# The City Lit Institute

##### Department of Computing

## Keeley Street, Holborn, London WC2B 4BA

**MySQL**

**(with Apache server)**

Introduction to MongoDB

**LECTURER** : **ALEXANDER ADU-SARKODIE**

MSc. Telematics (IT & Telecom), MSc. Eng., Dip. Russ. Lang., Teach. Cert, AMIAEng (UK), MBCS (UK), MIfL(UK)

**Blog** : <http://www.blogger.com/profile/14800490193632788559>

**Email:** [aadusarkodie1@gmail.com](mailto:aadusarkodie1@gmail.com)

**MongoDB**

MongoDB is a cross-platform document-oriented database. It is an Agile database. As it has no fixed schema. Schemas can be constructed dynamically.

A database schema of a database system is its structure described in a formal language supported by the Database Management System (DBMS). The term “schema” refers to the organisation of data as a blue print of how the database is constructed (divided into database tables in the case of relational databases).

Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. The **B** stands for Binary. First developed by the software company 10gen (now MongoDB Inc.) in October 2007.

As of 2014, MongoDB was the most popular NoSQL database system.

In MongoDB tables are collections and rows or records are referred to as documents

**Main features**

Some of the features include:

**Document-oriented**

Instead of taking a business subject and breaking it up into multiple relational structures, MongoDB can store the business subject in the minimal number of documents. For example, instead of storing title and author information in two distinct relational structures, title, author, and other title-related information can all be stored in a single document called Book, which is much more intuitive and usually easier to work with.[5]

Ad hoc queries

MongoDB supports search by field, range queries, regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions.

**Indexing**

Any field in a MongoDB document can be indexed (indices in MongoDB are conceptually similar to those in RDBMSes). Secondary indices are also available.

**Replication**

MongoDB provides high availability with replica sets.A replica set consists of two or more copies of the data. Each replica set member may act in the role of primary or secondary replica at any time. The primary replica performs all writes and reads by default. Secondary replicas maintain a copy of the data on the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary.

Secondaries can also perform read operations, but the data is eventually consistent by default.

**Load balancing**

MongoDB scales horizontally using sharding.

The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more slaves.)

MongoDB can run over multiple servers, balancing the load and/or duplicating data to keep the system up and running in case of hardware failure. Automatic configuration is easy to deploy, and new machines can be added to a running database.

**File storage**

MongoDB can be used as a file system, taking advantage of load balancing and data replication features over multiple machines for storing files.

This function, called GridFS, is included with MongoDB drivers and available with no difficulty for development languages. MongoDB exposes functions for file manipulation and content to developers.

Instead of storing a file in a single document, GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document.

In a multi-machine MongoDB system, files can be distributed and copied multiple times between machines transparently, thus effectively creating a load-balanced and fault-tolerant system.

**Aggregation**

MapReduce can be used for batch processing of data and aggregation operations. The aggregation framework enables users to obtain the kind of results for which the SQL GROUP BY clause is used.

**Server-side JavaScript execution**

JavaScript can be used in queries, aggregation functions (such as MapReduce), and sent directly to the database to be executed.

**Capped collections**

MongoDB supports fixed-size collections called capped collections. This type of collection maintains insertion order and, once the specified size has been reached, behaves like a circular queue.

**Collaboration**

https://www.youtube.com/watch?v=2ajlfURobd8