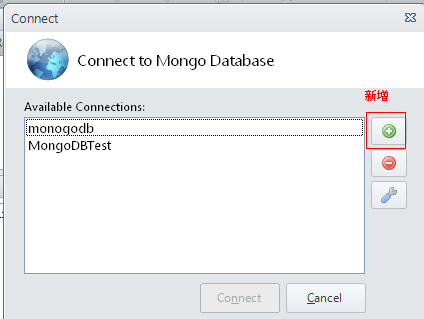
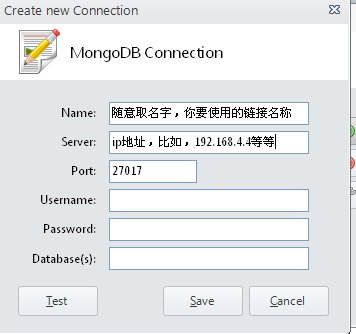
 MongoVUE 是个比较好用的MongoDB客户端，需要注册，但是可以变成永久使用，

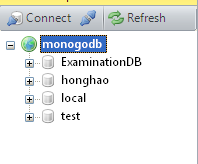
# 一、基础操作



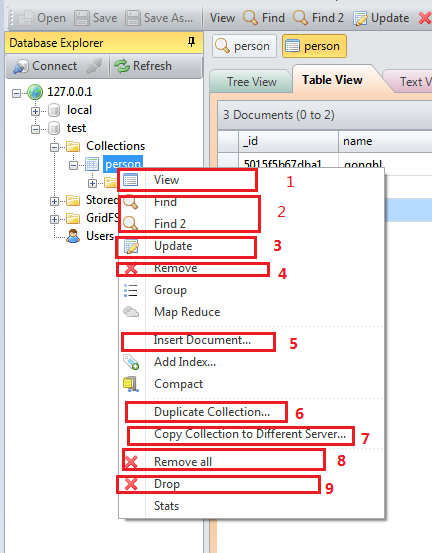
新增一个连接



进入的界面形如



# 二、进阶操作



1、查看所有数据，会有三种格式的观看方式，分别为树形，表格，bjosn，详情见1.1

2、查看特定的数据，使用的时候shell命令，需要一些基础的语法知识，详情见2.1

3、更新数据，详情见3.1

4、删除数据，**慎用，如果没有备份，删除之后无法恢复**

5、插入数据，详情见5.1

6、复制文档到统一集合，详情见6.1

7、复制集合到不同的数据库，详情见7.1

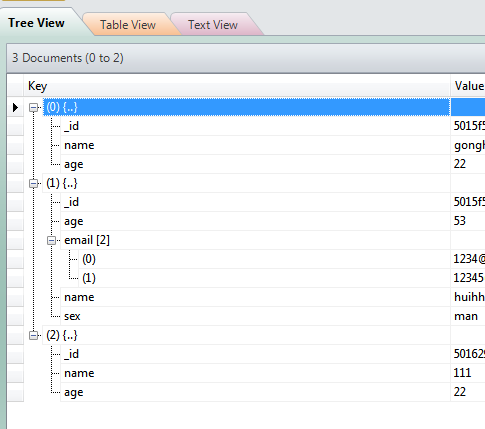
8、删除文档内所有数据、**慎用，如果没有备份，删除之后无法恢复**

9、删除数据库**慎用，如果没有备份，删除之后无法恢复**

## 1.1 查看所有数据

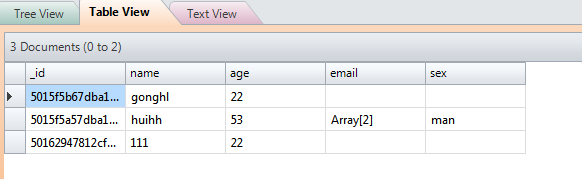
点击”view之后”，会出现文档内的数据

树形：

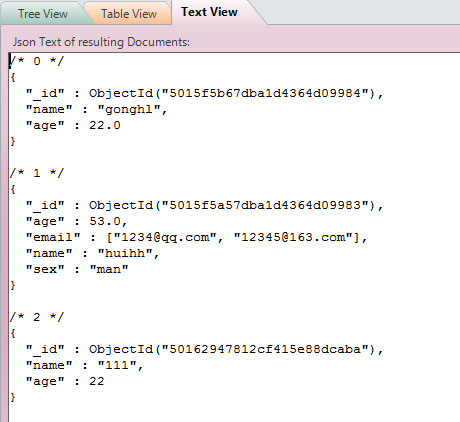


表格：

需要修改什么值的时候，可以直接点击要修改的字段进行修改，修改完成之后按回车

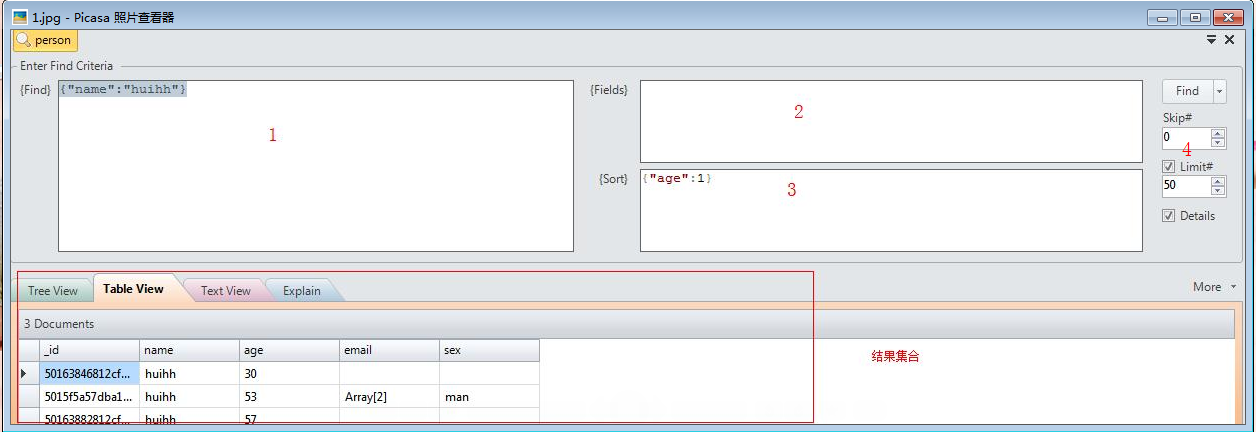


Txt形，Bjson格式的



## 2.1查看特定的数据

1. 点击”find”，输入条件之后点击find便可以进行查询，这里简单的讲一下基本的查询语句



有点模糊，可以放大点

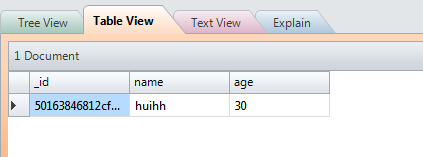
**1、区域为基本查询**

一组键值，key为要查询的字段，value为要查询字段的值，如有多个用逗号隔开

*语法 {“要查询的字段”：“要查询字段的值”，“第二个”…..}*

比如，我们想查一个名为huihh的人信息，并且年龄为30，那么写成

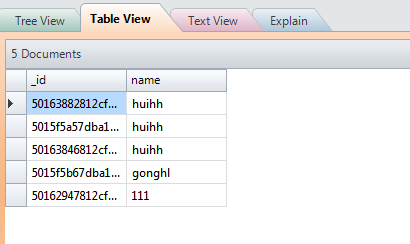
*{“name”:”huihh”,”age”:30}*



**2、字段查询**

和1有点不同，这么如果写查询语句

{"name":"huihh"}那么查询出来的数据，出了id，是一整列的name，并不会进行筛选，所以，这里我们可以写成{“name”，“”}，查询之后只显示name这一列，如果需要其他的特别查询条件，与1合用

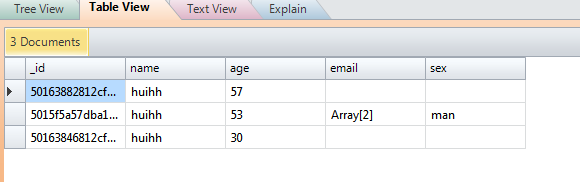


**3、排序**

方法使用一组键值对做参数，key是document里的key的名字，value是1升序或者-1降序。

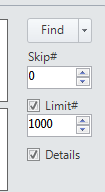
比如，查询huihh会有3个人呢，那么就使用age进行排序，

{“age”：-1}



**4、结果集的筛选**

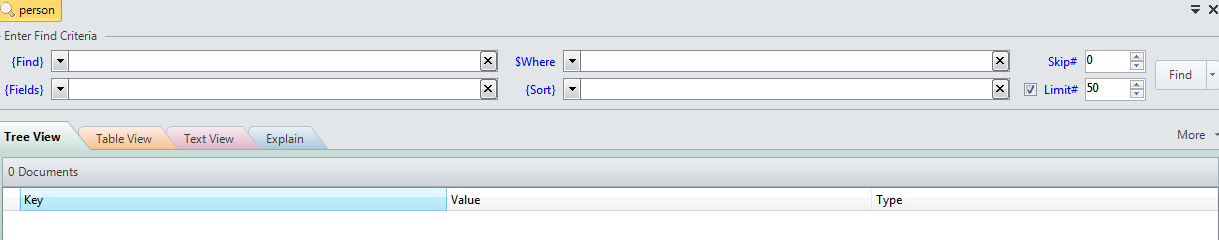
**在4区域会看到，形如**



limit函数限制返回的结果集的上限，如设置为1000那么就返回前面1000行

skip函数跳过前x个结果，返回剩余的，此处为0，就不跳过，如果设置为1，那么就跳过查询到的第一条数据，返回后面的

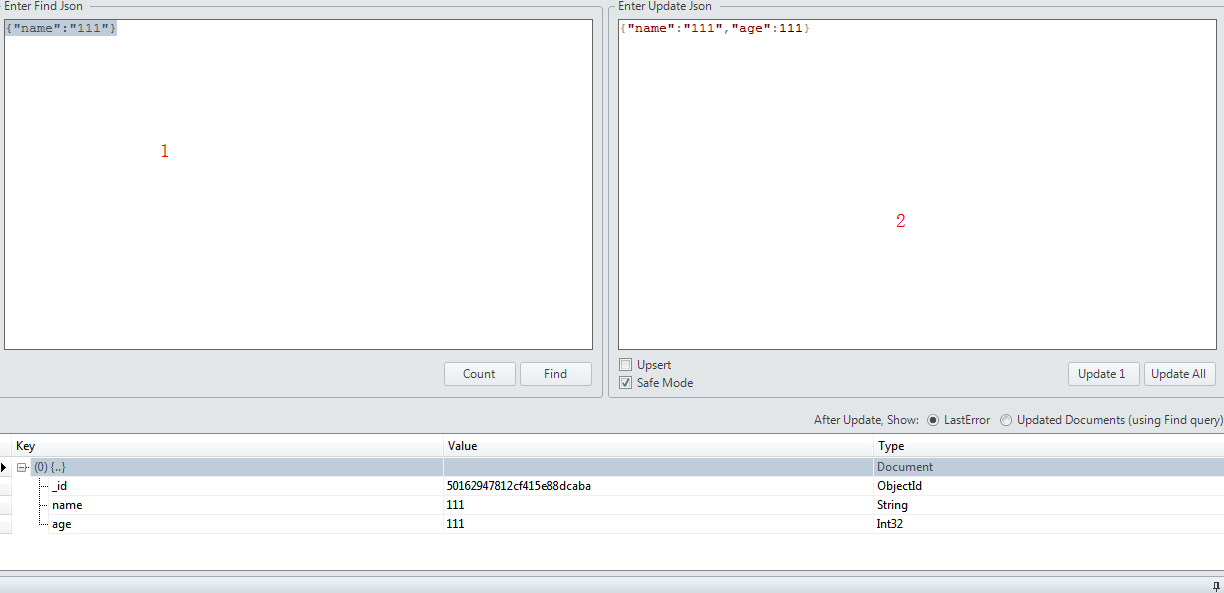
1. 点击”find2”



基本和1相同，唯一的区别是有个where语句，和这个里面是写js的，有js基础的兴趣的可以点下面的网站看看

<http://blog.163.com/wm_at163/blog/static/132173490201252610424458/>

## 3.1更新数据

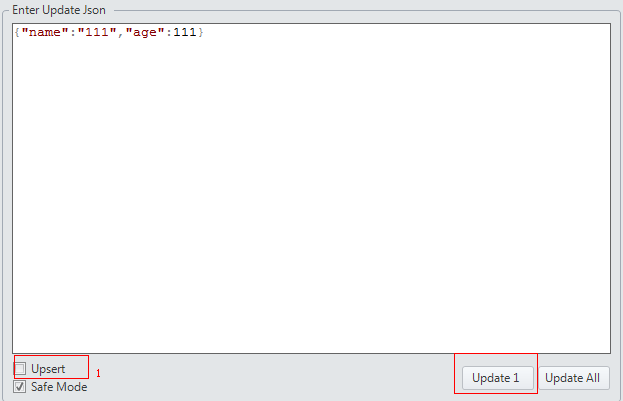


左边1为查询，就是要修改的地方，先查出来，然后再右边修改，比如这边要名为111的人，将其年龄改为111岁那么

左边写{“name”：“111”}右边写{"name":"111","age":111}，

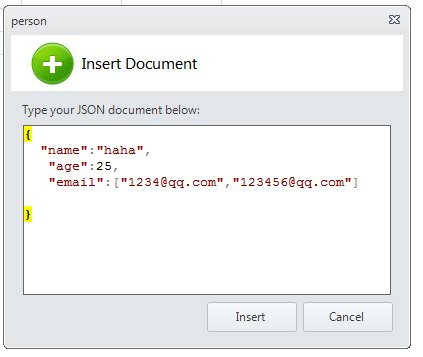
左边写完可以点击下面的find看看文档内关于这个名字的信息，

点击update1进行修改，



1：当此处选择的时候，点击update1会先查看这条数据，如果没有就插入，此时，左边不需要写

## 5.1 插入数据

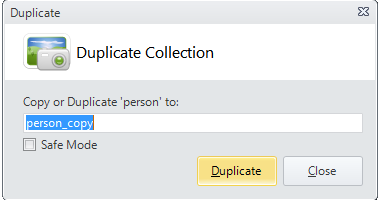


此处插入的数据，为bjosn格式，键值对

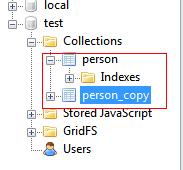
“key”:value

Key为要插入的字段名称，vlaue为对应的值，如果有多个值，可以用[]括起来

## 6.1复制文档到同一集合



点击之后，



7.1复制集合到不同的数据库

