MongoDB PHP MongoDB 数据库引用

MongoDB 关系

MongoDB 的关系表示多个文档之间在逻辑上的相互联系。

文档间可以通过嵌入和引用来建立联系。

MongoDB 中的关系可以是：

- 1:1 (1对1)
- 1: N (1对多)
- N: 1 (多对1)
- N: N (多对多)

接下来我们来考虑下用户与用户地址的关系。

一个用户可以有多个地址，所以是一对多的关系。

以下是 **user** 文档的简单结构：

```
{
   "_id":ObjectId("52ffc33cd85242f436000001"),
   "name": "Tom Hanks",
   "contact": "987654321",
   "dob": "01-01-1991"
}
```

以下是 **address** 文档的简单结构：

```
{
   "_id":ObjectId("52ffc4a5d85242602e000000"),
   "building": "22 A, Indiana Apt",
   "pincode": 123456,
   "city": "Los Angeles",
   "state": "California"
}
```

嵌入式关系

使用嵌入式方法，我们可以把用户地址嵌入到用户的文档中：

```
{
   "_id":ObjectId("52ffc33cd85242f436000001"),
   "contact": "987654321",
   "dob": "01-01-1991",
```

```
   "name": "Tom Benzamin",
   "address": [
      {
         "building": "22 A, Indiana Apt",
         "pincode": 123456,
         "city": "Los Angeles",
         "state": "California"
      },
      {
         "building": "170 A, Acropolis Apt",
         "pincode": 456789,
         "city": "Chicago",
         "state": "Illinois"
      }]
}
```

以上数据保存在单一的文档中，可以比较容易的获取和维护数据。 你可以这样查询用户的地址：

```
>db.users.findOne({"name":"Tom Benzamin"},{"address":1})
```

注意：以上查询中 **db** 和 **users** 表示数据库和集合。

这种数据结构的缺点是，如果用户和用户地址在不断增加，数据量不断变大，会影响读写性能。

引用式关系

引用式关系是设计数据库时经常用到的方法，这种方法把用户数据文档和用户地址数据文档分开，通过引用文档的 **id** 字段来建立关系。

```
{
   "_id":ObjectId("52ffc33cd85242f436000001"),
   "contact": "987654321",
   "dob": "01-01-1991",
   "name": "Tom Benzamin",
   "address_ids": [
      ObjectId("52ffc4a5d85242602e000000"),
      ObjectId("52ffc4a5d85242602e000001")
   ]
}
```

以上实例中，用户文档的 **address_ids** 字段包含用户地址的对象id（ObjectId）数组。

我们可以读取这些用户地址的对象id（ObjectId）来获取用户的详细地址信息。

这种方法需要两次查询，第一次查询用户地址的对象id（ObjectId），第二次通过查询的id获取用户的详细地址信息。

```
>var result = db.users.findOne({"name":"Tom Benzamin"},{"address_ids":1})
>var addresses = db.address.find({"_id":{"$in":result["address_ids"]}})
```

MongoDB PHP

MongoDB 数据库引用

点我分享笔记