1. Insertar un alumno mujer con dos materias y padre de apellido "Torres".

bd\_prueba> db.alumnos.insertOne({

... nombre: "María López",

... sexo: "F",

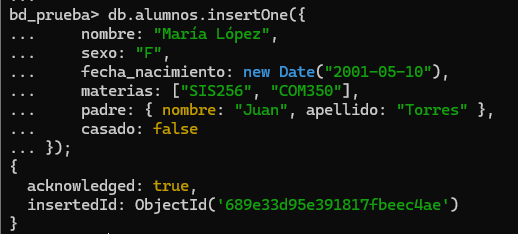
... fecha\_nacimiento: new Date("2001-05-10"),

... materias: ["SIS256", "COM350"],

... padre: { nombre: "Juan", apellido: "Torres" },

... casado: false

... });



1. Insertar cinco alumnos a la vez con datos variados.

db.alumnos.insertMany([

... {

... nombre: "Jorge Ramírez",

... sexo: "M",

... fecha\_nacimiento: new Date("2000-04-12"),

... materias: ["SIS256", "COM350"],

... padre: { nombre: "Alberto", apellido: "Ramírez" },

... casado: false

... },

... {

... nombre: "Lucía Fernández",

... sexo: "F",

... fecha\_nacimiento: new Date("1999-08-19"),

... materias: ["SIS258", "COM350"],

... padre: { nombre: "Miguel", apellido: "Fernández" },

... casado: true

... },

... {

... nombre: "Diego Morales",

... sexo: "M",

... fecha\_nacimiento: new Date("2001-02-05"),

... materias: ["SIS256"],

... padre: { nombre: "Carlos", apellido: "Morales" },

... casado: false

... },

... {

... nombre: "Paula Torres",

... sexo: "F",

... fecha\_nacimiento: new Date("2002-11-30"),

... materias: ["COM350", "SIS258"],

... padre: { nombre: "Pedro", apellido: "Torres" },

... casado: false

... },

... {

... nombre: "Matías Herrera",

... sexo: "M",

... fecha\_nacimiento: new Date("1998-06-18"),

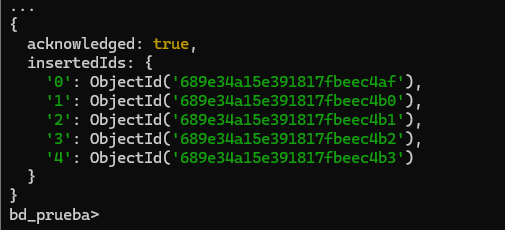
... materias: ["SIS256", "SIS258"],

... padre: { nombre: "Luis", apellido: "Herrera" },

... casado: true

... }

... ]);



1. Insertar un alumno sin campo padre.

db.alumnos.insertOne({

... nombre: "Sofía Castillo",

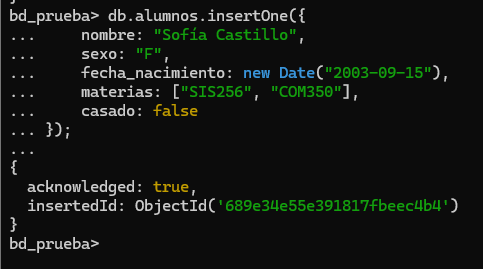
... sexo: "F",

... fecha\_nacimiento: new Date("2003-09-15"),

... materias: ["SIS256", "COM350"],

... casado: false

... });



1. Insertar un alumno hombre casado, nacido en 1990.

db.alumnos.insertOne({

... nombre: "Fernando Díaz",

... sexo: "M",

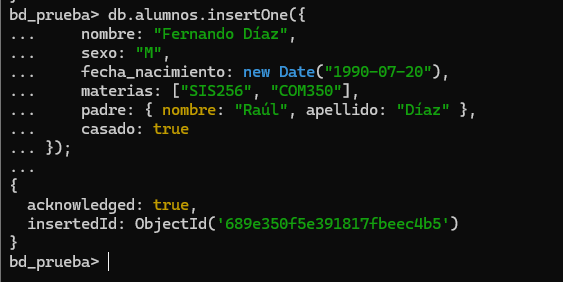
... fecha\_nacimiento: new Date("1990-07-20"),

... materias: ["SIS256", "COM350"],

... padre: { nombre: "Raúl", apellido: "Díaz" },

... casado: true

... });



1. Insertar un alumno con 4 materias y padre "Carlos Pérez".

db.alumnos.insertOne({

... nombre: "Valentina Rojas",

... sexo: "F",

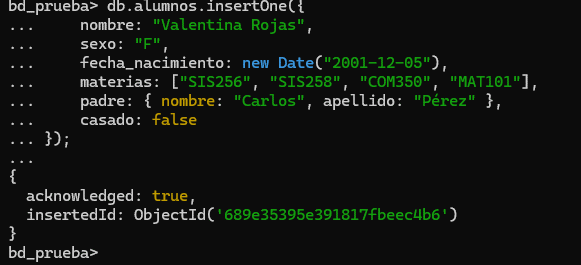
... fecha\_nacimiento: new Date("2001-12-05"),

... materias: ["SIS256", "SIS258", "COM350", "MAT101"],

... padre: { nombre: "Carlos", apellido: "Pérez" },

... casado: false

... });



1. Insertar un alumno solo con "COM350" como materia.

db.alumnos.insertOne({

nombre: "Andrés Salazar",

sexo: "M",

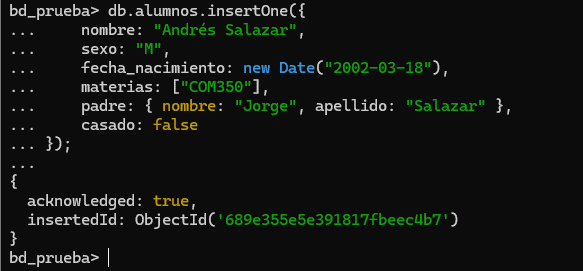
fecha\_nacimiento: new Date("2002-03-18"),

materias: ["COM350"],

padre: { nombre: "Jorge", apellido: "Salazar" },

casado: false

});



1. Listar alumnos cuyo padre tenga nombre "Luis" o "Miguel".

db.alumnos.find({

$or: [

{ "padre.nombre": "Luis" },

{ "padre.nombre": "Miguel" }

]

});



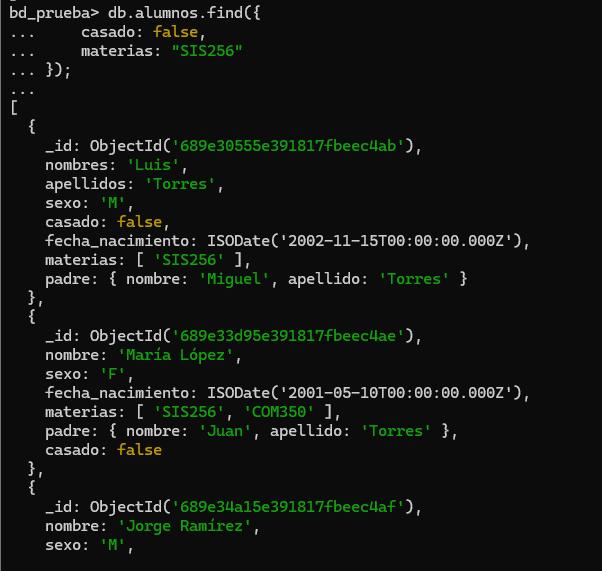
1. Mostrar alumnos no casados que tengan "SIS256".

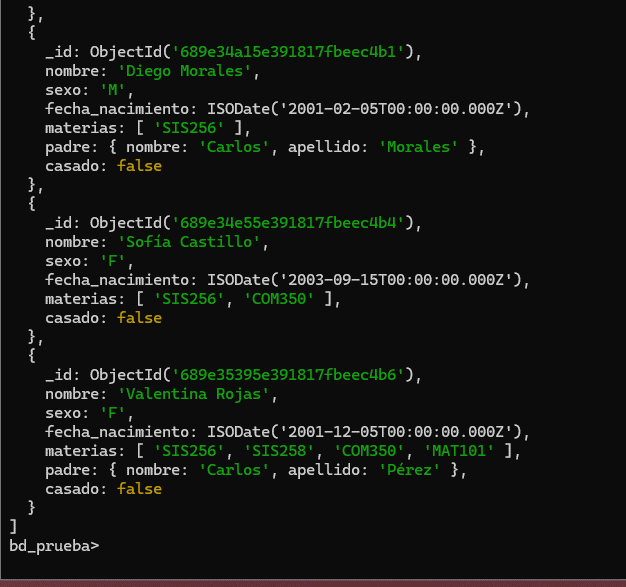
db.alumnos.find({

casado: false,

materias: "SIS256"

});



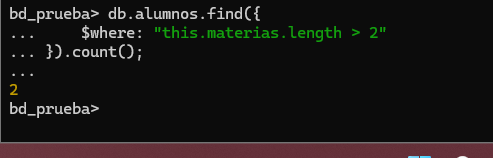


1. Contar alumnos con más de 2 materias.

db.alumnos.find({

$where: "this.materias.length > 2"

}).count();

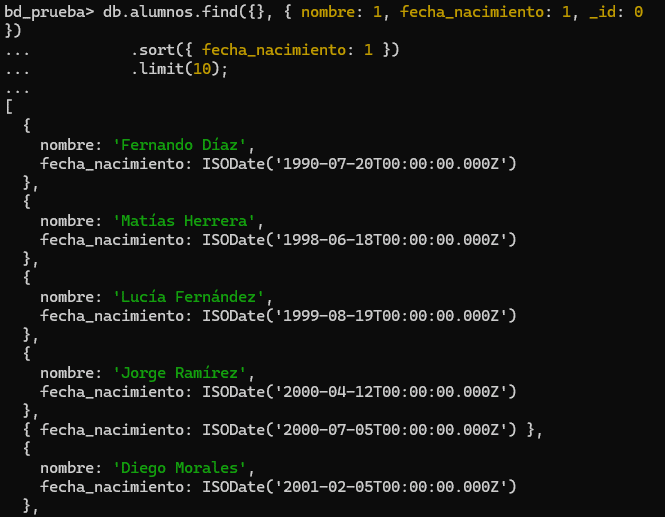


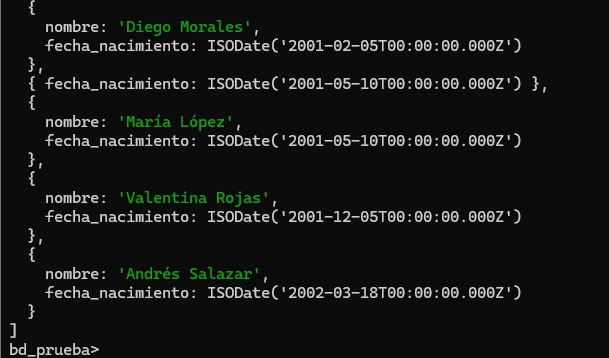
1. Mostrar los 10 más viejos con nombres y fecha\_nacimiento.

db.alumnos.find({}, { nombre: 1, fecha\_nacimiento: 1, \_id: 0 })

.sort({ fecha\_nacimiento: 1 })

.limit(10);



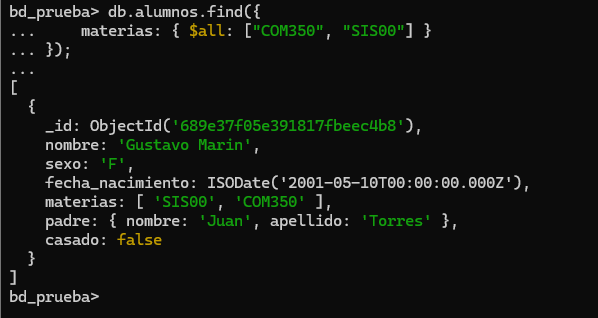


1. Buscar alumnos con "COM350" y "SIS00".

db.alumnos.find({

materias: { $all: ["COM350", "SIS00"] }

});



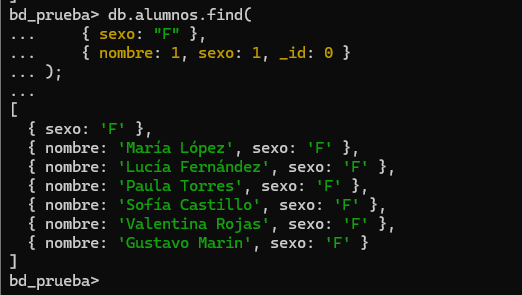
1. Mostrar solo nombres y sexo de los alumnos mujeres.

db.alumnos.find(

{ sexo: "F" },

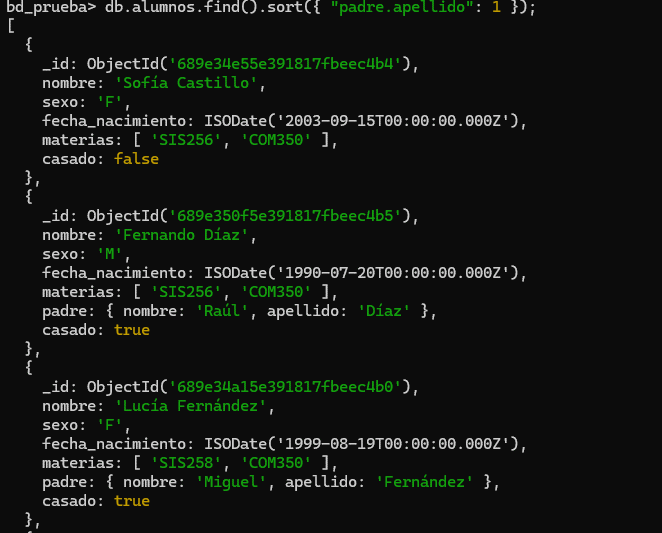
{ nombre: 1, sexo: 1, \_id: 0 }

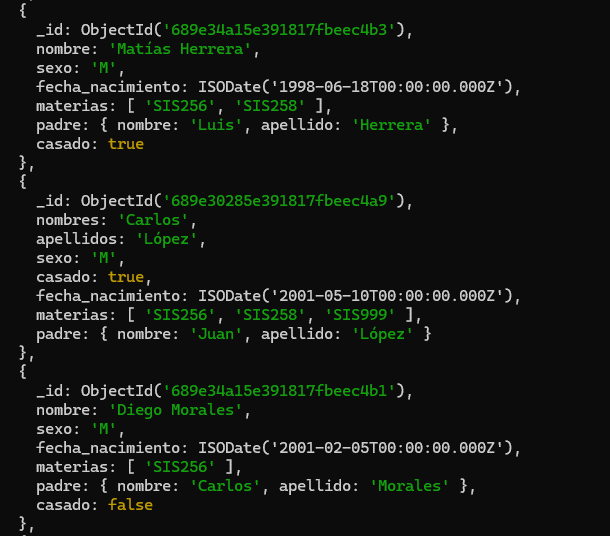
);

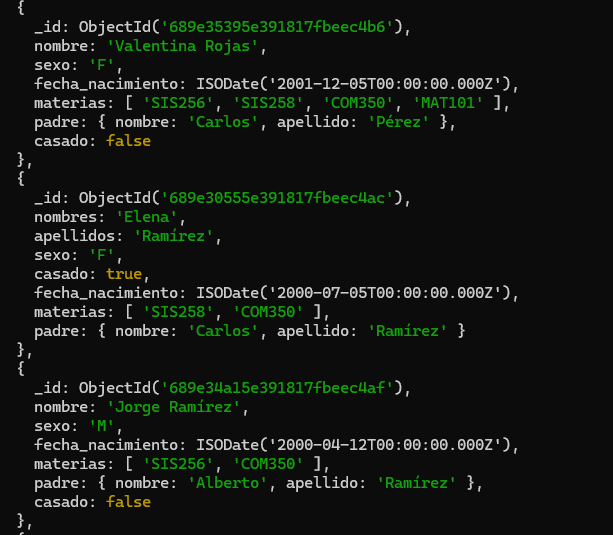


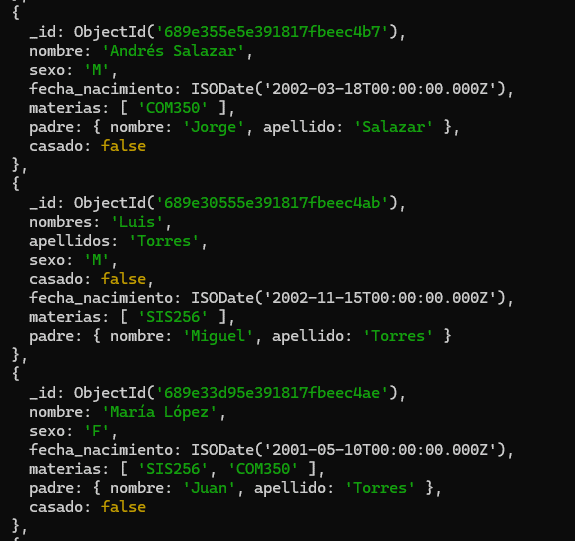
1. Ordenar todos los alumnos por apellido ascendente.

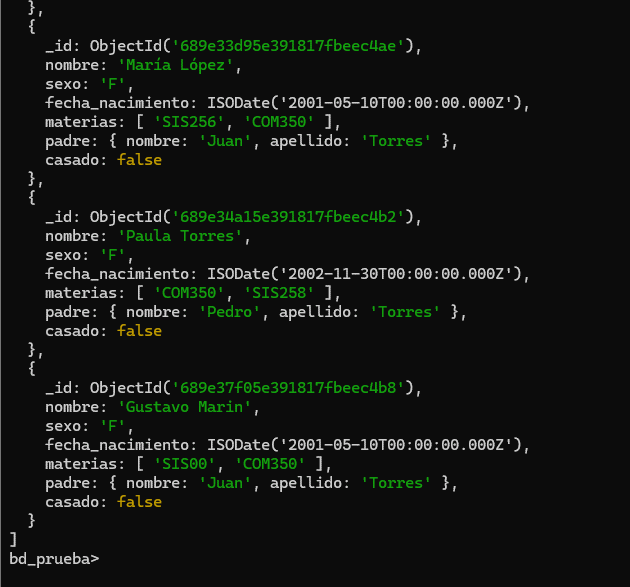
db.alumnos.find().sort({ "padre.apellido": 1 });











1. Listar alumnos nacidos entre 1998 y 2003.

db.alumnos.find({

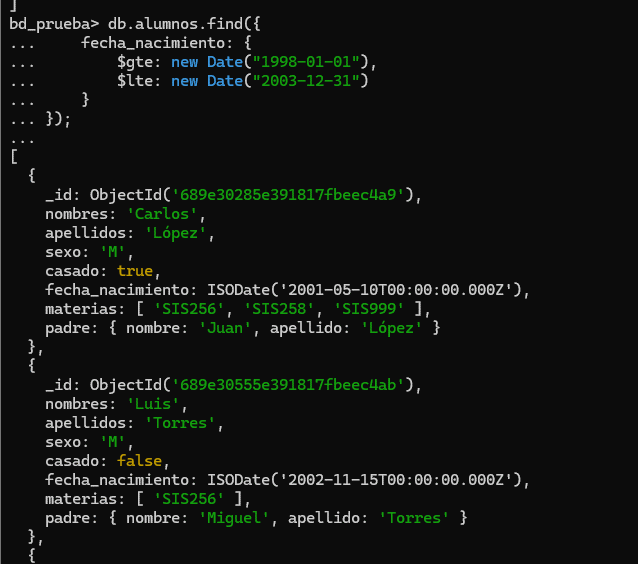
fecha\_nacimiento: {

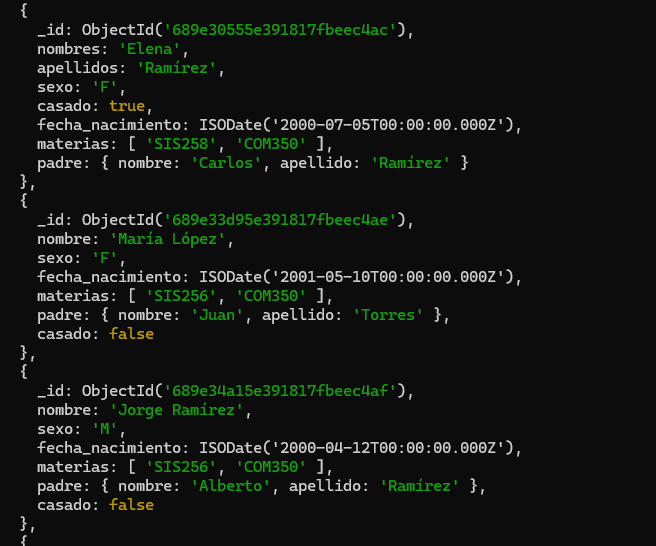
$gte: new Date("1998-01-01"),

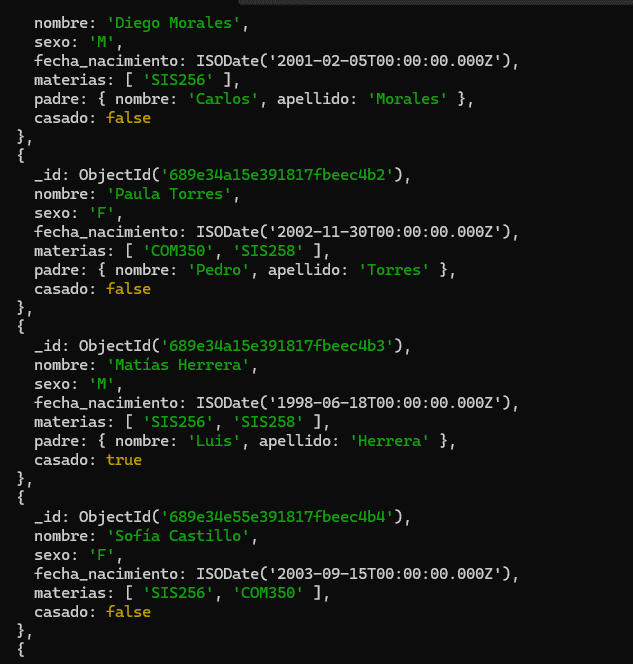
$lte: new Date("2003-12-31")

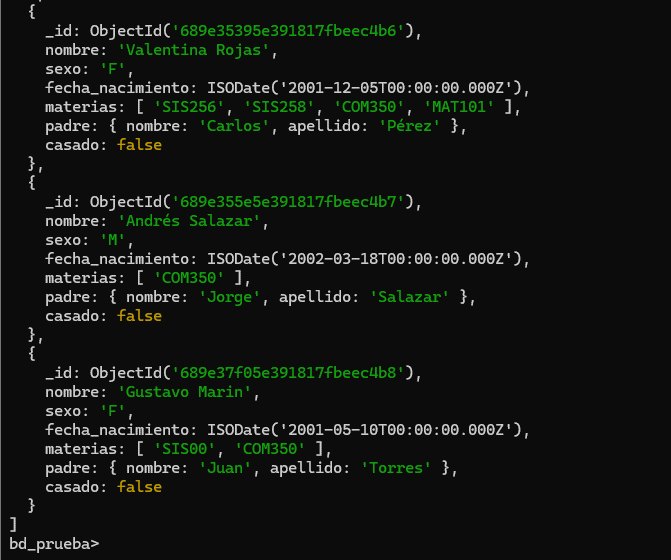
}

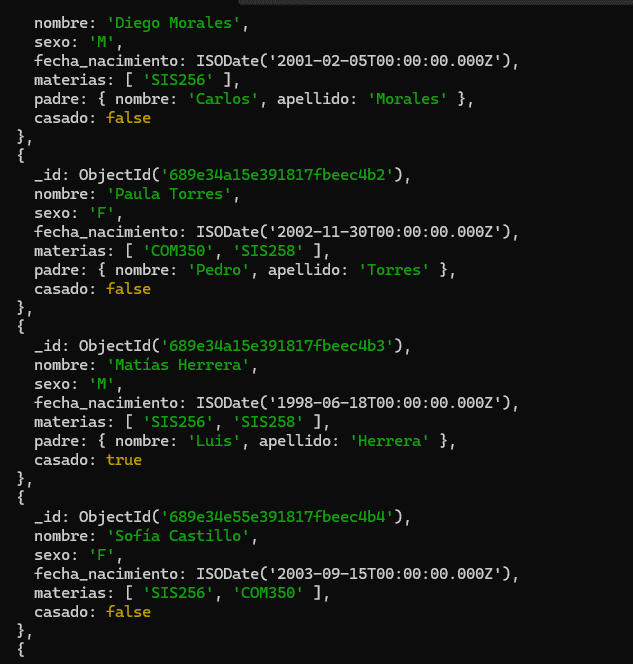
});









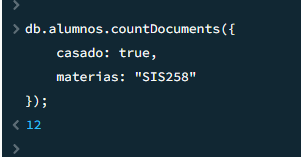


15. Mostrar alumnos cuyo apellido termine en "ez"

**db.alumnos.find({ apellidos: /ez$/ });**

16. Contar alumnos casados y con "SIS258"

**db.alumnos.countDocuments({casado: true, materias: "SIS258"});**

****

17. Cambiar el nombre "Luis" por "Luis Alberto"

**db.alumnos.updateMany({nombres:'Luis'},{$set:{nombres:'Luis Alberto'}})**

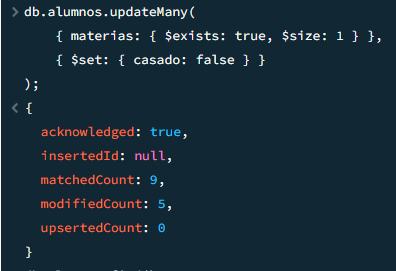
18. Agregar "SIS777" a todos los alumnos con "COM350"

**db.alumnos.updateMany({ materias: "COM350" },{ $addToSet: { materias: "SIS777" } });**

****

19. Poner casado: false a todos los alumnos con menos de 2 materias

**db.alumnos.updateMany({ materias: { $exists: true, $size: 1 } }, { $set: { casado: false } });**

****

20. Quitar "SIS256" de todos los que la tengan

**db.alumnos.updateMany({},{ $pull: { materias: "SIS256" } });**

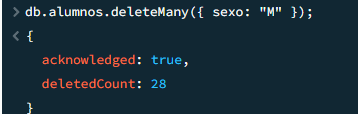
21. Cambiar el nombre del padre "Juan" a "Juan José"

**db.alumnos.updateMany( { "padre.nombre": "Juan" },{ $set: { "padre.nombre": "Juan José" } });**

22. Agregar un campo activo: true a todos los documentos

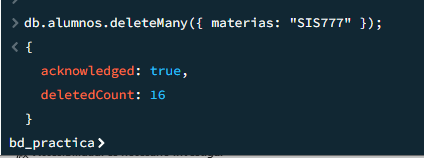
****

23. Eliminar todos los alumnos hombres

**db.alumnos.deleteMany({ sexo: "M" });**

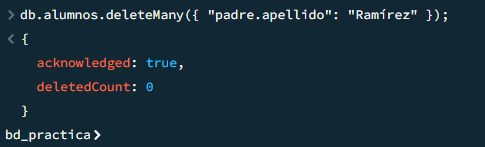
24. Eliminar a quien tenga "SIS777" como materia

**db.alumnos.deleteMany({ materias: "SIS777" });**

****

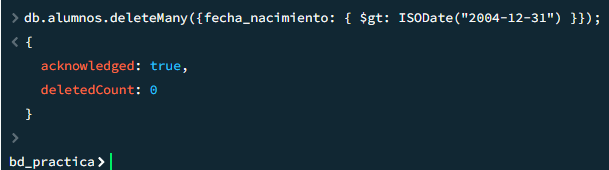
25. Eliminar a los alumnos cuyo padre tenga apellido "Ramírez"

**db.alumnos.deleteMany({ "padre.apellido": "Ramírez" });**

****

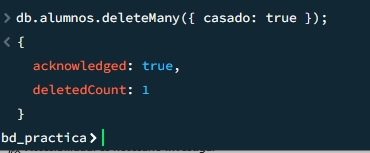
26. Eliminar a los que nacieron después de 2004

**db.alumnos.deleteMany({fecha\_nacimiento: { $gt: ISODate("2004-12-31") }});**

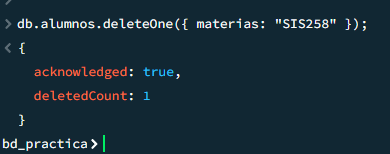
****

27. Eliminar todos los casados

**db.alumnos.deleteMany({ casado: true });**

****

28. Eliminar al primer alumno que tenga "SIS258"

**db.alumnos.deleteOne({ materias: "SIS258" });**