# 数据库笔记

## MySQL

## 第二节 MongoDB

### 第一章 MongoDB的安装

#### 1.1 MongoDB的安装

##### 1.1.1 windows上安装MongoDB

##### 1.1.2 linux 上安装MongoDB

#### 1.2 MongDB 的配置

##### 1.2.1 配置环境变量

##### 1.2.2 安装windows服务

1. 创建data目录

2. 创建data/logs/mongodb.log 文件

mongod --logpath "C:\Program Files\MongoDB\data\logs\mongodb.log" --logappend --dbpath "C:\Program Files\MongoDB\data" --directoryperdb --serviceName "MongoDB" --serviceDisplayName "MongoDB" –install

##### 1.2.3 安装可视化工具RoboMongoDB

### 第二章 MongoDB 基础

#### 2.1 MySQL与MongoDB的对应关系

1. MySQL中的数据表，对应MongoDB中的集合

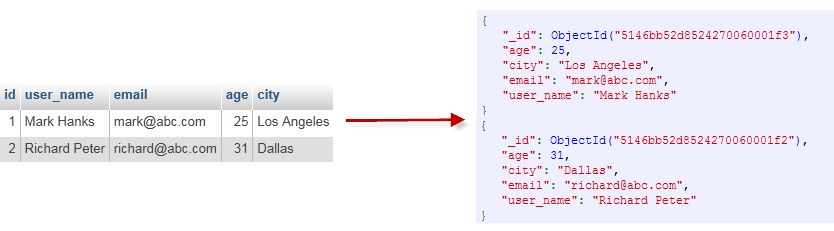
2. MySQL中的数据库，对应MongoDB中的数据库

3. MySQL中的每一行数据，对应MongoDB中的文档

4. MySQL中的每一列的数据字段，对应MongoDB中的field—域

5. MySQL中的主键，对应MongoDB中的"\_id",自动设置。

对应关系，如下图：



#### 2.2 MongoDB的基本操作

##### 2.2.1 MongoDB的连接

mongo

##### 2.2.2 创建/删除数据库

创建/切换数据库

use testdb //必须保证数据库不为空，才可以显示数据库

查看所有数据库

show dbs

删除数据库

use testdb

db.dropDatabase()

删除集合

db.集合名.drop()

##### 2.2.3 插入文档

use 数据库名 //切换到要操作的数据库

//文档插入操作

db.集合名.insert({

"name":"mike",

"age":23

})

查看已经插入的文档

db.集合名.find()

文档插入操作2

将文档定义为变量

document = {

"name":"mike",

"age":23,

"city":"beijing"

};

执行insert()命令

db.test.insert(document)

##### 2.2.4 更新文档

更新文档

db.collection.update(

<query>, //查询条件。类似MySQL中的where

<update>, /更新内容。类似MySQL中的set

{

upsert: <boolean>, //true,表示如果没查到，就插入

multi: <boolean>, //true，表示都更新；false，表示只更新第一条

writeConcern: <document> //指定抛出的异常级别

}

)

示例：

插入文档：

db.col.insert({

title:"mongoDB",

descript:"MongoDB教程",

by:"Gaylen"

likes:100

})

更新文档

db.col.update(

{'title':'mongoDB'},

{$set:{'title':'MongoDB'}},

{upsert:true}

)

查看文档

db.col.find().pretty()

替换文档

db.col.save(

{

"\_id":xxx,

"titile":xxx

}

{writeConcern:<document>} //可选，抛出异常级别

)

##### 2.2.5 删除文档

删除文档语法：

db.collection.remove(

<query>,

{

justOne: <boolean>, //true表示只删除一个文档

writeConcern: <document> //可选 抛出异常级别

}

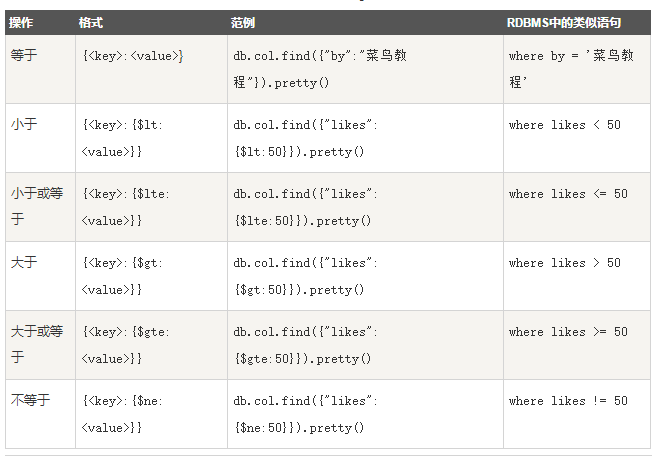
)

示例：

db.col.remove({title:"MongoDB"},1) //只删除一条，justOne = 1

##### 2.2.6 查询文档

MongoDB查询与MySQL查询比较:



$gt -------- greater than >

$gte ------- gt equal >=

$lt -------- less than <

$lte ------- lt equal <=

$ne -------- not equal !=

$eq ------ equal =

MongoDB的“and" 查询：

db.col.find({key1:value1, key2:value2}).pretty()

MongoDB 的 "or" 查询

db.col.find(

{

$or: [

{key1: value1},

{key2:value2}

]

}

).pretty()

### 第三章 MongoDB 进阶

#### 3.1 聚合

#### 3.2 索引

#### 3.3 复制

#### 3.4 分片

### 第四章 MongoDB 高级

#### 4.1 MongoDB 的Map Reduce

#### 4.2 MongoDB 的GridFS

#### 4.3 MongoDB 的全文检索

## 第三节 Redis开发