CSCI-UA.60-1 Database Design and Implementation

Database Environments for Class Discussion; An Introduction to NoSQL

CSCI-UA.60-1
Prof Deena Engel
Department of Computer Science
deena@cs.nyu.edu

Overview – Clarification of Terms

- Additional SQL environments we have not discussed:
  - Desktop Database Environments:
    - MS-Access and Open Office ("Base")
    - Filemaker Pro
  - Server-side Databases:
    - Oracle and PostGreSQL
- Data Formats we have not yet discussed:
  - XML
  - JSON
- NoSQL products (e.g. MongoDB)

FileMaker Pro

- See http://www.filemaker.com/
- Popular desktop database
- Offers flexible and dynamic web interface for data integration
- Popular among small firms
- Filemaker has a long history and has been a successful product in many environments as it is cross-platform (Mac OS and Windows); it was an early desktop database to integrate multimedia, images, etc.; available for iPhone / iPad applications and one of the first to integrate with clients' websites
- Current stories - http://www.filemaker.com/solutions/customers/

PostgreSQL

- PostgreSQL
  - open source database – see http://www.postgresql.org/
  - Featured users – http://www.postgresql.org/about/users

- PostgreSQL is considered the best open source example of an RDBMS and supports:
  - Natural language parsing
  - Multidimensional indexing
  - Geographic queries
  - Custom datatypes
  - … and more

Benefits of an RDBMS / SQL:

PostgreSQL is widely used:

- As long as the schema is designed in a normalized way, SQL provides excellent query flexibility.
- SQL supports modules, tuning and indexing to generally yield good performance results.
- Can handle multiple terabytes of data with small resource consumption.
- ACID compliant transactions mean that commits are *atomic, consistent, isolated* and *durable*.
- … and more!

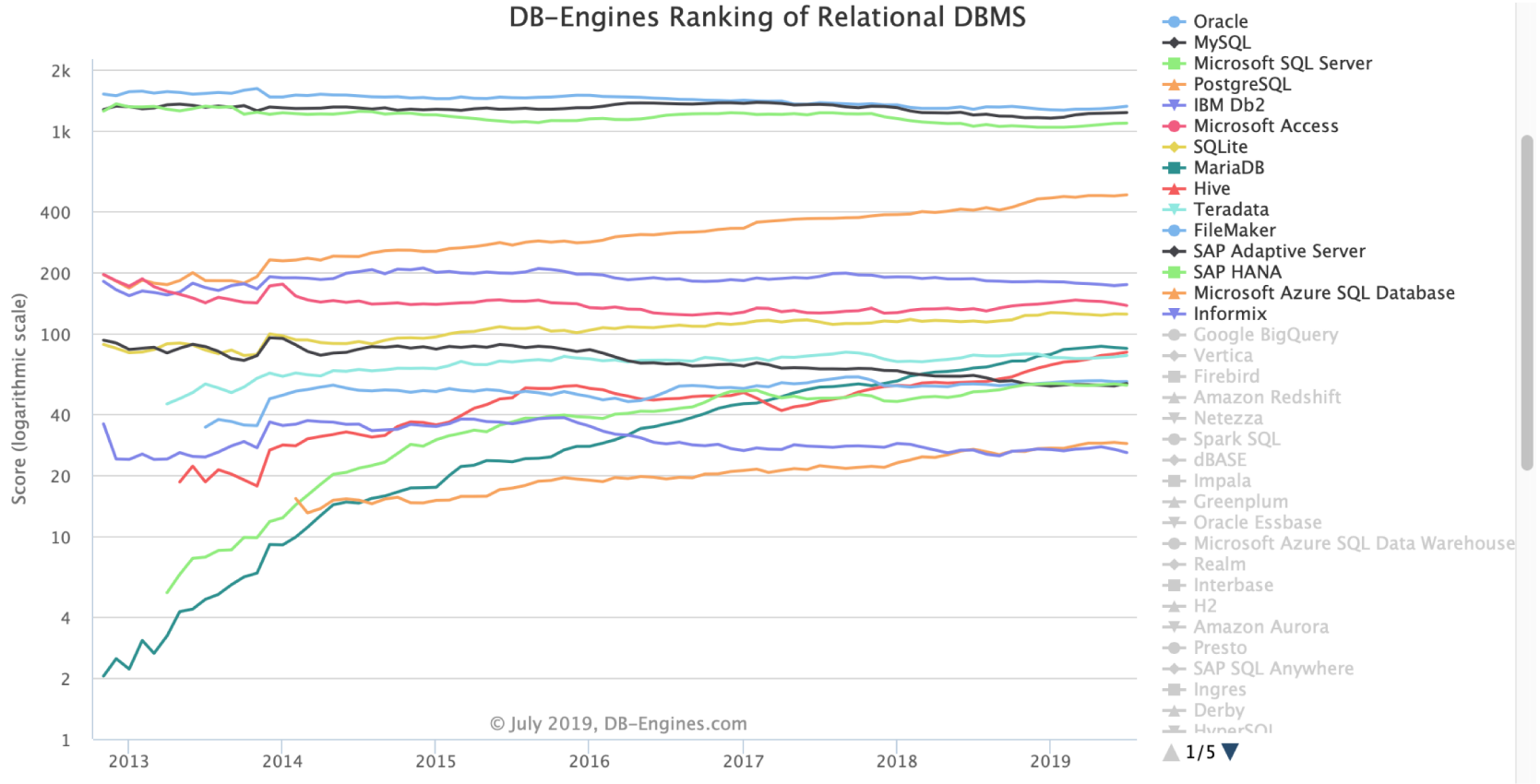
◦ From Seven Databases in Seven Weeks by Eric Redmond and Jim Wilson, page 50

Oracle

- Oracle
  - see http://www.oracle.com/index.html
  - Customers – see http://www.oracle.com/us/corporate/customers/index.html

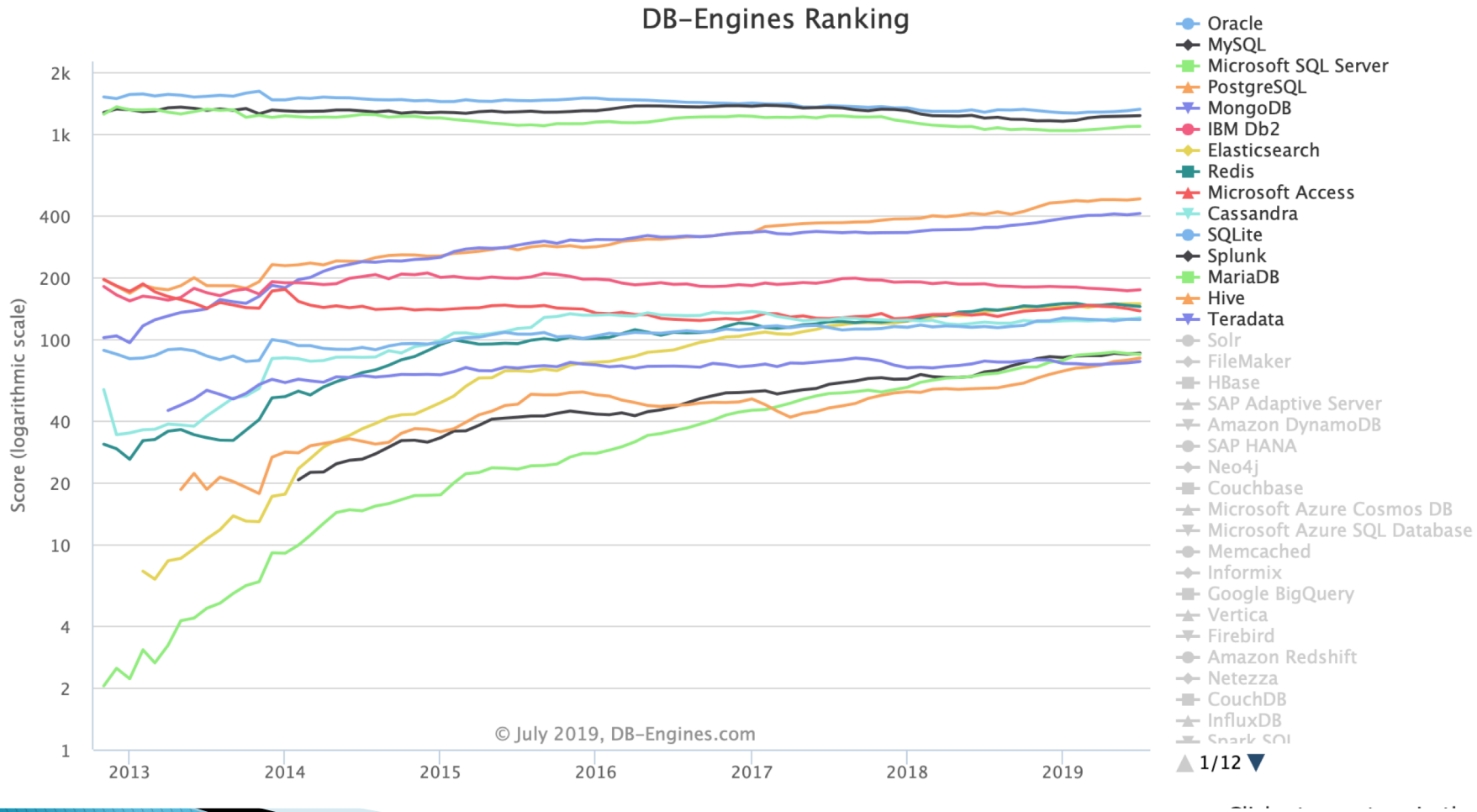
Overview of Popular RDBMS'

- https://db-engines.com/en/ranking_trend/relational+dbms



Overview of Popular DB Engines

- https://db-engines.com/en/ranking_trend



Additional Data Formats

XML and XSL / XSLT

- XML is NOT a database!!
- See XML: http://www.w3schools.com/xml/default.asp
- However, XML can be used with datasets that are displayed on the web.
- Earlier versions of XML were rendered using CSS but XSL / XSLT is a more powerful tool to render (transform) data sets for the web.
  - Use PHP, Perl or any other interface language to programmatically produce an XML-compliant file from data results that have been retrieved.
  - A better approach is to use the XML import/export capabilities that are typically available (Oracle, PostGreSQL) and MySQL has some capabilities as well (http://zetcode.com/databases/mysqltutorial/exportimport/)

JSON

- JSON: *JavaScript Object Notation* – a data interchange format (text-based)
- See http://www.json.org/
- Although JSON uses a JavaScript syntax, it remains language- and platform independent and is considered "self-describing".
- For example: the NYC Open Data site that we used earlier this semester to gather .CSV files, also offers options for JSON. (https://nycopendata.socrata.com/)
- JSON is also *NOT* a database!

JSON

- JSON: *JavaScript Object Notation*

```
{”Greeting": ”Hello, world}
```

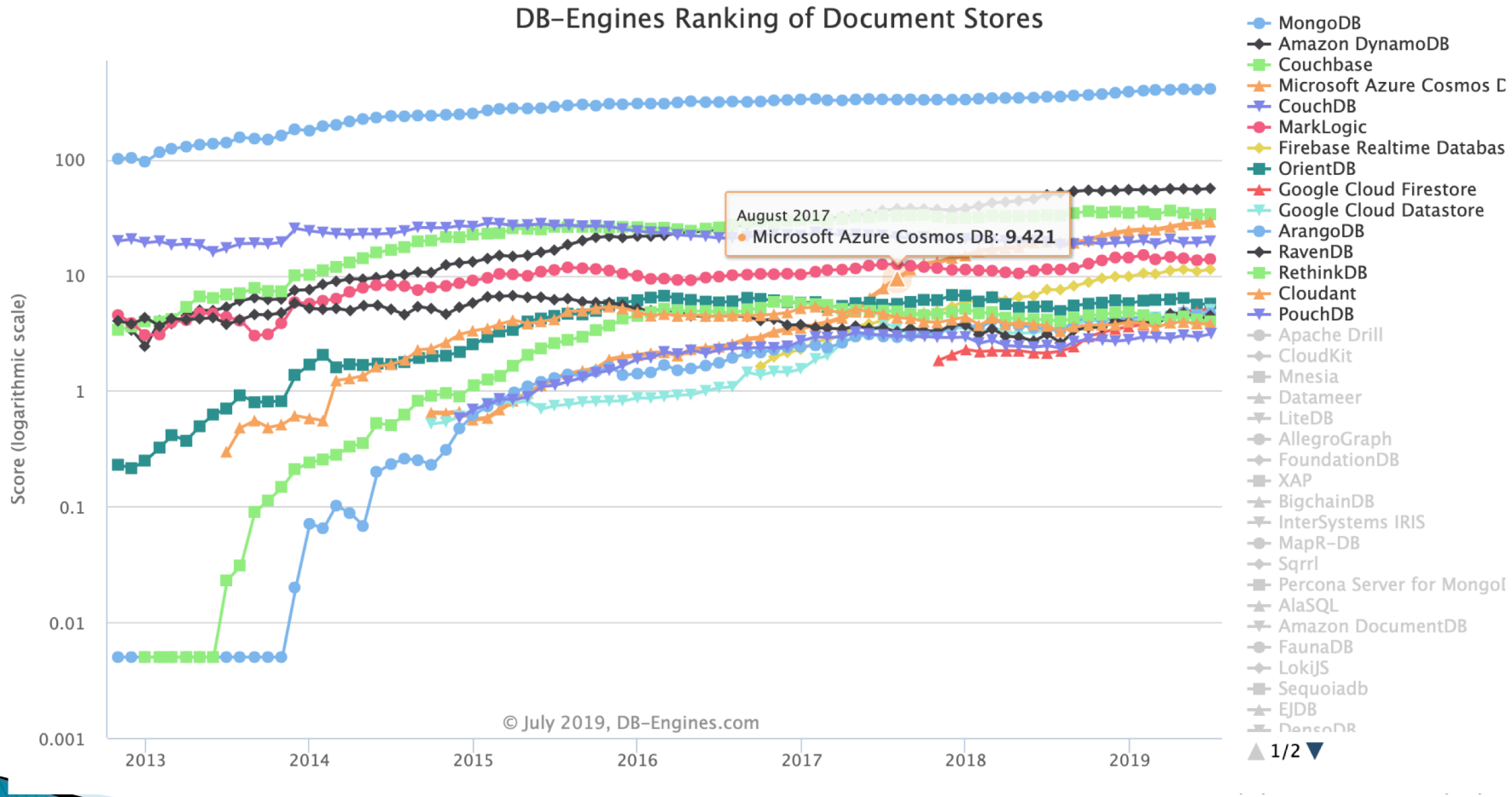
Introduction to NoSQL

NoSQL

- NoSQL: http://nosql-database.org/
- A list of NoSQL Examples: http://en.wikipedia.org/wiki/NoSQL#Document_store
- Some of the features include:
  - Document-Oriented Storage / JSON
  - Full Index Support
  - Mirror across LANs and WANs for scale and stability
  - Scales horizontally without compromising functionality
  - Querying: Rich, document-based queries.
  - Atomic ("in place") updates
  - … and other features

Popular NoSQL Document Stores

- https://db-engines.com/en/ranking_trend/document+store



MongoDB

- MongoDB: http://www.mongodb.org/
  - MongoDB as a document-oriented database (so a good option for certain types of websites but not necessarily for transactions-based systems)
- Some well-known MongoDB clients- http://www.10gen.com/customers
  - Foursquare
  - Disney Interactive Media Group
  - MTV

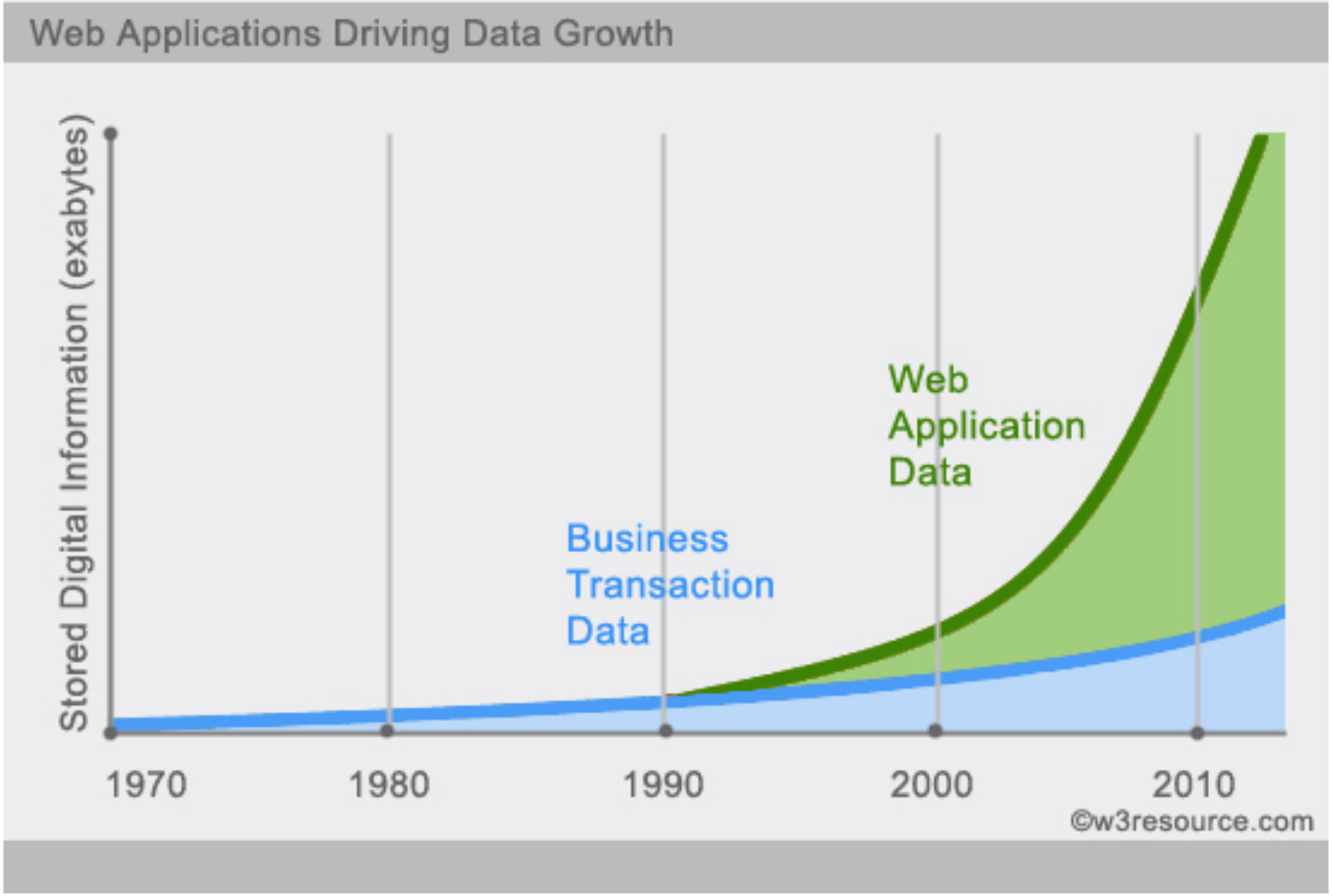
Notes about MongoDB

- First publicly released in 2009
- Designed as a scalable database
  - The name 'Mongo' supposedly comes from the word *'humongous'*
- Called a 'document database'
- Can query nested data
- Does not enforce a schema (different from SQL & RDBMS') so adding new fields to a document is easy as you will see!
- Supports powerful queryability
- Ease of use: MongoDB was designed with similarity in mind for SQL users to learn to write queries in MongoDB. You will see many similarities as we write queries and work with our datasets.

MongoDB

- Open source, written in C++
- Some of the features include:
  - Document-Oriented Storage / JSON
  - Full Index Support
  - Mirror across LANs and WANs for scale and stability
  - Scales horizontally without compromising functionality
  - Querying: Rich, document-based queries.
  - Atomic ("in place") updates and supports all CRUD operations
  - Geo-spatial queries
  - … and other features

Growth in digital information



http://www.w3resource.com/mongodb/nosql.php

Structured vs Unstructured Data

- Examples of unstructured and semi-structured data: log files, blogs, tweets, multi-media (audio, video)
  - No declarative query language; No predefined schema; Key-Value pair storage; eventual consistency is allowed rather than ACID transactions; can handle unstructured and unpredictable data.
  - horizontal scalability
- Examples of structured data: business transactions (financial transactions, medical records systems, university and academic records systems):
  - Structured and organized data; SQL; data and its relationships are stored in separate tables; consistency.
  - vertical scalability

Versions of MongoDB

- The current version running on *i6.cims.nyu.edu* supports aggregate queries in addition to geo-spatial data queries.

The “Mean” Stack

- The MEAN stack is
  - MongoDB – the database
  - Express.js, – web application framework
  - AngularJS – JavaScript framework, supports HTML
  - Node.js. – server-side environment
- All of the components in the MEAN stack support programs written in JavaScript.
  - MEAN applications can be written in one language for both server-side and client-side execution environments.

PyMongo

- For working with MongoDB from Python:
- https://api.mongodb.com/python/current/

NoSQL Terminology

DBMS	MongoDB
Table	Collection
Column / Field	Key
Value	Value
Row / Record	Document / Object

http://www.w3resource.com/mongodb/databases-documents-collections.php

MongoDB: Datatypes

Data type	description
string	empty string or characters
integer	digits
boolean	logical values (True / False)
double	floating point number
null	not zero, not empty
array	a list of values (similar to a list in python)
object	an entity which can be used in programming; could be a value, variable, function or data structure
Object ID	Every MongoDB object or document must have an Object ID which is unique. *This is a BSON(Binary JavaScript Object Notation, which is the binary interpretation of JSON) object id, a 12-byte binary value which has a very rare chance of getting duplicated. This id consists of a 4-byte timestamp (seconds since epoch), a 3-byte machine id, a 2-byte process id, and a 3-byte counter.*