column	field
index	index
table joins	embedded documents / linking

More at: http://docs.mongodb.org/manual/reference/sql-comparison/

Consider using a NoSQL database like MongoDB instead of a Relational Database like MySQL when:

- You don’t have a predetermined schema for your data, and instead need something more flexible
- You don’t really need to do joins between databases from different servers
- Your data is rather large (5-10 GB per table or more if you put it in a SQL database)