

# Proyecto MongoDB-Mongo Atlas

## 2º Trimestre:



Nombre: Pablo Jesús  
Apellidos: Ramos Fernández  
Curso: 1º ASIR  
Asignatura: Gestión de Base de Datos

## Mis consultas:

```
db.coleccionVideoClub.aggregate([{$group:{_id: "$dependiente", numeroVentas: {$count: {}},}},  
{ $sort:{ numeroVentas:1}} ] )
```

En esta consulta vamos a comprobar el numero de ventas realizada por cada uno de los dependientes que trabaja en el videoclub ordenados de manera ascendente.

```
db.coleccionVideoClub.aggregate([{$unwind: {path: "$peliculas",}},{$group:{_id:  
"$dependiente",mediaDependiente: {$avg:{$multiply: ["$peliculas.precioAlquiler",  
"$peliculas.stock"],}}}},{$sort:{numeroVentas:-1}}])
```

Realiza la media de los precios de las películas alquiladas, cuya venta ha sido realizada por los diferentes dependientes y ordenadas de manera descendente.

```
db.coleccionVideoClub.aggregate([{$unwind:{ path:"$datosAbonado"}},{$group:  
{_id:"$datosAbonado.nombreCliente", alquileresRealizados:  
{$sum:"$datosAbonado.alquileres"}}, {$sort:{alquileresRealizados:1}}])
```

En esta nueva consulta vamos a mostrar el número de películas alquiladas por cada cliente.

```
db.coleccionVideoClub.aggregate([{$unwind: { path: "$peliculas", }, },{$project: {dependiente: 1,  
diferenciaAlquileresVentas: {$subtract: [{ $multiply:  
["$peliculas.precioVenta", "$peliculas.stock"],},{$multiply:  
["$peliculas.precioAlquiler", "$peliculas.stock"]}}}},{$group: { _id: "$dependiente",totalLiquido:  
{$sum: "$diferenciaAlquileresVentas",}} }])
```

Realiza la suma de los fondos recaudados de los alquileres y ventas de las películas realizadas por cada uno de los dependientes que conforman el videoclub.

```
db.peliculas.aggregate([{$count: "peliculas.nombrePeliculas"}])
```

Cuentas la diversidad de películas de las que consta el videoclub.

```
db.peliculas.aggregate([{$group:{_id: "$nombrePelicula"}},{$project:{_id: 0, nombrePelicula:  
"$_id",}},{$count: "nombrePelicula"}])
```

Contabiliza el número de películas de las que dispone nuestro videoclub montando incluso su nombre.

```
db.peliculas.aggregate([{$group:{_id: "$nombrePelicula", numeroPelis: {$sum: 1}}},  
{$project: {_id: 0,nombrePelicula: "$_id",numeroPelis: 1}}])
```

Muestras los los nombres de las películas y el número de unidades de las que el videoclub dispone.

```
db.clientes.aggregate([{$count: "dependiente"}])
```

Número de ventas realizadas por cada dependiente que conforman el videoclub.

```
db.clientes.aggregate([ { $group: { _id: "$dependiente", numeroVentasRealizadas: { $sum: 1 } } },  
{ $project: { _id: 0, nombreDependiente: "$_id", numeroVentasRealizadas: 1 } } ] )
```

Número de ventas realizadas por los vendedores del videoclub, pero diferenciando entre los distintos dependientes de la tienda.

```
db.clientes.aggregate([ { $sortByCount: "$dependiente" }, { $project: { _id: 0, nombreDependiente:  
"$_id", numeroVentasRealizadas: "$count" } } ] )
```

Número de ventas realizadas por los vendedores del videoclub, pero diferenciando entre los distintos dependientes de la tienda. (usando el operador \$sortByCount)

```
db.clientes.aggregate([ { $sortByCount: "$datosAbonado.nombreCliente" }, { $project: { _id: 0,  
nombreCliente: "$_id", numeroAlquileresRealizados: "$count" } } ] )
```

Número de alquileres realizados por cada uno de los clientes, mostrando tanto el número de alquileres como el nombre de los diferentes clientes.

```
db.peliculas.aggregate([ { $lookup: { from: 'clientes', localField: 'id', foreignField: '_id', as: 'abonados'  
} }, { $addFields: { abonadosPelis: { $arrayElemAt: [ '$abonados', 0 ] } } }, { $project: { abonados: 0 }  
} ].pretty()
```

Agrupar los datos de las películas con los de los clientes correspondientes que alquilaron o compraron dichas películas.

## Aportación personal:

```
db.peliculas.aggregate([ { $bucketAuto: { groupBy: "$precioAlquiler", buckets: 4 } } ] )
```

En la siguiente operación, los documentos de entrada se agrupan en cuatro cubos según los valores del campo precio alquiler, mostrando un límite inferior y un límite superior.

```
db.peliculas.aggregate([ { $unset: [ "detalles.director" ] } ] ).pretty()
```

Elimina el campo que le especifiquemos en mi caso dentro de detalles: director.

```
db.peliculas.aggregate([ { $group: { _id: "$nombrePelicula", precioMaximoUnidadesPrecioVenta:  
{ $max: { $multiply: [ "$precioVenta", "$stock" ] } }, precioMaximoPelicula: { $max:  
"$precioVenta" } } } ] )
```

Muestra el nombre de la película junto a dinero generado por las ventas de todas las películas.

```
db.clientes.aggregate([ { $project: { varianzaNumeroAbonado: { $pow: [ { $stdDevPop:  
"$datosAbonado.numeroAbonado" }, 2 ] } } } ] )
```

Calcula la varianza al número de Abonado del cliente.

## División de la base de datos : (Cliente y Películas)

```
db.coleccionVideoClub.aggregate([ { $project: {"películas":0}}, {$out:
{db:"videoclub",coll:"clientes"}}]])
```

Separa de la colección VideoClub la parte de los datos del abonado junto con el correspondiente dependiente.

```
db.coleccionVideoClub.aggregate([{$unwind: "$películas"},{$project: { id: 1, películas: 1,
_id:0}}, {$project: {id: 1, nombrePelicula: "$películas.nombrePelicula", precioAlquiler:
"$películas.precioAlquiler", precioVenta: "$películas.precioVenta", disponibilidad:
"$películas.disponibilidad", stock: "$películas.stock", detalles: "$películas.detalles"}},
{$out:{ db:"videoclub",coll:"películas"}}]])
```

Por otro lado haremos un \$out para separar en esta parte las diferentes películas con sus respectivos detalles y datos técnicos.