University of Southampton

Msc Cyber Security

Foundation of Data Science

Data Storage

Author:
Gerard Tio Nogueras

Supervisor:
Dr. Huw Fryer

November 18, 2016

1 Introduction

Write a document of up to 2 pages describing the manner in which you would store the data you downloaded in question 1, for use in a Data Science application, and make mention of the following:

1. The difference between a relational database and a NoSQL database 20%
2. The costs and benefits of using different forms of storage, both in general, and with the particular data here. 20%
3. Conclude which method you would use for the data described in questions 1 and 2, and justify your choice 30%.
4. Presentation, grammar, and good academic practice 20%

This report is longer then asked but I opted for space and clarity against density. I assessed that what I have produced is equivalent to what was asked in number of pages.

2 Content

The choice of the type of storage is not a easy option, multiple arguments have to be taken in account. Here we will discuss them and conclude the best solution for our data.

2.1 General arguments

Main differences between SQL and NoSQL summarized:

- Data storage models: Relational models(SQL) where we have the entries as rows and the data points as columns. NoSQL have multiple options as documents, graphs, key-value and columnar.[McNulty-Holmes2014]
- Normalization(SQL) vs Denormalization(NoSQL): Normalisation helps avoiding replication in the data by separating tables properly and connecting them by ids. For NoSQL Databases we could apply normalization aswell but the CRUD syntax will work better with duplicated data.[Buckler, 2015]
- Relational JOIN vs NoSQL: JOIN is one of the powerfull clauses for SQL databases which allows to combine multiple tables. NoSQL doesn't have the equivalent this is why denormalization is preferred.[Buckler, 2015]
- CRUD(Create, Read, Update & Delete) syntax: Because of its model SQL is very lightweight with great capacities. For NoSQL, simple operations work very well but since information can be nested into deep levels, complexity can rise faster for more complicated queries.[Buckler, 2015]

- Data integrity: SQL databases are able to enforce constraints on any CRUD action, for example make sure some ids match or items can not be removed if a certain id is still present. Unfortunately NoSQL can not enforce any rules concerning integrity, all data is accepted.[Buckler, 2015]
- Tables vs Documents/Data: Tables are only build to accept expected data whereas Documents can easily adapt to unexpected data.[Gentz, 2016]
- Schemas vs schemaless: All the tables must be defined before starting the development of the program and changes can be hard to achieve. With Documents, there is no design, they are dynamic.[Gentz, 2016] [Buckler, 2015]
- Transactions(multiple simultaneous actions concerning the same data): SQL deals with those transactions as if they were a single one to ensure that either both succeed or both fail. NoSQL modifications are atomic wich is equivalent to SQL in this regard but may vary depending on the solution used. [Buckler, 2015] [Gentz, 2016]
- ACID(Atomicity, Consistency, Isolation, Durability): NoSQL will focus on on performance and scalability whereas SQL databases are mostly compliant with ACID.
- Scalability: Here we have vertical scaling(better hardware) for SQL and an horizontal scaling(more nodes) for NoSQL.[Gentz, 2016] [Buckler, 2015] [McNulty-Holmes2014]

2.1.1 Little conclusion to outline where these different types are perfect

To have a good idea on how to choose between both here are projects which one type or the other or perfectly designed for it.
Projects where SQL is ideal:

- Strong logical connections between Data and stable structure
- Important necessity of ACID compliance.
- Moderate growth

Projects where NoSQL is ideal:

- Data on the fly, unspecific data changing rapidly
- High throughput to handle viral growth
- Scalability for fast growth + Fast development
- examples: IoT, Mobile(scaling), Real-Time Analytics(stocks market), content management.

[MongoDB, 2016] [McNulty-Holmes2014] [Buckler, 2015]

2.2 Points of discussion applied to our data

First of all, many of the points cited in the argumentation before won't be of any use since there are many incognitos to what exactly this data is going to be used for and the possible structure of the whole database is unknown. Therefore we will have make a decision only based on what we have and maybe propose different answers depending on assumption for the use of the data.

Our first data concerns authorities, establishments and the second one concerns health institutions.
Let us analyse what would suite the data from the 1st question. Trying to apply the major factors to chose which type fits best: The type of data is pretty static, it won't scale since there a fixed number of those entities, there is an IdCode which can be used to link all the tables if needed after normalization. I think an SQL database would fit best for our first question. Because even if we think of the different possibilities

For the 2nd question we are facing dynamic data, the number of Address lines might differ from one to another. Then this is data which could probably be used for mobile applications and therefore scalability might be a factor to take into account. Some of the values are objects which fits a NoSQL database as wels as the other features mentioned. This is why a NoSQL database seems more fit for the type of data used in the 2nd question.

References

[Fryer, 2016] Fryer, H. (2016) Data Management Available at http://www.edshare.soton.ac.uk/17507/ (Accessed 18 November 2016)

[Wenzel, 2014] Wenzel, K. (2014) Database normalization explained in simple English. Available at: http://www.essentialsql.com/get-ready-to-learn-sql-database-normalization-explained-in-simple-english/ (Accessed: 18 November 2016).

[McNulty-Holmes2014] McNulty-Holmes, E. (2014) :SQL vs. NoSQL- what you need to know. Available at: http://dataconomy.com/sql-vs-nosql-need-know/ (Accessed: 18 November 2016).

[Buckler, 2015] Buckler, C. (2015) SQL vs NoSQL: The differences. Available at: https://www.sitepoint.com/sql-vs-nosql-differences/ (Accessed: 18 November 2016).

[Gentz, 2016] Gentz, M. (2016) When to use NoSQL vs SQL. Available at: https://docs.microsoft.com/en-gb/azure/documentdb/documentdb-nosql-vs-sql (Accessed: 18 November 2016).

[MongoDB, 2016] MongoDB (2016) Giant ideas brought to life. Available at: https://www.mongodb.com/use-cases (Accessed: 18 November 2016).