DB프로그래밍 과제03 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[과제 실습] MongoDB 를 실행하고 다음과 같이 실습해 보시기 바랍니다.

1. employees Collection 생성

=> [답안]

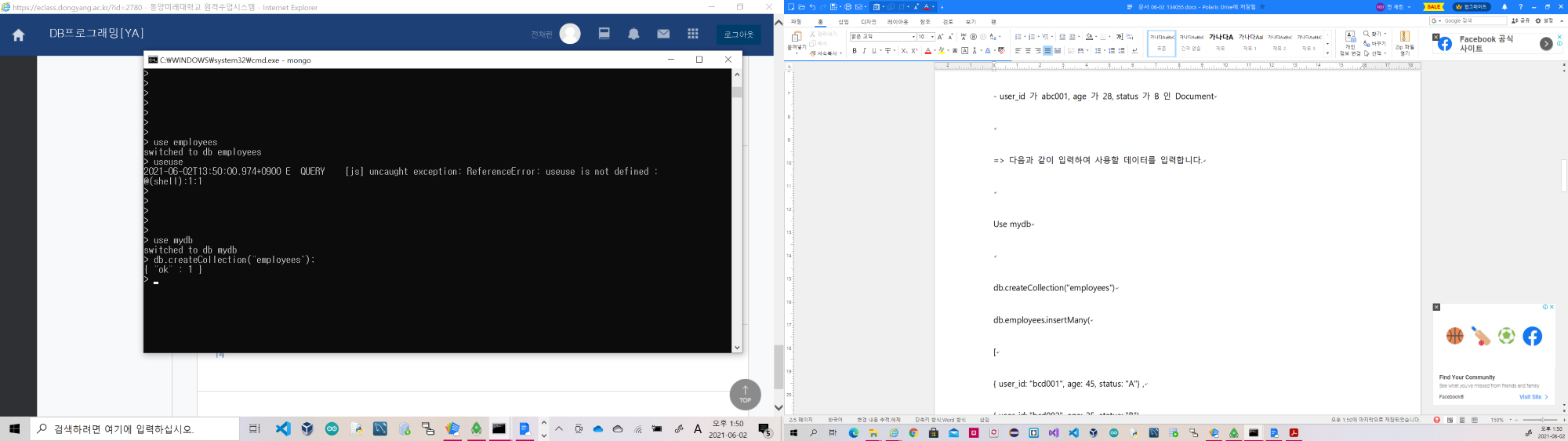
[소스코드]

use mydb

db.createCollection("employees");

[실행결과]

{ "ok" : 1 }



2. 다음 Document 데이터 넣기

- user\_id 가 bcd001, age 가 45, status 가 A 인 Document

- user\_id 가 bcd002, age 가 25, status 가 B 인 Document

- user\_id 가 bcd003, age 가 50, status 가 A 인 Document

- user\_id 가 bcd004, age 가 35, status 가 A 인 Document

- user\_id 가 abc001, age 가 28, status 가 B 인 Document

=> [답안]

​[소스코드]

db.employees.insertMany(

... [

... { user\_id: "bcd001", age: 45, status: "A" } ,

... { user\_id: "bcd002", age: 25, status: "B" } ,

... { user\_id: "bcd003", age: 50, status: "A" } ,

... { user\_id: "bcd004", age: 35, status: "A" } ,

... { user\_id: "abc001", age: 28, status: "B" }

... ]

... );

[실행결과]

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("60b70f370fc067b2b9026aac"),

ObjectId("60b70f370fc067b2b9026aad"),

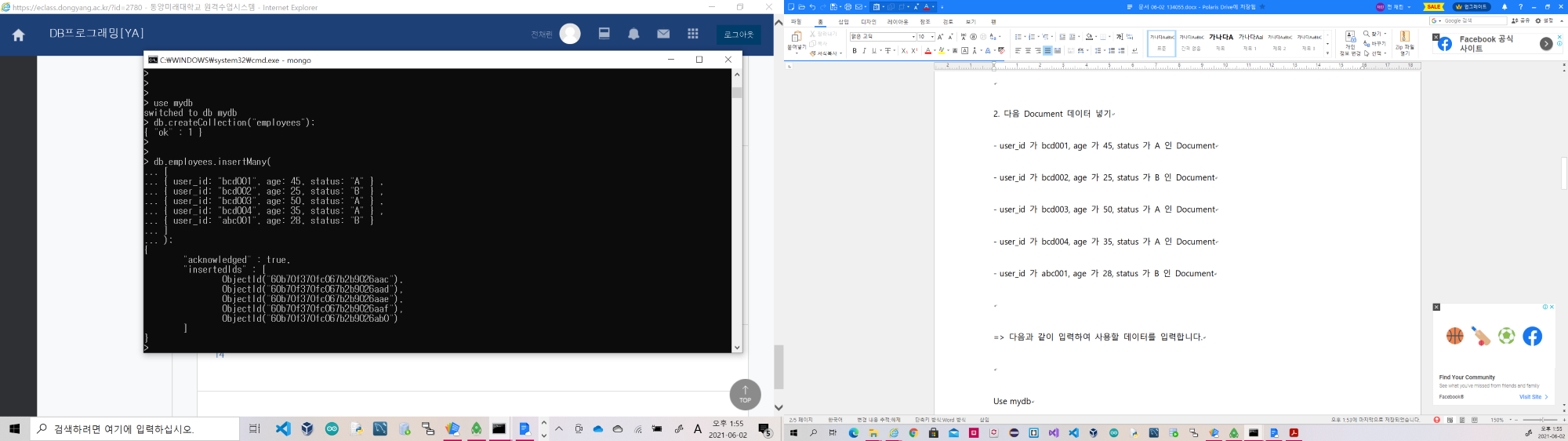
ObjectId("60b70f370fc067b2b9026aae"),

ObjectId("60b70f370fc067b2b9026aaf"),

ObjectId("60b70f370fc067b2b9026ab0")

]

}



[과제03-01]

1. 다음 Document 데이터 조회

- age 가 30 보다 큰 Document 의 user\_id, age, \_id 출력

=> [답안]

[소스코드]

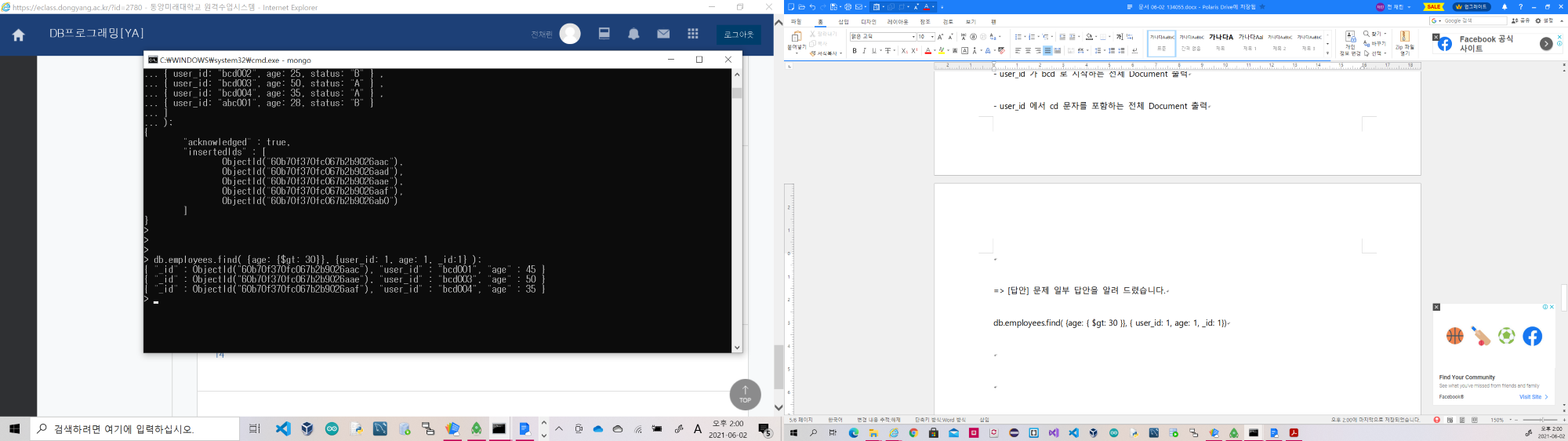
db.employees.find( {age: {$gt: 30}}, {user\_id: 1, age: 1, \_id:1} );

[실행결과]

{ "\_id" : ObjectId("60b70f370fc067b2b9026aac"), "user\_id" : "bcd001", "age" : 45 }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aae"), "user\_id" : "bcd003", "age" : 50 }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aaf"), "user\_id" : "bcd004", "age" : 35 }



- age 가 50 이고 status 가 A 인 Document 의 user\_id, age 출력

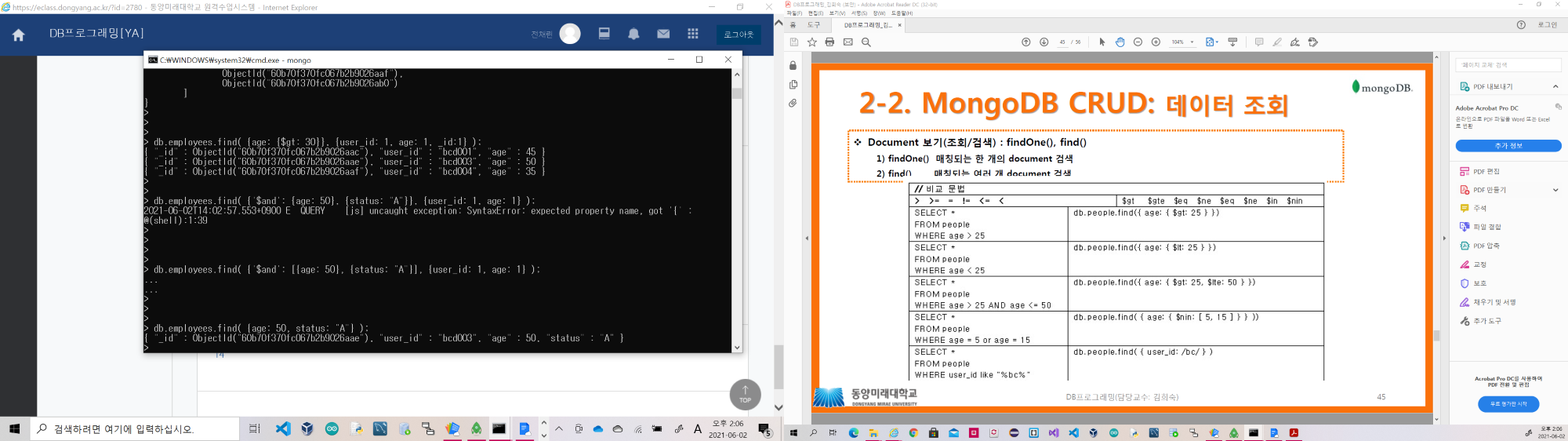
=> [답안]

[소스코드]

db.employees.find( {age: 50, status: "A"} );

[실행결과]

{ "\_id" : ObjectId("60b70f370fc067b2b9026aae"), "user\_id" : "bcd003", "age" : 50, "status" : "A" }



- age 가 30 보다 작은 Document 의 user\_id 와 age 출력

=> [답안]

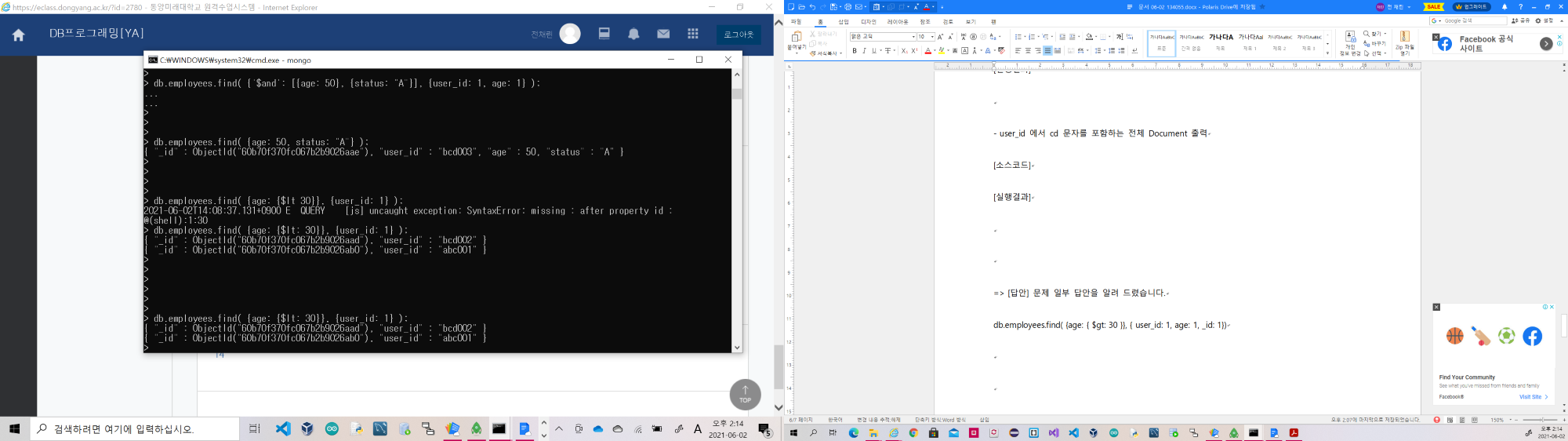
[소스코드]

db.employees.find( {age: {$lt: 30}}, {user\_id: 1} );

[실행결과]

{ "\_id" : ObjectId("60b70f370fc067b2b9026aad"), "user\_id" : "bcd002" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026ab0"), "user\_id" : "abc001" }



​

- user\_id 종류 출력

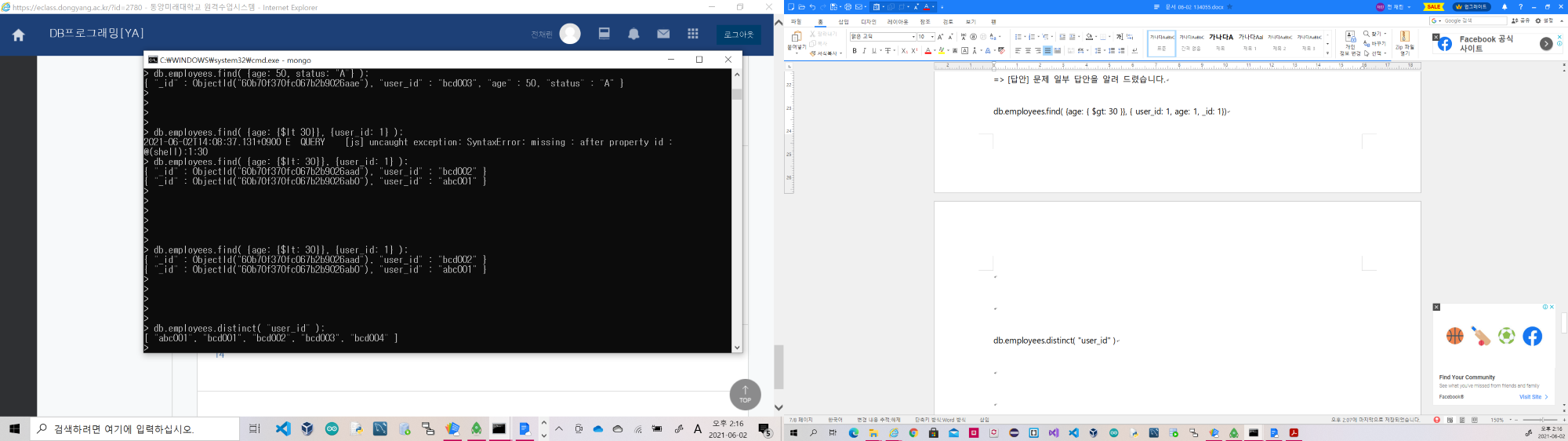
=> [답안]

[소스코드]

db.employees.distinct( "user\_id" );

[실행결과]

[ "abc001", "bcd001", "bcd002", "bcd003", "bcd004" ]



- user\_id 가 bcd 로 시작하는 전체 Document 출력

=> [답안]

[소스코드]

db.employees.find( {user\_id: /bcd/} );

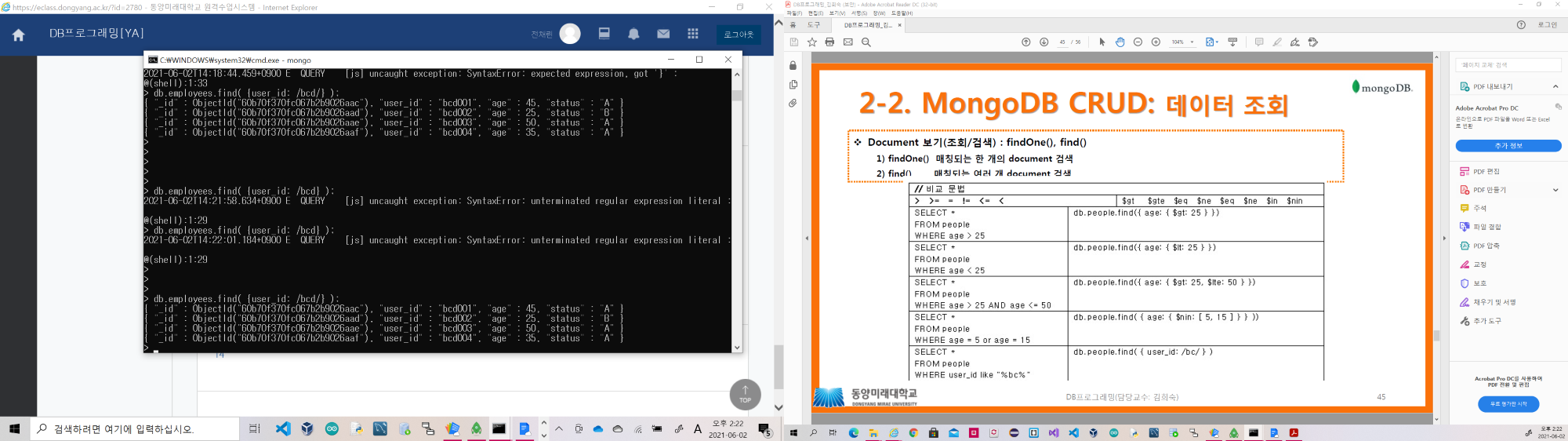
[실행결과]

{ "\_id" : ObjectId("60b70f370fc067b2b9026aac"), "user\_id" : "bcd001", "age" : 45, "status" : "A" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aad"), "user\_id" : "bcd002", "age" : 25, "status" : "B" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aae"), "user\_id" : "bcd003", "age" : 50, "status" : "A" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aaf"), "user\_id" : "bcd004", "age" : 35, "status" : "A" }



- user\_id 에서 cd 문자를 포함하는 전체 Document 출력

=> [답안]

[소스코드]

db.employees.find( {user\_id: /cd/} );

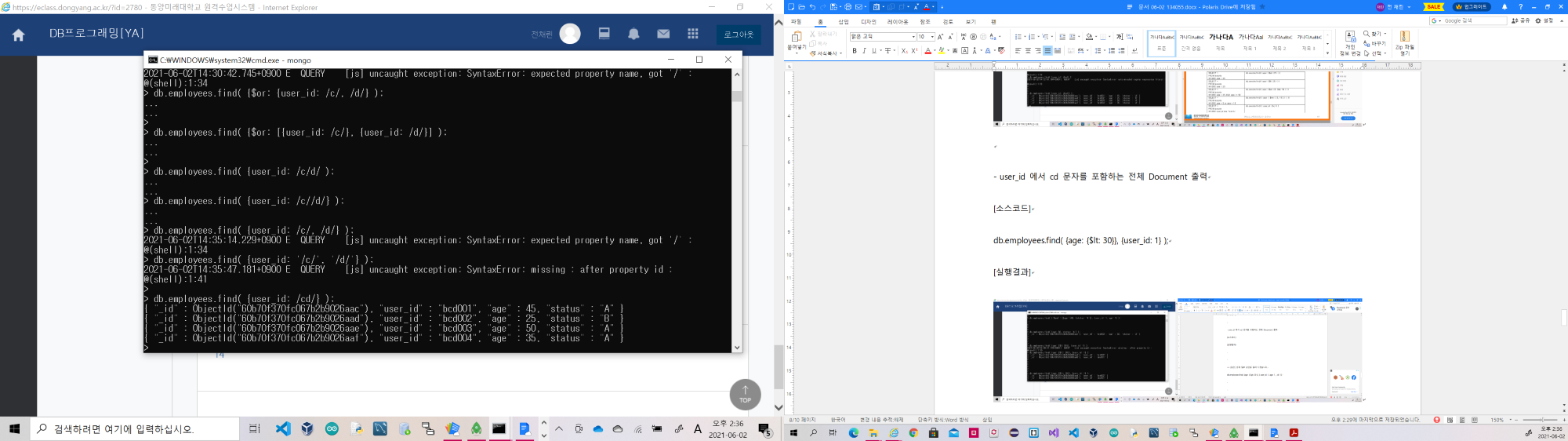
[실행결과]

{ "\_id" : ObjectId("60b70f370fc067b2b9026aac"), "user\_id" : "bcd001", "age" : 45, "status" : "A" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aad"), "user\_id" : "bcd002", "age" : 25, "status" : "B" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aae"), "user\_id" : "bcd003", "age" : 50, "status" : "A" }

{ "\_id" : ObjectId("60b70f370fc067b2b9026aaf"), "user\_id" : "bcd004", "age" : 35, "status" : "A" }



​

[과제03-02]

2. 다음 Document 데이터 수정하기

- age 가 40 보다 큰 Document 의 status 를 B 로 변환하기

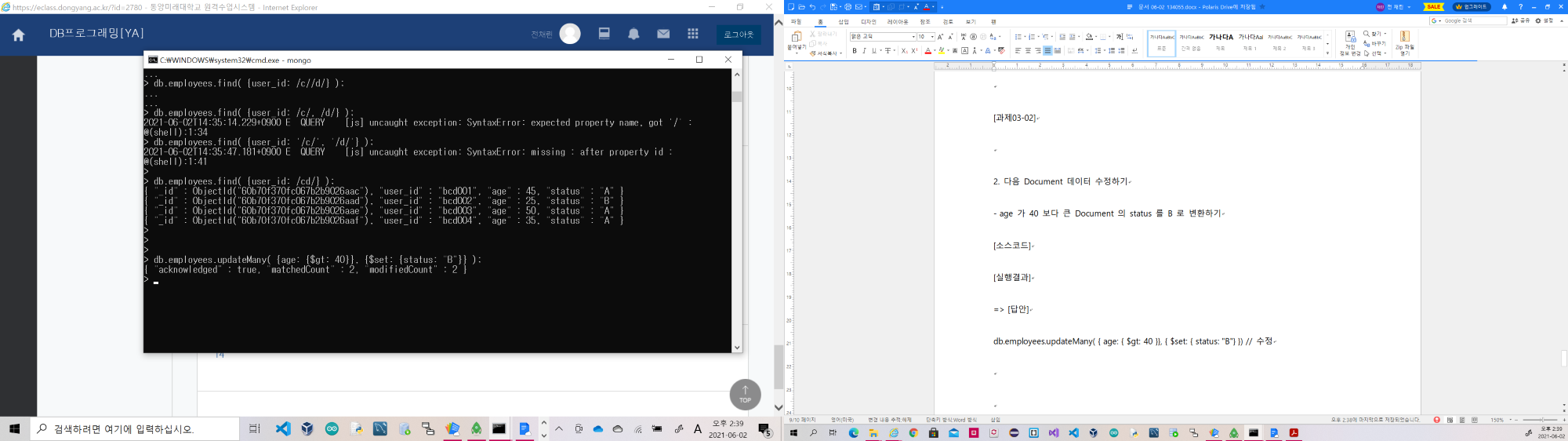
=> [답안]

[소스코드]

db.employees.updateMany( {age: {$gt: 40}}, {$set: {status: "B"}} );

​[실행결과]

{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }



​

[과제03-03]

3. 다음 Document 데이터 삭제하기

- age 가 30 보다 작은 Document 삭제하기

=> [답안]

​[소스코드]

db.employees.deleteMany( {age: {$lt: 30}} );

[실행결과]

{ "acknowledged" : true, "deletedCount" : 2 }

