

접속 및 셸 기본 조작 (mongosh)

MongoDB는 JSON 형태의 BSON 문서를 저장하는 유연하고 확장성이 뛰어난 NoSQL 문서 지향 데이터베이스입니다.

- 접속: `mongosh "mongodb://localhost:27017/mydb"`
- 데이터베이스 목록 확인: `show dbs`
- 데이터베이스 선택/전환: `use mydb`
- 컬렉션 목록 확인: `show collections`
- 컬렉션 생성/삭제: `db.createCollection("users") / db.users.drop()`
- 현재 DB 상태 및 통계: `db.stats(), db.users.stats()`

생성 (Create)

- 단일 문서 삽입:


```
db.users.insertOne({
  name: "홍길동",
  email: "hong@example.com",
  age: 28,
  tags: ["개발", "DB"],
  createdAt: new Date()
});
```
- 다중 문서 삽입:


```
db.users.insertMany([
  { name: "이순신", age: 35 },
  { name: "강감찬", age: 40 }
]);
```

조회 (Read & Query Operators)

기본 검색 및 프로젝션

- `db.users.find()`: 전체 조회
- `db.users.findOne({ _id: ObjectId("...") })`: 단일 문서 조회
- 프로젝션 (필드 선택): `db.users.find({}, { name: 1, email: 1, _id: 0 })`
- 정렬/제한/건너뛰기: `db.users.find().sort({ age: -1 }).skip(10).limit(5)`

주요 쿼리 연산자

- 비교: `$eq, $ne, $gt, $gte, $lt, $lte, $in: [1, 2], $nin`
- 논리: `$and, $or, $not, $nor`

- ```
db.users.find({
 $or: [{ age: { $gte: 30 } },
 { status: "VIP" }]
});
```
- 요소 & 정규식:
    - ▶ 필드 존재 여부: `db.users.find({ phone: { $exists: true } })`
    - ▶ 정규표현식: `db.users.find({ name: { $regex: /^홍/ } })`
    - ▶ 배열 검색: `db.users.find({ tags: "개발" })` (배열 내 요소 일치)
    - ▶ 배열 크기: `db.users.find({ tags: { $size: 2 } })`

## 수정 (Update Operators)

- 단일/다중 수정:
 

```
db.users.updateOne(
 { name: "홍길동" },
 {
 $set: { status: "active",
 updatedAt: new Date() },
 $inc: { visits: 1 },
 $unset: { tempField: "" },
 $push: { tags: "신규" },
 $addToSet: { tags: "중복방지" }
 },
 { upsert: true } // 없으면 새로 생성
);
```
- `db.users.updateMany({ age: { $lt: 20 } }, { $set: { isMinor: true } })`

## 삭제 (Delete)

- `db.users.deleteOne({ name: "홍길동" })`
- `db.users.deleteMany({ status: "inactive" })`

## 집계 파이프라인 (Aggregation Pipeline)

여러 단계를 거쳐 문서를 변환하고 집계합니다.

```
db.orders.aggregate([
 // 1단계: 조건 필터링
 { $match: { status: "completed" } },
```

```
 // 2단계: 그룹화 및 집계 연산
 {
 $group: {
 _id: "$customerId",
 totalSpent: { $sum: "$amount" },
 avgQuantity: { $avg: "$quantity" },
 orderCount: { $sum: 1 },
 items: { $push: "$item" }
 }
 },

 // 3단계: 정렬
 { $sort: { totalSpent: -1 } },

 // 4단계: 상위 5개 제한
 { $limit: 5 }
]);
```

### 주요 집계 단계 (Stages)

- `$project`: 출력 문서의 필드 구조 재구성
- `$unwind`: 배열 필드를 풀어 개별 문서로 분리
- `$lookup`: 다른 컬렉션과의 좌측 외부 조인(Left Outer Join)
 

```
{
 $lookup: {
 from: "users",
 localField: "userId",
 foreignField: "_id",
 as: "userInfo"
 }
}
```

## 인덱스 관리 (Indexes)

- 인덱스 생성: `db.users.createIndex({ email: 1 }, { unique: true })`
- 복합 인덱스: `db.users.createIndex({ age: 1, name: 1 })`
- TTL 인덱스 (자동 만료): `db.sessions.createIndex({ createdAt: 1 }, { expireAfterSeconds: 3600 })`
- 인덱스 목록 확인: `db.users.getIndexes()`
- 인덱스 삭제: `db.users.dropIndex("email_1")`

## 백업 및 내보내기 (CLI Tools)

- `mongodump --db=mydb --out=/backup/`

- `mongorestore --db=mydb /backup/mydb/`
- `mongoexport --db=mydb --collection=users --out=users.json`
- `mongoimport --db=mydb --collection=users --file=users.json`