

# Q&A&BugFix

RELEASE 模式下，ASSERT 语句会被优化，从而不被执行。在小规模数据测试时，为了确保 ASSERT 能够被正常运行起到检测作用，请使用 DEBUG 模式进行。RELEASE 模式下 DBStorageEngine 类的 Bug 修复详见：[BugFix](#)。该 Bug 会导致 DBStorageEngine 时预分配 CATALOG\_META\_PAGE 以及 INDEX\_ROOTS\_PAGE 这两个数据页的过程被跳过。

相较于去年，minisql 做出了如下更新：

1. 增加了 Recovery Manager 和 Lock Manager 两个模块
2. 修复了去年实验过程中发现的若干 bug

## 10.0 START

1. 在编译时，碰到了包含 'march=native' 相关的错误时，只需在 CMakeList.txt 文件中搜索 'march=native'，将其去掉即可。
2. 运行单个测试时，出现 IMPORTANCE NOTICE: This test program did not call testing::InitGoogleTest() before calling RUN\_ALL\_TESTS(). This is INVALID. 类似的错误，重新 cmake 一下，再编译单个测试文件应该就能够解决。如果仍未解决，目前可以暂时使用运行所有测试来代替。运行所有测试时，可以通过指定 Filter（在 main\_test.cpp 中注释的那行）来选择需要运行的测试用例。
3. 很多同学在用 GDB 调试时发现调试信息很少，大概率是没有用 DEBUG 模式。在执行 cmake 命令时，指定 -DBUILD\_TYPE=Debug 再重新编译即可。
4. 编译时遇到如下类似错误，是因为环境中已经安装了 GTest，与 ThirdPart 中的 GTest 产生冲突。如果将报错的 GTestTargets.cmake 文件暂时移走或删除还不能解决问题，有以下解法：

1. 暂时移除 anaconda 的环境变量，或者删除 anaconda 里面的 gtest
2. 寻找 debug 生成文件里面的 CMakeCache.txt，检查里面和 glog 以及 gflag 相关的变量。如果如下图所示，有变量仍然链接到 anaconda 里面的 gflags，则在 cmake 的时候增加编译选项 -Dgflags\_DIR=/opt/homebrew/lib/cmake/gflags，其中路径替换成自己安装的 gflags 路径。

```
Ubuntu-20.04: /usr/bin/cmake -DCMAKE_BUILD_TYPE=Debug -G 'CodeBlocks - Unix Makefiles' -S /mnt/d/minisql -B /mnt/d/minisql/cmake-build-debug-wsl
CMake Warning at CMakeLists.txt:25 (MESSAGE):
  Could NOT find clang-format.

CMake Error at /mnt/d/anaconda_download/Library/lib/cmake/GTest/GTestTargets.cmake:37 (message):
  Some (but not all) targets in this export set were already defined.

  Targets Defined: GTest::gtest;GTest::gtest_main

  Targets not yet defined: GTest::gmock;GTest::gmock_main

Call Stack (most recent call first):
  /mnt/d/anaconda_download/Library/lib/cmake/GTest/GTestConfig.cmake:32 (include)
  thirdparty/glog/CMakeLists.txt:71 (find_package)

-- Configuring incomplete, errors occurred!
See also "/mnt/d/minisql/cmake-build-debug-wsl/CMakeFiles/CMakeOutput.log".
See also "/mnt/d/minisql/cmake-build-debug-wsl/CMakeFiles/CMakeError.log".

[Finished]

< gflags_DIR:PATH=/usr/lib/x86_64-linux-gnu/cmake/gflags
---
> gflags_DIR:PATH=/mnt/d/anaconda/Library/lib/cmake/gflags
413c430
```

## 10.3 INDEX

1. `BPlusTree::FindLeafPage` 函数中，参数 `page_id` 的含义是什么？

是以 `page_id` 的那一页为 b+ 树的初始节点来找 leafpage。如果 `page_id == INVALID_PAGE_ID`，那么从根节点开始查找。

1. `InternalPage::MoveAllTo` 函数中，这个函数的意思是什么？参数 `middle_key` 的含义是什么？

这个函数是在合并的时候使用，把右节点合并到左节点。因为在右节点里，最左边 value 的 key 值是存在 parent 里的，所以要从 parent 获取。这个 `middle_key` 即是最左边 value 的 key 值。

## 10.4 CATALOG

1. `IndexMetadata::Create` 函数中，参数 `key_map` 和 `CatalogManager::CreateIndex` 函数中，参数 `index_keys` 的含义分别是？

PS：善用全局搜索，用全局搜索一下这些变量在哪里被用到可以更好地帮助理解这个变量的作用。

对于前者，通过全局搜索引用，可以发现它在 `Schema::ShallowCopySchema` 中有被用到，且该函数的注释中给出了一个例子。实际上 `key_map` 可以简单地理解成，索引键分别位于元组中的哪几列（即它们在元组中的下标）。对于后者，表示创建索引时，哪几列（列名，字符串类型）是需要作为索引键的。

1. `TableMetadata` 类中，`root_page_id_` 指的是 `TableHeap` 的 `root_page_id_`
2. `CatalogTableTest` 中，`schema` 采用 `make_shared<Schema>` 方式建立，发生了重复析构的问题，这是怎么回事呢？

这是因为 `CreateTable` 函数中，没有采用正确的方式拷贝 `schema`。建议看一下 `DeepCopySchema` 和 `ShallowCopySchema` 函数，并选择正确的拷贝方式。

## 10.5 EXECUTOR

1. 对于 `PRIMARY KEY` 的列如何进行区分？

对于 PK 列，可以在表创建的时候，将 PK 列的信息写入 `TableMetadata` 中，此外，在 `TableInfo` 对象中，也可以增加有关 PK 列的属性，以在内存中维护 PK 信息。

1. 语法树可视化时，生成的 `DOT` 文件在哪里？

正常情况下，是在 `build` 或 `build/bin` 或者 `build/test` 下面，可以通过 `find` 命令查找。

1. 事务 `Transaction` 模块需要实现吗？

今年不需要实现，因为没有确定好事务的框架。但是明年估计就要了（所以说好好学，争取不重修）。

1. `ExecuteContext` 这个类有什么用呢？

用于向上层传递一些信息，保留一些上下文的状态。但如果你在执行层直接打印结果的话，那么这个类其实就不太用得到，忽略即可。

1. `drop index` 语句目前只给出了 `index name` 而没有给出 `table name`（设计上的问题），这里可以通过不考虑重复 `index name` 的问题或是对所有 `table` 的 `index` 进行搜索来解决。
2. `create index` 语句使用 `using` 时，`using` 关键字的语法树结点在打印时会出现 `error type`（缺少 `using` 关键字的语法树结点定义）。如果没有选做另外的索引类型，可以直接忽略这个问题。如果选做了另外的索引类型，那么可以暂且忽略 `using` 关键字的语法树结点的类型，在其子结点中获取索引类型。

## 10.6 评论

---

在minisql 文件夹执行 `find . -name syntax_tree*` 可以找到DOT文件

table\_heap\_test中显示unknown file怎么处理，就是说我没找到table\_heap\_test.db文件但test中要使用到它破案了，是我没开权限创建不了这个文件

请问，#4的Table与Index有没有将其数据“写文件”的接口呀（或者到缓冲区）



[YingChengJun2022-05-31 15:33](#)

Table和Index的MetaData通过序列化写入到BufferPool的Page中，Page会有BPM自行写入到文件。

SimpleMemHeap的malloc +placement new，但却只用free释放，会造成内存泄露  
可用valgrind运行下./bin/main  
另外，DestroySyntaxTree时还有一处内存没有free



[VatineDJC2022-06-04 19:30](#)

#4中的IndexMetadata调用Create的时候并没有传入和page\_id相关的参数，那么如何确定IndexMetadata数据在哪个页上呢