Alumno: **Luis Alberto Jaramillo Pulido**

Práctica 5 semana 12: Utilizar una BD en MongoDB

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

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

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

Solución

Para no sacar el identificador basta con poner _id:0

Comando

```
db.aficiones.find({"Puntuacion" :{$gte:9}},{ _id:0,"Tema":1,"Nombre":2,"Puntuacion":3});
```

Salida

```
{ "Tema" : "Fútbol", "Nombre" : "Cristiano Ronaldo", "Puntuacion" : 9 }
{ "Tema" : "Rugby", "Nombre" : "Torrejon R.C.", "Puntuacion" : 10 }
{ "Tema" : "Rugby", "Nombre" : "San Isidro R.C", "Puntuacion" : 10 }
{ "Tema" : "Series", "Nombre" : "Juego de Tronos", "Puntuacion" : 10 }
{ "Tema" : "Videojuegos", "Nombre" : "GTA V", "Puntuacion" : 10 }
{ "Tema" : "Videojuegos", "Nombre" : "Red Dead Redemption", "Puntuacion" : 9 }
{ "Tema" : "Videojuegos", "Nombre" : "Battlefield 4", "Puntuacion" : 9 }
{ "Tema" : "Videojuegos", "Nombre" : "Need for Speed", "Puntuacion" : 9 }
{ "Tema" : "Libros", "Nombre" : "El resplandor", "Puntuacion" : 9 }
{ "Tema" : "Libros", "Nombre" : "Marina", "Puntuacion" : 9 }
{ "Tema" : "Libros", "Nombre" : "Odio", "Puntuacion" : 9 }
{ "Tema" : "Fútbol", "Nombre" : "Real Madrid", "Puntuacion" : 10 }
{ "Tema" : "Fútbol", "Nombre" : "FC Barcelona", "Puntuacion" : 10 }
{ "Tema" : "Fútbol", "Nombre" : "Bayern Munich", "Puntuacion" : 10 }
{ "Tema" : "Fútbol", "Nombre" : "Juventus", "Puntuacion" : 9 }
{ "Tema" : "Baloncesto", "Nombre" : "Allen Iverson", "Puntuacion" : 10 }
{ "Tema" : "Baloncesto", "Nombre" : "Stephen Curry", "Puntuacion" : 9.5 }
{ "Tema" : "Baloncesto", "Nombre" : "Carmelo Anthony", "Puntuacion" : 10 }
{ "Tema" : "Música", "Nombre" : "Maluma", "Puntuacion" : 10 }
{ "Tema" : "Música", "Nombre" : "Tom Odell", "Puntuacion" : 10 }
```

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.

Solución

Comando

```
db.aficiones.aggregate([
                    { $match: { "Puntuacion":{$gte:9} } },
                    { $group: { _id: "$Tema",totalGasto: { $sum: "$Precio" },totalComponentes: { $sum: 1 } } },
                    {$sort: { _id:1}}
                    ]);
```

Salida

```
{ "_id" : "Baloncesto", "totalGasto" : 135000000, "totalComponentes" : 3 }
{ "_id" : "Fútbol", "totalGasto" : 100003300, "totalComponentes" : 5 }
{ "_id" : "Libros", "totalGasto" : 28.4, "totalComponentes" : 3 }
{ "_id" : "Música", "totalGasto" : 50.97, "totalComponentes" : 4 }
{ "_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 las puntuaciones (10,9,8,7,6 y 5) por separado: Para cada puntuación, obtener los nombres de componentes que la tienen. (cuando ya funcione, incluye también el nombre su tema)

Solución

_id:-1 para que me ordene de forma descendente

Comando

```
db.aficiones.aggregate(

        {$match :{ $or :[
                    {Puntuacion : {$eq : 5}},
                    {Puntuacion : {$eq : 6}},
                    {Puntuacion : {$eq : 7}},
                    {Puntuacion : {$eq : 8}},
                    {Puntuacion : {$eq : 9}},
                    {Puntuacion : {$eq : 10}},
                    ]
        }
},
{$group : { _id : "$Puntuacion", Nombres : {$push : "$Tema"} }},{$sort: { _id:-1}});
```

Salida

```
{ "_id" : 10, "Nombres" : [ "Rugby", "Rugby", "Series", "Videojuegos", "Fútbol", "Fútbol", "Fútbol", "Baloncesto", "Baloncesto",
"Música", "Música" ] }
{ "_id" : 9, "Nombres" : [ "Fútbol", "Videojuegos", "Videojuegos", "Videojuegos", "Libros", "Libros", "Libros", "Fútbol", "Música",
"Música" ] }
{ "_id" : 8, "Nombres" : [ "Libros", "Libros", "Fútbol", "Rugby", "Videojuegos", "Videojuegos", "Libros", "Libros", "Fútbol", "Fútbol",
"Fútbol", "Música", "Arte", "Arte", "Arte" ] }
{ "_id" : 7, "Nombres" : [ "Rugby", "Rugby", "Rugby", "Videojuegos", "Videojuegos", "Videojuegos", "Libros", "Fútbol", "Arte",
"Arte", "Arte", "Arte", "Arte" ] }
{ "_id" : 6, "Nombres" : [ "Videojuegos", "Libros", "Libros", "Música", "Arte", "Arte" ] }
{ "_id" : 5, "Nombres" : [ "Rugby", "Fútbol", "Música", "Arte", "Arte" ] }
```

4. Lista de Apodos para cada afición (Tema)

Solución

Comando

```
db.aficiones.aggregate(
[
        {$group : { _id : "$Tema", Apodos_Aficion: {$addToSet : "$Apodo"} }},
        {$sort: { _id:1}}
]
);
```

Salida

```
{ "_id" : "Ajedrez", "Apodos_Aficion" : [ "Bobby" ] }
{ "_id" : "Arte", "Apodos_Aficion" : [ "lujarami" ] }
{ "_id" : "Baloncesto", "Apodos_Aficion" : [ "NBA" ] }
{ "_id" : "Fútbol", "Apodos_Aficion" : [ "Mike", "soFIFA" ] }
{ "_id" : "Libros", "Apodos_Aficion" : [ "Turre", "Ceor" ] }
{ "_id" : "MotoGP", "Apodos_Aficion" : [ "MGP" ] }
{ "_id" : "Música", "Apodos_Aficion" : [ "dios" ] }
{ "_id" : "Rugby", "Apodos_Aficion" : [ "Infor" ] }
{ "_id" : "Series", "Apodos_Aficion" : [ "Tyrion Lanister" ] }
{ "_id" : "Videojuegos", "Apodos_Aficion" : [ "Giorgio" ] }
```

b) (para nota) A quien le gusta los mismos componentes?: Queremos obtener para cada Tema: Apodo, Nombre y Tema de los componentes (que tengan apodo distinto) en los que coincide al menos un nombre del mismo Tema.

Solución

Para comprobar si hace bien la consulta, he agregado una fila más, correspondiente a mi afición, otorgándole un nombre repetido, “La tentación de San Antonio”.

En el tema “MotoGP” , se muestra el nombre “null”, porque no tiene un nombre asignado, existen 11 filas que no tienen un nombre asignado.

$First Devuelve un valor del primer documento para cada grupo.

Comando

```
db.aficiones.aggregate(

[

{"$group" : {"_id": "$Nombre", "Apodos": {"$addToSet": "$Apodo"}, "Temas": {"$first": "$Tema"}, "count": {"$sum": 1}}},

{"$match": {"count": {"$gte": 2}}}

]

);
```

Salida

```
{ "_id" : "La tentación de San Antonio", "Apodos" : [ "lujarami" ], "Temas" : "Arte", "count" : 2 }
{ "_id" : null, "Apodos" : [ "MGP" ], "Temas" : "MotoGP", "count" : 11 }
{ "_id" : "Mikhail", "Apodos" : [ "Bobby" ], "Temas" : "Ajedrez", "count" : 2 }
```

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

Solución

Comando

```
db.aficiones.aggregate([
        {"$match": {"$and": [{Puntuacion: {"$gt":4}}, {Puntuacion: {"$lt":9}}]}},
        {"$group" : {"_id": "$Nombre","Puntuacion": {"$first": "$Puntuacion"} ,"apodos": {"$addToSet":
"$Apodo"}, "temas": {"$first": "$Tema"}, "count": {"$sum": 1}}},
        {"$match": {"count": {"$gte": 2}}}
                ]);
```

Salida

```
{ "_id" : "La tentación de San Antonio", "Puntuacion" : 8, "apodos" : [ "lujarami" ], "temas" : "Arte", "count" : 2 }
```

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

Solución

Consulta1

Me muestra el nombre y el precio, cuando el precio es menor o igual a una cantidad 50, y me los ordena por nombre(ascendente).

Me resulta interesante el uso del comando “$project” para especificar los campos que quiero que me devuelva el documento, aunque no solo se limita a los campos existentes, también puede incluir nuevos campos que se pueden calcular.

Comando

```
db.aficiones.aggregate([
        { $match: { "Precio":{$lte:50} } },
        { $project: { _id: "$Nombre",Precio:"$Precio"} },
        {$sort: { _id:1}}
        ]);
```

Salida

```
{ "_id" : "Abraham Mateo", "Precio" : 4.63069745194075 }
{ "_id" : "Al filo de las sombras", "Precio" : 20.7 }
{ "_id" : "Arrow", "Precio" : 9.95 }
{ "_id" : "Battlefield 4", "Precio" : 39.95 }
{ "_id" : "Cafe Quijano", "Precio" : 0.9246570742370688 }
{ "_id" : "Call Of Duty Black Ops 3", "Precio" : 45.95 }
{ "_id" : "Corazón inquieto: la vida de San Agustin", "Precio" : 17 }
{ "_id" : "Cuentame", "Precio" : 0 }
{ "_id" : "Drake", "Precio" : 2.7776772630889086 }
{ "_id" : "El beso", "Precio" : 18.6121064540979 }
{ "_id" : "El caminante sobre el mar de nubes", "Precio" : 14.0932338704019 }
{ "_id" : "El camino de las sombras", "Precio" : 9.95 }
{ "_id" : "El jardin de las delicias", "Precio" : 10.1392203596679 }
{ "_id" : "El nombre del viento", "Precio" : 20.7 }
{ "_id" : "El ojo fragmentado", "Precio" : 10.95 }
```

```
{ "_id" : "El prisma negro", "Precio" : 10.95 }
{ "_id" : "El resplandor", "Precio" : 9.95 }
{ "_id" : "El temor del hombre sabio", "Precio" : 24.9 }
{ "_id" : "Harry Potter: La piedra filosofal", "Precio" : 2.779530283277762 }
{ "_id" : "Juego de Tronos", "Precio" : 9.95 }
```

Consulta 2

Me muestra el nombre de las obras que acaben en la letra “e” y de apodo “lujarami”.

Me resulta interesante el uso del comando $regex para especificar en la búsqueda la letra de terminación.

Comando

```
db.aficiones.find(
        {
                $and:
                [
                 {Apodo: "lujarami" },
                {"Nombre": {$regex:".*e$"}}
                ]

        }
  ,{Nombre:1}
);
```

Salida

```
{ "_id" : ObjectId("5ebea5888f7a2c3705eefb2b"), "Nombre" : "El hijo del hombre" }
```

Consulta 3

Me muestra toda la información de objeto, donde el tema sea “Rugby” y la puntuación sea <=4

Me resulta interesante el uso de “$where” para pasar una cadena que contenga una expresión de JavaScript o una función completa de JavaScript

Comando

```
db.aficiones.find({"Tema": "Rugby" ,"Puntuacion" : {$exists: true}, $where : "this.Puntuacion <= 4"}).pretty();
```

Salida

```
{
    "_id" : ObjectId("5746cf7de6046b47d407362e"),
    "Tema" : "Rugby",
    "Apodo" : "Infor",
    "Nombre" : "CAU Metropolitano",
    "Puntuacion" : 3,
    "Precio" : 64.85570660981443,
    "direccion" : [
        {
            "Campo" : "Polideportivo Orcasitas",
            "calle" : "Avda. Rafaela Ybarra",
            "Ciudad" : "Madrid",
            "Provincia" : "Madrid",
```

```
                "coord" : [
                    "40.3773531",
                    "-3.7131738"
                ]
            }
        ],
        "Fundacion" : 1962,
        "Equipos" : [
            {
                "Senior" : "1ª Regional",
                "Sub18" : "1ª Regional",
                "Sub16" : "Si",
                "Infantil" : "Si",
                "Femenino" : "No"
            }
        ],
        "ValorCalidad" : 0.8571428571428571,
        "Descuento" : 7
}
{
        "_id" : ObjectId("5746cf7de6046b47d4073630"),
        "Tema" : "Rugby",
        "Apodo" : "Infor",
        "Nombre" : "Alcala R.C",
        "Puntuacion" : 1,
        "Precio" : 64.85570660981443,
        "direccion" : [
            {
                "Campo" : "Campo de rugby Antonio Machado",
                "calle" : "C/ Jorge Guillén s/n",
                "Ciudad" : "Alcala de Henares",
                "Provincia" : "Madrid",
                "coord" : [
                    "40.4933645",
                    "-3.3728013"
                ]
            }
        ],
        "Fundacion" : 1967,
        "Equipos" : [
            {
                "Senior" : "1ª Regional",
                "Sub18" : "1ª Regional",
                "Sub16" : "Si",
                "Infantil" : "Si",
                "Femenino" : "No"
            }
        ],
        "ValorCalidad" : 0.2857142857142857,
        "Descuento" : 9
}
{
        "_id" : ObjectId("5746cf7de6046b47d4073634"),
        "Tema" : "Rugby",
        "Apodo" : "Infor",
        "Nombre" : "Club de Rugby Ingenieros Industriales Las Rozas",
        "Puntuacion" : 4,
        "Precio" : 74.12080755407365,
        "direccion" : [
```

```
            {
                "Campo" : "El cantizal",
                "calle" : "Calle Higuera",
                "Ciudad" : "Las Rozas",
                "Provincia" : "Madrid",
                "coord" : [
                    "40.5189782",
                    "-3.9250557"
                ]
            }
        ],
        "Fundacion" : 1971,
        "Equipos" : [
            {
                "Senior" : "Division de honor B",
                "Sub18" : "2ª Regional",
                "Sub16" : "Si",
                "Infantil" : "Si",
                "Femenino" : "Si"
            }
        ],
        "ValorCalidad" : 1,
        "Descuento" : 6
}
```


Consulta 4

Me muestra el plan de ejecución de la consulta anterior devolviendo las estadísticas de ejecución

Me resulta interesante porque devuelve las estadísticas que describen la ejecución del plan ganador y explain() devuelve la información de queryPlanner y executionStats, sin embargo, executionStats no proporciona información de ejecución de consultas para los planes rechazados.

Comando

```
db.aficiones.find({"Tema": "Rugby" ,"Puntuacion" : {$exists: true}, $where : "this.Puntuacion <= 4"}).explain("executionStats");
```

Salida


```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "pracmongo.aficiones",
        "indexFilterSet" : false,
        "parsedQuery" : {
            "$and" : [
                {
                    "Tema" : {
                        "$eq" : "Rugby"
                    }
                },
                {
                    "Puntuacion" : {
                        "$exists" : true
                    }
```

```
                },
                {
                    "$where" : {
                        "code" : "this.Puntuacion <= 4",
                        "scope" : {

                        }
                    }
                }
            ]
        },
        "winningPlan" : {
            "stage" : "COLLSCAN",
            "filter" : {
                "$and" : [
                    {
                        "Tema" : {
                            "$eq" : "Rugby"
                        }
                    },
                    {
                        "Puntuacion" : {
                            "$exists" : true
                        }
                    },
                    {
                        "$where" : {
                            "code" : "this.Puntuacion <= 4",
                            "scope" : {

                            }
                        }
                    }
                ]
            },
            "direction" : "forward"
        },
        "rejectedPlans" : [ ]
    },
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 3,
        "executionTimeMillis" : 38,
        "totalKeysExamined" : 0,
        "totalDocsExamined" : 100,
        "executionStages" : {
            "stage" : "COLLSCAN",
            "filter" : {
                "$and" : [
                    {
                        "Tema" : {
                            "$eq" : "Rugby"
                        }
                    },
                    {
                        "Puntuacion" : {
                            "$exists" : true
                        }
                    },
```

```
                        {
                            "$where" : {
                                "code" : "this.Puntuacion <= 4",
                                "scope" : {

                                }
                            }
                        }
                    ]
                },
                "nReturned" : 3,
                "executionTimeMillisEstimate" : 1,
                "works" : 102,
                "advanced" : 3,
                "needTime" : 98,
                "needYield" : 0,
                "saveState" : 0,
                "restoreState" : 0,
                "isEOF" : 1,
                "direction" : "forward",
                "docsExamined" : 100
            }
        },
        "serverInfo" : {
            "host" : "DESKTOP-OA0P071",
            "port" : 27017,
            "version" : "4.2.6",
            "gitVersion" : "20364840b8f1af16917e4c23c1b5f5efd8b352f8"
        },
        "ok" : 1
}
```

e) 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.

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.

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.

Solución

Comando

```
db.aficiones.aggregate( [
{ $group: {_id: { tema: "$Tema" },Documentos: { $addToSet : "$Nombre" }}}]);
```

Salida

{ "_id" : { "tema" : "Libros" }, "Documentos" : ["Odio", "Al filo de las sombras", "El camino de las sombras", "El ojo fragmentado", "Se lo que estas pensando", "Mas allá de las sombras", "El nombre del viento", "El temor del hombre sabio", "La daga de la ceguera", "Marina", "El resplandor", "Corazón inquieto: la vida de San Agustin", "La llamada Cthulhu", "Los juegos del hambre 1", "El prisma negro", "Harry Potter: La piedra filosofal", "La sombra del viento"] }
{ "_id" : { "tema" : "Rugby" }, "Documentos" : ["Club de Rugby Ingenieros Industriales Las Rozas", "Rugby Guadalajara", "Olimpico Rugby Club", "Alcala R.C", "Rivas R.C", "CAU Metropolitano", "San Isidro R.C", "Quijote Rugby", "Torrejon R.C.", "XV Hortaleza"] }
{ "_id" : { "tema" : "Videojuegos" }, "Documentos" : ["World of Warships", "Need for Speed", "Rome II Total War", "Battlefield 4", "GTA V", "Mafia II", "Red Dead Redemption", "Call Of Duty Black Ops 3", "Medal of Honor Warfighter", "Uncharted 4"] }
{ "_id" : { "tema" : "Fútbol" }, "Documentos" : ["Manchester United", "FC Barcelona", "Sergio Ramos", "Machester City", "Isco", "Paris Saint-Germain", "Rayo Vallecano", "Real Madrid", "Bayern Munich", "Cristiano Ronaldo", "Juventus", "Leicester City"] }
{ "_id" : { "tema" : "Series" }, "Documentos" : ["Arrow", "Waliking Dead", "Cuentame", "Juego de Tronos"] }
{ "_id" : { "tema" : "Ajedrez" }, "Documentos" : ["Bobby", "Mikhail", "Veselin", "Wilhelm", "Alexander", "Anatoly", "Garry", "Magnus", "Paul", "José"] }
{ "_id" : { "tema" : "Baloncesto" }, "Documentos" : ["Kobe Bryant", "Carmelo Anthony", "Stephen Curry", "Allen Iverson"] }
{ "_id" : { "tema" : "MotoGP" }, "Documentos" : [ ] }
{ "_id" : { "tema" : "Música" }, "Documentos" : ["Katy Perry", "Cafe Quijano", "Tom Odell", "Kiko Rivera", "Maluma", "Muse", "Abraham Mateo", "Drake", "Maroon 5", "Nicky Jam"] }
{ "_id" : { "tema" : "Arte" }, "Documentos" : ["La tentación de San Antonio", "La gran ola de Kanagawa", "Venus del espejo", "La noche estrellada", "El 3 de mayo en Madrid", "El beso", "La persistencia de la memoria", "El jardin de las delicias", "El hijo del hombre", "El caminante sobre el mar de nubes"] }

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.

Solución

Comando

```
db.aficiones.aggregate( [
{ $group: {_id: { tema: "$Tema" },NumeroDocumentos: { $sum: 1 },Documentos: { $addToSet : "$Nombre" }}},
{$sort: { _id:1}} ] ).forEach( function(miDocumento) { print(miDocumento);});
```

Salida

```
[object BSON]
[object BSON]
[object BSON]
[object BSON]
[object BSON]
[object BSON]
[object BSON]
[object BSON]
[object BSON]
[object BSON]
```

He usado el printjson(myDoc) para que me imprima cada documento de formato BSON, luego he hecho un sort para que me lo clasifique siguiendo el orden del nombre del tema.

Comando

```
db.aficiones.aggregate( [

{ $group: {_id: { tema: "$Tema" },NumeroDocumentos: { $sum: 1 },Documentos: { $addToSet : "$Nombre" }}},

{$sort: { _id:1}} ] ).forEach( function(miDocumento) { printjson(miDocumento);})
```

Salida

```
{
	"_id" : {
		"tema" : "Ajedrez"
	},
	"NumeroDocumentos" : 11,
	"Documentos" : [
		"Garry",
		"Magnus",
		"Paul",
		"José",
		"Wilhelm",
		"Veselin",
		"Mikhail",
		"Bobby",
		"Anatoly",
		"Alexander"
	]
}
{
	"_id" : {
		"tema" : "Arte"
	},
	"NumeroDocumentos" : 11,
	"Documentos" : [
		"El beso",
		"La persistencia de la memoria",
```

```
            "El jardin de las delicias",
            "El hijo del hombre",
            "El caminante sobre el mar de nubes",
            "La tentación de San Antonio",
            "La gran ola de Kanagawa",
            "Venus del espejo",
            "La noche estrellada",
            "El 3 de mayo en Madrid"
        ]
}
{
        "_id" : {
            "tema" : "Baloncesto"
        },
        "NumeroDocumentos" : 4,
        "Documentos" : [
            "Stephen Curry",
            "Kobe Bryant",
            "Allen Iverson",
            "Carmelo Anthony"
        ]
}
{
        "_id" : {
            "tema" : "Fútbol"
        },
        "NumeroDocumentos" : 12,
        "Documentos" : [
            "Isco",
            "Paris Saint-Germain",
            "Machester City",
            "Sergio Ramos",
            "Rayo Vallecano",
            "FC Barcelona",
            "Manchester United",
            "Leicester City",
            "Cristiano Ronaldo",
            "Juventus",
            "Real Madrid",
            "Bayern Munich"
        ]
}
{
        "_id" : {
            "tema" : "Libros"
        },
        "NumeroDocumentos" : 17,
        "Documentos" : [
            "El resplandor",
            "El temor del hombre sabio",
            "La daga de la ceguera",
            "Marina",
            "Corazón inquieto: la vida de San Agustin",
            "La llamada Cthulhu",
            "Los juegos del hambre 1",
            "Odio",
            "Harry Potter: La piedra filosofal",
            "El prisma negro",
            "El camino de las sombras",
            "La sombra del viento",
            "Al filo de las sombras",
            "Mas allá de las sombras",
            "El ojo fragmentado",
            "Se lo que estas pensando",
            "El nombre del viento"
        ]
```

```
}
{
    "_id" : {
        "tema" : "MotoGP"
    },
    "NumeroDocumentos" : 11,
    "Documentos" : [ ]
}
{
    "_id" : {
        "tema" : "Música"
    },
    "NumeroDocumentos" : 10,
    "Documentos" : [
        "Abraham Mateo",
        "Drake",
        "Maroon 5",
        "Nicky Jam",
        "Katy Perry",
        "Cafe Quijano",
        "Tom Odell",
        "Kiko Rivera",
        "Maluma",
        "Muse"
    ]
}
{
    "_id" : {
        "tema" : "Rugby"
    },
    "NumeroDocumentos" : 10,
    "Documentos" : [
        "Alcala R.C",
        "San Isidro R.C",
        "Torrejon R.C.",
        "XV Hortaleza",
        "Olimpico Rugby Club",
        "Rugby Guadalajara",
        "Club de Rugby Ingenieros Industriales Las Rozas",
        "CAU Metropolitano",
        "Rivas R.C",
        "Quijote Rugby"
    ]
}
{
    "_id" : {
        "tema" : "Series"
    },
    "NumeroDocumentos" : 4,
    "Documentos" : [
        "Cuentame",
        "Juego de Tronos",
        "Arrow",
        "Waliking Dead"
    ]
}
{
    "_id" : {
        "tema" : "Videojuegos"
    },
    "NumeroDocumentos" : 10,
    "Documentos" : [
        "Red Dead Redemption",
        "GTA V",
        "Battlefield 4",
        "World of Warships",
```

```
            "Mafia II",
            "Medal of Honor Warfighter",
            "Call Of Duty Black Ops 3",
            "Uncharted 4",
            "Need for Speed",
            "Rome II Total War"
        ]
}
```

f) 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 *save*.

Solución

Comando

```
db.aficiones.find().forEach(
        function (miDocumento) {
        var aux=0;
        var descuento = 0;
        if (miDocumento.Puntuacion < 7 ){
                descuento = miDocumento.Precio * 0.10;
                miDocumento.Precio = miDocumento.Precio - descuento;
        }
        aux = (1 - miDocumento.Puntuacion/10) * 10 ;
        miDocumento.Descuento = aux;
        printjson(miDocumento);
        db.aficiones.save(miDocumento);
});
```

Salida

A modo de simplificar la salida, he puesto de salida 5 filas de las 100 que me salen al ejecutar el comando con el nuevo campo “Descuento”

```
{
    "_id" : ObjectId("5746cf61e6046b47d4073615"),
    "Tema" : "Libros",
    "Apodo" : "Turre",
    "Nombre" : "El temor del hombre sabio",
    "Puntuacion" : 8.2,
    "Precio" : 24.9,
    "genero" : "Fantasia",
    "autor" : "Patrick Rothfuss",
    "paginas" : 1200,
    "ISBN" : 9788401339639,
    "ValorCalidad" : 32.93172690763052,
    "Descuento" : 1.8000000000000005
}
{
    "_id" : ObjectId("5746cf61e6046b47d4073616"),
    "Tema" : "Libros",
    "Apodo" : "Turre",
    "Nombre" : "El nombre del viento",
    "Puntuacion" : 7.9,
    "Precio" : 20.7,
    "genero" : "Fantasia",
    "autor" : "Patrick Rothfuss",
    "paginas" : 880,
    "ISBN" : 9788401337208,
    "ValorCalidad" : 38.16425120772947,
```

```
    "Descuento" : 2.0999999999999996
}
{
    "_id" : ObjectId("5746cf61e6046b47d4073617"),
    "Tema" : "Libros",
    "Apodo" : "Turre",
    "Nombre" : "El camino de las sombras",
    "Puntuacion" : 8.3,
    "Precio" : 9.95,
    "genero" : "Fantasia-epica",
    "autor" : "Brent Weeks",
    "paginas" : 592,
    "ISBN" : 9788499893679,
    "ValorCalidad" : 83.41708542713569,
    "Descuento" : 1.6999999999999993
}
{
    "_id" : ObjectId("5746cf61e6046b47d4073618"),
    "Tema" : "Libros",
    "Apodo" : "Turre",
    "Nombre" : "Mas allá de las sombras",
    "Puntuacion" : 8.3,
    "Precio" : 10.95,
    "genero" : "Fantasia-epica",
    "autor" : "Brent Weeks",
    "paginas" : 624,
    "ISBN" : 9788499894348,
    "ValorCalidad" : 75.79908675799088,
    "Descuento" : 1.6999999999999993
}
{
    "_id" : ObjectId("5746cf61e6046b47d4073619"),
    "Tema" : "Libros",
    "Apodo" : "Turre",
    "Nombre" : "Al filo de las sombras",
    "Puntuacion" : 8,
    "Precio" : 20.7,
    "genero" : "Fantasia-epica",
    "autor" : "Brent Weeks",
    "paginas" : 576,
    "ISBN" : 9788499893983,
    "ValorCalidad" : 38.64734299516908,
    "Descuento" : 1.9999999999999996
}
```

g) (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 : 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.

Solución

```
db.createCollection("PorNivel");
db.PorNivel.insert({NomCal: "nivel_1", Componentes: []});
db.PorNivel.insert({NomCal: "nivel_2", Componentes: []});
db.PorNivel.insert({NomCal: "nivel_3", Componentes: []});
db.PorNivel.insert({NomCal: "nivel_4", Componentes: []});
```

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

Solución

```
db.aficiones.find().forEach(
        function(doc) {
                var valor=doc.Puntuacion * 10;
                        if (valor <= 30) {
                                db.PorNivel.update(
                                        { NomCal: "nivel_1" },
                                        { $push: { Componentes: { Nombre: doc.Nombre, valorCalidad: valor } } }
                                )
                        }
                else if (valor <= 50) {
                        db.PorNivel.update(
                                { NomCal: "nivel_2" },
                                { $push: { Componentes: { Nombre: doc.Nombre, valorCalidad: valor } } }
                        )
                }
                else if (valor <= 70) {
                        db.PorNivel.update(
                                { NomCal: "nivel_3" },
                                { $push: { Componentes: { Nombre: doc.Nombre, valorCalidad: valor } } }
                        )
                }
                else if (valor <= 100) {
                        db.PorNivel.update(
                                { NomCal: "nivel_4" },
                                { $push: { Componentes: { Nombre: doc.Nombre, valorCalidad: valor } } }
                        )
                }
        }
);
```

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

Solución

Comando

```
db.PorNivel.find().forEach(
        function (doc) {
                for( i = 0; i <  3 ; ++i ) {
                        for( i = 0; i <  doc.Componentes.length ; ++i ) {
                                printjson("Nombre: "+ doc.Componentes[i].Nombre+ " calidad:
"+doc.Componentes[i].valorCalidad);
                        }
                        print("--------------------------------------------------------------------------");
                }
        }
);
```

Salida

```
"Nombre: CAU Metropolitano calidad: 30"
"Nombre: Alcala R.C calidad: 10"
"Nombre: Cuentame calidad: 30"
"Nombre: Kiko Rivera calidad: 20"
"Nombre: Abraham Mateo calidad: 0"
--------------------------------------------------------------------------
"Nombre: Quijote Rugby calidad: 50"
"Nombre: Club de Rugby Ingenieros Industriales Las Rozas calidad: 40"
"Nombre: Rayo Vallecano calidad: 50"
"Nombre: Katy Perry calidad: 50"
"Nombre: Drake calidad: 40"
"Nombre: El jardin de las delicias calidad: 50"
"Nombre: El caminante sobre el mar de nubes calidad: 50"
--------------------------------------------------------------------------
"Nombre: Rivas R.C calidad: 70"
"Nombre: XV Hortaleza calidad: 70"
"Nombre: Olimpico Rugby Club calidad: 70"
"Nombre: Call Of Duty Black Ops 3 calidad: 70"
"Nombre: Medal of Honor Warfighter calidad: 60"
"Nombre: World of Warships calidad: 70"
"Nombre: Uncharted 4 calidad: 70"
"Nombre: La sombra del viento calidad: 60"
"Nombre: Harry Potter: La piedra filosofal calidad: 60"
"Nombre: Los juegos del hambre 1 calidad: 70"
"Nombre: Leicester City calidad: 70"
"Nombre: Cafe Quijano calidad: 60"
"Nombre: La noche estrellada calidad: 70"
"Nombre: El beso calidad: 60"
"Nombre: La persistencia de la memoria calidad: 70"
"Nombre: El hijo del hombre calidad: 70"
"Nombre: El 3 de mayo en Madrid calidad: 70"
"Nombre: Venus del espejo calidad: 60"
--------------------------------------------------------------------------
"Nombre: El temor del hombre sabio calidad: 82"
"Nombre: El nombre del viento calidad: 79"
"Nombre: El camino de las sombras calidad: 83"
"Nombre: Mas allá de las sombras calidad: 83"
"Nombre: Al filo de las sombras calidad: 80"
"Nombre: El prisma negro calidad: 71"
"Nombre: La daga de la ceguera calidad: 74"
"Nombre: El ojo fragmentado calidad: 80"
"Nombre: Corazón inquieto: la vida de San Agustin calidad: 84"
"Nombre: Cristiano Ronaldo calidad: 90"
"Nombre: Sergio Ramos calidad: 80"
"Nombre: Torrejon R.C. calidad: 100"
```

"Nombre: San Isidro R.C calidad: 100"
"Nombre: Rugby Guadalajara calidad: 80"
"Nombre: Juego de Tronos calidad: 100"
"Nombre: Arrow calidad: 72.5"
"Nombre: GTA V calidad: 100"
"Nombre: Red Dead Redemption calidad: 90"
"Nombre: Battlefield 4 calidad: 90"
"Nombre: Need for Speed calidad: 90"
"Nombre: Rome II Total War calidad: 80"
"Nombre: Mafia II calidad: 80"
"Nombre: El resplandor calidad: 90"
"Nombre: La llamada Cthulhu calidad: 80"
"Nombre: Marina calidad: 90"
"Nombre: Odio calidad: 90"
"Nombre: Se lo que estas pensando calidad: 80"
"Nombre: Real Madrid calidad: 100"
"Nombre: FC Barcelona calidad: 100"
"Nombre: Bayern Munich calidad: 100"
"Nombre: Manchester United calidad: 80"
"Nombre: Juventus calidad: 90"
"Nombre: Paris Saint-Germain calidad: 80"
"Nombre: Machester City calidad: 80"
"Nombre: Allen Iverson calidad: 100"
"Nombre: Stephen Curry calidad: 95"
"Nombre: Carmelo Anthony calidad: 100"
"Nombre: Kobe Bryant calidad: 85"
"Nombre: Maluma calidad: 100"
"Nombre: Tom Odell calidad: 100"
"Nombre: Nicky Jam calidad: 90"
"Nombre: Maroon 5 calidad: 90"
"Nombre: Muse calidad: 80"
"Nombre: La gran ola de Kanagawa calidad: 80"
"Nombre: La tentación de San Antonio calidad: 80"
"Nombre: La tentación de San Antonio calidad: 80"
---

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

Solución

Comando

```
db.PorNivel.aggregate(
  [
  { $project: { _id:0 } } ,
  {"$unwind": "$Componentes"},
  { $sort : { "Componentes.Precio" : 1} },
  { $limit: 5 }
  ]
).pretty()
```

Salida


```
{
	"NomCal" : "nivel_1",
	"Componentes" : {
		"Nombre" : "Cuentame",
		"valorCalidad" : 30,
		"Precio" : 0
	}
}
{
	"NomCal" : "nivel_3",
	"Componentes" : {
		"Nombre" : "World of Warships",
		"valorCalidad" : 70,
		"Precio" : 0
	}
}
{
	"NomCal" : "nivel_3",
	"Componentes" : {
		"Nombre" : "Cafe Quijano",
		"valorCalidad" : 60,
		"Precio" : 1.5659148744891003
	}
}
{
	"NomCal" : "nivel_2",
	"Componentes" : {
		"Nombre" : "Katy Perry",
		"valorCalidad" : 50,
		"Precio" : 2.8211572588491007
	}
}
{
	"NomCal" : "nivel_2",
	"Componentes" : {
		"Nombre" : "Rayo Vallecano",
		"valorCalidad" : 50,
		"Precio" : 3.1381059608999995
	}
}
```

5. Elimina las 2 aficiones más caras de cada intervalo (NomCal). Puedes hacer una operación separada para cada intervalo.

Solución

Comando

```
var cursor=db.PorNivel.aggregate(
{ $unwind: "$Componentes" } ,
{ $sort: { "Componentes.Precio":-1 } } ,
{ $group: { _id : "$_id" , Componentes: { $push: "$Componentes" } } }
);
cursor.next();
var doc =cursor.next();
while(cursor.hasNext()){
        db.PorNivel.update({ },
        { $pull: { Componentes:{$in:  [cursor.Componentes] } }},
        {multi:true}
        );
        doc = cursor.next();
}
```

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
b) Crea las operaciones necesarias para poner los 5 mejores elementos de acuerdo al criterio de calidad explicado en el apartado 3.g., **no** hace falta que hagas el apartado 3.g.: calcula el nivel de calidad y asígna su valor en *aficiones*
c) Inserta un elemento a mano.
d) Lista todos los componentes para comprobar que mantiene los último cinco introducidos

Solución

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

b)

Comando

```
var cursor=db.PorNivel.aggregate(
{ $unwind: "$Componentes" } ,
{ $sort: { "Componentes.Precio":-1 } } ,
{ $group: { _id : "$_id" , Componentes: { $push: "$Componentes" } } }
);
cursor.next();
var doc =cursor.next();
while(cursor.hasNext()){
        db.superGuai.insert(doc);

        doc = cursor.next();
}
```

c) db.superGuai.insert({NomCal: "Nivel_3", Componentes: []});

d)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.

Solución

He diseñado una base de datos relacionado con el ámbito empresarial, la idea es de una empresa que están formadas por equipos de trabajo/profesion, donde cada equipo de trabajo estará formado por personas correspondiente a esa profesion.

({Nombre: "EmpresaABD",Provincia:"Madrid",Pais:"Spain", Equipo: [

Informaticos: [{ Nombre: "Pablo", Edad: 26, Correo:"pablo@hotmail.com"},{ Nombre: "Juan", Edad: 33,Correo:"pablo@hotmail.es" },{ Nombre: Maria, Edad: 26, Correo:"maria@hotmail.com"}]],

Fisicos: [{ Nombre: "Daniel", Edad: 29,Correo:"daniel@hotmail.com" },{ Nombre: "David", Edad: 33, Correo:"david@hotmail.com"},{ Nombre: "Ramon", Edad: 41,correo:"ramon@hotmail.com" }]]

La mayor parte de las operaciones van a ser de consultas, y los objetos serán objetos compuestos(Empresa), que contendrá otros objetos(equipo).

Descripciones de las operaciones

Operaciones de consulta:

Consultar sobre el nombre de la empresa

Consultar sobre el país que pertenece la empresa

Consultar sobre la profesión de la persona de un equipo

Consultar sobre la edad de la persona de un equipo

Consulta sobre el correo electrónico de la persona de un equipo

Operaciones de agregar

Operaciones de agregar una nueva empresa

Operaciones de agregar una nuevo equipo o nueva profesión

Operaciones de agregar una nueva persona a un equipo

Operaciones de eliminación

Eliminar empresa

Eliminar un equipo

Eliminar una persona de un equipo

Operaciones de modificación

Modificar el nombre de una empresa

Modificar el correo de una persona

APARTADO 5.- EXTRA

Deseamos introducir elementos compuestos, ej.: como en un equipo de futbol 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?

Solución

Convendría desnormalizar, porque los objetos(equipo de futbol) son objetos compuestos, es decir queremos incluir objetos(datos del jugador de futbol) dentro de otros objetos(equipo de futbol), que en este caso el objeto compuesto es el equipo de futbol, que a partir del objeto "equipo de futbol" podemos obtener los datos de cada jugador.

Esta forma de planificación de modelo de datos "Desnormalización de documentos", provoca que en una sola consulta podamos obtener todas las propiedades de ese objeto.

Una representación de la colección "Equipos de futbol" podría ser lo siguiente:

Equipo de Futbol

-Nombre:

-Pais

-Provincia

-Jugadores(Array):
Nombre,Edad,Estatura,Nacionalidad,Peso,Posicion,GolesLigamGolesChampion, GolesCopaRey

Ejemplo

({Nombre: "Barcelona",Provincia:"Barcelona",Pais:"Spain", Jugadores: [

{ Nombre: Messi, Edad: 31,Estatura:170,Nacionalidad: Argentina,Peso:68,Posicion:Delantero,GolesLiga:24, GolesChampion:11,GolesCopaRey:9 },

{ Nombre: Luis Suarez, Edad: 33,Estatura:180,Peso:73,Nacionalidad,Posicion:Delantero,GolesLiga:19, GolesChampion:5,GolesCopaRey:4 },…

]});