

19 Indexación: Índices únicos

A unique index ensures that the indexed fields do not store duplicate values; i.e. enforces uniqueness for the indexed fields. By default, MongoDB creates a unique index on the `_id` field during the creation of a collection.

Create a Unique Index

To create a unique index, use the `db.collection.createIndex()` method with the `unique` option set to `true`.

Sintaxis

```
db.collection.createIndex( <key and index type specification>, { unique: true } )
```

Práctica

```
> db.clientes.createIndex({dni: 1},{ unique: true })
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.clientes.createIndex({tie: 1},{ unique: true })
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 2,
  "numIndexesAfter" : 3,
  "ok" : 1
}

> db.clientes.insert({nombre:"Javier", apellidos: "Pérez López", dni: "44890567P"})
WriteResult({ "nInserted" : 1 })
> db.clientes.insert({nombre:"Laura", apellidos: "Pérez García", dni: "44890567P"})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: shop.clientes index:
dni_1 dup key: { : \"44890567P\" }"
  }
})
```

```

> db.clientes.dropIndexes()
{
  "nIndexesWas" : 3,
  "msg" : "non-_id indexes dropped for collection",
  "ok" : 1
}
> db.clientes.insert({nombre:"Javier", apellidos: "Pérez López", dni: "44890567P"})
WriteResult({ "nInserted" : 1 })
> db.clientes.insert({nombre:"Laura", apellidos: "Pérez García", dni: "44890567P"})
WriteResult({ "nInserted" : 1 })
> db.clientes.createIndex({dni: 1},{ unique: true })
{
  "ok" : 0,
  "errmsg" : "E11000 duplicate key error collection: shop.clientes index: dni_1
dup key: { : \"44890567P\" }",
  "code" : 11000,
  "codeName" : "DuplicateKey"
}

```

Unique Compound Index

You can also enforce a unique constraint on compound indexes. If you use the unique constraint on a compound index, then MongoDB will enforce uniqueness on the combination of the index key values.

Práctica

```

> db.oficinas.createIndex({"despachos.planta": 1, "despachos.letra": 1},{unique: true})
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.oficinas.insert({despachos: [{planta: 0, letra: "A"}, {planta: 0, letra: "B"}]})
WriteResult({ "nInserted" : 1 })
> db.oficinas.insert({despachos: [{planta: 0, letra: "C"}, {planta: 0, letra: "D"}]})
WriteResult({ "nInserted" : 1 })
> db.oficinas.insert({despachos: [{planta: 0, letra: "C"}, {planta: 0, letra: "E"}]})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: shop.oficinas index:
despachos.planta_1_despachos.letra_1 dup key: { : 0.0, : \"C\" }"
  }
})

```

```
}  
})
```

Unique Index and Missing Field

If a document does not have a value for the indexed field in a unique index, the index will store a null value for this document. Because of the unique constraint, MongoDB will only permit one document that lacks the indexed field. If there is more than one document without a value for the indexed field or is missing the indexed field, the index build will fail with a duplicate key error.

Práctica

```
> db.clientes.insert({nombre:"Javier", apellidos: "Pérez López"})  
WriteResult({ "nInserted" : 1 })  
> db.clientes.insert({nombre:"Sara", apellidos: "Fernández Gómez"})  
WriteResult({  
  "nInserted" : 0,  
  "writeError" : {  
    "code" : 11000,  
    "errmsg" : "E11000 duplicate key error collection: shop.clientes index:  
dni_1 dup key: { : null }"  
  }  
})
```