Cada consulta incluye una breve explicación para que puedas entender su propósito para aplicarla.

**Taller de Consultas en MongoDB**

**1. Insertar diez Documentos usando insertOne , insertmany**

**Explicación**: Inserta un documento con los campos nombre, edad, ciudad, fecha nacimiento y hobby en la colección.

**2. Encontrar un Documento**

**Explicación**: Encuentra el primer documento que coincida con el campo nombre igual a “John”.

R// db.taller.find({nombre:”John”})

**3. Encontrar Todos los Documentos**

**Explicación**: Recupera todos los documentos de la colección.

R// db.taller.find()

**4. Actualizar un Documento**

**Explicación**: Actualiza el campo edad a 31 en el documento donde nombre es “John”.

R// db.taller.updateOne( { nombre: 'juan' }, { $set: { edad: 31 } })

**5. Eliminar un Documento**

**Explicación**: Elimina el documento donde nombre es “John”.

R// db.taller.deleteOne( { nombre: 'juan' } )

**6. Contar Documentos**

**Explicación**: Cuenta el número total de documentos en la colección.

R//db.taller.countDocuments();

db.taller.countDocuments({ nombre: 'John' });

**7. Encontrar Documentos con una Condición**

**Explicación**: Encuentra documentos donde age es mayor que 25.

R// db.taller.find({ edad: { $gt: 25 }})

**8. Ordenar Documentos**

**Explicación**: Ordena los documentos por edad en orden descendente.

R// db.tuColeccion.find().sort({ edad: -1 });

**9. Limitar el Número de Resultados**

**Explicación**: Limita el resultado a los primeros 5 documentos.

R// db.tuColeccion.find().limit(5);

**10. Proyección de Campos Específicos**

**Explicación**: Muestra solo los campos nombre y edad de cada documento.

R// db.taller.find({}, { nombre: 1, edad: 1, \_id: 0 } )

**11. Buscar Documentos con Múltiples Condiciones**

**Explicación**: Encuentra documentos donde edad es mayor que 25 y ciudad es “New York”.

R// db.taller.find({edad:{$gt: 25},ciudad:'armenia'})

**12. Buscar Documentos con un Campo en un Array**

**Explicación**: Encuentra documentos donde el campo hobbies incluye “lectura” o “deporte”.

R// db.taller.find({ hobby: { $in: ["lectura", "deporte"] } } );

**13. Actualizar Múltiples Documentos**

**Explicación**: Actualiza el campo ciudad a “NYC” en todos los documentos donde ciudad es “New York”.

R// db.taller.updateMany( { ciudad: "New York" }, documentos { $set: { ciudad: "NYC" } }

**14. Eliminar Múltiples Documentos**

**Explicación**: Elimina todos los documentos donde edad es menor que 20.

R// db.taller.deleteMany({edad:{$lt:20}})

**15. Buscar Documentos con un Campo que No Existe**

**Explicación**: Encuentra documentos donde el campo fecha de nacimiento no existe.

R// db.taller.find(

{ canciones: { $exists: false } }

)

**16. Buscar Documentos con un Campo de Tipo Específico**

**Explicación**: Encuentra documentos donde el campo edad es de tipo entero.

R// db.taller.find({

edad: { $type: 'int' }

})

**17. Buscar Documentos con una Expresión Regular**

**Explicación**: Encuentra documentos donde el campo nombre contiene “John” (sin importar mayúsculas/minúsculas).

R// db.taller.find({

nombre: { $regex: /John/i }

})

**18. Buscar Documentos con un Campo Anidado**

**Explicación**: Encuentra documentos donde el campo anidado ciudad es “New York”.

R// db.taller.find({

" direccion.ciudad:" "armenia"

})

**19. Usar Agregaciones para Agrupar Datos**

**Explicación**: Agrupa documentos por ciudad y cuenta el número de documentos en cada grupo donde edad es mayor que 25.

R//db.taller.aggregate([

{ $match: { edad: { $gt: 25 } } },

{ $group: {

\_id: "$ciudad",

count: { $sum: 1 }

}

}

])

**20. Buscar Documentos con un Campo de Fecha**

**Explicación**: Encuentra documentos donde el campo es mayor o igual a la fecha especificada (Busquen una fecha ingresada ).

R// db.taller.find({ createdAt: { $gte: new ISODate("2023-01-01") } });