

Week 11: NoSQL and Cloud Databases

LSE MY472: Data for Data Scientists
<https://lse-my472.github.io/>

Autumn Term 2024

Ryan Hübert

Before we begin

Some important announcements

1. Please be sure to archive your course work from GitHub after this course is completed, as the GitHub classroom will be deleted before next autumn
2. The take home assignment will be posted this week and due Wednesday, 15th January 2025 at 5 pm London time
3. You will receive your feedback on the summative problem set (well) before the next due date (have mercy on me!)
4. Please complete the course survey

Outline

- Cloud solutions for databases
- SQL vs. noSQL
- Coding
 - Online database example with SQL: BigQuery
 - NoSQL example: MongoDB

Cloud solutions

Why remote solutions?

- Last week we learned about relational databases
- Worked with SQL to manipulate data stored within tables
- In our applications, the data were **local**
- At scale, we invariably want to store data **remotely**
- Trade-offs, as always!

Some example services

Database Type	Amazon Web Services (AWS)	Google Cloud Platform (GCP)	Microsoft Azure
Managed RDS	Amazon RDS	Cloud SQL	Azure SQL
Data Warehousing	Redshift	BigQuery	Snowflake
NoSQL (simple key-value)	DynamoDB	BigTable	Azure Tables
NoSQL (document)	DocumentDB	MongoDB on GC	Cosmos DB

Google Cloud Platform: BigQuery

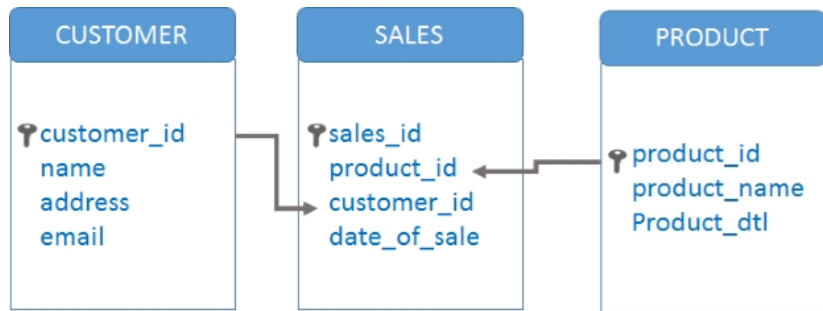
- To create and query online databases, we will look at Google BigQuery's sandbox version as an example
- Database warehouse with other features, used by many financial and commercial companies
- Queried via SQL syntax (API access allows integration with R or Python)
- Scalable to very large databases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

SQL vs noSQL

SQL

→ Relational databases have a strict structure

A simple e-commerce example:




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


Some examples from recent years:

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

© Simplelearn. All rights reserved.

simplelearn

From: **Simplelern**

noSQL: Pros and Cons

PROS	CONS
Massive scalability	Limited query capabilities
High availability	Not standardized
Schema flexibility	Not matured
Sparse and semistructured data	Developer heavy

MongoDB

- **Document-based** database
- Mapping of concepts
- Each document is constructed as a **BSON** (**B**inary **J**SON)
- Not UTF-8 string encoded document
- Like JSON, but binary - machine readable only (very lightweight)
- Can store more data types: Dates, separate kinds of numerics (int, float, etc.)

Reference:

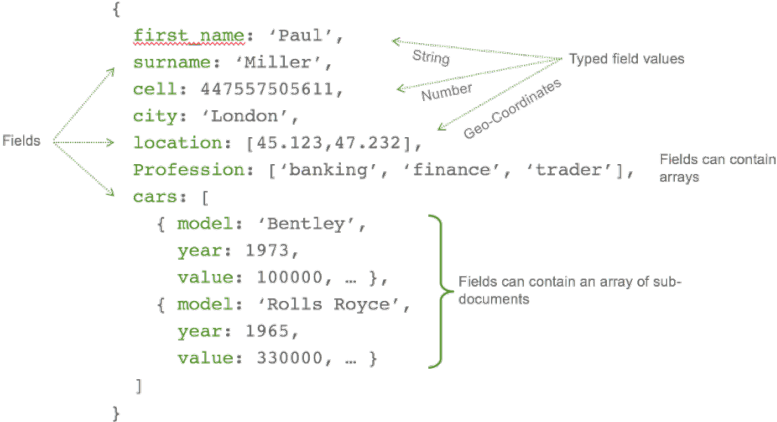
<https://docs.mongodb.com/manual/reference/sql-comparison/>

MongoDB vs. SQL

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document or BSON document
column	field
index	index
table joins	<code>\$lookup</code> , embedded documents
primary key	primary key
Specify any unique column or column combination as primary key.	In MongoDB, the primary key is automatically set to the <code>_id</code> field.

MongoDB documents

A document looks like this (but not the raw file, see [this example](#)):



From: [datawow.io](#)

MongoDB in R (optional)

There is an *optional* script available in the week's materials that shows you how to work with MongoDB in R

The script replicates basic queries from last week using MongoDB via R with the package `mongolite`

- For a simple selection of documents (i.e. rows in SQL), use its `find()` method
- For a bit more sophisticated queries, use its `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`

Coding

Coding

Remote databases:

- [01-bigquery-create-own-database.Rmd](#)
- [02-bigquery-examples.Rmd](#)

NoSQL databases (for your reference only):

- [03-mongodb-demo.Rmd](#)