

Diseño de agregados

Bases de datos NoSQL

UOC

Universitat Oberta
de Catalunya

Una de las mayores diferencias entre las bases de datos basadas en el modelo relacional y las bases de datos basadas en el modelo de agregados es la forma en cómo se almacena la información. El diseño de agregados requiere un cambio de perspectiva en el que las consultas son las que dirigen el diseño y donde la duplicación de datos está permitida.

Este documento pretende dar soporte al diseño de agregados mediante una breve descripción de los aspectos a considerar y algunos ejercicios resueltos en detalle.

Autores

Antonio Sarasa Cabezuelo

Joan Antón Pérez Braña

Jordi Conesa i Caralt (revisor)

Índice

Pensando en agregados	3
¿Qué considerar al diseñar un agregado?	5
Ejercicios	6
Ejercicio 1	6
Ejercicio 2	7
Soluciones	10
Ejercicio 1	10
Ejercicio 2	14

Histórico de revisiones	
Versión 1.0	Marzo 2020

Pensando en agregados

Una de las mayores diferencias entre las bases de datos basadas en el modelo relacional y las bases de datos basadas en el modelo de agregados es la forma en cómo se almacena la información.

El diseño de agregados requiere un cambio de perspectiva para los diseñadores, desarrolladores, usuarios y administradores de bases de datos pues se pasa del modelo de datos relacional, que almacena los datos en estructuras tabulares fijas, bidimensionales, formadas por filas y columnas, a un modelo de datos de agregado cuyo esquema es variable y que contiene atributos, matrices y sub-agregados embebidos.

Tal como se explica en el libro base de la asignatura, una base de datos relacional suele venir bien cuando no se conoce a priori como van a ser las consultas que se van a realizar, de manera que sólo se piensa en cómo se encuentran relacionadas las entidades de las cuales se guarda información. El problema de cómo se realizarán las consultas es algo secundario que se solucionara con consultas SQL más o menos complejas. Cuando se diseña el esquema conceptual de la base de datos, las consultas que se deberán realizar se consideran en segundo término, principalmente se piensa en términos de entidades, tipos de datos de las entidades y cómo se encuentran relacionadas. Por eso, el diseño lógico de las bases de datos relacionales es un modelo neutral con respecto al tipo de consulta que se va a realizar.

Sin embargo, en las bases de datos basadas en el modelo de agregación, las consultas son las que dirigen el diseño de los agregados. Para quien provenga de un contexto relacional debe tener en cuenta la ausencia de operaciones de combinación (o JOIN) en bases de datos no relacionales. Ello implica que el agregado debe contener toda la información necesaria para poder resolver adecuadamente las consultas planteadas. Así pues, se crean agregados específicos para resolver consultas específicas (o un conjunto de ellas) de la manera eficiente. Evidentemente, un mismo agregado difícilmente va a permitir realizar una consulta eficiente en todos los casos.

Para ilustrar este caso, utilizaremos el ejemplo de “el carro de la compra” (ver documento *B2_T3_1_ModelosAgregacionMotivacion.pdf*). En este caso se almacenaba un agregado con información de cada cliente y de los productos que había comprado. Supongamos que se quiere conocer qué clientes han comprado un determinado producto X. En este caso, el agregado propuesto, que contiene información por cliente, no sería una buena opción. El motivo es que habría que recorrer todos los clientes (todos los agregados) para verificar si el cliente en cuestión ha comprado X o no.

Sería más eficiente organizar los agregados por productos, donde hubiera un agregado por cada producto y en el mismo se incluyera información de los clientes que lo han comprado, cuando y en qué cantidad. De esta forma la consulta sobre quién ha comprado el producto X, se haría de forma óptima (en tiempo constante) consultando el campo, o subagregado, que almacena la lista de los clientes que compraron el producto.

Este ejemplo refleja perfectamente que, dependiendo la estructura del agregado, la consulta será más (o menos) eficiente. En el ejemplo, con la primera estructura de agregado podríamos haber recuperado la información que nos pedían, pero a costa de hacer un recorrido exhaustivo por todos los clientes. Por tanto, aunque con un único agregado se pueden realizar distintas consultas, es conveniente considerar la posibilidad de crear nuevos agregados cuando no se garantice una buena eficiencia de estas.

En las bases de datos NoSQL de agregación la redundancia de datos debida a la existencia de diferentes agregados con la misma información, pero con diferente estructura, no se considera un problema, sino una práctica habitual. Es parte de las características de este tipo de bases de datos.

Por otro lado, también hay que tener en cuenta que a las bases de datos NoSQL no se les exige que tengan un esquema prefijado, a diferencia de las bases de datos relacionales. Debido a que van a manipular diferentes tipos de agregados en función del tipo de consulta, y la falta de un esquema, se introduce un cambio en las responsabilidades del programador, que deberá conocer la estructura del agregado y realizar algunas de las tareas para el mantenimiento de la consistencia, asegurando que se usan los tipos de datos adecuados, que la información es coherente, etc.

¿Qué considerar al diseñar un agregado?

La norma de oro es *no modelar en torno a las relaciones, ni modelar alrededor de los objetos, sino modelar basándonos en las consultas a realizar*. Para ello se propone seguir las siguientes reglas al modelar un agregado:

Paso 1: Determinar exactamente qué consultas necesita admitir el agregado que vamos a diseñar. Esto puede incluir muchas consideraciones que quizás no se piensen en un inicio. Es necesario analizar los requisitos de la consulta en distintos aspectos como:

- Agrupación por un atributo,
- Ordenación por un atributo,
- Filtrado basado en algún conjunto de condiciones,
- Hacer cumplir la unicidad en el conjunto de resultados, y
- Todas las posibles consideraciones referentes a los requisitos de la consulta.

Normalmente, los cambios en alguno de estos requisitos de consulta, puede requerir un cambio en el diseño del agregado para proveer una máxima eficiencia.

Paso 2: Usar un tipo de agregado por patrón de consulta. Si se necesita dar soporte a múltiples patrones de consulta, generalmente se necesitará más de un tipo de agregado. Dicho de otra manera, cada tipo de agregado debe pre-compile la "respuesta" a una consulta de alto nivel que éste deba admitir. Así es como se optimizan las lecturas. Es importante recordar que la duplicación de datos está permitida y incluso es aconsejable en algunos casos.

A continuación, veremos distintos ejercicios con el objetivo de poner en práctica el diseño de agregados.

Ejercicios

A continuación proponemos algunos ejercicios para practicar el diseño de agregados. Las soluciones detalladas a los mismos se encuentran al final de este documento.

Ejercicio 1

La multinacional de venta de productos deportivos DLON desea explotar la información sobre los productos que vende en todas las tiendas que tiene distribuidas en las distintas ciudades de España. De cada producto se dispone de la siguiente información: nombre del producto, código del producto, precio de venta, sección de la tienda a la que pertenece el producto, y proveedor. Así mismo, se dispone de la información del número de unidades vendidas de cada producto, cuando se vendieron, en qué tiendas se vendieron (se conoce el nombre y código de las tiendas) los datos del comprador y del vendedor y de cuántos productos fueron devueltos.

Las consultas que se quieren realizar sobre la información son las siguientes:

1. Sobre cada cliente se dispone de DNI y nombre y se desea conocer a nivel mensual y por año, el número total de productos comprados mensualmente, el importe gastado mensualmente e información particularizada sobre cada producto comprado: código producto, fecha de compra, código tienda, unidades compradas, valor de la compra y código vendedor.
2. Sobre cada tienda se dispone de su nombre y su código, y se desea conocer a nivel mensual y por año, el número total de productos vendidos, el importe facturado por esos productos e información particularizada sobre cada producto vendido: código del producto, número de unidades vendidas, número de unidades que fueron devueltas y código de los clientes que compraron el producto.
3. Sobre cada vendedor se dispone de nombre y su código, y se desea conocer a nivel mensual y por año, el número total de productos vendidos, el importe total de las ventas realizadas e información particularizada sobre cada venta realizada: códigos productos vendidos, fecha de la venta, DNI del cliente, e importe de la venta. Observar que una venta consta de un conjunto de productos vendidos.

Analizadas el tipo de consultas que se desean llevar a cabo, se ha optado por usar un modelo de datos orientado a agregados. Se pide definir los agregados necesarios que satisfagan y optimicen las consultas planteadas. Concretamente por cada agregado se debe indicar:

- Breve explicación de la estructura del agregado.
- Representación gráfica del agregado propuesto usando un modelo gráfico de cajas anidadas como el que aparece en la Figura 1.1 del Capítulo 1 del libro básico de la asignatura, así como en los videos sobre agregación disponibles en el campus virtual.

Ejercicio 2

Considera las siguientes tablas de un sistema relacional para gestionar los libros de una biblioteca:

AUTORES					
Columna	Tipo	Es Nulo	Valor por defecto	Clave primaria	Es único
Autor_ID	INT	No		Sí	Sí
Nombre	VARCHAR(10)	No			
Apellidos	VARCHAR(35)	No			
Fecha nacimiento	DATE	No			
Fecha defunción	DATE	Sí	NULO		

LIBROS					
Columna	Tipo	Es Nulo	Valor por defecto	Clave primaria	Es único
ISBN	INT	No		Sí	Sí
Título	VARCHAR(35)	No			
AUTOR_ID	INT	No			
Editorial	VARCHAR(20)	No			
Soporte	VARCHAR(35)	No			
Fecha_registro	DATE	Sí	NULL		
País	VARCHAR(20)	Sí	NULL		

USUARIOS					
Columna	Tipo	Es Nulo	Valor por defecto	Clave primaria	Es único
Nombre	VARCHAR(10)	No			
Apellidos	VARCHAR(35)	No			
DNI	VARCHAR(15)	No		Sí	Sí
Email	VARCHAR(50)	No			Sí
Teléfono	VARCHAR(15)	Sí	NULL		

PRESTAMOS					
Columna	Tipo	Es Nulo	Valor por defecto	Clave primaria	Es único
Registro	INT	No		Sí	Sí
Id_Libro	INT	No			
Id_Usuario	VARCHAR(15)	No			
Fecha_prestamo	DATE	No			
Fecha_devolución	DATE	Sí			

Tras realizar un estudio sobre las consultas que se desean realizar, se ha llegado a la conclusión de que se quieren sólo tres tipos de consultas:

1. De cada usuario se quiere conocer el nombre y DNI, y un listado de actividad de usuario por meses. En cada mes se desea conocer los datos de cada libro prestado durante ese mes: ISBN, Título, Autor_ID, Editorial y Fecha devolución.
2. De cada editorial se quiere conocer nombre de la editorial, y un listado por meses de préstamo. En cada mes se desea conocer los datos de cada libro perteneciente a la editorial que ha sido prestado durante ese mes: ISBN, Título, Autor_ID, DNI del usuario al que se le prestó y fecha de préstamo.
3. De cada autor (identificador, nombre y apellido) se quiere conocer un listado préstamos agrupados por meses. En cada mes se desea conocer de cada libro del autor: el título del libro, el número de préstamos del libro, la editorial a la que pertenece cada libro, y un listado de los DNI de las personas que solicitaron el préstamo.

Analizadas el tipo de consultas que se desean llevar a cabo, se ha optado por usar un modelo de datos orientado a agregados. Se pide definir los agregados necesarios que satisfagan y optimicen las consultas planteadas. Concretamente por cada agregado se debe indicar:

- Breve explicación de la estructura del agregado.
- Representación gráfica del agregado propuesto usando un modelo gráfico de cajas anidadas como el que aparece en la Figura 1.1 del Capítulo 1 del libro básico de la asignatura, así como en los videos sobre agregación disponibles en el campus virtual

Soluciones

A continuación se encuentran los soluciones a los ejercicios propuestos.

Ejercicio 1

Consulta 1

Se propone un agregado que represente la información de un cliente concreto. Así, los campos más externos del agregado serán información identificativa del cliente: nombre y DNI. A continuación, existirá un campo denominado meses que contendrá una colección de agregados por cada mes-año. Cada uno de estos agregados contendrá como campos: mes, año, número total de productos comprados, importe gastado, y un campo productos. Este último campo almacenará un agregado por cada producto comprado por el cliente con los campos: código producto, fecha de compra, código de la tienda, unidades compradas, valor de la compra y código del vendedor.

En el siguiente diagrama se muestra el diseño planteado:

Nombre:
DNI:

Meses:

Mes:
Año:
Número total productos comprados:
Importe gastado:

Productos:

Código producto:
Fecha compra:
Código tienda:
Unidades compradas:
Valor de la compra:
Código vendedor:

Código producto:
Fecha compra:
Código tienda:
Unidades compradas:
Valor de la compra:
Código vendedor:

Mes:
Año:
Número total productos comprados:
Importe gastado:

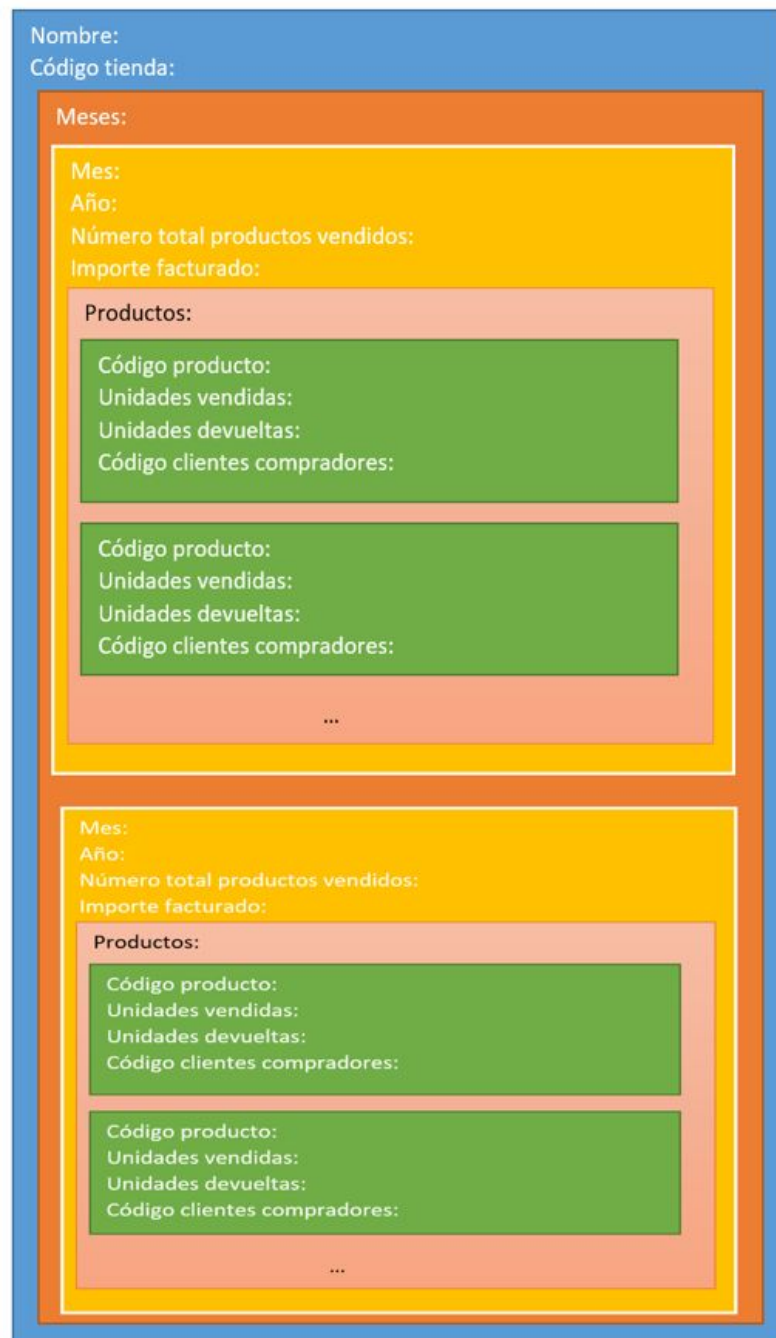
Productos:

Código producto:
Fecha compra:
Código tienda:
Unidades compradas:
Valor de la compra:
Código vendedor:

Código producto:
Fecha compra:
Código tienda:
Unidades compradas:
Valor de la compra:
Código vendedor:

Consulta 2

Se propone un agregado que represente la información de una tienda concreta. Así, los campos más externos del agregado serán información identificativa de la tienda: nombre y código tienda. A continuación, existirá un campo denominado meses que contendrá una colección de agregados por cada mes-año. Cada uno de estos agregados contendrá como campos: mes, año, número total de productos vendidos, importe facturado, y un campo productos. Este último campo almacenará un agregado por cada producto vendido en la tienda con los campos: código producto, unidades vendidas, unidades devueltas y código de los clientes que compraron el producto. En el siguiente diagrama se muestra el diseño planteado:



Consulta 3

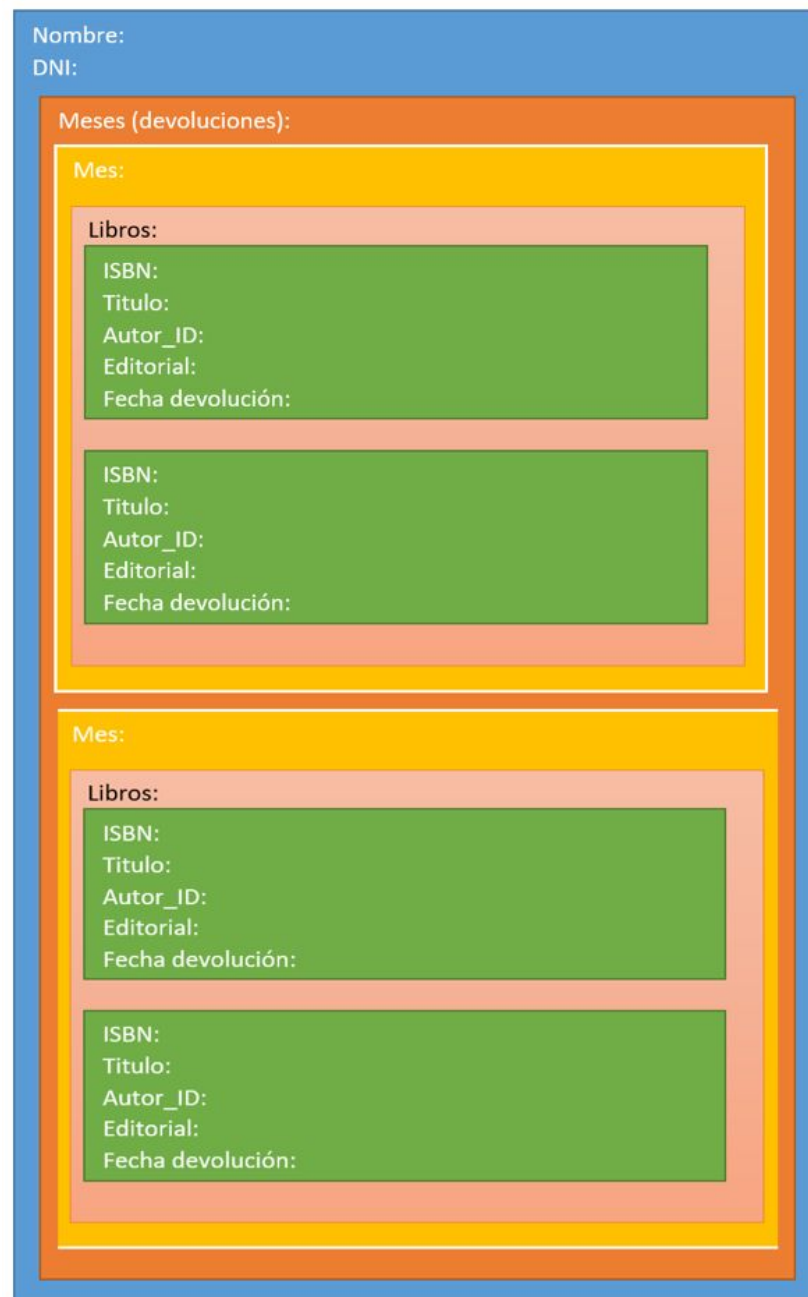
Se propone un agregado que represente la información de un vendedor concreto. Así, los campos más externos del agregado serán información identificativa del cliente: nombre y código vendedor. A continuación, existirá un campo denominado meses que contendrá una colección de agregados por cada mes-año. Cada uno de estos agregados contendrá como campos: mes, año, número total de productos vendidos, importe total de ventas realizadas, y un campo ventas. Este último campo almacenará un agregado por cada venta realizada por el vendedor con los campos: código productos vendidos, fecha de la venta, DNI del cliente, y el importe de la venta realizada. En el siguiente diagrama se muestra el diseño planteado:



Ejercicio 2

Consulta 1

Se propone un agregado que represente la información de un usuario concreto. Así, los campos más externos del agregado serán el nombre del usuario y su dni. A continuación existirá un campo denominado meses que contendrá una colección de agregados por cada mes. Cada uno de estos agregados contendrá como campos: mes y un campo libros. Este último campo almacenará un agregado por cada libro prestado al usuario con los campos: ISBN, Título, Autor_ID, Editorial y Fecha de devolución. En el siguiente diagrama se muestra el diseño planteado:



```
// si optamos por su representación en formato JSON: usuarios
{
  "nombre": "nombre01",
  "dni": "dni01",
  "meses": [
    {
      "mes": "mes01",
      "libros": [
        {
          "isbn": "isbn01",
          "titulo": "titulo01",
          "escritor": "escritor01",
          "editorial": "editorial01",
          "fecha_devolucion": "fecha01"
        },
        {
          "isbn": "isbn02",
          "titulo": "titulo02",
          "escritor": "escritor02",
          "editorial": "editorial02",
          "fecha_devolucion": "fecha02"
        }
      ]
    },
    {
      "mes": "mes02",
      "libros": [{...},{...}]
    }
  ]
}
```

Consulta 2

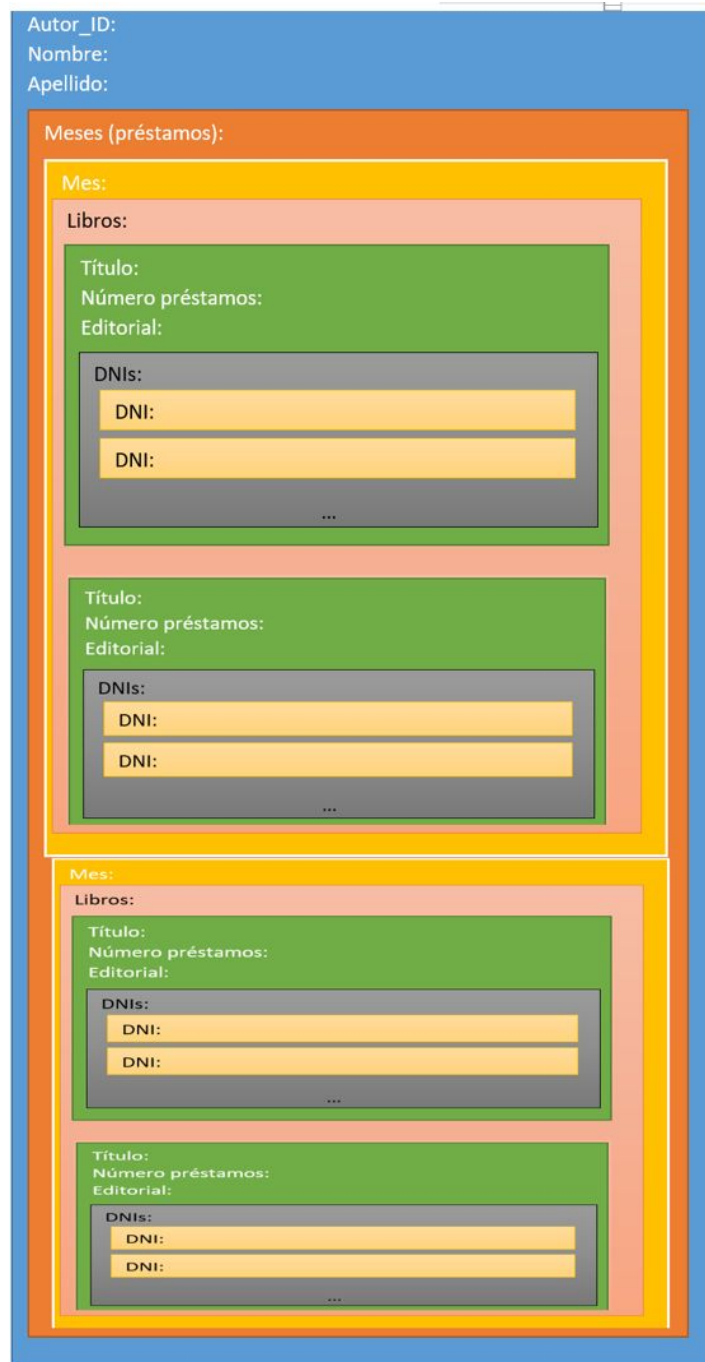
Se propone un agregado que represente la información de una editorial concreta. Así, el campo más externo del agregado será el nombre de la editorial. A continuación existirá un campo denominado meses que contendrá una colección de agregados por cada mes. Cada uno de estos agregados contendrá como campos: mes y un campo libros. Este último campo almacenará un agregado por cada ejemplar de un libro perteneciente a esa editorial con los campos: ISBN, Título, Autor_ID, DNI y Fecha de préstamo. En el siguiente diagrama se muestra el diseño planteado:




```
// si optamos por su representación en formato JSON:
editorial
{
  "editorial": "editorial01",
  "meses": [
    {
      "mes": "mes01",
      "libros": [
        {
          "isbn": "isbn01",
          "titulo": "titulo01",
          "escritor": "escritor01",
          "dni": "dni01",
          "fecha_prestamo": "fecha01"
        },
        {
          "isbn": "isbn02",
          "titulo": "titulo02",
          "escritor": "escritor02",
          "dni": "dni02",
          "fecha_prestamo": "fecha02"
        }
      ]
    },
    {
      "mes": "mes02",
      "libros": [{...},{...}]
    }
  ]
}
```

Consulta 3

Se propone un agregado que represente la información de un escritor concreto. Así, los campos más externos del agregado serán información identificativa del escritor: escritor e identificador. A continuación, existirá un campo denominado meses que contendrá una colección de agregados por cada mes. Cada uno de estos agregados contendrá como campos: mes, y un campo libros. Este último campo almacenará un agregado por cada libro prestado del escritor con los campos: Título, Número de préstamos del libro, la editorial a la que pertenece el libro, y un listado de los DNIs de las personas que solicitaron el préstamo del libro. En el siguiente diagrama se muestra el diseño planteado:



```
// si optáramos por su representación en formato JSON:
autores
{
  "autor_ID": 88,
  "Nombre": "Jack",
  "Apellido": "Kerouac",
  "meses": [
    {
      "mes": "mes01",
      "libros": [
        {
          "titulo": "titulo01",
          "numero_prestamos": 2,
          "editorial": "editorial01",
          "dnis": [
            {
              "dni": "dni01"
            },
            {
              "dni": "dni02"
            }
          ]
        }
      ]
    },
    {
      "titulo": "titulo02",
      "numero_prestamos": 1,
      "editorial": "editorial02",
      "dnis": [{"dni": "dni03"}, {"dni": "dni04"}]
    }
  ]
},
{
  "mes": "mes02",
  "libros": [{...},{...}]
}
]
```