## 配置

配置conf文件目录下的conf.xml文件：

<?xml version="1.0" encoding="UTF-8"?>

<recipe>

<mysql>

<url>jdbc:mysql://127.0.0.1:3306/wcache</url>

<user>root</user>

<password></password>

</mysql>

<table>

<name>test</name>

</table>

</recipe>

<mysql>标签以及子标签对应需要连接的数据库

<table>标签中存放需要建立缓存的数据库表

## 使用

用WCache.getInstance()获得WCache对象实例wc

Table test = wc.tables.get(“test”)获得表名为test的表的对象

### 普通查询

**Select语句：**

test.query().select()

.where(“属性名字”).eq(属性的值)

.andWhere(“属性的名字”).bt(属性的值)

.orWhere(“属性的名字”).lt(属性的值)

.execute(回调对象opcb)

opcb对象所属类继承OutPutCallBack，并重载public void out(List<Map<String, Object>> r)函数用于完成对查询结果的操作

eq()表示等于，bt()表示大于，lt()表示小于

**Delete语句：**

test.query().delete().where(“属性名字”).eq(属性的值).execute(null)

delete()后面跟着的where子句的用法和select语句中的一样

**Update语句**

test.query().update(params).where(“属性的名字”).eq(属性的值).execute(null)

params是一个HashMap，key是需要更新的属性名字，value是需要更新的属性的值

update()后面跟着的where子句的用法和select语句中的一样

**Insert语句**

test.query().insert(params).execute(null)

params是一个HashMap，key是属性名字，value是属性的值

### 事务查询

Trasaction t = new Trasaction() 获得事务对象

用select语句解释如何使用事务查询：

test.query().select().where(“属性名字”).eq(属性的值).addToTrasaction(t, opcb)

t.trasactionExecute() 执行事务