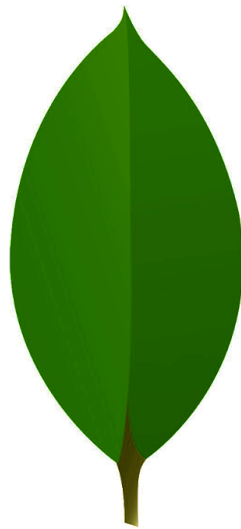


1º ASIR DAM Subgrupo B

Proyecto Bases de Datos 2ª Evaluación

Miguel Ferreira Cordero



mongoDB®

Introducción

En este proyecto se ha creado una base de datos sobre coches, en concreto de sus características más comunes y útiles. En otras colecciones a parte de la de coches, se almacenará también las categorías de coches en la base de datos, las marcas de estos y las competiciones que han ganado las marcas.

El proyecto consta de 3 consultas de agregación que muestra los datos que más suelen pedir los clientes a la hora de comprar un vehículo e información sobre las marcas.

Operadores empleados en el aggregate

Aggregate: Agrupa datos de varios documentos y opera de una manera en la que busca en esos documentos y devuelve un resultado único.

\$count: Devuelve un recuento del número de documentos en esta etapa de la canalización de agregación.

\$group: Agrupa los documentos de entrada por una expresión con un id y aplica las expresiones del field, si se especifica, a cada grupo. Lee todos los documentos de entrada y genera uno distinto por cada group. Los documentos de salida sólo tienen id, y si se especifica, un campo.

\$lookup: hace un outer join de la misma base de datos. Este incluye un from, que es la colección a la que se va a unir, un localField, que es , un ForeignField, que es, y un as, que es cómo se llamará el array de salida.

\$project: Pasa a la siguiente etapa los campos especificados. Estos campos pueden ser ya existentes de los documentos de entrada o campos creados en el mismo project.

\$match: Filtra los documentos para que sólo pasen a la siguiente etapa aquellos documentos que contengan unas condiciones específicas.

\$replaceWith: Reemplaza el documento de entrada con un documento especificado.

\$arrayToObject: Convierte un array en un único documento. El array debe ser de dos maneras: o ser un array de 2 elementos donde el primero es el nombre del campo y el segundo el valor de ese campo (ej: [["item", "abc123"], ["qty", 25]]), o debe de ser un array de documentos que tenga 2 campos, "k" y "v", donde K tiene el nombre del

campo y V el valor del mismo. Véase en el ejemplo: [{ "k": "item", "v": "abc123"}, { "k": "qty", "v": 25 }].

\$concatArrays: Concatena arrays y devuelve un array concatenado. (Ej: { \$concatArrays: [<array1>, <array2>, ...] })

\$sort: Ordena todos los documentos de entrada y los devuelve en un orden específico.

\$push: Adjunta un valor específico a un array. (Ej: { \$push: { <field1>: <value1>, ... } })

\$unwind: Deconstruye un array de los documentos de entrada para devolver un documento por cada elemento. Cada documento de salida es el documento de entrada con el valor del campo reemplazado por el elemento. (Ej: { \$unwind: <field path> })

\$merge: Escribe los resultados de una tubería de aggregations a una colección específica. Este operador debe estar en la última etapa de la tubería.

\$set: Reemplaza el valor de un campo con un valor específico. (Ej: { \$set: { <field1>: <value1>, ... } })

\$addFields: Añade un nuevo campo a los documentos. (Ej: { \$addFields: { <newField>: <expression>, ... } })

\$expr: Permite el uso de las aggregation expressions.

\$switch: Evalúa una serie de casos, y cuando encuentre una expresión que sea true devuelve un valor definido. Si no encuentra ningún true devolverá un valor por defecto.

Toda esta documentación se pueden encontrar en la página de Mongo DB: [Mongo DB Documentation](#)