Databases

Open Source Software

Wesley Turner

Rensselaer Polytechnic Institute

Department of Computer Science

Uses material from:

- https://github.com/rcos/CSCI2963-01-Spring2017/blob/master/Lectures/MongoDB.pdf
  - Thanks: Raymond Jacobson, Class of 2016 CS/CSE!
- https://www.tutorialspoint.com/mongodb

Reading Material

Reading Material

- **MongoDB Manual** – https://docs.mongodb.com/manual/
  - Read up through *MongoDB CRUD Operations*, especially **Introduction** and **SQL to MongoDB Mapping Chart**
  - You should be able to install MongoDB on your system, but you don't need to read those sections in detail.
- **Couch Manual** – https://docs.couchdb.org/en/stable/intro/index.html
  - Read the *Introduction*

Introduction

What is a Database?

But people have always used databases

Store Information

Through the Decades

1970s

The Relational Database

+--------------+------------------+ | Professors | +--------------+------------------+ | Name | Favorite Software | +================+====================+ | Goldschmidt | Notepad | +--------------+------------------+ | Moorthy | Observatory | +--------------+------------------+ | Magdon-Ismail | Latex | +--------------+------------------+ | Cutler | Dr. Memory | +--------------+------------------+

+-------------+---------------+-------------+ | Software | +-------------+---------------+-------------+ | Name | Platforms | Release Date | +==============+================+==============+ | Notepad | Windows | 1983 | +-------------+---------------+-------------+ | Dr. Memory | Cross-Platform | 2011 | +-------------+---------------+-------------+ | Observatory | Web | 2010 | +-------------+---------------+-------------+ | LaTeX | Cross-Platform | 1985 | +-------------+---------------+-------------+


```
> SELECT FAVORITE_SOFTWARE FROM PROFESSORS WHERE NAME='Moorthy'

>> "Observatory"
```

Joins

+-------------+------------------+---+------------+-------------+---------------+ | Professors | | Software | +-------------+------------------+---+------------+-------------+---------------+ | Name | Favorite Software | | Name | Platforms | Release Date | +==============+====================+===+============+=============+===============+ | Goldschmidt | Notepad | | Notepad | Windows | 1983 | +-------------+------------------+---+------------+-------------+---------------+ | Moorthy | Observatory | | Dr. Memory | **Cross-** | 2011 | | | | | **Platform** | | +-------------+------------------+---+------------+-------------+---------------+ | Magdon- | Latex | | Observatory | Web | 2010 | | Ismail | | | | | +-------------+------------------+---+------------+-------------+---------------+ | Cutler | Dr. Memory | | LaTeX | **Cross-** | 1985 | | | | | **Platform** | | +-------------+------------------+---+------------+-------------+---------------+

```
> SELECT PROFESSORS.NAME FROM PROFESSORS JOIN SOFTWARE ON \
PROFESSORS.FAVORITE_SOFTWARE=SOFTWARE.NAME WHERE \
SOFTWARE.PLATFORMS="Cross-platform"
>> "Magdon-Ismail", "Cutler"
```

Name	Favorite Software	Platforms	Release Date
Goldschmidt	Notepad	Windows	1983
Moorthy	Observatory	Web	2010
Magdon-Ismail	LaTeX	Cross-platform	1985
Cutler	Dr. Memory	Cross-platform	2011


```
> SELECT PROFESSORS.NAME FROM PROFESSORS JOIN SOFTWARE ON \
PROFESSORS.FAVORITE_SOFTWARE=SOFTWARE.NAME WHERE \
SOFTWARE.PLATFORMS="Cross-platform"
>> "Magdon-Ismail", "Cutler"
```


1970s/80s/90s

1970s/80s/90s

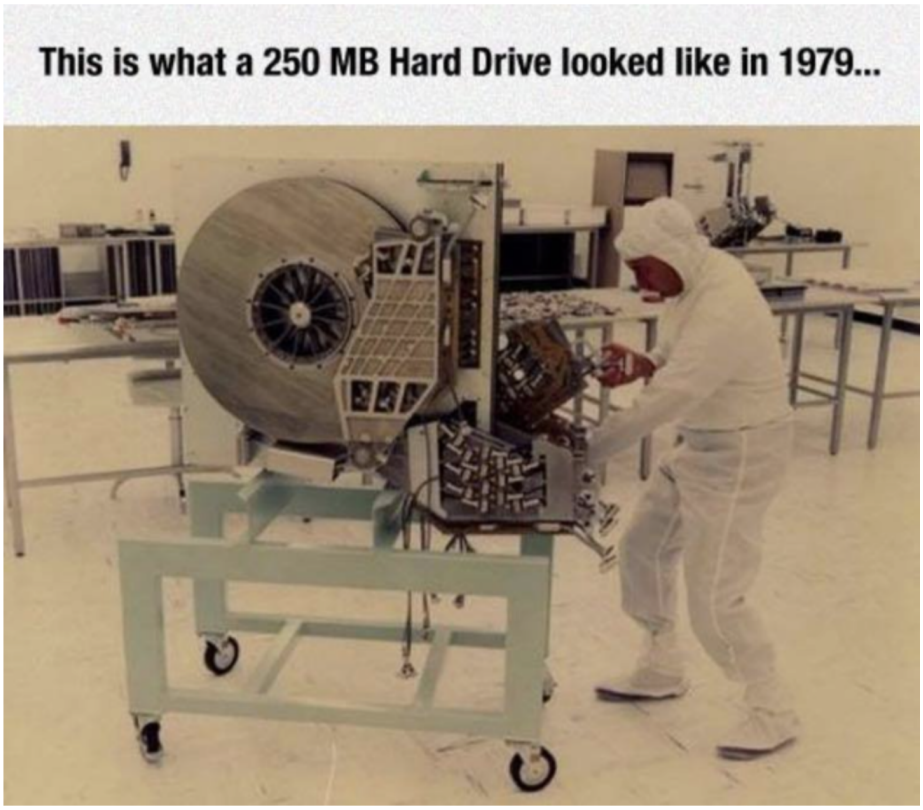
- Keep things separate (normalize)
- No redundant data
- Seems pretty sensible

- Keep things separate (normalize)
- No redundant data
- Seems pretty sensible

- Still widely used today in many applications

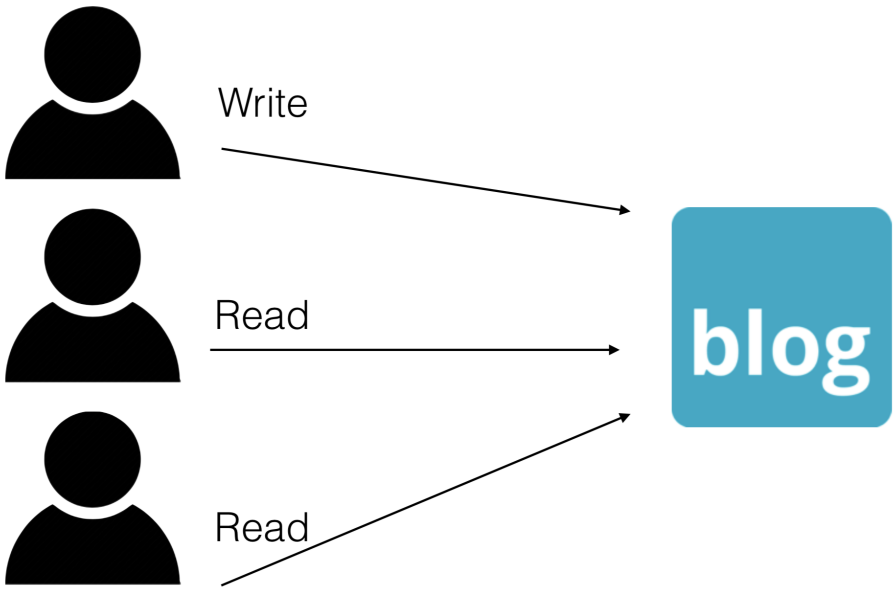
Name	Developers/Owner	Initial Version
OracleDB	Oracle	1978
DB2	IBM	1983
MySQL	Oracle	1995
Microsoft SQL Server	Microsoft	1989
PostgreSQL	PostgreSQL Global Development Group	1996
Sybase	SAP	1987

The World Then

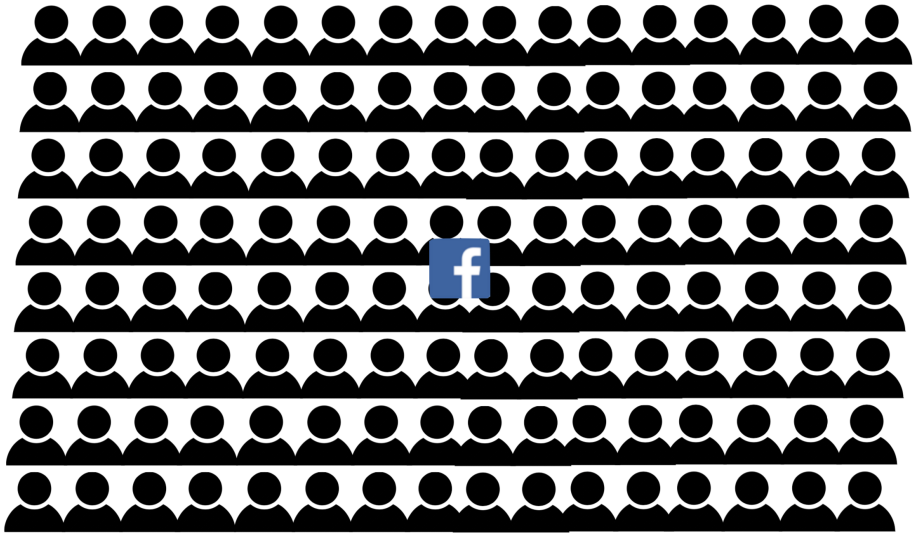


The World Now

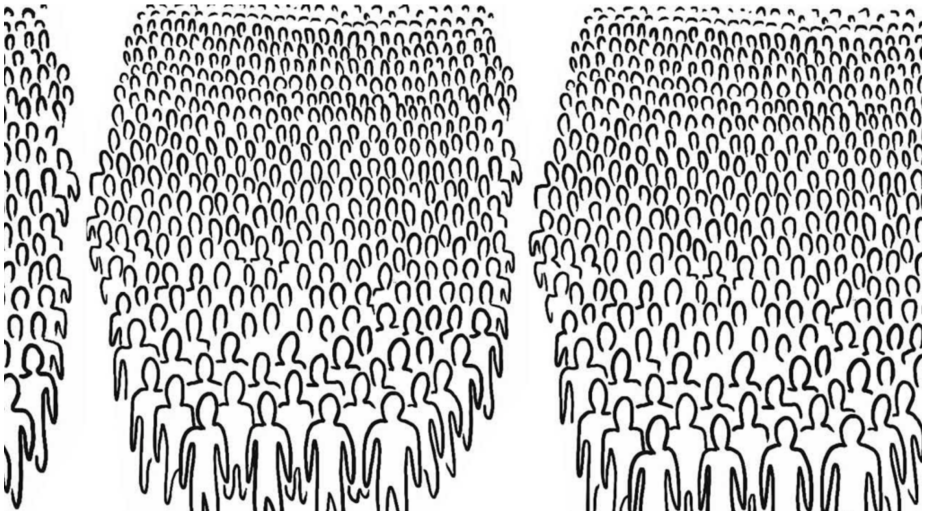
The World Then



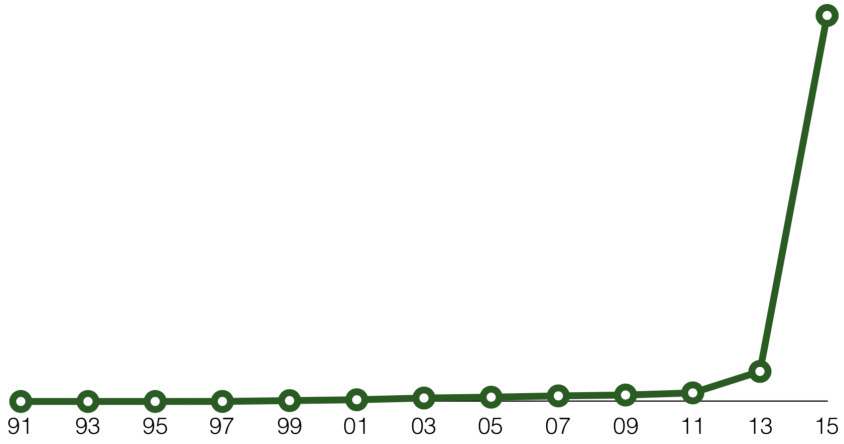
The World Now



But Really



As of 2015, a full 90 percent of all the data in the world has been generated over the previous two years



Vertical Scalability

Name	Favorite Software
Goldschmidt	Notepad
Moorthy	Observatory
Magdon-Ismail	LaTeX
Cutler	Dr. Memory

Name	Platforms	Release Date
Notepad	Windows	1983
Dr. Memory	Cross-platform	2011
Observatory	Web	2010
LaTeX	Cross-platform	1985

Horizontal Scalability

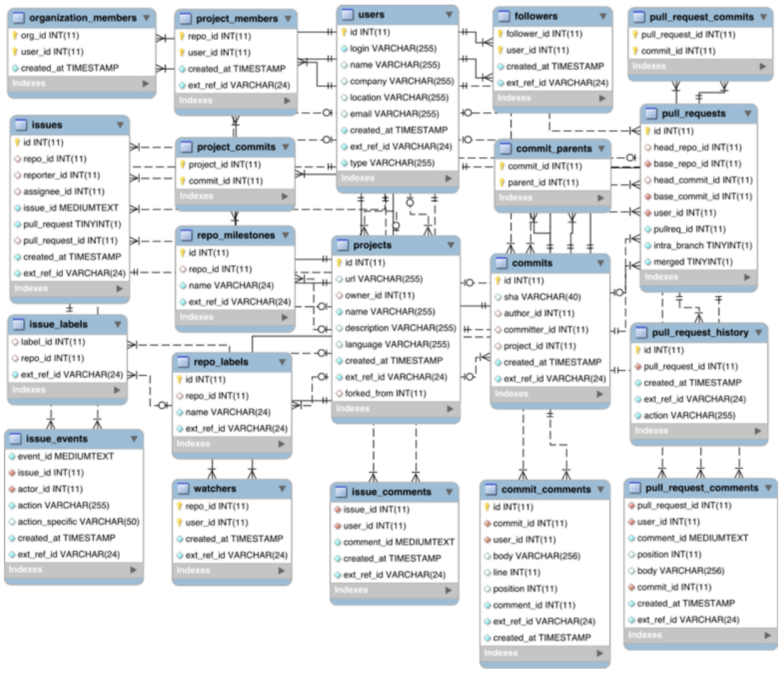
Which professors like software that runs on Windows?

Name	Favorite Software
Goldschmidt	Notepad
Moorthy	Observatory
Magdon-Ismail	LaTeX
Cutler	Dr. Memory

Expensive Join

Name	Platforms	Release Date
Notepad	Windows	1983
Dr. Memory	Cross-platform	2011
Observatory	Web	2010
LaTeX	Cross-platform	1985

Normalization




```
class AudioDevice:
    def __init__(self, manufacturer, device):
        self.manufacturer = manufacturer
        self.device = device

class MP3Player(AudioDevice):
    def __init__(self, manufacturer, device, file_format):
        AudioDevice.__init__(self, manufacturer, device)
        self.file_format = file_format

class RecordPlayer(AudioDevice):
    def __init__(self, manufacturer, device, speed):
        AudioDevice.__init__(self, manufacturer, device)
        self.speed = speed
```

Device	Manufacturer	File Format	Speed
iPod	Apple	.m4a	:-(
Turntable	Pro-Ject	:-(	33 rpm
Zune	Microsoft	.wma	:-(

Device	Manufacturer	File Format
iPod	Apple	.m4a
Zune	Microsoft	.wma

Device	Manufacturer	Speed
Turntable	Pro-Ject	33 rpm

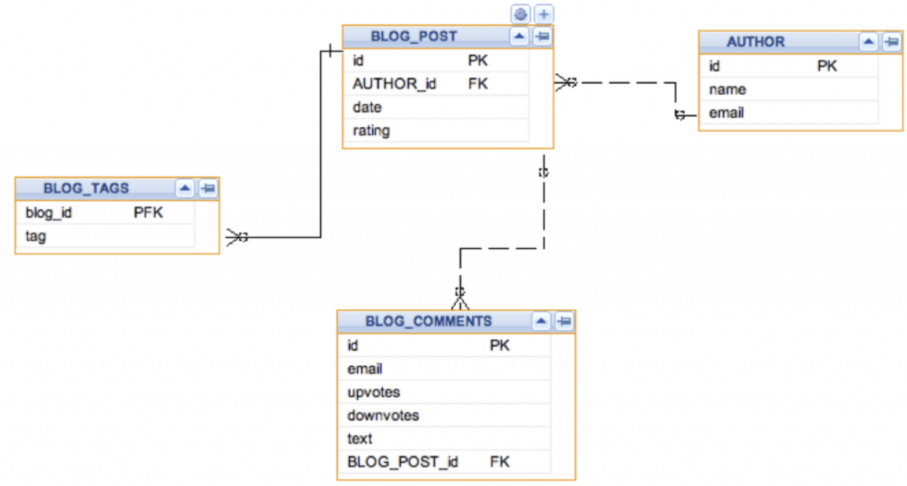
Device	File Format
iPod	.m4a
Zune	.wma

Device	Speed
Turntable	33 rpm

Device	Manufacturer
iPod	Apple
Turntable	Pro-Ject
Zune	Microsoft

So ... What is NoSQL?

Schema



Well ... MongoDB is a Document Database

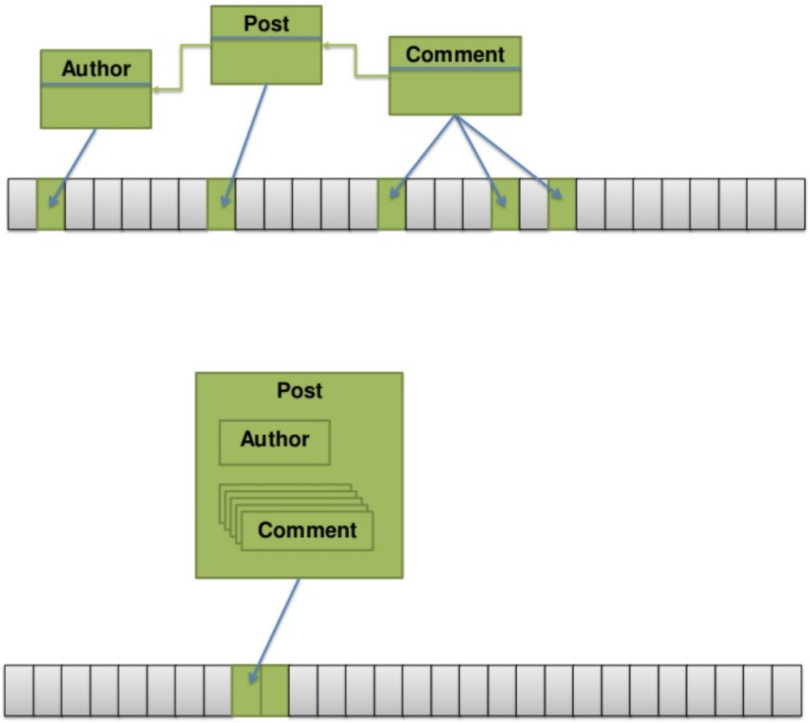
```
{ _id: 1234,
author: { name: "Bob Davis", email : "bob@bob.com" },
post: "In these troubled times I like to ...",
date: { $date: "2010-07-12 13:23UTC" },
location: [ -121.2322, 42.1223222 ],
rating: 2.2, comments: [
  { user: "jgs32@hotmail.com", upVotes: 22,
    downVotes: 14,
    text: "Great point! I agree" },
  { user: "holly.davidson@gmail.com", upVotes: 421,
    downVotes: 22,
    text: "You are a moron" } ],
tags: [ "Politics", "Virginia" ]
}
```

Actually, BSON (Binary JSON) http://bsonspec.org/

Other NoSQL Options May Have Different Terminology

But the concept is less formalization ... Think of storing a Python dictionary.

Normalization vs. Document Store



Comparison of Performance SQL vs NoSQL

https://static.epcc.ed.ac.uk/dissertations/hpc-msc/2012-2013/RDBMS%20vs%20NoSQL%20-%20Performance%20and%20Scaling%20Comparison.pdf

- Take aways:
  - NoSQL Databases handle complex queries faster at the expense of data duplication.
  - SQL Databases perform similarly on simple queries (no joins), but perform worse on complex queries; however, the enforce data consistency and have far less data duplication.

Flexible Schema

Relational	DB MongoDB
1. Set up schema	1. Insert data

Relational	DB MongoDB
2. Insert data	2. Insert data with new structure
3. Change schema	3. Insert data with new structure
4. Insert data with new structure	
5. How do I change the schema? Am I breaking something?	
6. Insert data with new structure	

Problems with Flexible

```
{
  _id: 1,
  author: { name: "Bob Davis", email : "bob@bob.com" },
  post: "In these troubled times I like to ...",
  date: { $date: "2010-07-12 13:23UTC" },
}
{
  _id: 1928571982758,
  author: { name: "Peter Brown", email : "brownp@rpi.edu" },
  post: "First blog post ever",
  date: { $date: "2014-11-12 13:23UTC" },
  tags: [ "Food", "DIY" ]
}
```

Why is MongoDB Open Source?

1. Community
2. Documentation
3. Ease of adoption
4. Trust in open source

Why is MongoDB Open Source?

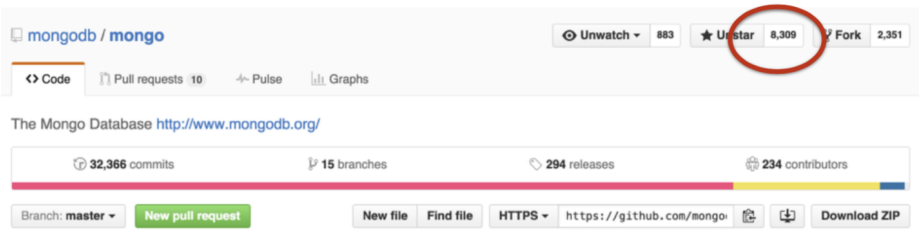
1. Community
2. Documentation

3. Ease of adoption
4. Trust in open source

What Happened?

Databases are an Unsolved Problem

- Scalability
- Fault tolerance/Availability Consistency
- Speed



Other Options

Database Top 10

Who's popular?

Database Top 10

- The following list is from:
  - https://www.databasejournal.com/features/oracle/slideshows/top-10-2019-databases.html

  - This no longer available
- A new list from 2021 is similar:
  - https://db-engines.com/en/ranking
- Although, Stackoverflow in 2020 sees it somewhat differently:
  - https://insights.stackoverflow.com/survey/2020#technology-databases-all-respondents4

Database Top 10

A The first commercially available SQL-based Relational Database Management System was released by Oracle in 1979. Oracle provides a range of industry-leading on-premises and cloud-based database solutions to meet the data management requirements from small businesses to large enterprises.

MySQL is the most popular Open Source SQL Database Management System (DBMS). MySQL databases are relational which stores data in separate organized tables. MySQL is Open Source which means that it is possible for anyone to use and modify the software. Anybody may download MySQL from the Internet and use it without paying a cent.

Security innovations in Microsoft's flagship database, Microsoft SQL Server, help secure data for mission-critical workloads with 'layers of protection', Always Encrypted technology, dynamic data masking, and transparent data encryption.

PostgreSQL is an object-relational database management system. PostgreSQL is transactional and ACID-compliant. PostgreSQL contains updatable views and materialized views, triggers, foreign keys and supports stored procedures and functions.

PostgreSQL is free and open source, so you are free to use, modify and distribute PostgreSQL in any form.

MongoDB is a cross-platform document-oriented database. It stores data in flexible, JSON-like documents. MongoDB's document model maps to the objects in your application thus making data easy to work with.

The Data warehouse includes a common SQL engine to support a wide range of

data structures and types. IBM Data Lake enables agile, data-driven decisions by utilizing vast amounts of unstructured data that historically could not be analyzed. IBM Fast Data combines fast data ingestion and concurrent analysis of real-time and historical data with machine learning.

Redis (Remote Dictionary Server) is an open source in-memory data structure store, functioning as a database. It has built-in replication, Lua scripting and transactions. Redis supports strings, hashes, lists, sets, bitmaps, hyper loglogs, geospatial indexes and streams.

Amazon SimpleDB is a NoSQL data store that offloads the work of database administration. Developers can easily store and query data items via web services requests while Amazon SimpleDB does the rest.

Amazon SimpleDB is not a relational database system, it instead creates and manages multiple geographically distributed replicas of your data automatically that enables high availability and data durability.

Microsoft Access is a lightweight database management system that combines the Microsoft Jet Database Engine with a user interface. An added benefit is that Microsoft Access is a member of the Microsoft Office suite of applications. Microsoft Access offers traditional Access desktop solutions as well as SharePoint web solutions.

SQLite is a C-language library that implements a small, very fast, self-contained SQL database engine. SQLite is the most used database engine in the world mainly due to it being built into all mobile phones and most computers.

SQLite is ACID-compliant. It implements most of the SQL standard making use of the PostgreSQL syntax. On the other hand, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee any domain integrity.

Quick Tutorial on MongoDB

Install MongoDB For Ubuntu 20.04 (Focal)

```
> sudo apt-get install gnupg
> wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | \
    sudo apt-key add -
> echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu \
    focal/mongodb-org/4.4 multiverse" | \
    sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
> sudo apt-get update
> sudo apt-get install mongodb-org
```

Other options and instructions can be found at https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/#install-mongodb-community-edition

Install MongoDB For WSL (Maybe)

```
> sudo apt-get update
> sudo apt-get install mongodb
```

This can be found at https://dev.to/seanwelshbrown/installing-mongodb-on-windows-subsystem-for-linux-wsl-2-19m9

These install different versions ... 4.4 vs 3.6.8?

Or you can use Docker (Untested)

YAML file:

```
version: "3.8"
services:
mongodb:
image : mongo
container_name: mongodb
environment:
- PUID=1000
- PGID=1000
volumes:
```

```
- /home/barry/mongodb/database:/data/db
ports:
- 27017:27017
restart: unless-stopped
```

This can be found at https://www.bmc.com/blogs/mongodb-docker-container/

Start the Database and Connect to It

```
> mkdir database
> sudo mongod --dbpath database
```

Then in a separate window,

```
> mongo
```

Get Help and Stats

```
> db.help()
DB methods:
  db.addUser(userDocument)
  db.adminCommand(nameOrDocument) - switches to 'admin' db, and runs command [ just calls db
  db.auth(username, password)
  db.cloneDatabase(fromhost)
  db.commandHelp(name) returns the help for the command
  db.copyDatabase(fromdb, todb, fromhost)
  db.createCollection(name, { size : ..., capped : ..., max : ... } )
  db.currentOp() displays currently executing operations in the db
  db.dropDatabase()
      ...
```

Get Help and Stats

```
> db.stats()
{
"db" : "test",
"collections" : 0,
"views" : 0,
"objects" : 0,
"avgObjSize" : 0,
"dataSize" : 0,
"storageSize" : 0,
"numExtents" : 0,
"indexes" : 0,
"indexSize" : 0,
"fileSize" : 0,
"fsUsedSize" : 0,
```

```
"fsTotalSize" : 0,
"ok" : 1
}
```

Create a New Database and Look At It

```
> use newdatabase
switched to db newdatabase
> db
newdatabase
> show dbs
local 0.078125GB
test  (empty)
> db.movie.insert({"name":"tutorials point"})
Mon Jul 23 03:12:49.382 [conn1] allocating new ns file database/newdatabase.ns, filling with
Mon Jul 23 03:12:49.621 [FileAllocator] allocating new datafile database/newdatabase.0, fill
...
> show dbs
local 0.078125GB
newdatabase   0.203125GB
test  (empty)
```

Collections

```
> show collections
movie
system.indexes
> db.movie.drop()
Mon Jul 23 03:24:20.458 [conn1] CMD: drop newdatabase.movie
true
> show collections
system.indexes
```

Add Some Data

```
>db.post.insert([
{
   title: 'MongoDB Overview',
   description: 'MongoDB is no sql database',
   by: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 100
},
```

Add Some Data

```
{
   title: 'NoSQL Database',
   description: "NoSQL database doesn't have tables",
   by: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 20,
   comments: [
      {
         user:'user1',
         message: 'My first comment',
         dateCreated: new Date(2013,11,10,2,35),
         like: 0
      }
   ]
}
])
```

What Happened?

```
> show collections
post
system.indexes
```

What Happened?

```
> db.post.stats()
{
  "ns" : "newdatabase.post",
  "count" : 2,
  "size" : 608,
  "avgObjSize" : 304,
  "storageSize" : 16384,
  "numExtents" : 1,
  "nindexes" : 1,
  "lastExtentSize" : 16384,
  "paddingFactor" : 1,
  "systemFlags" : 1,
  "userFlags" : 0,
  "totalIndexSize" : 8176,
  "indexSizes" : {
  "_id_" : 8176
},
  "ok" : 1
```

```
}
```

Find a Document

```
> db.post.find({"title": "MongoDB Overview"})
{ "_id" : ObjectId("5b554f0dc313b2ac9455e6cf"), "title" : "MongoDB Overview", "description"
> db.post.find({"title": "MongoDB Overview"}).pretty()
{
  "_id" : ObjectId("5b554f0dc313b2ac9455e6cf"),
  "title" : "MongoDB Overview",
  "description" : "MongoDB is no sql database",
  "by" : "tutorials point",
  "url" : "http://www.tutorialspoint.com",
  "tags" : [
      "mongodb",
      "database",
      "NoSQL"
  ],
  "likes" : 100
  }
```

Change a Document

```
> db.post.update({'title':'MongoDB Overview'}, \
    {$set:{'by':'New Author'}})
> db.post.find({'title':'MongoDB Overview'}).pretty()
{
  "_id" : ObjectId("5b554f0dc313b2ac9455e6cf"),
  "by" : "New Author",
  "description" : "MongoDB is no sql database",
  "likes" : 100,
  "tags" : [
    "mongodb",
    "database",
    "NoSQL"
  ],
  "title" : "MongoDB Overview",
  "url" : "http://www.tutorialspoint.com"
}
```

Quick Tutorial on CouchDB

Install CouchDB

- Follow the directions at:
  - https://docs.couchdb.org/en/stable/install/index.html

- We won't go into this here ... You will do it during Lab

Verify the Database using cURL

https://docs.couchdb.org/en/stable/intro/tour.html

```
> curl http://127.0.0.1:5984/

{
"couchdb": "Welcome",
"version": "3.0.0",
"git_sha": "83bdcf693",
"uuid": "56f16e7c93ff4a2dc20eb6acc7000b71",
"features": [
 "access-ready",
 "partitioned",
 "pluggable-storage-engines",
 "reshard",
 "scheduler"
],
"vendor": {
 "name": "The Apache Software Foundation"
  }
}
```

Create a new database

```
> curl -X GET http://admin:admin@127.0.0.1:5984/_all_dbs

["_replicator","_users"]

> curl -X PUT http://admin:admin@127.0.0.1:5984/baseball

{"ok":true}

> curl -X GET http://admin:admin@127.0.0.1:5984/_all_dbs

["baseball"]
```

(Required databases omitted from here on.)

```
> curl -X PUT http://admin:admin@127.0.0.1:5984/baseball

{"error":"file_exists","reason":"The database could not be created,
the file already exists."}

> curl -X PUT http://admin:admin@127.0.0.1:5984/plankton
```

```
{"ok":true}

> curl -X GET http://admin:admin@127.0.0.1:5984/_all_dbs

["baseball", "plankton"]

> curl -X DELETE http://admin:admin@127.0.0.1:5984/plankton

{"ok":true}

> curl -X GET http://admin:admin@127.0.0.1:5984/_all_dbs

["baseball"]
```

Add a Document

```
>  curl http://127.0.0.1:5984/_uuids

{"uuids":["9b1b50e6f7c792b0ca9371a79600edfa"]}

> curl -X PUT http://admin:admin@127.0.0.1:5984/baseball/9b1b50e6f7c792b0ca9371a79600edfa \
 -d '{"team":"Yankees"}'

{"ok":true,"id":"9b1b50e6f7c792b0ca9371a79600edfa", \
"rev":"1-6c7d6a1453648632433e1f18a59bc3bb"}

> curl -X GET http://admin:admin@127.0.0.1:5984/baseball/9b1b50e6f7c792b0ca9371a79600edfa

{"_id":"9b1b50e6f7c792b0ca9371a79600edfa",\
"_rev":"2-de43b90a11f24bfd4aea06f07bf98848","team":"Yankees"}

> curl -X PUT http://admin:admin@127.0.0.1:5984/baseball/9b1b50e6f7c792b0ca9371a79600edfa \
 -d '{"_rev":"1-6c7d6a1453648632433e1f18a59bc3bb","team":"Mighty Yankees"}'

{"ok":true,"id":"9b1b50e6f7c792b0ca9371a79600edfadoc",\
"rev":"2-de43b90a11f24bfd4aea06f07bf98848"}

> curl -X GET http://admin:admin@127.0.0.1:5984/baseball/9b1b50e6f7c792b0ca9371a79600edfa

{"_id":"9b1b50e6f7c792b0ca9371a79600edfa",\
"_rev":"2-de43b90a11f24bfd4aea06f07bf98848","team":"Mighty Yankees"}
```

Running from the Browser

- Our next few steps come from the CouchDB documents (well the previous ones did too!)
  - https://docs.couchdb.org/en/stable/intro/tour.html#welcome-to-fauxton

The End

by W. D. Turner