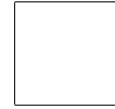


Apellidos:

Nombre: UVUS:



Bases de datos (GE/DGME)

12/01/2022

Primera prueba de evaluación alternativa

Ejercicio 1 (1 punto) Responda razonadamente a las siguientes cuestiones relacionadas con el ámbito de NoSQL:

1. (0,375 puntos) Hemos dedicado la mayor parte del cuatrimestre a trabajar con bases de datos relacionales, y hemos visto que tienen muchas bondades, pero... ¿es siempre el más adecuado el enfoque que proporcionan estas? ¿Qué aspectos o necesidades pueden motivar la adopción de otros enfoques, mediante bases de datos no relacionales?
2. (0,375 puntos) ¿Qué dos principios fundamentales hemos visto que priman en las bases de datos relacionales, y por qué otros dos principios se sustituyen en el ámbito de NoSQL? Puede emplear en su respuesta ACID, BASE o CAP, pero no es necesario, es suficiente con que describa un poco los principios fundamentales en cada uno de estos ámbitos, y sus diferencias fundamentales.
3. (0,25 puntos) ¿Qué tipo de base de datos NoSQL es MongoDB? ¿Por qué se caracteriza este tipo? ¿Qué otros tipos de bases de datos no relacionales conoce?

Ejercicio 2 (2 puntos) Haciendo uso de los principios de diseño analizados en clase, basados en el curso que estuvimos viendo, se pide plantear la estructura de colecciones que montaría para el problema planteado en el siguiente escenario (extraído y recortado del boletín sobre diseño conceptual), consistente en un Sistema de Información Geográfica:

- a) *De los ríos se almacenará código de río, nombre, descripción y longitud total. Además, se almacenarán las provincias por las que pasa el río, con la longitud del tramo del río para cada una.*
- b) *De las provincias se almacenará código de provincia, nombre y nº de habitantes, así como la capital de la provincia y la Comunidad Autónoma. Además, se almacenará para la provincia información sobre sus municipios, el censo de cada uno, etc.*
- c) *Los ríos pueden ser afluentes de otros ríos. Para cada afluente se desea conocer de qué río lo es, en qué provincia se une a dicho río y en qué punto kilométrico.*
- d) *De los sistemas montañosos se almacenará un código, nombre, orientación (norte, nordeste, etc.) y la longitud, así como la altura máxima y las provincias que ocupa.*
- e) *Los sistemas están formados por montes, de los que almacenar un código, nombre, descripción y altura. Un monte sólo pertenecerá a un sistema. De los montes también se quiere almacenar la provincia o provincias en las que se encuentra. Los montes pueden tener origen volcánico o de plegamiento. En el caso de que su origen sea volcánico, se desea almacenar el tipo de volcán y si es de plegamiento, se almacenará el periodo geológico de dicho plegamiento.*
- f) *Algunos ríos y montes son elementos geológicos monitorizados por satélite. De dichos elementos se desea almacenar la fecha en la que se comienza su monitorización y el satélite que realiza el seguimiento. Un elemento monitorizado sólo puede serlo por un satélite y un satélite podrá monitorizar varios elementos. De los satélites se desea almacenar su número identificativo, nombre y descripción.*

Nota: los siguientes ejercicios emplean una BD MongoDB sobre Pokemon, y otra sobre Lego. Puede generarlas mediante los archivos `pokemon.json` y `lego.json`. Tenga en cuenta la estructura de los archivos a la hora de importar con ciertos parámetros.

Ejercicio 3 (2,5 puntos) Realice las siguientes operaciones de consulta sobre la base de datos MongoDB anterior, empleando las funciones básicas de consulta (sin agregaciones):

- Devolver todos los datos de los pokemon cuyo `pkdx_id` sea menor o igual a 50.
- Devolver los tipos de pokemon que existen, en orden alfabético.
- Devolver el número de pokemons tipo `psychic`.
- Devolver los tres primeros pokemons que tengan una evolución de nivel mayor que 30.
- Devolver el `pkdx_id` y nombre de los pokemons que tengan tipo “fire” o “water”, y ordenarlos alfabéticamente por nombre. Resuélvalo de dos formas distintas, empleando distintos operadores dentro de su función básica de consulta.
- Devolver `name` y `evolutions` de los pokemons que tengan al menos una evolución antes del nivel 15, en orden descendente según dicho nivel.

Ejercicio 4 (1,5 puntos) Realice las siguientes operaciones de manipulación de los datos sobre la base de datos MongoDB anterior, empleando las funciones vistas en clase para la actualización, inserción y eliminación:

- Actualizar un pokemon que tenga el tipo “flying” para cambiarle el campo `image_url` al valor “voladora.jpg”.
- Actualizar los pokemon cuya `description` contenga la palabra “telekinetic”, incluyéndoles un nuevo campo “isTelekinetic” con el valor **true**.
- Añadir el tipo dragon a Charizard.
- Incorporar un nuevo Pokemon con algunos datos básicos similares a los que observe en los datos de la base de datos.
- Eliminar el pokemon con `pkdx_id` igual a 10.
- Cambiar por el 30 el nivel en que Wartortle evoluciona a blastoise.

Ejercicio 5 (3 puntos) Realice las siguientes operaciones de consulta sobre la base de datos MongoDB anterior, empleando ahora agregaciones:

- Devolver el número de sets de Lego disponibles para cada tema (`theme_id`). Deben aparecer primero los temas con mayor número de sets incluidos.
- Devolver el total de número de piezas (`num_parts`), así como su promedio de piezas, para cada tema (`theme_id`). Limite su resultado a los 7 temas con un mayor número de piezas totales.
- Devolver, para cada año posterior al 2000, el nombre y número de piezas del set con mayor número de piezas. Deben aparecer primero los datos más antiguos.
- Devolver, de aquellos sets cuyo año se encuentre entre 2015 y 2021 (ambos inclusive), el set con mayor número de piezas por cada tema. La consulta deberá devolver, además del nombre del set, el porcentaje que supone su número de piezas con respecto al total de piezas de su tema (siempre ciñéndonos al rango de años anterior).
- Devolver el número de pokemons de cada tipo. Ayuda: puesto que cada pokemon puede tener varios tipos, para que el resultado sea lo más correcto posible, puede usar para esta consulta el operador *unwind*.