| 实验题目 | 查询执行器实现 | | | 实验日期 | 2022.5.1 |
|---|---|---|---|---|---|
| 班级 | 1903103 | 学号 | 1190200208 | 姓名 | 李旻翀 |

# CS33503 数据库系统实验

## 实验检查记录

| 实验结果的正确性(60%) | | 表达能力(10%) | |
|---|---|---|---|
| 实验过程的规范性(10%) | | 实验报告(20%) | |
| 加分(5%) | | 总成绩(100%) | |

## 实验报告

### 一、实验目的（介绍实验目的）

1. 掌握各种关系代数操作的实现算法，特别是连接操作的实现算法。
2. 在实验 2 完成的缓冲区管理器的基础上，使用 C++面向对象程序设计方法实现查询执行器。

### 二、实验环境（介绍实验使用的硬件设备、软件系统、开发工具等）

在 Win10 + VsCode 环境下编写代码，实际运行在 Ubuntu 18.04 LTS 下。

### 三、实验过程（介绍实验过程、设计方案、实现方法、实验结果等）

#### 1. 实验内容概述

本次实验中，我完成了基于块的嵌套循环连接算法的实现。并在 Linux 环境下实际运行测试用例验证了其正确性。

#### 2. 实验内容：基于块的嵌套循环连接算法的实现

##### 2.1 算法概述

基于块的嵌套循环连接算法在邹老师 PPT 中有具体阐述，其算法内容如下：

基于块的嵌套循环连接(Block-based Nested-Loop Join)

假设: $B(S) \leq B(R)$

算法

```
for 外关系S的每M − 1块 do
    将这M − 1块读入缓冲池
    用一个内存查找结构来组织这M − 1块中的元组
    for 内关系R的每一块P do
        将P读入缓冲池
        for P中每条元组r do
            for 内存查找结构中能与r进行连接的元组s do
                连接r和s，并将结果写入输出缓冲区
```

| 实验题目 | 查询执行器实现 | | | 实验日期 | 2022.5.1 |
|---|---|---|---|---|---|
| 班级 | 1903103 | 学号 | 1190200208 | 姓名 | 李旻翀 |

本次实验基于上述算法内容，实现了 NestedLoopJoinOperator 类中的 execute 方法。

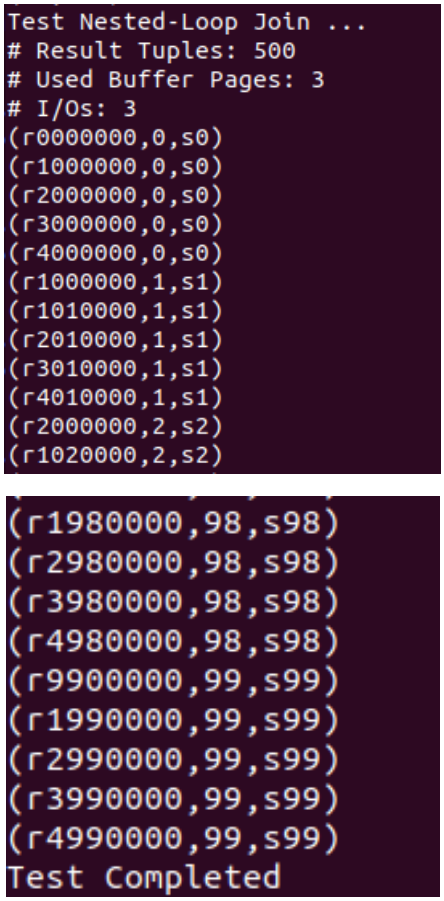
## 2.2 算法思想

由于算法代码较长，因此此处不作展示。算法的大致思想如下：

1. 首先通过循环将外关系 S 的 M-1 个块读入缓存池。通过结构 used_list 来保存读入缓存的块。更新使用的页面数 numUsedBufPages 与 IO 数 numIOs（各加一）。若 S 关系中的元组提前读完，则提前结束循环。
2. 每次读入并处理外关系 R 中的一个块 P，直到外关系 R 读完。在每次读进 R 的 1 个块的同时，更新使用的页面数 numUsedBufPages 与 IO 数 numIOs，并遍历缓存块，查找能与 R 中元组进行连接的 S 中的元组。如果能够连接，则通过 joinTuples 函数构建出连接后的 result_tuple。
3. 在循环完成后，通过 insertTuple 将结果写入文件，并更新 numResultTuples。

## 2.3 实验结果

在根目录下打开命令行，执行命令 make 完成编译，然后进入./src 文件夹执行./badgerdb_main 运行生成的程序，运行结果如下：




需要注意的是，每次运行后在./src 中会生成 3 个.tbl 文件，需要将其删除后才能再次运行。

| 实验题目 | 查询执行器实现 | | | 实验日期 | 2022.5.1 |
|---|---|---|---|---|---|
| 班级 | 1903103 | 学号 | 1190200208 | 姓名 | 李旻翀 |

## 四、实验结论（总结实验发现及结论）

通过此次实验，我实现了基于块的嵌套循环连接算法，在实验过程中，我有以下收获：
1.通过本次实验，我熟悉了基于块的嵌套循环连接算法，并通过 C++语言实现它。
2.加深了对查询执行器算法的了解。
3.锻炼了自己的编程能力以及阅读现有代码的能力。
4.了解了查询执行器的底层实现，对数据库整体运作原理的理解更为深刻。