1. Cree una base de datos llamada 'my\_first\_db'.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Crear colección de estudiantes.

Text

Description automatically generated

1. Cada documento que inserte en esta colección debe tener el siguiente formato:({name: STRING, home\_state: STRING, lucky\_number: NUMBER, birthday: {month: NUMBER, day: NUMBER, year: NUMBER}})
2. Crea 5 estudiantes con la información adecuada.

{name: "Lucas", home\_state: "San Jose Dojo", lucky\_number: 2, birthday: {month: 1, day: 25, year: 1986}}

{name: "Marcos", home\_state: "Seattle Dojo", lucky\_number: 5, birthday: {month: 5, day: 15, year: 1985}}

{name: "Santiago", home\_state: "Mountain View Dojo", lucky\_number: 7, birthday: {month: 11, day: 11, year: 1987}}

{name: "Pedro", home\_state: "Seattle Dojo", lucky\_number: 9, birthday: {month: 6, day: 24, year: 1986}}

{name: "Juan", home\_state: "UDLA Dojo", lucky\_number: 3, birthday: {month: 3, day: 8, year: 1988}}

Text

Description automatically generated

1. Consigue todos los estudiantes.

Consulta: db.students.find({}).pretty()

Text

Description automatically generated

1. Recupera a todos los estudiantes que son de California (San Jose Dojo) o Washington (Seattle Dojo).

Consulta: db.students.find({home\_state: {$in: ["San Jose Dojo", "Seattle Dojo"]}}).pretty()

Text

Description automatically generated

1. Obten todos los estudiantes cuyo número de la suerte sea mayor que 3

Consulta: db.students.find({lucky\_number: {$gt: 3}}).pretty()

Text

Description automatically generated

1. Obten todos los estudiantes cuyo número de la suerte sea menor o igual a 10

Consulta: db.students.find({lucky\_number: {$lte: 10}}).pretty()

Text

Description automatically generated

1. Obten todos los estudiantes cuyo número de la suerte esté entre 1 y 9 (inclusive)

Consulta: db.students.find({ $and: [ { lucky\_number: { $gt: 1 } }, { lucky\_number: { $lte: 9 } }] }).pretty()

Text

Description automatically generated

1. Agrega un campo a cada colección de estudiantes llamado 'intereses' que es un ARRAY. Debe contener las siguientes entradas: 'codificación', 'brunch', 'MongoDB'. Haz esto en UNA operación.

Consulta: db.students.updateMany({}, {$set: {interests: ["coding", "brunch", "MongoDB"]}})

Text

Description automatically generated

1. Agrega algunos intereses únicos para cada estudiante en particular en cada una de sus matrices de intereses.

db.students.update({name: "Lucas"}, {$push: {interests: { $each: ["react", "express"] }}})

db.students.find({name: "Lucas"}).pretty()

Text

Description automatically generated

db.students.update({name: "Marcos"}, {$push: {interests: { $each: [".net", "iis"] }}})

db.students.find({name: "Marcos"}).pretty()

Text

Description automatically generated

db.students.update({name: "Santiago"}, {$push: {interests: { $each: ["javascript", "nodejs"] }}})

db.students.find({name: "Santiago"}).pretty()

Text

Description automatically generated

db.students.update({name: "Pedro"}, {$push: {interests: { $each: ["java", "jboss"] }}})

db.students.find({name: "Pedro"}).pretty()

Text

Description automatically generated

db.students.update({name: "Juan"}, {$push: {interests: { $each: ["angular", "express"] }}})

db.students.find({name: "Juan"}).pretty()

Text

Description automatically generated

1. Agrega los 'impuestos' de intereses a la matriz de intereses de alguien.

db.students.update({name: "Lucas"}, {$push: {interests: "impuestos"}})

db.students.find({name: "Lucas"}).pretty()

Text

Description automatically generated

1. Elimina los intereses de 'impuestos' que acaba de agregar.

db.students.update({name: "Lucas"}, {$pop: {interests: 1}})

db.students.find({name: "Lucas"}).pretty()

Text

Description automatically generated

1. Eliminar a todos los estudiantes que son de California.

db.students.remove({home\_state: "San Jose Dojo"})

db.students.find({}).pretty()

Text

Description automatically generated

1. Eliminar a un alumno por su nombre.

db.students.remove({name: "Pedro"})

db.students.find({}).pretty()

Text

Description automatically generated

1. Retira a un estudiante cuyo número de la suerte sea mayor que 5 (SOLO UNO)

db.students.remove({lucky\_number: {$gt: 5}}, true)

db.students.find({}).pretty()

Text

Description automatically generated

1. Agrega un campo a cada colección de estudiantes llamado 'number\_of\_belts' y configúralo en 0.

db.students.updateMany({}, {$set: {number\_of\_belts: 0}})

db.students.find({}).pretty()

Text

Description automatically generated

1. Incrementa este campo en 1 para todos los estudiantes en Washington (Seattle Dojo).

db.students.updateMany({home\_state: "Seattle Dojo"}, {$inc: {number\_of\_belts: 1}})

db.students.find({}).pretty()

Text

Description automatically generated

1. Cambia el nombre del campo 'number\_of\_belts' a 'belts\_earned'

db.students.updateMany( {}, { $rename: { "number\_of\_belts": "belts\_earned" } } )

db.students.find({}).pretty()

Text

Description automatically generated

1. Elimina el campo 'lucky\_number'.

db.students.updateMany( {}, { $unset: { "lucky\_number": "" } } )

db.students.find({}).pretty()

Text

Description automatically generated

1. Agrega un campo 'updated\_on' y establece el valor como la fecha actual.

db.students.updateMany( {}, { $set: { "updated\_on": new Date() } } )

db.students.find({}).pretty()

Text

Description automatically generated