**P5 - MONGODB**

# **APARTADO 2.- Utilizar la colección aficiones para relacionar personas como en una Red Social**

1. db.aficiones.count() -> 89
2. Crear tus propios contenidos de la BD

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Fish",

"Puntuacion": 9.0,

"Precio": 450.0,

"Nivel": "Intermedio",

"Tamano": 6

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Longboard",

"Puntuacion": 9.5,

"Precio": 800.0,

"Nivel": "Intermedio",

"Tamano": 10

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Shortboard",

"Puntuacion": 8.5,

"Precio": 700.0,

"Nivel": "Pro",

"Tamano": 5.4

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Funboard",

"Puntuacion": 6.0,

"Precio": 200.0,

"Nivel": "Debuntante",

"Tamano": 7

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Gun",

"Puntuacion": 8.0,

"Precio": 900.0,

"Nivel": "Pro",

"Tamano": 9

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Hybrid",

"Puntuacion": 8.5,

"Precio": 500.0,

"Nivel": "Intermedio",

"Tamano": 7

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "StandUp",

"Puntuacion": 7.0,

"Precio": 300.0,

"Nivel": "Todos",

"Tamano": 10

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Malibu",

"Puntuacion": 8.0,

"Precio": 800.0,

"Nivel": "Intermedio",

"Tamano": 7.2

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "Bodyboard",

"Puntuacion": 5.0,

"Precio": 50.0,

"Nivel": "Principiante",

"Tamano": 10

})

db.aficiones.insert({

"Tema": "Surf",

"Apodo": "Helo",

"Nombre": "SurfSkate",

"Puntuacion": 7.0,

"Precio": 250.0,

"Nivel": "Todos",

"Tamano": 3

})

1. Probar consultas

db.aficiones.count() -> 99

db.aficiones.findOne({"Tema": "Surf" });

consulta = db.aficiones.findOne({"Tema": "Surf" });

print(consulta);

{

"\_id" : ObjectId("644a463096f860da97151cb7"),

"Tema" : "Surf",

"Apodo" : "Helo",

"Nombre" : "Fish",

"Puntuacion" : 9,

"Precio" : 450,

"Nivel" : "Intermedio",

"Tamano" : 6

}

[object BSON]

# **APARTADO 3.- Trabajando con la colección Aficiones en MongoDB**

1. Consultas: (componente es cada documento de la BD)

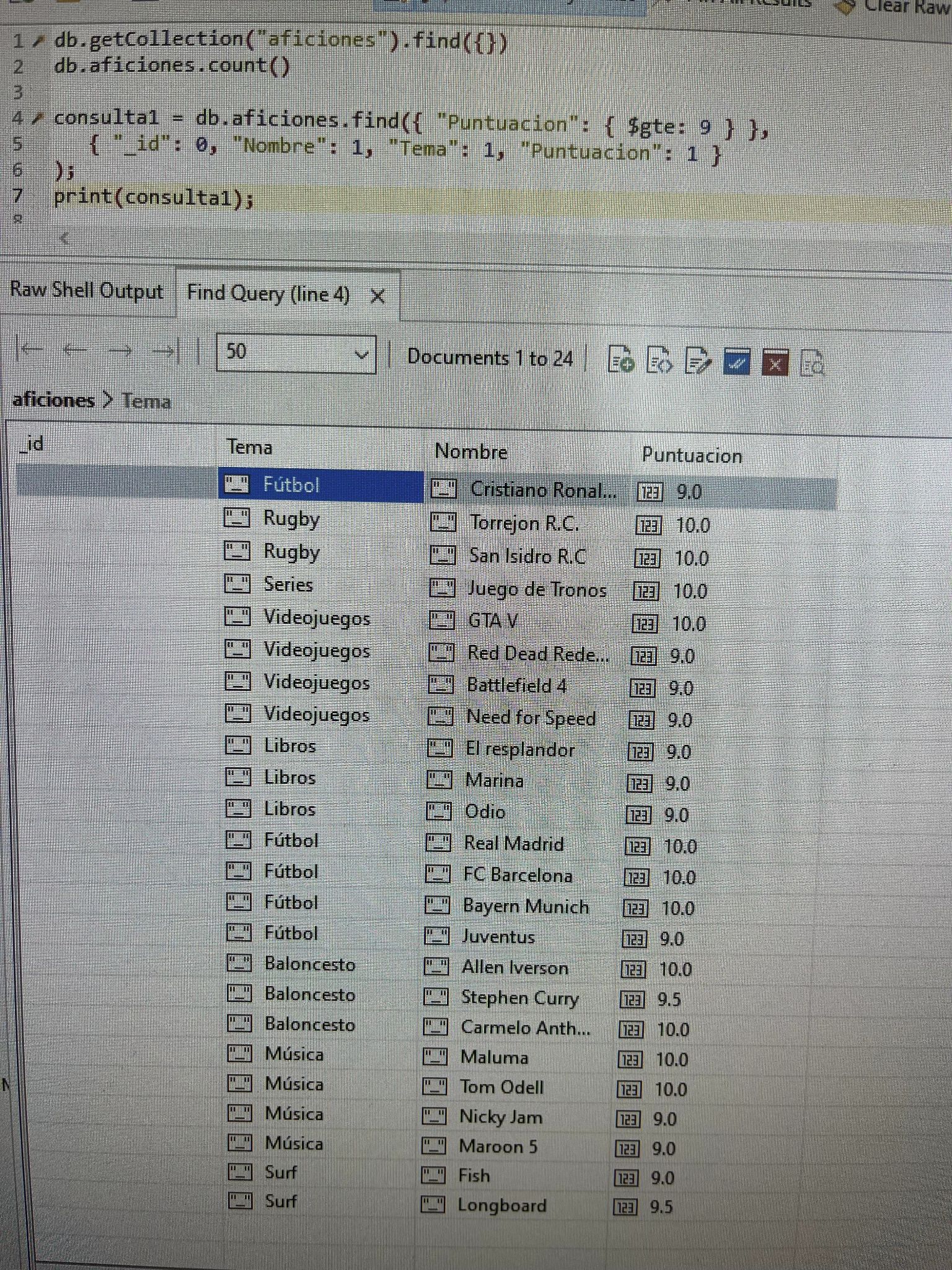
1. Obtener el nombre, el tema y la puntuación de los componentes mejor valorados (puntuaciones > = 9) pero no saques el identificador.

consulta1 = db.aficiones.find({ "Puntuacion": { $gte: 9 } },

{ "\_id": 0, "Nombre": 1, "Tema": 1, "Puntuacion": 1 }

);

print(consulta1);



2. Obtener para cada tema (agrupado): su nombre, el total de cuanto te gastarías si vas a todos los componentes mejor valorados y cuántos componentes has incluido.

consulta2 = db.aficiones.aggregate([{$match:{Puntuacion:{ $gte: 9}}},{$group:{\_id:

"$Tema",totalGasto:{$sum:"$Precio"},totalComponentes:{$sum:1}}}]) ;

print(consulta2);

{ "\_id" : "Música", "totalGasto" : 50.97, "totalComponentes" : 4 }

{ "\_id" : "Baloncesto", "totalGasto" : 135000000, "totalComponentes" : 3 }

{ "\_id" : "Surf", "totalGasto" : 1250, "totalComponentes" : 2 }

{ "\_id" : "Fútbol", "totalGasto" : 100003300, "totalComponentes" : 5 }

{ "\_id" : "Libros", "totalGasto" : 28.4, "totalComponentes" : 3 }

{ "\_id" : "Rugby", "totalGasto" : 605, "totalComponentes" : 2 }

{ "\_id" : "Series", "totalGasto" : 9.95, "totalComponentes" : 1 }

{ "\_id" : "Videojuegos", "totalGasto" : 179.8, "totalComponentes" : 4 }

3. para cada una de estas puntuaciones por separado: 10, 9, 8, 7, 6 y 5. Para cada puntuación, obtener los nombres de componentes que la tienen. (cuando ya funcione, incluye también el nombre su tema)

consulta3 = db.aficiones.aggregate([{$match:{Puntuacion:{$in:[10,9,8,7,6,5]}}},{$group:{\_id:"$Puntuacion",Componentes:{$push:{Nombre:"$Nombre",Tema:"$Tema"}}}},{$sort:{\_id:-1}}]);

print(consulta3);

{ "\_id" : 10, "Componentes" : [ { "Nombre" : "Torrejon R.C.", "Tema" : "Rugby" }, { "Nombre" : "San Isidro R.C", "Tema" : "Rugby" }, { "Nombre" : "Juego de Tronos", "Tema" : "Series" }, { "Nombre" : "GTA V", "Tema" : "Videojuegos" }, { "Nombre" : "Real Madrid", "Tema" : "Fútbol" }, { "Nombre" : "FC Barcelona", "Tema" : "Fútbol" }, { "Nombre" : "Bayern Munich", "Tema" : "Fútbol" }, { "Nombre" : "Allen Iverson", "Tema" : "Baloncesto" }, { "Nombre" : "Carmelo Anthony", "Tema" : "Baloncesto" }, { "Nombre" : "Maluma", "Tema" : "Música" }, { "Nombre" : "Tom Odell", "Tema" : "Música" } ] }

{ "\_id" : 9, "Componentes" : [ { "Nombre" : "Cristiano Ronaldo", "Tema" : "Fútbol" }, { "Nombre" : "Red Dead Redemption", "Tema" : "Videojuegos" }, { "Nombre" : "Battlefield 4", "Tema" : "Videojuegos" }, { "Nombre" : "Need for Speed", "Tema" : "Videojuegos" }, { "Nombre" : "El resplandor", "Tema" : "Libros" }, { "Nombre" : "Marina", "Tema" : "Libros" }, { "Nombre" : "Odio", "Tema" : "Libros" }, { "Nombre" : "Juventus", "Tema" : "Fútbol" }, { "Nombre" : "Nicky Jam", "Tema" : "Música" }, { "Nombre" : "Maroon 5", "Tema" : "Música" }, { "Nombre" : "Fish", "Tema" : "Surf" } ] }

{ "\_id" : 8, "Componentes" : [ { "Nombre" : "Al filo de las sombras", "Tema" : "Libros" }, { "Nombre" : "El ojo fragmentado", "Tema" : "Libros" }, { "Nombre" : "Sergio Ramos", "Tema" : "Fútbol" }, { "Nombre" : "Rugby Guadalajara", "Tema" : "Rugby" }, { "Nombre" : "Rome II Total War", "Tema" : "Videojuegos" }, { "Nombre" : "Mafia II", "Tema" : "Videojuegos" }, { "Nombre" : "La llamada Cthulhu", "Tema" : "Libros" }, { "Nombre" : "Se lo que estas pensando", "Tema" : "Libros" }, { "Nombre" : "Manchester United", "Tema" : "Fútbol" }, { "Nombre" : "Paris Saint-Germain", "Tema" : "Fútbol" }, { "Nombre" : "Machester City", "Tema" : "Fútbol" }, { "Nombre" : "Muse", "Tema" : "Música" }, { "Nombre" : "Gun", "Tema" : "Surf" }, { "Nombre" : "Malibu", "Tema" : "Surf" } ] }

{ "\_id" : 7, "Componentes" : [ { "Nombre" : "Rivas R.C", "Tema" : "Rugby" }, { "Nombre" : "XV Hortaleza", "Tema" : "Rugby" }, { "Nombre" : "Olimpico Rugby Club", "Tema" : "Rugby" }, { "Nombre" : "Call Of Duty Black Ops 3", "Tema" : "Videojuegos" }, { "Nombre" : "World of Warships", "Tema" : "Videojuegos" }, { "Nombre" : "Uncharted 4", "Tema" : "Videojuegos" }, { "Nombre" : "Los juegos del hambre 1", "Tema" : "Libros" }, { "Nombre" : "Leicester City", "Tema" : "Fútbol" }, { "Nombre" : "StandUp", "Tema" : "Surf" }, { "Nombre" : "SurfSkate", "Tema" : "Surf" } ] }

{ "\_id" : 6, "Componentes" : [ { "Nombre" : "Medal of Honor Warfighter", "Tema" : "Videojuegos" }, { "Nombre" : "La sombra del viento", "Tema" : "Libros" }, { "Nombre" : "Harry Potter: La piedra filosofal", "Tema" : "Libros" }, { "Nombre" : "Cafe Quijano", "Tema" : "Música" }, { "Nombre" : "Funboard", "Tema" : "Surf" } ] }

{ "\_id" : 5, "Componentes" : [ { "Nombre" : "Quijote Rugby", "Tema" : "Rugby" }, { "Nombre" : "Rayo Vallecano", "Tema" : "Fútbol" }, { "Nombre" : "Katy Perry", "Tema" : "Música" }, { "Nombre" : "Bodyboard", "Tema" : "Surf" } ] }

The find query will be run with Query Assist.

99

4. Lista de Apodos para cada afición (Tema)

consulta4 = db.aficiones.aggregate([{$group:{\_id:"$Tema",Apodos:{$addToSet:"$Apodo"}}}]);

print(consulta4);

{ "\_id" : "Baloncesto", "Apodos" : [ "NBA" ] }

{ "\_id" : "Música", "Apodos" : [ "dios" ] }

{ "\_id" : "MotoGP", "Apodos" : [ "MGP" ] }

{ "\_id" : "Surf", "Apodos" : [ "Helo" ] }

{ "\_id" : "Fútbol", "Apodos" : [ "Mike", "soFIFA" ] }

{ "\_id" : "Rugby", "Apodos" : [ "Infor" ] }

{ "\_id" : "Libros", "Apodos" : [ "Ceor", "Turre" ] }

{ "\_id" : "Series", "Apodos" : [ "Tyrion Lanister" ] }

{ "\_id" : "Videojuegos", "Apodos" : [ "Giorgio" ] }

{ "\_id" : "Ajedrez", "Apodos" : [ "Bobby" ] }

1. (para nota) A quien le gusta los mismos componentes? -> Para cada Tema, queremos saber qué Apodos tienen un mismo componente, es decir el mismo Nombre (ej,: el mismo nombre de libro, CD, equipo de futbol). Los atributos a mostrar de cada componente son: Apodo, Nombre y Tema. Para hacer pruebas inserta algún apodo y tema con componentes que ya existan en la BD.

consultab = db.aficiones.aggregate([{$group:{\_id:"$Tema",componentes:{$addToSet:{Tema:"$Tema",Apodo :"$Apodo",Nombre:"$Nombre"}}}}]);

print(consultab);

{ "\_id" : "Baloncesto", "componentes" : [ { "Tema" : "Baloncesto", "Apodo" : "NBA", "Nombre" : "Allen Iverson" }, { "Tema" : "Baloncesto", "Apodo" : "NBA", "Nombre" : "Kobe Bryant" }, { "Tema" : "Baloncesto", "Apodo" : "NBA", "Nombre" : "Carmelo Anthony" }, { "Tema" : "Baloncesto", "Apodo" : "NBA", "Nombre" : "Stephen Curry" } ] }

{ "\_id" : "Música", "componentes" : [ { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Maluma" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Tom Odell" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Abraham Mateo" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Maroon 5" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Muse" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Drake" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Cafe Quijano" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Nicky Jam" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Kiko Rivera" }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Katy Perry" } ] }

{ "\_id" : "MotoGP", "componentes" : [ { "Tema" : "MotoGP", "Apodo" : "MGP" } ] }

{ "\_id" : "Surf", "componentes" : [ { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Bodyboard" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Fish" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "StandUp" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Longboard" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Funboard" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "SurfSkate" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Shortboard" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Gun" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Malibu" }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Hybrid" } ] }

{ "\_id" : "Fútbol", "componentes" : [ { "Tema" : "Fútbol", "Apodo" : "Mike", "Nombre" : "Sergio Ramos" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Leicester City" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "FC Barcelona" }, { "Tema" : "Fútbol", "Apodo" : "Mike", "Nombre" : "Isco" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Real Madrid" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Bayern Munich" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Manchester United" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Paris Saint-Germain" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Machester City" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Juventus" }, { "Tema" : "Fútbol", "Apodo" : "Mike", "Nombre" : "Cristiano Ronaldo" }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Rayo Vallecano" } ] }

{ "\_id" : "Libros", "componentes" : [ { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El nombre del viento" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "Mas allá de las sombras" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El camino de las sombras" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "La llamada Cthulhu" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "El resplandor" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El temor del hombre sabio" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "Al filo de las sombras" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El ojo fragmentado" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Harry Potter: La piedra filosofal" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Odio" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El prisma negro" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "La daga de la ceguera" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Los juegos del hambre 1" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "La sombra del viento" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Marina" }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Se lo que estas pensando" }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "Corazón inquieto: la vida de San Agustin" } ] }

{ "\_id" : "Rugby", "componentes" : [ { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "CAU Metropolitano" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Quijote Rugby" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Club de Rugby Ingenieros Industriales Las Rozas" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Olimpico Rugby Club" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Rugby Guadalajara" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Rivas R.C" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Torrejon R.C." }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "San Isidro R.C" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "XV Hortaleza" }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Alcala R.C" } ] }

{ "\_id" : "Videojuegos", "componentes" : [ { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Battlefield 4" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Call Of Duty Black Ops 3" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Medal of Honor Warfighter" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Red Dead Redemption" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Need for Speed" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Rome II Total War" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Mafia II" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "World of Warships" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Uncharted 4" }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "GTA V" } ] }

{ "\_id" : "Series", "componentes" : [ { "Tema" : "Series", "Apodo" : "Tyrion Lanister", "Nombre" : "Juego de Tronos" }, { "Tema" : "Series", "Apodo" : "Tyrion Lanister", "Nombre" : "Cuentame" }, { "Tema" : "Series", "Apodo" : "Tyrion Lanister", "Nombre" : "Arrow" }, { "Tema" : "Series", "Apodo" : "Tyrion Lanister", "Nombre" : "Waliking Dead" } ] }

{ "\_id" : "Ajedrez", "componentes" : [ { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Alexander" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Mikhail" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Veselin" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Wilhelm" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Paul" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Bobby" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "José" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Magnus" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Garry" }, { "Tema" : "Ajedrez", "Apodo" : "Bobby", "Nombre" : "Anatoly" } ] }

1. (para nota) Repite la búsqueda anterior para puntuaciones intermedias: más de 4 y menos de 9. Muestra la puntuación también.

consultac = db.aficiones.aggregate([{$match:{"Puntuacion":{$gt:4,$lt:9}}},{$group:{\_id:"$Tema",componentes:{$addToSet:{Tema:"$Tema",Apodo:"$Apodo",Nombre:"$Nombre",Puntuacion:"$Puntuacion"}}}}]);

print(consultac);

{ "\_id" : "Música", "componentes" : [ { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Cafe Quijano", "Puntuacion" : 6 }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Muse", "Puntuacion" : 8 }, { "Tema" : "Música", "Apodo" : "dios", "Nombre" : "Katy Perry", "Puntuacion" : 5 } ] }

{ "\_id" : "Series", "componentes" : [ { "Tema" : "Series", "Apodo" : "Tyrion Lanister", "Nombre" : "Arrow", "Puntuacion" : 7.25 } ] }

{ "\_id" : "Videojuegos", "componentes" : [ { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Call Of Duty Black Ops 3", "Puntuacion" : 7 }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Mafia II", "Puntuacion" : 8 }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Rome II Total War", "Puntuacion" : 8 }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Uncharted 4", "Puntuacion" : 7 }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "World of Warships", "Puntuacion" : 7 }, { "Tema" : "Videojuegos", "Apodo" : "Giorgio", "Nombre" : "Medal of Honor Warfighter", "Puntuacion" : 6 } ] }

{ "\_id" : "Libros", "componentes" : [ { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El temor del hombre sabio", "Puntuacion" : 8.2 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "Al filo de las sombras", "Puntuacion" : 8 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "Corazón inquieto: la vida de San Agustin", "Puntuacion" : 8.4 }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "La sombra del viento", "Puntuacion" : 6 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El camino de las sombras", "Puntuacion" : 8.3 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "Mas allá de las sombras", "Puntuacion" : 8.3 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "La daga de la ceguera", "Puntuacion" : 7.4 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El ojo fragmentado", "Puntuacion" : 8 }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Harry Potter: La piedra filosofal", "Puntuacion" : 6 }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "La llamada Cthulhu", "Puntuacion" : 8 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El prisma negro", "Puntuacion" : 7.1 }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Se lo que estas pensando", "Puntuacion" : 8 }, { "Tema" : "Libros", "Apodo" : "Ceor", "Nombre" : "Los juegos del hambre 1", "Puntuacion" : 7 }, { "Tema" : "Libros", "Apodo" : "Turre", "Nombre" : "El nombre del viento", "Puntuacion" : 7.9 } ] }

{ "\_id" : "Rugby", "componentes" : [ { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Quijote Rugby", "Puntuacion" : 5 }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "XV Hortaleza", "Puntuacion" : 7 }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Rugby Guadalajara", "Puntuacion" : 8 }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Rivas R.C", "Puntuacion" : 7 }, { "Tema" : "Rugby", "Apodo" : "Infor", "Nombre" : "Olimpico Rugby Club", "Puntuacion" : 7 } ] }

{ "\_id" : "Fútbol", "componentes" : [ { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Leicester City", "Puntuacion" : 7 }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Manchester United", "Puntuacion" : 8 }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Paris Saint-Germain", "Puntuacion" : 8 }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Machester City", "Puntuacion" : 8 }, { "Tema" : "Fútbol", "Apodo" : "Mike", "Nombre" : "Sergio Ramos", "Puntuacion" : 8 }, { "Tema" : "Fútbol", "Apodo" : "soFIFA", "Nombre" : "Rayo Vallecano", "Puntuacion" : 5 } ] }

{ "\_id" : "Surf", "componentes" : [ { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "StandUp", "Puntuacion" : 7 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Bodyboard", "Puntuacion" : 5 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Gun", "Puntuacion" : 8 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Shortboard", "Puntuacion" : 8.5 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Hybrid", "Puntuacion" : 8.5 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Funboard", "Puntuacion" : 6 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "Malibu", "Puntuacion" : 8 }, { "Tema" : "Surf", "Apodo" : "Helo", "Nombre" : "SurfSkate", "Puntuacion" : 7 } ] }

1. Describe al menos cuatro consultas interesantes para tí, descríbela primero el texto y luego su código y ejecútalas.

Consultad1:

Me gusta también leer y quiero saber cuales son los mejores libros (puntuación >= 9) de la base de datos: con el nombre del libro, el autor, el género y la puntuación asociada.

consultad1 = db.aficiones.find({ "Tema": "Libros", "Puntuacion": { $gte: 9 } }, { "Nombre": 1, "Autor": 1, "Genero": 1, "Puntuacion": 1, "\_id": 0 });

{ "Nombre" : "El resplandor", "Autor" : "Stephen King", "Puntuacion" : 9, "Genero" : "Terror" }

{ "Nombre" : "Marina", "Autor" : "Carlos Ruiz Zafon", "Puntuacion" : 9, "Genero" : "Thriller" }

{ "Nombre" : "Odio", "Autor" : "David Moody", "Puntuacion" : 9, "Genero" : "Ciencia Ficcion" }

Consultad2:

Me gusta descubrir nuevos ámbitos así que quiero tener una recomendación con el mejor componente de cada Tema. (He añadido un filtrado para que no aparezcan los componentes con campos nulos).

consultad2 = db.aficiones.aggregate([{ $match: { “Nombre”: { $ne: null }, “Puntuacion”: { $ne: null } } }, { $sort: { “Tema”: 1, “Puntuacion”: -1 } }, { $group: { \_id: "$Tema", “Nombre”: { $first: "$Nombre" }, “Puntuacion”: { $first: "$Puntuacion" } } }]);

{ "\_id" : "Música", "Nombre" : "Maluma", "Puntuacion" : 10 }

{ "\_id" : "Baloncesto", "Nombre" : "Allen Iverson", "Puntuacion" : 10 }

{ "\_id" : "Series", "Nombre" : "Juego de Tronos", "Puntuacion" : 10 }

{ "\_id" : "Videojuegos", "Nombre" : "GTA V", "Puntuacion" : 10 }

{ "\_id" : "Libros", "Nombre" : "El resplandor", "Puntuacion" : 9 }

{ "\_id" : "Fútbol", "Nombre" : "Real Madrid", "Puntuacion" : 10 }

{ "\_id" : "Rugby", "Nombre" : "Torrejon R.C.", "Puntuacion" : 10 }

{ "\_id" : "Surf", "Nombre" : "Longboard", "Puntuacion" : 9.5 }

Consultad3:

Me gustaria aprender más sobre el fútbol entonces quiero ver lo más destacado. Por eso quiero obtener los componentes del tema “fútbol” con una puntuación mayor o igual a 8, y solo mostrar el nombre y la puntuación de los componentes.

consultad3 = db.aficiones.find({ "Tema": "Fútbol", "Puntuacion": { $gte: 8 } }, { "Nombre": 1, "Puntuacion": 1, "\_id": 0 });

{ "Nombre" : "Cristiano Ronaldo", "Puntuacion" : 9 }

{ "Nombre" : "Sergio Ramos", "Puntuacion" : 8 }

{ "Nombre" : "Real Madrid", "Puntuacion" : 10 }

{ "Nombre" : "FC Barcelona", "Puntuacion" : 10 }

{ "Nombre" : "Bayern Munich", "Puntuacion" : 10 }

{ "Nombre" : "Manchester United", "Puntuacion" : 8 }

{ "Nombre" : "Juventus", "Puntuacion" : 9 }

{ "Nombre" : "Paris Saint-Germain", "Puntuacion" : 8 }

{ "Nombre" : "Machester City", "Puntuacion" : 8 }

Consultad4:

Quiero saber cuáles son los temas en la base de datos, junto con la cantidad de componentes y la puntuación promedio de los mismos.

consultad4 = db.aficiones.aggregate([{ $group: { \_id: "$Tema", "Cantidad": { $sum: 1 }, "PuntuacionPromedio": { $avg: "$Puntuacion" } } }]);

{ "\_id" : "Música", "Cantidad" : 10, "PuntuacionPromedio" : 6.3 }

{ "\_id" : "Libros", "Cantidad" : 17, "PuntuacionPromedio" : 7.858823529411764 }

{ "\_id" : "Videojuegos", "Cantidad" : 10, "PuntuacionPromedio" : 8 }

{ "\_id" : "Rugby", "Cantidad" : 10, "PuntuacionPromedio" : 6.2 }

{ "\_id" : "MotoGP", "Cantidad" : 11, "PuntuacionPromedio" : null }

{ "\_id" : "Series", "Cantidad" : 4, "PuntuacionPromedio" : 6.75 }

{ "\_id" : "Ajedrez", "Cantidad" : 11, "PuntuacionPromedio" : null }

{ "\_id" : "Baloncesto", "Cantidad" : 4, "PuntuacionPromedio" : 9.5 }

{ "\_id" : "Fútbol", "Cantidad" : 12, "PuntuacionPromedio" : 8.363636363636363 }

{ "\_id" : "Surf", "Cantidad" : 10, "PuntuacionPromedio" : 7.65 }

1. Obtener todos los componentes de tu colección clasificados por tema.

e.1) Queremos imprimir una línea para cada documento de este modo:

TEMA: <su nombre> - NombreDoc: <del documento>.

Usa un cursor que llama a una función sin nombre (definida dentro del cursor). Esa función imprime cada línea.

var cursorE= db.aficiones.find();

while(cursorE.hasNext()){

doc = cursorE.next();

print("TEMA: " + doc.Tema + " - NombreDoc: " + doc.Nombre + ".");

}

TEMA: Libros - NombreDoc: El temor del hombre sabio.

TEMA: Libros - NombreDoc: El nombre del viento.

TEMA: Libros - NombreDoc: El camino de las sombras.

TEMA: Libros - NombreDoc: Mas allá de las sombras.

TEMA: Libros - NombreDoc: Al filo de las sombras.

TEMA: Libros - NombreDoc: El prisma negro.

TEMA: Libros - NombreDoc: La daga de la ceguera.

TEMA: Libros - NombreDoc: El ojo fragmentado.

TEMA: Libros - NombreDoc: Corazón inquieto: la vida de San Agustin.

TEMA: Fútbol - NombreDoc: Cristiano Ronaldo.

TEMA: Fútbol - NombreDoc: Isco.

TEMA: Fútbol - NombreDoc: Sergio Ramos.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: MotoGP - NombreDoc: undefined.

TEMA: Rugby - NombreDoc: Torrejon R.C..

TEMA: Rugby - NombreDoc: San Isidro R.C.

TEMA: Rugby - NombreDoc: CAU Metropolitano.

TEMA: Rugby - NombreDoc: Rivas R.C.

TEMA: Rugby - NombreDoc: Alcala R.C.

TEMA: Rugby - NombreDoc: XV Hortaleza.

TEMA: Rugby - NombreDoc: Rugby Guadalajara.

TEMA: Rugby - NombreDoc: Quijote Rugby.

TEMA: Rugby - NombreDoc: Club de Rugby Ingenieros Industriales Las Rozas.

TEMA: Rugby - NombreDoc: Olimpico Rugby Club.

TEMA: Series - NombreDoc: Juego de Tronos.

TEMA: Series - NombreDoc: Cuentame.

TEMA: Series - NombreDoc: Waliking Dead.

TEMA: Series - NombreDoc: Arrow.

TEMA: Videojuegos - NombreDoc: GTA V.

TEMA: Videojuegos - NombreDoc: Red Dead Redemption.

TEMA: Videojuegos - NombreDoc: Battlefield 4.

TEMA: Videojuegos - NombreDoc: Need for Speed.

TEMA: Videojuegos - NombreDoc: Call Of Duty Black Ops 3.

TEMA: Videojuegos - NombreDoc: Medal of Honor Warfighter.

TEMA: Videojuegos - NombreDoc: World of Warships.

TEMA: Videojuegos - NombreDoc: Rome II Total War.

TEMA: Videojuegos - NombreDoc: Mafia II.

TEMA: Videojuegos - NombreDoc: Uncharted 4.

TEMA: Ajedrez - NombreDoc: Paul.

TEMA: Ajedrez - NombreDoc: Wilhelm.

TEMA: Ajedrez - NombreDoc: José.

TEMA: Ajedrez - NombreDoc: Alexander.

TEMA: Ajedrez - NombreDoc: Mikhail.

TEMA: Ajedrez - NombreDoc: Mikhail.

TEMA: Ajedrez - NombreDoc: Bobby.

TEMA: Ajedrez - NombreDoc: Anatoly.

TEMA: Ajedrez - NombreDoc: Garry.

TEMA: Ajedrez - NombreDoc: Veselin.

TEMA: Ajedrez - NombreDoc: Magnus.

TEMA: Libros - NombreDoc: El resplandor.

TEMA: Libros - NombreDoc: La llamada Cthulhu.

TEMA: Libros - NombreDoc: La sombra del viento.

TEMA: Libros - NombreDoc: Harry Potter: La piedra filosofal.

TEMA: Libros - NombreDoc: Los juegos del hambre 1.

TEMA: Libros - NombreDoc: Marina.

TEMA: Libros - NombreDoc: Odio.

TEMA: Libros - NombreDoc: Se lo que estas pensando.

TEMA: Fútbol - NombreDoc: Real Madrid.

TEMA: Fútbol - NombreDoc: FC Barcelona.

TEMA: Fútbol - NombreDoc: Bayern Munich.

TEMA: Fútbol - NombreDoc: Manchester United.

TEMA: Fútbol - NombreDoc: Juventus.

TEMA: Fútbol - NombreDoc: Leicester City.

TEMA: Fútbol - NombreDoc: Paris Saint-Germain.

TEMA: Fútbol - NombreDoc: Machester City.

TEMA: Fútbol - NombreDoc: Rayo Vallecano.

TEMA: Baloncesto - NombreDoc: Allen Iverson.

TEMA: Baloncesto - NombreDoc: Stephen Curry.

TEMA: Baloncesto - NombreDoc: Carmelo Anthony.

TEMA: Baloncesto - NombreDoc: Kobe Bryant.

TEMA: Música - NombreDoc: Maluma.

TEMA: Música - NombreDoc: Tom Odell.

TEMA: Música - NombreDoc: Nicky Jam.

TEMA: Música - NombreDoc: Maroon 5.

TEMA: Música - NombreDoc: Kiko Rivera.

TEMA: Música - NombreDoc: Muse.

TEMA: Música - NombreDoc: Katy Perry.

TEMA: Música - NombreDoc: Drake.

TEMA: Música - NombreDoc: Cafe Quijano.

TEMA: Música - NombreDoc: Abraham Mateo.

TEMA: Surf - NombreDoc: Fish.

TEMA: Surf - NombreDoc: Longboard.

TEMA: Surf - NombreDoc: Shortboard.

TEMA: Surf - NombreDoc: Funboard.

TEMA: Surf - NombreDoc: Gun.

TEMA: Surf - NombreDoc: Hybrid.

TEMA: Surf - NombreDoc: StandUp.

TEMA: Surf - NombreDoc: Malibu.

TEMA: Surf - NombreDoc: Bodyboard.

TEMA: Surf - NombreDoc: SurfSkate.

e.2) Agrupa con aggregate para obtener un documento con el nombre de cada tema y una lista de nombres de sus documentos, además de cuántos documentos son.

consultaE = db.aficiones.aggregate([{$group:{\_id: {"Tema": "$Tema"},count:{$sum:1},documentos:{$addToSet :"$Nombre"}}}]);

{ "\_id" : { "Tema" : "Fútbol" }, "count" : 12, "documentos" : [ "Machester City", "Cristiano Ronaldo", "FC Barcelona", "Manchester United", "Leicester City", "Real Madrid", "Juventus", "Sergio Ramos", "Bayern Munich", "Isco", "Paris Saint-Germain", "Rayo Vallecano" ] }

{ "\_id" : { "Tema" : "Libros" }, "count" : 17, "documentos" : [ "El resplandor", "Harry Potter: La piedra filosofal", "Marina", "El nombre del viento", "Mas allá de las sombras", "Se lo que estas pensando", "La daga de la ceguera", "El ojo fragmentado", "El temor del hombre sabio", "El camino de las sombras", "Corazón inquieto: la vida de San Agustin", "La sombra del viento", "El prisma negro", "Odio", "Los juegos del hambre 1", "La llamada Cthulhu", "Al filo de las sombras" ] }

{ "\_id" : { "Tema" : "Rugby" }, "count" : 10, "documentos" : [ "Rugby Guadalajara", "XV Hortaleza", "Torrejon R.C.", "Olimpico Rugby Club", "Quijote Rugby", "CAU Metropolitano", "San Isidro R.C", "Alcala R.C", "Club de Rugby Ingenieros Industriales Las Rozas", "Rivas R.C" ] }

{ "\_id" : { "Tema" : "Baloncesto" }, "count" : 4, "documentos" : [ "Stephen Curry", "Kobe Bryant", "Allen Iverson", "Carmelo Anthony" ] }

{ "\_id" : { "Tema" : "MotoGP" }, "count" : 11, "documentos" : [ ] }

{ "\_id" : { "Tema" : "Surf" }, "count" : 10, "documentos" : [ "Malibu", "Hybrid", "Bodyboard", "Funboard", "Fish", "Gun", "Longboard", "StandUp", "Shortboard", "SurfSkate" ] }

{ "\_id" : { "Tema" : "Ajedrez" }, "count" : 11, "documentos" : [ "Alexander", "Mikhail", "Anatoly", "Wilhelm", "Garry", "Veselin", "Magnus", "José", "Bobby", "Paul" ] }

{ "\_id" : { "Tema" : "Música" }, "count" : 10, "documentos" : [ "Tom Odell", "Katy Perry", "Kiko Rivera", "Maluma", "Nicky Jam", "Maroon 5", "Drake", "Cafe Quijano", "Abraham Mateo", "Muse" ] }

{ "\_id" : { "Tema" : "Series" }, "count" : 4, "documentos" : [ "Cuentame", "Juego de Tronos", "Arrow", "Waliking Dead" ] }

{ "\_id" : { "Tema" : "Videojuegos" }, "count" : 10, "documentos" : [ "Rome II Total War", "Call Of Duty Black Ops 3", "World of Warships", "Mafia II", "Red Dead Redemption", "Medal of Honor Warfighter", "GTA V", "Need for Speed", "Uncharted 4", "Battlefield 4" ] }

1. Rebaja un 10% al precio de todos los componentes peor valorados (puntuación < 7). Y en la misma actualización añades el atributo Descuento a todas las aficiones: su valor no es un porcentaje fijo, sino que se lo asignas tú de acuerdo a esta regla: cuanto mayor puntuación, menor % de descuento (inventa la fórmula). Usa forEach y una function con replaceOne.

consultaF = db.aficiones.find().forEach(

function (myDoc) {

var descuento = myDoc.Precio \* 0.10;

var porcentaje = (1 - myDoc.Puntuacion/10) \* 10 ;

if (myDoc.Puntuacion < 7 ){

myDoc.Precio = myDoc.Precio - descuento;

}

myDoc.Descuento = porcentaje;

print(myDoc);

db.aficiones.save(myDoc);

});

1. (para nota) Queremos tener una colección PorNivel donde vas a crear (cuatro documentos) que representan cuatro niveles de calidad calculados, cada uno tendrá un vector con los documentos de aficiones que le correspondan. Sigue estos pasos:

1. Crear la colección PorNivel. Cada documento tendrá estos campos:

NomCal: Nombre del intervalo de calidad, con uno de estos valores : nivel\_1, nivel\_2, nivel\_3 y nivel\_4

Componentes: un array/vector que contenga, como elementos, los componentes (documentos) de la colección aficiones que correspondan a ese nivel. Además, cada elemento del array, debe tener un campo con el valor de calidad calculado del componente, obtenido multiplicando por 10 la puntuación del componente. Ese campo es el valor usado para asignarle un nivel, dentro de estos cuatro intervalos de valores: 0 < = nivel\_1 <=30, 30 <

nivel\_2 <=50, 50 < nivel\_3 <=70 y 70 < nivel\_4.

// definir los límites de los intervalos de calidad

var nivel\_1\_min = 0;

var nivel\_1\_max = 30;

var nivel\_2\_min = 30;

var nivel\_2\_max = 50;

var nivel\_3\_min = 50;

var nivel\_3\_max = 70;

var nivel\_4\_min = 70;

var nivel\_4\_max = 100;

// definir la función para asignar un nivel a un componente

function asignarNivel(puntuacion) {

var valor = puntuacion \* 10;

if (valor >= nivel\_1\_min && valor <= nivel\_1\_max) {

return "nivel\_1";

} else if (valor > nivel\_2\_min && valor <= nivel\_2\_max) {

return "nivel\_2";

} else if (valor > nivel\_3\_min && valor <= nivel\_3\_max) {

return "nivel\_3";

} else {

return "nivel\_4";

}

}

// inicializar la variable para los documentos de PorNivel

var por\_nivel = {

"nivel\_1": { "NomCal": "nivel\_1", "Componentes": [] },

"nivel\_2": { "NomCal": "nivel\_2", "Componentes": [] },

"nivel\_3": { "NomCal": "nivel\_3", "Componentes": [] },

"nivel\_4": { "NomCal": "nivel\_4", "Componentes": [] }

};

// consultar la colección aficiones

var cursor = db.aficiones.find();

// iterar sobre los documentos de la colección aficiones

while (cursor.hasNext()) {

var componente = cursor.next();

var nivel = asignarNivel(componente.Puntuacion);

por\_nivel[nivel].Componentes.push({

"Nombre": componente.Nombre,

"Apodo": componente.Apodo,

"Precio": componente.Precio,

"Puntuacion": componente.Puntuacion,

"ValorCalidad": componente.Puntuacion \* 10

});

}

// insertar los documentos de PorNivel en la colección correspondiente

for (var key in por\_nivel) {

db.PorNivel.insertOne(por\_nivel[key]);

}

2. Carga en la colección PorNivel todos los componentes de la colección aficiones que correspondan.

3. Imprime el contenido de la colección PorNivel formateado, poniendo una línea de guiones entre cada nivel.

4. Consulta PorNivel para obtener los 5 elementos más baratos independientemente del nivel al que pertenezcan: su nombre, su precio y su NomCal

5. Elimina los 2 componentes más caros de cada intervalo (NomCal). Puedes hacer una operación separada para cada intervalo.

# **APARTADO 4.- Usando Colecciones limitadas (capped)**

Queremos mantener en una colección *superGuai*, los 5 mejores componentes de la colección aficiones. Para ello hacemos lo siguiente:

a) Crear dicha colección

consulta4a = db.createCollection("superGuai",{capped:true,size:100000,max:5});

b) Crea las operaciones necesarias para poner los 5 mejores elementos de acuerdo al criterio de calidad explicado en el apartado 3.g., **pero usando** **la colección** aficiones **no uses** PorNivel. Antes de obtener los mejores elementos: calcula el nivel de calidad y crea el atributo Calidad y su valor, actualizando la colección aficiones.

consulta4b = db.PorNivel.aggregate(

{ $unwind: "$Componentes" } ,

{ $project: { \_id:0 , Componente: "$Componentes" } } ,

{ $limit: 5 }

).forEach(function(doc) {

db.superGuai.insert(doc);

} );

c) Inserta un elemento a mano.

consulta4c = db.superGuai.insert({Insertar: "Test"});

d) Lista todos los componentes para comprobar que mantiene los último cinco introducidos

db.superGuai.find({})

# **APARTADO 5.-**

Siguiendo las pautas para diseñar una BD no-sql en las diapositivas de la Teoría: diseña tú una BD de tema libre y describe qué operaciones quiere hacer. Teniendo en cuenta que sea un tema donde una BD tipo SQL no sea adecuada.

**APARTADO 5.- EXTRA**

Deseamos introducir elementos compuestos, ej.: como en un equipo de fútbol si incluimos cada jugador con sus datos personales. Y queremos hacer muchas consultas sobre esos elementos compuestos ej.: datos personales. ¿Conviene normalizar o desnormalizar?. ¿Cómo debería quedar la representación de la colección?

# **APARTADO 6.- Trabajando con la colección aficiones en MongoDB: Actualizaciones**

a) Repasar ejemplos de update e insert: ejemplos-Libro-MongoDB-v3.pdf

b) Arreglar la colección aficiones. Usando varias update sucesivas puedes añadir y quitar atributos sin

perder sus valores. Arregla los errores más destacados que son:

b.1) MotoGP: cambiar el atributo NombreEquipo por Nombre (sin perder sus valores)

consulta6b = db.aficiones.update( { Tema: "MotoGP" } , { $rename: { NombreEquipo: "Nombre" } } , { multi: true } );

WriteResult({ "nMatched" : 11, "nUpserted" : 0, "nModified" : 11 })

Averiguamos con: db.aficiones.find({Tema:"MotoGP"})

b.2) Precio: En Futbol y Baloncesto hay valores exagerados: hacer una función con un bucle que , en cada ciclo, quite un cero hasta dejar cifras de tres dígitos.

consulta6b2 = db.aficiones.find({$or:[{Tema:"Fútbol"},{Tema:"Baloncesto"}]}).forEach

(

function (undoc) {

print ("Antes: " + undoc.Precio);

while (undoc.Precio >=1000)

undoc.Precio /=10;

print ("Una vez ya modificado: " + undoc.Precio);

db.aficiones.update ({\_id: undoc.\_id}, undoc);

}

);

Antes: 100000000

Una vez ya modificado: 100

Antes: 60000000

Una vez ya modificado: 600

Antes: 80000000

Una vez ya modificado: 800

Antes: 1000

Una vez ya modificado: 100

Antes: 900

Una vez ya modificado: 900

Antes: 700

Una vez ya modificado: 700

Antes: 900

Una vez ya modificado: 900

Antes: 700

Una vez ya modificado: 700

Antes: 500

Una vez ya modificado: 500

Antes: 1000

Una vez ya modificado: 100

Antes: 1000

Una vez ya modificado: 100

Antes: 5.9049

Una vez ya modificado: 5.9049

Antes: 50000000

Una vez ya modificado: 500

Antes: 45000000

Una vez ya modificado: 450

Antes: 40000000

Una vez ya modificado: 400

Antes: 35000000

Una vez ya modificado: 350

b.3) Añadir el precio a MotoGP y Ajedrez con valor fijo de 100 (en un solo update)

consulta6b3 = db.aficiones.update( { $or: [ { Tema: "MotoGP" } , { Tema: "Ajedrez" } ] } , { $set: { Precio: 100 } } , { multi: true });

b.4) En MotoGP: Atributo incorrecto: hay "Puntuacón" donde debería haber "Puntuacion" sin perder su valor

consulta6b4 = db.aficiones.update( { Tema: "MotoGP" } , { $rename: { Puntuacón: "Puntuacion" } } , { multi: true } );

b.5) (sacar nota) En Ajedrez : unir los valores de Nombre y Apellidos en el atributo Nombre. Quitar el atributo Apellidos.

consulta6b45 = db.aficiones.find( { Tema: "Ajedrez" } ).forEach(

function(nomb) {

var Nombrenuevo = nomb.Nombre + " " + nomb.Apellidos;

db.aficiones.update( { \_id: nomb.\_id } , { $unset: { Apellidos: "" }} );

db.aficiones.update( { \_id: nomb.\_id } , { $set: { Nombre:

Nombrenuevo } } );

}

);

Averiguamos con: db.aficiones.find({Tema:"Ajedrez"})

c) Esos errores son ocasionados porque la colección no tenía un validador automático. Sigue estos pasos para crearlo:

- En Studio3T, crea una colección vacía misAficiones, solo para tu Tema (tu afición)

- Define el validador adecuado siguiendo las pautas del Esquema en validar-con-Schema.js.

- Para crear y validar el Esquema de validación:

- En la colección misAficiones, botón dcho + “Add/Edit Validator”. Se abre ventana

- Copia el Esquema del validador en la ventana

- Validar el documento : en el botón izq, esquina inferior izquierda.

- Botón para Salvar

- A partir de ahora, todos los documentos insertados en la IntelliShell, se validan automáticamente.

- Inserta en misAficiones los documentos de tu tema que están en colección aficiones. Deben superar la validación para que se inserten. Para ello usa menu de la barra arriba, opción Document + Edit Document(JSON) + Validate +

Update. Si no existe lo crea

Nota: ver otro ejemplo del $jsonSchema en <https://docs.mongodb.com/manual/core/schema-validation/>

He usado el ejemplo que nos has dado para hacer una validación donde he expuesto

que para hacer la validación tenga estos campos principalmente, Tema, Nombre,

Puntuacion, Apodo, como se ve en la imagen y lo demás del ejemplo lo deje igual

/\* db.createCollection("misAficiones", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "Tema", "Nombre", "Puntuacion","Apodo"],

properties: {

Tema: {

bsonType: "string",

description: "Nombre de aficion "

},

Nombre: {

bsonType: "string",

description: " Nombre de la actividad"

},

Puntuacion: {

bsonType: "double",

minimum: 0.0,

maximum: 10.0,

description: "Un campo decimal entre 0 y 10"

},

Apodo:{

bsonType: "string",

description: "Nombre de apodo "

},

Curso: {

enum: [ "Primero", "Segundo", "Tercero", "Cuarto" ],

description: "Solo puede ser alguno de los valores posibles"

},

PeriodoPractica: {

bsonType: [ "int" ],

minimum: 1960,

maximum: 2020,

description: "Años entre 1960 y 2020"

},

"viveEn.Ciudad" : {

bsonType: "string",

description: "Nombre de ciudad"

},

"viveEn.Pais" : {

bsonType: "string",

description: "Nombre de pais"

}

}

}

}

})

\*

d) Queremos evitar repeticiones de datos, ej.: tener el mismo libro repetido para todos los que les guste. Para ello queremos normalizar aficiones para tener una colección solodatos con la lista de detalles de las aficiones. Además, para no estropear aficiones, crea otra colección soloaficiones con lo que debe quedar en aficiones después de normalizarla: quedarán los atributos más frecuentes en las consultas: los Campos Obligatorios del enunciado, y el identificador del nuevo objeto con los detalles en solodatos.

db.createCollection("solodatos")

db.createCollection("soloaficiones")

db.aficiones.aggregate( { $project: { CamposOblig: "$$ROOT" , Detalles: "$$ROOT" } } , { $project: { CamposOblig: {Nombre: 1, Precio: 1, Tema: 1, Apodo: 1, Puntuacion: 1, Ref: "$\_id" } , Detalles: "$$ROOT.Detalles" } } , { $project: { \_id: 0, Detalles: { Tema: 0, Puntuacion: 0, Apodo: 0, Precio: 0 } } }).forEach(

function(doc) {

db.solodatos.insert(doc.Detalles);

db.soloaficiones.insert(doc.CamposOblig);

}

)

e) El efecto del apartado anterior es que ahora la consulta completa uniendo ambas colecciones es más compleja y más lenta. Cómo se hace?

db.soloaficiones.aggregate( [{ $lookup: { from: "solodatos" , localField: "Ref", foreignField: "\_id" , as: "Detalles" } }] )

f) (para nota) Crea una colección reconstruida que sea el resultado de la unión de solodatos y soloaficiones (el resultado del apartado anterior)

db.createCollection("reconstruida")

db.soloaficiones.aggregate( [{ $lookup: { from:"solodatos", localField:"Ref", foreignField:"\_id", as:"Detalles" } } , { $out: "reconstruida" } ])