

# Proyecto MongoDB:



Nombre: Pablo Jesús

Apellidos: Ramos Fernández

Curso: 1º ASIR

Asignatura: Gestión de Base de Datos

Curso escolar: 2021/2022

## Mis consultas:

```
db.coleccionGrupos.find({$and:[{$fechaLanzamiento:{$gte: new Date ("1969-01-01")}},  
{fechaLanzamiento:{$lt: new Date("1980-01-01")}}]},{$fechaLanzamiento:1, nombreGrupo:1}  
).pretty()
```

En esta consulta vamos a mostrar los grupos de rock de principios de los años 70 y finales del 79 por medio de los operadores \$and, \$gte y \$lt y mostrará la fecha al completo por medio del método “new Date”. Para ser más preciso, nos mostrará el nombre del grupo y su fecha de inicio.

```
db.coleccionGrupos.find({$and:[{nombreGrupo:{$regex: /^The/i}},  
{pais: {$eq: "Londres, Inglaterra, Reino Unido"}}]}).pretty()
```

Vamos a mostrar en esta consulta aquellos grupos de rock que empiecen por las iniciales “The” sin tener en cuenta si está escrito “THE”, “the” y diferentes variantes. Y que pertenezcan a Londres ,Inglaterra, Reino Unido mediante los operadores \$regex(/^texto-/i) y el operador \$eq.

```
db.coleccionGrupos.find({$or:[{genero: {$ne:"Blues rock"}},{enActivo: {$eq: true}} ]}).pretty()
```

En esta otra nos va a mostrar los grupos cuyo género no sea “Blues rock” ó que sigan haciendo actuaciones, grabando discos, celebrando conciertos, etc. Mediante los operadores \$or, \$ne y \$eq.

```
db.coleccionGrupos.find({$and:[{web:{$exists:false}},{genero:{$eq:"Hard rock"}}]}).pretty()
```

Aquí nos mostrará aquellos grupos de rock que estén un poco “chapado a la antigua” ya que no tienen página web y que su género musical sea igual a “Hard rock”. Para ello hacemos uso de los operadores \$and, \$exists y \$eq.

```
db.coleccionGrupos.find({"discografia.unidadesVendidas": {$elemMatch: {$gte: 200, $lt:  
1500000}}},{nombreGrupo:1, "discografia.unidadesVendidas":1}).pretty()
```

En esta consulta veremos los operadores \$elemMatch, \$gte, \$lt. Mediante la selección de aquellos de los grupos que hayan vendido 200 unidades o más, ó menos de 1500000 de unidades, pero siempre que un grupo tenga alguno de esos dos valores, nos lo mostrará. Para afinar más en los criterios de la consulta he filtrado por el nombre del grupo y dentro de documento, el array de unidades vendidas.

```
db.coleccionGrupos.find({$and:[{web:{$regex: /es$/}},{pais:{$ne: "Londres, Inglaterra, Reino  
Unido"}},]},{$pais:1, nombreGrupo:1, web:1}).pretty()
```

Aquí nos da como resultado aquellos grupos que su extensión de la web finalice en “.es” y que su país de origen no sea Londres, Inglaterra, Reino Unido. Los operadores usados han sido \$and, \$regex(/-texto-\$/) y \$ne. Para que nos muestre los valores consultados he filtrado por el país, nombre del grupo y su web.

```
db.coleccionGrupos.find({precioEntradaConcierto: {$eq: 0.00}},{nombreGrupo:1,  
precioEntradaConcierto:1}).pretty()
```

Para esta nueva consulta vamos a encontrar aquellos grupos en que la entrada del concierto sea 0€. Para ello hacemos uso de \$eq.

```
db.coleccionGrupos.find({pais:{$regex: /^E/, $options: 'm'}},{pais:1, nombreGrupo:1}).pretty()
```

Nos muestra aquellos grupos de rock en que su nombre comience por la letra “E”, como por ejemplo que su origen sea España. Para ello el operador que he utilizado han sido \$regex(/^-texto-/,\$option: ‘m’)

```
db.coleccionGrupos.find( {$and: [{fechaLanzamiento:{$gte: new Date ("1980-01-01")}},{$and: [
{precioEntradaConcierto: { $gt : 10 } }, { precioEntradaConcierto : { $lt: 40 } } ]},{ $and: [
{ enActivo: true },{ "discografia.unidadesVendidas": { $lt : 500 } } ] }},{fechaLanzamiento:1,
precioEntradaConcierto:1, enActivo:1, "discografia.unidadesVendidas":1}).pretty()
```

En esta consulta nos muestra los grupos cuyo fecha de lanzamiento sea mayor o igual que la indicada y que el precio de la entrada del concierto sea mayor que 10€ y menor que 40€ y que aún sigan haciendo giras y que el número de unidades vendidas sea menor de 500 uds. Los operadores usados han sido: \$and, \$gte, \$gt y \$lt.

```
db.coleccionGrupos.find( {$and: [{fechaLanzamiento:{$gte: new Date ("1980-01-01")}},{$or: [
{precioEntradaConcierto: { $gt : 10 } }, { precioEntradaConcierto : { $lt: 40 } } ]},{ $or: [
{ enActivo: true },{ "discografia.unidadesVendidas": { $lt : 500 } } ] }},{fechaLanzamiento:1,
precioEntradaConcierto:1, enActivo:1, "discografia.unidadesVendidas":1}).pretty()
```

En esta otra consulta parecida a la que hemos visto anteriormente, pero no igual, nos muestra los grupos cuyo fecha de lanzamiento sea mayor o igual que la indicada y que el precio de la entrada del concierto sea mayor que 10€ ó menor que 40€ y que aún sigan haciendo giras ó que el número de unidades vendidas sea menor de 500 uds. Los operadores usados han sido: \$and, \$or, \$gte, \$gt y \$lt.

```
db.coleccionGrupos.find( {"discografia.unidadesVendidas": { $in: [ 500, 1500000 ] } },
{nombreGrupo:1, "discografia.unidadesVendidas":1}).pretty()
```

Aquí nos mostrará las unidades de discos vendidas que estén comprendidas entre 500 y 1500000. Para ello he usado el operador \$in. Para ser más concreto he filtrado por el nombre del grupo y dentro de la discografía por las unidades vendidas.

```
db.coleccionGrupos.find({ $nor:[ { enActivo: true },{ "discografia.unidadesVendidas": { $lt : 500 } } ] }, {enActivo:1, "discografia.unidadesVendidas":1}).pretty()
```

En esta consulta nos visualizará los grupos que o no estén activos ya o que los álbumes que hayan vendido sea inferior a 500 unidades. Operadores usados: \$nor y \$lt.

```
db.coleccionGrupos.find( { $and: [{genero: { $not : {$regex: /^Hard/i} } },
{ precioEntradaConcierto : { $not: { $gt: 30.00 } } } ] }, {genero:1, precioEntradaConcierto:1,
nombreGrupo:1}).pretty()
```

Esta consulta nos muestra los grupos de rock que no contengan en su género musical la cadena “Hard” y que el precio de la entrada al concierto sea inferior a 30€. Para ello he usado \$and, \$not y \$regex(/^-texto-/). Para que no nos muestre todas las propiedades de los grupos he filtrado por el género, el precio de la entrada del concierto y el nombre del grupo.

```
db.coleccionGrupos.find({año: {$eq: new Date("1988-05-11").getFullYear()}}, {nombreGrupo:1, año:1, fechaLanzamiento:1}).pretty()
```

Nos muestra aquellos grupos que empezaron sus andadas en el mundo musical en una fecha en concreto mediante los operadores \$eq y getFullYear().

```
db.coleccionGrupos.find( { discografica: { $nin: [ "Warner Music", "EMI", "Columbia Records" ] } }, {discografica:1,nombreGrupo:1}).pretty()
```

Encuentra aquellos grupos de rock que sus discográficas no sean las indicadas mediante el operador \$nin.

```
db.coleccionGrupos.find({ genero: { $all: [ "Hard rock", "Pop rock", "Blues rock" ] } }, {genero:1, nombreGrupo:1}).pretty()
```

Nos muestra los grupos que en su género musical contengan “Hard rock”, “Pop rock” y “Blues rock”, pero si tienen esos tres y algunos otro más, también nos lo muestra.

```
db.coleccionGrupos.find({gira: { $size: 4 }}, {nombreGrupo:1, gira:1}).pretty()
```

Visualizaremos los grupos que hayan realizado un total de 4 giras en total. Usaremos el operador \$size y filtro por los campos nombreGrupo y gira.

## **Aportación personal:**

```
db.coleccionGrupos.find({pais: {$eq: "Londres, Inglaterra, Reino Unido"},$expr: { $lt: [0.5, {$rand: {} } ] } }, { nombreGrupo: 1, pais:1 }).pretty()
```

Este es uno de los métodos que he realizado por cuenta propia y muestra de los grupos que sean de dicho país , la mitad de ellos. Operadores usados: \$eq, \$expr, \$lt y \$rand.

```
db.coleccionGrupos.find( { $expr: { $gt: [ "$numeroFans" , "$numeroManagers" ] } }, {nombreGrupo:1, numeroFans:1, numeroManagers:1}).pretty()
```

Otro de las consultas realizadas autónomamente ha sido que nos muestre de entre el número de fans y el número de managers, nos visualice aquellos grupos en que el número de fans sea superior al número de personal técnico que forma el grupo.

```
var bulk = db.coleccionGrupos.initializeUnorderedBulkOp();
```

```
bulk.insert({ nombreGrupo: "Blues Cal Experience", precioEntradaConcierto: 7.55, discografica: "DOMI Music"});
```

```
bulk.insert({ nombreGrupo: "Los niños del Ventorrillo", precioEntradaConcierto: 20.00, discografica: "Pruna Records"});
```

```
bulk.insert({ nombreGrupo: "Borneo", precioEntradaConcierto: 2.00, discografica: "DOMI Music"});
```

```
bulk.execute();
```

Esto es otra variedad para insertar nuevos grupos sin la necesidad de hacer uso de insertOne o insertMany. Haremos del operador bulk. La primera línea inicializa la variable bulk. Acto seguido insertaremos los nuevos grupos. Una vez añadidos todos los requeridos, ejecutaremos ese bloque de datos.