

UD3.T6 Ejercicio Agregaciones

Crear una BD competicion.

Crear una colección llamada pruebas con los siguientes datos:

```
{nombre:"Carlos", mes:"Marzo", distKm:6, tiempoMin:42}
{nombre:"Elena", mes:"Marzo", distKm:10, tiempoMin:60}
{nombre:"Carlos", mes:"Marzo", distKm:2, tiempoMin:12}
{nombre:"Elena", mes:"Marzo", distKm:10, tiempoMin:61}
{nombre:"Carlos", mes:"Abril", distKm:5, tiempoMin:33}
{nombre:"Elena", mes:"Abril", distKm:42, tiempoMin:285}
{nombre:"Angel", mes:"Abril", distKm:5, tiempoMin:33}
```

Hacer las siguientes operaciones usando agregaciones:

- Calcular el número de pruebas que ha realizado cada corredor.

```
competicion> db.pruebas.aggregate([{$group:{_id:"$nombre",nPruebas:{$sum:1}}}]])
[
  { _id: 'Carlos', nPruebas: 3 },
  { _id: 'Elena', nPruebas: 3 },
  { _id: 'Angel', nPruebas: 1 }
]
```

- Calcular el número de pruebas que ha realizado cada corredor por mes.

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes:"$mes"},nPruebas:{$sum:1}}}]])
[
  { _id: { nombre: 'Elena', mes: 'Marzo' }, nPruebas: 2 },
  { _id: { nombre: 'Carlos', mes: 'Marzo' }, nPruebas: 2 },
  { _id: { nombre: 'Angel', mes: 'Abril' }, nPruebas: 1 },
  { _id: { nombre: 'Carlos', mes: 'Abril' }, nPruebas: 1 },
  { _id: { nombre: 'Elena', mes: 'Abril' }, nPruebas: 1 }
]
```

- Calcular el número total de Kms recorridos por cada corredor.

```
competicion> db.pruebas.aggregate([{$group:{_id:"$nombre",nKms:{$sum:"$distKm"}}}]])
[
  { _id: 'Carlos', nKms: 13 },
  { _id: 'Elena', nKms: 62 },
  { _id: 'Angel', nKms: 5 }
]
```

- Calcular los kilómetros que corre cada uno de media al mes.

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes:"$mes"},nKmsAvg:{$avg:"$distKm"}}}]])
[
  { _id: { nombre: 'Elena', mes: 'Marzo' }, nKmsAvg: 10 },
  { _id: { nombre: 'Carlos', mes: 'Marzo' }, nKmsAvg: 4 },
  { _id: { nombre: 'Angel', mes: 'Abril' }, nKmsAvg: 5 },
  { _id: { nombre: 'Carlos', mes: 'Abril' }, nKmsAvg: 5 },
  { _id: { nombre: 'Elena', mes: 'Abril' }, nKmsAvg: 42 }
]
```

- Visualizar las distintas distancias que ha corrido cada corredor en las pruebas.

```
competicion> db.pruebas.aggregate([{$group:{_id:"$nombre",distancias:{$addToSet:"$distKm"}}}])
[
  { _id: 'Carlos', distancias: [ 6, 2, 5 ] },
  { _id: 'Elena', distancias: [ 10, 42 ] },
  { _id: 'Angel', distancias: [ 5 ] }
]
```

- Calcular el número medio de pruebas por corredor al mes (se cuenta el número de pruebas por persona y mes y a continuación se hace la media de este dato).

Idea:

- Sabemos contar el número de pruebas por mes.
- Sabemos hacer la media de unos valores ¡usemos 2 etapas!

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes:"$mes"},nPruebas:{$sum:1}}},{$group:{_id:'$_id.nombre',mediaAlMes:{$avg:'$nPruebas'}}}])
[
  { _id: 'Elena', mediaAlMes: 1.5 },
  { _id: 'Carlos', mediaAlMes: 1.5 },
  { _id: 'Angel', mediaAlMes: 1 }
]
```

- Visualizar en cada mes qué distancias se han hecho en alguna sesión. (Si una distancia se ha corrido varias veces en ese mes debe aparecer varias veces).

```
competicion> db.pruebas.aggregate([{$group:{_id:{mes:"$mes"},distancias:{$push:"$distKm"}}}])
[
  { _id: { mes: 'Abril' }, distancias: [ 5, 42, 5 ] },
  { _id: { mes: 'Marzo' }, distancias: [ 6, 10, 2, 10 ] }
]
```

- Obtener la mayor y menor distancia recorrida por cada corredor.

```
competicion> db.pruebas.aggregate([{$group:{_id:"$nombre", "minDist":{$min:"$distKm"}, "maxDist":{$max:"$distKm"}}}])
[
  { _id: 'Carlos', minDist: 2, maxDist: 6 },
  { _id: 'Elena', minDist: 10, maxDist: 42 },
  { _id: 'Angel', minDist: 5, maxDist: 5 }
]
```

- Queremos disponer de los datos de distancias recorridas en millas, sabiendo que una milla = 1,60934 km

```
competicion> db.pruebas.aggregate([{$group:{_id:"$nombre", distanciaMillas:{$push:{$multiply:[$distKm,1.60934]}}}])
[
  { _id: 'Elena', distanciaMillas: [ 16.0934, 16.0934, 67.59228 ] },
  { _id: 'Angel', distanciaMillas: [ 8.0467 ] },
  { _id: 'Carlos', distanciaMillas: [ 9.65604, 3.21868, 8.0467 ] }
]
```

- Obtener la media en kilómetros mensuales de cada corredor, pero solo para aquellos valores medios que sean superiores a 5km.

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes:"$mes"},nKmsTotal:{$sum:"$distKm"}},{$group:{_id:"$_id.nombre",nKmsMedia:{$avg:"$nKmsTotal"}},{$match:{nKmsMedia:{$gt:5}}}],{$group:{_id:"$_id.nombre",nKmsMedia:{$avg:"$nKmsTotal"}}}])
[ { _id: 'Elena', nKmsMedia: 31 }, { _id: 'Carlos', nKmsMedia: 6.5 } ]
```

- Calcular la media de kilómetros por corredor y mes, ordenado por mes.

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes:"$mes"},nKmsTotal:{$sum:"$distKm"}},{$group:{_id:"$_id.nombre",nKmsMedia:{$avg:"$nKmsTotal"}},{$sort:nKmsMedia}],{$group:{_id:"$_id.nombre",nKmsMedia:{$avg:"$nKmsTotal"}}}])
[
  { _id: 'Elena', nKmsMedia: 31 },
  { _id: 'Carlos', nKmsMedia: 6.5 },
  { _id: 'Angel', nKmsMedia: 5 }
]
```

- Mostrar el corredor que tiene mayor media absoluta.

```
competicion> db.pruebas.aggregate([{$group: {_id: {nombre: "$nombre", mes: "$mes"}, nKmsTotal: {$sum: "$distKm"} }}, {$group: {_id: "$_id.nombre", nKmsMedia: {$avg: "$nKmsTotal"} }}, {$sort: {nKmsMedia: -1}}
,{$limit:1}}]
[ { _id: 'Elena', nKmsMedia: 31 } ]
```

- Calcula la media de pruebas realizadas por cada persona mensualmente.

Idea: Redirige la salida de una agrupación creando una nueva colección. No muestra nada en la salida, pero puedes comprobarlo haciendo un find sobre la nueva colección de salida.

```
competicion> db.pruebas.aggregate([{$group: {_id: {nombre: "$nombre", mes: "$mes"}, nKmsTotal: {$sum: "$distKm"} }}, {$group: {_id: "$_id.nombre", nKmsMedia: {$avg: "$nKmsTotal"} }}, {$out: "Pruebas2"}])
```

- Creamos una nueva colección en la BD llamada gustos, con los siguientes datos:

```
{nombre:"Carlos", aficiones:["siesta","cine"]}
{nombre:"Elena", aficiones:["correr","cine"]}
{nombre:"Ana", aficiones:["viajar","cine"]}
{nombre:"Angel", aficiones:["correr","montaña", "cine"]}
```

- Queremos saber el número de personas con el que cuenta cada afición.

Idea: Cuando tenemos documentos que contienen un array y queremos agrupar por valores del array, a veces conviene eliminar los arrays y convertirlos en múltiples documentos. En realidad estamos "normalizando" (primera forma normal). Y agrupar sobre esos documentos.

```
competicion> db.gustos.aggregate([{$unwind: "$aficiones"}, {$group: {_id: "$aficiones", suma: {$sum: 1}} }])
[
  { _id: 'viajar', suma: 1 },
  { _id: 'siesta', suma: 1 },
  { _id: 'montaña', suma: 1 },
  { _id: 'cine', suma: 4 },
  { _id: 'correr', suma: 2 }
]
```