MY472 - Week 10
NoSQL and Working with Online Databases
Outline

- Introduction
- SQL vs. noSQL
- Cloud solutions
- Coding session
  - Online database example with SQL: BigQuery
  - NoSQL example: MongoDB
Introduction
Big Data

- Today's data can be very large and also quite complicated.
Database solutions for Big Data

- Different types of databases (SQL vs. NoSQL)
- Cloud solutions using fully managed services
SQL or noSQL?
SQL

- Relational databases have a strict structure

A simple e-commerce example:

![Database Diagram](image-url)
noSQL

- Originally referring to “non SQL”, “non relational” or “not only SQL”
- Provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases
- No strict structure/schema
- noSQL databases are good for data with
  - High **velocity** – Lots of data coming in very quickly
  - High **variety** – Data can be structured, semi-structured, and unstructured
  - High **volume** – Total size of data
  - High **complexity** – Stored in many locations
noSQL types

- **Key Value**
  - Example: Riak, Tokyo Cabinet, Redis server, Memcached, Scalaris

- **Document-Based**
  - Example: MongoDB, CouchDB, OrientDB, RavenDB

- **Column-Based**
  - Example: BigTable, Cassandra, Hbase, Hypertable

- **Graph-Based**
  - Example: Neo4J, InfoGrid, Infinite Graph, Flock DB

From: Simplelern
## noSQL: Pros and Cons

<table>
<thead>
<tr>
<th>PROS</th>
<th>CONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Massive scalability</td>
<td>Limited query capabilities</td>
</tr>
<tr>
<td>High availability</td>
<td>Not standardized</td>
</tr>
<tr>
<td>Schema flexibility</td>
<td>Not matured</td>
</tr>
<tr>
<td>Sparse and semistructured data</td>
<td>Developer heavy</td>
</tr>
</tbody>
</table>
MongoDB

- **Document-based** database
- Mapping of concepts

<table>
<thead>
<tr>
<th>SQL Terms/Concepts</th>
<th>MongoDB Terms/Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>database</td>
</tr>
<tr>
<td>table</td>
<td>collection</td>
</tr>
<tr>
<td>row</td>
<td>document or BSON document</td>
</tr>
<tr>
<td>column</td>
<td>field</td>
</tr>
</tbody>
</table>

- **Reference**
- Each document is constructed as a **BSON** ([Binary JSON](https://en.wikipedia.org/wiki/Binary_Jackson))
- Not UTF-8 string encoded like JSON, but binary - machine readable
- Can store more data types: Dates, separate kinds of numerics (int, float, etc.)
MongoDB documents

A document looks like this:

```javascript
{
    first name: 'Paul',
    surname: 'Miller',
    cell: 447557505611,
    city: 'London',
    location: [45.123, 47.232],
    profession: ['banking', 'finance', 'trader'],
    cars: [  
        {  
            model: 'Bentley',
            year: 1973,
            value: 100000, ... 
        },
        {  
            model: 'Rolls Royce',
            year: 1965,
            value: 330000, ... 
        }
    ]
}
```

From: datawow.io
MongoDB this week

- We will look at MongoDB as an example of a popular noSQL database this week
- We thereby try to replicate basic queries from last week using MongoDB via R with the package `mongolite`
- For a simple selection of documents (i.e. rows in SQL), we will use its `find()` method
- For a bit more sophisticated queries, we will use the `aggregate()` method
- Search queries are in JSON like notation
- Detailed documentation of MongoDB commands and operators
- [Resource 1 (pdf)](#) and [resource 2 (website)](#) for the R package `mongolite`
Cloud solutions
Some exemplary services

<table>
<thead>
<tr>
<th>Database Type</th>
<th>AWS</th>
<th>GCP</th>
<th>Azure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed RDS</td>
<td>Amazon RDS</td>
<td>Cloud SQL</td>
<td>Azure SQL</td>
</tr>
<tr>
<td>Data Warehousing</td>
<td>Redshift</td>
<td>BigQuery</td>
<td>Snowflake</td>
</tr>
<tr>
<td>NoSQL (simple key-value)</td>
<td>DynamoDB</td>
<td>BigTable</td>
<td>Azure Tables</td>
</tr>
<tr>
<td>NoSQL (document)</td>
<td>MongoDB on EC2</td>
<td>MongoDB on GCE</td>
<td>DocumentDB</td>
</tr>
</tbody>
</table>
Google Cloud Platform: BigQuery

- To create and query online databases, we will look at Google BigQuery’s sandbox version as an example
- Database warehouse with other features, used by many financial and commercial companies
- Queried via SQL syntax (API access allows integration with R or Python)
- Scalable to very large databases
- Good documentation
- Many similar databases exist from other providers
Coding session
Files this week

- 01-bigquery-create-own-database.Rmd
- 02-bigquery-examples.Rmd
- 03-mongodb-demo.Rmd