Introduction to NoSQL Database

C9a

Relational Database Concepts (recap)

- A relational database is a collection of data records with pre-defined relationships between them.
- These records are organized as a set of tables with columns and rows.
- Tables are used to hold information about the entities to be represented in the database (e.g. customer, item).
- Each column in a table holds data of an attribute/field of that entity.
- The rows in the table represent a collection of related data values, which form records/tuples for that entity.
- Each record has unique identifier called a primary key, and rows among multiple tables can be made related using foreign keys.

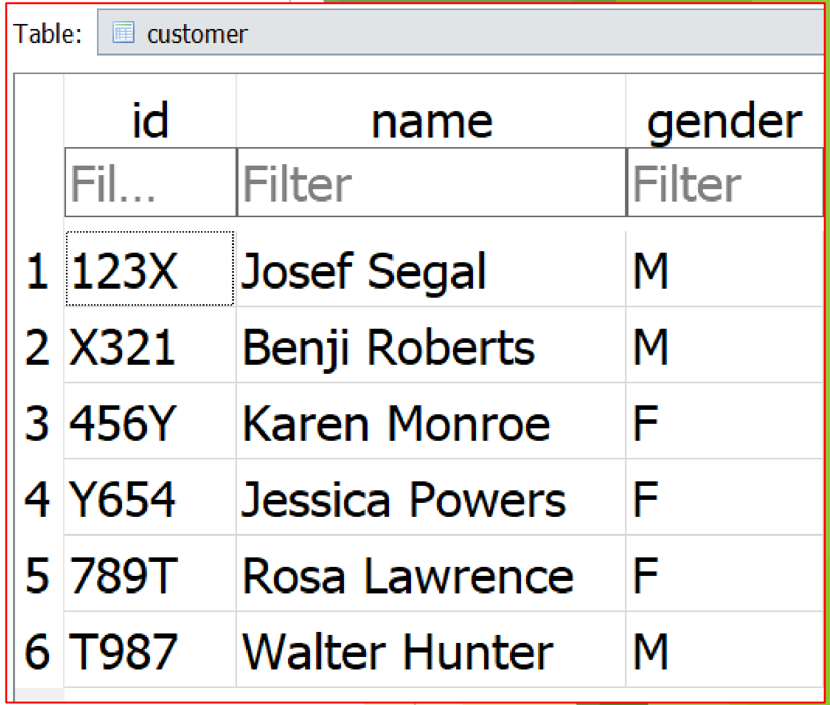
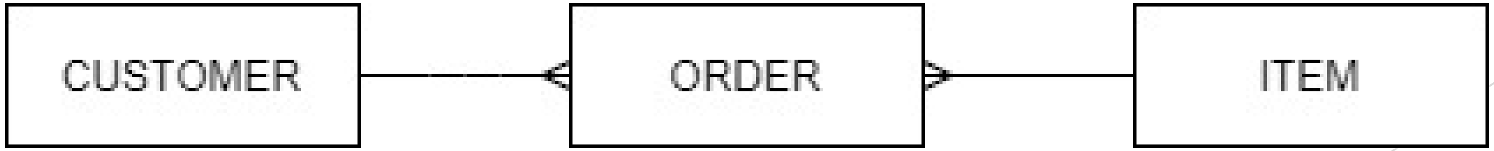


Table: customer

	id	name	gender
	Fil...	Filter	Filter
1	123X	Josef Segal	M
2	X321	Benji Roberts	M
3	456Y	Karen Monroe	F
4	Y654	Jessica Powers	F
5	789T	Rosa Lawrence	F
6	T987	Walter Hunter	M



https://aws.amazon.com/relational-database/

SQL concepts (recap)

- Structured Query Language (SQL) is used to add, update or delete rows of data, retrieving subsets of data for transaction processing and analytics applications, and to manage all aspects of the relational database.
- Relational databases use a set of constraints to enforce data integrity (overall completeness, accuracy and consistency of data)in the database.
- These include primary Keys, Foreign Keys, 'Not NULL' constraint, 'Unique' constraint, and 'Check' constraints.

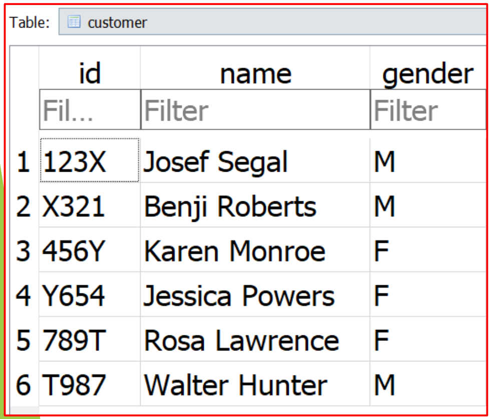


Table: customer

	id	name	gender
	Fil...	Filter	Filter
1	123X	Josef Segal	M
2	X321	Benji Roberts	M
3	456Y	Karen Monroe	F
4	Y654	Jessica Powers	F
5	789T	Rosa Lawrence	F
6	T987	Walter Hunter	M

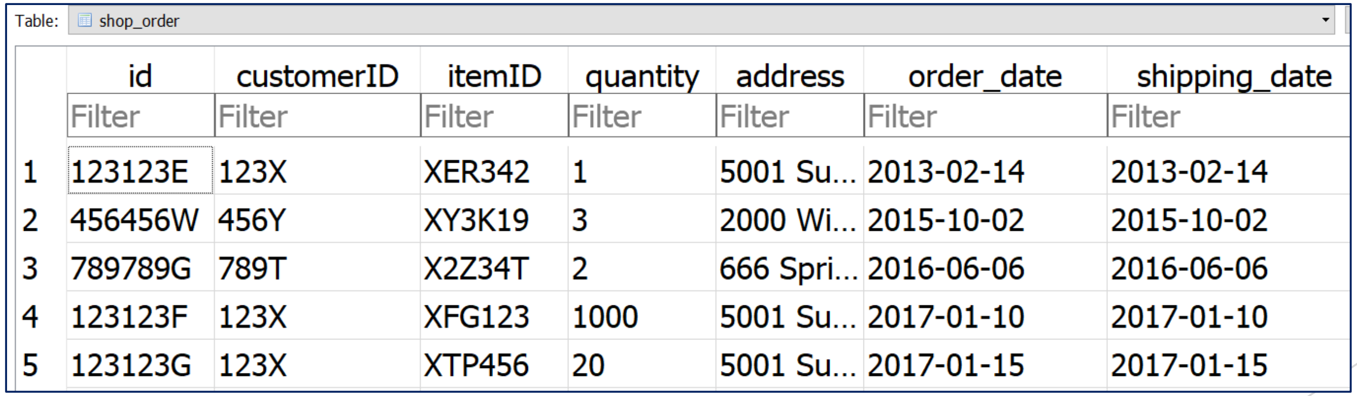


Table: shop_order

	id	customerID	itemID	quantity	address	order_date	shipping_date
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	123123E	123X	XER342	1	5001 Su...	2013-02-14	2013-02-14
2	456456W	456Y	XY3K19	3	2000 Wi...	2015-10-02	2015-10-02
3	789789G	789T	X2Z34T	2	666 Spri...	2016-06-06	2016-06-06
4	123123F	123X	XFG123	1000	5001 Su...	2017-01-10	2017-01-10
5	123123G	123X	XTP456	20	5001 Su...	2017-01-15	2017-01-15

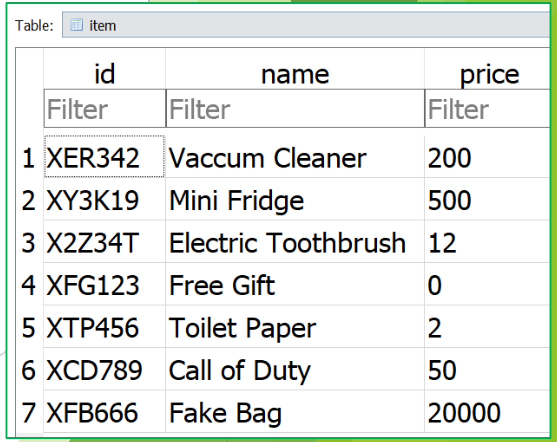
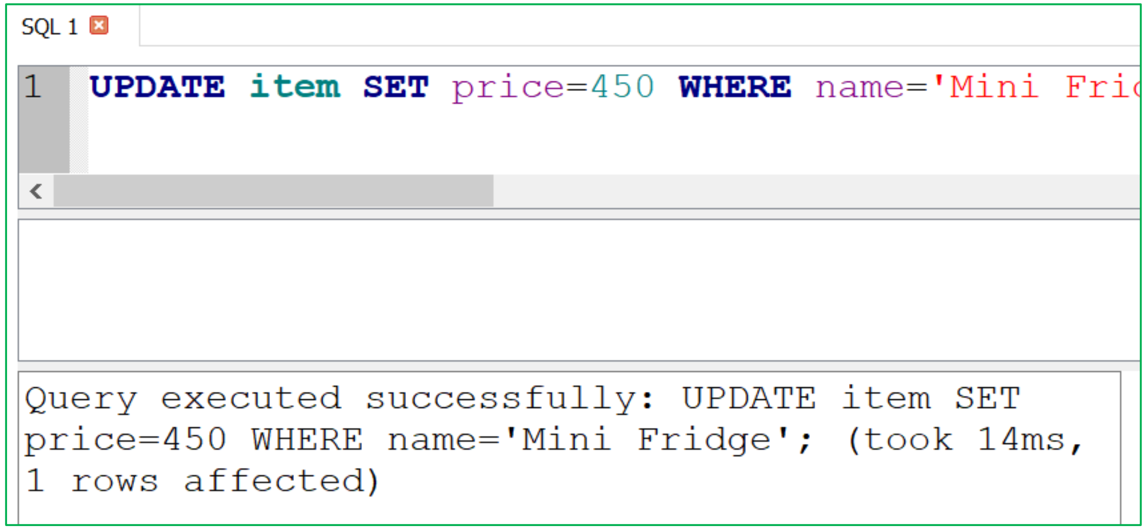


Table: item

	id	name	price
	Filter	Filter	Filter
1	XER342	Vaccum Cleaner	200
2	XY3K19	Mini Fridge	500
3	X2Z34T	Electric Toothbrush	12
4	XFG123	Free Gift	0
5	XTP456	Toilet Paper	2
6	XCD789	Call of Duty	50
7	XFB666	Fake Bag	20000

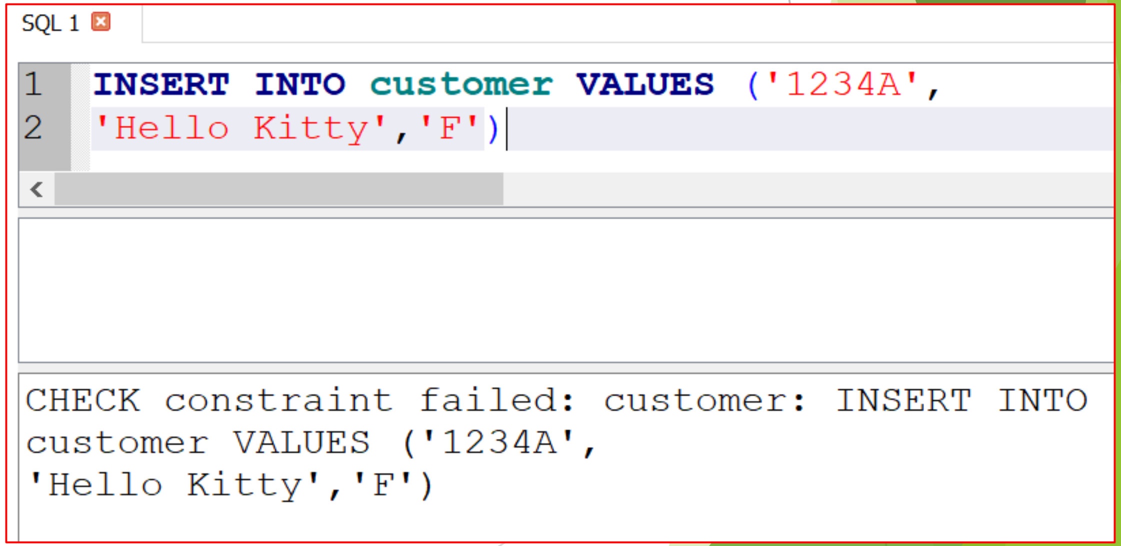
Transaction in Relational Database

- A transaction is one or more SQL statements that are executed as a sequence of operations that form a single logical unit of work.
- Transactions provide an "all-or-nothing" proposition, meaning that the entire transaction must complete as a single unit and be written to the database or none of the individual components of the transaction should go through.
- In the relation database terminology, a transaction results in a COMMIT or a ROLLBACK.



```
UPDATE item SET price=450 WHERE name='Mini Fric
```

```
Query executed successfully: UPDATE item SET
price=450 WHERE name='Mini Fridge'; (took 14ms,
1 rows affected)
```



```
INSERT INTO customer VALUES ('1234A',
'Hello Kitty','F')
```

```
CHECK constraint failed: customer: INSERT INTO
customer VALUES ('1234A',
'Hello Kitty','F')
```

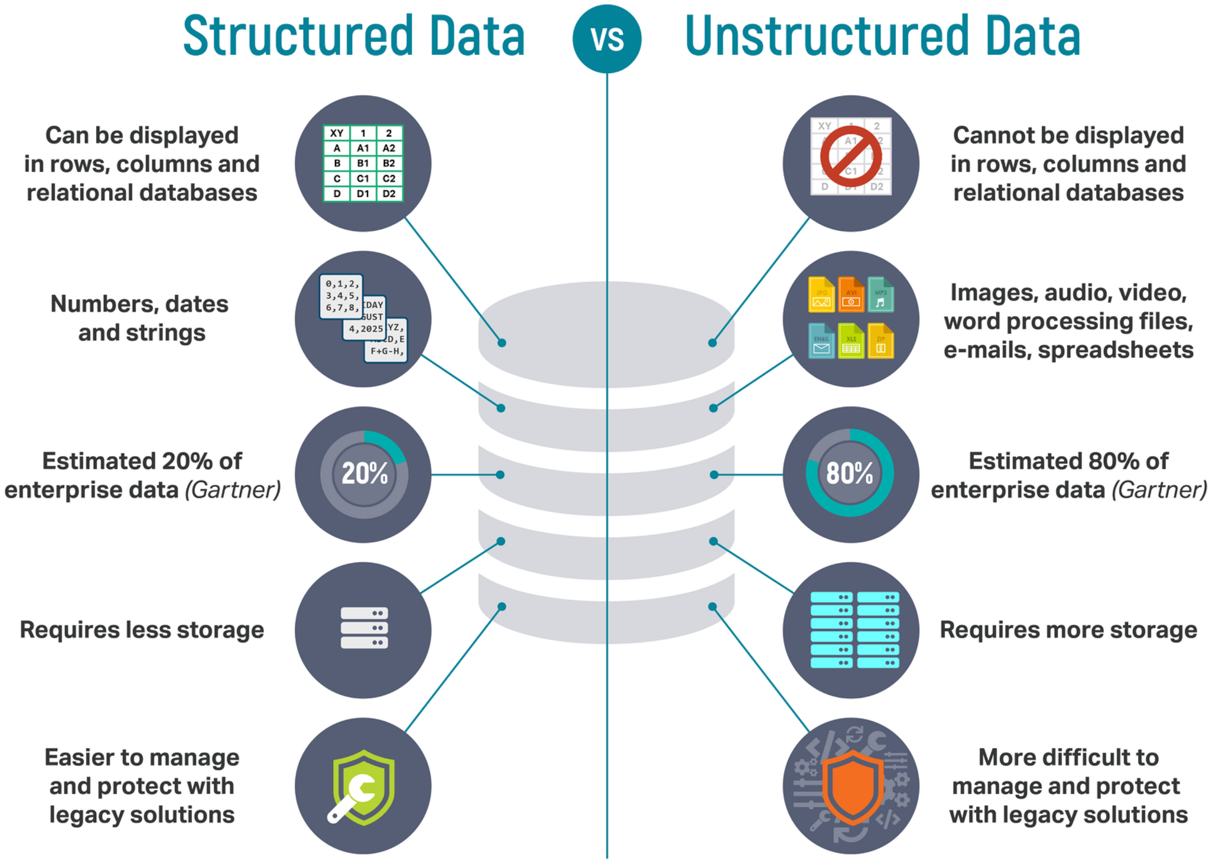
Transaction in Relational Database

- All database transactions must be ACID compliant to ensure data integrity.
- **Atomicity** requires that either transaction as a whole be successfully executed or if a part of the transaction fails, then the entire transaction be invalidated.
- **Consistency** mandates the data written to the database as part of the transaction must adhere to all defined rules, and restrictions including constraints, cascades, and triggers.
- **Isolation** is critical to achieving concurrency control and makes sure each transaction is independent unto itself.
- **Durability** requires that all of the changes made to the database be permanent once a transaction is successfully completed.

https://www.tutorialspoint.com/sql/sql-transactions.htm

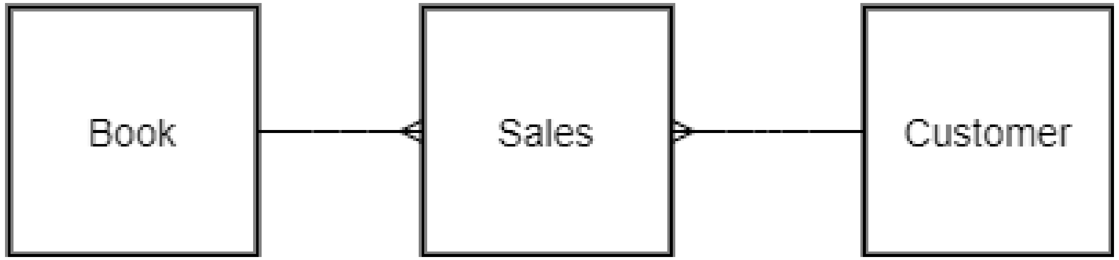
Structured data vs unstructured data

- Relational databases work well with structured data since each table's **schema** is always clearly defined.
- However, with the increasing number of ways to gather and generate data, we often need to deal with unstructured data.

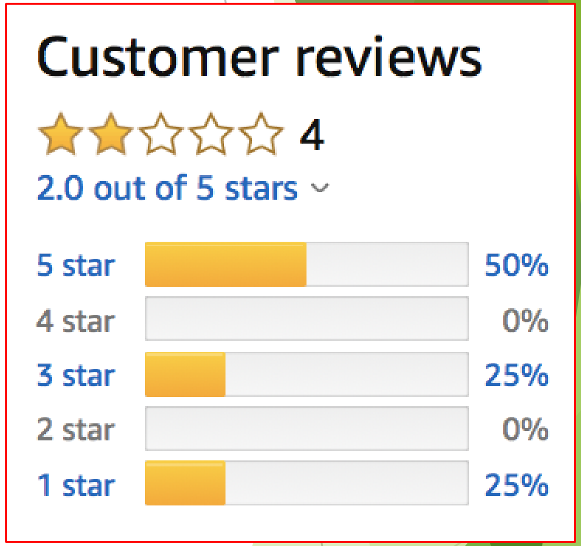

Structured Data
VS
Unstructured Data
Can be displayed in rows, columns and relational databases
Cannot be displayed in rows, columns and relational databases
Numbers, dates and strings
Images, audio, video, word processing files, e-mails, spreadsheets
20%
Estimated 20% of enterprise data *(Gartner)*
80%
Estimated 80% of enterprise data *(Gartner)*
Requires less storage
Requires more storage
Easier to manage and protect with legacy solutions
More difficult to manage and protect with legacy solutions

Issues with Relational database

- Consider an online bookstore that uses a RDBMS to manage its customer, book and sales data.



- To keep up with times, the business owner decides to add a book rating and review feature to allow customers to review the books they have purchased.
- He also wants to incorporate social media elements on his website to allow users to create profile, add other users who share similar interests in books so that they can post on each others' walls and like each others' pictures/posts.



Customer reviews

★★☆☆☆ 4

2.0 out of 5 stars

5 star	▇▇▇▇▇	50%
4 star		0%
3 star	▇▇▇	25%
2 star		0%
1 star	▇▇▇	25%

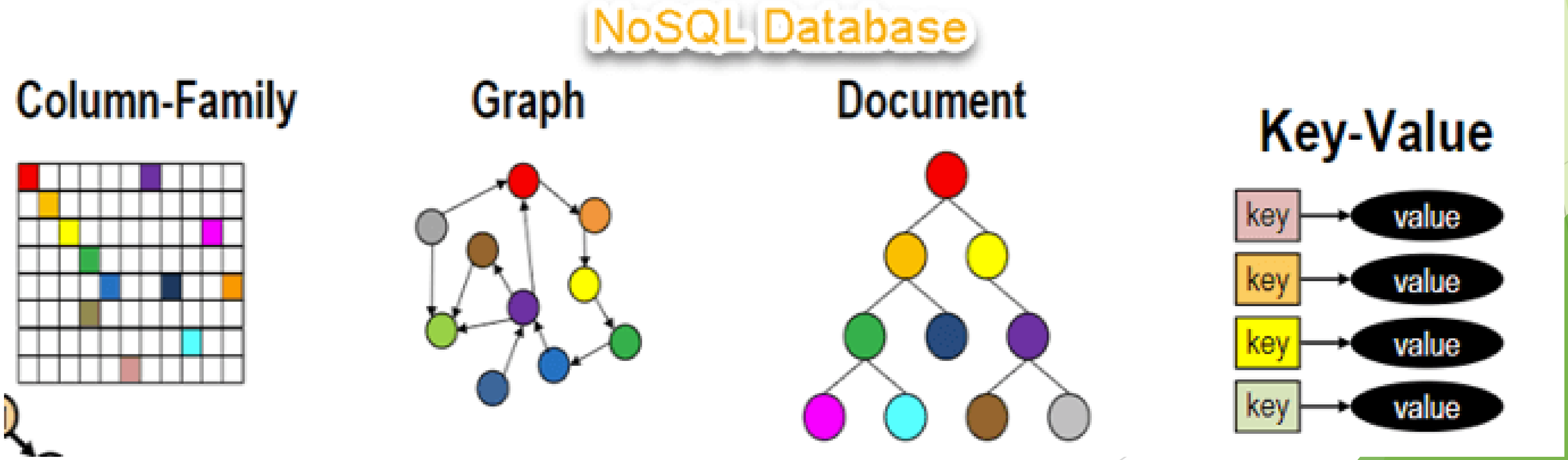
Issues with Relational database

- But in order to do that, he would need to insert new columns (e.g. rating, review) to existing tables, add new tables into the database, and re-establish the relationships between tables and so on……
- In this case, non-relational databases, also referred to as **NoSQL databases**, can offer a better choice.
- NoSQL databases used by Google, Amazon, eBay, Facebook, SEGA, Carousell, Uber, Grab, SG buses, Governments and many more!

https://www.mongodb.com/who-uses-mongodb

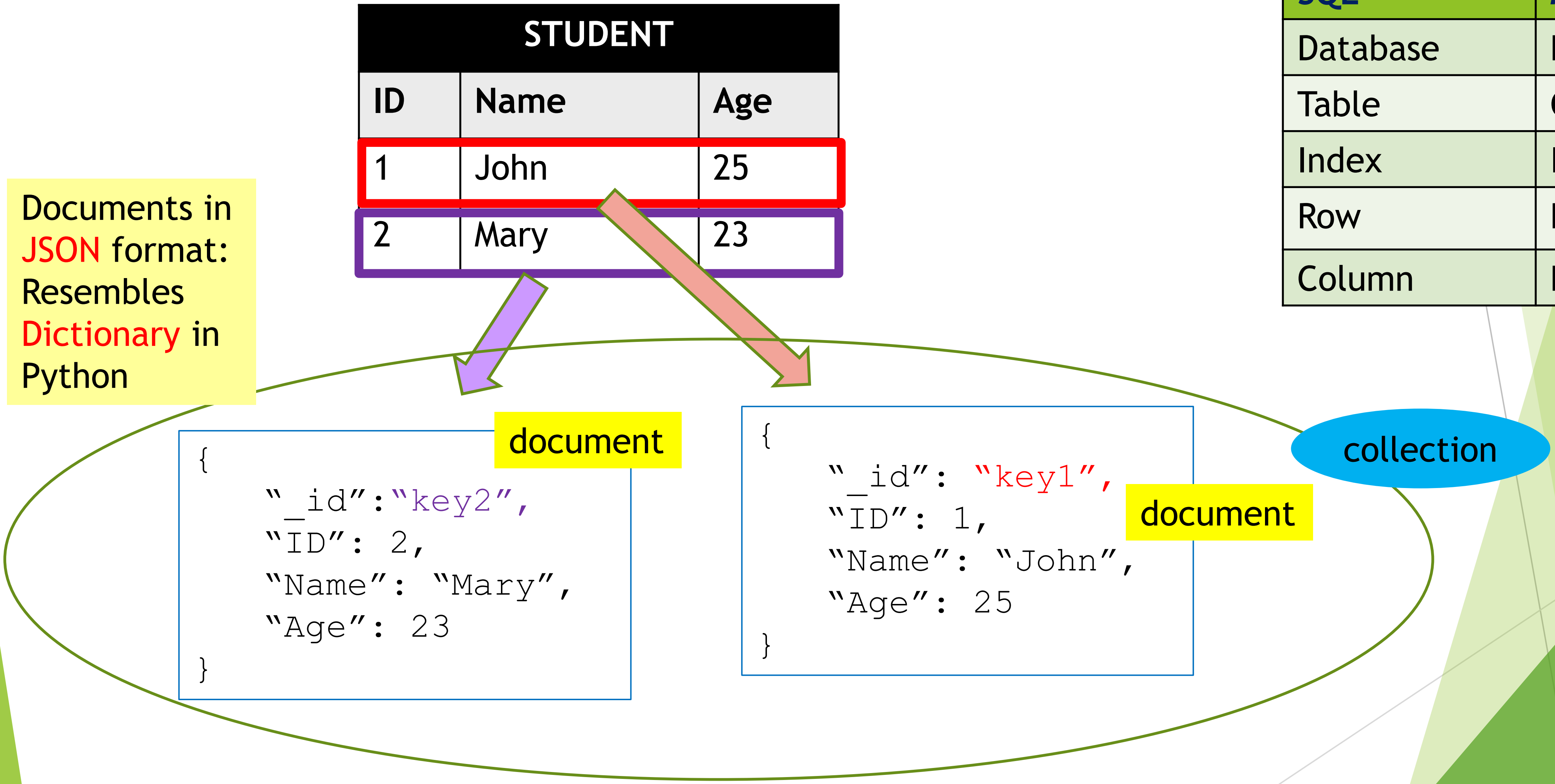
NoSQL database

- 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.
- Each key can point to an embedded key-value structure, also known as a **document**, instead of just a single value.
- The documents are in JavaScript Object Notation (JSON) format.



NoSQL document database structure

Documents in JSON format: Resembles Dictionary in Python



STUDENT		
ID	Name	Age
1	John	25
2	Mary	23

```
{
      “_id”:“key2”,
      “ID”: 2,
      “Name”: “Mary”,
      “Age”: 23
}
```

```
{
      “_id”: “key1”,
      “ID”: 1,
      “Name”: “John”,
      “Age”: 25
}
```

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

Differences between relational and NoSQL

Relational		NoSQL
Rigid, fixed, predefined schema	**Data model (schema)**	Flexible, no predefined schema, dynamic and can change easily
Represents data in tables and rows	**Database structure**	Stores data as collections of documents
Relationships between tables are established using foreign keys.	**Data relationships**	Relationships between collections are established via referencing and embedding
Joins are usually used to get data across tables, hence easier for complex queries	**Performing queries**	Can only query one collection at a time, limited query capabilities.
Stores data in a single data server	**Storage of data in server**	Provides sharding (data spread across different servers)

How NoSQL addresses shortcomings of RDBMS

- **1. High flexibility/adaptability:**
  - Relational db: predefined schema- difficult/costly to change, unable to support storing and processing of unstructured data
  - NoSQL databases: flexible schema, able to accommodate changes to business requirements, suitable for storing and processing unstructured data, and more natural compared to relational

New column

STUDENT			
ID	Name	Age	CG
1	John	25	S15
2	Mary	23	S16

document

```
{
      “_id”: “key1”
      “ID”: 1,
      “Name”: “John”,
      “Age”: 25,
      “CG”: “S15”
}
```

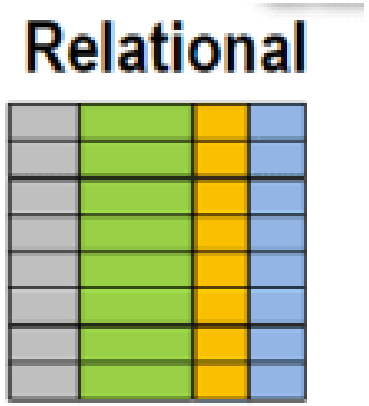
document

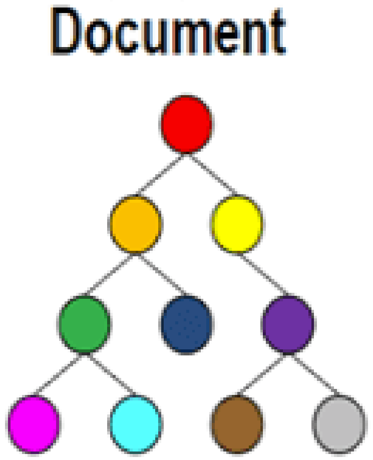
```
{
      “_id”: “key2”
      “ID”: 2,
      “Name”: “Mary”,
      “Age”: 23,
      “CG”: “S16”,
      “CCA”: “dance”,
      “tutors”: [
                {“name”: “Peter”,
                  “subject”: “Math”},
                {“name”: “Lenin”,
                  “subject”: “Hist”},
                {“name”: “Adam”,
                  “subject”: “Eng”},
               ]
}
```

Embedded- document

How NoSQL addresses shortcomings of RDBMS

- **2. Lower Cost of storage:**
  - Unlike NoSQL databases, relational databases do not usually support **hierarchical data storage** where less frequently-used data is moved to 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.





How NoSQL addresses shortcomings of RBDMS

- **3. Better Performance Scalability:**
  - Relational databases are mainly vertically scalable while NoSQL databases are mainly horizontally scalable.

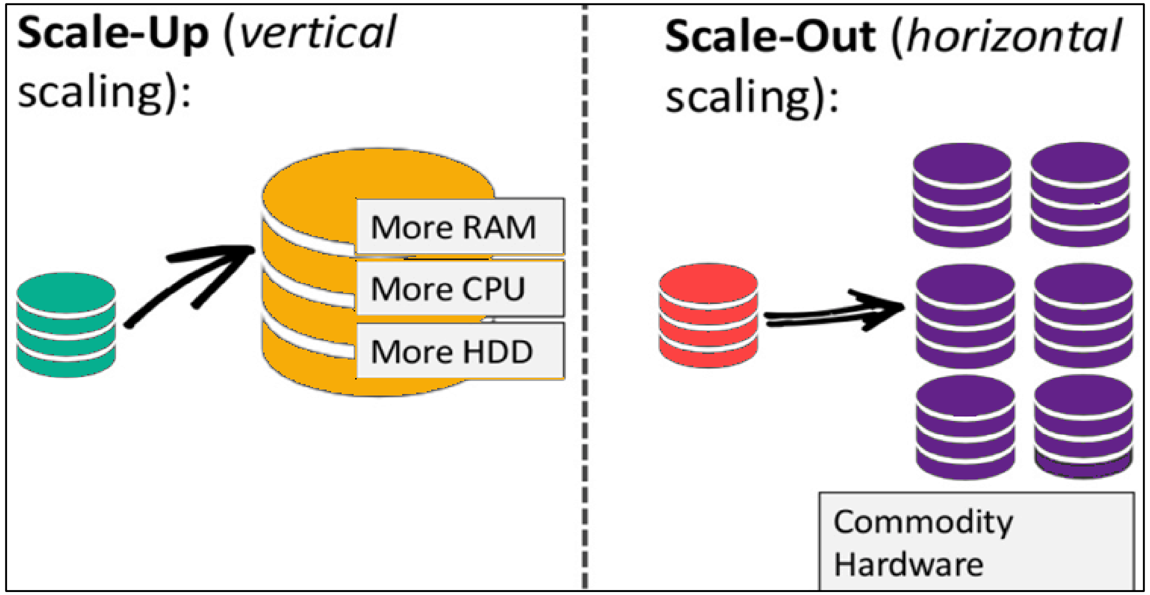
Vertically scalable:

→ requires upgrading an existing server with faster processors and more memory.

→ high-performance components can be expensive and upgrades are limited by the capacity of a single machine

Relational

NoSQL



Scale-Up (*vertical* scaling):
More RAM
More CPU
More HDD
Scale-Out (*horizontal* scaling):
Commodity Hardware

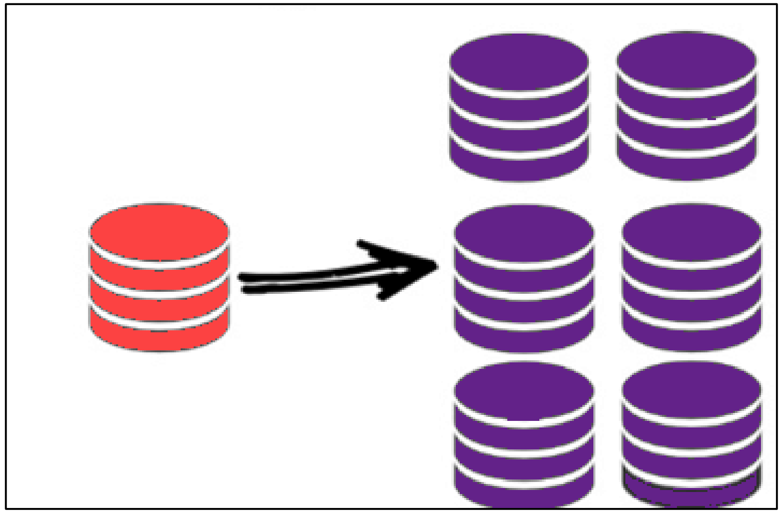
Horizontally scalable:

→ performance improved by simply increasing the number of servers

→ relatively cheaper as mass-produced average-performance computers are easily available at low prices

How NoSQL addresses shortcomings of RBDMS

- **4. High Availability:**
  - Relational databases are stored in a server, which makes the database unavailable when the server fails (down time).
  - NoSQL databases are designed to take advantage of multiple servers so that if one server fails, the other servers can continue to support applications (100% up time), therefore higher availability compared to relational.



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.

Use SQL databases if:	Use NoSQL databases if:
The data being stored has a fixed schema.	The data being stored has a dynamic schema, (i.e., unstructured data with flexible data types).
Complex and varied queries will be frequently performed.	Data storage needs to be performed quickly.
The atomicity, consistency, isolation and durability (ACID) properties are critical to the database (e.g sales transaction, banking/finance)	There will be an extremely large amount of data (i.e., Big Data for real time analytics).
There will be a high number of simultaneous transactions.	ACID properties are not critical

Designing schema in Document-based NoSQL

- Always design according to user requirements
- Objects which you want to use together should be combined into one document.
- Otherwise they should be separated.
- Optimize schema for more frequent use cases
- Model relationships:
  - Embedded
  - Reference

Embedded Relationships

- Documents within Documents

one-to-one

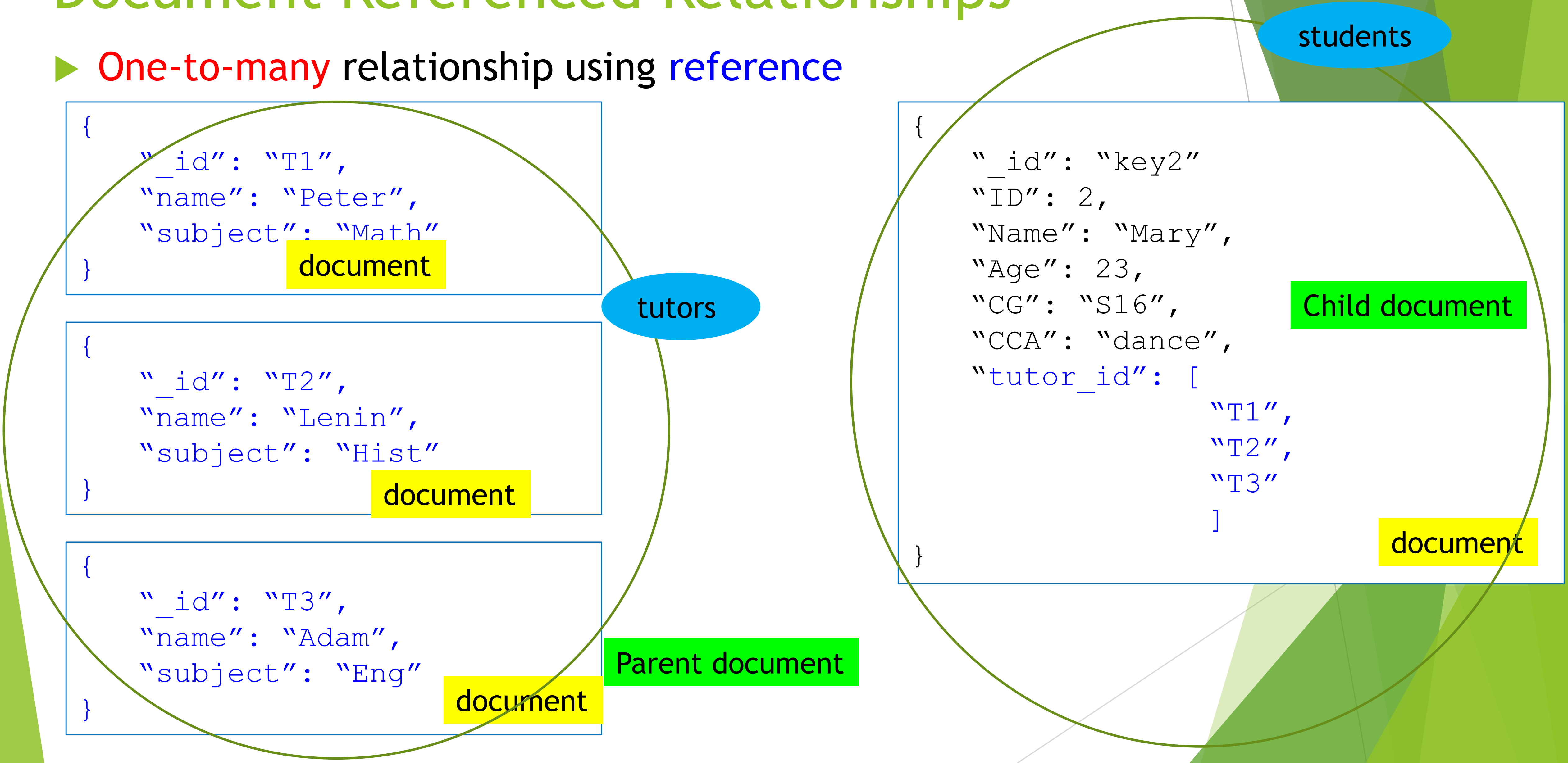
```
{
    “_id”: “key1”
    “ID”: 1,
    “Name”: “John”,
    “Age”: 25,
    “Address”: {
                  “street”: “Summer Drive”,
                  “city”: “Las Vegas”,
                  “state”: “Nevada”,
                  “country”: “United States”
               }
}
```

one-to-many

```
{
    “_id”: “key2”
    “ID”: 2,
    “Name”: “Mary”,
    “Age”: 23,
    “CG”: “S16”,
    “CCA”: “dance”,
    “tutors”: [
                {“name”: “Peter”,
                  “subject”: “Math”},
                {“name”: “Lenin”,
                  “subject”: “Hist”},
                {“name”: “Adam”,
                  “subject”: “Eng”},
              ]
}
```

Document Referenced Relationships

- One-to-many relationship using reference



```
{
    “_id”: “T1”,
    “name”: “Peter”,
    “subject”: “Math”
}
```

```
{
    “_id”: “T2”,
    “name”: “Lenin”,
    “subject”: “Hist”
}
```

```
{
    “_id”: “T3”,
    “name”: “Adam”,
    “subject”: “Eng”
}
```

```
{
    “_id”: “key2”
    “ID”: 2,
    “Name”: “Mary”,
    “Age”: 23,
    “CG”: “S16”,
    “CCA”: “dance”,
    “tutor_id”: [
                   “T1”,
                   “T2”,
                   “T3”
                   ]
}
```

Relational → NoSQL

- Consider the online bookstore example previously.

Customer

ID	Name
A1	John

Sales

CID*	ISBN*	Qty	Date
A1	9781935182870	1	18/4/20

Book

ISBN	Title	Author	Price
9781935182870	MongoDB in Action	Kyle Banker	31.20

```
{   "_id": "custA1",
    "ID": A1,
    "Name": "John",
    "SalesRecords": [ { "Date": "18/4/20",
                        "Book": { "_id": "book1",
                                  "ISBN": 9781935182870,
                                  "Title": "MongoDB in Action",
                                  "Author": Kyle Banker,
                                  "Price": 31.20 },
                        "Qty": 1
                      }
                    ]
}
```

customer

Embedded-document

Relational → NoSQL

- Consider the online bookstore example previously.

Customer

ID	Name
A1	John

Sales

CID*	ISBN*	Qty	Date
A1	9781935182870	1	18/4/20

Book

ISBN	Title	Author	Price
9781935182870	MongoDB in Action	Kyle Banker	31.20

customer

```
{   “_id”: “custA1”,
    “ID”: A1,
    “Name”: “John”,
    “SalesRecords”: [ { “Date”: “18/4/20”,
                        “Book_id”: “book1”
                        “Qty”: 1
                        }
                    ]
}
```

Document Reference

Child document

Book

```
{   “_id”: “book1”,
    “ISBN”: 9781935182870,
    “Title”: “MongoDB in Action”,
    “Author”: Kyle Banker,
    “Price”: 31.20 }
```

Parent document

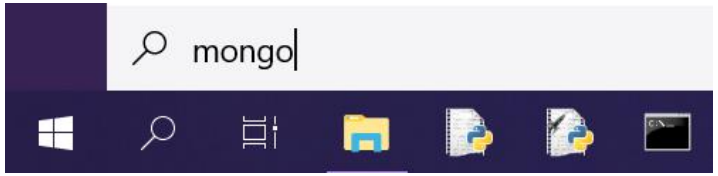
Launching MongoDB server and client

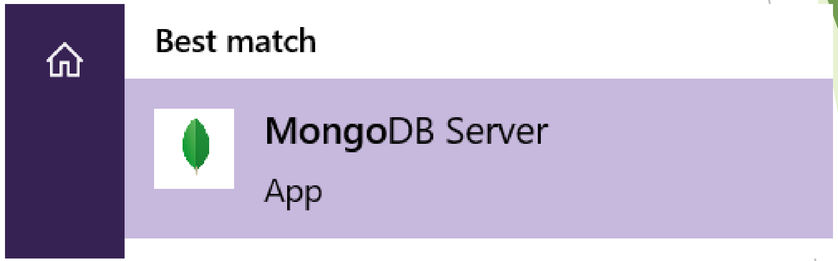
Using applications

Note: This section is for those with Mongo installed properly. If you followed the instructions for this section and Mongo does not launch, proceed to the next section on using command prompt to launch Mongo.

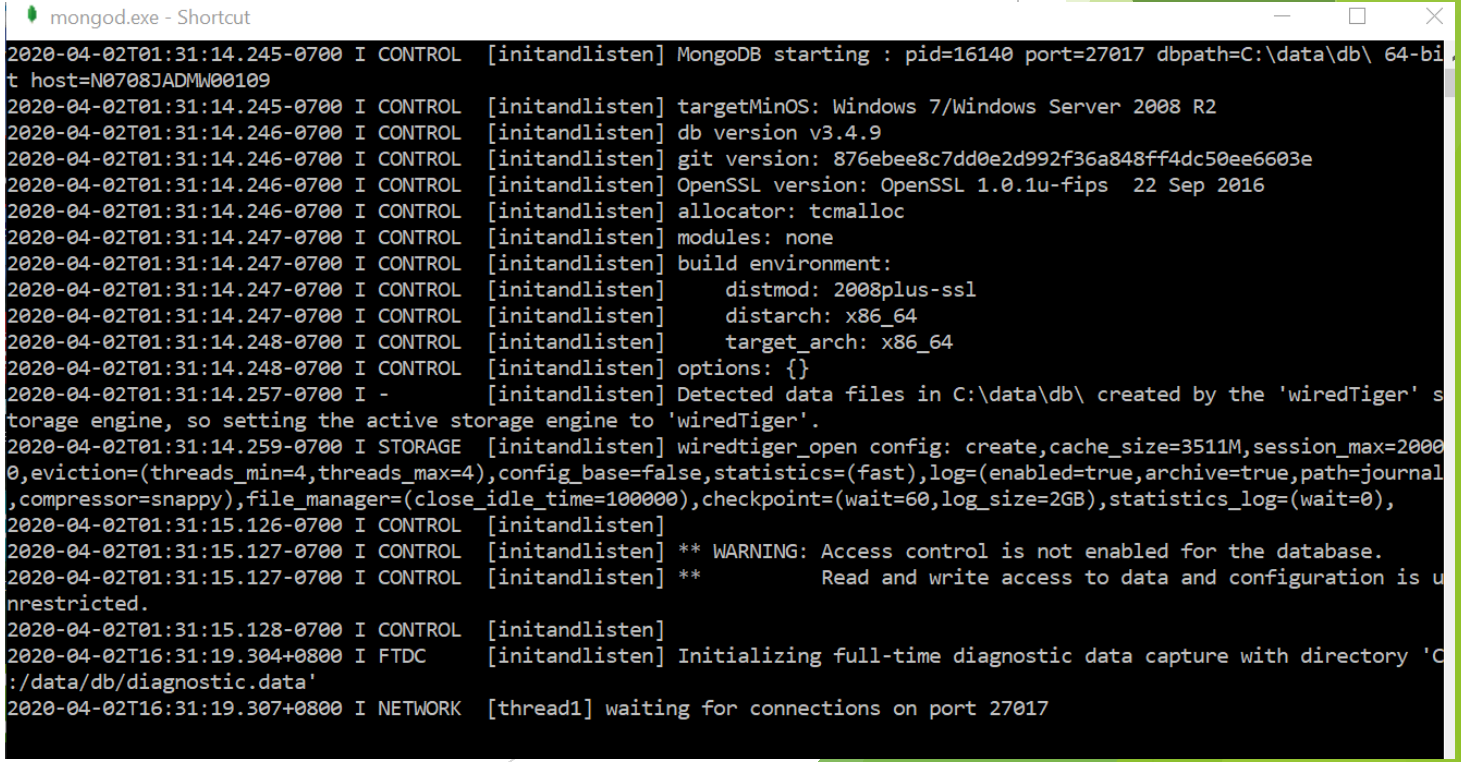
Check that your MongoDB server is installed successfully on your laptop

- Search for MongoDB Server App on your computer





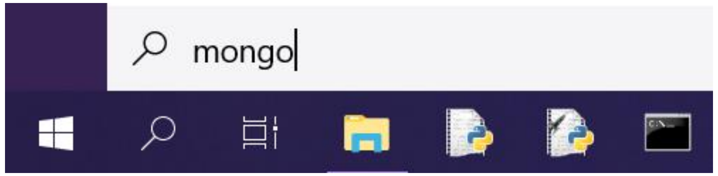
- If MongoDB is installed on your computer, you would be able to find the MongoDB Server App. Click it and you will see the command shell with the ‘waiting for connections on port 27017’



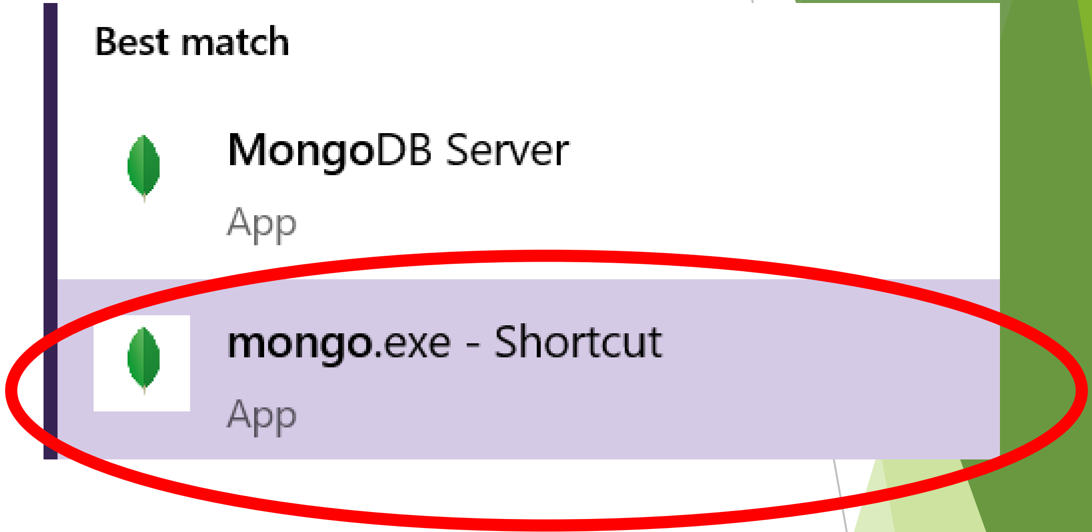
```
2020-04-02T01:31:14.245-0700 I CONTROL  [initandlisten] MongoDB starting : pid=16140 port=27017 dbpath=C:\data\db\ 64-bi
t host=N0708JADMW00109
2020-04-02T01:31:14.245-0700 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2020-04-02T01:31:14.246-0700 I CONTROL  [initandlisten] db version v3.4.9
2020-04-02T01:31:14.246-0700 I CONTROL  [initandlisten] git version: 876ebee8c7dd0e2d992f36a848ff4dc50ee6603e
2020-04-02T01:31:14.246-0700 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips  22 Sep 2016
2020-04-02T01:31:14.246-0700 I CONTROL  [initandlisten] allocator: tcmalloc
2020-04-02T01:31:14.247-0700 I CONTROL  [initandlisten] modules: none
2020-04-02T01:31:14.247-0700 I CONTROL  [initandlisten] build environment:
2020-04-02T01:31:14.247-0700 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2020-04-02T01:31:14.247-0700 I CONTROL  [initandlisten]     distarch: x86_64
2020-04-02T01:31:14.248-0700 I CONTROL  [initandlisten]     target_arch: x86_64
2020-04-02T01:31:14.248-0700 I CONTROL  [initandlisten] options: {}
2020-04-02T01:31:14.257-0700 I -        [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' s
torage engine, so setting the active storage engine to 'wiredTiger'.
2020-04-02T01:31:14.259-0700 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=3511M,session_max=2000
0,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),
2020-04-02T01:31:15.126-0700 I CONTROL  [initandlisten]
2020-04-02T01:31:15.127-0700 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-02T01:31:15.127-0700 I CONTROL  [initandlisten] **          Read and write access to data and configuration is u
nrestricted.
2020-04-02T01:31:15.128-0700 I CONTROL  [initandlisten]
2020-04-02T16:31:19.304+0800 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory 'C
:/data/db/diagnostic.data'
2020-04-02T16:31:19.307+0800 I NETWORK  [thread1] waiting for connections on port 27017
```

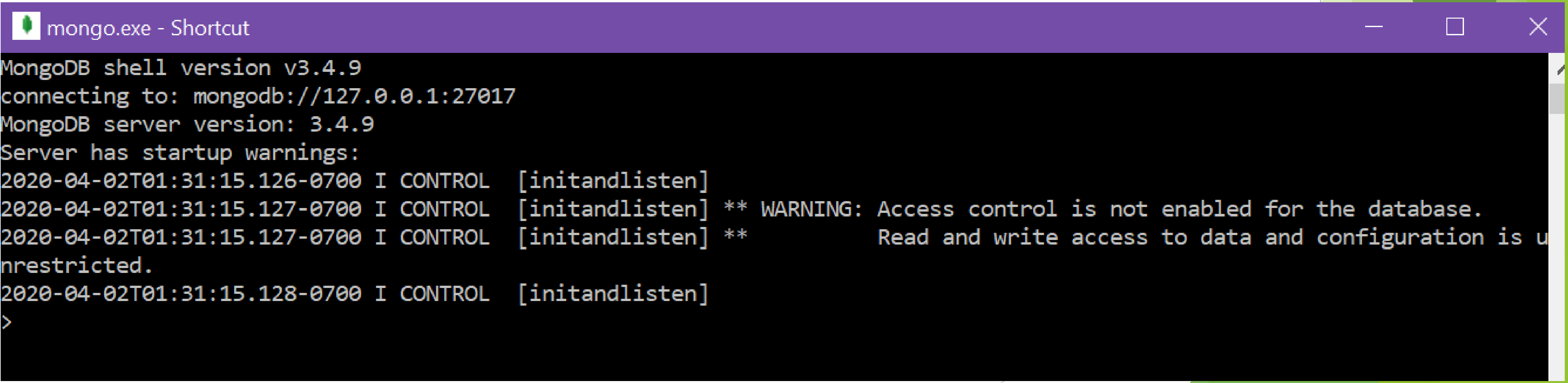
Launch the Mongo client

- Search for mongo.exe on your computer



- Click it and you will see the mongo command shell

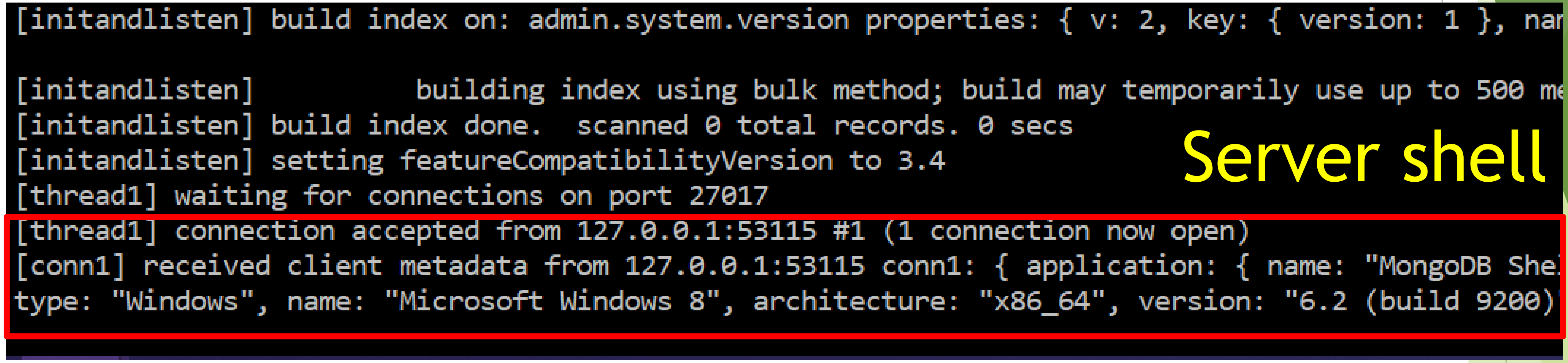




```
MongoDB shell version v3.4.9
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.9
Server has startup warnings:
2020-04-02T01:31:15.126-0700 I CONTROL  [initandlisten]
2020-04-02T01:31:15.127-0700 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-02T01:31:15.127-0700 I CONTROL  [initandlisten] **          Read and write access to data and configuration is u
nrestricted.
2020-04-02T01:31:15.128-0700 I CONTROL  [initandlisten]
>
```

Launching Mongo Client

- Notice on the server shell that a connection has established:



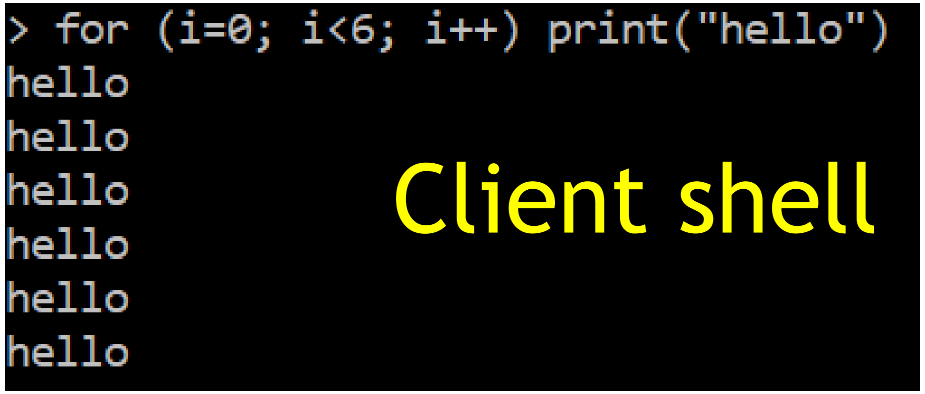
```
[initandlisten] build index on: admin.system.version properties: { v: 2, key: { version: 1 }, nar

[initandlisten]          building index using bulk method; build may temporarily use up to 500 me
[initandlisten] build index done.  scanned 0 total records. 0 secs
[initandlisten] setting featureCompatibilityVersion to 3.4
[thread1] waiting for connections on port 27017
[thread1] connection accepted from 127.0.0.1:53115 #1 (1 connection now open)
[conn1] received client metadata from 127.0.0.1:53115 conn1: { application: { name: "MongoDB She
type: "Windows", name: "Microsoft Windows 8", architecture: "x86_64", version: "6.2 (build 9200)
```

Server shell

Test Mongo Client

The Mongo Shell uses JavaScript. Test the MongoDB client shell by typing:

- 2+2 ##you should get 4
- for (i=0; i<5;i++) print(“Hello”) ##you should get ‘Hello’ printed 5 times



```
> for (i=0; i<6; i++) print("hello")
hello
hello
hello
hello
hello
hello
```

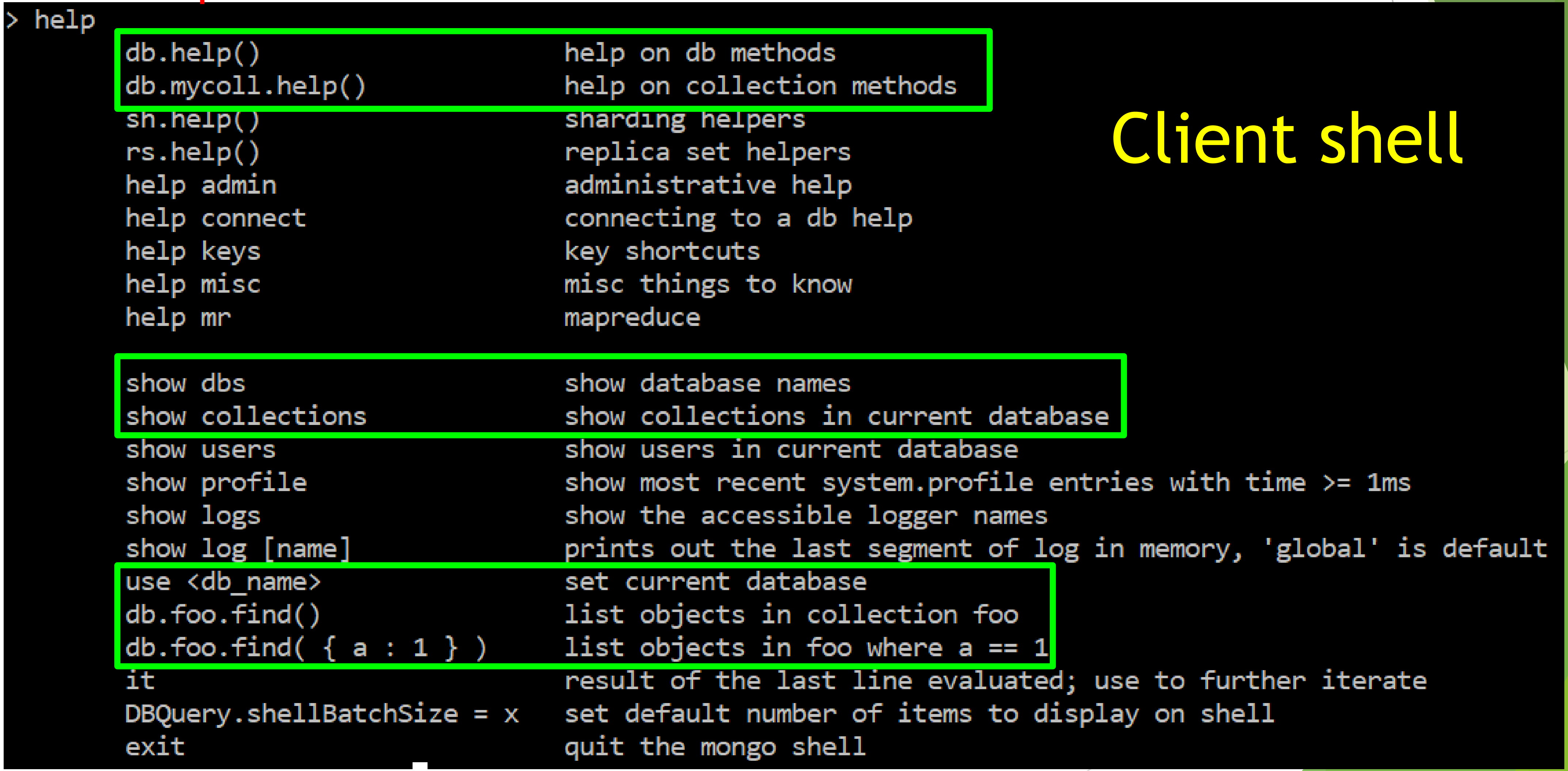
Client shell

- “Hello, World!”.replace(“World”, “MongoDB”) ##you should get ‘Hello’ MongoDB
- x=100
- x/5 ##you should get 20

Test Mongo Client

Access the help menu by typing help and explore the options boxed up in green:

- help

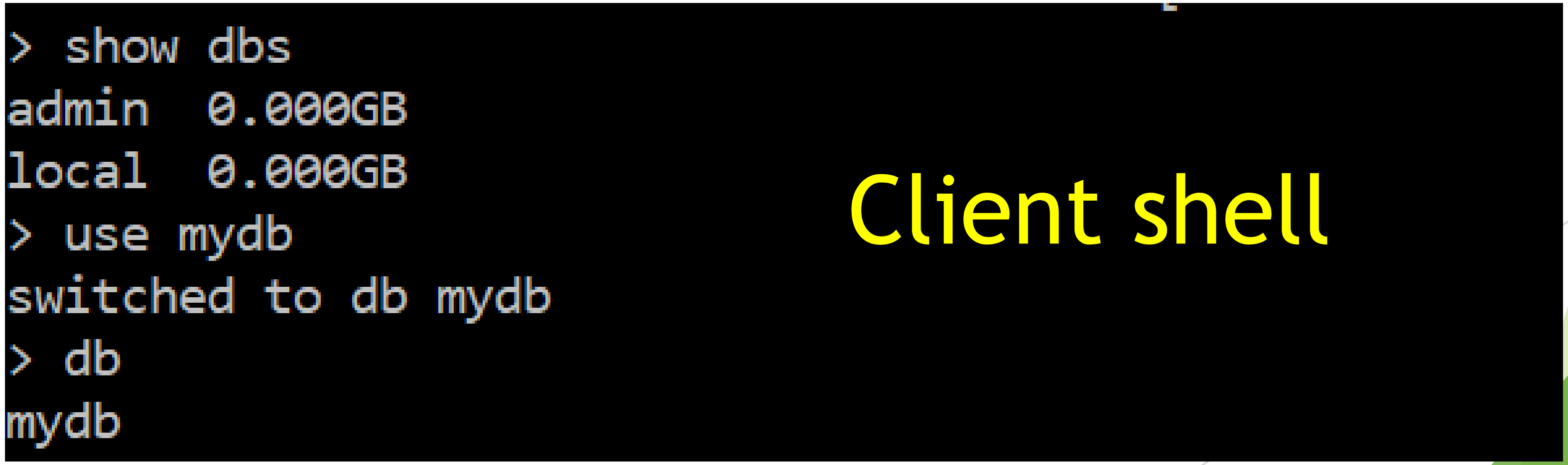


```
> help
        db.help()                    help on db methods
        db.mycoll.help()             help on collection methods
        sh.help()                    sharding helpers
        rs.help()                    replica set helpers
        help admin                   administrative help
        help connect                 connecting to a db help
        help keys                    key shortcuts
        help misc                    misc things to know
        help mr                      mapreduce

        show dbs                     show database names
        show collections             show collections in current database
        show users                   show users in current database
        show profile                 show most recent system.profile entries with time >= 1ms
        show logs                    show the accessible logger names
        show log [name]              prints out the last segment of log in memory, 'global' is default
        use <db_name>                set current database
        db.foo.find()                list objects in collection foo
        db.foo.find( { a : 1 } )     list objects in foo where a == 1
        it                           result of the last line evaluated; use to further iterate
        DBQuery.shellBatchSize = x   set default number of items to display on shell
        exit                         quit the mongo shell
```

Test Mongo Client

Creating your first database

- Show all available databases
  - show dbs #you should get local (and maybe also admin)
- Select a particular database to access, e.g. mydb. This will create mydb if it does not already exist:
  - use mydb #you should get ‘switched to db mydb’
- Check for current database which is in use:
  - db #you should get ‘mydb’



```
> show dbs
admin   0.000GB
local   0.000GB
> use mydb
switched to db mydb
> db
mydb
```

Client shell

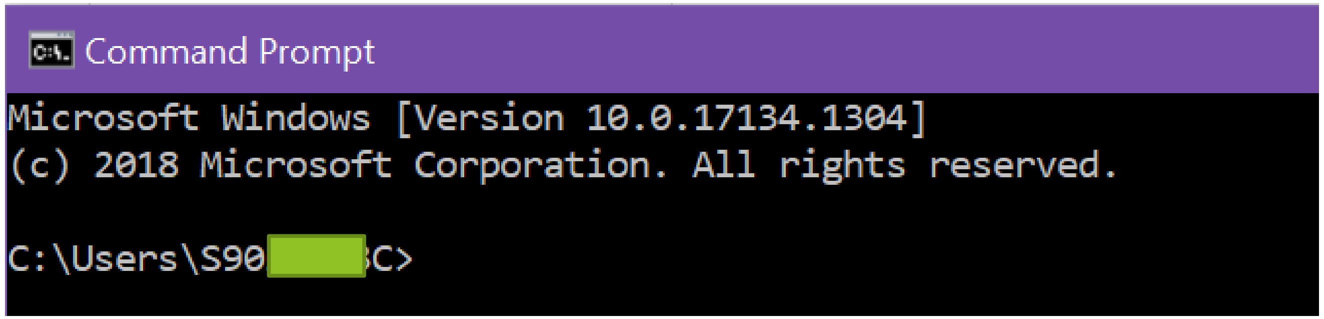
OPTIONAL: This slide onwards are meant for those with issues with launching MongoDB

Setting up MongoDB server and client

Using command prompt

1) Launching Mongo Server

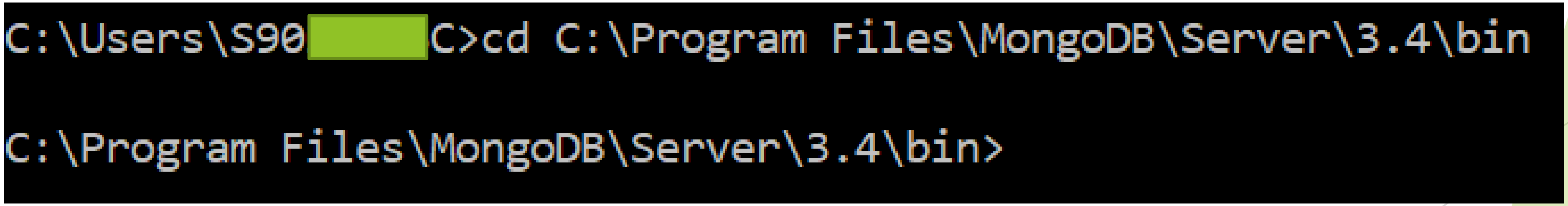
1. Launch Command Prompt by typing cmd after clicking windows icon



```
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\S90[redacted]C>
```

2. Change directory to where the MongoDB is install by typing:

- cd C:\Program Files\MongoDB\Server\3.4\bin



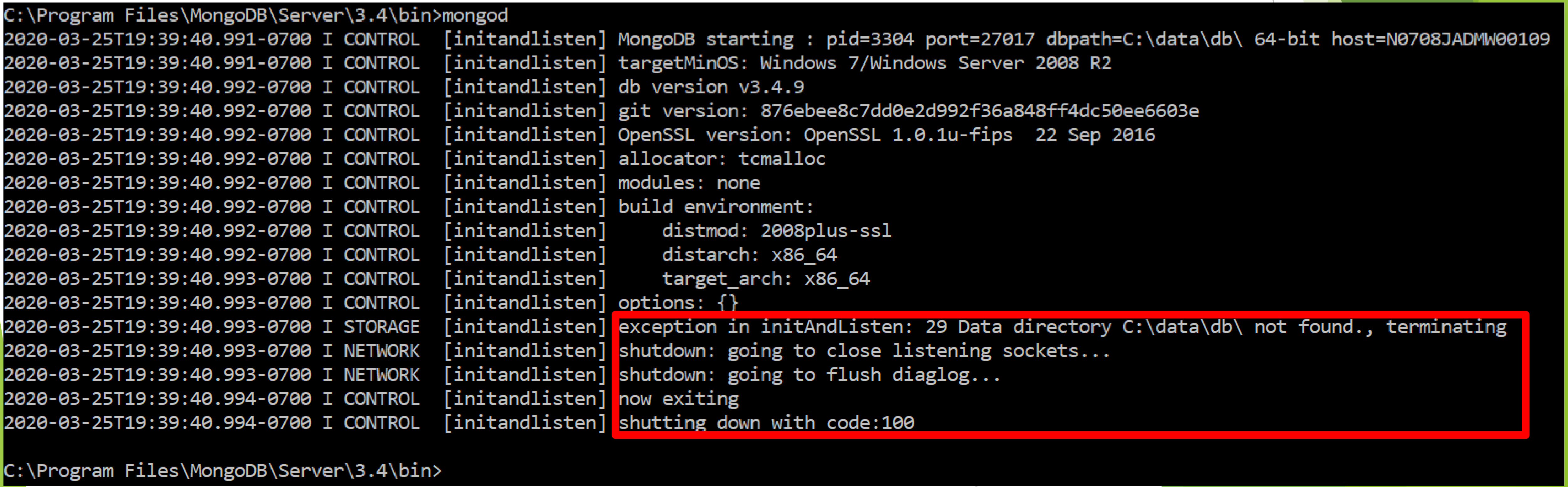
```
C:\Users\S90[redacted]C>cd C:\Program Files\MongoDB\Server\3.4\bin

C:\Program Files\MongoDB\Server\3.4\bin>
```

1) Launching Mongo Server

3. Run the MongoDB server by typing:

    - mongod

4. If you see the following exception, that means the server is not launched successfully due to a missing Data directory in C:\data\db\

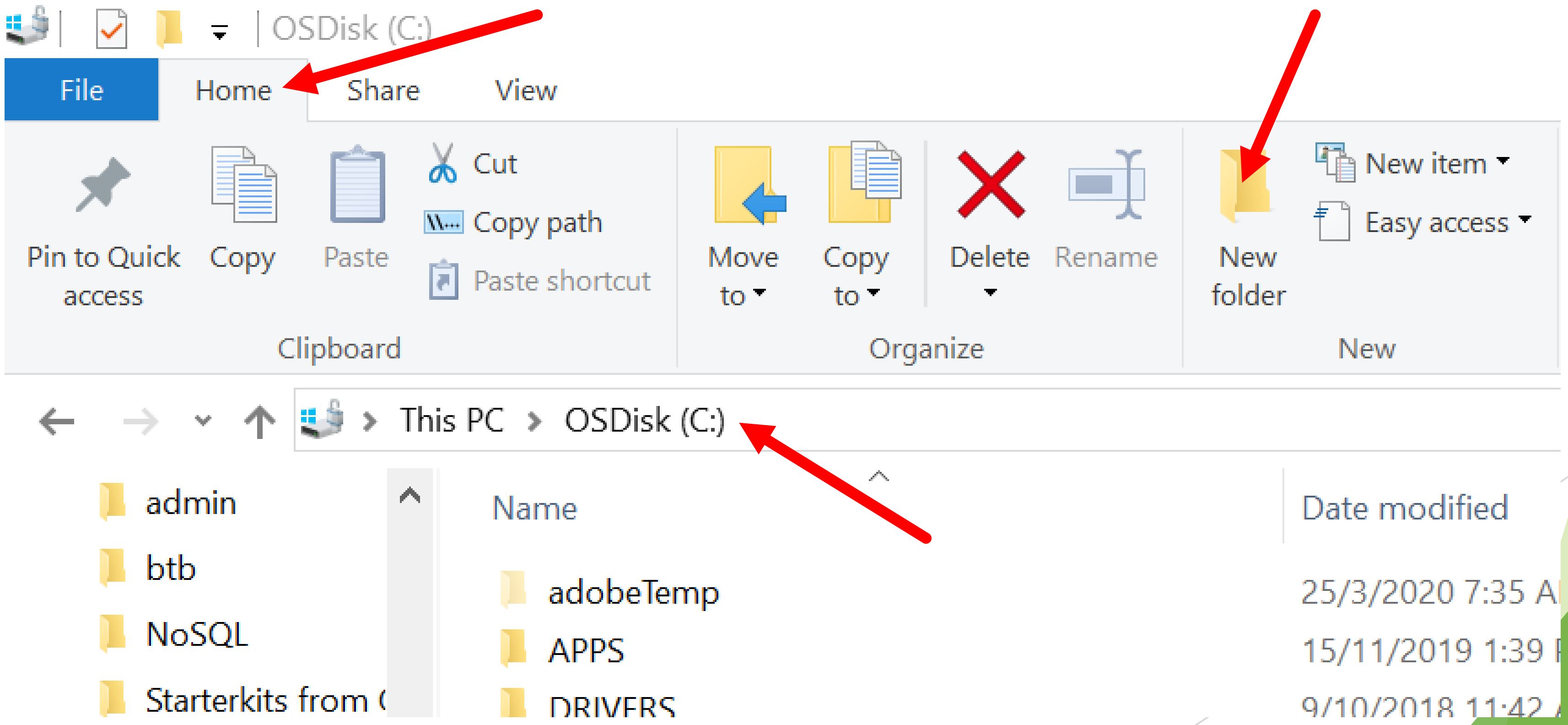


```
C:\Program Files\MongoDB\Server\3.4\bin>mongod
2020-03-25T19:39:40.991-0700 I CONTROL  [initandlisten] MongoDB starting : pid=3304 port=27017 dbpath=C:\data\db\ 64-bit host=N0708JADMW00109
2020-03-25T19:39:40.991-0700 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten] db version v3.4.9
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten] git version: 876ebee8c7dd0e2d992f36a848ff4dc50ee6603e
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips  22 Sep 2016
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten] allocator: tcmalloc
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten] modules: none
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten] build environment:
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2020-03-25T19:39:40.992-0700 I CONTROL  [initandlisten]     distarch: x86_64
2020-03-25T19:39:40.993-0700 I CONTROL  [initandlisten]     target_arch: x86_64
2020-03-25T19:39:40.993-0700 I CONTROL  [initandlisten] options: {}
2020-03-25T19:39:40.993-0700 I STORAGE  [initandlisten] exception in initAndListen: 29 Data directory C:\data\db\ not found., terminating
2020-03-25T19:39:40.993-0700 I NETWORK  [initandlisten] shutdown: going to close listening sockets...
2020-03-25T19:39:40.993-0700 I NETWORK  [initandlisten] shutdown: going to flush diaglog...
2020-03-25T19:39:40.994-0700 I CONTROL  [initandlisten] now exiting
2020-03-25T19:39:40.994-0700 I CONTROL  [initandlisten] shutting down with code:100

C:\Program Files\MongoDB\Server\3.4\bin>
```

1) Launching Mongo Server

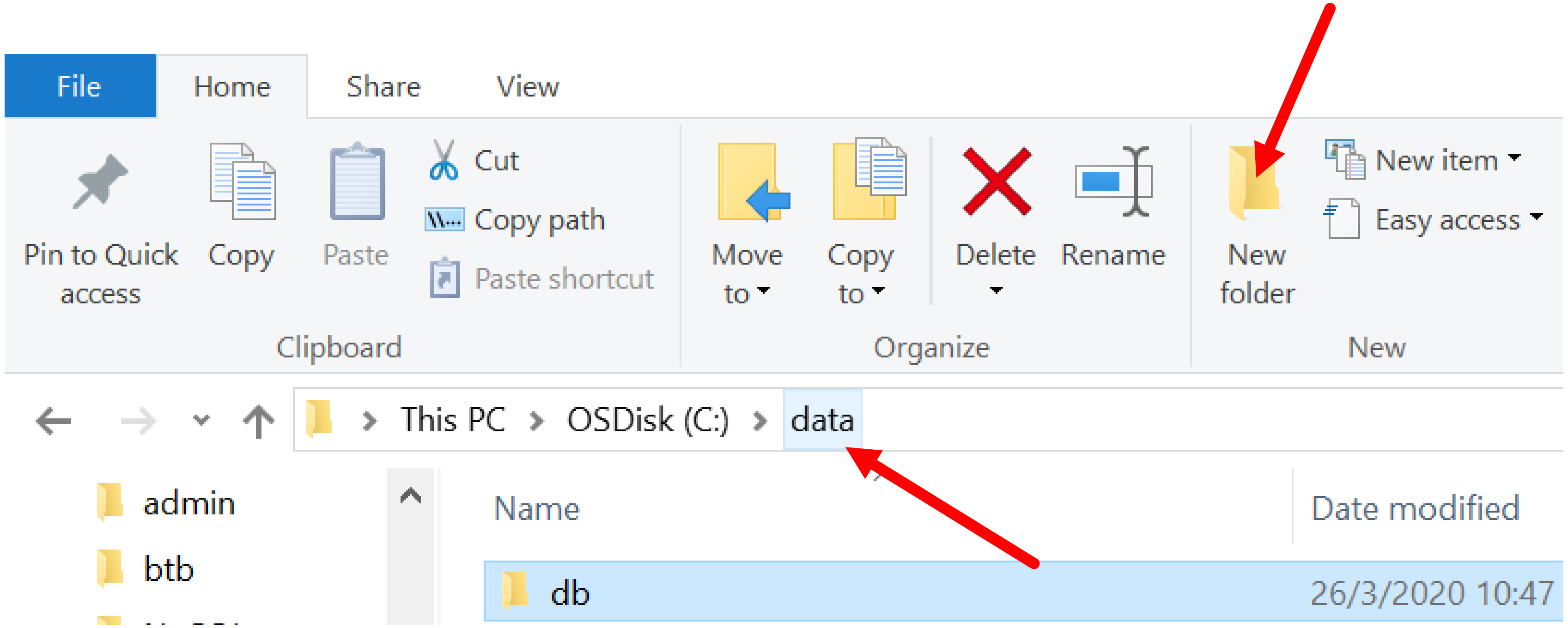
5. To rectify the issue, we create the data directory C:\data\db in C drive as follows:

- Go to file explorer> This PC > C:
- Click on New folder and create a new folder, naming it as data



1) Launching Mongo Server

6. In the data folder, create another new folder, naming it db:



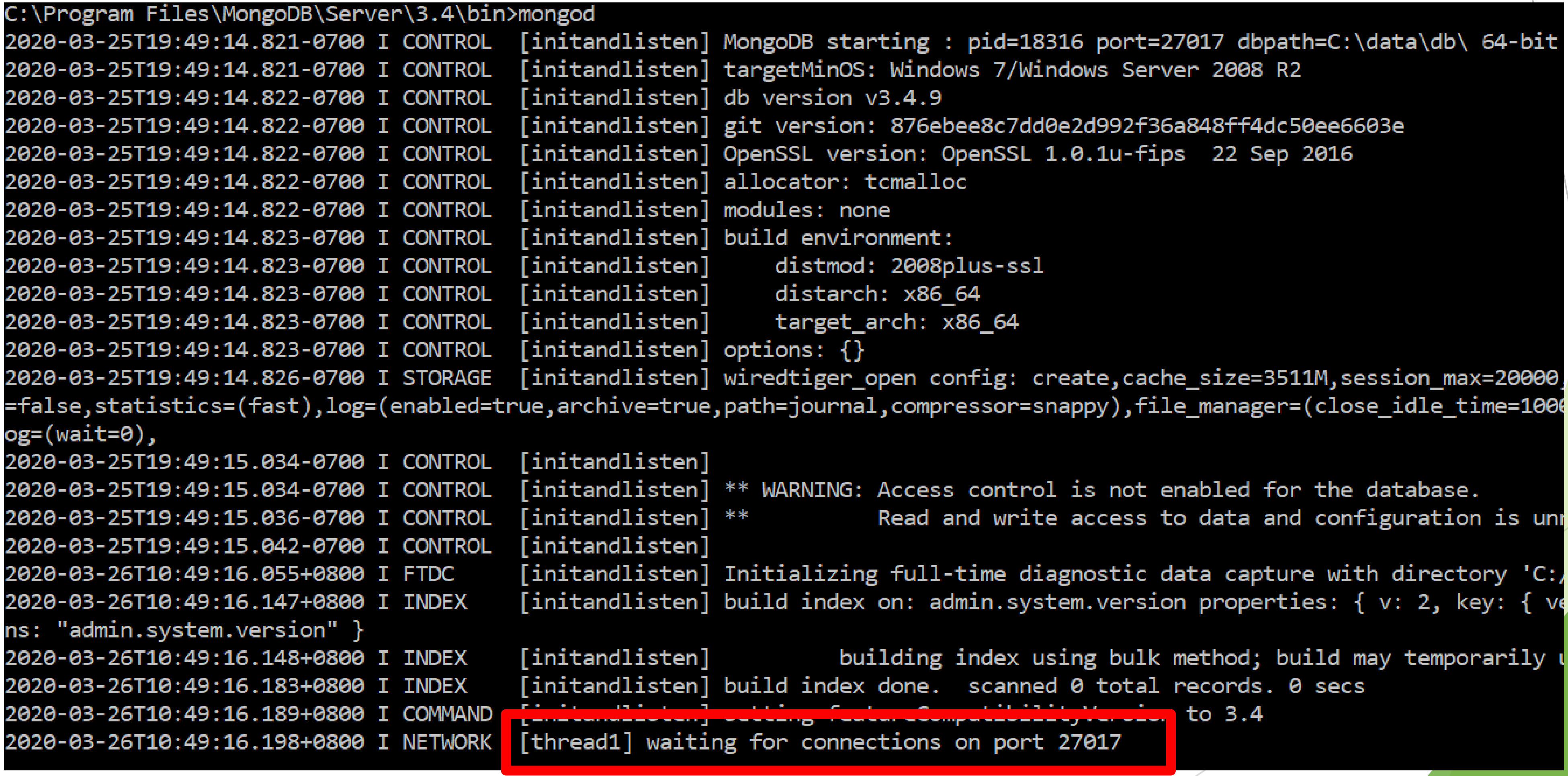
db 26/3/2020 10:47

1) Launching Mongo Server

7. Re-Run the MongoDB server by typing:

   - mongod

- This time, there should be no error:



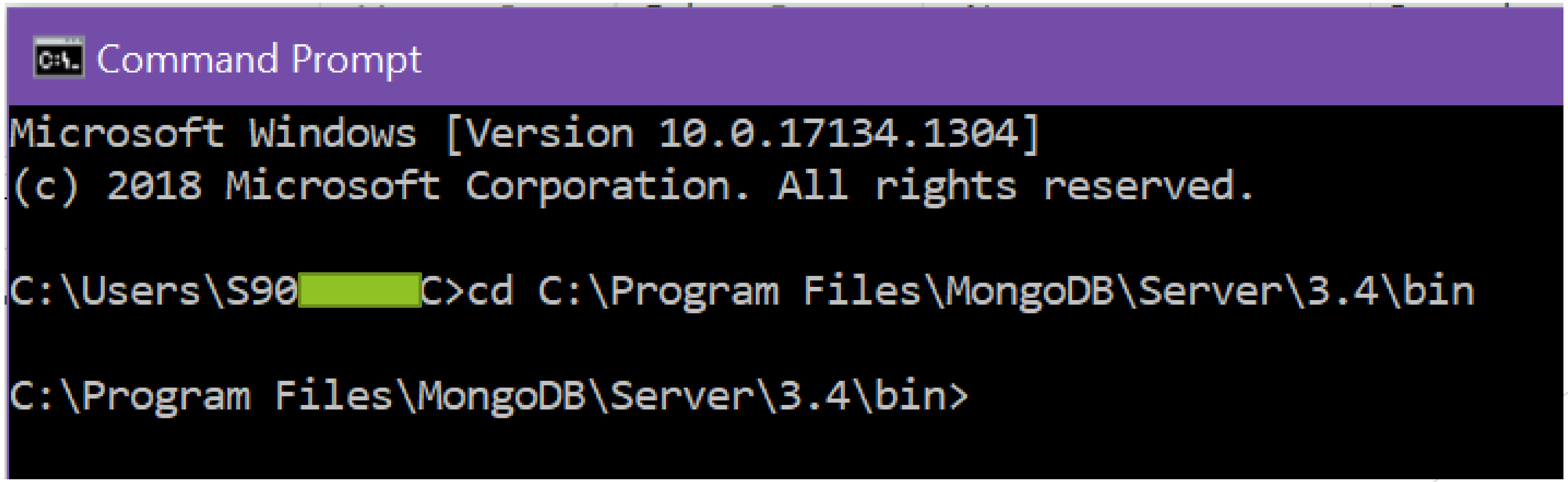
```
C:\Program Files\MongoDB\Server\3.4\bin>mongod
2020-03-25T19:49:14.821-0700 I CONTROL  [initandlisten] MongoDB starting : pid=18316 port=27017 dbpath=C:\data\db\ 64-bit
2020-03-25T19:49:14.821-0700 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2020-03-25T19:49:14.822-0700 I CONTROL  [initandlisten] db version v3.4.9
2020-03-25T19:49:14.822-0700 I CONTROL  [initandlisten] git version: 876ebee8c7dd0e2d992f36a848ff4dc50ee6603e
2020-03-25T19:49:14.822-0700 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips  22 Sep 2016
2020-03-25T19:49:14.822-0700 I CONTROL  [initandlisten] allocator: tcmalloc
2020-03-25T19:49:14.822-0700 I CONTROL  [initandlisten] modules: none
2020-03-25T19:49:14.823-0700 I CONTROL  [initandlisten] build environment:
2020-03-25T19:49:14.823-0700 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2020-03-25T19:49:14.823-0700 I CONTROL  [initandlisten]     distarch: x86_64
2020-03-25T19:49:14.823-0700 I CONTROL  [initandlisten]     target_arch: x86_64
2020-03-25T19:49:14.823-0700 I CONTROL  [initandlisten] options: {}
2020-03-25T19:49:14.826-0700 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=3511M,session_max=20000,
=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=1000
og=(wait=0),
2020-03-25T19:49:15.034-0700 I CONTROL  [initandlisten]
2020-03-25T19:49:15.034-0700 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-25T19:49:15.036-0700 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unr
2020-03-25T19:49:15.042-0700 I CONTROL  [initandlisten]
2020-03-26T10:49:16.055+0800 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory 'C:/
2020-03-26T10:49:16.147+0800 I INDEX    [initandlisten] build index on: admin.system.version properties: { v: 2, key: { ve
ns: "admin.system.version" }
2020-03-26T10:49:16.148+0800 I INDEX    [initandlisten]          building index using bulk method; build may temporarily u
2020-03-26T10:49:16.183+0800 I INDEX    [initandlisten] build index done.  scanned 0 total records. 0 secs
2020-03-26T10:49:16.189+0800 I COMMAND  [initandlisten] setting featureCompatibilityVersion to 3.4
2020-03-26T10:49:16.198+0800 I NETWORK  [thread1] waiting for connections on port 27017
```

2) Launching Mongo Client

- The message ‘waiting for connections on port 27017’ implies that the mongoDB server is up and running. The default port is 27017.

1. While keeping the server shell running, launch another command prompt shell and change directory to where the MongoDB is install by typing:

    - cd C:\Program Files\MongoDB\Server\3.4\bin

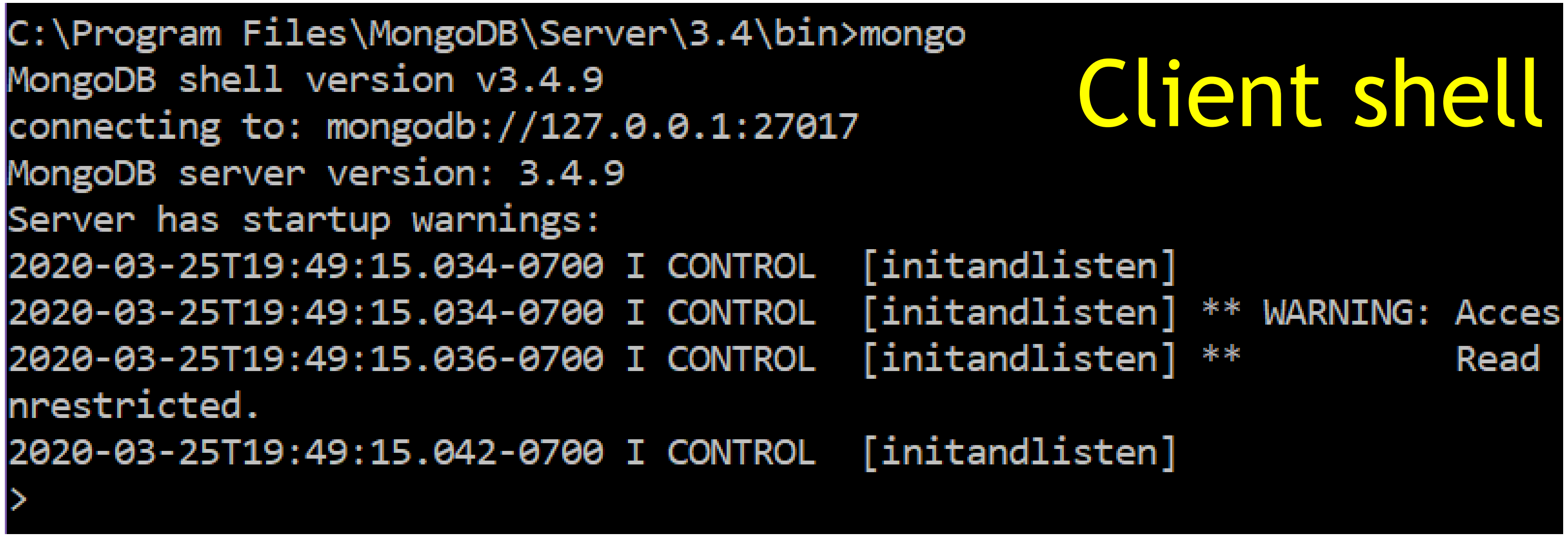


```
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\S90[redacted]C>cd C:\Program Files\MongoDB\Server\3.4\bin

C:\Program Files\MongoDB\Server\3.4\bin>
```

2) Launching Mongo Client

2. Run the MongoDB client shell by typing:

- mongo



```
C:\Program Files\MongoDB\Server\3.4\bin>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:
2020-03-25T19:49:15.034-0700 I CONTROL  [initandlisten]
2020-03-25T19:49:15.034-0700 I CONTROL  [initandlisten] ** WARNING: Acces
2020-03-25T19:49:15.036-0700 I CONTROL  [initandlisten] **          Read
nrestricted.
2020-03-25T19:49:15.042-0700 I CONTROL  [initandlisten]
>
```

Client shell

Creating desktop shortcuts for MongoDB server and client (optional)

- For those who prefer to access your MongoDB server and client using desktop icons, you may create shortcuts by following the next 2 slides.

3) Shortcut for Mongo Server

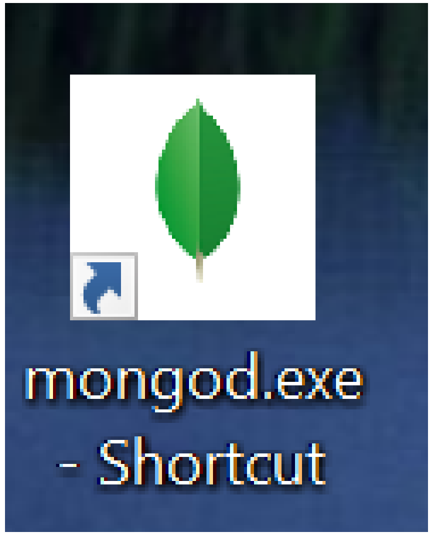
1. Proceed to the following directory using file explorer:

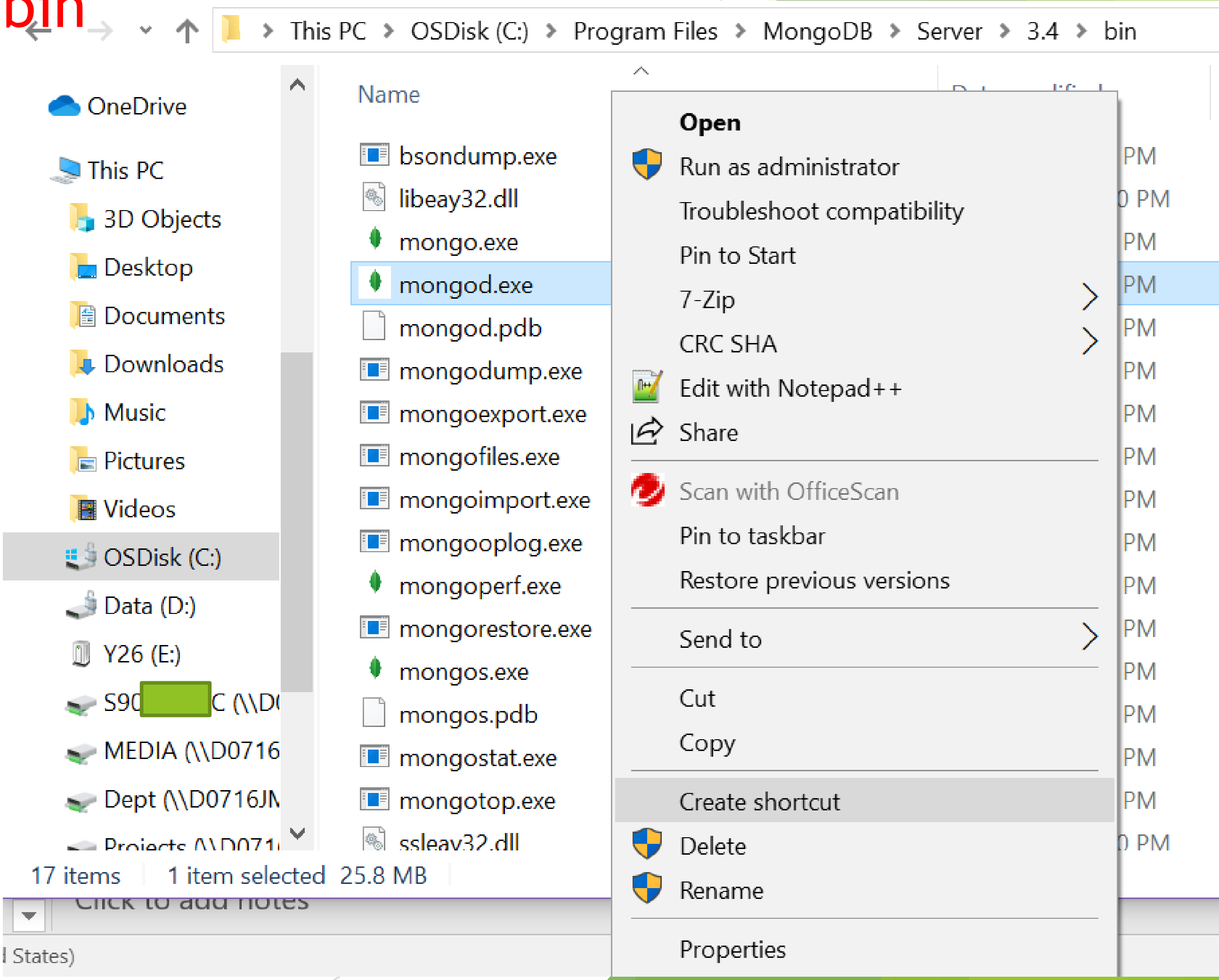
- C:\Program Files\MongoDB\Server\3.4\bin

2. Right click mongod and select

   Create shortcut:

- Place the shortcut on desktop





4) Shortcut for Mongo Client

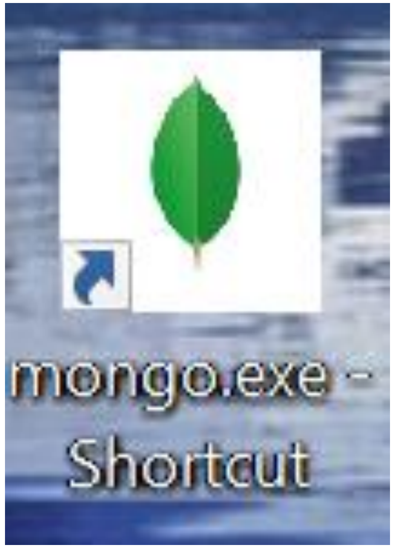
1. Proceed to the following directory using file explorer:

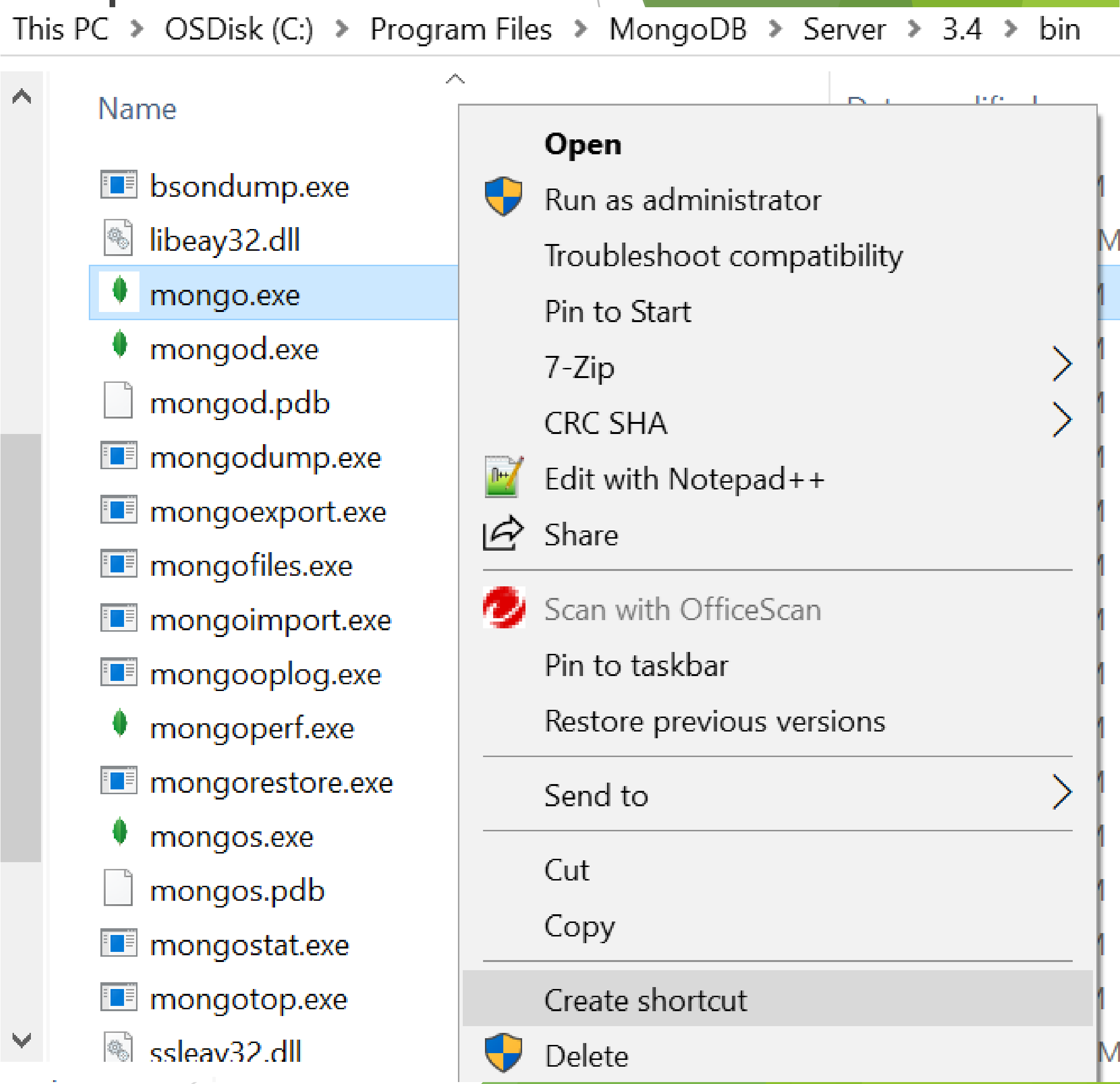
- C:\Program Files\MongoDB\Server\3.4\bin

2. Right click mongo and select

   Create shortcut:

- Place the shortcut on desktop




This PC › OSDisk (C:) › Program Files › MongoDB › Server › 3.4 › bin

Name

bsondump.exe
libeay32.dll
mongo.exe
mongod.exe
mongod.pdb
mongodump.exe
mongoexport.exe
mongofiles.exe
mongoimport.exe
mongooplog.exe
mongoperf.exe
mongorestore.exe
mongos.exe
mongos.pdb
mongostat.exe
mongotop.exe
ssleay32.dll