

[2 PUNTOS] CREAR Y POBLAR LA BASE DE DATOS

/*NOTA: al declarar variables Compass no tiene problema, pero al pegar todo de golpe puede fallar, especialmente con las primeras instrucciones: use, dropDatabase() y createCollection(""). */

/* Limpia la consola - en el resto de scripts se omitirá por comodidad */

cls

/* Creamos la BD. */

use MDBElecciones

/* 1) Borramos la BD. */

db.dropDatabase()

/* 2) Volvemos a crear la BD después de borrarla. */

use MDBElecciones

/* 3) Creamos la colección "Partidos". */

db.createCollection("Partidos")

/* 4.1) Creamos primero los votos de cada partido como array para poder insertarlos en el partido correspondiente. Cada voto en una línea del array de votos correspondiente. */

```
var votos_p01 = [  
    {"id_dig": "F6E8D2C", "circunscripcion": "Impunícia", "gratificacion": 500},  
    {"id_dig": "3A7F9B4", "circunscripcion": "Impunícia", "gratificacion": 1000},  
    {"id_dig": "H5I4J3K", "circunscripcion": "Impunícia"},  
    {"id_dig": "3A7F9B4", "circunscripcion": "Impunícia"}  
]
```

```
var votos_p02 = [  
    {"id_dig": "5X7W9Y4", "circunscripcion": "Impunícia"},  
    {"id_dig": "P1Q3R5S", "circunscripcion": "Ladronia"},  
    {"id_dig": "C1D2E3F", "circunscripcion": "Ladronia"},  
    {"id_dig": "G6H7I8J", "circunscripcion": "Ladronia"},  
    {"id_dig": "Y2X4Z6W", "circunscripcion": "Ladronia"},  
    {"id_dig": "Z5A4B3C", "circunscripcion": "Impunícia"},  
    {"id_dig": "J5K4L3M", "circunscripcion": "Ladronia"},  
]
```

```
var votos_p04 = [  
    {"id_dig": "1A2B3C4", "circunscripcion": "Impunícia", "gratificacion": 1500},  
    {"id_dig": "7G8H9I0", "circunscripcion": "Ladronia", "gratificacion": 1000},  
    {"id_dig": "J1K2L3M", "circunscripcion": "Impunícia"},  
    {"id_dig": "R7S8T9U", "circunscripcion": "Ladronia", "gratificacion": 1200},  
    {"id_dig": "4D5E6F7", "circunscripcion": "Ladronia"},  
    {"id_dig": "A1B2C3D", "circunscripcion": "Impunícia", "gratificacion": 700},  
    {"id_dig": "N4O5P6Q", "circunscripcion": "Ladronia", "gratificacion": 800},  
    {"id_dig": "V1W2X3Y", "circunscripcion": "Impunícia"}  
]
```

```
var votos_p05 = [
  {"id_dig": "Z1X4C7V", "circunscripcion": "Impunícia", "gratificacion": 100},
  {"id_dig": "O1P2Q3R", "circunscripcion": "Impunícia", "gratificacion": 50},
  {"id_dig": "S7T8U9V", "circunscripcion": "Impunícia", "gratificacion": 200},
  {"id_dig": "K0L9M8N", "circunscripcion": "Ladronia", "gratificacion": 1000},
  {"id_dig": "T4U5V6W", "circunscripcion": "Impunícia", "gratificacion": 100},
  {"id_dig": "X1Y2Z3A", "circunscripcion": "Ladronia", "gratificacion": 150}
]
```

/* 4.2) Creamos cada partido con el array de votos correspondiente. El partido p03 no tiene.*/

```
var p01 = {"_id": "p01", "siglas": "ASIR", "nombre": "Alianza Social de Izquierda Radical", "votos": votos_p01}
```

```
var p02 = {"_id": "p02", "siglas": "DAM", "nombre": "Democracia, Acción y Modernidad", "votos": votos_p02,
"afiliados": 6000}
```

```
var p03 = {"_id": "p03", "siglas": "DAW", "nombre": "Diversidad, Ambiente y Wifi gratis", "afiliados": 4000}
```

```
var p04 = {"_id": "p04", "siglas": "PHT", "nombre": "Partido de la Honestidad y la Transparencia", "votos": votos_p04,
"afiliados": 12000}
```

```
var p05 = {"_id": "p05", "siglas": "PRO", "nombre": "Partido del Robo Organizado", "votos": votos_p05, "afiliados": 8000}
```

/* 4.3) Insertamos los partidos en la colección de la base de datos */

```
db.Partidos.insertMany([p01,p02,p03,p04,p05])
```

/* 5) Listamos todos los campos de todos los registros ordenados por las siglas del partido */

```
db.Partidos.find().sort({"siglas": 1})
```

[2 PUNTOS] ACTUALIZAR LA INFORMACIÓN

1)

```
/* Utilizamos la BD */
use MDBElecciones

/* Creamos las variables necesarias para filtros, proyecciones y updates */
/* Partidos que muestren afiliados */
var j_cond_exists = {"afiliados":{"$exists:true}}

/* Partidos que NO muestren afiliados */
var j_cond_not_exists = {"afiliados":{"$exists:false}}

/* Partidos con MENOS de 5000 afiliados */
var j_cond_caso1 = {"afiliados":{"$lt:5000}}

/* Partidos ENTRE 5000 y 10000 afiliados (mayor o igual que 5000 y menor o igual que 10000) */
var j_cond_caso2 = {"and":[{"afiliados":{"$gte:5000"}},{"afiliados":{"$lte:10000"}}]}

/* Partidos con MÁS de 10000 afiliados */
var j_cond_caso3 = {"afiliados":{"$gt:10000}}

/* Un filtro para cada tipo de subvencion (básicamente que muestre afiliados y una condición para cada caso) */
/* Muestran afiliados y son menos de 5000 (usando AND lógica) */
var j_filtro1 = {"and":[j_cond_exists,j_cond_caso1]}

/* Muestran afiliados y están entre 5000 y 10000 (usando AND lógica) */
var j_filtro2 = {"and":[j_cond_exists,j_cond_caso2]}

/* Muestran afiliados y son más de 10000 (usando AND lógica) */
var j_filtro3 = {"and":[j_cond_exists,j_cond_caso3]}

/* Se incluye el campo "subvencion" y se actualiza a 0€ */
db.Partidos.updateMany(j_cond_not_exists, {"$set":{"subvencion":0}})

/* Se incluye el campo "subvencion" y se actualiza a 1000000€ */
db.Partidos.updateMany(j_filtro1, {"$set":{"subvencion":1000000}})

/* Se incluye el campo "subvencion" y se actualiza a 3000000€ */
db.Partidos.updateMany(j_filtro2, {"$set":{"subvencion":3000000}})

/* Se incluye el campo "subvencion" y se actualiza a 10000000€ */
db.Partidos.updateMany(j_filtro3, {"$set":{"subvencion":10000000}})
```

2)

```
/* Ahora se muestran los partidos y sus subvenciones mediante una proyección */
/* Proyeccion: muestra nombre y subvención (omite el "_id") */
var j_proyeccion = {"nombre":1,"subvencion":1,"_id":0}

/* Muestra el nombre del partido y la subvención ordenados por subvención de mayor a menor (no tiene filtro) */
db.Partidos.find({},j_proyeccion).sort({"subvencion":-1})
```

[3 PUNTOS] MINERÍA DE DATOS

1)

```
/* Comenzamos siempre asegurando que usamos la BD */  
use MDBElecciones
```

```
/* separamos el array de votos para tratarlo */  
var stage_separar = {$unwind:"$votos"}
```

```
/* Que tengan gratificacion y sea mayor que 500€ */  
var j_filtro = {$and:[{"votos.gratificacion":{$exists:true}},{"votos.gratificacion":{$gt:500}}]}
```

```
/* aplicamos el filtro al stage del match */  
var stage_filtro = {$match:j_filtro}
```

```
/* creamos la proyeccion (consulta a prueba de cambios, en lugar de "votos":1) */  
var j_proyeccion = {"_id":0,"siglas":1,"votos.id_dig":1,"votos.circunscripcion":1,"votos.gratificacion":1}
```

```
/* stage de proyeccion */  
var stage_proyeccion = {$project:j_proyeccion}
```

```
/* ordenamos por gratificacion de menor a mayor */  
var stage_sort = {$sort:{"siglas":1,"votos.gratificacion":1}}
```

```
/* para mostrar las siglas y los datos de los votos necesitamos una agregación con los stage que hemos definido */  
db.Partidos.aggregate([stage_separar,stage_filtro,stage_proyeccion,stage_sort])
```

2)

```
/* separamos el array de votos para tratarlo – o podemos usar el mismo unwind de antes */  
var stage_separar = {$unwind:"$votos"}
```

```
/* proyeccion del total gastado por partido (siglas) */  
var j_proyeccion = {"_id":0, "siglas":1, "total_gastado":{$sum:"$votos.gratificacion"}}
```

```
/* stage de la proyeccion */  
var stage_proyeccion = {$project:j_proyeccion}
```

```
/* ordenamos por coste total (de mayor a menor) */  
var j_sort = {"total_gastado":-1}
```

```
/* stage del sort */  
var stage_sort = {$sort:j_sort}
```

```
/* limitamos el resultado a 1 (el que más) con el stage limit */  
var stage_limit = {$limit:1}
```

```
/* mediante una agregación, proyectamos, ordenamos y mostramos el partido con más gasto en gratificaciones */  
db.Partidos.aggregate([stage_proyeccion, stage_sort, stage_limit])
```

[2 PUNTOS] CONTADORES

```
/* Comenzamos siempre asegurando que usamos la BD */  
use MDBElecciones  
  
/* Creamos las distintas condiciones del filtro para contar los partidos que lo cumplan */  
/* No tiene afiliados (no los muestra) */  
var j_cond_a1 = {"afiliados":{"$exists:false}}  
  
/* No tienen afiliados (=0) */  
var j_cond_a2= {"afiliados":{"$eq:0}}  
  
/* No tienen afiliados (cumple alguna de las dos anteriores, OR lógica) */  
var j_cond_afiliados = {$or:[j_cond_a1,j_cond_a2]}  
  
/* Tiene votos (no importa si están vacíos, bienvenido a Corrupoly) */  
var j_cond_v1 = {"votos": { $exists: true}}  
  
/* No tiene gratificacion (que te paguen 0 es una gratificación en Corrupoly) */  
var j_cond_v2 = {"votos.gratificacion": { $exists: false }}  
  
/* Cumple ambas condiciones a la vez (AND lógica) */  
var j_cond_votos = {$and:[j_cond_v1,j_cond_v2]}  
  
/* Filtro con todas las condiciones (OR lógica) */  
var j_filtro = {$or:[j_cond_afiliados,j_cond_votos]}  
  
/* Contamos cuántos partidos cumplen con las condiciones */  
db.Partidos.countDocuments(j_filtro);
```

[1 PUNTO] BORRADO

1)

```
/* Nos aseguramos de usar la BD */  
use MDBElecciones
```

```
/* Proyección de siglas y nombres */  
var j_proyec = {"_id":0, "siglas":1, "nombre":1}
```

```
/* El partido NO tiene votos */  
var j_cond_votos = {"votos":{"$exists:false}}
```

```
/* No votados en Ladronia */  
var j_cond_ladronia = {"votos.circunscripcion":{"$ne:"Ladronia"}}
```

```
/* No tienen votos o no han sido votados en Ladronia */  
var j_filtro = {"$or":[j_cond_votos,j_cond_ladronia]}
```

```
/* Listamos las siglas y los partidos que cumplen el filtro */  
db.Partidos.find(j_filtro,j_proyec)
```

2)

```
/* Borramos los partidos que cumplen con el filtro anterior */  
db.Partidos.deleteMany(j_filtro)
```