

# OPERACIONES BÁSICAS

OPERACION	DESCRIPCION
show dbs	Muestra todas las BD.
show collections	Muestra las colecciones de la BD.
use miDB	Cambia a la base miDb, si no existe la crea.
db.miColeccion.insert({nombre:'mario'})	Inserta un objeto en miColeccion.
db.miColeccion.update({nombre:'mario', {\$set{comentarios:[]}}}	Modifica el documento
db.miColeccion.remove({nombre:'mario'})	Borra el documento cuyo nombre es mario.
db.miColeccion.find()	Muestra los datos de miColeccion
db.miColeccion.find().pretty()	Muestra la consulta más legible
db.miColeccion.find().explain()	Nos dice que es lo que está haciendo la consulta
db.miColeccion.find({nombre:'mario'})	Busca un documento que contenga el nombre = mario en miColeccion
db.miColeccion.findOne()	Muestra los datos de "miColeccion", al encontrar el primero, termina el proceso. Se usa para ver qué información contiene una colección.
db.miColeccion.count()	Obtiene la cantidad de documentos de miColeccion
db.miColeccion.createIndex({nom	Crea un índice ascendente sobre

bre:1})	el campo nombre
db.miColeccion.drop()	Borra toda la coleccion, es mucho mas rapido que remove().

## Find:

db.miColecion.find(consulta, proyección)

db.miColeccion.find({edad:20}, {nombre:1})

db.miColeccion.find({edad:20}).limit(3)

db.miColeccion.find({edad:20}).sort({nombre:1})

Descendente: 1.

Ascendente: -1.

## Selectores de Búsqueda:

### Comparativos:

- \$eq: Iguala al valor especificado.
- \$gt: Mayor que.
- \$gte: Mayor o igual.
- \$lt: Menor que.
- \$lte: Menor igual.
- \$ne: Distinto.
- \$in: Entre los valores [a, b, c, ...].
- \$nin: No esta entre los valores [a, b, c, ...].

db.miColeccion.find({edad:{\$gt: 20}, {nombre: 1, \_id:0})

### Logicos:

- \$or:

- \$and:
- \$nor:
- \$not:

## Elementales:

- \$exists: Coincide con documentos que tienen el campo especificado.
- \$type: Selecciona documentos si un campo es del tipo especificado.

## Valores de \$type:

- 1 - Double.
- 2 - String.
- 3 - Objeto.
- 4 - Arreglo.
- 5 - Data binaria.
- 6 - Indefinido (deprecado).
- 7 - Id de objeto.
- 8 - Booleano.
- 9 - Fecha.
- 10 - Nulo.
- 11 - Expresión regular.
- 13 - Javascript.
- 14 - Símbolo.
- 15 - Javascript con alcance definido.
- 16 - Entero de 32bit.
- 17 - Estampilla de tiempo.
- 18 - Entero de 64bit.
- 127 - Llave máxima.
- 255 - Llave mínima.

```
db.miColeccion.find({edad: {$type: 1}})
```

# SQL vs NoSQL

## SQL INSERT Statements

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

## MongoDB insert() Statements

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

## SQL SELECT Statements

```
SELECT *
FROM users
```

## MongoDB find() Statements

```
db.users.find()
```

```
SELECT id,
       user_id,
       status
FROM users
```

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

```
SELECT user_id,
status
```

```
db.users.find(
  { },
```

---

<b>FROM</b> users	{ user_id: 1, status: 1, _id: 0 }
	)

---

<b>SELECT</b> *	db.users.find(
<b>FROM</b> users	{ status: "A" }
<b>WHERE</b> status = "A"	)

---

<b>SELECT</b> user_id,	db.users.find(
status	{ status: "A" },
<b>FROM</b> users	{ user_id: 1, status: 1, _id: 0 }
<b>WHERE</b> status = "A"	)

---

<b>SELECT</b> *	db.users.find(
<b>FROM</b> users	{ status: { \$ne: "A" } }
<b>WHERE</b> status != "A"	)

---

<b>SELECT</b> *	db.users.find(
<b>FROM</b> users	{ status: "A",
<b>WHERE</b> status = "A"	age: 50 }
<b>AND</b> age = 50	)

---

---

<b>SELECT *</b>	<b>db.users.find(</b>
<b>FROM users</b>	<b>{ \$or: [ { status: "A" },</b>
<b>WHERE status = "A"</b>	<b>{ age: 50 } ] }</b>
<b>OR age = 50</b>	<b>)</b>

---

<b>SELECT *</b>	<b>db.users.find(</b>
<b>FROM users</b>	<b>{ age: { \$gt: 25 } }</b>
<b>WHERE age &gt; 25</b>	<b>)</b>

---

<b>SELECT *</b>	<b>db.users.find(</b>
<b>FROM users</b>	<b>{ age: { \$lt: 25 } }</b>
<b>WHERE age &lt; 25</b>	<b>)</b>

---

<b>SELECT *</b>	<b>db.users.find(</b>
<b>FROM users</b>	<b>{ age: { \$gt: 25, \$lte: 50 } }</b>
<b>WHERE age &gt; 25</b>	<b>)</b>
<b>AND age &lt;= 50</b>	

---

<b>SELECT *</b>	<b>db.users.find( { user_id: /bc/ } )</b>
<b>FROM users</b>	
<b>WHERE user_id like</b>	
<b>"%bc%"</b>	

---

---

**SELECT \***  
**FROM** users  
**WHERE** user\_id like  
"bc%"

db.users.find( { user\_id: /^bc/ } )

---

**SELECT \***  
**FROM** users  
**WHERE** status = "A"  
**ORDER BY** user\_id  
**ASC**

db.users.find( { status: "A" } ).sort( { user\_id: 1 }  
)

---

**SELECT \***  
**FROM** users  
**WHERE** status = "A"  
**ORDER BY** user\_id  
**DESC**

db.users.find( { status: "A" } ).sort( { user\_id: -1  
} )

---

**SELECT COUNT(\*)**  
**FROM** users

db.users.count()

*or*

db.users.find().count()

---

---

<b>SELECT</b>	<code>db.users.count( { user_id: { \$exists: true } } )</code>
<b>COUNT(user_id)</b>	
<b>FROM users</b>	

*or*

`db.users.find( { user_id: { $exists: true } }  
).count()`

---

<b>SELECT COUNT(*)</b>	<code>db.users.count( { age: { \$gt: 30 } } )</code>
<b>FROM users</b>	
<b>WHERE age &gt; 30</b>	

*or*

`db.users.find( { age: { $gt: 30 } } ).count()`

---

<b>SELECT</b>	<code>db.users.distinct( "status" )</code>
<b>DISTINCT(status)</b>	
<b>FROM users</b>	

---

<b>SELECT *</b>	<code>db.users.findOne()</code>
<b>FROM users</b>	

---



---

**LIMIT 1**

*or*

**db.users.find().limit(1)**

---

**SELECT \***  
**FROM users**  
**LIMIT 5**  
**SKIP 10**

**db.users.find().limit(5).skip(10)**

---

**EXPLAIN SELECT \***  
**FROM users**  
**WHERE status = "A"**

**db.users.find( { status: "A" } ).explain()**

---

SQL Update Statements

MongoDB update() Statements

---

**UPDATE users**  
**SET status = "C"**  
**WHERE age > 25**

**db.users.update(**  
    **{ age: { \$gt: 25 } },**  
    **{ \$set: { status: "C" } },**  
    **{ multi: true }**  
**)**

---

---

**UPDATE** users  
**SET** age = age + 3  
**WHERE** status = "A"

db.users.update(  
 { status: "A" },  
 { \$inc: { age: 3 } },  
 { multi: true }  
)

---

SQL Delete Statements

MongoDB remove() Statements

---

**DELETE FROM** users  
**WHERE** status = "D"

db.users.remove( { status: "D" }  
)

---

**DELETE FROM** users

db.users.remove({})

---