中国人民大学信息学院

数据库系统原理与实现

实验三《SPJ 算法》实验报告

Github: https://github.com/taoyouxian/tinydb（可查看 Wiki 相关介绍与 FAQ）

小组成员：张恒、陶友贤、赵旺、修晔良

目录

1 实验内容.....2
1.1 选择操作算法.....2
1.2 投影操作算法.....2
1.3 链接操作算法.....2
2 实验要求.....2
2.1 选择操作.....2
2.2 连接操作.....2
3 实验分工安排.....2
4 实验环境.....3
5 实验原理与程序设计.....3
5.1 存储管理模块的改进与完善.....3
5.2 查询执行.....6
5.3 选择操作：表扫描.....6
5.4 投影操作.....7
5.5 连接操作.....7
6 实验结果.....9
6.1 初始化数据库.....9
6.2 读入 employee 表.....9
6.3 读入 department 表.....10
6.4 基于表扫描的等值选择.....11
6.5 基于表扫描的范围选择.....12
6.6 投影.....12
6.7 嵌套循环连接.....13
6.8 基于排序的等值连接.....14
6.9 基于散列的等值连接.....15
6.10 三表连接.....15
6.11 不同数据类型的多条件等值连接.....16
6.12 不同数据类型的多条件非等值连接.....16
6.13 执行完之后数据库系统参数.....17
6.14 关闭数据库.....17
6.15 其他功能.....17
7 使用手册.....18
8 实验总结.....18

1 实验内容

1.1 选择操作算法

a) Table scan(表扫描)
b) Index scan(索引扫描)

1.2 投影操作算法

1.3 链接操作算法

a）嵌套循环连接
b） 基于排序的连接
c）基于散列的连接
d） 基于索引的连接

2 实验要求

2.1 选择操作

a） 无条件选择
b） 等值条件
c） 非等值条件
d） 范围条件

2.2 连接操作

a) 卡氏积（排序连接和散列连接例外）
b) 等值条件
c) 非等值条(排序连接和散列连接例外)

3 实验分工安排

任务名称	人员
系统设计	张恒、陶友贤、赵旺、修晔良
程序实现	陶友贤、张恒

系统测试	陶友贤、张恒
文档编写	张恒
构图设计	赵旺、修晔良
文档修改	张恒、陶友贤

4 实验环境

- 操作系统平台：图形界面程序 windows 10/win 8/win 7
- 编程语言：C++、C
- 程序开发环境：Visual Studio 2013 + QT（Version：5.7.2）

5 实验原理与程序设计

5.1 存储管理模块的改进与完善

补充了实验一未实现的映射表和关系数据字典，结合本次实验的要求，在实验一的基础上，对数据库的一些结构体进行了修 改与完善，并增添了 Relation 和 Attribute 两个类。但整体采用的仍是“记录→页 →文件”这样的组织结构层次来对数据进行组织、存储等操作。下面将主要的数 据结构罗列于此，并加以说明。其中 Relation 是描述数据字典的类，对每一张关系表，都对应一个 Relation 类的实例

```
Class Relation{
Public:
      int fileID;                              //该关系模式对应的文件号
       char relationName[NAME_MAX_LENGTH];    //关系名
       int attributeNum;                       //属性个数
       Attribute atb[ATTRIBUTE_NUM];           //属性表
       bool isIndexed[ATTRIBUTE_NUM]; //是否在对应属性上建立了索引
       bool isOrdered[ATTRIBUTE_NUM];  //是否在对应属性上有序
       int recordLength;
       int recordNum;
```

```
}
```

```
class Attribute{
private:
      char attributeName[NAME_MAX_LENGTH];//属性名
      int type;                           //属性类型
      int length;                         //属性长度
      int recordDeviation;                 //记录内偏移（在关系属性表中的偏移）
}
```

以上是新增的两个类，顾名思义，分别是关系与属性，在构建关系数据字典的时候有很大的应用。

在实验一中，用户是直接对文件进行操作的。增加了“Relation”之后，用户的 操作就保留在关系表层面上，简化了一些用户操作，使系统的底层部分对用户透 明，更符合常见的数据库管理系统，也便于使用。 因此，在下图中可以看出，数据库管理系统的头部结构增加了一个 Relation 类型的数组 data_dict，即关系数据字典，这是为了保存该数据库中所包含的所有 关系表的相关信息。DBMS 会为此而维护一个表模式文件，在关闭数据库时将数 据字典写回到磁盘中，而在初始化数据库时，则会从该文件中读取已存储的关系 模式信息，以保证后续的操作。

另外，基于上一次的设想（即不去管每个元组是属于哪个表的，在向该数据库中插入一条元组时，会给它一个唯一的逻辑号，作为其在数据库中的逻辑地址）， 又增加了一个位示图，原理和作用都类似空闲空间位示图，用于标识数据库中逻 辑号是否可用。

```
struct dbSysHead{
   struct SysDesc desc; //数据库系统的相关参数
   unsigned long *FreeSpace_bitMap;//空闲空间位示图
   unsigned long *FreeLogicID_bitMap;//空闲逻辑号位示图
   struct buffSpace buff;//缓冲区
   Relation data_dict[MAX_FILE_NUM];//关系数据字典，先存在这里，最后关闭数据库时要刷回到文件中
   FILE *fpdesc;//文件指针
};
```

为了使逻辑号能够起到“准确定位一个元组”的作用，数据库管理系统需要维 护一个“映射表”文件，与用户文件相同的是，该文件也是以页为单位（因为页/块 是数据库进行 I/O 操作的最小单位），每页中会有页头来记录当前页的相关信息。 不同的是，不需要记录每条记录的偏移量，因为该文件中的每条记录即是映射表 表项，其结构与大小是固定的，如下所示：

```
struct dbMapTable {
   bool isdeleted;//该逻辑号指向的元组是否已删除
   long logicID;//逻辑号
   long pageNo;//元组所在的页号
```

```
    long recordID;//页内记录号（用于计算偏移量）
};
```

映射表			
是否删除	DB 逻辑地址（逻辑号）	DB 物理地址	
		页号	页内记录号
1	1	0	0
0	2	0	1
1	3	0	2
1	4	1	0
0	5	1	1
0	6	1	2
0	7	2	0
…	…	…	…

这样，给定一个逻辑号，就可以通过查询该文件来获得该元组所在的页号和页内记录号，从而准确定位一个元组，获取元组内容。

本实验设计的关系表包括 employee 表，department 表，birthday 表，其中 employee 和 department 通过 did 相互连接。Deparment 和 birthday 表通过 idcard 相连接。

```
struct employee{
    int rid;    //职工号
    int did;    //所在部门的 id
    int age;
    char rname[20];
};

struct department{
    int did;            //部门 id
    int manager_id;     //部门经理职工号
    int num;            //部门总人数
    char dname[20];     //部门名字
    char idcard[18];    //部门经理身份证
};

struct birthday{
    char idcard[18];    //部门经理身份证
    char birthtime[15];    //部门经理出生时间
    int addressid;    //部门经理出生地址，表示出生地址数组中的下标
```

```
};
```

5.2 查询执行

查询处理器是 DBMS 中的一个部件集合，它将用户的查询、修改等命令转化为数据库上的操作序列,并执行这些操作， 供了查询将被如何执行的大量细节。查询执行作为操作数据库数据的算法，是查询处理器的主要部分之一，实现了关系代数操作的基本方法,在 DBMS 的层次结构中应处于数据存取层,即把上层 的集合操作转为单记录操作。处理对象是单个元组。

5.3 选择操作：表扫描

和表扫描相关的函数有

```
int tableScanEqualSelector(struct dbSysHead *head, int dictID, char* attribute_name, char* value);//等值选择
int tableScanRangeSelector(struct dbSysHead *head, int dictID, char* attribute_name, char* min, char* max);//范围选择
```

扫描就是将关系的元组从磁盘读入到内存中的操作，是其他复杂操作的基础与关键。数据库的磁盘中存放了关系 employee 的所有元组，分布在多个数据块中，而数据库管理系统知道哪些块中包含了 R 的元组。以块为单位，将包含拉关 系 R 的元组的数据块依次读入内存中，这种操作称为表扫描。 选择操作是一元操作符，其特点是不需要一次在内存中装入整个关系，而是 一次读一个块即可。 描述基于表扫描的选择操作可用 SQL 语句“select * from employee where age = 20”为例，其中“where age = 20”即为选择的条件，而“from employee”意味 着需要对表 employee 进行扫描，这就是一个基于表扫描的等值选择，将 where 后面的等值条件改为范围条件，即变成了范围选择。 该功能的伪代码大致如下：

```
创建临时关系表 PageNo←表
    employee 占用的第一页
```

```
do
    读取该页页头
    for i←0 to 该页包含的记录个数
        读取对应的一条元组
        解析该元组在指定属性上的值，比较是否满足给定条件
        if 满足
            将该元组插入到临时表中
        repeat
        PageNo←当前页的下一页
    until 当前页没有后继页了
```

5.4 投影操作

投影操作在查询执行的过程中多是处于最后一步，与选择操作一样，都是一元操作符，同样是按块操作即可。在处理每一个元组时，需将整个元组进行切分后，获得用户查询语句指定的属性对应的值，再将结果投影到临时表中。

5.5 连接操作

a) 嵌套循环链接

```
int nestedLoopJoin(struct dbSysHead *head, int employee_dictID, int department_dictID);
```

因为在我们的数据库中，元组是以块聚集的，因而选择了基于块的嵌套 循环连接，以减少 I/O 操作，提高效率。其实质就是嵌套着遍历两个关系表的所有块中的每个元组，找出在公共属性上满足指定要求的，就将两个元组 连接起来，再输出到临时表中。 需要注意的是，通常会将较小的表作为外层循环（仅进行一次读取），把 尽可能多的缓冲区块分给这个表，只留一个缓冲区块用于读取较大表的页。 这样可以减少外层循环的迭代次数，也即减少了内层表的每一块被重复读入 到缓冲区中的次数。以对表 employee 和 department 进行连接为例（这两个表的公共属性为 did，employee 是较大的表，department 是较小的表），将该过程以伪代码形 式在下面展出：

```
创建临时关系表
遍历 department 的每一页
    遍历该页中的每一条记录，当读取到一条 record_dept 时
        遍历 employee 的每一页
            遍历该页中的每一条记录，当读取到一条 record_emp 时
                比较 record_dept 与 record_emp，在公共属性 did 上的值是否相同
                if 相同
                    将这两个元组拼接起来，插入到临时表中
```

虽然直观上来看，该方法有四层循环，但实际上，将一页读入到缓冲区 之后，再进行的操作就都是内存操作了，因而不会造成大量的 IO 读写，是一 种比较简单的连接方法。

b)基于排序的等值连接

```
int SortJoin(struct dbSysHead *head, int employee_dictID, int department_dictID);
```

在嵌套循环连接中，如果两个表是有序的，当判断两个元组在公共属性 上是否有相同值的时候，就会减少很多不必要的比较，对于外层表的每一条 记录，都能尽早结束内层循环，相比于简单的嵌套循环连接，其效率有了跟 大的 高。该方法的处理过程与上述方法类似，只需要在进入循环之前，先对两个 表进行排序，进而对两个有序的表进行连接操作。另外，在内层循环进行比 较时，如果发现 record_emp > record_dept，那么直接 break 即可，因为表已 经是有序的了，内层表往后的记录都只能比 record_dept 更大，而不会出现 相同值了。

c)基于散列的等值连接

```
int HashJoin(struct dbSysHead *head, int employee_dictID, int department_dictID);
```

目的与基于排序的方法类似，都是为了减少不必要的比较次数。选择合 适的散列函数，应用到公共属性上，先对表中的元组进行“分组”——将哈希 值相同的元组放到同一个桶里。该过程的伪代码如下所示：

```
选择一个散列函数
对表 employee 和 department 分别进行哈希，各生成 n 个桶
创建一个临时表
for i←1 to n
    将 employee 的第 i 个桶和 department 的第 i 个桶中的所有元组进行等值连接，并将 连接后的结果插入到临时表中
repeat
```

6 实验结果

6.1 初始化数据库

从文件中初始化数据库，下图是一些系统参数。

```
**************************************************
**************************************************
**********                                      **********
**********   Welcome to use [TinyDB]            **********
**********                                      **********
**************************************************
**************************************************

Init database...
database not exist, start to create database...
Init database successfully.

******************** Database related parameters ****************
Total database size: 205520896
The size of each page: 4096
The total number of database pages: 50176
Currently available pages: 50176
The starting address of free space: 820
The size of bitmap size in free space: 6272
The start address of the diagram in idle logic bit: 7092
The size of the bitmap in idle logic number: 128000
The starting address of the data area: 135092
The file number of the mapping table file: -1
File number of the data dictionary file: -1
The total number of files in the current database: 0
The total number of records in the current database: 0

**************************************************
```

6.2 读入 employee 表

从我们自己生成的数据文件读入 employee 表，此处也支持 Create SQL 解析。

```
create table employee(rid INT(11), did INT, rname VARCHAR(20), age INT, PRIMARY KEY(rid))
```

```
File Type: User file
Tablename: employee
Attributes number: 4
Record length: 39
Record number: 200
The file occupies: 2 page
The starting page number of the file is: 2

**************************************************
Tablename: employee
Attributes number: 4
Record length: 39
Record number: 200
-------- Page Head Info -------
PageNo  2
PageFreePalce   26
RecordNum Stored in CurPage     142
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
0       0       11      0       0|1|47|abc0
1       1       22      0       1|4|40|abc0
2       2       33      0       2|9|24|abc0
3       3       44      0       3|8|58|abc0
4       4       55      0       4|2|44|abc0
5       5       66      0       5|5|45|abc0
6       6       77      0       6|1|47|abc0
7       7       88      0       7|1|31|abc0
```

```
-------- Page Head Info -------
PageNo  3
PageFreePalce   2336
RecordNum Stored in CurPage     58
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
142     0       14      0       142|5|43|abc14
143     1       28      0       143|2|45|abc14
144     2       42      0       144|2|25|abc14
145     3       56      0       145|8|46|abc14
146     4       70      0       146|7|57|abc14
147     5       84      0       147|2|32|abc14
148     6       98      0       148|9|54|abc14
149     7       112     0       149|1|49|abc14
150     8       126     0       150|6|29|abc15
151     9       140     0       151|8|32|abc15
152     10      154     0       152|5|55|abc15
153     11      168     0       153|4|49|abc15
154     12      182     0       154|1|52|abc15
155     13      196     0       155|5|40|abc15
156     14      210     0       156|8|53|abc15
157     15      224     0       157|9|43|abc15
158     16      238     0       158|9|56|abc15
159     17      252     0       159|7|29|abc15
```

6.3 读入 department 表

从我们自己生成的数据文件读入 department 表

```
Tablename: department
Attributes number: 5
Record length: 50
Record number: 10
-------- Page Head Info -------
PageNo  7
PageFreePalce   3574
RecordNum Stored in CurPage     10
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
200     0       34      0       0|0|18|manager0|340521199501033310
201     1       68      0       1|2|15|manager1|340521199501033311
202     2       101     0       2|4|8|manager2|340521199501033312
203     3       135     0       3|6|13|manager3|340521199501033313
204     4       169     0       4|8|17|manager4|340521199501033314
205     5       204     0       5|10|10|manager5|340521199501033315
206     6       238     0       6|12|7|manager6|340521199501033316
207     7       272     0       7|14|5|manager7|340521199501033317
208     8       307     0       8|16|18|manager8|340521199501033318
209     9       342     0       9|18|12|manager9|340521199501033319
```

6.4 基于表扫描的等值选择

我们以如下 SQL 语句为例：

```
select * from employee
where did = 6
```

```
This a temporary tables.
-------- Page Head Info -------
PageNo  4
PageFreePalce   3526
RecordNum Stored in CurPage     19
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       12      0       30|6|55|abc3
-1      1       24      0       35|6|25|abc3
-1      2       36      0       38|6|41|abc3
-1      3       48      0       42|6|20|abc4
-1      4       60      0       43|6|36|abc4
-1      5       72      0       46|6|23|abc4
-1      6       84      0       60|6|30|abc6
-1      7       96      0       67|6|50|abc6
-1      8       108     0       76|6|50|abc7
-1      9       120     0       87|6|41|abc8
-1      10      134     0       124|6|51|abc12
-1      11      148     0       150|6|29|abc15
-1      12      162     0       161|6|49|abc16
-1      13      176     0       164|6|45|abc16
-1      14      190     0       168|6|51|abc16
-1      15      204     0       169|6|43|abc16
-1      16      218     0       176|6|42|abc17
-1      17      232     0       178|6|42|abc17
-1      18      246     0       187|6|53|abc18
```

6.5 基于表扫描的范围选择

我们以如下 SQL 语句为例：

```
select * from employee
where age > 30 and age < 40
```

```
This a temporary tables.
-------- Page Head Info -------
PageNo  5
PageFreePalce   3768
RecordNum Stored in CurPage     11
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       12      0       30|6|55|abc3
-1      1       24      0       31|0|22|abc3
-1      2       36      0       32|8|46|abc3
-1      3       48      0       33|0|42|abc3
-1      4       60      0       34|4|28|abc3
-1      5       72      0       35|6|25|abc3
-1      6       84      0       36|0|29|abc3
-1      7       96      0       37|0|50|abc3
-1      8       108     0       38|6|41|abc3
-1      9       120     0       39|3|48|abc3
-1      10      132     0       40|9|43|abc4
```

6.6 投影

我们以如下 SQL 语句为例：

```
SELECT rid,did from employee
```

```
-------- Page Head Info -------
PageNo  6
PageFreePalce   76
RecordNum Stored in CurPage     200
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       4       0       47|1
-1      1       8       0       40|4
-1      2       12      0       24|9
-1      3       16      0       58|8
-1      4       20      0       44|2
-1      5       24      0       45|5
-1      6       28      0       47|1
-1      7       32      0       31|1
-1      8       36      0       42|5
-1      9       40      0       56|7
-1      10      44      0       24|1
-1      11      48      0       53|2
-1      12      52      0       42|2
-1      13      56      0       56|1
-1      14      60      0       35|8
-1      15      64      0       26|7
-1      16      68      0       38|1
-1      17      72      0       52|9
-1      18      76      0       39|7
-1      19      80      0       34|5
-1      20      84      0       31|3
-1      21      88      0       33|2
-1      22      92      0       44|3
-1      23      96      0       51|1
```

6.7 嵌套循环连接

我们以如下 SQL 语句为例（接下来的三个连接查询都是此 SQL）：

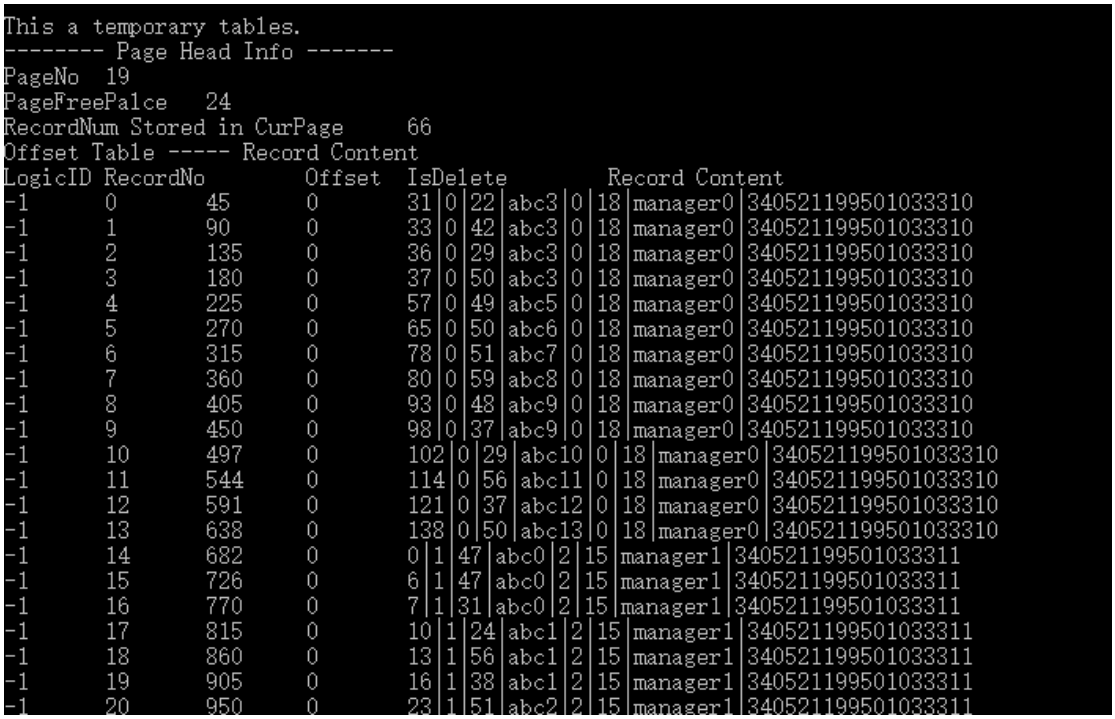
```
select * from employee, department
where employee.did = department.did
```

```
-------- Page Head Info -------
PageNo  8
PageFreePalce   50
RecordNum Stored in CurPage     66
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       44      0       0|1|47|abc0|2|15|manager1|340521199501033311
-1      1       88      0       1|4|40|abc0|8|17|manager4|340521199501033314
-1      2       133     0       2|9|24|abc0|18|12|manager9|340521199501033319
-1      3       178     0       3|8|58|abc0|16|18|manager8|340521199501033318
-1      4       221     0       4|2|44|abc0|4|8|manager2|340521199501033312
-1      5       266     0       5|5|45|abc0|10|10|manager5|340521199501033315
-1      6       310     0       6|1|47|abc0|2|15|manager1|340521199501033311
-1      7       354     0       7|1|31|abc0|2|15|manager1|340521199501033311
-1      8       399     0       8|5|42|abc0|10|10|manager5|340521199501033315
-1      9       443     0       9|7|56|abc0|14|5|manager7|340521199501033317
-1      10      488     0       10|1|24|abc1|2|15|manager1|340521199501033311
-1      11      532     0       11|2|53|abc1|4|8|manager2|340521199501033312
-1      12      576     0       12|2|42|abc1|4|8|manager2|340521199501033312
-1      13      621     0       13|1|56|abc1|2|15|manager1|340521199501033311
-1      14      667     0       14|8|35|abc1|16|18|manager8|340521199501033318
-1      15      712     0       15|7|26|abc1|14|5|manager7|340521199501033317
-1      16      757     0       16|1|38|abc1|2|15|manager1|340521199501033311
-1      17      803     0       17|9|52|abc1|18|12|manager9|340521199501033319
-1      18      848     0       18|7|39|abc1|14|5|manager7|340521199501033317
-1      19      894     0       19|5|34|abc1|10|10|manager5|340521199501033315
-1      20      939     0       20|3|31|abc2|6|13|manager3|340521199501033313
-1      21      983     0       21|2|33|abc2|4|8|manager2|340521199501033312
-1      22      1028    0       22|3|44|abc2|6|13|manager3|340521199501033313
```

6.8 基于排序的等值连接

```
-------- Page Head Info -------
PageNo  12
PageFreePalce   24
RecordNum Stored in CurPage     66
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       45      0       31|0|22|abc3|0|18|manager0|340521199501033310
-1      1       90      0       33|0|42|abc3|0|18|manager0|340521199501033310
-1      2       135     0       36|0|29|abc3|0|18|manager0|340521199501033310
-1      3       180     0       37|0|50|abc3|0|18|manager0|340521199501033310
-1      4       225     0       57|0|49|abc5|0|18|manager0|340521199501033310
-1      5       270     0       65|0|50|abc6|0|18|manager0|340521199501033310
-1      6       315     0       78|0|51|abc7|0|18|manager0|340521199501033310
-1      7       360     0       80|0|59|abc8|0|18|manager0|340521199501033310
-1      8       405     0       93|0|48|abc9|0|18|manager0|340521199501033310
-1      9       450     0       98|0|37|abc9|0|18|manager0|340521199501033310
-1      10      497     0       102|0|29|abc10|0|18|manager0|340521199501033310
-1      11      544     0       114|0|56|abc11|0|18|manager0|340521199501033310
-1      12      591     0       121|0|37|abc12|0|18|manager0|340521199501033310
-1      13      638     0       138|0|50|abc13|0|18|manager0|340521199501033310
-1      14      682     0       0|1|47|abc0|2|15|manager1|340521199501033311
-1      15      726     0       6|1|47|abc0|2|15|manager1|340521199501033311
-1      16      770     0       7|1|31|abc0|2|15|manager1|340521199501033311
-1      17      815     0       10|1|24|abc1|2|15|manager1|340521199501033311
-1      18      860     0       13|1|56|abc1|2|15|manager1|340521199501033311
-1      19      905     0       16|1|38|abc1|2|15|manager1|340521199501033311
-1      20      950     0       23|1|51|abc2|2|15|manager1|340521199501033311
-1      21      995     0       44|1|48|abc4|2|15|manager1|340521199501033311
-1      22      1040    0       55|1|45|abc5|2|15|manager1|340521199501033311
-1      23      1085    0       61|1|56|abc6|2|15|manager1|340521199501033311
```

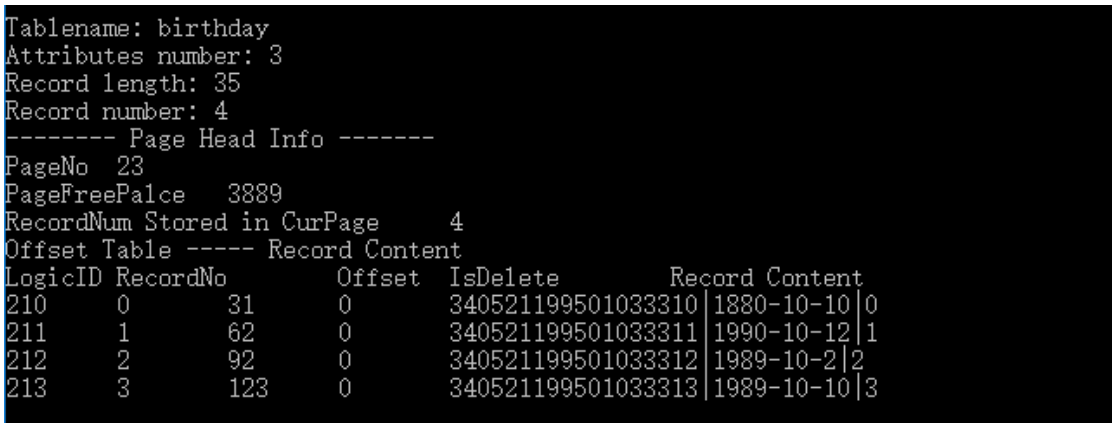
6.9 基于散列的等值连接



```
This a temporary tables.
-------- Page Head Info -------
PageNo  19
PageFreePalce   24
RecordNum Stored in CurPage     66
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       45      0       31|0|22|abc3|0|18|manager0|340521199501033310
-1      1       90      0       33|0|42|abc3|0|18|manager0|340521199501033310
-1      2       135     0       36|0|29|abc3|0|18|manager0|340521199501033310
-1      3       180     0       37|0|50|abc3|0|18|manager0|340521199501033310
-1      4       225     0       57|0|49|abc5|0|18|manager0|340521199501033310
-1      5       270     0       65|0|50|abc6|0|18|manager0|340521199501033310
-1      6       315     0       78|0|51|abc7|0|18|manager0|340521199501033310
-1      7       360     0       80|0|59|abc8|0|18|manager0|340521199501033310
-1      8       405     0       93|0|48|abc9|0|18|manager0|340521199501033310
-1      9       450     0       98|0|37|abc9|0|18|manager0|340521199501033310
-1      10      497     0       102|0|29|abc10|0|18|manager0|340521199501033310
-1      11      544     0       114|0|56|abc11|0|18|manager0|340521199501033310
-1      12      591     0       121|0|37|abc12|0|18|manager0|340521199501033310
-1      13      638     0       138|0|50|abc13|0|18|manager0|340521199501033310
-1      14      682     0       0|1|47|abc0|2|15|manager1|340521199501033311
-1      15      726     0       6|1|47|abc0|2|15|manager1|340521199501033311
-1      16      770     0       7|1|31|abc0|2|15|manager1|340521199501033311
-1      17      815     0       10|1|24|abc1|2|15|manager1|340521199501033311
-1      18      860     0       13|1|56|abc1|2|15|manager1|340521199501033311
-1      19      905     0       16|1|38|abc1|2|15|manager1|340521199501033311
-1      20      950     0       23|1|51|abc2|2|15|manager1|340521199501033311
```

6.10 三表连接

Birthday 数据：



```
Tablename: birthday
Attributes number: 3
Record length: 35
Record number: 4
-------- Page Head Info -------
PageNo  23
PageFreePalce   3889
RecordNum Stored in CurPage     4
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
210     0       31      0       340521199501033310|1880-10-10|0
211     1       62      0       340521199501033311|1990-10-12|1
212     2       92      0       340521199501033312|1989-10-2|2
213     3       123     0       340521199501033313|1989-10-10|3
```

注：目前只写了嵌套循环连接（在公共属性上的）

```
select * from employee e, department d, birthday b where e.did = d.did and d.idcard = b.idcard
```

```
PageNo  25
PageFreePalce   2270
RecordNum Stored in CurPage     24
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       60      0       134|1|39|abc13|2|15|manager1|340521199501033311|1990-10-12|1
-1      1       118     0       136|2|49|abc13|4|8|manager2|340521199501033312|1989-10-2|2
-1      2       178     0       138|0|50|abc13|0|18|manager0|340521199501033310|1880-10-10|0
-1      3       238     0       139|3|49|abc13|6|13|manager3|340521199501033313|1989-10-10|3
-1      4       298     0       140|1|28|abc14|2|15|manager1|340521199501033311|1990-10-12|1
-1      5       358     0       141|1|49|abc14|2|15|manager1|340521199501033311|1990-10-12|1
-1      6       416     0       143|2|45|abc14|4|8|manager2|340521199501033312|1989-10-2|2
-1      7       474     0       144|2|25|abc14|4|8|manager2|340521199501033312|1989-10-2|2
-1      8       532     0       147|2|32|abc14|4|8|manager2|340521199501033312|1989-10-2|2
-1      9       592     0       149|1|49|abc14|2|15|manager1|340521199501033311|1990-10-12|1
-1      10      652     0       154|1|52|abc15|2|15|manager1|340521199501033311|1990-10-12|1
-1      11      712     0       162|3|55|abc16|6|13|manager3|340521199501033313|1989-10-10|3
-1      12      772     0       170|3|55|abc17|6|13|manager3|340521199501033313|1989-10-10|3
-1      13      830     0       172|2|28|abc17|4|8|manager2|340521199501033312|1989-10-2|2
-1      14      888     0       173|2|45|abc17|4|8|manager2|340521199501033312|1989-10-2|2
-1      15      948     0       174|3|56|abc17|6|13|manager3|340521199501033313|1989-10-10|3
-1      16      1008    0       175|1|58|abc17|2|15|manager1|340521199501033311|1990-10-12|1
-1      17      1068    0       177|1|44|abc17|2|15|manager1|340521199501033311|1990-10-12|1
-1      18      1128    0       181|3|46|abc18|6|13|manager3|340521199501033313|1989-10-10|3
-1      19      1186    0       183|2|59|abc18|4|8|manager2|340521199501033312|1989-10-2|2
-1      20      1246    0       184|3|44|abc18|6|13|manager3|340521199501033313|1989-10-10|3
-1      21      1304    0       192|2|45|abc19|4|8|manager2|340521199501033312|1989-10-2|2
-1      22      1362    0       196|2|59|abc19|4|8|manager2|340521199501033312|1989-10-2|2
-1      23      1422    0       197|3|50|abc19|6|13|manager3|340521199501033313|1989-10-10|3
```

6.11 不同数据类型的多条件等值连接

我们以如下 SQL 语句为例：

select * from department d, birthday b where d.idcard = b.idcard and d.birthtime like '%19', 0-begin, 1-contains, 2-end

```
This a temporary tables.
-------- Page Head Info -------
PageNo  26
PageFreePalce   3952
RecordNum Stored in CurPage     2
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       45      0       340521199501033312|1989-10-2|2|2|4|8|manager2
-1      1       92      0       340521199501033313|1989-10-10|3|3|6|13|manager3
```

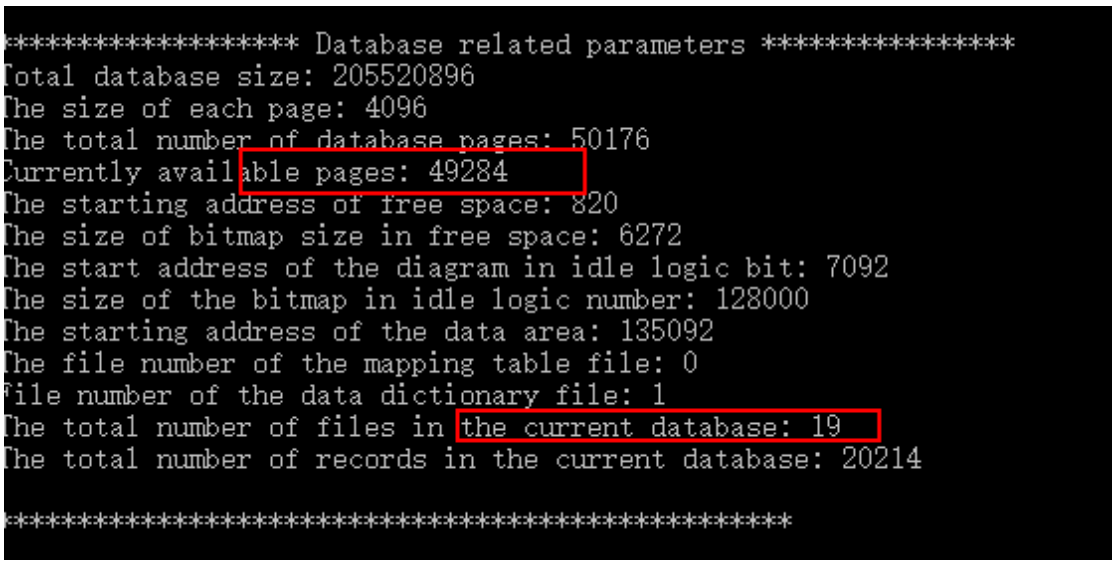
6.12 不同数据类型的多条件非等值连接

我们以如下 SQL 语句为例：

select * from department d, birthday b
where d.idcard = b.idcard and d.birthtime like '%19',

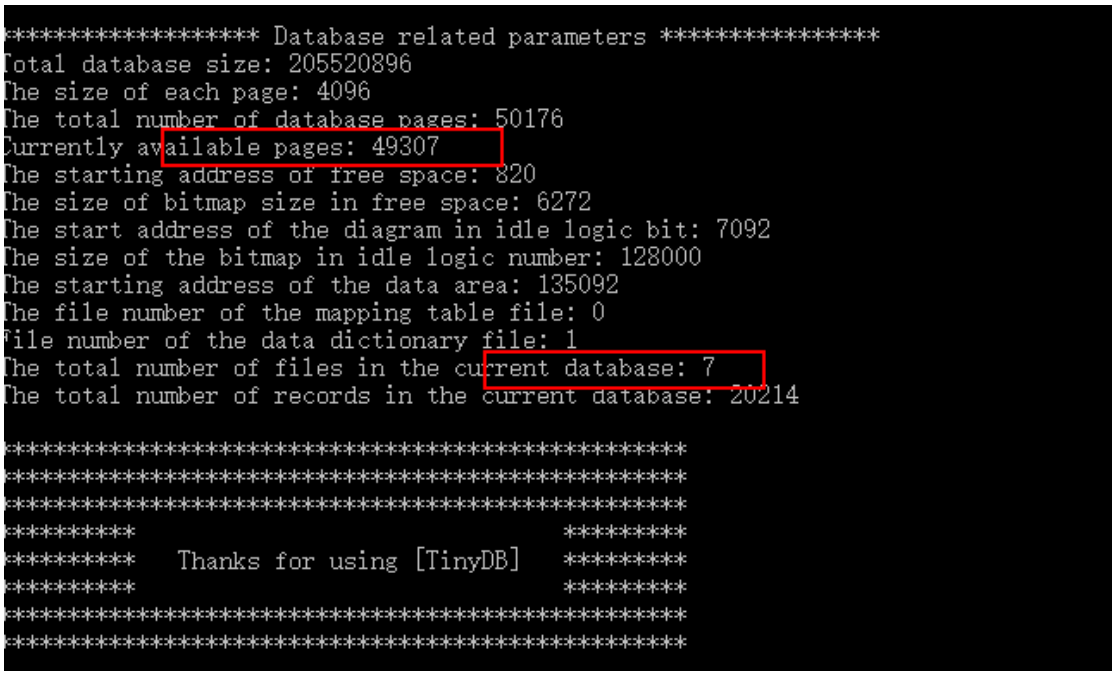
```
This a temporary tables.
-------- Page Head Info -------
PageNo  27
PageFreePalce   3950
RecordNum Stored in CurPage     2
Offset Table ----- Record Content
LogicID RecordNo        Offset  IsDelete        Record Content
-1      0       47      0       340521199501033310|1880-10-10|0|0|0|18|manager0
-1      1       94      0       340521199501033311|1990-10-12|1|1|2|15|manager1
```

6.13 执行完之后数据库系统参数



6.14 关闭数据库

回收临时表所占用的数据块、文件号等，显示更新数据库之后的系统参数。



6.15 其他功能

（1） 查询数据导出 csv

```
writeFile(&head, j5);
```

（2）简单 SQL（Create 语句）解析

```
string table_name;
    map<string, pair<int, int> > paras_map;
    createSql(sql, table_name, paras_map);
```

（3）除了自身为了系统方便使用了生成数据，也支持 TPCH 中的数据，详情请看（https://github.com/taoyouxian/tinydb/blob/master/SPJ/main.cpp）

7 使用手册

【Tiny TinyDB 】管理界面布局——控制台

```
*****************************************************************************************************
*      Welcome to TinyDB demonstration, choose your function:
*               1. Init tables(employee,department,birthday)
*               2. Insert datas to tables
*               3. Equivalent selection
*               4. Range selection
*               5. Projection
*               6. Join(Nested loop/Sort-based/Hash-based)
*               . Three tables connected
*               8. Multi-condition equivalent connection for different data types
*               9. Multi-conditional non-equivalentconnections of different data types
*               10. Show record bt logicID
*               11. Show mapping table file
*               12. Show system desc
*               13. Show file desc
*               14. Output(.csv)
*               15. Backup(.mat)
*               16. Main Window
*               0. Exit
*****************************************************************************************************

TinyDB> Please input your choise:
```

如上图所示，由于本次实验三的内容接口未详细实现，因此还未用 QT 管理界面布局，后面会加进去，用户可以根据控制台操作方法来进行自己的操作。

8 实验总结

由于实验二和实验三的完成人员并不相同，目前尚未做到将索引相关的功能加入到本 DBMS 系统中。目前本实验还有很多不足之处，功能不健全，代码风格略混乱，有待完善。希望能够通过实验四有更大的进步，进而更好地完成实验五。

由于 SPA 算法实现上有很大难度，因此我们也在这部分讨论了很长时间，以及讨论各个数据结构之间的交互与通信。然后接下来的任务就是怎样具体实现的问题了。我们团队采用的方式主要是分组进行实现各个模块进而整合的方式进行。