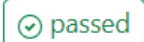





# DBTrain Lab 4 Report

刘泓尊 2018011446 计84

CI job ID #145685

Status	Job	Pipeline
 passed	#145685  master  0d658710	#71537 by  刘泓尊's avatar

## MVCC

实现了一种简单的只支持Insert+Search相关操作的MVCC机制，支持索引。具体的逻辑如下。

### Transaction

每个事务维护：事务ID、当前活跃事务、当前事务的Insert序列和该事务所操作的表。

**Begin:** 在事务Begin的时候，分配ID并创建事务，将该事务添加到活跃列表中。事务ID的分配是递增的。

**Commit:** 在事务Commit的时候，将该事务从活跃列表中删除。

**Abort:** 在事务Abort的时候，通过后入先出顺序将该事务Insert序列的操作还原（因为只有Insert操作，所以只需删除对应行），并将该事务从活跃列表中删除。

### Operations

**Create:** 在创建表的时候，如果属于某事务，则增加一列，用于记录该事务的ID。

**Insert:** 在插入时如果 `txn` 不是空指针，那么创建一个Field存放事务ID，并存入表中。同时将该插入操作记录在事务的“插入记录”中

**Search:** 查找记录时，根据当前事务ID、record中的事务ID、当前活跃事务来判断该记录是否可见：如果record中事务ID小于当前事务ID且已经不在活跃列表中，就是可见的。在返回结果的时候，注意删除事务ID这一Field。

### Update & Delete

如果要支持Update和Delete就比较繁琐了，因为在Abort的时候不能简单通过删除来还原操作了，而是要记录本次修改之前的数据。所以可以通过一个链表来实现，每个记录隐式增加一个 `prevVersion` 指针，指向上一版本的记录，同时两个记录都需要记录事务ID。这样，在事务Abort之后，就可以通过遍历链表找到对应版本，并覆盖当前版本。其实也是一个update/delete undo log。我查了一下Mysql的实现方式，基本也是这样做的。

## 测试结果

**Start to test Lab4**

Database Init.

Build Finish.

[=====] Running 4 tests from 1 test suite.

[-----] Global test environment set-up.

[-----] 4 tests from MVCCTests

[ RUN ] MVCCTests.CommitInsert1

[ OK ] MVCCTests.CommitInsert1 (2 ms)

[ RUN ] MVCCTests.CommitInsert2

[ OK ] MVCCTests.CommitInsert2 (1 ms)

[ RUN ] MVCCTests.AbortInsert1

[ OK ] MVCCTests.AbortInsert1 (0 ms)

[ RUN ] MVCCTests.AbortInsert2

[ OK ] MVCCTests.AbortInsert2 (0 ms)

[-----] 4 tests from MVCCTests (3 ms total)

[-----] Global test environment tear-down

[=====] 4 tests from 1 test suite ran. (3 ms total)

[ PASSED ] 4 tests.

YOU HAVE 4 DISABLED TESTS

**Start to send result**

**Result has been sent**

**Passed lab4 test**