COUCHDB - INTRODUCTION

http://www.tutorialspoint.com/couchdb/couchdb_introduction.htm

Database management system provides mechanism for storage and retrieval of data. There are three main types of database management systems namely −

- RDBMS
- NoSQL

RDBMS

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

A Relational database management system *RDBMS* is a database management system *DBMS* that is based on the relational model as introduced by E. F. Codd.

The data in RDBMS is stored in database objects called **tables**. The table is a collection of related data entries and it consists of columns and rows. It stores only structured data.

NoSQL Databases

A NoSQL database *sometimescalledasNotOnlySQL* is a database that provides a mechanism to store and retrieve data other than the tabular relations used in relational databases. These databases are schema-free, support easy replication, have simple API, eventually consistent, and can handle huge amounts of data *bigdata*.

The primary objective of a NoSQL database is to have the following −

- Simplicity of design,
- Horizontal scaling, and
- Finer control over availability.

NoSQL databases use different data structures compared to relational databases. It makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it must solve. These databases store both structured data and unstructured data like audio files, video files, documents, etc. These NoSQL databases are classified into three types and they are explained below.

Key-value Store − These databases are designed for storing data in key-value pairs and these databases will not have any schema. In these databases, each data value consists of an indexed key and a value for that key.

Examples − BerkeleyDB, Cassandra, DynamoDB, Riak.

Column Store − In these databases, data is stored in cells grouped in columns of data, and these columns are further grouped into Column families. These column families can contain any number of columns.

Examples − BigTable, HBase, and HyperTable.

Document Store − These are the databases developed on the basic idea of key-value stores where "documents" contain more complex data. Here, each document is assigned a unique key, which is used to retrieve the document. These are designed for storing, retrieving, and managing document-oriented information, also known as semi-structured data.

Examples − CouchDB and MongoDB.

What is CouchDB?

CouchDB is an open source database developed by Apache software foundation. The focus is on the ease of use, embracing the web. It is a NoSQL document store database.

It uses JSON, to store data *documents*, java script as its query language to transform the documents, http protocol for api to access the documents, query the indices with the web browser. It is a multi master application released in 2005 and it became an apache project in 2008.

Why CouchDB?

- CouchDB have an HTTP-based REST API, which helps to communicate with the database easily. And the simple structure of HTTP resources and methods *GET, PUT, DELETE* are easy to understand and use.
- As we store data in the flexible document-based structure, there is no need to worry about the structure of the data.
- Users are provided with powerful data mapping, which allows querying, combining, and filtering the information.
- CouchDB provides easy-to-use replication, using which you can copy, share, and synchronize the data between databases and machines.

Data Model

- Database is the outermost data structure/container in CouchDB.
- Each database is a collection of independent documents.
- Each document maintains its own data and self-contained schema.
- Document metadata contains revision information, which makes it possible to merge the differences occurred while the databases were disconnected.
- CouchDB implements multi version concurrency control, to avoid the need to lock the database field during writes.

Features of CouchDB

Document Storage

CouchDB is a document storage NoSQL database. Documents are the primary unit of data, each field is uniquely named and contain values of various data types such as text, number, Boolean, lists, etc... In these documents there is no set limit to text size or element count.

CouchDB provides an API called RESTful HTTP API for reading and updating *add, edit, delete* database documents.

Below given is the sample JSON document structure followed by couchDB..

```
{
   "field" : "value",
   "field" : "value",
   "field" : "value",
}
```

ACID Properties

The CouchDB file layout and commitment system features all the ACID properties.

Once the data is entered in to the disc it will not be overwritten. Document updates *add, edit, delete* follow Atomicity, i.e., they will be saved completely or not saved at all. The database will not have any partially saved or edited documents.

Almost all of these update are serialized and any number of clients can read a document without waiting and without being interrupted.

Compaction

Compaction is an operation to avail extra disc space for the database by removing unused data. While performing compaction operation on a particular file, a file with the extension **.compaction** is created and all the active/actual data is copied *cloned* to that file, when the copying process is finished then the old file is discarded. The database remains online during the compaction and all updates and reads are allowed to complete successfully.

Views

Data in CouchDB is stored in semi-structured documents that are flexible with individual implicit structures, but it is a simple document model for data storage and sharing. If we want see our data in many different ways, we need a way to filter, organize and report on data that hasn’t been decomposed into tables.

To solve this problem, CouchDB provides a view model. Views are the method of aggregating and reporting on the documents in a database, and are built on-demand to aggregate, join and report on database documents. Because views are built dynamically and don’t affect the underlying document, you can have as many different view representations of the same data as you like.

History

A brief of CouchDB's past is as follows −

- CouchDB was written in Erlang programming language.
- It was started by Damien Katz in 2005.
- CouchDB became an Apache project in 2008.

The current version of CouchDB is 1.61.

Loading [MathJax]/jax/output/HTML-CSS/jax.js