

# 数据库中期报告

计34 桥本优 2013011370

计34 钱雨杰 2013011372

计34 董胤蓬 2013011367

## 一. 阶段功能

我们小组的数据库实现了记录管理模块与系统管理模块的基础功能，具体的功能为：

1. 新建文件、删除文件、打开文件、关闭文件；
2. 插入记录、删除记录、更新记录；
3. 遍历记录、获取属性值满足条件的记录；
4. 创建、删除、切换数据库以及列出当前数据库的信息；
5. 创建表、删除表、列出现有的表及信息。
6. 对系统管理部分用户命令的解析。

## 二. 小组分工

前期的分工为：

桥本优：对文件的操作，包括新建、删除、打开、关闭文件；对数据库和表的创建删除等操作。

钱雨杰：插入、删除、更新记录；文件头以及页头的实现；用户命令的解析与执行。

董胤蓬：遍历查找满足条件的记录；通过查找显示表及当前数据库信息；中期报告。

后期分工：（待定）

桥本优：索引模块

钱雨杰：查询解析模块

董胤蓬：UI设计等

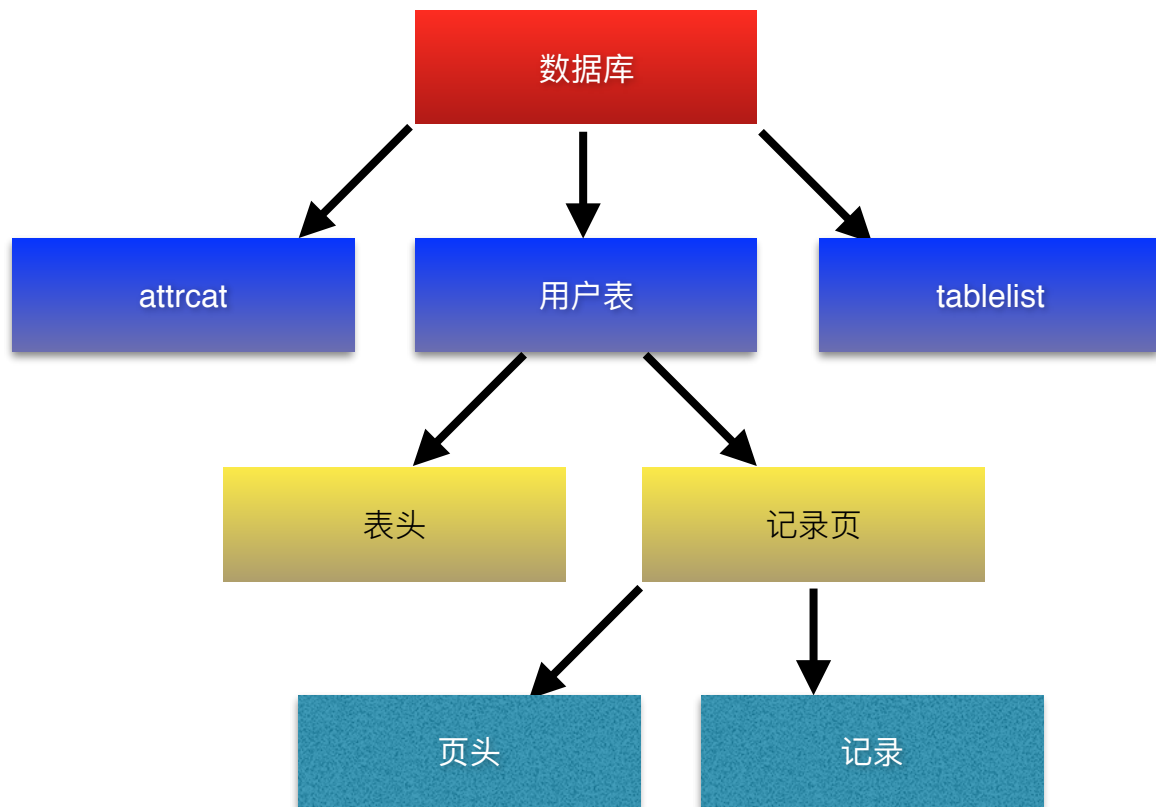
### 三. 具体实现

#### • 数据库设计

每个数据库在存储空间上对应于一个文件夹，其中包含很多文件，对应于每一个表。每个数据库中有两个专用的表，分别为attrcat与tablelist. attrcat表用于存储当前数据库中每个用户表的属性值，而tabellist表用于存储当前数据库所有用户表的信息。

数据库中每个表中的记录大小可以不一样，在新建表时指定，但是同一个表中的记录记录是定长的。在每个表中，第一页为存储文件信息的页，后续的各页用于存储记录。为了方便索引，在每页中页头部分记录当前页的使用情况。

下图可以表示出我们设计的数据库的基本结构：



下面将按照上图的结构从下至上介绍各个模块的类与接口定义，以及底层记录和RID类的接口。

## • 记录管理模块

### **class RM\_Record**

功能 ： 提供文件记录的基本操作

函数 ：

int Set(char \*pData, int size, RID rid\_) ： 设置记录内容

int GetData(char \*&pData) ： 获取记录内容

int GetRid(RID &rid) ： 获取记录RID

### **class RID**

功能 ： 记录的Identifier

**函数：**

RID() : 构造函数

~RID() : 析构函数

int GetPageNum (int &pageNum) const : 获取record的页数

int GetSlotNum(int &slotNum) const : 获取record的槽编号

int Page() const : 获取record的页数

int Slot() const : 获取record的槽编号

### **class FileHead**

**功能：** 文件第一页格式，文件第一页存储

**变量：**

int recordSize : 记录长度

int pageNumber : 页的个数

int recordPerPage : 每页记录个数

int recordNumber : 记录的总数

### **class PageHead**

**功能：** 数据页页头格式，暂时格式为一个int和84位的位图组成

**函数：**

bool getRecordHead(int RecordID) : 查询RecordID对应记录是否存在

int setRecordHead(int RecordID, bool IsRecord) : 设置RecordID对应的槽

### **class RM\_Manager**

**功能：** 处理RM部分的新建，删除，打开，关闭文件记录，程序仅实例化一个这样的类，所有请求均由这一个实例来处理

**函数：**

RM\_Manager(PF\_Manager&pfm)：构造函数,其中有页式文件系统的对象

~RM\_Manager()：析构函数

int CreateFile (const char \*fileName, int recordSize)：新建文件名为  
fileName的文件

int DestroyFile (const char \*fileName)：删除文件名为fileName的文件

int OpenFile (const char \*fileName,  
RM\_FileHandle &fileHandle)：打开fileName的文件

int CloseFile(RM\_FileHandle &fileHandle)：通过fileHandle获得fileID，调  
用FileManager::CloseFile

## **class RM\_FileHandle**

**功能：**操作使用RM Interface打开的文件中的记录

**函数：**

RM\_FileHandle()：构造函数

~RM\_FileHandle()：析构函数

int GetRec(const RID &rid, RM\_Record &rec) const：如果不存在rid，需要  
提示错误，如果成功，rec中存放记录的拷贝

int InsertRec(char \*pData, RID &rid)：插入一个pData指向的数据作为一个  
新的记录，如果成功，&rid将指向这个新插入记录的RID

int DeleteRec(RID&rid)：删除文件中记录为RID的记录，如果页中所含记录  
为空，调用FileManager::DisposePage处理该页

int UpdateRec(RM\_Record &rec)：更新所有和rec相关的记录

## **class RM\_FileScan**

**功能** : 提供RM文件记录的检索功能, 检索功能依赖于指定的几个条件

**函数** :

RM\_FileScan() : 构造函数

~RM\_FileScan() : 析构函数

int OpenScan(const RM\_FileHandle &fileHandle, AttrType attrType,  
int attrLength, int attrOffset, CompOp compOp,  
void \*value, ClientHint pinHint = NO\_HINT)

: 当value为空指针时, 查询所有记录, 如果不为空时, 指向value,用于查询比较。attrType的定义在database.h中INT, FLOAT和STRING. attrOffset为查询记录的初始位置(查询某个字段的值).

int GetNextRec(RM\_Record &rec) : 获取下一条record

int CloseScan() : 终止检索

## **• 系统管理模块**

### **struct DataAttrInfo**

**功能** : 用于在attrcat表中存放属性信息, 类中定义了属性的值

**变量** :

int offset : 属性的偏移量

int attrLength : 属性的长度

int indexNo : 属性的编号

AttrType attrType : 属性的类型

char relName[MAXNAME+1] : 关系的名称

char attrName[MAXNAME+1] : 属性名称

bool notNull : 属性是否为空

bool primaryKey : 主关键字

### **struct TableInfo**

功能 : 在tabellist表中存储用户表名称

变量 :

char tableName[MAXNAME + 1] : 用户表名称 (可扩展其他表信息)

### **class SM\_Manager**

功能 : 系统管理模块, 用于数据库的创建、删除、切换、显示; 某个数据库中表的创建、删除、显示

函数 :

int CreateDb(const char \*dbName) : 创建名为dbName的数据库

int OpenDb(const char \*dbName) : 打开名为dbName的数据库

int CloseDb() : 关闭数据库

int DropDb(const char \*dbName) : 删除名为dbName的数据库

int ShowDb(const char \*dbName) : 显示名为dbName的数据库信息

int CreateTable(const char \*tableName, int attrCount, AttrInfo \*attrs)

: 创建名为tableName的表, attrs为表中记录的属性

int DropTable(const char \*tableName) : 删除名为tableName的表

int Exec(const char \*instruction) : 解析用户指令并执行

## **四. 问题与困难**

数据库大作业进展过半, 我们遇到了很多的问题, 也解决了很多编程中的问题, 现在总结如下:

1. 由于我们使用助教给的页式文件系统，但是在完成第一阶段任务时发现编译不过，找了很长时间的bug，最后发现在pagedef.h中把一定变量定义在了全局，而之后编程时有多个cpp包含了这个头文件，导致了链接错误。所以也希望助教在之后的教学中可以修复这个问题。
2. 我们实现过程中参考stanford课程项目的框架来编写代码的，所以很多的类的接口都是借用stanford工程中的框架。虽然这个框架为我们提供了很多的便利，但是实现上有可能过于依赖其接口，我们在后面的编程中逐步摆脱其束缚，将实际需求与代码融合在一起。
3. 用户指令解析部分我们只实现了对数据库和表操作的指令。所以我们只是很简单的用if判断，没有添加很多可选的参数以及错误命令的错误信息输出。我们会考虑后续用Yacc工具来对用户命令进行语法分析，进而执行用户命令。

## 五. 课程建议

1. 冯老师的课程生动形象，针对具体的问题深入浅出地与我们讨论，我们在课堂上学习到了很多的东西。但是我们还是希望冯老师可以将上课的板书整理一下发给我们一份讲稿，因为我们可能因为熬夜的原因在早上第一节课不能集中注意力，所以错过了很多关键的东西。所以希望可以讲稿来课后学习。
2. 关于期中考试，我们觉得对于数据库的结构、B+树索引以及SQL语句有很大的必要去熟悉与掌握，但是开卷的期中考试会减低我们复习基础知识的积极性，所以建议将考试移到期末，进行正规一些的考试，让同学可以对数据库的基础知识多加了解。



3. 对于大作业的建议：大作业可以提升我们对数据库的理解与实现上的技巧，还可以提升我们的编程能力，所以这个大作业很有必要。建议鼓励不按照stanford的框架编程，这样编程的思维可以不被框架所束缚，写出符合需求的高效代码。