

Проектирование баз данных, ч.2

ФИО преподавателя: Смирнов Михаил Вячеславович
e-mail: smirnovmgupi@gmail.com

Лекция 8

CRUD операции в noSQL

Виды инструкций для работы с документами mongo

- Инструкции вставки документов (Insert Documents)
- Инструкции выборки из документов (Query Documents)
- Инструкции изменения документов (Update Documents)
- Инструкции удаления документов (Delete Documents)

Инструкции вставки документов

```
db.collection.insertOne(  
  <document>, //документ, вставляемый в коллекцию  
  {  
    writeConcern: <document> //условия записи  
  }  
)
```

Инструкции вставки документа (mongo shell, пример)

```
db.inventory.insertOne(  
  { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom:  
    "cm" } }  
)
```

```
db.inventory.find( { item: "canvas" } )
```

Инструкции вставки документа (C#, пример)

```
var document = new BsonDocument
{
    { "item", "canvas" },
    { "qty", 100 },
    { "tags", new BsonArray { "cotton" } },
    { "size", new BsonDocument { { "h", 28 }, { "w", 35.5 }, { "uom", "cm" } } }
};
collection.InsertOne(document);

var filter = Builders<BsonDocument>.Filter.Eq("item", "canvas");
var result = collection.Find(filter).ToList();
```

Инструкции вставки нескольких документов (mongo shell, пример)

```
db.inventory.insertMany([  
  { item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },  
  { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },  
  { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom:  
"cm" } }  
])
```

Выборка всех документов из коллекции (mongo shell, C#)

```
db.inventory.find( {} )
```

```
var filter = Builders<BsonDocument>.Filter.Empty;
```

```
var result = collection.Find(filter).ToList();
```

Инструкция эквивалентна следующей инструкции языка SQL:

```
SELECT * FROM inventory
```


Выборка документов по условию эквивалентности (mongo shell, C#)

```
db.inventory.find( { status: "D" } )
```

```
var filter = Builders<BsonDocument>.Filter.Eq("status", "D");  
var result = collection.Find(filter).ToList();
```

Инструкция эквивалентна следующей инструкции языка SQL:

```
SELECT * FROM inventory WHERE status = "D"
```

Выборка документов по условию эквивалентности нескольким значениям
(mongo shell, C#)

```
db.inventory.find( { status: { $in: [ "A", "D" ] } } )
```

```
var filter = Builders<BsonDocument>.Filter.In("status", new[] { "A", "D" });  
var result = collection.Find(filter).ToList();
```

Инструкция эквивалентна следующей инструкции языка SQL:

```
SELECT * FROM inventory WHERE status in ("A", "D")
```

Выборка документов по условию AND (mongo shell, C#)

```
db.inventory.find( { status: "A", qty: { $lt: 30 } } )
```

```
var builder = Builders<BsonDocument>.Filter;  
var filter = builder.And(builder.Eq("status", "A"), builder.Lt("qty", 30));  
var result = collection.Find(filter).ToList();
```

Инструкция эквивалентна следующей инструкции языка SQL:

```
SELECT * FROM inventory WHERE status = "A" AND qty < 30
```

Выборка документов по условию OR (mongo shell, C#)

```
db.inventory.find( { $or: [ { status: "A" }, { qty: { $lt: 30 } } ] } )
```

```
var builder = Builders<BsonDocument>.Filter;  
var filter = builder.Or(builder.Eq("status", "A"), builder.Lt("qty", 30));  
var result = collection.Find(filter).ToList();
```

Инструкция эквивалентна следующей инструкции языка SQL:

```
SELECT * FROM inventory WHERE status = "A" OR qty < 30
```

Выборка документов с комбинированием условий AND и OR (mongo shell, C#)

```
db.inventory.find( {  
    status: "A",  
    $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ]  
})
```

```
var builder = Builders<BsonDocument>.Filter;  
var filter = builder.Or(builder.Eq("status", "A"), builder.Lt("qty", 30));  
var result = collection.Find(filter).ToList();
```

Инструкция эквивалентна следующей инструкции языка SQL:

```
SELECT * FROM inventory WHERE status = "A" AND ( qty < 30 OR item LIKE "p%")
```

Пример документа для инструкций изменения документа (mongo shell)

```
db.inventory.insertMany( [  
  { item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" }, status: "A" },  
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },  
  { item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status: "A" },  
  { item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" }, status: "P" },  
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" },  
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },  
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },  
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" },  
  { item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" }, status: "A" },  
  { item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" }, status: "A" }  
] );
```

Инструкция изменения одного документа (mongo shell, пример)

```
db.inventory.updateOne(
  { item: "paper" },
  {
    $set: { "size.uom": "cm", status: "P" },
    $currentDate: { lastModified: true }
  }
)
```

Инструкция изменения одного документа (C#, пример)

```
var filter = Builders<BsonDocument>.Filter.Eq("item", "paper");
```

```
var update = Builders<BsonDocument>.Update.Set("size.uom", "cm").Set("status",  
"P").CurrentDate("lastModified");
```

```
var result = collection.UpdateOne(filter, update);
```


Инструкция изменения группы документов (mongo shell, пример)

```
db.inventory.updateMany(  
  { "qty": { $lt: 50 } },  
  {  
    $set: { "size.uom": "in", status: "P" },  
    $currentDate: { lastModified: true }  
  }  
)
```

Инструкция изменения группы документов (C#, пример)

```
var filter = Builders<BsonDocument>.Filter.Lt("qty", 50);
```

```
var update = Builders<BsonDocument>.Update.Set("size.uom", "in").Set("status",  
"P").CurrentDate("lastModified");
```

```
var result = collection.UpdateMany(filter, update);
```

Инструкция замены документа (mongo shell, пример)

```
db.inventory.replaceOne(  
  { item: "paper" },  
  { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse:  
"B", qty: 40 } ] }  
)
```

Инструкция замены документа (C#, пример)

```
var filter = Builders<BsonDocument>.Filter.Eq("item", "paper");  
var replacement = new BsonDocument  
{  
    { "item", "paper" },  
    { "instock", new BsonArray  
        {  
            new BsonDocument { { "warehouse", "A" }, { "qty", 60 } },  
            new BsonDocument { { "warehouse", "B" }, { "qty", 40 } } }  
        }  
};  
var result = collection.ReplaceOne(filter, replacement);
```

Пример документа для инструкций удаления документа (mongo shell)

```
db.inventory.insertMany( [  
  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },  
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" },  
  { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },  
  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },  
  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" },  
]);
```

Инструкция удаления всех документов коллекции (mongo shell, C#, пример)

```
db.inventory.deleteMany({})
```

```
var filter = Builders<BsonDocument>.Filter.Empty;  
var result = collection.DeleteMany(filter);
```

Инструкция удаления документов по условию (mongo shell, C#, пример)

```
db.inventory.deleteMany({ status : "A" })
```

```
var filter = Builders<BsonDocument>.Filter.Eq("status", "A");  
var result = collection.DeleteMany(filter);
```

Полезные ссылки:

- MongoDB Manual <https://docs.mongodb.com/manual/crud/>
- Пример web-API в JSON, API Spotify
<https://developer.spotify.com/documentation/web-api/reference/>
- Видеопрактикум “Основы работы с JSON-документами через приложение в QT”:
<https://www.youtube.com/playlist?list=PLbE0Wv5gl4StADy4cKZ3503aNggE4cHgN>
- Видеопрактикум “Создание клиент-серверного приложения TCP/Socket/JSON/SQLite, создание приложения для работы с API веб-сервиса в QT”: <https://www.youtube.com/playlist?list=PLbE0Wv5gl4Ssgx-Otlcl9WsrzgWQfUXMH>

Спасибо за внимание!