

<파이썬 데이터 분석 및 MongoDB 평가>

1. CSV 파일 BigData 수집& 저장

특정 폴더에 들어있는 CSV 파일 모두 합쳐 하나의 CSV 파일로 생성하는 프로그램을 작성하시오.
(사용한 CSV 파일과 생성된 CSV 파일, 프로그램 소스 제출)

```
import csv
import glob
import os

## 전역 변수부
file_list = glob.glob(os.path.join('C:/Users/admin/Desktop/평가/CSV/folder/', '*.csv'))
firstYN = True

## 메인 코드부
for input_file in file_list :
    with open(input_file, "r") as inFp :
        with open("C:/Users/admin/Desktop/평가/CSV/강원인구통합.csv", "a", newline='')
as outFp:

    csvReader = csv.reader(inFp)
    csvWriter = csv.writer(outFp)
    header_list = next(csvReader)

    if firstYN == True :
        csvWriter.writerow(header_list)
        firstYN = False

    for row_list in csvReader:
        csvWriter.writerow(row_list)

print('Save. OK~')
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\admin\Desktop\평가\1번 (Ex06-01).py
Save. OK~
>>>
```

강원인구통합.csv [C:\Users\admin\Desktop\평가\CSVW] - 혼셀

파일 편집 보기 입력 서식 수식 데이터 도구

오려두기 복사하기 붙여넣기 셀 내용 맞추기 모양 복사 셀 서식 일반 % , .00 .00 줄 병합 테두리 셀 편집 숨기기 채우기 자동

앞은 고정 11.0 pt 가 가 가 각 가 가

	A	B	C	D	E	F	G	H	I	J	K	L
시점	세대수	인구수	남자인구	여자인구	한국인인구	한국인남자	한국인여자	외국인인구	외국인남자	외국인여자		
1961	296732	1585869	791267	794602								
1962	295754	1593037	808319	784718	1593037							
1963	305792	1678126	852652	825474	1678126	852652	825474	0				
1964	308605	1722005	878558	843447	1722005	878558	843447	0				
1965	312688	1771035	902167	868868	1771035	902167	868868	0	0	0		
1966	334196	1832383	936824	895559	1832383	936824	895559	0	0	0		
1967	326508	1825018	930319	894699	1825018	930319	894699	0	0	0		
1968	325301	1823612	928804	894808	1823612	928804	894808	0	0	0		
1969	328938	1810527	924250	886277	1810527	924250	886277	0	0	0		
1970	352484	1866928	953653	913275	1866928	953653	913275	0	0	0		
1971	343835	1851866	944748	907118	1851866	944748	907118	0	0	0		
1972	341985	1863419	950793	912626	1863419	950793	912626	0	0	0		
1973	339751	1852456	946034	906422	1852456	946034	906422	0	0	0		
1974	338744	1865738	953938	911800	1865738	953938	911800	0	0	0		
1975	362165	1862107	955258	906849	1862107	955258	906849	0	0	0		
1976	355710	1842363	941292	901071	1842363	941292	901071	0	0	0		
1977	355909	1847177	942097	905080	1847177	942097	905080	0	0	0		
1978	362478	1852755	942117	910638	1852755	942117	910638	0	0	0		
1979	369774	1844559	939862	904697	1844559	939862	904697	0	0	0		
1980	375265	1790954	915397	875557	1790226	914983	875243	728	414	314		
1981	376332	1804853	921268	883585	1804853	921268	883585	0	0	0		
1982	381965	1809871	925008	884863	1809871	925008	884863	0	0	0		
1983	391296	1824324	931933	892391	1823717	931611	892106	607	322	285		
1984	393853	1816365	927598	888767	1815719	927253	888466	646	345	301		
1985	404694	1724809	880464	844345	1724146	880090	844056	663	374	289		
1986	406934	1750707	893424	857283	1750119	893107	857012	588	317	271		
1987	407745	1751252	891588	859664	1750698	891280	859418	554	308	246		
1988	412209	1730905	879530	851375	1730307	879205	851102	598	325	273		
1989	413688	1702025	864011	838014	1701502	863724	837778	523	287	236		

강원인구통합

준비 NUM 100%

2. (Excel 파일 BigData 수집& 저장)

특정 폴더에 들어있는 Excel 파일을 모두 합쳐 하나의 CSV 파일로 생성하는 프로그램을

작성하시오.(사용한 Excle 파일과 생성된 CSV 파일, 프로그램 소스를 제출)

```
import csv
import glob
import os
import openpyxl

## 전역 변수부
file_list = glob.glob(os.path.join('C:/Users/admin/Desktop/평가/Excel/folder/', '*.xlsx'))
firstYN = True
## 메인 코드부
for input_file in file_list :
    workbook = openpyxl.load_workbook(input_file)
    wsheetList = workbook.sheetnames
    for wsName in wsheetList:
        worksheet = workbook[wsName]
        with open("C:/Users/admin/Desktop/평가/Excel/강원인구통합.csv", "a", newline='') as
outFp:
            csvWriter = csv.writer(outFp)
            if firstYN == True:
                header_list = []
                for col in range(1, worksheet.max_column + 1):
                    header_list.append(worksheet.cell(row=1, column=col).value)
                csvWriter.writerow(header_list)
                firstYN = False
            for row in range(2, worksheet.max_row + 1):
                row_list = []
                for col in range(1, worksheet.max_column + 1):
                    row_list.append(worksheet.cell(row=row, column=col).value)
                csvWriter.writerow(row_list)

print('Save. OK~')
```

```

Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\admin\Desktop\평가\2번 (Ex07-01).py
Save. OK~
>>>

```

강원인구통합.csv [C:\Users\Wadmin\Desktop\평가\Excel\W] - 홈셀

파일 편집 보기 입력 서식 수식 데이터 도구

오려 두기 복사하기 붙여넣기 셀 내용 맞추기 모양 복사 셀 서식

일반 % , .00 .00

출 병합 테두리 셀 편집 숨기기 채우기

맑은 고딕 11.0 pt

시점	세대수	인구수	남자인구	여자인구	한국인인구	한국인남자	한국인여자	외국인인구	외국인남자	외국인여자
1961	296732	1585869	791267	794602						
1962	295754	1593037	808319	784718	1593037					
1963	305792	1678126	852652	825474	1678126	852652	825474	0		
1964	308605	1722005	878558	843447	1722005	878558	843447	0		
1965	312688	1771035	902167	868868	1771035	902167	868868	0	0	0
1966	334196	1832383	936824	895559	1832383	936824	895559	0	0	0
1967	326508	1825018	930319	894699	1825018	930319	894699	0	0	0
1968	325301	1823612	928804	894808	1823612	928804	894808	0	0	0
1969	328938	1810527	924250	886277	1810527	924250	886277	0	0	0
1970	352484	1866928	953653	913275	1866928	953653	913275	0	0	0
1971	343835	1851866	944748	907118	1851866	944748	907118	0	0	0
1972	341985	1863419	950793	912626	1863419	950793	912626	0	0	0
1973	339751	1852456	946034	906422	1852456	946034	906422	0	0	0
1974	338744	1865738	953938	911800	1865738	953938	911800	0	0	0
1975	362165	1862107	955258	906849	1862107	955258	906849	0	0	0
1976	355710	1842363	941292	901071	1842363	941292	901071	0	0	0
1977	355909	1847177	942097	905080	1847177	942097	905080	0	0	0
1978	362478	1852755	942117	910638	1852755	942117	910638	0	0	0
1979	369774	1844559	939862	904697	1844559	939862	904697	0	0	0
1980	375265	1790954	915397	875557	1790226	914983	875243	728	414	314
1981	376332	1804853	921268	883585	1804853	921268	883585	0	0	0
1982	381965	1809871	925008	884863	1809871	925008	884863	0	0	0
1983	391296	1824324	931933	892391	1823717	931611	892106	607	322	285
1984	393853	1816365	927598	888767	1815719	927253	888466	646	345	301
1985	404694	1724809	880464	844345	1724146	880090	844056	663	374	289
1986	406934	1750707	893424	857283	1750119	893107	857012	588	317	271
1987	407745	1751252	891588	859664	1750698	891280	859418	554	308	246
1988	412209	1730905	879530	851375	1730307	879205	851102	598	325	273
1989	413688	1702025	864011	838014	1701502	863724	837778	523	287	236

강원인구통합

준비 NUM

3.(Crawling BigData 수집& 저장)

네이트 뉴스 속보를 크롤링하여 기사 제목/뉴스 링크/언론사/출간일시에 따라 2분 간격으로 반복해서 출력하시오.(출력한 화면을 캡처하여 파일로 저장하고 프로그램 소스와 함께 제출)

```
import time
import bs4
import urllib.request

# 파일에 저장할 데이터를 담은 문자열 초기화
data_to_save = ""

nateUrl = "https://news.nate.com/recent?mid=n0100"
while True:
    htmlObject = urllib.request.urlopen(nateUrl)
    webPage = htmlObject.read()
    bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
    tag_list = bsObject.findAll('div', {'class': 'mlt01'})

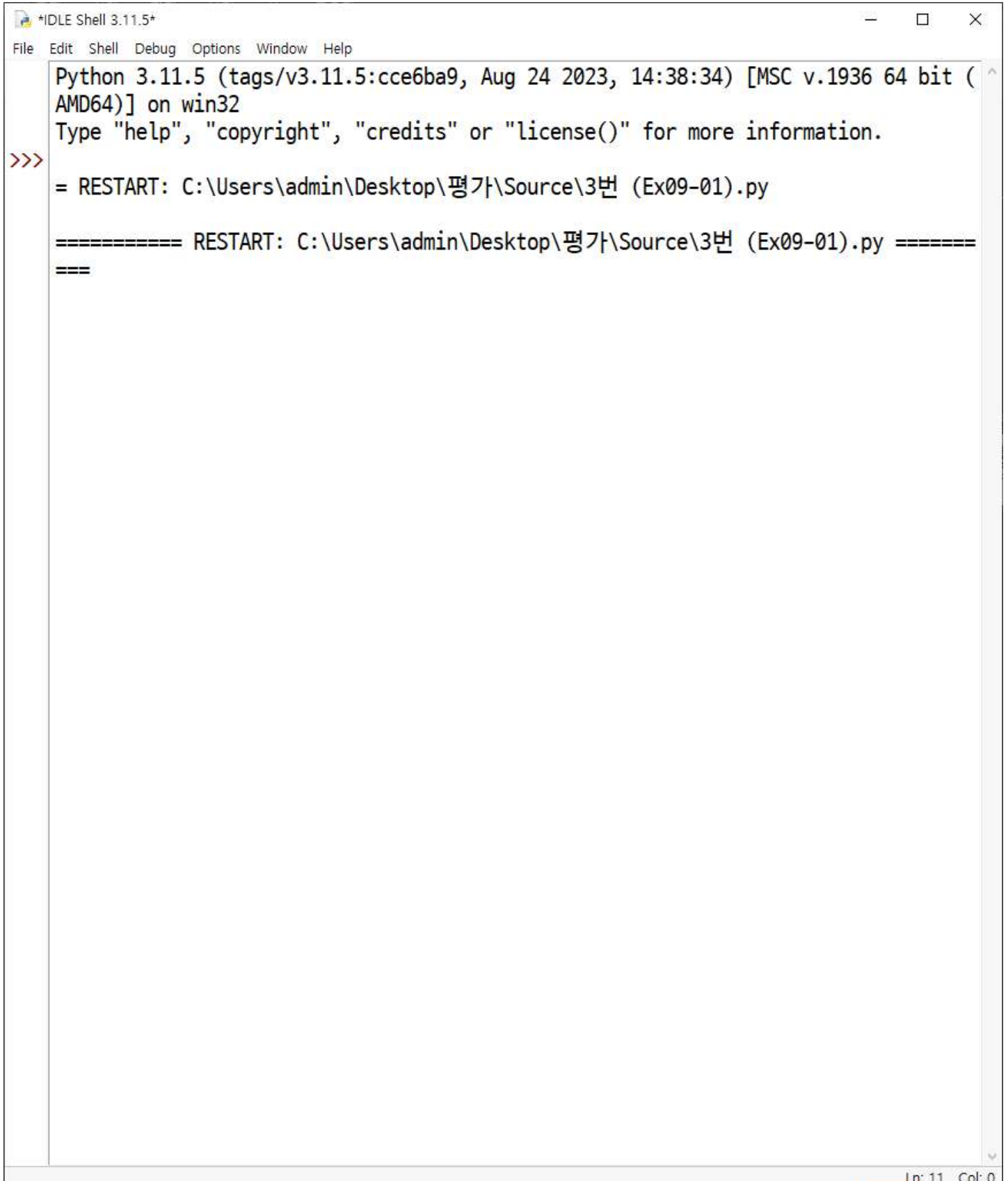
    data_to_save += '##### 실시간 뉴스 속보 #####\n'
    num = 1
    for tag in tag_list:
        subject = tag.find('h2', {'class': 'tit'}).text
        link = tag.find('a', {'class': 'lt1'})['href']
        pressAndDate = tag.find('span', {'class': 'medium'}).text
        pressAndDate.replace('\t', ' ')
        pressAndDate.replace('\n', '')

        if len(pressAndDate.split()) == 3:
            press, pDate, pTime = pressAndDate.split()
        elif len(pressAndDate.split()) == 4:
            press1, press2, pDate, pTime = pressAndDate.split()
            press = press1 + press2
        else:
            continue

        data_to_save += '(' + str(num) + ')' + subject + '\n'
        data_to_save += '\t https:' + link + ' ' + press + ' ' + pDate + ' ' + pTime + '\n'
        num += 1

# 데이터를 파일에 저장
with open("nate_news.txt", "w", encoding="utf-8") as file:
    file.write(data_to_save)
```

time.sleep(120)



The screenshot shows a Python IDLE Shell window titled '*IDLE Shell 3.11.5*'. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the following content:

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\admin\Desktop\평가\Source\3번 (Ex09-01).py
===== RESTART: C:\Users\admin\Desktop\평가\Source\3번 (Ex09-01).py =====
===
```

The status bar at the bottom right indicates 'Ln: 11 Col: 0'.

4. (MongoDB 생성, 검색, 수정, 삭제)

MongoDB를 설치한 후 간단한 게시판의 database, collection, document를 만들고 생성, 검색, 수정, 삭제하는 Query문을 작성하여 제출하시오. MongoDB Compass를 이용하여 만든 MongoDB를 볼 수 있도록 화면을 아래와 같이 캡처하여 함께 제출하시오.

<소스 코드>

[도큐먼트 작성 - 컬렉션 생성]

```
use board
```

```
switched to db board
```

[자유게시판 - 도큐먼트 board에 컬렉션 생성]

```
freeboard_result = db.board.insertOne({name: "자유게시판"})
```

```
{
  acknowledged: true,
  insertedId: ObjectId("650bda605fdb941e926cc241")
}
```

```
freeboard_id = freeboard_result.insertedId
```

```
ObjectId("650bda605fdb941e926cc241")
```

[자유게시판 글 작성]

```
db.article.insertMany([
```

```
  {
    board_id: "freeboard_id",
    title: 'hello',
    content: 'hi, hello1',
    author: 'Karoid'
```

```
},
```

```
{
  board_id: "freeboard_id",
  title: 'hi',
  content: 'hi, hello2',
  author: 'Jeong'
```

```
},
```

```
{
  board_id: "freeboard_id",
  title: 'hi',
  content: 'hi, hello3',
  author: 'Hong',
  comments: [
```

```
    {
      author: 'karoid',
      content: 'hello Hong!'
    }
```

```
]
```

```
}]
```

```
)
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("650bda765fdb941e926cc242"),
    '1': ObjectId("650bda765fdb941e926cc243"),
    '2': ObjectId("650bda765fdb941e926cc244")
  }
}
```

[비밀게시판 작성]

```
secretboard_id = db.board.insertOne({name: '비밀게시판'}).insertedId
ObjectId("650bda7e5fdb941e926cc245")
```

[비밀게시판 글 작성]

```
db.article.insertOne({
  board_id: "secretboard_id",
  title: 'my Secret Title',
  content: 'hi, hello1',
  author: 'noname'
})
{
  acknowledged: true,
  insertedId: ObjectId("650bda965fdb941e926cc246")
}
```

[자유게시판 문서 글 불러오기]

```
freeboard_id = db.board.find({name: '자유게시판'}).toArray()[0]._id
ObjectId("650bda605fdb941e926cc241")
```

[자유게시판 글만 나타내게 하기]

```
db.article.find({board_id: "freeboard_id"},{
  _id: false,
  board_id: false,
  author: false,
  comments: false})
{
  title: 'hello',
  content: 'hi, hello1'
}
```



```
{
  title: 'hi',
  content: 'hi, hello2'
}
{
  title: 'hi',
  content: 'hi, hello3'
}
```

[Karoid라는 특정 도큐먼트 찾기]

```
db.article.find({"comments.0.author": "Karoid"})
```

[모든글에 대한 추천수 필드 추가]

```
db.article.updateMany({},{$set: {upvote: 0}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
```

[비밀게시판 _id를 찾아서 게시판 추천수 모두 +1 증가하기]

```
secretboard_id = db.board.find({name: "비밀게시판"}).toArray()[0]._id
ObjectId("650bda7e5fdb941e926cc245")
```

```
db.article.updateMany({board_id: secretboard_id}, {$inc: {upvote: 1}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[댓글이 달린 자유게시판 글의 _id(작성자 Hong) 저장(ObjectId 값 저장)]

```
freeboard_id = db.board.find({name: "자유게시판"}).toArray()[0]._id
ObjectId("650bda605fdb941e926cc241")
```

```
doc_id = db.article.find({board_id: "freeboard_id", author: "Hong"}).toArray()[0]._id
ObjectId("650bda765fdb941e926cc244")
```

[Content 값 수정]

```
db.articel.updateOne({_id: 'doc_id'}, {$set: {content: 'updated'}})
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[필드 안에 있는 요소 추가]

```
db.article.updateOne(
  { _id: 'doc_id' }, {
    $push: {comments: {author: "Quote", content: "reply"}}
  })
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[Comment 필드 안에 있는 요소 추가]

```
db.article.updateOne(
  { _id: 'doc_id' }, {
    $set: { "comments.${karoidcomment}.upvote": 0 }
  },
  {
    arrayFilters: [{ "karoidcomment.author": "Quote" }]
  }
)
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

[board의 article 컬렉션 삭제]

```
db.article.deleteMany({})
{
  acknowledged: true,
  deletedCount: 4
}
```

[board의 article 도큐먼트 삭제]

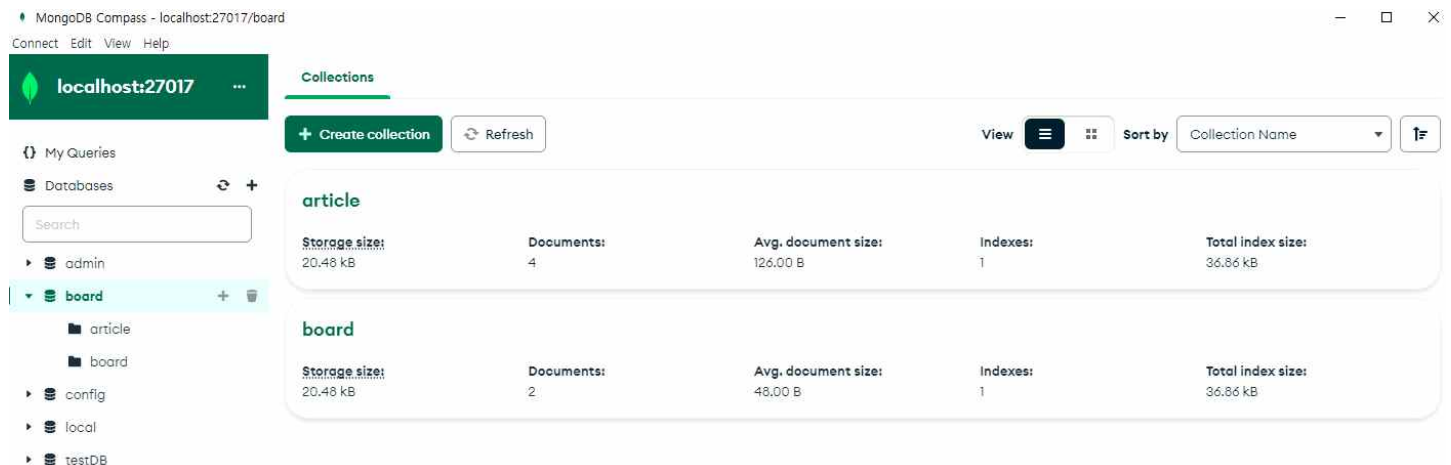
```
db.article.drop()
true
```

[board 도큐먼트 삭제]

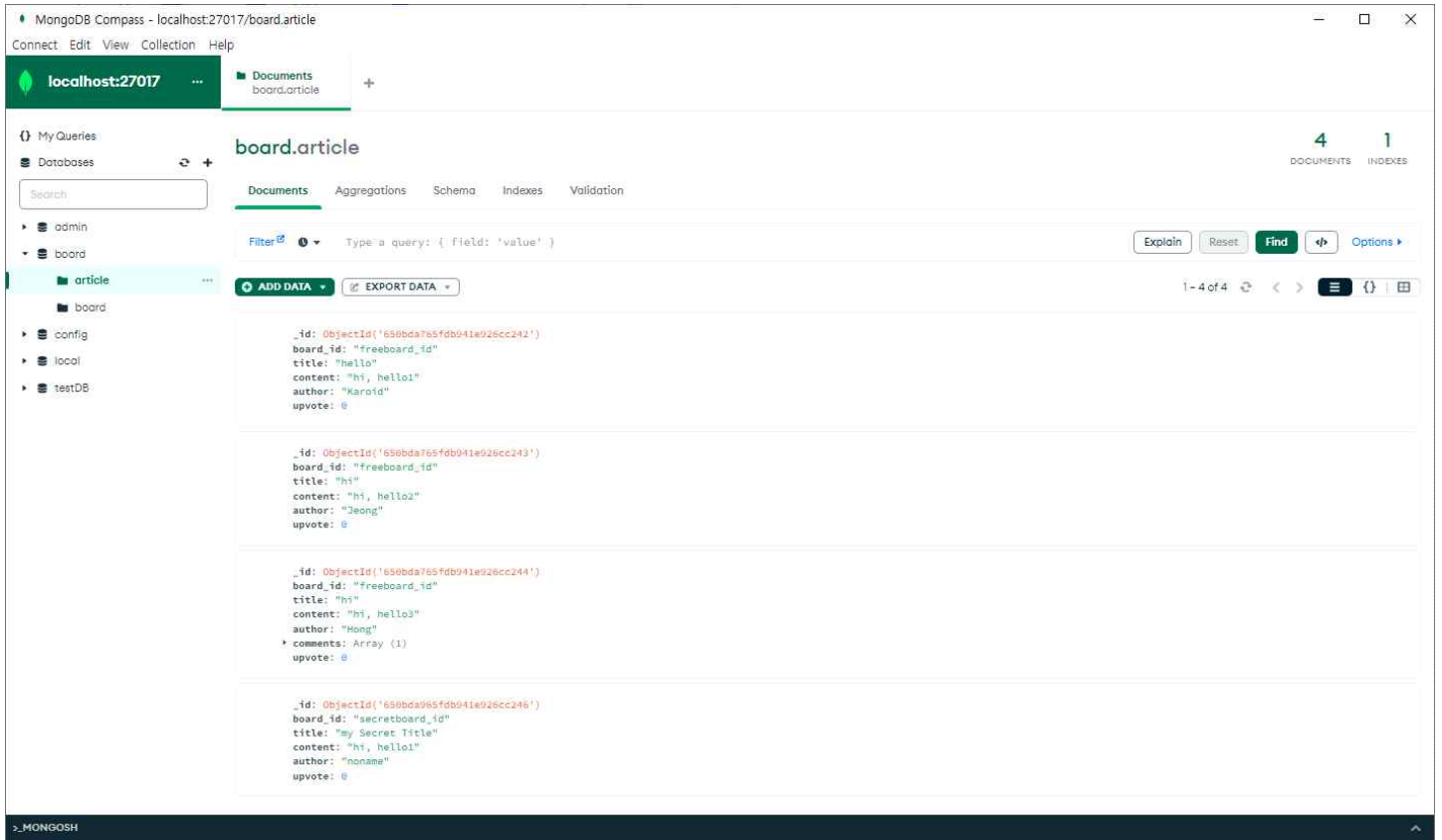
```
db.board.drop()
true
```

[전체 도큐먼트 삭제]

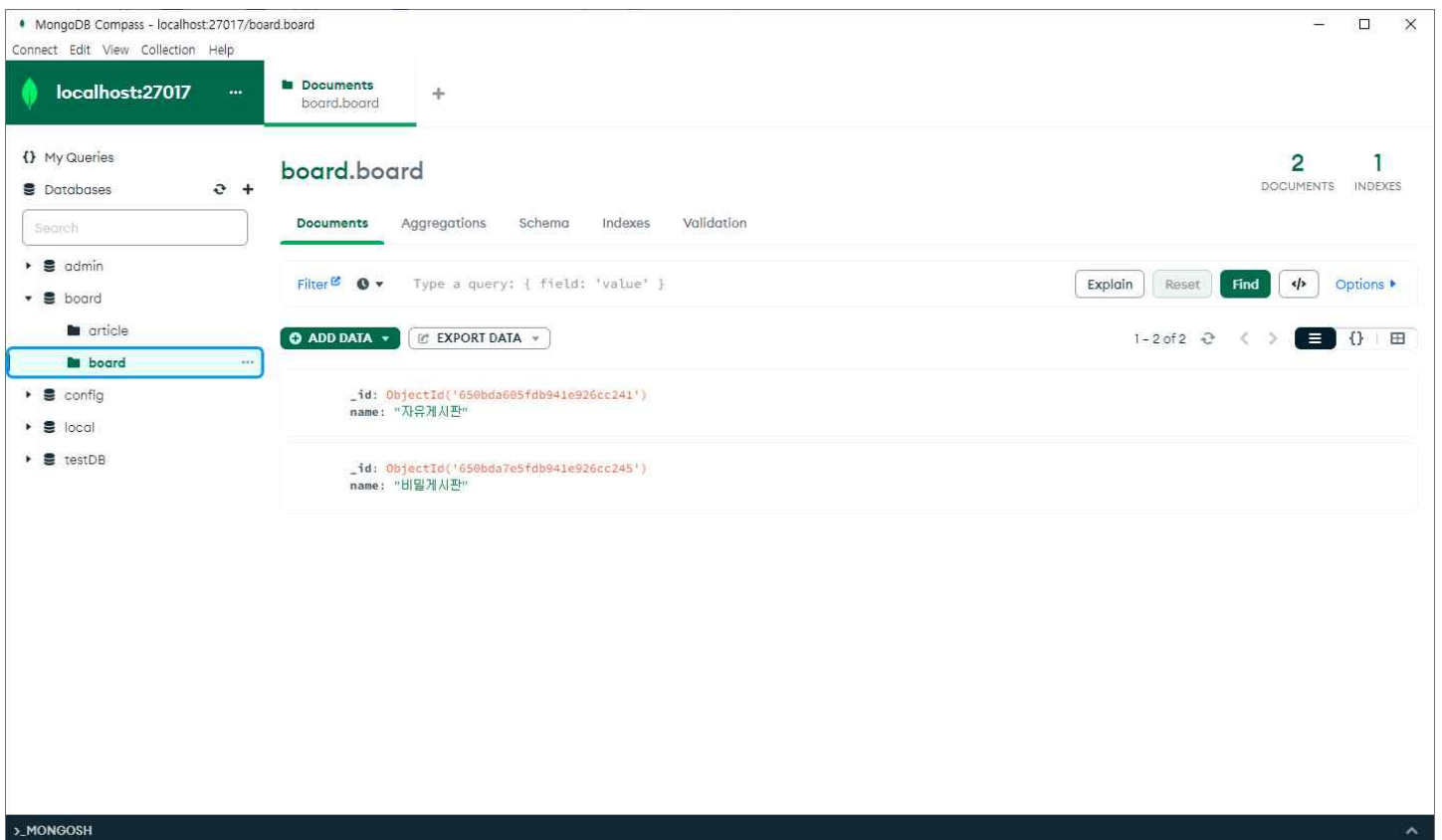
```
db.dropDatabase()
{ ok: 1, dropped: 'board' }
```



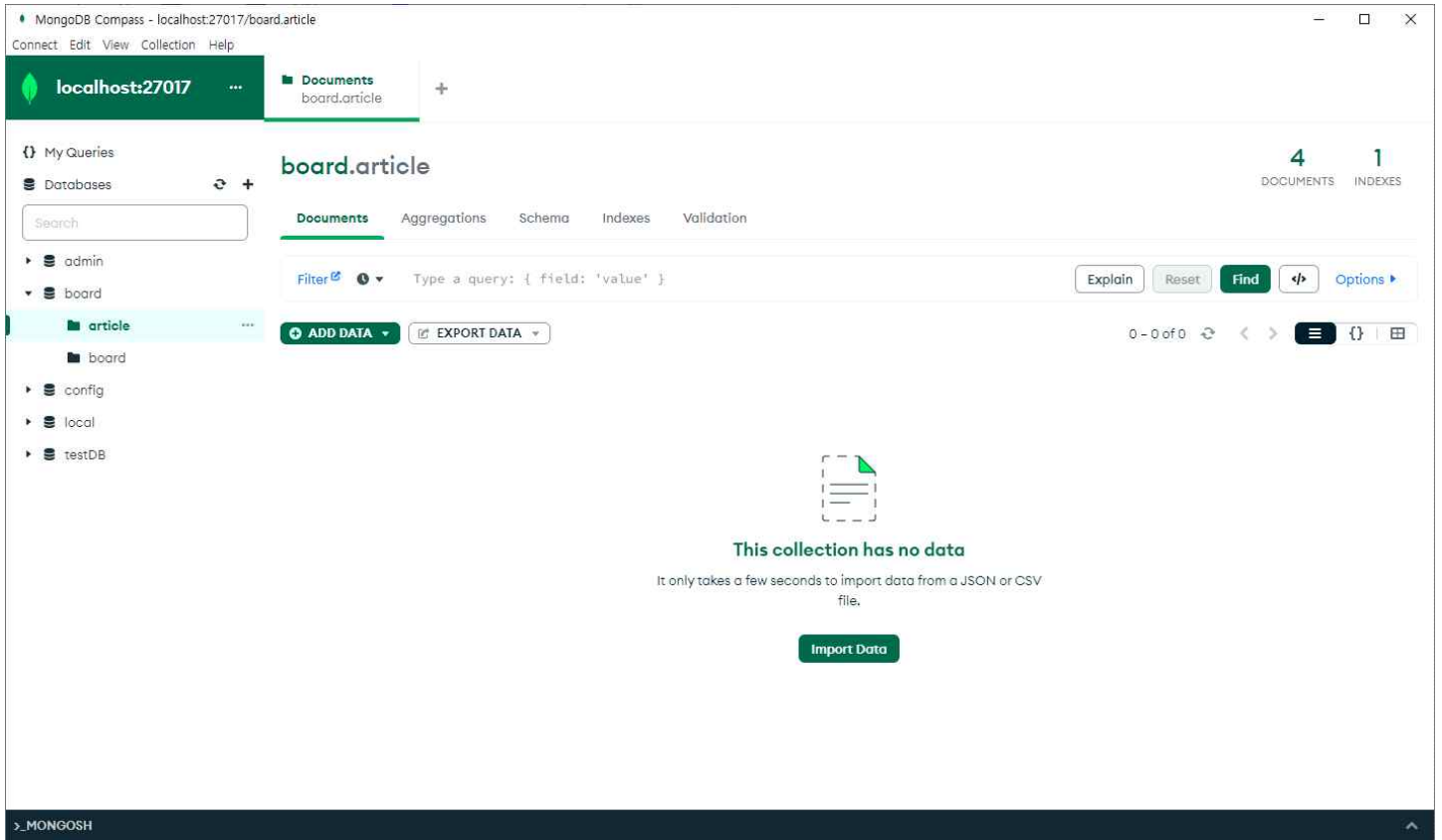
board Document & Collection 생성



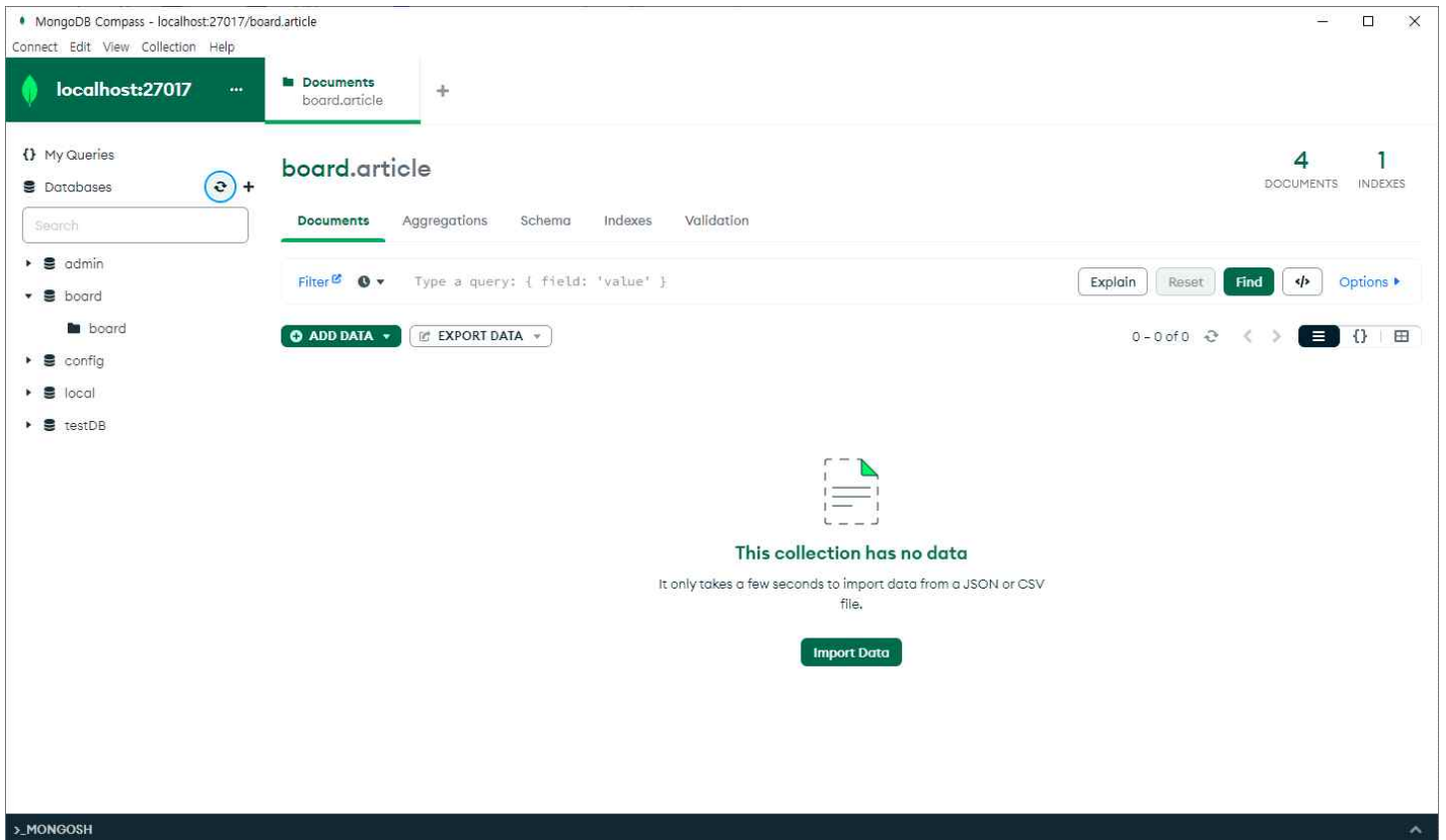
board.article의 Collection



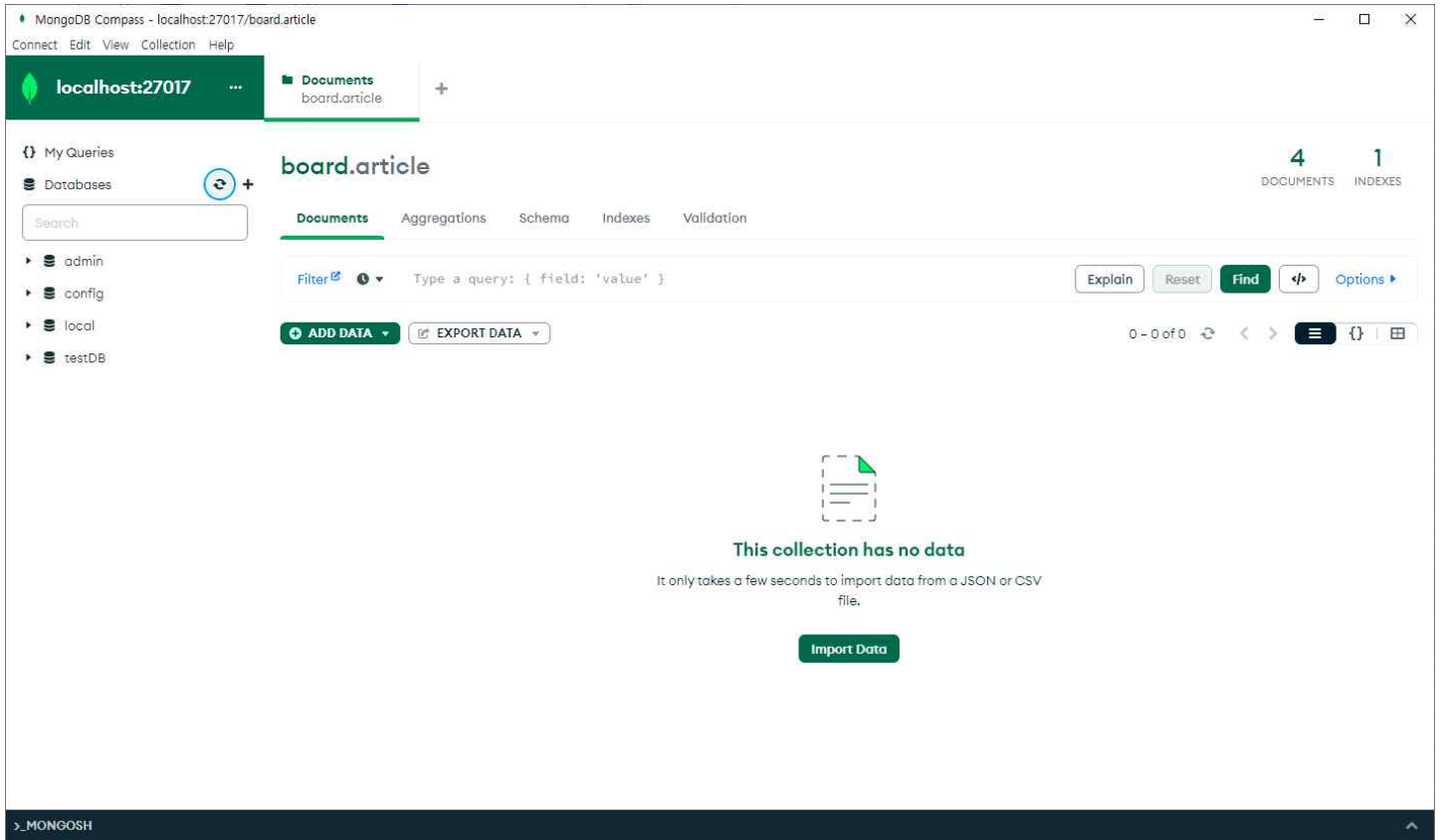
board.board Collection



board.article Collection 삭제 후



board.article 도큐먼트 삭제



board DB 전체 삭제 후