



Operaciones CRUD

Índice

1. Create
2. Read
3. Update
4. Delete



1. Create

Insert(<document or array of documents>).

Inserta uno o varios documentos, crea colección si no existe.

```
db.inventory.insert([
  { 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" } }
])
```

```
CREATE TABLE people (
  id MEDIUMINT NOT NULL
    AUTO_INCREMENT,
  user_id Varchar(30),
  age Number,
  status char(1),
  PRIMARY KEY (id)
)
```

Implicitly created on first `insertOne()` or `insertMany()` operation. The primary key `_id` is automatically added if `_id` field is not specified.

```
db.people.insertOne( {
  user_id: "abc123",
  age: 55,
  status: "A"
} )
```

However, you can also explicitly create a collection:

```
db.createCollection("people")
```

```
INSERT INTO people(user_id,
  age,
  status)
VALUES ("bcd001",
  45,
  "A")
```

```
db.people.insertOne(
  { user_id: "bcd001", age: 45, status: "A" }
)
```

<https://docs.mongodb.com/manual/reference/insert-methods/>



2. Read

find(<filter>,<projection>).

Devuelve un cursor con los documentos que cumplen la condición

```
SELECT *  
FROM people
```

```
db.people.find()
```

```
SELECT *  
FROM people  
WHERE status != "A"
```

```
db.people.find(  
  { status: { $ne: "A" } }  
)
```

```
SELECT *  
FROM people  
WHERE status = "A"  
AND age = 50
```

```
db.people.find(  
  { status: "A",  
    age: 50 }  
)
```

```
SELECT *  
FROM people  
WHERE status = "A"  
OR age = 50
```

```
db.people.find(  
  { $or: [ { status: "A" }, { age: 50 } ] }  
)
```

```
SELECT *  
FROM people  
WHERE age > 25
```

```
db.people.find(  
  { age: { $gt: 25 } }  
)
```

```
SELECT *  
FROM people  
WHERE age < 25
```

```
db.people.find(  
  { age: { $lt: 25 } }  
)
```



Proyecciones

Las consultas en MongoDB devuelven todos los campos en documentos coincidentes. Para limitar la cantidad de datos que MongoDB envía a las aplicaciones, puede incluir un documento de proyección para especificar o restringir los campos para regresar.

- Devuelve los campos especificados y campo `_id`
`db.inventory.find({ status: "A" }, { item: 1, status: 1 })`
- Suprimir campo `_id`
`db.inventory.find({ status: "A" }, { item: 1, status: 1, _id: 0 })`
- Devolver todos los campos menos campos excluidos
`db.inventory.find({ status: "A" }, { status: 0, instock: 0 })`

```
SELECT id,  
       user_id,  
       status  
FROM people
```

```
db.people.find(  
  { },  
  { user_id: 1, status: 1 }  
)
```

```
SELECT user_id, status  
FROM people
```

```
db.people.find(  
  { },  
  { user_id: 1, status: 1, _id: 0 }  
)
```



3. Update

update(<filter>, <update>, <options>).

Actualiza el primer o todos los registros que cumplan el filtro.

```
db.restaurant.updateMany(  
  { "name" : "Pizza Rat's Pizzeria" },  
  { $set: { "_id" : 4, "violations" : 7, "borough" : "Manhattan" } }  
);
```

Opcion upsert. Si es true, si no cumple la condición de filtro, creará el documento. Por defecto, false

Opcion Multi. Si es true, modificará todos los documentos que cumplan la condición.

```
UPDATE people  
SET status = "C"  
WHERE age > 25
```

```
db.people.updateMany(  
  { age: { $gt: 25 } },  
  { $set: { status: "C" } }  
)
```

```
UPDATE people  
SET age = age + 3  
WHERE status = "A"
```

```
db.people.updateMany(  
  { status: "A" },  
  { $inc: { age: 3 } }  
)
```



Operadores <Update>

- **\$currentDate**. Establece el valor de un campo en la fecha actual.
- **\$inc**. Incrementa el valor del campo en la cantidad especificada.
- **\$min**. Modifica al valor especificado si es menor que este.
- **\$max**. Modifica al valor especificado si es mayor que este.
- **\$mul**. Multiplica el valor del campo por la cantidad especificada.
- **\$rename**. Renombra el nombre de un campo.
- **\$set**. Actualiza o añade el valor de/a un campo.
- **\$setOnInsert**. Actualiza o añade el valor de/a un campo si update inserta un documento.
- **\$unset**. Elimina el campo especificado.

```
db.students.updateMany( {}, { $rename: { "nmae": "name" } } )
```

```
db.products.update( {}, { $inc: { quantity: -2, "metrics.orders": 1 } } )
```

```
db.products.update( {}, { $set: {
```

```
    quantity: 500,
```

```
    details: { model: "14Q3", make: "xyz" },
```

```
    tags: [ "coats", "outerwear", "clothing" ]
```

```
  } } )
```

```
db.products.update( { },
```

```
  { $unset: { quantity: "", instock: "" } } )
```



4. Delete

DeleteMany(<filter>).

Borra todos los documentos

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

```
DELETE FROM people  
WHERE status = "D"
```

```
db.people.deleteMany( { status: "D" } )
```

```
DELETE FROM people
```

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



