BIRMINGHAM CITY
University

CMP6207 - Assignment 1

IOThings Application Report

Lewis Higgins - Student ID 22133848

CMP6207 - Modern Data Stores

Module Coordinator: Konstantinos Vlachos

Contents

Introduction

The report will be based on the design and implementation of a MongoDB NoSQL database system for the company "IoThings Home Automation Solutions". There are two assignments in this module, where this report is worth **60%**, and a presentation is the remaining **40%**. This module will also incorporate elements of web design, with HTML and CSS also playing a role alongside the primary use of JavaScript.

- Konstas cares more for the functionality over aesthetics. While you should put effort into the HTML and CSS parts, they're a lesser concern than the overall usability of the system.
- A literature review is expected in this report.
- 4,000 word count, so 4,400 hard limit.
- The presentation is about this report. It must cover the design, implementation and data management.
  - You would show him your database cluster (local or Atlas?) as part of it.
- IT WILL BE THIRTY MINUTES. If the report contains all the screenshots (which it will) you might not need to make slides, though consider it anyway. You can still show the report alongside the slides where needed.
- He wants you to make an Atlas account even if you do it locally.
- You are creating the dataset for this assignment. You will describe it in an appendix rather than the report's main body.
- This is a "professional report", and as such the title page with the BCU logo and your info should probably change. It needs to also show the date.

Types of NoSQL databases

Structured Query Language, or SQL, was developed by IBM following Codd (1970)'s groundbreaking publication in the ACM journal, with the first commercial SQL implementation being published by Oracle in 1979 (Oracle, 2025). SQL powers many relational database systems even today, though the problems associated with its age, most notably in the speed of its operations, are beginning to show in modern systems. Therefore, NoSQL ("Not Only SQL") was developed as an extension of SQL, allowing data to be stored in a non-tabular, non-relational format for efficient storage of semi-structured and unstructured data in a flexible, functional and scalable model for faster operations than standard relational databases in most scenarios (Google Cloud, 2025; AWS, 2025b). There are a wide variety of NoSQL database types which vary in complexity, functionality and purpose, meaning that identification of the most suitable type is paramount for maximum efficiency.

1.1 Document database

Document databases are intuitive, flexible and horizontally scalable databases that work well in a wide variety of use cases for both transactional and analytical purposes, including IoT data and real-time analytics (MongoDB, 2025a). They store records as "documents", which store an object's data and metadata, in a format such as JSON, BSON, or XML[1]. Details and examples of these file types can be found in Appendix A.

Figure 1.1: An example of a JSON Document.

[1]JavaScript Object Notation, Binary JSON and Extensible Markup Language, respectively.

Figure 1.1 depicts an example JSON document in MongoDB[2], a popular DBMS for document databases. It contains the data of a singular example school teacher, storing details such as their name, age and subject expertise, as well as a unique internal object ID used by MongoDB to identify that document. The data itself is of varying types including strings, integers and arrays, which makes document databases easily integrable into a development workflow due to the direct storage of object types used in programming languages like Python and JavaScript.

Popular services for document databases include Databricks (Databricks, Fri, 10/13/2023 - 11:57) Couchbase (Couchbase, 2025), and the previously mentioned MongoDB (MongoDB, 2025a).

1.2 Key-value database

Key-value databases are primarily reputed for their speed and simplicity, functioning by storing each record as a key-value pair. Rather than having to search through massive amounts of irrelevant data for a query, key-value databases can instead search through their stored keys to retrieve results within milliseconds or even microseconds if used in-memory (Redis, 2025a). While this is excellent for simple queries to retrieve specific known records, this same property also causes major limitations in that retrieving data based on values, such as finding all users over a given age, would require the entire database to be searched, making key-value databases best suited for real-time data access and caching where simpler queries are used (MongoDB, 2025b).

Notable software options used across industry for key-value databases include Amazon's DynamoDB (AWS, 2025a) and Redis (Redis, 2025b).

1.3 Column-oriented database

1.4 Graph database

[2]MongoDB actually stores data as BSON, though it translates between the two when queried.

Comparing NoSQL and Relational databases

- This section "shouldn't be too long, but enough to convince them."
- **This is NOT comparing *MongoDB* to relational databases.** The brief mentions that it should be "generic just in case after seeing your MongoDB database they decide to go with another software provider."

Database design and implementation

MongoDB, an acronym from "hu**mongo**us **DB**", aims to address some of SQL's antiquation issues...

API Implementation and Documentation

Conclusion

Overall, something was done...

Bibliography

AWS (2025a). *Fast NoSQL Key-Value Database – Amazon DynamoDB – AWS*. Amazon Web Services, Inc. URL: https://aws.amazon.com/dynamodb/ (visited on 17/02/2025).

AWS (2025b). *What Is a NoSQL Database? - Nonrelational Databases Explained - AWS*. Amazon Web Services, Inc. URL: https://aws.amazon.com/nosql/ (visited on 04/02/2025).

Codd, E. F. (1st June 1970). 'A Relational Model of Data for Large Shared Data Banks'. In: *Commun. ACM* 13 (6), pp. 377–387. ISSN: 0001-0782. DOI: 10.1145/362384.362685.

Couchbase (2025). *Couchbase: Best Free NoSQL Cloud Database Platform*. Couchbase. URL: https://www.couchbase.com/ (visited on 17/02/2025).

Databricks (Fri, 10/13/2023 - 11:57). *The Data and AI Company*. Databricks. URL: https://www.databricks.com/ (visited on 17/02/2025).

Google Cloud (2025). *What Is NoSQL? Databases Explained*. Google Cloud. URL: https://cloud.google.com/discover/what-is-nosql (visited on 04/02/2025).

MongoDB (2025a). *Document Database - NoSQL*. MongoDB. URL: https://www.mongodb.com/resources/basics/databases/document-databases (visited on 16/02/2025).

MongoDB (2025b). *What Is A Key-Value Database?* MongoDB. URL: https://www.mongodb.com/resources/basics/databases/key-value-database (visited on 17/02/2025).

Oracle (2025). *History of SQL*. URL: https://docs.oracle.com/cd/B13789_01/server.101/b10759/intro001.htm (visited on 04/02/2025).

Redis (2025a). *Redis FAQ*. Docs. URL: https://redis.io/docs/latest/develop/get-started/faq/ (visited on 17/02/2025).

Redis (2025b). *What Is a Key-Value Database?* Redis. URL: https://redis.io/nosql/key-value-databases/ (visited on 17/02/2025).

Appendix A - Data types

The types of data stored in databases vary dependent on their file format. Section 1.1 referred to three major file formats used in document databases - JavaScript Object Notation (JSON), Binary JSON (BSON), and Extensible Markup Language (XML). Each of these have their own distinct attributes, which will be thoroughly described in this appendix.

A.1 JavaScript Object Notation (JSON)

JSON files are...

A.2 Binary JSON (BSON)

BSON files are...

A.3 Extensible Markup Language (XML)

XML files are not common in many databases in recent times. However, they do still have some benefits not found in JSON or BSON, such as the creation of Document Type Definitions (DTDs) where data can be validated as it is retrieved by forming constraints such as what elements exist or what attributes an element can or must have. DTDs do not allow for constraining data types (i.e. String, Integer, Float), which is a feature instead provided by XML Schemas, an updated schema language that addresses DTD drawbacks through allowing not only type definitions but also enforcing uniqueness and foreign key constraints as well as inheritance. Additionally, XML schemas are written using XML syntax unlike DTDs. The only considerable drawback of XML schemas are the complexity of writing them, as they can be considerably more complicated than DTDs.