* 文档数据库设计的基本原则
  1. 概念设计（概念模型）：
     + 对象
     + 对象之间的联系，
       - n:1 文章：用户
       - n:m 用户：用户
  2. 结构设计：
     + 对象<->文档
     + 多个对象<->文档（嵌入式）
       - 用户 {[文章]} （不合适）
       - 文章 {[评论]} （合适）
     + 应用->关系存放的位置
     + 冗余数据->提高读取速度，降低更新速度
     + 索引

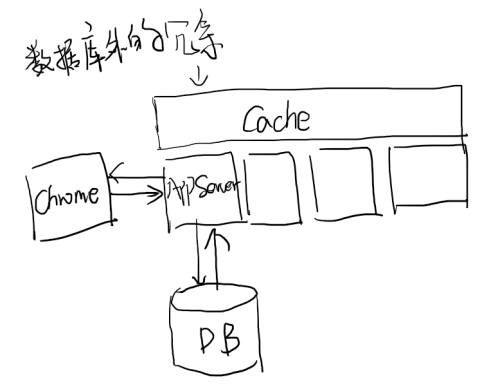
**第*1*题：在数据库设计中加入冗余属性可以有什么好处？**

A：有利于提升数据查询的性能 √

B：有利于提升数据更新的性能

C：有利于数据库的维护

D：有利于降低软件开发的复杂度



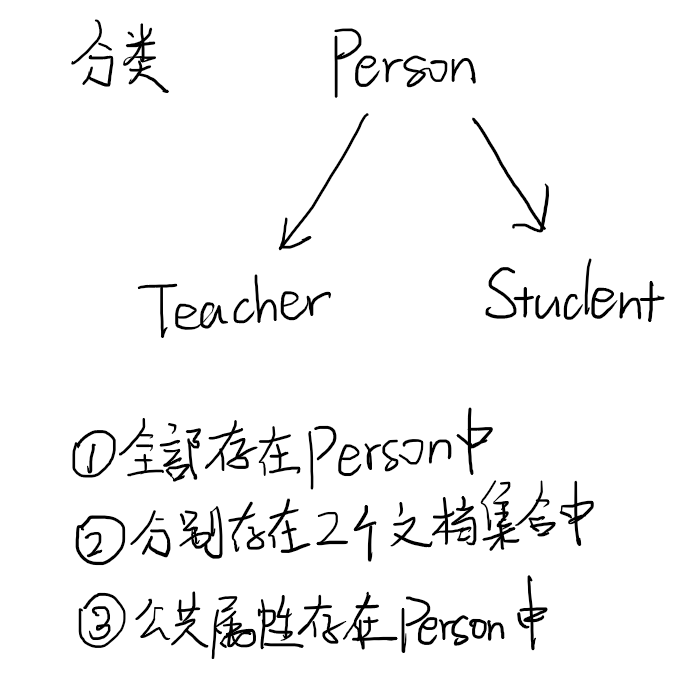
**第*2*题：我们在描述人的文档中记录他以导演、演员或编剧的身份参与过的影片。一种方式是按照身份对影片进行分类 {导演的影片[{...}], 演员的影片[{...}], 编剧的影片[{...}]}。另一种方式是在每一部影片中对参与方式进行标注 {影片[{..., 参与方式:(导演/演员/编剧)...}]}。对这两种方式的优缺点描述正确的是：**

A：前一种方式便于统计一个人参与影片的个数可以身兼数职，需要去重

B：前一种方式便于查找一个人是否参与过某部影片后一种方式可以在数组上建索引

C：前一种方式便于统计一个人导演过影片的个数 √

D：前一种方式比后一种方式几乎没有优势



**第*3*题：什么时候适合将一种对象嵌入到另一种对象中存储？**

A：子对象对父对象有明确的依附关系，比如一个书中的章节或者一个账户中的优惠券。

B：应用总是通过父对象去访问子对象，比如一篇文章的评论

C：一个子对象只属于唯一一个父对象，否则子对象会被多次存储

D：以上皆是 √

作业：

请在上一次作业中关于电影网站的数据库设计基础上进行追加设计，实现以下功能：

每一位演员在其参演的电影中都扮演一定的角色。用户在浏览电影时，除了能看到演员信息，还能看到每位演员扮演了什么角色。用户在浏览演员时，除了能看到他（或她）参演的电影，还能看到他（或她）在每一部电影中扮演的角色。

注意：一位演员可以在一部电影中扮演多个角色。同一个角色也可能由多名演员扮演，比如，年少时由一位演员扮演，年老时由另一位演员扮演。

User {

name, info

}

Staff {

name, gender, age, intro,

movie[{m\_ID, m\_name, positions[], character[]}],

comment[{content, score, u\_ID, u\_name}]

}

Movie {

name, intro, year, type, length, rating,

director[s\_ID, s\_name], scriptwriter[s \_ID, s\_name], actor[{s\_ID, s\_name, character[]}],

comment[{content, score, u\_ID, u\_name}]

}

