Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
(6549) Bases de Datos NoSQL

PROYECTO PRÁCTICO FASE I:

Star Wars – Episodio I:
El Vacuum Fantasma

Profesora: Mercy Ospina
Auxiliar Docente: Juan Pablo Rivas

Estudiantes:
Rafael Contreras Pimentel, C.I.: 30.391.915
Rafael Contreras Agudelo, C.I.: 28.484.629
Yarima Contreras Blanco, C.I.: 29.706.291

Caracas, 06 de julio de 2025

Creación de Esquemas: Consideraciones

Para la creación de los esquemas requeridos se tomaron las siguientes consideraciones:

- Planeta (planet): Para la creación de este esquema, el tipo de dato de la población fue establecido de manera que sea aceptado tanto ***int*** como ***long***. Esto es debido a que al existir planetas que cuentan con una población muy elevada puede suceder que el valor completo sobrepase el límite permitido por ***int***, por lo cual se debe usar un tipo de dato que soporte un valor más alto.

- Especie (specie): En el esquema especie se mantienen los campos solicitados, sin hacer referencia a ningún otro. Dado que las especies están relacionadas tanto a planeta como a personaje, se decide mantener la colección y utilizarla mediante referenciación.

- Facción (faction): En la creación de este esquema, se trata al nombre del líder como un simple ***string*** y no como un personaje referenciado.

- Locaciones (location): Este esquema referencia a planeta en su campo con el mismo nombre. Además, se asumieron las coordenadas como un subdocumento comprendiendo latitud y longitud.

- Armas (weapon): Para este esquema, se toma al fabricante como un simple ***string*** ya que el fabricante no es necesariamente un personaje. Por otro lado, dado que su información se utiliza únicamente al relacionarse con personajes, se ha decidido embeber estos datos dentro de la colección de Personajes, para obtener los resultados rápidamente y estudiar la aplicación del caso.

- Vehículos (vehicle): Al igual que en el esquema de **Armas**, se toma al fabricante como un simple ***string*** ya que, nuevamente, el fabricante no es necesariamente un personaje. Esta colección se utiliza como información adicional, debido a que no presenta campos relacionados con las demás.

- Personajes (character): El esquema correspondiente a la colección de Personajes posee gran riqueza relacional, tomándose los campos planeta, especie y la lista de

facciones como referencias a las colecciones con sus mismos nombres. Asimismo, para establecer la relación entre arma y personaje, se ha embebido la información de las armas dentro de cada documento.

- Naves Espaciales (spaceship): Para el esquema de Naves Espaciales, se ha considerado al fabricante como un ***string***, ya que, al igual que otros esquemas, el fabricante no es necesariamente un personaje. Mientras tanto, las referencias presentes corresponden a piloto, quien sí debe ser un personaje, y a las facciones relacionadas.

- Películas (movie): Para este esquema, el director es establecido como un simple ***string*** y no como un personaje ya que el director no es necesariamente un personaje. A su vez, presenta relaciones embebidas al almacenar también datos de los personajes y naves espaciales involucradas.

- Eventos históricos (historical-event): El esquema de eventos históricos presenta como relación referenciada el campo correspondiente a película, mientras que las facciones, localidades y personajes que hicieron vida en dicho evento, guardan su información de manera embebida dentro de esta colección.

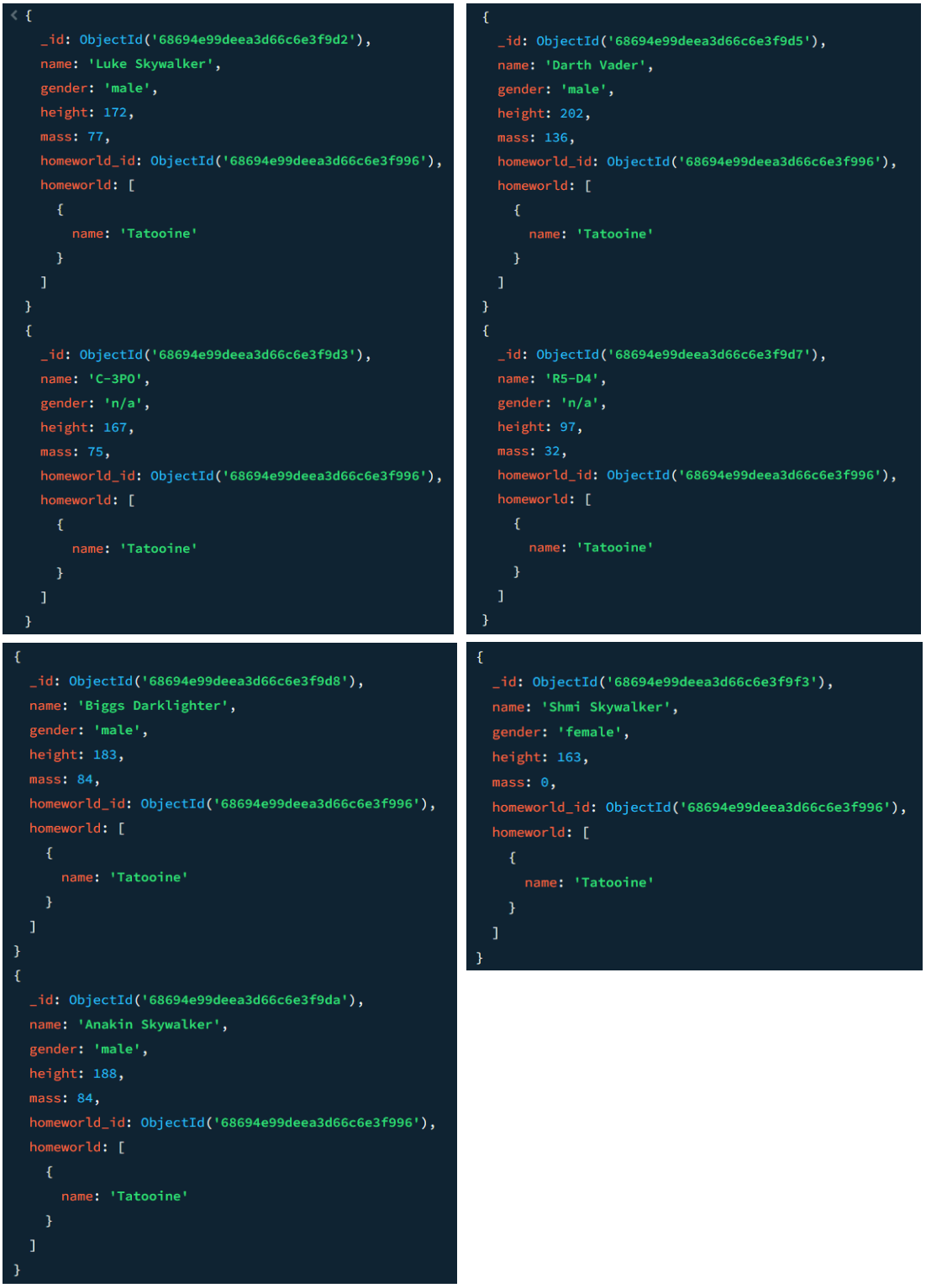
Consultas: Requerimientos originales

1. **Obtener los personajes de Tatooine**

   Para obtener de manera efectiva a todos los personajes cuyo planeta natal (homeworld) sea “Tatooine”, en primer lugar, se limitan los campos a estudiar, de modo que los Guardianes del Conocimiento eviten distracciones, luego, se filtran aquellos cuyo homeworld.name sea el indicado.

```
db.character.aggregate([
  {
   $lookup: {
     from: "planet",
     localField: "homeworld_id",
     foreignField: "_id",
     as: "homeworld"
   }
  },
  {
   $match: {
     "homeworld.name": { $in: ["Tatooine"] }
   }
  },
  {
   $project: {
     _id: 1,
     name: 1,
     gender: 1,
     height: 1,
     mass: 1,
     homeworld_id: 1,
     "homeworld.name": 1
   }
  }
])
```

   Resultado obtenido:



2. Encontrar naves espaciales pilotadas por humanos de la Alianza Rebelde

Debido a que se trabajó con un esquema y datos en inglés, la especie a buscar es “Human” y la facción “Rebel Alliance”. Así, es necesario iniciar uniendo la colección de naves espaciales (spaceship) con la de personajes (character), gracias al id coincidente en el campo piloto (pilot_id). Luego, es necesario descomponer el array

resultante, acción realizada gracias a $unwind, resultado con el cual se procede con otra unión, esta vez para detectar a aquellos pilotos cuya especie sea humana (Human), asimismo, se realiza el mismo proceso, pero verificando que la facción a la que pertenece sea la Alianza Rebelde (Rebel Alliance).

```
db.spaceship.aggregate([
  {
   $lookup: {
     from: "character",
     localField: "pilot_id",
     foreignField: "_id",
     as: "pilot"
   }
  },
  { $unwind: "$pilot" },
  {
   $lookup: {
     from: "specie",
     localField: "pilot.species_id",
     foreignField: "_id",
     as: "species"
   }
  },
  { $unwind: "$species" },
  {
   $lookup: {
     from: "faction",
     localField: "pilot.faction_ids",
     foreignField: "_id",
     as: "factions"
   }
  },
  {
   $match: {
     "species.name": "Human",
     "factions.name": { $in: ["Rebel Alliance"] }
   }
  },
  {
   $project: {
     name: 1,
     model: 1,
     "pilot.name": 1,
     "species.name": 1,
     "factions.name": 1
   }
  }
])
```

Resultado obtenido:

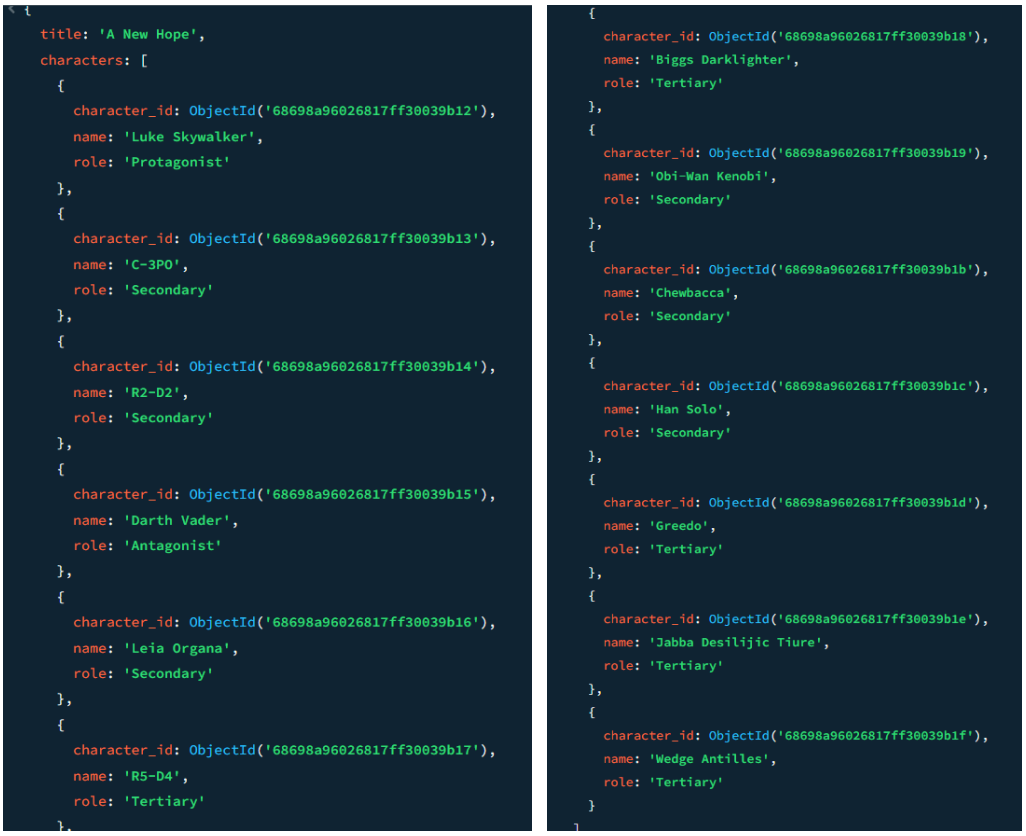
3. **Obtener los personajes que participaron en la película "Una Nueva Esperanza"**

Debido al esquema en inglés, el nombre de la película es “A New Hope”. Para obtener estos datos, la Inteligencia Espacial debe agrupar filtrar el nombre de la película y listar los personajes que posee embebidos en la colección. Tal como se muestra a continuación.

```
db.movie.aggregate([
 {
  $match: { title: "A New Hope" }
 },
 {
  $project: {
    _id: 0,
    title: 1,
    characters: 1
```

```
    }
  }
])
```

Resultado obtenido:



4. **Obtener los nombres de todos los personajes que han tenido en su poder sables de luz con un color de cristal igual al de “Mace Windu”**

   El primer paso para obtener esta información debe ser determinar el color del cristal del sable utilizado por el icónico Mace Windu. Para ello, se filtran los personajes, utilizando su nombre y se accede a la información del arma embebida, extrayendo el color de su cristal.

   Con ese dato, se busca entre los documentos con el objetivo de hallar coincidencias de sables y de color, para finalmente mostrar los nombres de los personajes resultantes.

```
db.character.aggregate([
  {
    $match: { name: "Mace Windu" }
  },
  {
    $project: {
      _id: 0,
```

```
    color: "$weapon.crystal_color"
   }
  },
  {
   $lookup: {
    from: "character",
    let: { targetColor: "$color" },
    pipeline: [
     {
      $match: {
       $expr: {
        $and: [
         { $eq: ["$weapon.name", "Lightsaber"] },
         { $eq: ["$weapon.crystal_color", "$$targetColor"] }
        ]
       }
      }
     },
     {
      $project: { _id: 0, name: 1 }
     }
    ],
    as: "matches"
   }
  },
  {
   $unwind: "$matches"
  },
  {
   $replaceRoot: { newRoot: "$matches" }
  }
])
```

Resultado obtenido:

```
< {
    name: 'Mace Windu'
  }
NoSQLProy>
```

Nota adicional: Cabe resaltar que la decisión de embeber las armas dentro de personaje, también se vio influenciada por el objetivo de entrenar al Equipo de Inteligencia de la Galaxia para que sea capaz de obtener datos estelares de forma tanto referenciada como embebida.

5. **Nombre, nombre del planeta natal, nombre de la especie y altura de las Jedi que hayan participado en “El Imperio Contraataca” o que piloten una nave espacial de tipo “Caza estelar”**

   Con el objetivo de conseguir la información especificada, se deben seguir los siguientes pasos:
   Primero, se debe evaluar si el personaje aparece en la película “The Empire Stickes Back” (equivalente a “El Imperio Contraataca” en el esquema inglés), acción que se logra gracias al primer $lookup y se almacena en movie_participation.
   De manera similar, se verifica si el personaje ha sido piloto de alguna nave, de ser así, se guardan las naves correspondientes.
   Luego, sabiendo que se quiere obtener a los personajes femeninos afiliados a la facción “Jedi Order”, se agregan estas condiciones, así como la existencia de movie_participation o que la nave que pilotan sea de tipo “Starfighter”.
   Así, se limpian los resultados para sólo mostrar los campos deseados por los Guardianes del Conocimiento.

```
db.character.aggregate([
  {
   $lookup: {
     from: "movie",
     pipeline: [
       { $match: { title: "The Empire Strikes Back" } },
       { $project: { characters: 1 } },
       { $unwind: "$characters" },
       { $replaceRoot: { newRoot: "$characters" } }
     ],
     as: "movie_participation"
   }
  },
  {
   $lookup: {
     from: "spaceship",
     localField: "_id",
     foreignField: "pilot_id",
     as: "ships"
   }
  },
  {
   $lookup: {
     from: "faction",
     localField: "faction_ids",
     foreignField: "_id",
     as: "factions"
   }
  },
  {
   $match: {
     $and: [
       { "gender": "female" },
```

```
      { "factions.name": "Jedi Order" },
      {
       $or: [
        { "movie_participation.name": { $exists: true } },
        { ships: { $elemMatch: { class: "Starfighter" } } }
       ]
      }
     ]
    }
   },
   {
    $lookup: {
     from: "planet",
     localField: "homeworld_id",
     foreignField: "_id",
     as: "planet"
    }
   },
   { $unwind: "$planet" },
   {
    $lookup: {
     from: "specie",
     localField: "species_id",
     foreignField: "_id",
     as: "specie"
    }
   },
   { $unwind: { path: "$specie", preserveNullAndEmptyArrays: true } },
   {
    $project: {
     _id: 0,
     name: 1,
     height: 1,
     homeworld: "$planet.name",
     species: "$specie.name"
    }
   }
])
```

Resultado obtenido:


{
name: 'Ayla Secura',
height: 178,
homeworld: 'Ryloth',
species: "Twi'lek"
}
{
name: 'Adi Gallia',
height: 184,
homeworld: 'Coruscant',
species: 'Tholothian'
}
{
name: 'Luminara Unduli',
height: 170,
homeworld: 'Mirial',
species: 'Mirialan'
}


{
name: 'Barriss Offee',
height: 166,
homeworld: 'Mirial',
species: 'Mirialan'
}
{
name: 'Jocasta Nu',
height: 167,
homeworld: 'Coruscant',
species: 'Human'
}
{
name: 'Shaak Ti',
height: 178,
homeworld: 'Shili',
species: 'Togruta'
}

Consultas: Pipeline de Agregación

1. Cantidad de sables de cada color de cristal

Es bien sabido que el color de los cristales de los sables de luz tienen un importante significado simbólico y están asociados a distintos aspectos de la Fuerza y las filosofías Jedi y Sith. Por esta razón, se quiere que los Guardianes del Conocimiento analicen los sables de luz más utilizados, identifiquen los ideales más seguidos y tomen sus previsiones para garantizar la armonía.

Para lograr este resultado, es necesario filtrar a los personajes según el nombre de arma que poseen en el subdocumento respectivo, de forma que sea “Lightsaber”. Luego, se agrupan los documentos filtrados con base en el color del cristal (crystal_color) y se suman las ocurrencias.

```
db.character.aggregate([
  {
   $match: { "weapon.name": "Lightsaber" }
  },
  {
   $group: {
    _id: "$weapon.crystal_color",
    count: { $sum: 1 }
   }
  }
])
```

Resultado obtenido:

```
< {
    _id: 'Green',
    count: 7
  }
  {
    _id: 'Red',
    count: 3
  }
  {
    _id: 'Blue',
    count: 10
  }
  {
    _id: 'Purple',
    count: 1
  }
NoSQLProy >
```

2. **Personajes que pertenecen o pertenecieron a más de dos facciones**

Las facciones suelen representar ideologías o intereses opuestos. Es por esta razón que el Equipo de Inteligencia se ha interesado en conocer los personajes que pertenecen a más de dos facciones, de modo que puedan evaluar su nivel de lealtad.

Esta información se obtiene contando la cantidad de elementos (identificadores únicos) guardados en el arreglo de facciones del personaje. Así, se filtra para obtener sólo aquellos cuyo conteo sea mayor a dos.

```
db.character.aggregate([
  {
   $project: {
    name: 1,
    faction_count: { $size: "$faction_ids" }
   }
  },
  {
   $match: { faction_count: { $gt: 2 } }
  }
])
```

Resultado obtenido:

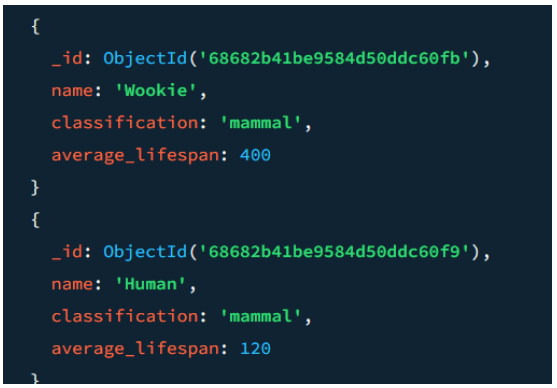
3. Top 5 especies con mayor longevidad

En la infinita galaxia, siempre hay algo desconocido y aspectos nuevos por aprender, tantos, que a veces los años de una vida no son suficientes para lograr todo. Las especies más longevas pueden acumular vastos conocimientos y experiencia, influenciando la historia a través de generaciones. De esta manera, los Guardianes del Conocimiento quieren recolectar los datos estelares de las 5 especies registradas con mayor tiempo de vida.

Dado que se quieren obtener los resultados cuyo tiempo promedio de vida (average_lifespan) sea mayor, entonces, el primer paso consiste en ordenar este dato de manera descendente. Así, es posible limitar los resultados a los 5 primeros.

```
db.specie.aggregate([
  {
    $sort: { average_lifespan: -1 }
  },
  {
    $project: {
      name: 1,
      classification: 1,
      average_lifespan: 1
    }
  },
  { $limit: 5 }
])
```

Resultado obtenido:



4. **Listar todas las facciones, su tipo y su líder que estuvieron involucradas con un rol antagonista en eventos históricos de la película “A New Hope”**

   Una Nueva Esperanza (A New Hope) es una película fundamental en el universo, debido a que es la primera estrenada, introduciendo toda la cultura y mitología de la galaxia. Por esta razón, se quiere saber cuáles fueron las facciones y líderes antagonistas, marcando conflicto en estos eventos tan importantes.

   Esta información se obtiene vinculando los eventos históricos con la película “A New Hope”, mediante un lookup que termina almacenando el resultado en el arreglo movie_match. Luego, se expande el arreglo de facciones, debido a que un mismo evento puede poseer varias facciones involucradas. Con esta descomposición, se filtran las facciones según su rol, tomando específicamente aquellas que correspondan a un papel antagonista. Así, se accede a la información de las facciones obtenidas, de modo que se muestren sólo los campos de interés.

```
db["historic-event"].aggregate([
 {
  $lookup: {
   from: "movie",
   let: { movieRef: "$movie_id" },
   pipeline: [
    {
     $match: {
      $expr: {
       $and: [
        { $eq: ["$title", "A New Hope"] },
        { $eq: ["$_id", "$$movieRef"] }
       ]
      }
     }
    },
    { $project: { _id: 1 } }
   ],
   as: "movie_match"
  }
 },
 {
  $match: {
   movie_match: { $ne: [] }
  }
 },
 { $unwind: "$factions" },
 {
  $match: {
   "factions.role": "Antagonist"
  }
 },
 {
  $lookup: {
```

```
      from: "faction",
      localField: "factions.faction_id",
      foreignField: "_id",
      as: "faction_info"
    }
  },
  { $unwind: "$faction_info" },
  {
    $project: {
      _id: 0,
      faction_name: "$factions.name",
      role: "$factions.role",
      type: "$faction_info.type",
      leader: "$faction_info.leader_name"
    }
  },
  {
    $group: {
      _id: {
        faction_name: "$faction_name",
        leader: "$leader"
      },
      role: { $first: "$role" },
      type: { $first: "$type" }
    }
  },
  {
    $project: {
      _id: 0,
      faction_name: "$_id.faction_name",
      leader: "$_id.leader",
      role: 1,
      type: 1
    }
  }
])
```

Resultado obtenido:

```
< {
    role: 'Antagonist',
    type: 'Government',
    faction_name: 'Galactic Empire',
    leader: 'Emperor Palpatine'
  }
NoSQLProy >
```

5. **Fabricantes de naves espaciales involucradas en películas que se estrenaron entre los años 1983 y 2002**

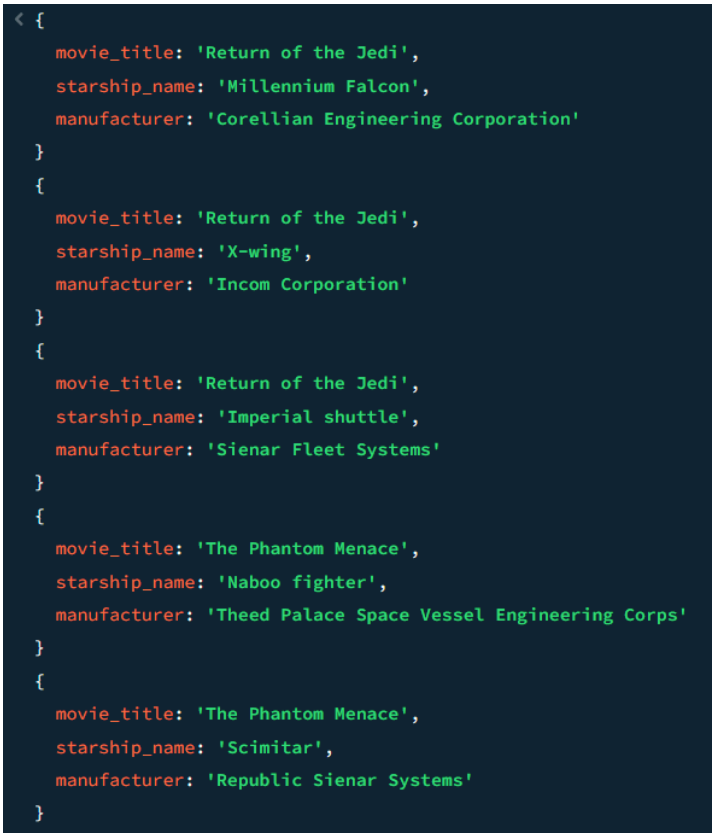
   Las naves espaciales son una herramienta sumamente necesaria para diversas tareas en la amplia galaxia. Por esta razón, se desea saber cuáles son los fabricantes de las naves que participaron en películas de cierta antigüedad, de modo que sea posible conocer un poco más su trayectoria y fortalecer la confianza en sus productos.

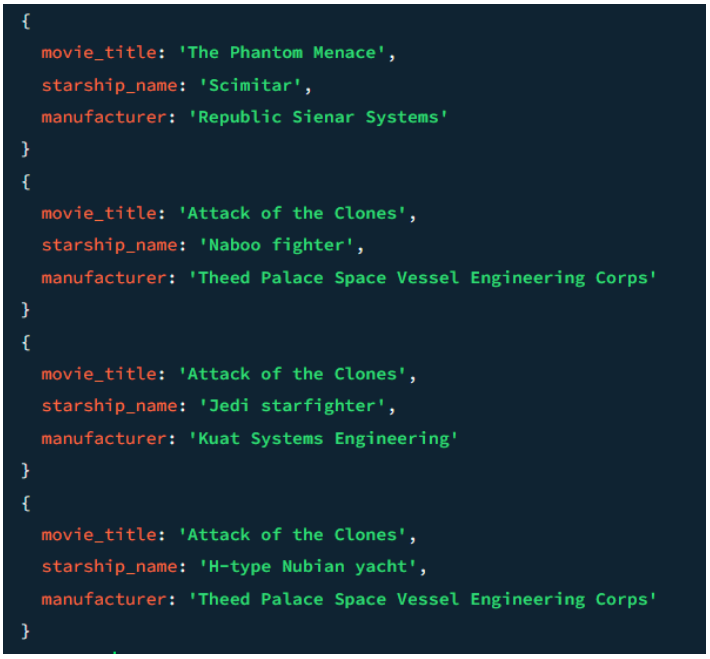
   Así, se buscan las películas cuyas fechas de estreno se encuentren entre inicios de 1983 y finales de 2002. Este resultado es descompuesto para obtener la información de las naves espaciales, tomando así el nombre del fabricante.

   Para facilitar a la Inteligencia Espacial la identificación de la información, se decidió mostrar como resultado la película, nave y fabricante de cada documento.

```
db.movie.aggregate([
  {
   $match: {
    release_date: {
     $gte: ISODate("1983-01-01T00:00:00Z"),
     $lte: ISODate("2002-12-31T23:59:59Z")
    }
   }
  },
  {
   $unwind: "$starships"
  },
  {
   $lookup: {
    from: "spaceship",
    localField: "starships.starship_id",
    foreignField: "_id",
    as: "ship"
   }
  },
  {
   $unwind: "$ship"
  },
  {
   $project: {
    _id: 0,
    movie_title: "$title",
    starship_name: "$ship.name",
    manufacturer: "$ship.manufacturer"
   }
  }
])
```

Resultado obtenido:





Consulta: Map Reduce

En la vasta galaxia existen muchos planetas y con ellos, infinidad de tipos de clima. Climas muy adversos ocasiona que su población viva situaciones complicadas y opten por viajar a otros planetas. Por esta razón, el Imperio Galáctico ha solicitado información a los Guardianes del Conocimiento para conocer la población total agrupada por cada tipo de clima.

De esta forma, el procedimiento para satisfacer esta consulta es el siguiente:

Primero se realiza la fase Map, para esto se recorre cada documento en la colección planet, verificando el valor de su población y emitiendo una clave/valor, en donde se almacene el clima y la población correspondiente:

```
map = function() {
  if (this.population) {
    emit(this.climate, this.population);
  }
};
```

Luego, es hora de la fase Reduce, en esta función se agrupan todos los valores emitidos con la misma clave, de forma que suma las poblaciones de todos los planetas que poseen el mismo clima.

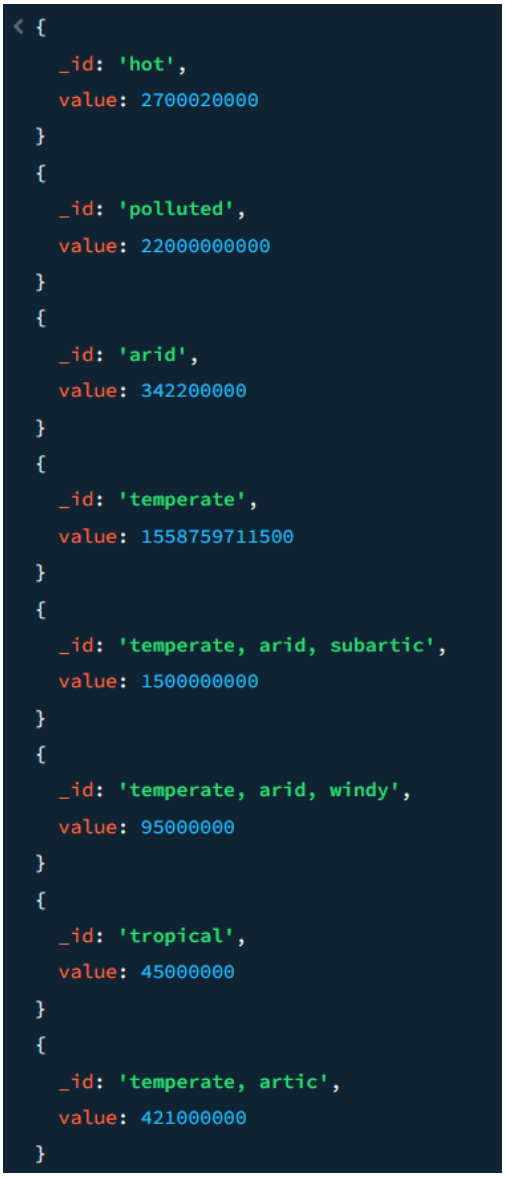
```
reduce = function(climate, populations) {
  return Array.sum(populations);
};
```

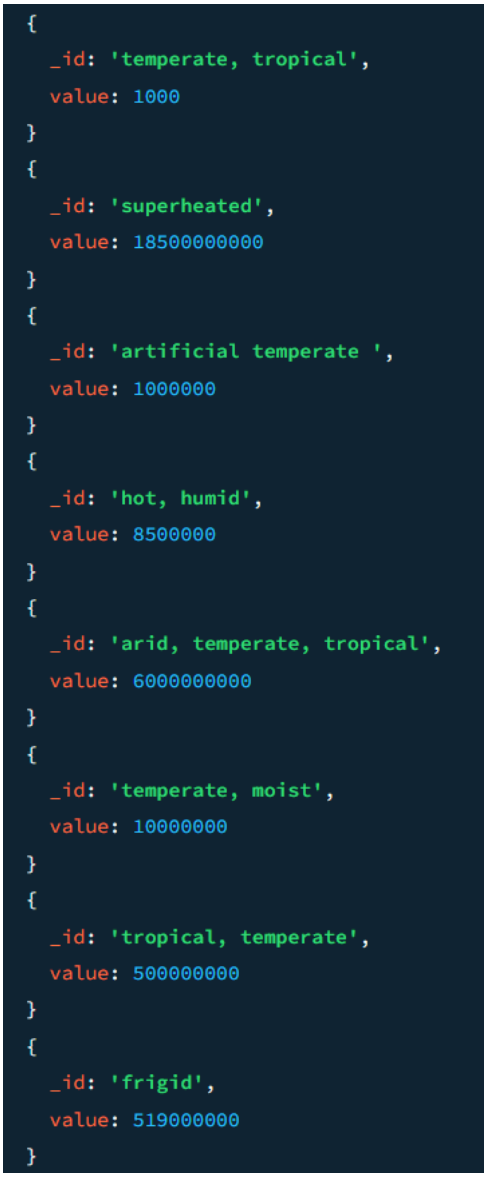
Por último, se crea una nueva colección, en donde se guarda el clima con su población acumulada.

```
db.planet.mapReduce(map, reduce, {
  out: "total_population_by_climate"
});
```

Visualización de resultados:

```
db.total_population_by_climate.find()
```


{
_id: 'hot',
value: 2700020000
}
{
_id: 'polluted',
value: 22000000000
}
{
_id: 'arid',
value: 342200000
}
{
_id: 'temperate',
value: 1558759711500
}
{
_id: 'temperate, arid, subartic',
value: 1500000000
}
{
_id: 'temperate, arid, windy',
value: 95000000
}
{
_id: 'tropical',
value: 45000000
}
{
_id: 'temperate, artic',
value: 421000000
}


{
_id: 'temperate, tropical',
value: 1000
}
{
_id: 'superheated',
value: 18500000000
}
{
_id: 'artificial temperate ',
value: 1000000
}
{
_id: 'hot, humid',
value: 8500000
}
{
_id: 'arid, temperate, tropical',
value: 6000000000
}
{
_id: 'temperate, moist',
value: 10000000
}
{
_id: 'tropical, temperate',
value: 500000000
}
{
_id: 'frigid',
value: 519000000
}

Consideraciones Adicionales

Debido a la decisión de embeber el documento de armas (weapon) dentro de personajes (character), los archivos adjuntos a esta entrega resultan en un total de 20, siendo los especificados a continuación:

Archivos anexos:

1. esquema_character.json
2. esquema_faction.json
3. esquema_historic-event.json
4. esquema_location.json
5. esquema_movie.json
6. esquema_planet.json
7. esquema_spaceship.json
8. esquema_specie.json
9. esquema_vehicle.json
10. documentos_character.json
11. documentos_faction.json
12. documentos_historic-event.json
13. documentos_location.json
14. documentos_movie.json
15. documentos_planet.json
16. documentos_spaceship.json
17. documentos_specie.json
18. documentos_vehicle.json
19. consultas.txt
20. rafaelContrerasPimentel_rafaelContrerasAgudelo_yarimaContrerasBlanco_Informe.pdf

    Nota: Aunque el formato indica "<nombreApellido1>_<nombreApellido2>_<nombreApellido3>_Informe.pdf", debido a la repetición del mismo apellido para todos los integrantes del equipo y de nombre y apellido para dos de los mismos, se procedió a especificar los segundos apellidos para facilitar la identificación.