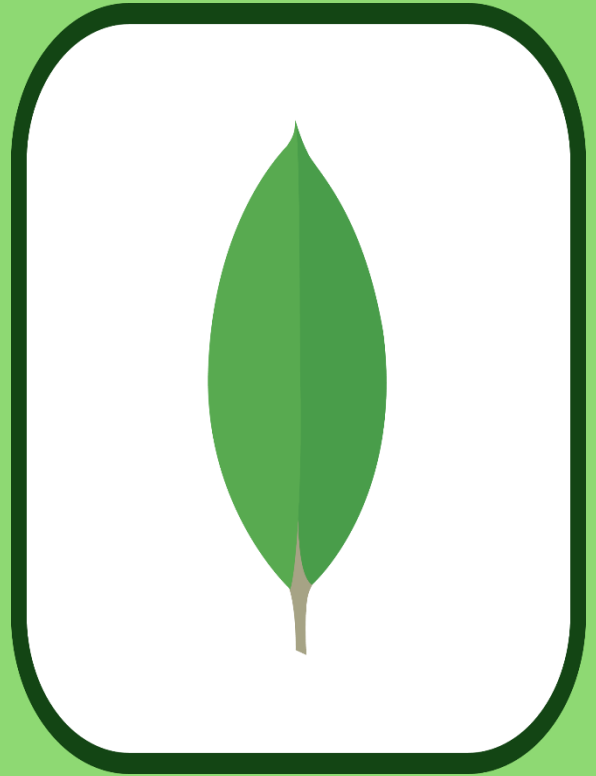


Consultas en MongoDB



Adrián García-Herras Herrero
Víctor Marín Escribano
Ariel Orlando Cahuaza Huanca

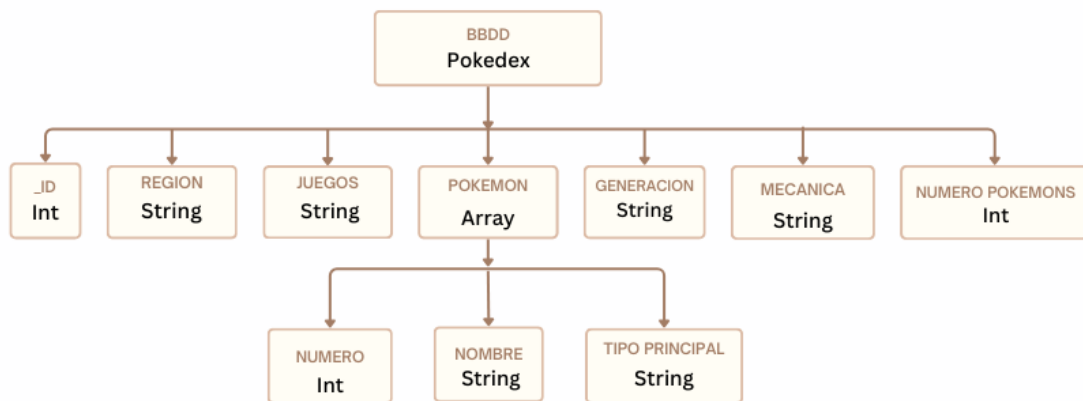
Índice

Índice	1
Introducción.....	2
EJER 1	3
EJER 2	3
EJER3.....	4
EJER4.....	4
EJER5.....	5
EJER6.....	5
EJER7.....	6
EJER8.....	6

Introducción

La base de datos de Pokedex tiene 10 documentos de las regiones de los juegos de pokemon, estas están compuestas por el `_id`, Región, Juegos, Generación, Mecánica, Numero_pokemons, Pokémon.

Por orden estas representan, el identificador en la base de datos, el nombre de la región, los juegos en los cuales la historia transcurre en dicha región, generación en la cual aparece, la mecánica o mecánicas que aporoto a la saga, el número de pokemons que añadió y sus pokemons iniciales, estos están compuestos por su número de la pokedex, el nombre del pokemon y sus tipos.



Esquema de datos

EJER 1

En este caso se quiere ordenar todos los pokemon por la región de manera alfabética. Usando find({}) para seleccionar todos para luego usar la proyección para excluir el id e incluir la región, juegos y generación. Y el sort para ordenarlos de manera alfabética.

Resultado:

	Generación	Juegos	Region
1	Séptima	Pokémon Sol/Luna	AloLa
2	Octava	Pokémon Espada/Escudo	Galar
3	Cuarta (precuela)	Pokémon Leyendas: Arceus	Hisui
4	Tercera	Pokémon Rubí/Zafiro/Esmeralda	Hoenn
5	Segunda	Pokémon Oro/Plata/Cristal	Johto
6	Sexta	Pokémon X/Y	Kalos
7	Primera	Pokémon Rojo/Verde/Azul/Amarillo	Kanto
8	Novena	Pokémon Escarlata/Púrpura	Paldea
9	Cuarta	Pokémon Diamante/Perla/Platino	Sinnoh
10	Quinta	Pokémon Negro/Blanco	Unova

Ilustración 1-Resultado de la consulta del primer ejercicio

EJER 2

En este caso buscamos la región con más de 150 pokemons. Y lo hacemos utilizando un find() para luego filtrarlo con \$gt seleccionado número de pokemons mayor a 150 y en la proyección excluimos el id.

Resultado:

	Generación	Juegos	Region
1	Primera	Pokémon Rojo/Verde/Azul/Amarillo	Kanto
2	Quinta	Pokémon Negro/Blanco	Unova

Ilustración 2-Resultado de la consulta del segundo ejercicio.

EJER3

En este caso encontrar las regiones con los pokemons de tipo Agua y que en pokedex sean superior a 500.

Para ello usamos find() con un \$eleMatch para buscarlas en los arrays que sean de tipo: Agua y el Numero con \$gt mayor a 500 y excluyendo en la proyección el id y la mecánica.

Resultado:

Generación	Juegos	Numero_pokemons	Pokémon	Region
Quinta	Pokémon Negro/Blanco	156	[{"Número": new NumberInt("495"), "Nombre": "Snivy", "Tipo_pr	Unova
Sexta	Pokémon X/Y	72	[{"Número": new NumberInt("650"), "Nombre": "Chespin", "Tipo_	Kalos
Séptima	Pokémon Sol/Luna	88	[{"Número": new NumberInt("722"), "Nombre": "Rowlet", "Tipo_p	Alola
Octava	Pokémon Espada/Escudo	82	[{"Número": new NumberInt("810"), "Nombre": "Grookey", "Tipo_	Galar
Novena	Pokémon Escarlata/Púrpura	120	[{"Número": new NumberInt("906"), "Nombre": "Sprigatito", "Ti	Paldea
Cuarta (precuela)	Pokémon Leyendas: Arceus	8	[{"Número": new NumberInt("722"), "Nombre": "Rowlet", "Tipo_p	Hisui

Ilustración 3-Resultado de la consulta del tercer ejercicio

EJER4

En este caso buscamos obtener la región con más juegos con más de 100 pokemons y donde no aparezcan Hoenn y que no sean de la 1 generación.

Usaremos un find con los operadores \$gt, \$not (\$eq) para seleccionar los que tienen más de 100 pokemon, la región hoenn y la primera generación. Y descartamos el id.

Resultado:

Juegos	Mecánica	Numero_pokemons	Region
Pokémon Diamante/Perla/Platino	Movimientos divididos en físico/especial	107	Sinnoh
Pokémon Negro/Blanco	Combates triples y rotatorios	156	Unova
Pokémon Escarlata/Púrpura	Teracristalización	120	Paldea

Ilustración 4-Resultado de la consulta del cuarto ejercicio.

EJER5

Filtra regiones con más de 100 pokémons, ordena por región y limita a 5 resultados

Usamos el aggregate con un match y gt para filtrar el número de pokemons > 100, el sort para ordenarlos por de manera alfabéticamente y el limit mostrar los 5 primeros.

Resultado:

_id	Generación	Juegos	Mecánica	Numero_pokemo...	Pokémon	Region
3	Tercera	Pokémon Rubi/Zafiro/Es...	Habilidades y concurso...	135	[{"Número": "252", "Nom	Hoenn
1	Primera	Pokémon Rojo/Verde/Azu...	El juego completo	151	[{"Número": "001", "Nom	Kanto
9	Novena	Pokémon Escarlata/Púrp...	Teracristalización	120	[{"Número": "906", "Nom	Paldea
4	Cuarta	Pokémon Diamante/Perla...	Movimientos divididos ...	107	[{"Número": "387", "Nom	Sinnoh
5	Quinta	Pokémon Negro/Blanco	Combates tripl...	156	[{"Número": "495", "Nom	Unova

Ilustración 5-Resultado de la consulta del quinto ejercicio.

EJER6

Divide regiones en pares/impares (según _id) y calcula media de pokémons por grupo y un total de regiones por grupo

Resultado:

grupo	mediaPokemons	regiones	totalRegiones
Regiones pares	73.8	["Johto", "Sinnoh", "Kalos", "Galar", "Hisui"]	5
Regiones impares	130	["Kanto", "Hoenn", "Unova", "Alola", "Paldea"]	5

Ilustración 6-Resultado de la consulta del sexto ejercicio.

EJER7

Descompone array Pokémon y filtra solo tipos de Fuego

En este caso usaremos un aggregate con un unwind para generar un archivo individual de cada pokemon, el match para filtrarlo el tipo fuego. Y con project para mostrar la región, nombre y el tipo.

Resultado:

_id	Pokémon	Region
1	{"Nombre": "Charmander", "Tipo_principal": "Fuego"}	Kanto
2	{"Nombre": "Cyndaquil", "Tipo_principal": "Fuego"}	Johto
3	{"Nombre": "Torchic", "Tipo_principal": "Fuego"}	Hoenn
4	{"Nombre": "Chimchar", "Tipo_principal": "Fuego"}	Sinnoh
5	{"Nombre": "Tepig", "Tipo_principal": "Fuego"}	Unova
6	{"Nombre": "Fennekin", "Tipo_principal": "Fuego"}	Kalos
7	{"Nombre": "Litten", "Tipo_principal": "Fuego"}	Alola
8	{"Nombre": "Scorbunny", "Tipo_principal": "Fuego"}	Galar
9	{"Nombre": "Fuecoco", "Tipo_principal": "Fuego"}	Paldea
10	{"Nombre": "Cyndaquil", "Tipo_principal": "Fuego"}	Hisui

Ilustración 7-Resultado de la consulta del séptimo ejercicio.

EJER8

Calcula densidad de pokémons (nº Pokémon/100) y crea descripción concatenada indicando la región y la generación

En este caso con project creamos dos campos calculados, el primero par calcular la densidad de los pokemon con \$divide y la segunda para una descripción con \$concat de la región y generación.

Resultado:

_id	Densidad	Descripción
1	1.51	La región Kanto pertenece a la Primera generación
2	1	La región Johto pertenece a la Segunda generación
3	1.35	La región Hoenn pertenece a la Tercera generación
4	1.07	La región Sinnoh pertenece a la Cuarta generación
5	1.56	La región Unova pertenece a la Quinta generación
6	0.72	La región Kalos pertenece a la Sexta generación
7	0.88	La región Alola pertenece a la Séptima generación
8	0.82	La región Galar pertenece a la Octava generación
9	1.2	La región Paldea pertenece a la Novena generación
10	0.08	La región Hisui pertenece a la Cuarta (precuela) generación

Ilustración 8-Resultado de la consulta del octavo ejercicio.