

## 5、mongoDb文档设计（关系处理）

### 1、mongoDb集合（表）无模式

文档在同一集合，但它们【不需要具有相同的字段】或结构集合，集合文档中的【公共字段可以包含不同类型】的数据。插入数据之前，集合不需要设计结构，可以任意插入不同的结构数据，不过实际使用一般是统一结构的数据。

### 2、在MongoDB中设计架构时有一些考虑：

根据用户设计架构（不需要三范式）。

将对象合并到一个文档中，否则分开它们（但确保不需要连接）。

为最常用的用例优化架构。

在模式中执行复杂聚合。

### 3、实例

假设客户需要他的博客/网站的数据库设计，并查看RDBMS和MongoDB架构设计之间的区别。网站有以下要求。

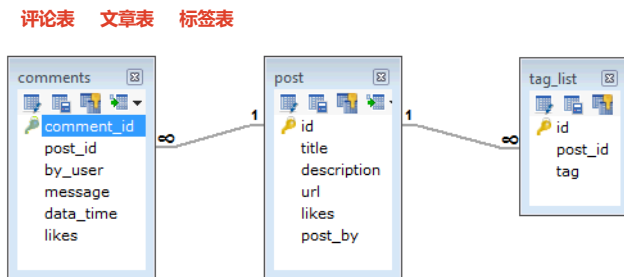
每一个文章都有：标题，url，文章内容

每一个文章都有（**关联一方**）：一个发表的用户信息（用户名，手机号，邮箱）

每一个文章都有（**关联多方**）：多个文章的标签信息（标签名1，标签名2。。。）

每一个文章都有（**关联多方**）：多个文章的评论信息（用户1-评论1，用户2-评论2。。。）

Sql的设计表：



用户表（登录的用户）

user\_id

uname

email

phone

而在MongoDB模式中，设计到【一个集合结构】--利用【数组】和【文档】中，其结构以下 -

ps：注意，尽量不要使用集合连接（表外键关联）--》a）性能比较弱 b）查询不方便（表连接查询比较弱---》3.2后好一些）

```
{
  _id: 1
  title: 'mongodb不错哟',
  description: '学习了一会儿，感觉就上手了，妈的再也不用写sql咯，不用设计表咯！',
  url: 'baidu.com/q/111',
  post_user: {uname: '曾小贤', phone: '119', '111@qq.com'}
  tags: ['mongoDb', '数据库', 'noSql'],
  comments: [
    {
      user: '曾小贤',
      message: '好厉害啊',
      dateCreated: '2015-12-12',
      like: true
    },
    {
      user: '胡一菲',
      message: '挺好玩儿的呢',
      dateCreated: '2017-11-14',
      like: false
    }
  ]
}
```

## 4、集合连接问题

### a ) DBRefs

\$ref - 此字段指定引用文档的集合

\$id - 此字段指定引用文档的\_id字段

\$db - 这是一个可选字段，并包含引用文档所在的数据库的名称

实例：

```
address: {  
  "$ref": "集合名",  
  "$id": '引用的地址id'  
  "$db": "实例名"  
},
```

查询步骤

```
>var user = db.users.findOne({"name":"Maxsu"})
```

```
>var dbRef = user.address
```

```
>db[dbRef.$ref].findOne({"_id":(dbRef.$id)})
```

**b ) 使用\$lookup ( 3.2后支持--》类似于sql的外键，可以直接表连接查询 )**