MongoDB data management

- Query in mongo shell
- Export & import MongoDB
- Using and understanding iot data with Python or R

A5.9.8 MongoDB management

1. Query in Mongo shell

db.sensors.count() → sensors collection에 있는 도큐먼트 (문서)의 수

db.sensors.find().sort({_id: 1}).limit(10) → 오래된 document 10개 추출

db.sensors.find().sort({_id: -1}).limit(10) → 최근 document 10개 추출

db.sensors.find({date: {$gt: "2018-05-29 22:26:05"}}) → 특정 시간 이후 document 추출

db.sensors.find({temperature: {$gt: 29}}) → 온도가 29도를 넘는 document 추출

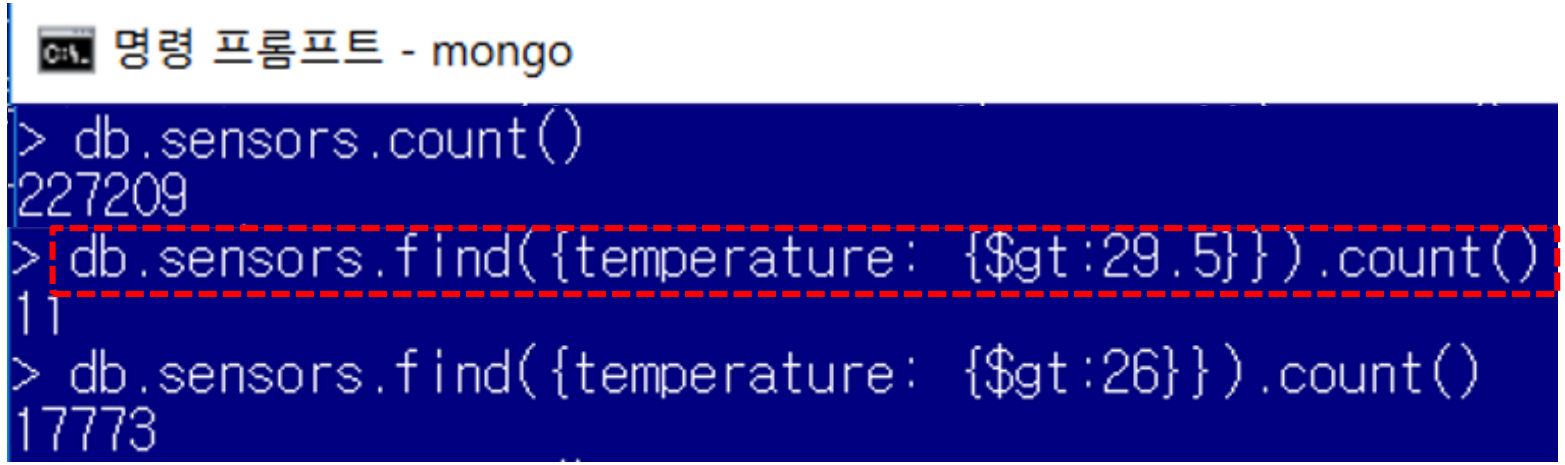
https://docs.mongodb.com/manual/tutorial/query-documents/

A5.9.8 MongoDB management

1.1 Query in Mongo shell

db.sensors.count() → sensors collection 에 있는 문서의 총수

db.sensors.find({temperature: {$gt: 29.5}}).count()
→ sensors collection 에 있는 온도가 **29.5**를 초과하는 문서의 수



A5.9.8 MongoDB management

1.2 Query in Mongo shell

db.sensors.find().sort({_id: -1}).limit(10) → 최근 데이터 10개 추출

명령 프롬프트 - mongo

```
> show dbs
Warning  0.000GB
iot11    0.013GB
local    0.000GB
> use iot11
switched to db iot11
> show collections
sensors
> db.sensors.find().sort({_id:-1}).limit(10)
{ "_id" : ObjectId("5b0d51f82d151211a8b9e2ef"), "date" : "2018-05-29 22:13:28.218", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51ed2d151211a8b9e2ee"), "date" : "2018-05-29 22:13:17.958", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51e32d151211a8b9e2ed"), "date" : "2018-05-29 22:13:07.713", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51d92d151211a8b9e2ec"), "date" : "2018-05-29 22:12:57.453", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51cf2d151211a8b9e2eb"), "date" : "2018-05-29 22:12:47.208", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51c42d151211a8b9e2ea"), "date" : "2018-05-29 22:12:36.947", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51ba2d151211a8b9e2e9"), "date" : "2018-05-29 22:12:26.687", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51b02d151211a8b9e2e8"), "date" : "2018-05-29 22:12:16.442", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d51a62d151211a8b9e2e7"), "date" : "2018-05-29 22:12:06.182", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
{ "_id" : ObjectId("5b0d519b2d151211a8b9e2e6"), "date" : "2018-05-29 22:11:55.937", "temperature" : "26.3", "humidity" :
 "49.8", "luminosity" : "0", "__v" : 0 }
>
```

사용 중인 **db** 이름으로 변경이 필요! **--- use iot**

A5.9.8 MongoDB management

1.3 Query in Mongo shell

db.sensors.find({temperature: {$gt: 29}}) → 29도 초과하는 문서추출

명령 프롬프트 - mongo

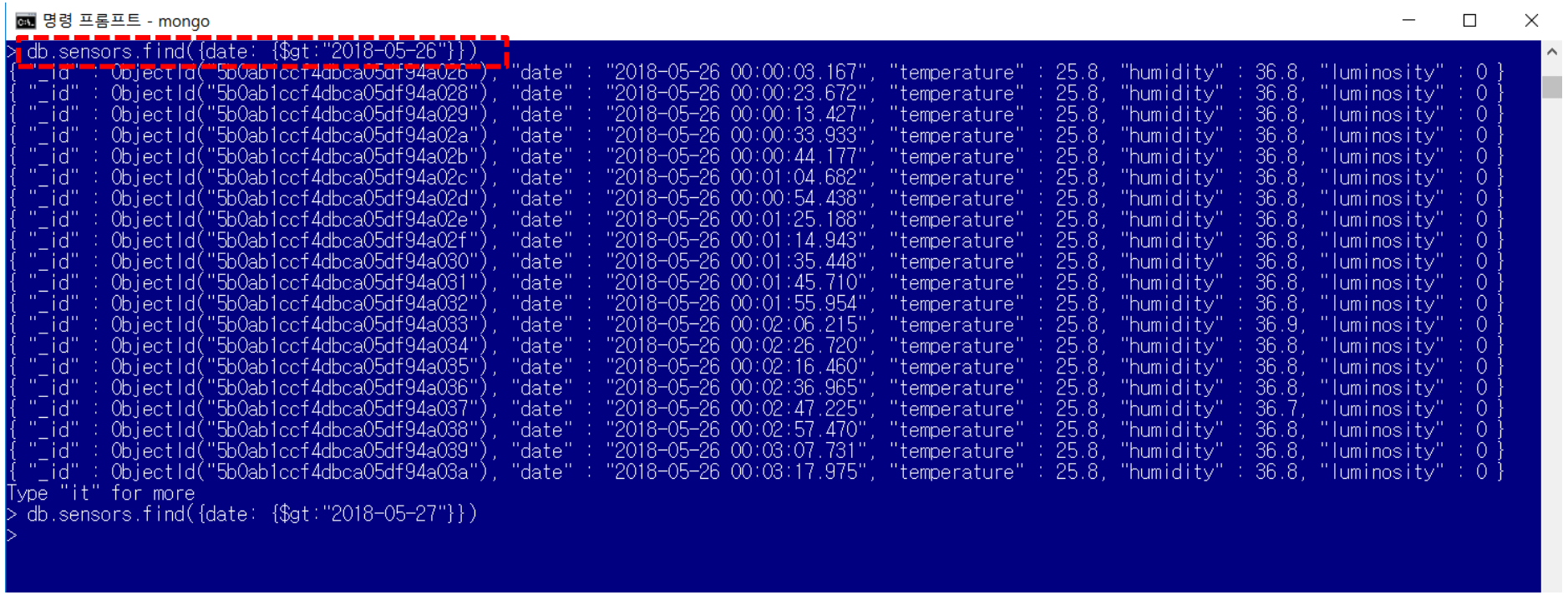
```
{ "_id" : ObjectId("5b0ab1c7f4dbca05df913fec"), "date" : "2018-03-12 09:17:59.512", "temperature" : 28.6, "humidity" : 13.7, "luminosity" : 60 }
Type "it" for more
> db.sensors.find({temperature: {$gt:29}})
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914266"), "date" : "2018-03-12 11:06:51.069", "temperature" : 29.1, "humidity" : 14.4, "luminosity" : 60 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91426b"), "date" : "2018-03-12 11:07:01.330", "temperature" : 29.2, "humidity" : 14.3, "luminosity" : 60 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91426c"), "date" : "2018-03-12 11:07:11.575", "temperature" : 29.1, "humidity" : 14.2, "luminosity" : 60 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914377"), "date" : "2018-03-12 11:52:49.318", "temperature" : 29.1, "humidity" : 14.4, "luminosity" : 57 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914378"), "date" : "2018-03-12 11:52:59.563", "temperature" : 29.2, "humidity" : 14.4, "luminosity" : 58 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914379"), "date" : "2018-03-12 11:53:09.826", "temperature" : 29.2, "humidity" : 14.3, "luminosity" : 58 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91437b"), "date" : "2018-03-12 11:53:20.069", "temperature" : 29.1, "humidity" : 14.3, "luminosity" : 57 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143a9"), "date" : "2018-03-12 12:01:21.996", "temperature" : 29.2, "humidity" : 14.7, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143aa"), "date" : "2018-03-12 12:01:32.258", "temperature" : 29.1, "humidity" : 14.6, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143ad"), "date" : "2018-03-12 12:02:03.008", "temperature" : 29.1, "humidity" : 14.5, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143ae"), "date" : "2018-03-12 12:02:13.268", "temperature" : 29.2, "humidity" : 14.4, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143af"), "date" : "2018-03-12 12:02:23.529", "temperature" : 29.3, "humidity" : 14.3, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b0"), "date" : "2018-03-12 12:02:33.774", "temperature" : 29.4, "humidity" : 14.2, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b1"), "date" : "2018-03-12 12:02:54.280", "temperature" : 29.4, "humidity" : 14.1, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b2"), "date" : "2018-03-12 12:02:44.035", "temperature" : 29.4, "humidity" : 14.2, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b3"), "date" : "2018-03-12 12:03:04.541", "temperature" : 29.4, "humidity" : 14, "luminosity" : 55 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b4"), "date" : "2018-03-12 12:03:14.785", "temperature" : 29.3, "humidity" : 13.9, "luminosity" : 54 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b5"), "date" : "2018-03-12 12:03:25.046", "temperature" : 29.2, "humidity" : 13.9, "luminosity" : 54 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143b6"), "date" : "2018-03-12 12:03:35.291", "temperature" : 29.1, "humidity" : 14, "luminosity" : 54 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9143eb"), "date" : "2018-03-12 12:12:38.735", "temperature" : 29.2, "humidity" : 14.7, "luminosity" : 53 }
Type "it" for more
> db.sensors.find({temperature: {$gt:31}})
> db.sensors.find({temperature: {$gt:30}})
> db.sensors.find({temperature: {$gt:29.5}})
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914427"), "date" : "2018-03-12 12:22:53.957", "temperature" : 29.6, "humidity" : 13.7, "luminosity" : 50 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914428"), "date" : "2018-03-12 12:23:04.218", "temperature" : 29.7, "humidity" : 13.6, "luminosity" : 50 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914429"), "date" : "2018-03-12 12:23:14.479", "temperature" : 29.7, "humidity" : 13.4, "luminosity" : 50 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91442a"), "date" : "2018-03-12 12:23:24.724", "temperature" : 29.7, "humidity" : 13.4, "luminosity" : 51 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91442b"), "date" : "2018-03-12 12:23:34.985", "temperature" : 29.7, "humidity" : 13.4, "luminosity" : 51 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91442d"), "date" : "2018-03-12 12:23:45.229", "temperature" : 29.6, "humidity" : 13.4, "luminosity" : 51 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df9149d6"), "date" : "2018-03-12 16:32:03.827", "temperature" : 29.6, "humidity" : 14.8, "luminosity" : 46 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914a0e"), "date" : "2018-03-12 16:40:46.764", "temperature" : 29.6, "humidity" : 14.8, "luminosity" : 46 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df914a0f"), "date" : "2018-03-12 16:40:57.025", "temperature" : 29.6, "humidity" : 14.8, "luminosity" : 46 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df916289"), "date" : "2018-03-13 10:30:48.354", "temperature" : 29.6, "humidity" : 19.2, "luminosity" : 63 }
{ "_id" : ObjectId("5b0ab1c7f4dbca05df91628a"), "date" : "2018-03-13 10:30:38.108", "temperature" : 29.6, "humidity" : 19.3, "luminosity" : 64 }
>
```

A5.9.8 MongoDB management

1.4 Query in Mongo shell

db.sensors.find({date: {$gt: “2018-05-26”}})

→ 5월 26일 이후 데이터 전부 추출



A5.9.8 MongoDB management

2. Import or export MongoDB (windows cmd 창에서 실행)

- **mongoimport -d dbName -c collectionName --type csv --headerline --file fileName.csv**

- **mongoexport -d dbName -c collectionName --fields <field1,field2,…> --limit=nn --type csv --out fileName.csv**

json 또는 csv 파일로 import/export

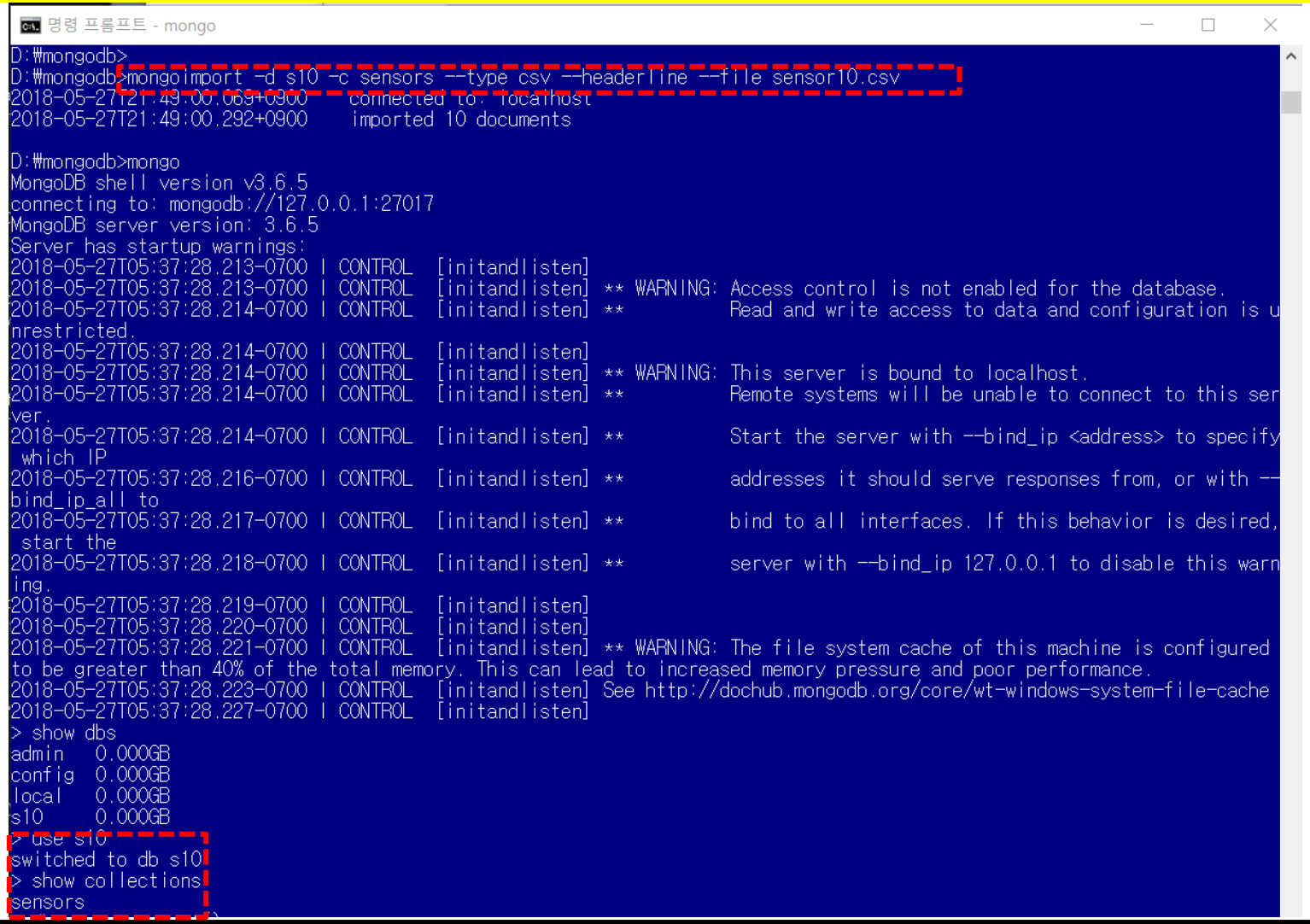
https://docs.mongodb.com/manual/reference/program/mongoimport/

https://docs.mongodb.com/manual/reference/program/mongoexport/

A5.9.8 MongoDB management

2.1.1 Import MongoDB (windows cmd 창에서 실행)

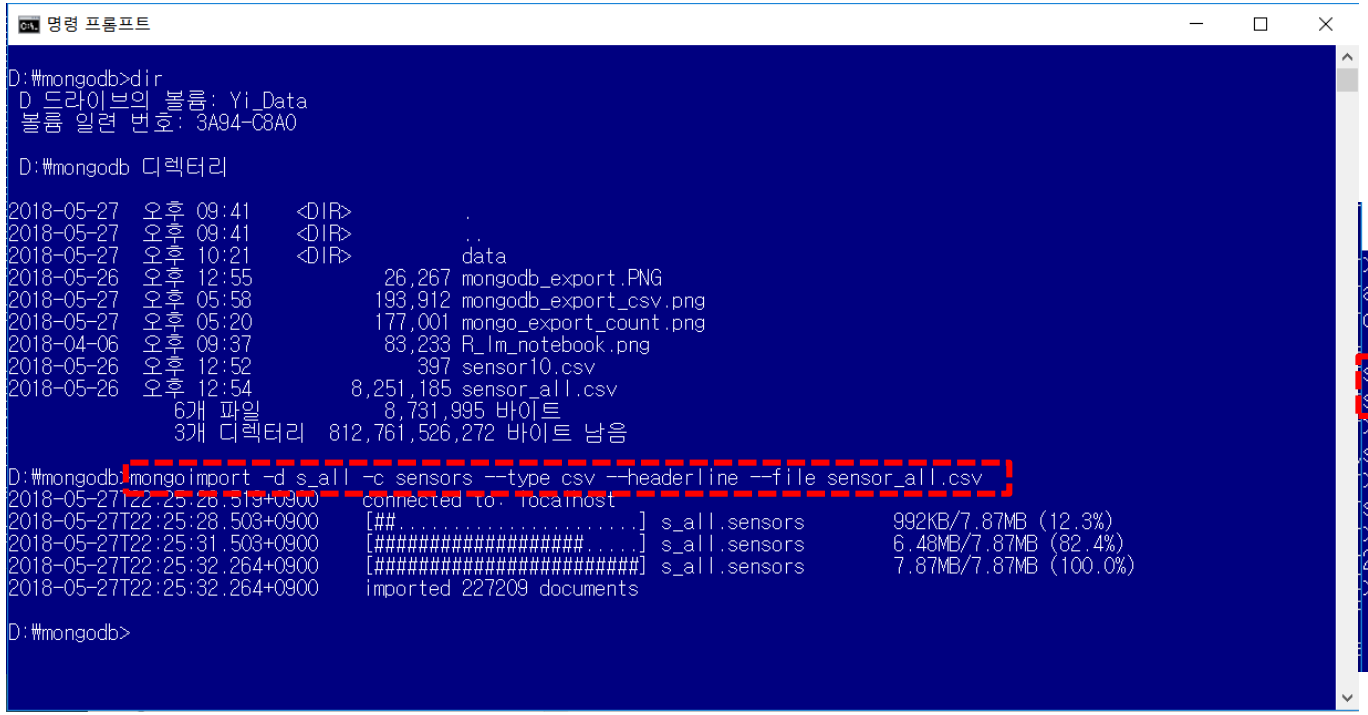
- mongoimport -d s10 -c sensors --type csv --headerline --file sensor10.csv



A5.9.8 MongoDB management

2.1.2 Import MongoDB (windows cmd 창에서 실행)

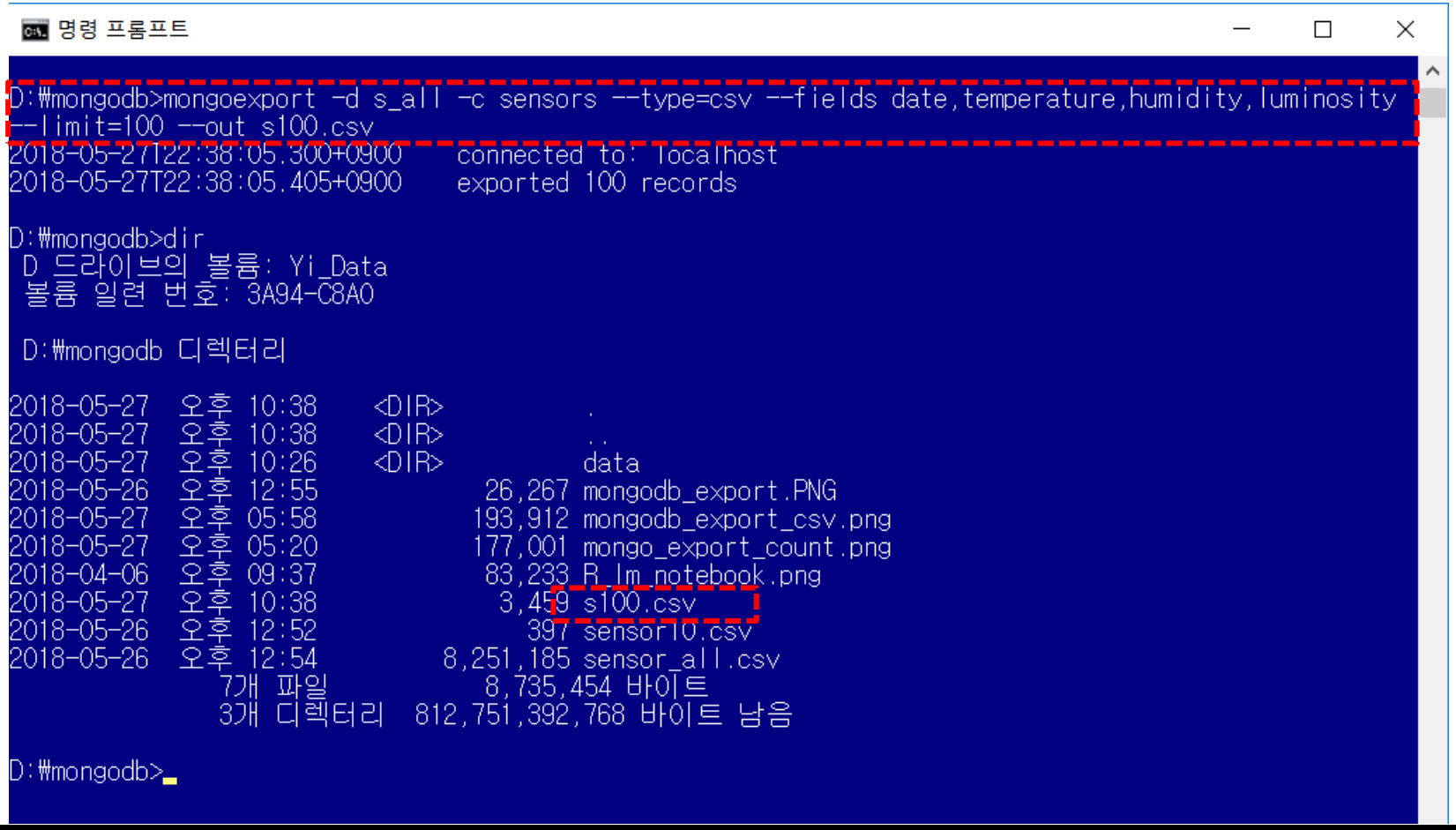
- **mongoimport -d s_all -c sensors --type csv –headerline --file sensor_all.csv**



A5.9.8 MongoDB management

2.2.1 Export MongoDB (windows cmd 창에서 실행)

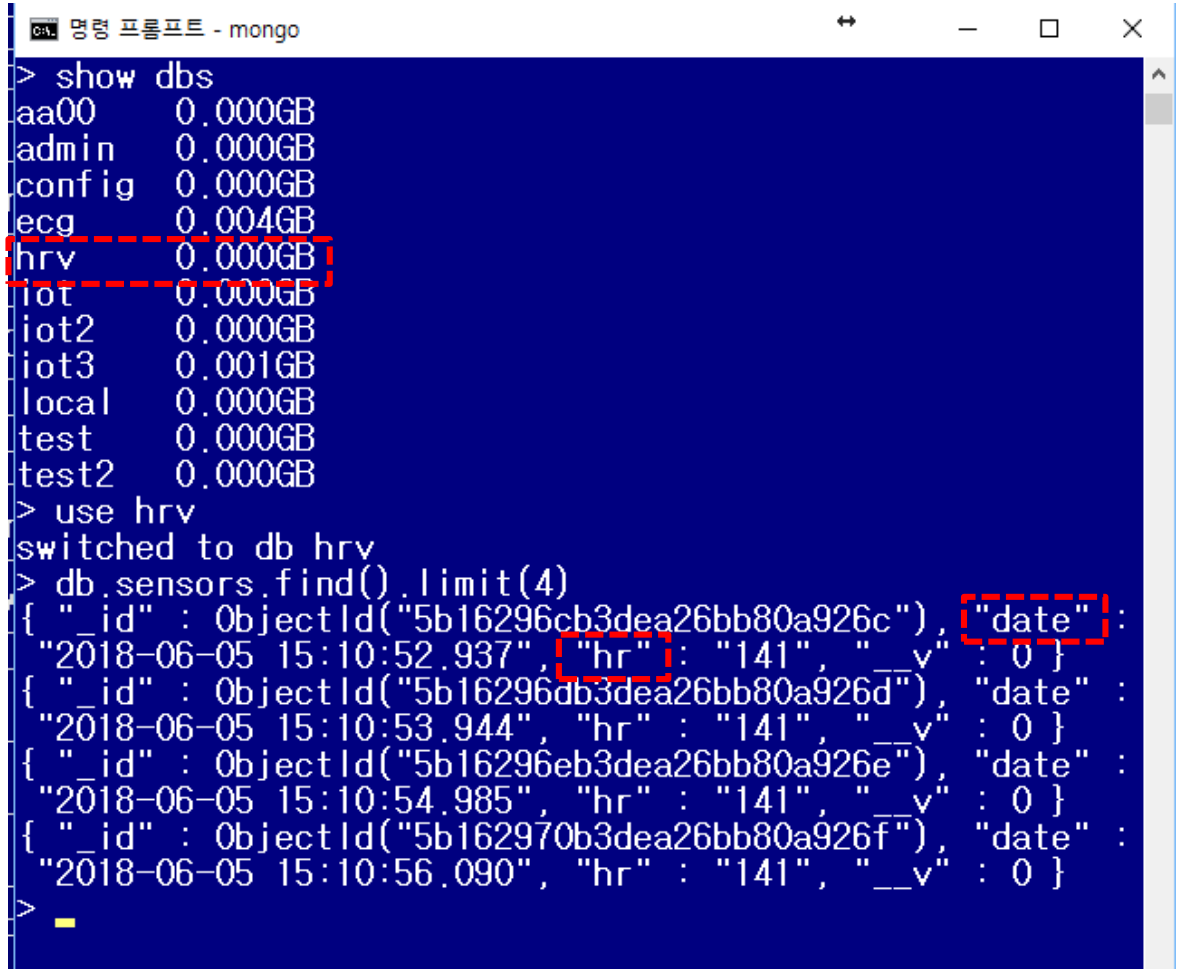
- **mongoexport -d s_all -c sensors --type=csv --fields date,temperature,humidity,luminosity --limit=100 --out s100.csv**



A5.9.8 MongoDB management

2.2.2 Export MongoDB (windows cmd 창에서 실행, dbName을 hrv로 변경!)

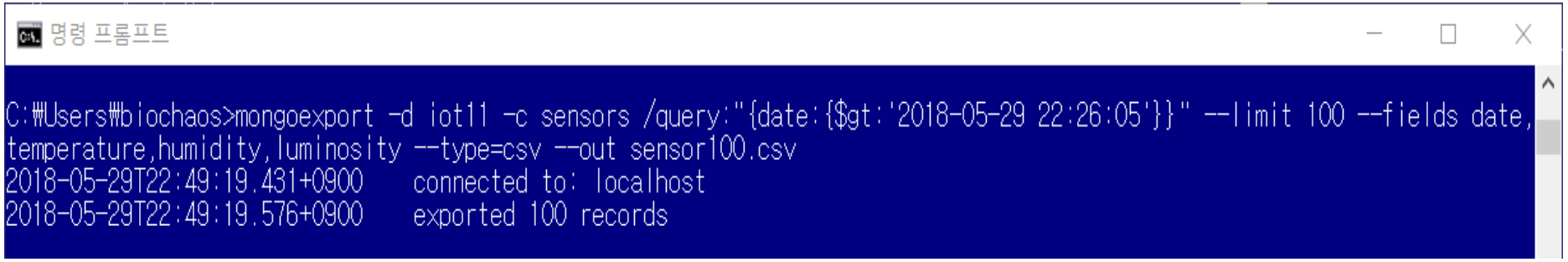
- mongoexport -d hrv -c sensors --type=csv --fields date,hr --out hs99hr.csv



A5.9.8 MongoDB management

2.3 Advanced export with query (windows cmd 창에서 실행)
iot11 db의 특정 시간 이후의 데이터 100개를 csv 파일 (s100.csv)로 저장

- **mongoexport -d iot11 -c sensors /query:"{date: {$gt: '2018-05-29 22:26:06'}}" --limit=100 --fields date,temperature,humidity,luminosity --type=csv --out s100.csv**



[Tip] iot db의 최근 데이터 500개를 csv 파일 (s500.csv)로 저장할 때,

- **mongoexport –d iot -c sensors --sort "{_id: -1}" --limit=500 --fields date,temperature,humidity,luminosity --type=csv --out s500.csv**