HOMEWORK: HOMEWORK 6.1

For this week's homework we will start with a standalone MongoDB database, turn it into a sharded cluster with two shards, and shard one of the collections. We will create a "dev" environment on our local box: no replica sets, and only one config server. In production you would almost always use three config servers and replica sets as part of a sharded cluster. In the final of the course we'll set up a larger cluster with replica sets and three config servers.

Download the handout.

Start an initially empty `mongod` database instance.

Connect to it with the shell and week6.js loaded:

```
$ mongo --shell localhost/week6 week6.js
```

Run `homework.init()`. It will take some time to run as it inserts quite a few documents. When it is done run

```
> db.trades.stats()
```

to check the status of the collection.

At this point we have a single mongod and would like to transform it into a sharded cluster with one shard. (We'll use this node's existing `week6.trades` data in the cluster.)

Stop the mongod process. Now, restart the mongod process adding the option `--shardsvr`. If you started mongod with a `--dbpath` option, specify that as well.

```
$ mongod --shardsvr …
```

Note that with --shardsvr specified the default port for mongod becomes 27018.

Start a mongo config server:

```
$ mongod --configsvr …
```

(Note with `--configsvr` specified the default port for listening becomes 27019 and the

default data directory /data/configdb. Wherever your data directory is, it is suggested that you verify that the directory is empty before you begin.)

Start a mongos:

```
$ mongos --configdb your_host_name:27019
```

Connect to mongos with the shell:

```
$ mongo --shell localhost/week6 week6.js
```

Add the first shard ("*your_host_name*:27018").

Verify that the week6.trades data is visible via mongos. Note at this point the week6 database isn't "sharding enabled" but its data is still visible via mongos:

```
> db.trades.find().pretty()
> db.trades.count()
> db.trades.stats()
```

Run homework.a() and enter the result below. This method will simply verify that this simple cluster is up and running and return a result key.

```
1
```

SHOW ANSWER

Submission deadline has passed.