

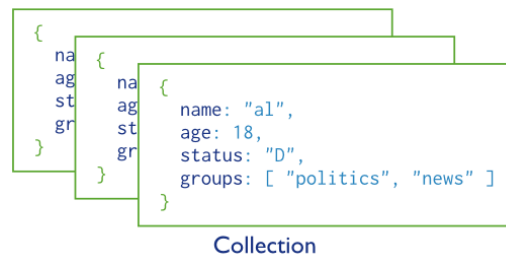
Introducción a CRUD (Create, Read, Update, Delete)

MongoDB almacena la información en documentos, los cuales son de formato JSON. Estos documentos son similares a las estructuras de programación que son asociadas con un identificador y un valor. Estas llaves puede tener también asociada otras llaves y valor como por ejemplos diccionarios, hash, maps y vectores relacionados. En MongoDB los documentos son almacenados como documentos BSON, el cuales es la representación binaria de JSON con un tipo adicional de información.

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

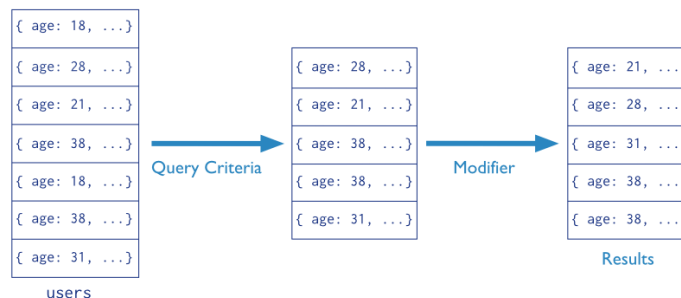
Los documentos son almacenados en colecciones una colección es un grupo de documentos relacionados que tienen un grupo de índices compartidos. Las colecciones son similares a una tabla en las bases relaciones.



Consultas

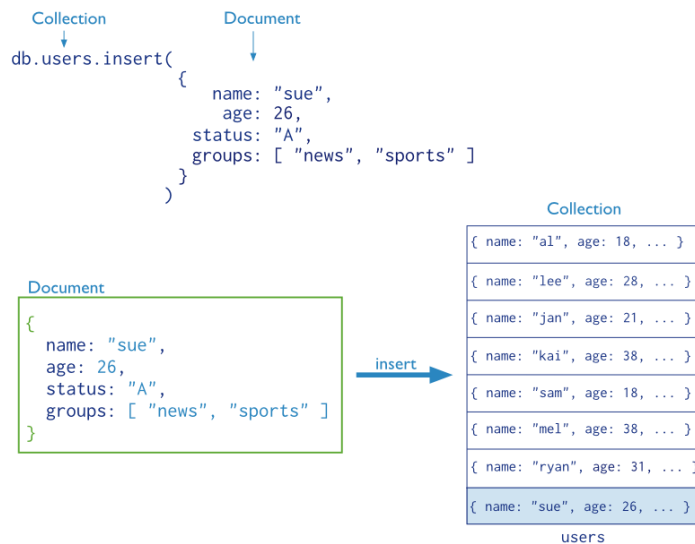
En MongoDB una consulta apunta a una específica colección de documentos. Las consultas deberán especificar un criterio o condiciones que identifica los documentos que MongoDB retorna a los clientes. Una consulta debe incluir una proyección que especifica los campos que concuerden con los documentos que retorna. Usted opcionalmente puede modificar las consultas para imponer límites, no tener en cuenta documentos, y orden.

Collection Query Criteria Modifier
`db.users.find({ age: { $gt: 18 } }).sort({age: 1 })`



Modificación de datos

La modificación de datos se refiere a las operaciones para crear, actualizar o eliminar datos. En MongoDB, estas operaciones modifican la información de una sola colección. Para las operaciones de actualización y eliminación usted podrá seleccionar varios documentos para actualizar o eliminarlos.



Operaciones de Lectura

Las operaciones de lectura devuelven la información almacenada en la base de datos, las consultas seleccionan los documentos de una única colección.

Mongodb posee el método `db.coleccion.find()`. Este método acepta los criterios de proyección y devuelve un cursor a los documentos correspondientes. Usted podrá modificar opcionalmente la consulta de imponer límites, omitir información y criterios de ordenaciones.

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

```
SELECT _id, name, address  
FROM users  
WHERE age > 18  
LIMIT 5
```

← projection
← table
← select criteria
← cursor modifier

Operadores de consultas

Comparación

Nombre	Descripción
\$gt	Mayor que.
\$gte	Mayor igual que.
\$in	Existe en el arreglo.
\$lt	Menor que.
\$lte	Menor igual que.
\$ne	Diferente.
\$nin	No existe en el array.

Lógicas

Nombre	Descripción
\$or	OR
\$and	AND
\$not	NOT
\$nor	NOR

Elementos

Nombre	Descripción
\$exists	Busca un campo específico
\$type	Busca que un campo específico sea de un tipo específico

Evaluación

Nombre	Descripción
\$mod	Realiza el módulo de un campo específico y retorna documentos con un resultado especificado.
\$regex	Selecciona los documentos donde los valores se ajustan a un valor especificado como una expresión regular.

\$text	Busca textos
\$where	Busca que se satisfaga una expresión de JavaScript

Geoespacial

Nombre	Descripción
\$geoWithin	Selecciona la geometría que se encuentre en los límites geométricos. GeoJSON.
\$geoIntersects	Selecciona la geometría que intersecta con los límites geométricos. GeoJSON.
\$near	Retorna los objetos geoespaciales cerca de un punto.
\$nearSphere	Retorna los objetos geoespaciales cerca de un punto sobre la esfera.

Arreglos

Nombre	Descripción
\$all	Coincide con matrices que contienen todos los elementos especificados en la consulta.
\$elemMatch	Selecciona los documentos si los elementos del Arreglo que concuerdan con la condición especificada.
\$size	Selecciona los documentos si el arreglo tiene una dimensión especificada.

Operadores de Proyección

Nombres	Descripción
\$	Proyecta el primer elemento de un arreglo que concuerde con la condición de consulta.
\$elemMatch	Proyecta solo el primer elemento de un arreglo que concuerde con una condición especificada.
\$meta	Proyecta el puntaje asignado durante la operación \$text.
\$slice	Limita el número de elementos proyectados de un arreglo.

Operaciones de Escritura

Creación

La creación agrega nuevos documentos a la colección. En MongoDB el metodo es db.coleccion.insert().

```
db.users.insert (  ← collection
{
  name: "sue",      ← field: value
  age: 26,          ← field: value
  status: "A"       ← field: value
}                  } document
)

INSERT INTO users  ← table
      ( name, age, status ) ← columns
VALUES      ( "sue", 26, "A" ) ← values/row
```

Para crear la llave primario debemos agregar el campo `_id`, si no se agrega esta campo agregara uno por defecto.

Actualización

Para actualizar documentos en una colección. En mongoDB, existen los métodos `db.coleccion.update()` y el `db.coleccion.save()`. El método `update` acepta un criterio de búsqueda que determina que documentos se debe actualizar como también modifica múltiples filas. El método puede también aceptar opciones que afecten su comportamiento como la opción `multi` que permite modificar los múltiples documentos.

```
db.users.update(  ← collection
{ age: { $gt: 18 } }, ← update criteria
{ $set: { status: "A" } }, ← update action
{ multi: true }      ← update option
)

UPDATE users  ← table
SET   status = 'A' ← update action
WHERE age > 18  ← update criteria
```

El método `save()` reemplaza el archivo existente que concuerde con el `_id`.

Eliminación

Elimina un documento de una colección. En MongoDB se hace con el método `db.coleccion.remove()`. Este método puede aceptar criterios de búsqueda.

```
db.users.remove(      ← collection  
  { status: "D" }    ← remove criteria  
)
```

```
DELETE FROM users    ← table  
WHERE status = 'D'   ← delete criteria
```