Research Document

Name: Tony Jiang

Semester: 6

Project Music trivia game

Contents

1 Introduction

2 Research questions

Main question:

What kind of storage solution fits for storing and using large files in a music trivia web-based game?

Sub-questions:

The strategy and methodology for the FHICT can be found at this link: https://ictresearchmethods.nl/ Methods and https://cmdmethods.nl/.

1. **Which databases should be considered for a comparison test?**
    - **Strategy: Library**
    - **Methodology: Literature Study, Community Research**

2. **Which of the chosen databases is easy to implement on the web?**
    - **Strategy: Field, Library**
    - **Methodology: Document Analysis, Community Research**

3. **What is the storage size of the two chosen databases for the free version?**
    - **Strategy: Field, Workshop**
    - **Methodology: Document Analysis, Prototyping**

4. **Among the chosen databases, which one offers greater scalability for free?**
    - **Strategy: Library, Workshop, Showroom**
    - **Methodology: Literature Study, Prototyping, Gap Analysis, Benchmark Test**

5. **How does the performance of the chosen databases compare?**
    - **Strategy: Library Workshop Showroom**
    - **Methodology: Literature Study, Prototyping, Gap Analysis, Benchmark Test**

3 Sub-questions result

3.1 Which databases should be considered for a comparison test?

I need to decide which databases to compare for this project due to time constraints. The ideal criteria for selecting databases to compare include:

- The database must be popular.
- The database must have a free version available.
- The databases must be of different types from each other.
- The database must have a cloud-based option.

Additionally, I'll limit the maximum number of comparisons to three.

3.1.1 Types of databases

I'm going to write down all the databases type that I found and summarize them and decide which types of databases to do a comparison test based on this project.

- **Relational databases**

Relational databases are a type of database that store and provide access to data points that are related to one another. They are based on the relational model, an approach to managing data using a structure and language consistent. Relational databases are fundamental in many applications and industries due to their structured approach to data management and robust querying capabilities.

The relational databases are:

1. MySQL
2. PostgreSQL
3. MariaDB
4. SQLite

- **NoSQL databases**

NoSQL databases (Not Only SQL databases) are a category of database management systems that do not adhere strictly to the traditional relational database model. They are designed to handle a wide variety of data models, including key-value, document, column-family, and graph formats. NoSQL databases are particularly well-suited for large-scale data storage and for applications requiring flexible, scalable, and high-performance data management.

NoSQL databases are increasingly popular due to their ability to handle diverse data types, scale efficiently, and provide high performance for specific use cases.

Document-oriented Databases:

MongoDB

CouchDB

Key-Value Stores:

Redis

Amazon DynamoDB

Riak

Column-family Stores:

Apache Cassandra

HBase

Graph Databases:

Neo4j

Amazon Neptune

ArangoDB

- **Object-oriented databases**

Object-oriented databases (OODBMS) are databases that integrate object-oriented programming principles with database technology to store and manage complex data. They are designed to handle data as objects, similar to how data is managed in object-oriented programming languages. This approach aims to provide a seamless integration between application programming and database management, allowing objects to be stored persistently without the need for conversion to a relational format.

They object-oriented databases are:

1. SQLite
2. Berkeley DB
3. H2

- **Cloud databases**

Cloud databases are databases that run on cloud computing platforms, providing various advantages such as scalability, accessibility, and reduced infrastructure management overhead. They can be fully managed by cloud service providers or partially managed by the user. Cloud databases can follow different database models, including relational, NoSQL, and others, and they offer the flexibility to meet diverse application needs.

The cloud databases are:

1. Amazon RDS
2. Google Cloud Spanner
3. Microsoft Azure SQL Database

- **Embedded databases**

Embedded databases are database management systems (DBMS) that are integrated directly into an application, providing data storage and retrieval functionality within the application itself. Unlike traditional DBMSs, which run as separate services, embedded databases run as a part of the application, eliminating the need for separate database management processes. They are designed to be lightweight, fast, and require minimal configuration, making them ideal for use in applications where performance and resource constraints are critical.

The embedded databases are:

1. SQLite
2. Berkeley DB
3. H2

3.1.2 Chosen databases

The two databases I will be testing are MySQL and MongoDB. Both are very popular. MySQL is widely used by many applications and industries, while MongoDB is known for its scalability, flexibility, and high-performance data management, making it ideal for scaling the project's data.

3.2 Which of the chosen databases is easy to implement on the web?

The 2 chosen databases are MySQL and MongoDB. The project's environment uses Java and Gradle. I need to determine which database, between MySQL and MongoDB, is easier to implement in a Java and Gradle environment.

3.2.1 MySQL

Java has built-in support for JDBC (Java Database Connectivity), a standard API for connecting to and executing queries with relational databases like MySQL. There are well-established libraries and Gradle dependencies for MySQL, such as “mysql-connector-java.” Additionally, MySQL requires a predefined schema, meaning you need to define tables and their relationships before inserting data.

It is easy to make a connection from Java and Gradle to a MySQL database. Gradle has dependencies that facilitate creating a schema in the database from the project and make database connections easy.

3.2.2 MongoDB

MongoDB provides a Java driver that allows you to interact with the database. This is not as standardized as JDBC but is well-documented and easy to use. You can use the official MongoDB Java driver or Spring Data MongoDB for easier integration. Additionally, MongoDB doesn't require a predefined schema. You can store documents with different structures in the same collection.

MongoDB is also easy to implement with the provided Gradle dependencies. You can refer to the official MongoDB documentation for instructions on integrating it into a Java and Gradle project. However, if you are new to MongoDB, it might take some time to establish the connection.

3.2.3 Summary

MySQL and MongoDB are both easy to implement in a Java and Gradle project using the provided Gradle dependencies for each database. However, in school, you typically learn about relational databases first, such as MySQL and Microsoft SQL Server. This means you already have some familiarity with how to establish connections from your project. On the other hand, MongoDB requires you to learn it independently and refer to the documentation. Once you get the hang of it, implementing MongoDB becomes straightforward.

3.3 What is the storage size of the two chosen databases for the free version?

The size of the community editions for MySQL and MongoDB Community Server has no specific limit; it only depends on server configuration and hardware resources. If you are using the cloud version of these databases, there may be limits for the free version.

3.3.1 MySQL cloud version

For the cloud version of MySQL, we can consider cloud providers like AWS (Amazon Web Services). They offer a free tier of MySQL RDS (Relational Database Service), which includes 750 hours of db.t2.micro instances per month for up to 12 months, with a storage limit of up to 20 GB.

3.3.2 MongoDB cloud version

For the cloud version of MongoDB, MongoDB offers MongoDB Atlas (Cloud-hosted). They provide a free tier known as the M0 cluster, which includes 512 MB of storage.

3.3.3 Summary

The community versions of MySQL and MongoDB have no specific storage limits. However, the cloud versions of MySQL and MongoDB impose limits on data storage. MongoDB's cloud version offers 512 MB of data storage capacity, which may be restrictive for extensive data storage tests.

3.4 Among the chosen databases, which one offers greater scalability for free?

3.5 How does the performance of the chosen databases compare?

4 Conclusion to the main question