

MongoDB CRUD

Índice

- 01 [Colecciones](#)
- 02 [Documentos](#)
- 03 [Comandos para la utilización en el Mongosh](#)



01

Colecciones

Colecciones

MongoDB almacena documentos en las colecciones.

Las colecciones son análogas a las tablas en la base de datos relacionales.

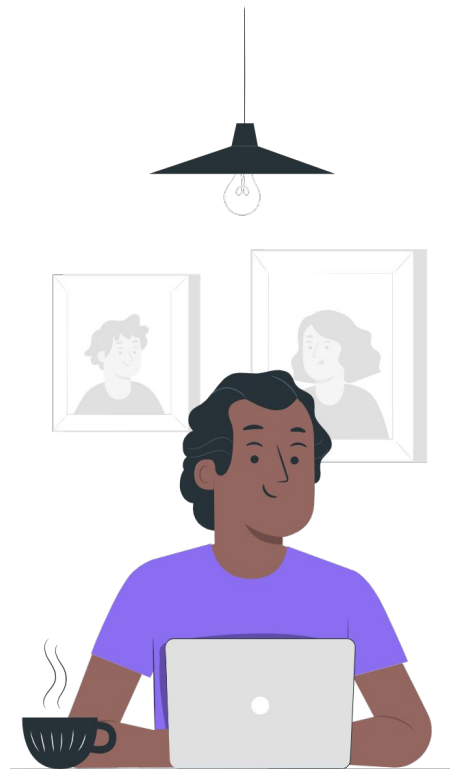


Creando una colección

Si una colección no existe, MongoDB crea la colección cuando almacena los datos por primera vez en dicha colección.

```
db.myNewCollection2.insertOne( { x: 1 } )  
db.myNewCollection3.createIndex( { y: 1 } )
```

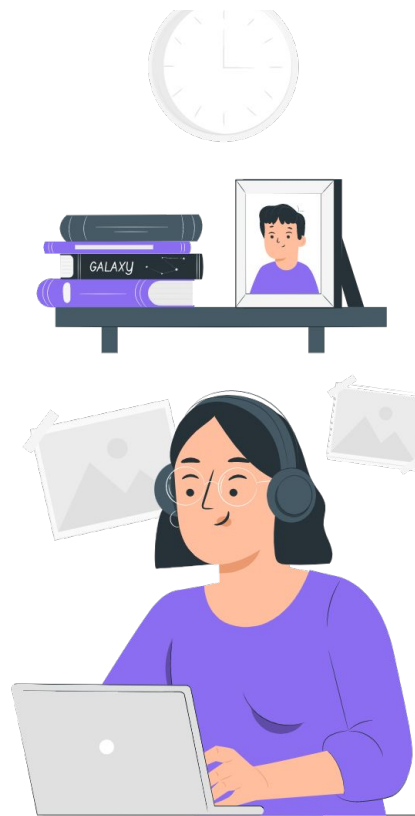
Tanto insertOne() como createIndex() crean sus respectivas colecciones si aún no existen.



Creación explícita

MongoDB proporciona el método `db.createCollection()` para crear explícitamente una colección con múltiples opciones. Como, por ejemplo, definir el tamaño máximo o las reglas de validación de la documentación.

Si no se va a especificar estas opciones, no se necesita crear explícitamente la colección. Después de todo, MongoDB lo crea automáticamente la primera vez que almacena datos para colecciones.



02

Documentos

Documentos

De forma predeterminada, los documentos de una sola colección no necesitan tener el mismo conjunto de campos. El tipo de datos de un campo puede diferir entre los documentos de una colección.

Cada documento debe contener una identificación única asociada, especificada por el campo `_id`.

Si el usuario no especifica un valor para este campo, MongoDB lo genera automáticamente, definido como `ObjectId()`.

MongoDB también crea un índice para el campo `_id` para que las consultas sean más eficientes.

Podemos especificar un `_id` (que será único en la colección):
`db.personas.insert({_id: 001, nombre: "Ana Maria Braga"})`

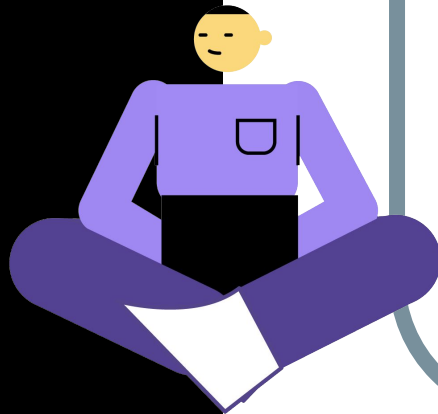
03

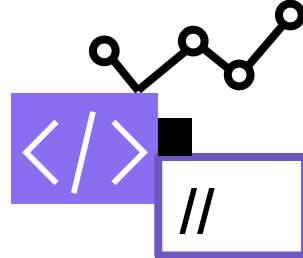
Comandos para utilización en Mongosh



A continuación, conocerás algunos comandos de **Mongosh** para:

- Insertar documentos en una colección.
- Seleccionar documentos en una colección.
- Seleccionar datos de un conjunto específico.
- Actualizar documentos.
- Eliminar documentos.





FIND - Seleccionando los documentos de la colección

Las operaciones de lectura recuperan documentos de una colección. MongoDB proporciona algunos métodos para leer documentos. Vean a continuación:

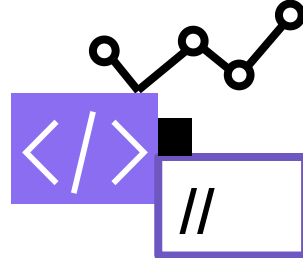
- Para seleccionar todos los documentos de la colección, pasen un documento vacío como parámetro de filtro de consulta al método de búsqueda. El parámetro de filtro de consulta determina los criterios de selección.

```
{ }  
  
db.mycol.find( { } )  
{ _id: 83, titulo: 'MongoDB' }  
{ _id: 84, titulo: 'MySQL' }  
{ _id: 85, titulo: 'Cassandra' }  
{ _id: 86, titulo: 'SQL Server' }  
{ _id: 87, titulo: 'SAS' }  
{ _id: 88, titulo: 'Javascript' }  
{ _id: 89, titulo: 'HTML' }
```

- Esta es la devolución de la consulta.

Esta operación
corresponde a la
siguiente
instrucción SQL:

```
SELECT * FROM  
mycol
```



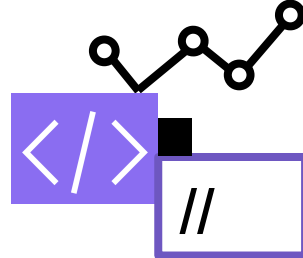
INSERT - Ingresando documentos en la colección

MongoDB proporciona los siguientes métodos para insertar documentos en una colección:

```
db.collection.insertOne()
db.collection.insertMany()

db.mycol.insertOne({_id: 87, titulo: 'SAS'})
{} { acknowledged: true, insertedId: 87 } // retorno de la consulta

db.mycol.insertMany([{_id: 88, titulo: 'Javascript'}, {_id: 89, titulo: 'HTML'}])
{ acknowledged: true, insertedIds: { '0': 88, '1': 89 } } // retorno de la consulta
```

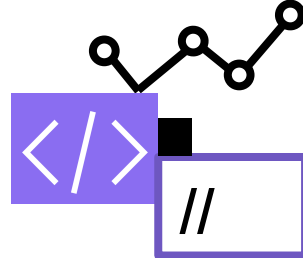


FIND - Seleccionando los datos de un conjunto

- Para seleccionar todos los datos de un conjunto específico:

```
db.mycol.find( { titulo: { $in: [ "MongoDB", "MySQL" ] } } )  
{ _id: 83, titulo: 'MongoDB' } // retorno de la consulta  
{ _id: 84, titulo: 'MySQL' }
```

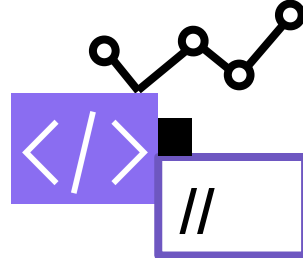
Este comando es equivalente a: **SELECT * FROM mycol WHERE titulo in ("MongoDB, "MySQL")**



FIND - Seleccionando los datos sin conjunto

- Para seleccionar todos los datos que no pertenecen a un conjunto específico:

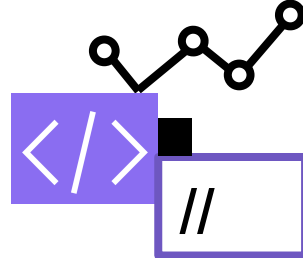
```
{ } db.mycol.find( { titulo: { $nin: [ "MongoDB", "MySQL" ] } } )  
{ _id: 87, titulo: 'SAS' }  
{ _id: 88, titulo: 'Javascript' }  
{ _id: 89, titulo: 'HTML' } // Resultado de la consulta  
{ _id: 90, titulo: 'CSS' }  
{ _id: 91, titulo: 'Bootstrap' }
```



UPDATE - Actualizando los documentos

Para actualizar un documento, MongoDB provee operadores de actualización. Como **\$set**, para modificar los valores de los campos.

```
{  
  db.mycol.update({titulo : "HTML"}, {$set : {titulo: "HTML5"}})  
  { acknowledged: true,  
    insertedId: null,  
    matchedCount: 1,    // retorno de la instrucción  
    modifiedCount: 1,  
    upsertedCount: 0 }  
}
```



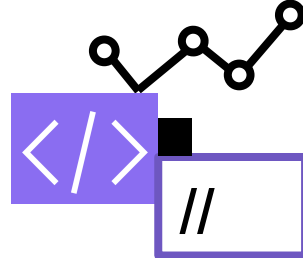
UPDATE - Actualizando los documentos

- Para verificar la actualización, podemos utilizar el comando find:

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

- Podemos utilizar, también, el comando **updateOne**, para actualizar solo un registro:

```
{ } db.mycol.updateOne({titulo:"SAS"}, {$set : {titulo : "Java"}})
```

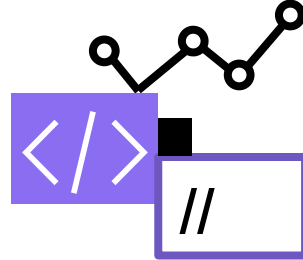



UPDATE - Actualizando los documentos

- Podemos también utilizar la opción **updateMany**, para actualizar varios registros de una sola vez:

```
{  
  db.mycol.updateMany({_id: {$lt : 85}}, {$set:{titulo:"Teste"}})  
  { acknowledged: true,  
    insertedId: null,  
    matchedCount: 2,           // retorno de la instrucción  
    modifiedCount: 2,  
    upsertedCount: 0 }  
}
```

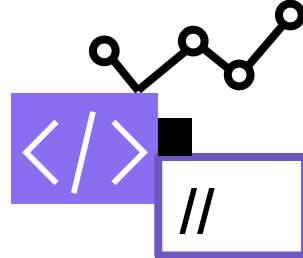
En esta instrucción, fueron modificados 2 registros, o sea, todos los registros cuyo id eran menores que 85.



DELETE - Excluyendo los documentos

- Para excluir un documento, utilicen la operación **deleteOne**.
Observen el ejemplo: en él, excluimos el registro que contenía la palabra test como valor del campo título.

```
{ } db.mycol.deleteOne({ titulo : "Test" })
```



DELETE - Excluyendo los documentos

- Para excluir varios documentos, utilicen la operación **deleteMany**. En el ejemplo abajo, excluimos todos los registros que tenían **_id >= 87**.

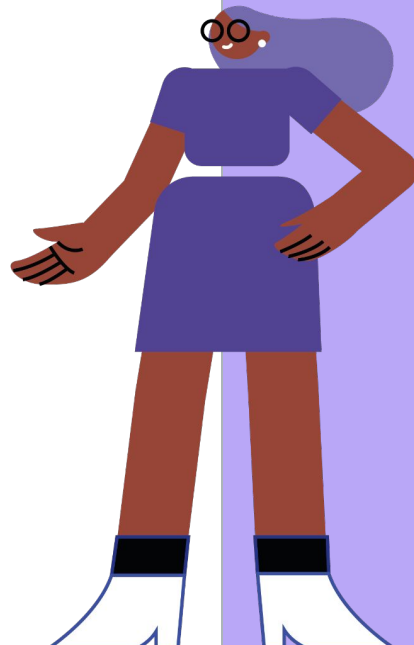
```
{ } db.mycol.deleteMany({ _id : { $gte : 87 } })  
{ acknowledged: true, deletedCount: 3 } // retorno de la instrucción
```

Como la colección tenía los registros 87, 88 y 89, los registros fueron excluidos.

Conclusiones

En esta clase, repasamos los conceptos de **Colecciones** y **Documentos** y practicamos los **comandos básicos de MongoDB**.

¡Hasta pronto!



¡Muchas gracias!