

Catalog & Index Manager 模块详细设计

3180103608 杨凌霄

1. Catalog Manager 模块设计

1) 模块综述

本模块主要对表格元数据进行管理，需要完成的功能如下所示：

- 定义数据库中所有表的信息，包括表的名称、表中主键、定义在表上的索引等
- 提供给 Interpreter 模块接口，以调用保存表格元数据的接口
- 接口调用 Master Server 或 Region Server 的 thrift 接口，实现保存与删除
- 迎合缓存策略，已经保存的表格元数据再次查询时不用再调用 Master Server 方面的问询接口

2) 涉及语句

创建表语句

```
CREATE TABLE 表名 (  
    列名 类型  
    列名 类型  
    .....  
    PRIMARY KEY(列名)  
);
```

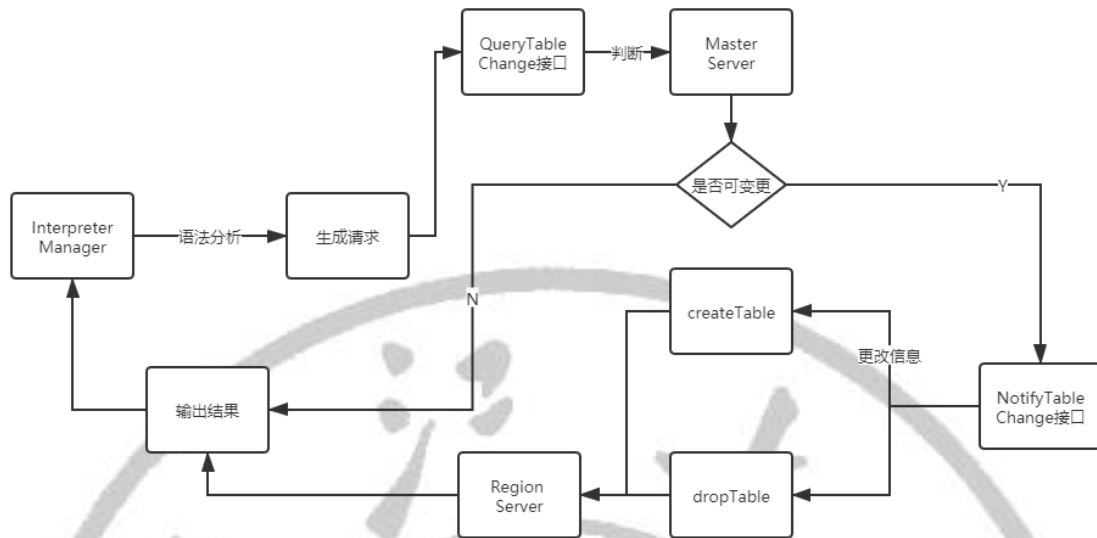
删除表语句

```
DROP TABLE 表名;
```

3) 相关部分接口

```
public static void createTable(Table newTable) {}  
public static void dropTable(Table oldTable) {}
```

4) 模块流程图



2. Index Manager 模块设计

1) 模块综述

本模块是 Client 中建立 B+树以及管理索引的模块。需要完成的功能如下所示：

- 管理和维护表格上的索引，其中包括表格的主键（主索引）
- 在新建 Table 时，默认创建主键索引，以 attribute 的值为 Key，数据结构 Row 为 Value 构建 B+树
- B+树类实现 Serializable 接口，序列化保存到服务器上
- 由于数据结构 Data 的定义，B+树要分别对 Integer, String, Float 定义判断策略
- 在新建索引时也要完成 B+树的创建
- 提供给 Record Manager 接口，方便在插入查询等语句时实现速度优化

2) 涉及语句

创建索引语句

CREATE INDEX 索引名 ON 表名 (列名);

删除索引语句

DROP INDEX 索引名;

插入记录语句

INSERT INTO 表名 VALUES (值 1, 值 2, ……., 值 n);

删除记录语句

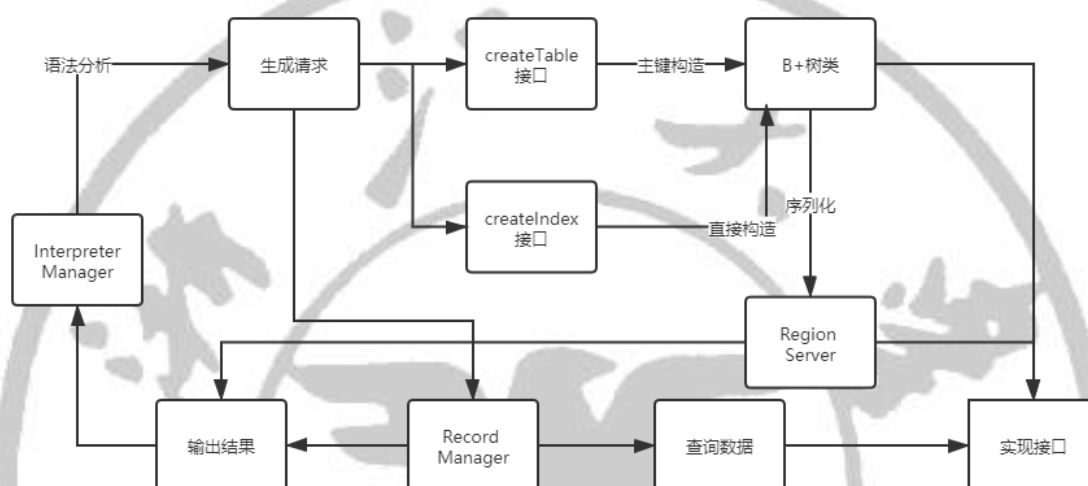
DELETE FROM 表名;

DELETE FORM 表名 WHERE 条件;

3) 相关部分接口

```
public static void createIndex(Index newIndex, Table tmpTable){}
public static void dropIndex(Index oldIndex, Table tmpTable){}
```

4) 模块流程图



3. 开发心得

我在这次的分布式 MiniSQL 的编写过程中，学习到了很多新知，在 debug 的过程中也训练了自己的代码调试能力和耐心。在传统的 MiniSQL 实验中，我们在大二下的数据库课程中也制作过本地的数据库，但比如表格元数据的保存都是在本地磁盘上的，这次新颖的分布式架构让我感觉很有挑战，但又学会了很多。比如我在保存表格元数据时，就要调用 thrift 接口，一些数据结构因为不是同一台机器，也要定义不同的地方，而且还要完成属性的复制，调用接口时还要兼顾安全性、容错性以及副本维护的方面。所以分布式的数据库要考虑到的地方还是较为繁杂的。

另外，一些令人耳目一新的代码调试手段、代码开发框架还是很令人感兴趣的，比如我在运用语法糖、Beanutils、thrift、zookeeper 等规模大小不一工具时，都学到了很多，加快了代码编写和调试时候的速率。开发一整个项目，我不仅学习了诸多的框架如何使用，还提高了自己在处理问题时候的全面性思考能力，最后答辩中鲁老师也提出了架构的另一种策略，更适用于大规模数据，受益良多。