Abstract

Semistructured data is on rise, with some people embracing its use. Its data characteristics that is flexible in schema definition, makes it easy for developer to answer application demand faster. Flexibility schema that is given is also appropriate for certain use case, especially where given domain data is not relational.

One of the popular semi-structured database model are document-oriented database. On this database, the data and its structure is encapsulated in a document. One reason, is to better fit in the data model of the real world, where data is not yet normalized. Concepts and structures where data is encapsulated in a single document is known as self-contained data.

In addition to differences in data models, document databases also has the distinction of doing its query. Because the database is not yet standardized, and is still actively developed by various parties, then the query model are different from one another. Several document databases using a declarative language, while others use a procedural language. One of the procedural models will be studied in this final project, is MapReduce.

In this final Project, data migration of employees schema will be done, which was using relational data, to document database. Migration is done to better understand the differences between the relational model, and document model. After that, benchmark will be done to measure the performance differences between relational databases and documents.

The result of this Final Project is application that is capable of doing migration from relational table to document with three different model: Self Contained, Normalized, and Mixed. In addition, the results of testing conducted found that the flexibility of the data semistrukutral has impact on larger disk space consumption, index creation time that is longer, as well as limitations in terms of query that can be done through a MapReduce model.

Key words : *Semi Structured Data, Data Migration, MapReduce, Document Oriented Database, Relational Database, Database Benchmark*.