Politecnico di Torino

DBMG

Design pattern exercises (1)

DANIELE APILETTI

POLITECNICO DI TORINO

Exercise 1 – «Books and publishers»

- We want to design a MongoDB database to store the following data about books and publishers.
- Each publisher is characterized by the name, the year of foundation and the country. A list of contact methods is available. The possibile contact methods are: email, telephone, website. At most one contact for each method can be recorded. Additional methods might be added in the future.
- Each book is identified by the ISBN code and characterized by the title, the subtitle, the number of pages, the language and the publication date. Each book is published by only one publisher. A publisher distributes several books.
- The design should be optimized to display for each book all its information and the publisher's name
  - Indicate for each collection in the database the document structure and the strategies used for modelling.

Exercise 1 – «Books and publishers»

publisher

```
{_id: <ObjectId>,
 name:<string>,
 year:<date>,
 country: <string>,
 contacts: {email: <string>,
            tel: <string>,
            website: <string> }
}
```

book

```
{_id: <string>, // ISBN
 title: <str>,
 subtitle: <string>,
 n: <int>,
 language: <string>,
 pub_date: <date>
 publisher: {_id: <ObjectId>,
             name:<string>}
}
```

Polymorphic pattern to track only the contact methods that are available

Extended reference pattern to display the name of the publisher. The application can access the book collection to show both the book information and the name of the publisher. In this way we avoid a lookup to the publisher collection.

Exercise 2 – «Blog website»

- We want to design a MongoDB database for the management of a blog.
- Each user subscribed to the blog is identified by a username. We want to keep track of user contact methods, such as email, phone, Twitter account, etc. The contact methods tracked by the platform may change over the time. A user might have more contacts of the same method. Each contact method can have a status, e.g., "active", "inactive", "confirmed", etc.
- Blog posts are characterized by the title, the description and the author (i.e., username). The posts receive comments from users. Each comment is characterized by the text, the user who published it and its timestamp.
- The application should display for each post only the top 10 most recent comments. The user profile should also include the total number of posts and comments written by the user.
  - Indicate for each collection in the database the document structure and the strategies used for modelling.

Exercise 2 – «Blog website»

```
user
{_id: <string>, // username
 contacts:[
          {k: <string>, // contact method
           v: <string>, // contact value
           s: <string>, // status
          ],
 nPosts: <int>,
 nComments: <int>
}
post
{_id: <ObjectId>,
 title: <str>,
 description: <string>,
 author: <string>, // username
 comments: [       // most recent comments
          {_id: <ObjectId>
           user: <string>,
           text: <string>,
           timestamp: <datetime>}
          ]
}
comment
{_id: <ObjectId>
 user: <string>,
 text: <string>,
 timestamp: <datetime>
}
```

Attribute pattern to track the contact methods. The embedded documents in the contact fields consists of the k field which represents the contact method, the v field providing the corresponding value, and the s field for the status

Computed pattern to store the number of posts and comments published by the user.

Subset pattern to track only the most recent comments.

Exercise 3 – «Breadcrumbs»

- We want to design the database for the management of a sitemap.
- Each page of the webisite is characterized by the relative path, the title, the body, the creation timestamp, the last update timestamp and a list of tags.
- The pages are organized in a tree strcture. The home page is the root. All other pages are in sub paths from the root.
- The application should build the breadcrumbs code for each page. The breadcrumbs are an ordered list of links containing all the ancestors of the current page back to the home page, i.e., the root.
  - Indicate for each collection in the database the document structure and the strategies used for modelling.

Exercise 3 – «Breadcrumbs»

page

```
{_id: < ObjectId >,
 path: <string>,
 title: <string>,
 body: <string>,
 creation: <datetime>,
 update: <datetime>,
 tags: [ <string> ],
 parent: {_id: <ObjectId>,
         path: <string>}
 ancestors: [
         {_id: <ObjectId>, // parent
          path: <string>},
         {_id: <ObjectId>, // grandpa
          path: <string>},
         …
         ]
}
```

Tree pattern to track the ancestors.
The parent field is added to use, if necessary, the `$lookup` and `$graphlookup` functions.

For each ancestor, the **extended reference** pattern is exploited to include the path in the embedded document and avoid join operations

Acknowledgment

Bibliography

For further information on the content of these slides, please refer to the book

"Design with MongoDB"

Best Models for Applications

by Alessandro Fiori

https://flowygo.com/en/projects/design-with-mongodb/