

**本科实验报告**

|  |  |
| --- | --- |
| 课程名称： | 大规模信息系统构建技术导论 |
| 实验名称： | 分布式MiniSQL |
| 姓 名： | 米博宇 |
| 学 院： | 计算机学院 |
| 系： | 软件工程系 |
| 专 业： | 软件工程 |
| 学 号： | 3200102888 |
| 指导教师： | 鲍凌峰 |

2023年 5 月 15 日

**浙江大学实验报告**

实验名称： 分布式MiniSQL客户端 实验类型： 设计实验

同组学生： 李毅桐、王粤龙 实验地点：

## 一、实验内容

* 设计分布式MiniSQL的客户端模块
* 客户端模块启动后，首先要求用户名输入，然后连接zookeeper服务器并创建相应的客户端节点。
* 成功创建节点后，在命令行中持续接收用户的SQL语句输入。
* 根据用户的SQL语句输入，调用业务逻辑得到对应的输出结果
* 在用户输入”quit”后，客户端结束运行。

## 二、功能描述

### 1. 语法分析

客户端支持对用户输入的SQL语句进行语法分析。输入的SQL语句有语法错误时，直接提示用户语法错误，无需调用其他方法。SQL语句没有问题时，客户端可以判断其操作类型，并执行相应的业务逻辑。

### 2. 支持的SQL语句类型

* 创建数据表
  + 支持创建数据表，支持MySQL的所有数据类型
* 删除数据表
  + 支持删除存在的数据表
* 插入数据
  + 支持向存在的数据表中插入制定值的数据
* 删除数据
  + 支持从表中删除一条或多条数据
* 查询数据
  + 支持多条件查询，条件之间可以AND，OR等逻辑词分隔
  + 支持多表查询，笛卡尔积和left join等复杂join操作

### 3. 与Master服务器交互

客户端连接成功后，会在zookeeper集群中注册一个对应的节点，由Master服务器对应的节点持续监听该节点的变化。需要进行数据表操作时，客户端会将操作的信息（操作类型、参数等）写入对应节点的缓冲区。Master服务器对应的节点监听到变化时，根据操作信息执行相应的数据库操作。执行完成后，将执行结果写入客户端对应的节点缓冲区，客户端节点监听到缓冲区变化后，将执行结果打印在命令行中。

## 三、接口说明

### 1. 外部接口

客户端程序从命令行中接受用户键入的SQL语句，在获得SQL语句执行结果后，将结果输出至命令行供用户查看。

### 2. 内部接口

本模块调用JSQLParser的解析函数接口获取SQL语句的类型并得到所有操作表的名称。调用zookeeper的getData接口向Master服务器获取当前数据库内所有表名，判断操作表是否存在。

## 四、工作原理

### 1. 连接zookeeper集群

新客户端打开后，首先要求用户输入用户名，作为在zookeeper集群中的唯一识别方式（如果用户名被占用则提示重新输入）。然后DistributedClient类的成员变量zkClient被初始化为一个zookeeper对象，名称为用户名，与zookeeper集群连接。

zkClient对象监听Master节点。Master节点缓冲区中存放着数据库中所有数据表和region服务器的对应关系。对应关系发生变化时，zkClient重新从Master获取对应关系表，并存储在自身的成员变量tableRegionMap中。

zkClient对象监听自身节点，自身节点存放SQL语句的执行结果。当Master或Region将SQL的执行结果写入自身节点缓冲区时，DistributedClient将其打印到命令行中。

### 2. 获取用户输入

连接和初始化操作完成后，客户端开始获取SQL语句输入。输入规则与MySQL相同：用户可输入任意多行，以换行符分隔，分号作为结束标记。

### 3. 对SQL进行解析

调用JSQL Parser模块对输入的SQL语句进行解析。首先检查是否有语法错误，如有，提示用户并要求重新输入。否则，判断SQL语句的操作类型，并提取SQL语句中所有的表名。支持的操作有：CREATE\_TABLE, DROP\_TABLE, SELECT, DELETE, INSERT, UPDATE。

#### CREATE\_TABLE

检查要创建的表是否存在。如果存在，则提示用户；否则将用户名和SQL语句写入Master节点的缓冲区，由Master节点负责完成对应操作。

对于其他操作，检查SQL语句的表名在tableRegionMap中是否都存在。如果有表不存在，提示用户；否则进行对应操作，如下所示。

#### DROP\_TABLE

将用户名和SQL语句写入Master节点的缓冲区，由Master节点负责完成对应操作。

#### DELETE, INSERT, UPDATE

根据tableRegionMap查询得知需要修改操作的表所在的Region服务器。直接向对应服务器节点的缓冲区写入用户名和该SQL语句，由Region负责执行。

#### SELECT

首先判断是单表查询或多表查询。

如果是单表查询：处理同上：根据tableRegionMap查询得知需要查询操作的表所在的Region服务器。直接向对应服务器节点的缓冲区写入用户名和该SQL语句，由Region负责执行。

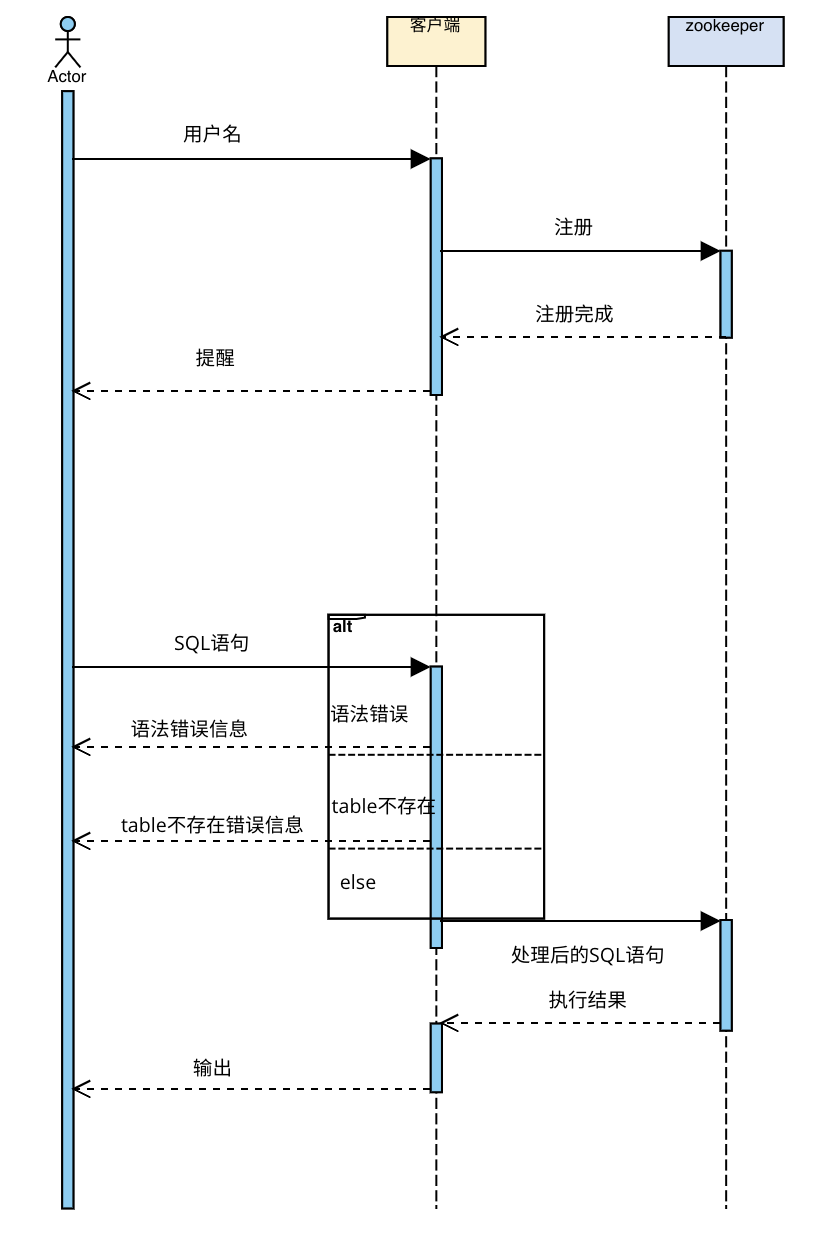
如果是多表查询：根据tableRegionMap判断需要查询的表是否在同一个region服务器上。

如果需要查询的表都在同一个region服务器上。仅向对应服务器节点的缓冲区写入用户名和该SQL语句，由该Region负责执行。否则，将需要查询的表和用户名、SQL语句写入Master的缓冲区，由Master负责执行并返回查询结果。

### 4. 输出执行结果

在SQL语句有问题（语法错误，操作的Table不存在等）。客户端将直接打印相关的报错。SQL语句执行完毕后，将向客户端节点的缓冲区中写入执行结果，监听到变化后直接打印结果。如果是SELECT语句，还需要逐行打印查询结果。

客户端服务的时序图如下：



## 单元测试

### 5.1用户注册

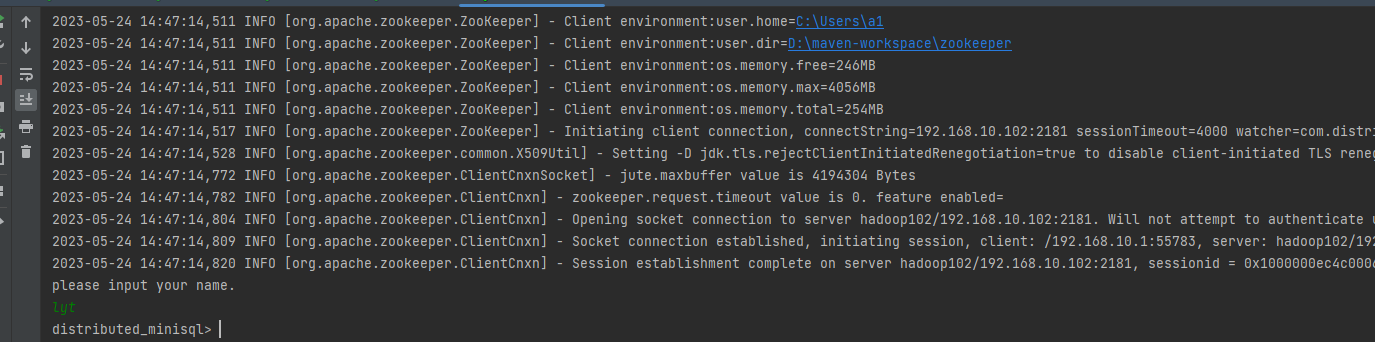
#### 注册新用户名

输入：打开客户端后，输入新的用户名

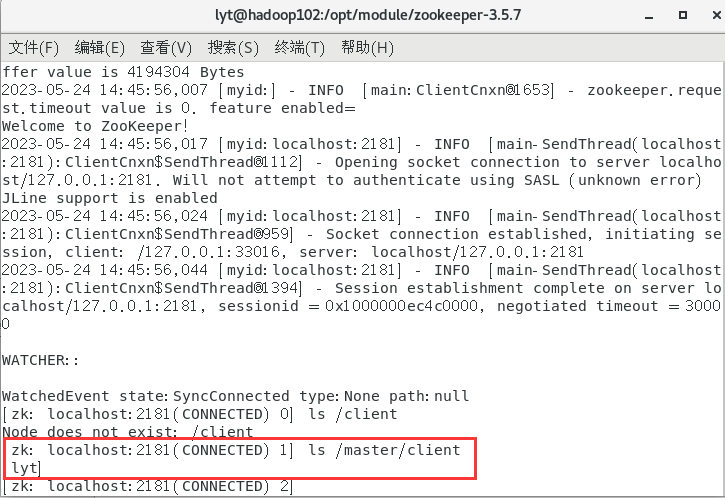
预期输出：客户端提示注册成功，zookeeper中对应节点被创建

实际输出：

客户端提示：显示sql语句输入的提示符



Zookeeper 新增名称为用户名的节点



是否通过：通过

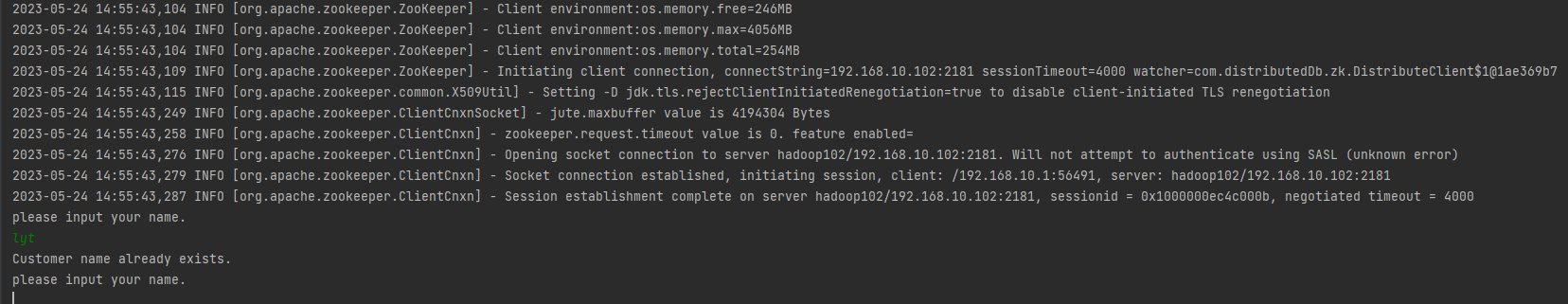
#### 注册已存在的用户名

输入：打开客户端后，输入已被注册的用户名

预期输出：客户端提示用户名已存在

实际输出：

客户端提示用户名已存在



是否通过：通过

### 5.2 SQL语句异常处理

#### SQL语句有语法错误

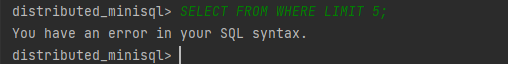
输入：有语法错误的SQL语句

预期输出：客户端提示语法错误

实际输入：“SELECT FROM WHERE LIMIT 5;”

实际输出：

客户端提示语法错误



是否通过：通过

#### 待创建的表已存在

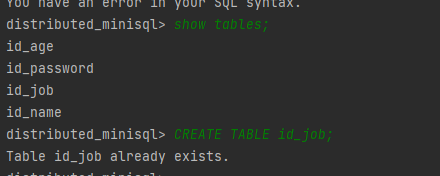
输入：CRETE TABLE操作的SQL语句，名称为语句中表名的数据表已存在

预期输出：客户端提示该表已存在

测试输入：“id\_job” 表已存在后，输入“CREATE TABLE id\_job;”

实际输出：

客户端提示表已存在



是否通过：通过

#### 查询/更改/删除的表不存在

输入：DROP TABLE，INSERT，UPDATE，DELETE，SELECT操作的SQL语句中，名称为语句中表名的数据表之一不存在。

预期输出：客户端提示数据表不存在

测试输入：“student”表不存在时，分别输入

“DROP TABLE student;”

“INSERT INTO student VALUES (1234, ‘Alice’);”

“UPDATE student SET id=1233 WHERE id=1234;”

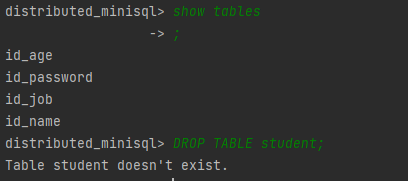
“DELETE FROM student WHERE id=1234;”

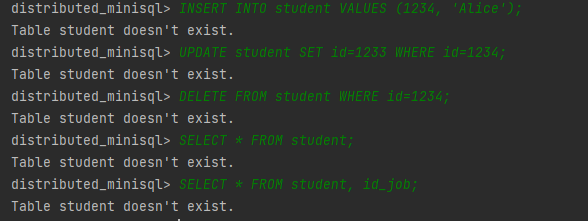
“SELECT \* FROM student;”

“SELECT \* FROM student, id\_job;”

实际输出：

客户端提示“student”表不存在





是否通过：通过

### 5.3 SQL语句发送

**注：本节测试经过初步检查（5.2中异常检测）无异常的SQL语句能否被正确发送至期望的服务器。**

#### 建立/删除数据表

输入：CREATE, DROP类型的SQL语句

预期输出：客户端将带有客户端用户名、SQL语句的消息写入mastersql节点缓冲区。

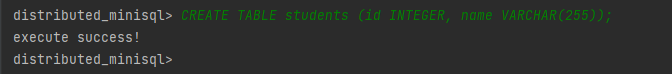
实际输入：输入

“CREATE TABLE students (id INTEGER, name VARCHAR(255));”

“DROP TABLE students;”

实际输出：

每当输入一条SQL语句后，mastersql节点中被写入包含用户名和SQL语句的信息









是否通过：通过

#### 更改数据表

输入：INSERT，UPDATE，DELETE类型的SQL语句

预期输出：客户端将带有客户端用户名、SQL语句的消息写入SQL语句操作的数据表所在Region服务器对应的的zookeeper节点的缓冲区。

实际输入：输入

“INSERT INTO student VALUES (1234, ‘Alice’);”

“UPDATE student SET id=1233 WHERE id=1234;”

“DELETE FROM student WHERE id=1234;”

实际输出：

输入第一条SQL语句后，student表所在的Region服务器对应的zookeeper节点中被写入包含用户名和SQL语句的信息









是否通过：通过

#### 单表查询

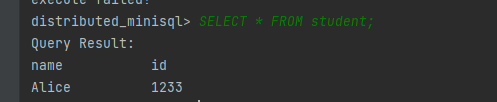
输入：仅查询一张数据表的SQL语句

预期输出：客户端将带有客户端用户名、SQL语句的消息写入S待查询所在Region服务器对应的的zookeeper节点的缓冲区。

实际输入： “SELECT \* FROM student;”

实际输出：

student表所在的Region服务器对应的zookeeper节点中被写入包含用户名和SQL语句的信息。





是否通过：通过

#### 多表查询

输入：仅查询多张数据表的SQL语句

预期输出：如果查询的数据表均在同一个Region服务器上，将带有客户端用户名、SQL语句的消息写入其所在Region服务器对应的的zookeeper节点的缓冲区。否则，将用户名和SQL语句的信息写入mastersql节点的缓冲区。

实际输入：

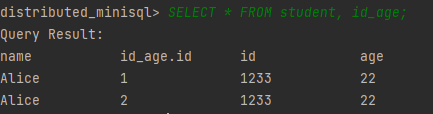
假设表，id\_job在Region1上，student, id\_age在Region2上，输入

“SELECT \* FROM student, id\_age;”

“SELECT \* FROM student, id\_job;”

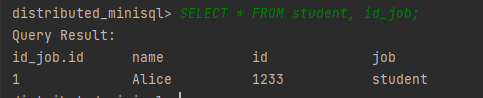
实际输出：

输入第一条SQL语句时，Region2对应的节点缓冲区写入用户名和SQL语句。





输入第二条SQL语句时，mastersql节点缓冲区写入用户名和SQL语句。





是否通过：通过

### 结果显示

输入：Region或Master完成数据库操作后，将结果写入客户端对应的节点

预期输出：结果显示在命令行界面

实际输入：使用zookeeper命令行在客户端节点中写入“Successfully executed”

实际输出：命令行显示“Successfully executed”





是否通过：通过

## 六、开发心得

在开发中，我体会到了客户端设计的难度所在。需要仔细设计接收用户输入并显示输出的逻辑，并进行全方面的测试确保模块的可靠性。

在进行小组合作的时候，我体会到了模块之间解耦并制定统一的信息传输格式的重要性。必须完整地考虑到需要传输的各种信息，才能避免开发过程中格式的更改带来的额外工作量。

此外，在设计客户端模块的时候，也必须参与其他模块的设计并熟悉相关的函数，调用时才能做到得心应手。

总之，这个项目让我了解并掌握了zookeeper的使用和编程，并增强了系统设计能力和沟通合作能力。