Crear una base de datos PRUEBA y dentro de ella, crear una colección llamada CICLOS

```
use PRUEBA
db.createCollection("CICLOS")
```

Insertar los siguientes 4 registros

```
db.ciclos.insertMany([
 {
  "_id": "1",
  "Nombre": "DAM",
  "Grupo": 1,
  "NumAlumnos": 2,
  "Nivel": "Medio",
  "Alumnos": [
   {
    "Nombre": "Carlos",
    "Edad": 23
   },
   {
    "Nombre": "Sandra",
    "Edad": 25
   }
  ]
 },
 {
  "_id": "2",
  "Nombre": "DAW",
  "Grupo": 1,
  "NumAlumnos": 2,
```

```
 "Nivel": "Medio",
  "Alumnos": [
    {
      "Nombre": "Fernando",
      "Edad": 23
    },
    {
      "Nombre": "Julian",
      "Edad": 28
    }
  ]
},
{
  "_id": "3",
  "Nombre": "ASIR",
  "Grupo": 1,
  "NumAlumnos": 2,
   "Nivel": "Bajo",
  "Alumnos": [
    {
      "Nombre": "Jose Carlos",
      "Edad": 20
    },
    {
      "Nombre": "Isabel",
      "Edad": 24
    }
```

```
    ]
  },
  {
    "_id": "4",
    "Nombre": "DAM",
    "Grupo": 2,
    "NumAlumnos": 2,
    "Nivel": "Alto",
    "Alumnos": [
      {
        "Nombre": "Juan",
        "Edad": 22
      },
      {
        "Nombre": "Carmen",
        "Edad": 18
      }
    ]
  }
]);
```

1. **Mostrar todos los datos de la colección ciclos.**

```
db.ciclos.find()
```

2. **Contar todos los datos**

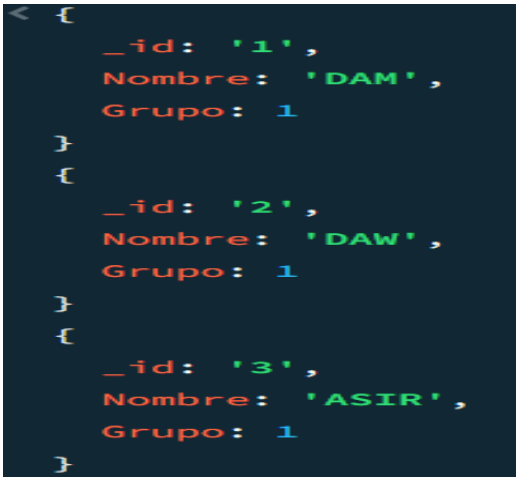
```
< 4
```

```
db.ciclos.countDocuments()
```

3. **Mostrar solo los ciclos de 2º.**

```
db.ciclos.find({"Grupo" : 2})
```

4. **Mostrar solo los ciclos de 1º pero solo el nombre y el grupo**



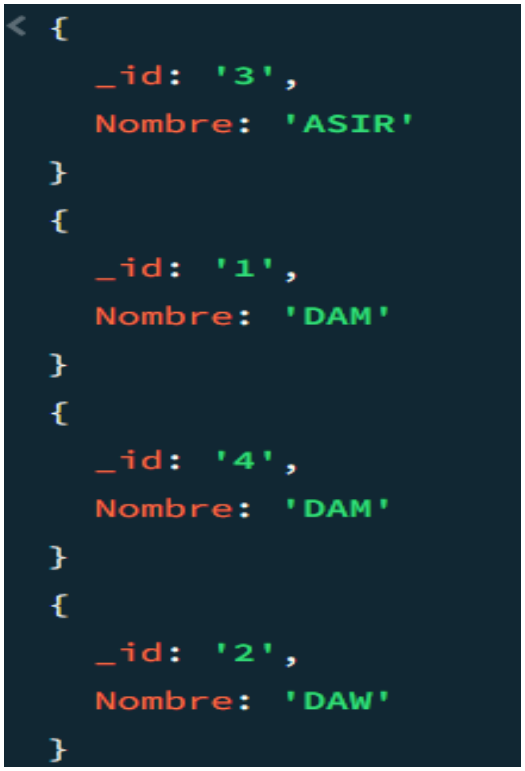

```
db.ciclos.find({ Grupo: 1 },{ Nombre: 1, Grupo: 1, _id: 0 })
db.ciclos.find({ "Grupo": 1 },{ Nombre: 1, Grupo: 1, _id: 0 })
```

5. Mostrar de los ciclos de 1º los que sean de Nivel Medio

```
db.ciclos.find({ "Grupo": 1 , Nivel : "Medio"})
```

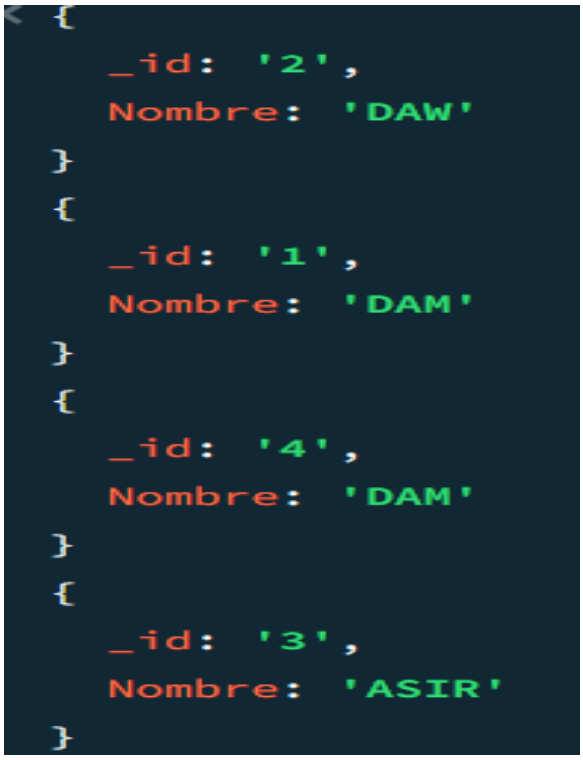
```
{
  _id: '2',
  Nombre: 'DAW',
  Grupo: 1,
  NumAlumnos: 2,
  Nivel: 'Medio',
  Alumnos: [
    {
      Nombre: 'Fernando',
      Edad: 23
    },
    {
      Nombre: 'Julian',
      Edad: 28
    }
  ]
}
```

6. Ordenar descendentemente por nombre



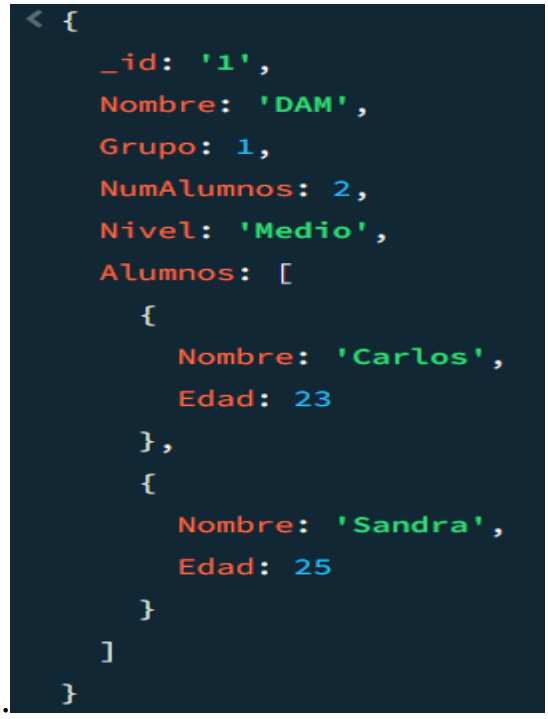

```
db.ciclos.find({},{ Nombre: 1, _id: 1 }).sort({ Nombre: 1 })
```

7. **Ordenar ascendentemente por nombre**



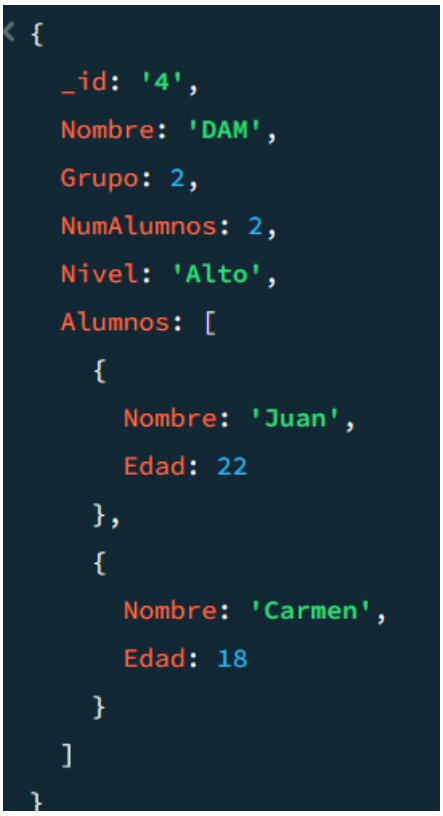
db.ciclos.find({},{ Nombre: 1, _id: 1 }).sort({ Nombre: -1 })

8. **Mostrar solo el primer registro de la colección ciclos.**



db.ciclos.findOne()

9. **Mostrar solo el ultimo registro de la colección ciclos.**




```
db.ciclos.find().sort({ _id: -1 }).limit(1)
```

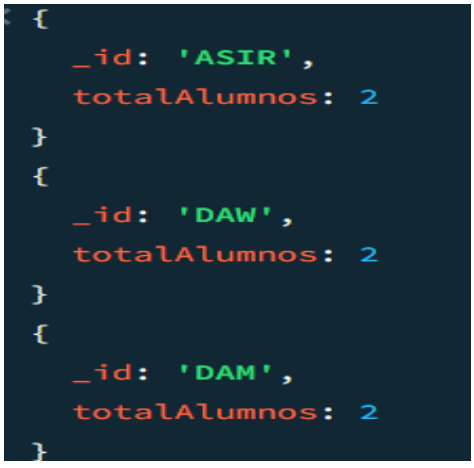
AGREGACIONES (SIMILAR GROUP BY Y HAVING EN SQL)

Se usa para realizar operaciones avanzadas y permite realizar cálculos, filtrados y agrupaciones.

- **$match**: Filtra los documentos según ciertos criterios, similar a una cláusula WHERE en SQL.
- **$group**: Agrupa los documentos por un campo específico y permite realizar operaciones de agregación, como contar, sumar o calcular promedios.
- **$sort**: Ordena los documentos según uno o más campos.
- **$project**: Modifica la forma de los documentos, permitiendo incluir o excluir campos específicos y crear nuevos campos.
- **$limit**: Limita el número de documentos devueltos.
- **$skip**: Omite un número específico de documentos en la salida.

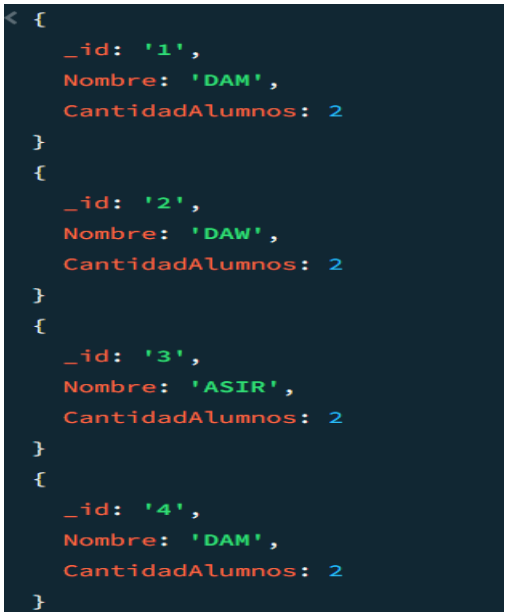
Ejemplo: Sacar los documentos de ciclos (ordenado por el total de alumnos en orden descendente) incluyendo solo aquellos del grupo 1, agrupando los resultados por el nombre del ciclo y sumando el número de alumnos, y finalmente

```
db.ciclos.aggregate([
  { $match: { Grupo: 1 } }, // Filtra solo los grupos 1
  { $group: { _id: "$Nombre", totalAlumnos: { $sum: "$NumAlumnos" } } }, // Agrupa por Nombre y suma NumAlumnos
  { $sort: { totalAlumnos: -1 } } // Ordena por totalAlumnos de forma descendente
]);
```



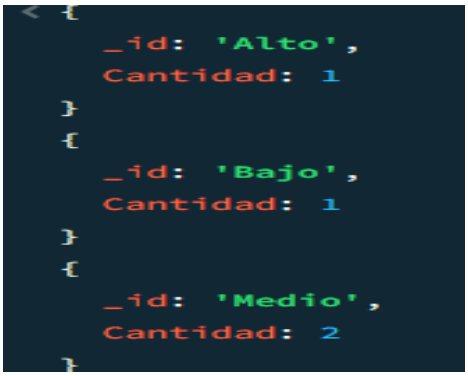
10.Contar cuántos alumnos hay en cada ciclo usando agregación**:

```
db.ciclos.aggregate({ $group: { _id: "$_id", Nombre: { $first: "$Nombre" }, CantidadAlumnos: { $sum: "$NumAlumnos" }}})
```



11.Mostrar el número de ciclos por nivel:

```
db.ciclos.aggregate({ $group: { _id:  "$Nivel" , Cantidad: { $sum: 1 }}})
```



12.- Mostrar el promedio de edad de los alumnos por ciclo:

```
db.ciclos.aggregate([
  {$unwind: "$Alumnos"},
  {$group: {_id: "$Nombre", PromedioEdad: { $avg: "$Alumnos.Edad" }}}
])
```