Lecture 6a: MongoDB Store

History

- MongoDB was created by Eliot and Dwight (founders of DoubleClick) in 2007 when they faced scalability issues while working with relational databases.
- The organization that developed MongoDB was originally known as 10gen. In 2009 they changed their business model and released MongoDB as an open source Project. In 2013, they changed their name to MongoDB Inc.

Features

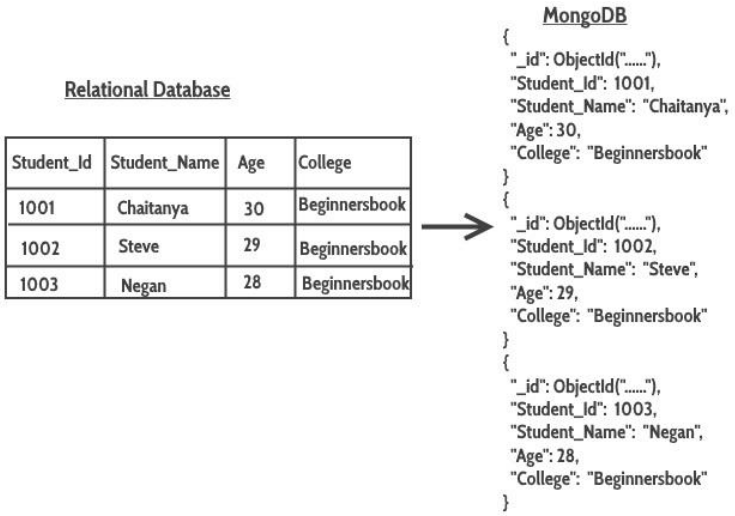
- High performance – most of the operations in the MongoDB are faster compared to relational databases
- Auto Replication – allows you to quickly recover data in case of a failure
- Scalability – Horizontal scaling is possible in MongoDB because of sharding. Sharding is partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved
- Load balancing – horizontal scaling allows MongoDB to balance the load
- High Availability – Auto replication improves the availability of MongoDB database
- Indexing – index is a single field within the document. Indexes are used to quickly located data without having a search every document in a MongoDB database. This improves the performance of operations performed on the MongoDB database.

MongoDB vs. RDBMS

- Advantages of MongoDB over Relational Databases
  - Schema less − MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
  - Structure of a single object is clear.
  - No complex joins.
  - Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
  - Tuning.
  - Ease of scale-out − MongoDB is easy to scale.
  - Conversion/mapping of application objects to database objects not needed.
  - Uses internal memory for storing the (windowed) working set, enabling faster access of data
- MongoDB data model is document-oriented which means it stores its information in documents rather than rows
- Collections in MongoDB is equivalent to the tables is RDBMS
- Documents in MongoDB is equivalent to the rows in RDBMS
- Fields in MongoDB is equivalent to the columns in RDBMS
- Fields (key and value pairs) are stored in document, documents are stored in a collection and collections are stored in a database.

Relational Database	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Table Join	Embedded Documents
Primary key	Primary Key (default key_id is provided by MongoDB itself)
Oracle	MongoDB
Sqlplus	Mongo

- Example of a relational table and MongoDB representation of the same data



- MongoDB automatically inserts a unique _id(12-byet field) field in every document, this serves as primary key for each document
- MongoDB support dynamic schema which means one document of a collection can have 4 fields while the other document has only 3 fields. This is not possible in relational database.
- Document:
  - A document is a set of property names and their values. The values can be simple data types such as strings, numbers and dates, arrays and even other documents.
  - Documents have dynamic schema that is documents in the same collection do not need to have the same set of fields or structure
  - Common fields in a collection's documents may hold different types of data
  - A document can be thought of as a row in a relational table
  - The document contains all the information about the object so for example if you are storing information about "Peter" his name, address, cell, and his email. But what if he has more than one email, in a relational database you would have to create a separate table and store the email along with some ID Peter, so that the table is normalize. Then you would need to join the tables through SQL code in order to retrieve all the information about Peter.

- However, in MongoDB you can put all the information about Peter in one document and you don't have to worry about repetition or normalization.
- With a document model all the information belonging to the object can be retrieved at once so that you have the complete picture. No joins are required.
- With MongoDB you can query the data in an SQL like language.
- MongoDB stores documents in a format called Binary JSON or BSON
- Example of a document:
  {name:"al", age:18, status:"d", groups:["politics","news"]}

- Collection
  - A collection in MongoDB is a group of documents.
  - All documents in a collection can have a similar or related purpose.
  - A collection in MongoDB can be thought of as a table in a relational database
  - Collections do not enforce a schema. Collections have dynamic schemas which means that the documents within a single collection can have any number of different "shapes." For example, both of the following documents can be stored in the same collection even though one contains string and the other string and integer:
    - {"greeting":"Hello, world!"}
    - {"hi": 5}
  - Naming conventions for collections
    - Empty string not a valid collection name
    - May not contain a space
    - Cannot use system as a name
    - Cannot use $
  - Example of a collection:
    {name:"al", age:18, status:"d", groups:["politics","news"]}
    {name:"di",age:31, status:"r"}
    {name:"jim", age:45,status:"I", groups:"news"}
- Database
  - A database in MongoDB is a physical container for collections.
  - A single instance of MongoDB (a single MongoDB server) can host several databases each grouping together zero or more collections
  - A database has its own permission and each database is stored in separate files on disk
  - A good rule of thumb is to store all data for a single application in the same database
  - Separate databases are useful when storing data for several application or users on the same MongoDB server
  - Naming convention for databases
    - The empty string is not a valid database name
    - Cannot contain special characters just alphanumeric
    - Database names are case sensitive
    - Limited to 64 bytes
    - Cannot use admin, local, config as names

- Data Types used in MongoDB
  - Strings – most common datatype. Alphanumeric
  - Integer – store a numerical value can be 32 bit or 64 bit
  - Nulls –can be used to represent both a null value and a nonexistent field
  - Boolean –can be used for the values true and false
  - Double – 64-bit floating point number (whole number and decimal)
  - Date – used to store the current data or time in UNIX time format. You can specify your own date time by creating object of data and passing day, month, year into it
  - ObjectID – used to store the document's ID
  - Array – sets or lists of values are represented as arrays
  - Timestamp – record when a document has been modified or added
  - Embedded document – documents can contain entire documents embedded as values in a parent document
  - Code – can contain JavaScript code

<u>MongoDB logistics</u>

- MongoDB is written in C++
- Data is stored and queried in BSON – binary serialized JSON-like data
- The project compiles on all major operating systems including Mac OS X, Windows, Solaris and most flavors of Linux
- MongoDB is open source
- The source code is freely available on the MongoDB website for download https://www.mongodb.com/download-center/community
- You can use MongoDB for free online but you cannot store any data https://docs.mongodb.com/manual/tutorial/query-documents/
- The project is guided by the MongoDB Inc. core server team
- Why use MongoDB
  - Document Oriented Storage − Data is stored in the form of JSON style documents.
  - Index on any attribute
  - Replication and high availability
  - Auto-sharding
  - Rich queries
  - Fast in-place updates
  - Professional support by MongoDB
- Where is MongoDB Used
  - Big Data
  - Content Management and Delivery
  - Mobile and Social Infrastructure
  - User Data Management
  - Data Hub
- MongoDB Use Cases
  - The Business Insider (TBI)
    - has used MongoDB since 2008.

    - TBI is a news site serving more than a million unique page views per day
    - MongoDB handles the site's main content posts, comments, users as well as real-time analytics data
    - the analytic data is used to generate dynamic heat maps indicating click-through rates for the various news stories
  - Other businesses that use MongoDB include Shutterfly and The New York Times

Important MongoDB Concepts

- Replication
  - MongoDB provides database replication via a topology known as a replica set
  - Replica sets distribute data across two or more machines for redundancy and automate failover in the event of server and network outages
  - Replica sets consist of many MongoDB servers, usually with each server on a separate physical machine; which are called nodes
  - At any given time, one node serves as the replica set primary node and one or more nodes serve as secondaries
  - The primary can accept both read and write operations but the secondaries only read
  - If the primary node fails, the cluster will pick a secondary node and automatically promote it to primary and when the former primary comes back it will become the secondary
  - Replication is one of MongoDB's most useful features
- Journaling
  - MongoDB guarantees every write is flushed to the journal file every 100 ms. If the server is ever shut down uncleanly say in a power outage, the journal will be used to ensure that MongoDB data files are restored to a consistent state when you restart the server.
- Scaling
  - database systems with large data sets and high throughput applications can challenge the capacity of a single sever. To address this issue of scale, database systems have two basic approaches; vertical scaling or Sharding/Horizontal scaling
    - Vertical scaling: it adds more CPU and storage resources to increase capacity. But such arrangements are disproportionately expensive. As a result, there is a practical maximum capability for vertical scaling.
    - Sharding or Horizontal Scaling: by contrast, it divides the data set and distributes the data over multiple servers-shards. Each shard is an independent database and collectively shards make up a single database.
  - MongoDB supports sharding through the configuration of sharded clusters.
  - Shards are used to store the data
  - Query Routers or mongos instances interface with client applications and direct operations to the appropriate shard or shards and then return results to the clients.
  - Config servers stores the cluster's metadata. This data contains a mapping of the cluster's data set to the shards. The query router uses this metadata to target operations to specific shards.
  - MongoDB distributes data at the collection level.

- Sharding partitions a collection's data by the shard key
- A shard key is either an indexed field or an indexed compound field that exists in every document in the collection
- MongoDB divides the shard key values into chunks and distributes the chunks evenly across the shards

- Indexes
  - Every document gets a default index on the _id attribute which enforces uniqueness.
  - Indexes can be set on any attribute or embedded attributes and documents. Indexes can also be created on multiple attributes
- Concurrency
  - MongoDB refrains from using any kind of locking on data
  - Concurrent writes will simply overwrite each other's data as they go straight to memory
  - There is also no guarantee that when you query information that it is the most current because there are no locks so that someone maybe updating that information as you are viewing it
- Components of MongoDB
  - Core Server: core database server runs via an executable called mongod
  - JavaScript shell: the MongoDB command shell is a JavaScript based tool for administering the database and manipulating data
  - Database drivers: is the code used in an application to communicate with a MongoDB server.
  - Command line tools:
    - Mongodump and mongorestore – standard utilities for backing up and restoring a database
    - Mongoexport and mongoimport – export and import JSON, CVS and TSV data this is useful if you need your data in a widely supported format
    - Mongosniff – a wire-sniffing tool for viewing operations sent to the database.
    - Mongostat – polls MongoDB and the system to provide helpful stats
    - Mongotop – polls MongoDB and shows the amount of time it spends reading and writing data in each collection
    - Mongoperf – helps you understand the disk operations happening in a running MongoDB instance
    - Mongooplog – shows what is happening in the MongoDB oplog
    - Bsondump – Converts BSON files into human readable formats including JSON