

Validación de Esquemas

Bases de Datos 2022

Esquemas Flexibles

- Por defecto los documentos de una colección pueden tener diferentes estructuras
- Cambio en la estructura implica actualizar los documentos a la nueva estructura
- En la práctica los documentos de una colección comparten una estructura similar
- Opcionalmente se pueden arreglar reglas de **validación de esquemas**

Validación de Esquemas

- Permite especificar reglas de validación a los documentos
- La validación de esquema se ejecuta durante los inserts y updates
- Cómo se especifican las reglas de validación
 - Especificación con [Validación de Esquemas JSON](#)
 - [Especificación con Operadores de Selección](#)

Validación de Esquemas JSON

```
db.createCollection( "<name>", { validator: <document>, validationLevel: <string>, validationAction: <string> } )
```

```
db.runCommand( { collMod: "<name>", validator: <document>, validationLevel: <string>, validationAction: <string> } )
```

➤ Crear la colección employees

```
db.createCollection("employees", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: [ "name", "age" ],
      properties: {
        name: {
          bsonType: "string",
          minLength: 3,
          description: "full name of the employee and is required"
        },
        age: {
          bsonType: "int",
          minimum: 16,
          description: "age of the employee and is required"
        },
        category: {
          enum: [ "Full-time", "Part-time", "Temporary" ],
          description: "can only be one of the enum values if the field exists"
        }
      }
    }
  }
})
```

➤ Ejemplo de validación 1

```
db.employees.insertOne( { name: "Kate MacDonell", category: "Full-time" } )
```

MongoServerError: Document failed validation

➤ Ejemplo de validación 2

```
db.employees.insertOne( { name: "Kate MacDonell", age: 15, category: "Part-time" } )
```

MongoServerError: Document failed validation

➤ Ejemplo de validación 3

```
db.employees.insertOne( { name: "Kate MacDonell", age: 21, category: "On-call" } )
```

MongoServerError: Document failed validation

➤ Ejemplo de validación 4

```
db.employees.insertOne( { name: "Kate MacDonell", age: 21, category: "Part-time" } )
```

```
{ acknowledged: true, insertedId: ObjectId("63648dff89f286435339acf3") }
```

Validación de Esquemas JSON - JSONSchema

```
{ $jsonSchema: <JSON Schema object> }
```

```
{  
  $jsonSchema: {  
    <keyword1>: <value1>,  
    <keyword2>: <value2>,  
    <keyword3>: <value3>,  
    ...  
  }  
}
```

Validación de Esquemas JSON - JSONSchema - KEYWORDS

➤ bsonType

- Acepta los mismos alias en string usados por el operador \$type

➤ required

- El documento debe contener todos los elementos especificados en el arreglo

➤ properties

- Un esquema JSON válido donde cada valor es un esquema JSON válido

➤ additionalProperties

- Especifica si campos si se permiten campos adicionales

➤ minimum, maximum

- Indica el valor mínimo (máximo) del campo

➤ minItems, maxItems

- Indica la longitud mínima (longitud máxima) del arreglo

➤ otros keywords

- [enum](#), [description](#), [pattern](#), [minLength](#), [maxLength](#), [uniqueItems](#), [más keywords](#)
- [Tips y buenas prácticas](#)

Validación de Esquemas con Operadores de Selección

- Permite especificar validaciones que comparan múltiples campos
- Se puede combinar validación con operador de selección con validación de esquemas JSON
- Crear la colección orders

```
db.createCollection( "orders",  
  {  
    validator: {  
      $expr: {  
        $eq: [  
          "$totalWithIVA",  
          { $multiply: [ "$total", "$IVA" ] }  
        ]  
      }  
    }  
  }  
)
```

➤ Ejemplo de validación 1

```
db.orders.insertOne( {  
  total: NumberDecimal("4000"),  
  IVA: NumberDecimal("1.21"),  
  totalWithIVA: NumberDecimal("4800")  
})
```

MongoServerError: Document failed validation

➤ Ejemplo de validación 2

```
db.orders.insertOne( {  
  total: NumberDecimal("4000"),  
  IVA: NumberDecimal("1.21"),  
  totalWithIVA: NumberDecimal("4840")  
})
```

```
{ acknowledged: true, insertedId: ObjectId("6364977e89f286435339acf4") }
```

Validación de Esquemas - ValidationLevel - ValidationAction

- **validationLevel** : permiten especificar cómo aplicar las reglas de validación a documentos ya existentes
 - **strict**: (valor por defecto) Las reglas de validación se aplican a todos los inserts y updates.
 - **moderate** : Las reglas de validación solo se aplican a los documentos existentes válidos.
- **validationAction** : permiten especificar cómo manejar los documentos que no cumplen la validación
 - **error**: (valor por defecto) MongoDB rechaza cualquier insert o update que no cumple la regla de validación
 - **warn**: MongoDB permite que operación continúe, pero registra la infracción en los logs de MongoDB

Validación de Esquemas - ValidationLevel - ValidationAction

```
db.contacts.insertMany([
  { "_id": 1, "name": "Anne", "phone": "+1 555 123 456", "city": "London", "status": "Complete" },
  { "_id": 2, "name": "Ivan", "city": "Vancouver" }
])
```

➤ Agregar una validación a una colección existente

```
db.runCommand({
  collMod: "contacts",
  validator: { $jsonSchema: {
    bsonType: "object",
    required: [ "phone", "name" ],
    properties: {
      phone: {
        bsonType: "string",
        description: "phone must be a string and is required"
      },
      name: {
        bsonType: "string",
        description: "name must be a string and is required"
      }
    }
  }
},
  validationLevel: "moderate",
  validationAction: "error"
})
```

➤ Ejemplo de validación 1

```
db.contacts.updateOne(
  { "_id": 1 },
  { $set: { phone: null } }
)
```

MongoServerError: Document failed validation

➤ Ejemplo de validación 2

```
db.contacts.updateOne(
  { "_id": 2 },
  { $set: { phone: null } }
)
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Validación de Esquemas - Metadata

- Mostrar la regla de validación de una colección
 - `db.getCollectionInfos("<collection>")`

Temas a estudiar

- Referencias
 - [Validación de esquemas](#)