

NoSQL Databases

Name: _____ () Class: _____ Date: _____

Lesson 1: Introduction to NoSQL

Instructional Objectives:

By the end of this task, you should be able to:

- Understand how NoSQL database management system address the shortcomings of relational database management system (SQL)
- Explain the applications of SQL and NoSQL
- Use MongoDB shell to create, edit and remove databases

Why learn NoSQL?

Relational databases (commonly referred as SQL databases) work well with structured data since each table's **schema** (the precise description of the data to be stored and the relationships between them) is always clearly defined. However, with the increasing number of ways to gather and generate data, we often need to deal with unstructured data. For example, a convenience store that frequently refreshes the services it provides may sell both mobile phones as well as groceries. To run the store, information about both mobile phones (e.g., model names and prices) and groceries (e.g., prices and descriptions) need to be stored. In the future, the store may also start selling mobile phone subscription plans as well. Storing all this data in the same relational database may not be easy. In this case, non-relational databases, also referred to as **NoSQL databases**, can offer a better choice.

There are four main types of NoSQL databases: key-value databases, document databases, wide-column databases and graph databases. For the syllabus, we focus on MongoDB, a type of document database. Document databases work like a hash table, but each key can point to an embedded key-value structure, also known as a **document**, instead of just a single value. (Recall that in a hash table, each key points to a single value or data item.)

How NoSQL addresses shortcomings of relational databases

- Relational databases have a predefined schema that is difficult to change. Even if you wish to add a field to a small number of records, you still need to include the field for the entire table. Therefore, it can be difficult to support the processing of unstructured data using relational databases compared to NoSQL databases.
- Unlike NoSQL databases, relational databases do not usually support **hierarchical data storage** where less frequently-used data is moved to

NoSQL Databases

cheaper, slower storage devices. This means that the cost of storing data in a relational database is more expensive than storing the same amount of data in a NoSQL database.

- Relational databases are mainly vertically scalable while NoSQL databases are mainly horizontally scalable. **Vertically scalable** means that improving the performance of a relational database server usually requires upgrading an existing server with faster processors and more memory. Such high-performance components can be expensive and upgrades are limited by the capacity of a single machine. On the other hand, **horizontally scalable** means that the performance of a NoSQL database can be improved by simply increasing the number of servers. This is relatively cheaper as mass-produced average-performance computers are easily available at low prices.
- Relational databases are stored in a server, which makes the database unavailable when the server fails. NoSQL databases are designed to take advantage of multiple servers so that if one server fails, the other servers can continue to support applications.

Differences between relational databases and NoSQL databases

- Relational databases have a fixed, predefined schema that its tables follows but NoSQL databases usually have no predefined schema, which is dynamic and can change easily.
- Relational databases contain tables while NoSQL databases like MongoDB contain collections. The data types of each field in the table is fixed for relational databases but it is flexible for NoSQL databases like MongoDB.
- Relational databases represent data in tables and rows while NoSQL databases usually store data as collections of documents.
- For relational databases, joins are usually used to get data across tables, while for NoSQL databases like MongoDB there is usually no such joins. Thus it is easier to use relational databases for complex queries rather than NoSQL databases.

Applications of SQL and NoSQL Databases

The choice of whether to use a SQL or NoSQL database depends on the type of data being stored as well as the nature of tasks that the database is required to perform.

SQL databases should be used if:

- The data being stored has a fixed schema.
- Complex and varied queries will be frequently performed.
- The atomicity, consistency, isolation and durability (ACID) properties are critical to the database.
- There will be a high number of simultaneous transactions.

NoSQL Databases

NoSQL databases should be used if:

- The data being stored has a dynamic schema, (i.e., unstructured data with flexible data types).
- Data storage needs to be performed quickly.
- There will be an extremely large amount of data (i.e., Big Data).

Terms used in MongoDB

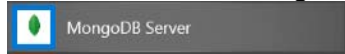
Here are the terms in MongoDB with the corresponding terms in SQL.

MongoDB Term	SQL Term
Database	Database
Collection	Table
Document	Row
Field	Column

Introduction to MongoDB

MongoDB is a NoSQL database. To start the MongoDB server, click on MongoDB

Server icon in the Start menu.



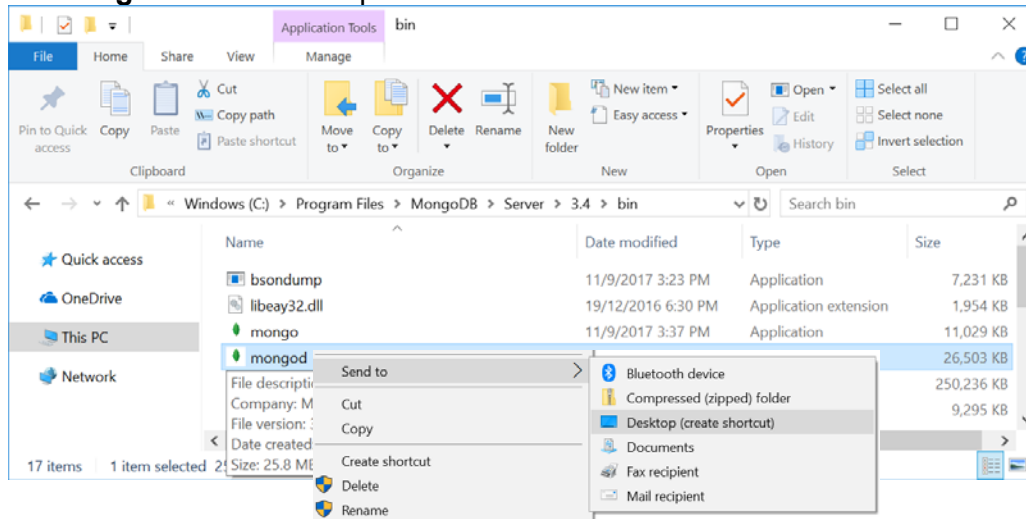
You will see a screen similar to the following.

```
MongoDB Server
2018-10-31T00:57:33.214-0700 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2018-10-31T00:57:33.216-0700 I CONTROL [initandlisten] db version v3.4.9
2018-10-31T00:57:33.216-0700 I CONTROL [initandlisten] git version: 876ebee8c7dd0e2d992f36a848ff4dc50ee6603e
2018-10-31T00:57:33.216-0700 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips 22 Sep 2016
2018-10-31T00:57:33.216-0700 I CONTROL [initandlisten] allocator: tcmalloc
2018-10-31T00:57:33.216-0700 I CONTROL [initandlisten] modules: none
2018-10-31T00:57:33.217-0700 I CONTROL [initandlisten] build environment:
2018-10-31T00:57:33.217-0700 I CONTROL [initandlisten]   distmod: 2008plus-ssl
2018-10-31T00:57:33.217-0700 I CONTROL [initandlisten]   distarch: x86_64
2018-10-31T00:57:33.217-0700 I CONTROL [initandlisten]   target_arch: x86_64
2018-10-31T00:57:33.217-0700 I CONTROL [initandlisten] options: { storage: { dbPath: "C:\Users\zhihao\AppData\Local\Temp\p" } }
2018-10-31T00:57:33.219-0700 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=7605M,session_max=2000,
eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten]
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten] **      Read and write access to data and configuration is unrestricted.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten]
2018-10-31T15:57:33.586+0800 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'C:\Users\zhihao\AppData\Local\Temp\diagnostic.data'
2018-10-31T15:57:33.620+0800 I INDEX [initandlisten] build index on: admin.system.version properties: { v: 2, key: {
version: 1 }, name: "incompatible_with_version_32", ns: "admin.system.version" }
2018-10-31T15:57:33.620+0800 I INDEX [initandlisten] building index using bulk method; build may temporarily use up to 500 megabytes of RAM
2018-10-31T15:57:33.627+0800 I INDEX [initandlisten] build index done. scanned 0 total records. 0 secs
2018-10-31T15:57:33.628+0800 I COMMAND [initandlisten] setting featureCompatibilityVersion to 3.4
2018-10-31T15:57:34.098+0800 I NETWORK [thread1] waiting for connections on port 27017
```

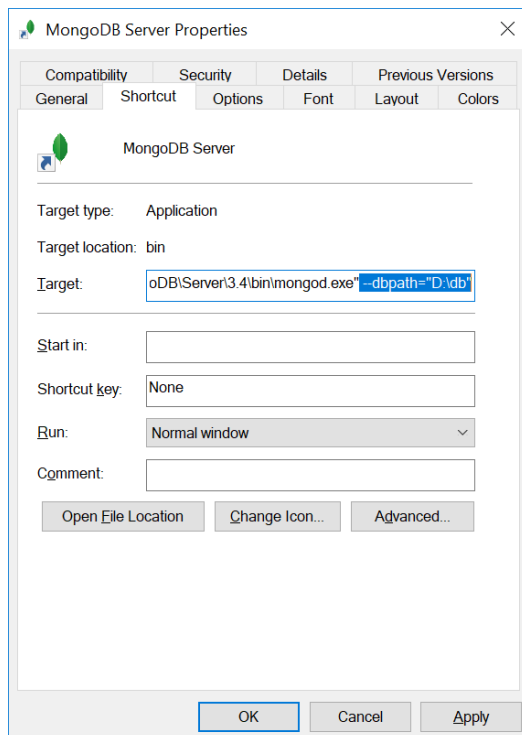
NoSQL Databases

Note!

If you don't see the icon, go to the MongoDB installation directory and create a shortcut for **mongod** on the desktop.



Right-click on properties and add in `--dbpath="D:\db"`, where `D:\db` is the directory storing the MongoDB database.

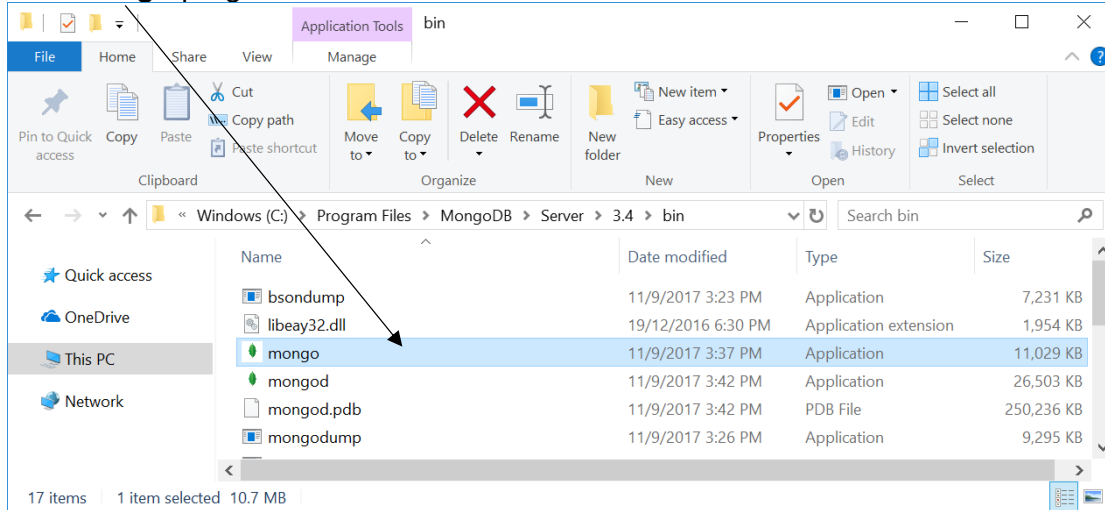


NoSQL Databases

Use of MongoDB shell

To interact with MongoDB databases, you need to start a connection to the server. To do so, use Windows Explorer and go to the MongoDB directory.

Start **mongo** program.



Starting the mongo shell program allows you to connect to the MongoDB server. You will then enter an interactive shell designed specifically for MongoDB.

```
Command Prompt - mongo
MongoDB shell version v3.4.9
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.9
Server has startup warnings:
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten]
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten] ** WARNING: Access
control is not enabled for the database.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten] **
Read a
nd write access to data and configuration is unrestricted.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten]
>
```

You can do use variables and mathematical operations similar to Python shell.

```
Command Prompt - mongo
nd write access to data and configuration is unrestricted.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten]
> 6+7
13
> y=4
4
> 3*4-2
10
> y/3
1.3333333333333333
>
```

NoSQL Databases

Go further! Look at the output on the MongoDB Server window as you type statements into the mongo shell, and try to figure out what the server is doing.

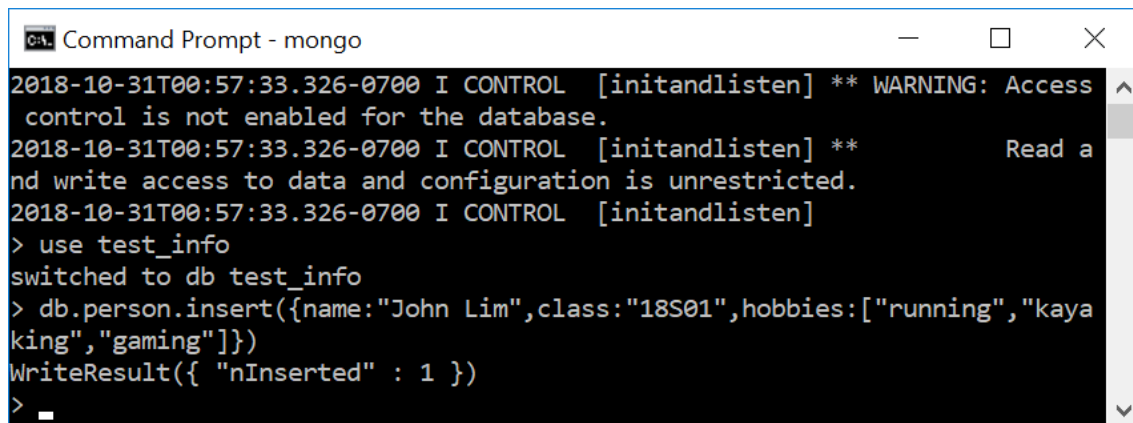
You can create a database with the **use** statement. You can insert documents into a collection using **insert** statement.

Try typing the following statements:

```
use test_info
db.person.insert({name:"John Lim",class:"18S01",hobbies:["running","kayaking","gaming"]})
```

The first statement tells MongoDB to use the database test_info. The second statement will insert a document into the person collection. Your screen will look like below. This states that one document has been inserted.

Note! The database test_info is created only when a document is inserted.



```
Command Prompt - mongo
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten] ** Read a
nd write access to data and configuration is unrestricted.
2018-10-31T00:57:33.326-0700 I CONTROL [initandlisten]
> use test_info
switched to db test_info
> db.person.insert({name:"John Lim",class:"18S01",hobbies:["running","kayaking","gaming"]})
WriteResult({ "nInserted" : 1 })
> _
```

Note! The collection is automatically created when a document is added. You can manually create the person collection using the statement `db.createCollection("person")`.

1. What do you think is the output when you type the following?
`db.person.insert({name:"Ben",class:"18S01"})`

.....
Check your answer by typing into MongoDB.

2. How many documents are there in the person collection now?

.....

3. Write down the statement to insert the student Mary from class 18S02 with hobby running into the person collection.

.....

NoSQL Databases

To find a particular document, you can use the `find` function.

For example, to find Ben's document, you can type
`db.person.find({name:"Ben"})`

Another example is to find the people with running as a hobby. You can type
`db.person.find({hobbies:"running"})`

To find all documents, you can type
`db.person.find( {} )`

4. What is the statement to find all information on students in class "18S01"?

.....

To list all the names of students in class "18S01", the following can be used.
`db.person.find({name:1}, class: "18S01")`

5. What is the statement to get all the names of students in class "18S02"?

.....

To get all the names and classes of the students, you can use the following.
`db.person.find({}, {name:1, class:1})`

Try out the various NoSQL statements.

To remove the person collection, use the following:
`db.person.drop()`

To remove the `test_info` database, use the following:
`db.dropDatabase()`

To exit Mongo shell, type `quit()`.

To shutdown the MongoDB server, press Ctrl-C.

References

Content	Link
NoSQL Databases	https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/
	https://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases
	https://www.mongodb.com/scale/types-of-nosql-databases
	Dan Sullivan (2015). <i>NoSQL for Mere Mortals</i> . Person Education

NoSQL Databases

MongoDB/ PyMongo	https://www.mongodb.com/what-is-mongodb http://api.mongodb.com/python/current/tutorial.html
---------------------	--