哈尔滨工业大学(深圳)

《数据库》实验报告

实验四

查询处理算法的模拟实现

学　院: 计算机科学与技术

姓　名: 胡聪

学　号: 180110505

专　业: 计算机科学与技术

日　期: 2021-05-05

一、 实验目的

理解索引的作用，掌握关系选择、连接、集合的交、并、差等操作的实现算法，理解算法的 I/O 复杂性。

二、 实验环境

Windows 10 操作系统、Visual Studio Code

三、 实验内容

关系 R 具有两个属性 A 和 B，其中 A 和 B 的属性值均为 int 型（4 个字节），A 的值域为[20, 60]，B 的值域为[1000, 2000]。

关系 S 具有两个属性 C 和 D，其中 C 和 D 的属性值均为 int 型（4 个字节）。C 的值域为[40, 80]，D 的值域为[1000, 3000]。

① 实现基于线性搜索的关系选择算法：基于 ExtMem 程序库，使用 C 语言实现线性搜索算法，选出 S.C=50 的元组，记录 IO 读写次数，并将选择结果存放在磁盘上。（模拟实现 select S.C, S.D from S where S.C = 50）

② 实现两阶段多路归并排序算法（TPMMS）：利用内存缓冲区将关系 R 和 S 分别排序，并将排序后的结果存放在磁盘上。

③ 实现基于索引的关系选择算法：利用（2）中的排序结果为关系 S 建立索引文件，利用索引文件选出 S.C=50 的元组，并将选择结果存放在磁盘上。记录 IO 读写次数，与（1）中的结果对比。（模拟实现 select S.C, S.D from S where S.C = 50 ）

④ 实现基于排序的连接操作算法（Sort-Merge-Join）：对关系 S 和 R 计算 S.C 连接 R.A ，并统计连接次数，将连接结果存放在磁盘上。（模拟实现 select S.C, S.D, R.A, R.B from S inner join R on S.C = R.A）

⑤ 实现基于排序或散列的两趟扫描算法，实现并（S∪R） 、交（S∩R） 、

差（S-R）其中一种集合操作算法，将结果存放在磁盘上，并统计并、交、差操作后的元组个数。

四、 实验过程

(1) 实现基于线性搜索的关系选择算法

问题分析：从头到尾依次进行搜索即可找到符合题目要求的数据。

实验结果：

```
------------------------------
基于线性搜索的选择算法 S.C=50:
------------------------------
读入数据块17
读入数据块18
(X=50,Y=2225)
(X=50,Y=2741)
(X=50,Y=2766)
(X=50,Y=2537)
(X=50,Y=2883)
(X=50,Y=1885)
读入数据块19
读入数据块20
读入数据块21
读入数据块22
读入数据块23
读入数据块24
读入数据块25
(X=50,Y=1503)
读入数据块26
读入数据块27
读入数据块28
读入数据块29
读入数据块30
读入数据块31
读入数据块32
读入数据块33
读入数据块34
读入数据块35
读入数据块36
读入数据块37
读入数据块38
读入数据块39
(X=50,Y=1016)
(X=50,Y=2913)
(X=50,Y=1770)
读入数据块40
```

```
(X=50,Y=1647)
(X=50,Y=2588)
读入数据块41
读入数据块42
读入数据块43
读入数据块44
读入数据块45
读入数据块46
读入数据块47
读入数据块48

满足选择条件的元组一共12个

IO读写一共 34次
```

(2) 实现两阶段多路归并排序算法（TPMMS）

问题分析：两阶段多路归并排序算法常用来解决数据量较大的问题，特别是数据大小超过了数据库的可用主存大小，无法一次将数据载入主存的情况。该算法包括两个阶段。

阶段 1：将数据片段进行排序，使得每条记录都是一个有序表的一部分，该表即为排序子表，这样的表产生了多个；

阶段 2：归并排序所有的排序子表，最后形成一个排序表。

该算法只让磁盘中的数据写入主存两次，完成了数据的排序，节省了 IO 时间。

实验结果：

```
关系R排序

输出到文件301.blk
输出到文件302.blk
输出到文件303.blk
输出到文件304.blk
输出到文件305.blk
输出到文件306.blk
输出到文件307.blk
输出到文件308.blk
输出到文件309.blk
输出到文件310.blk
输出到文件311.blk
输出到文件312.blk
输出到文件313.blk
输出到文件314.blk
输出到文件315.blk
输出到文件316.blk
```

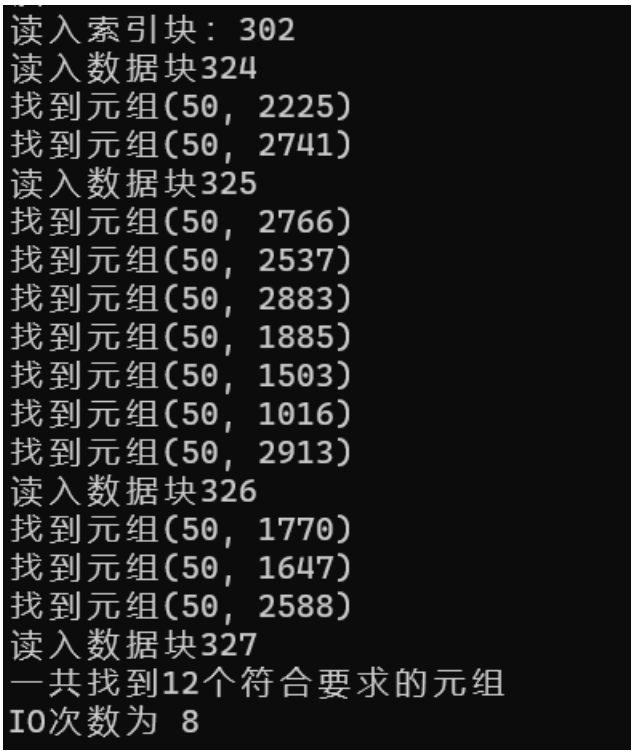
```
关系S排序

输出到文件317.blk
输出到文件318.blk
输出到文件319.blk
输出到文件320.blk
输出到文件321.blk
输出到文件322.blk
输出到文件323.blk
输出到文件324.blk
输出到文件325.blk
输出到文件326.blk
输出到文件327.blk
输出到文件328.blk
输出到文件329.blk
输出到文件330.blk
输出到文件331.blk
输出到文件332.blk
输出到文件333.blk
输出到文件334.blk
输出到文件335.blk
输出到文件336.blk
输出到文件337.blk
输出到文件338.blk
输出到文件339.blk
输出到文件340.blk
输出到文件341.blk
输出到文件342.blk
输出到文件343.blk
输出到文件344.blk
输出到文件345.blk
输出到文件346.blk
输出到文件347.blk
输出到文件348.blk
```

(3) 实现基于索引的关系选择算法

问题分析：在做选择操作时，如果选择条件中有属性没有被索引，那么需要将关系表中的元组加载到内存中以判断是否满足条件，如果满足则输出，如果表中查询涉及到的属性有索引，则在执行选择操作的时候，首先将索引文件加载到内存中，然后通过索引找到元组，判断是否满足条件并且输出。

实验结果：

```
------------------------------
基于索引的关系选择算法 S.C=50:
------------------------------
读入数据块317
读入数据块318
读入数据块319
读入数据块320
读入数据块321
读入数据块322
读入数据块323
读入数据块324
读入数据块325
读入数据块326
读入数据块327
读入数据块328
读入数据块329
读入数据块330
读入数据块331
读入数据块332
读入数据块333
读入数据块334
读入数据块335
读入数据块336
读入数据块337
读入数据块338
读入数据块339
读入数据块340
读入数据块341
读入数据块342
读入数据块343
读入数据块344
读入数据块345
读入数据块346
读入数据块347
读入数据块348
读入索引块: 301
```



(4) 实现基于排序的连接操作算法（Sort-Merge-Join）

问题分析：第一趟：划分 R 和 S 的子表并进行子表排序，排序均基于连接属性排序。

第二趟：归并时注意是 R 的输入还是 S 的输入。R 和 S 的两路输入之间进行连接检查并连接后输出。

实验结果：

```
----------基于排序的连接操作算法----------

注：结果写入磁盘1001
注：结果写入磁盘1002
注：结果写入磁盘1003
注：结果写入磁盘1004
注：结果写入磁盘1005
注：结果写入磁盘1006
注：结果写入磁盘1007
注：结果写入磁盘1008
注：结果写入磁盘1009
注：结果写入磁盘1010
注：结果写入磁盘1011
注：结果写入磁盘1012
注：结果写入磁盘1013
注：结果写入磁盘1014
注：结果写入磁盘1015
注：结果写入磁盘1016
注：结果写入磁盘1017
注：结果写入磁盘1018
注：结果写入磁盘1019
注：结果写入磁盘1020
注：结果写入磁盘1021
注：结果写入磁盘1022
注：结果写入磁盘1023
注：结果写入磁盘1024
注：结果写入磁盘1025
注：结果写入磁盘1026
注：结果写入磁盘1027
注：结果写入磁盘1028
注：结果写入磁盘1029
注：结果写入磁盘1030
```

```
注：结果写入磁盘1031
注：结果写入磁盘1032
注：结果写入磁盘1033
注：结果写入磁盘1034
注：结果写入磁盘1035
注：结果写入磁盘1036
注：结果写入磁盘1037
注：结果写入磁盘1038
注：结果写入磁盘1039
注：结果写入磁盘1040
注：结果写入磁盘1041
注：结果写入磁盘1042
注：结果写入磁盘1043
注：结果写入磁盘1044
注：结果写入磁盘1045
注：结果写入磁盘1046
注：结果写入磁盘1047
注：结果写入磁盘1048
注：结果写入磁盘1049
注：结果写入磁盘1050
注：结果写入磁盘1051
注：结果写入磁盘1052
注：结果写入磁盘1053
注：结果写入磁盘1054
注：结果写入磁盘1055
注：结果写入磁盘1056
注：结果写入磁盘1057
注：结果写入磁盘1058
注：结果写入磁盘1059
注：结果写入磁盘1060
注：结果写入磁盘1061
注：结果写入磁盘1062
注：结果写入磁盘1063
注：结果写入磁盘1064
注：结果写入磁盘1065
注：结果写入磁盘1066
注：结果写入磁盘1067
注：结果写入磁盘1068
注：结果写入磁盘1069
```

```
注：结果写入磁盘1070
注：结果写入磁盘1071
注：结果写入磁盘1072
注：结果写入磁盘1073
注：结果写入磁盘1074
注：结果写入磁盘1075
注：结果写入磁盘1076
注：结果写入磁盘1077
注：结果写入磁盘1078
注：结果写入磁盘1079
注：结果写入磁盘1080
注：结果写入磁盘1081
注：结果写入磁盘1082
注：结果写入磁盘1083
注：结果写入磁盘1084
注：结果写入磁盘1085
注：结果写入磁盘1086
注：结果写入磁盘1087
注：结果写入磁盘1088
注：结果写入磁盘1089
注：结果写入磁盘1090
注：结果写入磁盘1091
注：结果写入磁盘1092
注：结果写入磁盘1093
注：结果写入磁盘1094
注：结果写入磁盘1095
注：结果写入磁盘1096
总共连接336次
```

(5) 实现基于散列的两趟扫描算法，实现交、并、差其中一种集合操作算法

问题分析：基于散列的两趟扫描算法，第一趟划分子集，并使子集具有某种特性，如有序或相同散列值等，将原始关系通过 h_p 散列成；第二趟处理全局性内容的操作，形成结果关系。如多子集间的归并操作，相同散列值子集的操作等。

实验结果：

```
----------基于排序的两趟扫描算法（差操作）----------

(40, 2385)
(40, 1705)
(40, 1339)
(41, 2909)
(41, 2427)
(41, 2609)
(41, 1269)
注：结果写入磁盘2001
(42, 2152)
(43, 2908)
(43, 1236)
(43, 2223)
(43, 2333)
(43, 2880)
(43, 2857)
注：结果写入磁盘2002
(44, 1672)
(44, 1606)
(44, 1003)
(45, 1288)
(45, 2861)
(45, 2038)
(45, 2416)
注：结果写入磁盘2003
(46, 2285)
(46, 1664)
(46, 1691)
(46, 2642)
(46, 1998)
(46, 2410)
(46, 2564)
注：结果写入磁盘2004
(46, 2232)
(47, 1111)
(47, 1053)
(47, 2254)
(47, 2410)
```

```
注：结果写入磁盘2005
(48, 2553)
(48, 1810)
(48, 1506)
(48, 1509)
(48, 2132)
(49, 2531)
(49, 1605)
注：结果写入磁盘2006
(49, 1100)
(49, 2724)
(49, 1084)
(50, 2225)
(50, 2741)
(50, 2766)
(50, 2537)
注：结果写入磁盘2007
(50, 2883)
(50, 1885)
(50, 1503)
(50, 1016)
(50, 2913)
(50, 1770)
(50, 1647)
注：结果写入磁盘2008
(50, 2588)
(51, 2486)
(51, 2252)
(51, 1501)
(51, 2214)
(52, 1876)
(52, 2607)
注：结果写入磁盘2009
(52, 1134)
(52, 1242)
(52, 1783)
(52, 1839)
(53, 2809)
(53, 1012)
(53, 1476)
```

```
注：结果写入磁盘2010
(53, 1635)
(53, 2140)
(53, 1394)
(54, 2366)
(54, 2053)
(54, 1204)
(54, 1930)
注：结果写入磁盘2011
(54, 1906)
(55, 1133)
(55, 2512)
(55, 1536)
(55, 2886)
(56, 1023)
(56, 1544)
注：结果写入磁盘2012
(56, 1713)
(56, 1796)
(56, 1949)
(56, 2486)
(56, 1590)
(57, 1312)
(57, 1583)
注：结果写入磁盘2013
(57, 2735)
(58, 1303)
(59, 2527)
(59, 2154)
(59, 1680)
(59, 1973)
(59, 1518)
```

```
注：结果写入磁盘2014
(59, 2195)
(59, 1990)
(60, 2524)
(60, 1254)
(60, 2645)
(60, 1743)
(60, 2975)
注：结果写入磁盘2015
(60, 1447)
(60, 1398)
(60, 1083)
(60, 2196)
(60, 1920)
(60, 1428)
(60, 2716)
注：结果写入磁盘2016
(61, 1957)
(61, 2181)
(61, 1630)
(61, 2184)
(61, 1853)
(61, 2556)
(61, 2958)
注：结果写入磁盘2017
(61, 1366)
(62, 2906)
(62, 1775)
(62, 1668)
(62, 2676)
(62, 2574)
(62, 1564)
注：结果写入磁盘2018
(62, 1276)
(62, 1617)
(63, 2495)
(63, 2713)
(63, 1519)
(63, 1983)
(63, 1098)
```

```
注：结果写入磁盘2019
(64, 1758)
(64, 2552)
(64, 2376)
(64, 1106)
(64, 2478)
(64, 1779)
(64, 2053)
注：结果写入磁盘2020
(64, 1171)
(65, 1363)
(65, 2469)
(65, 1427)
(65, 2014)
(65, 1278)
(65, 2681)
注：结果写入磁盘2021
(65, 2535)
(66, 2712)
(66, 1057)
(66, 1406)
(66, 2808)
(67, 1529)
(67, 1243)
注：结果写入磁盘2022
(67, 2818)
(67, 1035)
(67, 1765)
(68, 1412)
(68, 1962)
(68, 2100)
(68, 1218)
```

```
注：结果写入磁盘2023
(69, 2441)
(69, 2917)
(69, 2721)
(69, 2596)
(70, 2216)
(70, 2042)
(70, 1852)
注：结果写入磁盘2024
(70, 1962)
(71, 2573)
(71, 2053)
(71, 1856)
(71, 2149)
(71, 1509)
(72, 1081)
注：结果写入磁盘2025
(72, 1015)
(72, 1082)
(72, 1325)
(72, 1102)
(73, 1283)
(73, 2814)
(74, 2340)
注：结果写入磁盘2026
(74, 1204)
(74, 1203)
(74, 1819)
(74, 1584)
(74, 1127)
(75, 2961)
(75, 1680)
注：结果写入磁盘2027
(76, 1384)
(76, 1111)
(76, 1811)
(76, 2369)
(77, 1626)
(77, 1218)
(77, 1775)
```

```
注：结果写入磁盘2028
(77, 2979)
(77, 2680)
(78, 1113)
(78, 2666)
(78, 1308)
(78, 1833)
(78, 1496)
注：结果写入磁盘2029
(79, 1612)
(79, 2975)
(79, 2576)
(79, 1271)
(79, 2542)
(79, 1725)
注：结果写入磁盘2030
S和R的差集(S-R)有209个
```

五、 附加题

六、 总结

本次实验，与之前的实验不同的是，涉及到了较为底层的数据库算法，且题目中要求有限的缓冲区限制，给实验过程增加了一些挑战。实验的一开始，首先需要了解 ExtMem 库的使用方法，仔细阅读指导书，了解库提供的一些接口，才能开始上手本实验。

实验过程中，也遇到了很多问题，例如 CMake 的使用再次出现了问题，调试过程中发现了段错误等问题却又很难定位到问题所在，好在后来通过查找最终解决了这些问题。虽然本次实验中很多东西以后在工作中用不上，但重要的是解决底层问题的思想，这些是很宝贵的经验。