

Introduction to Database Project Summary Report

Chenghua Liu* Xinghan Liu†
Department of Computer Science
Tsinghua University

目录

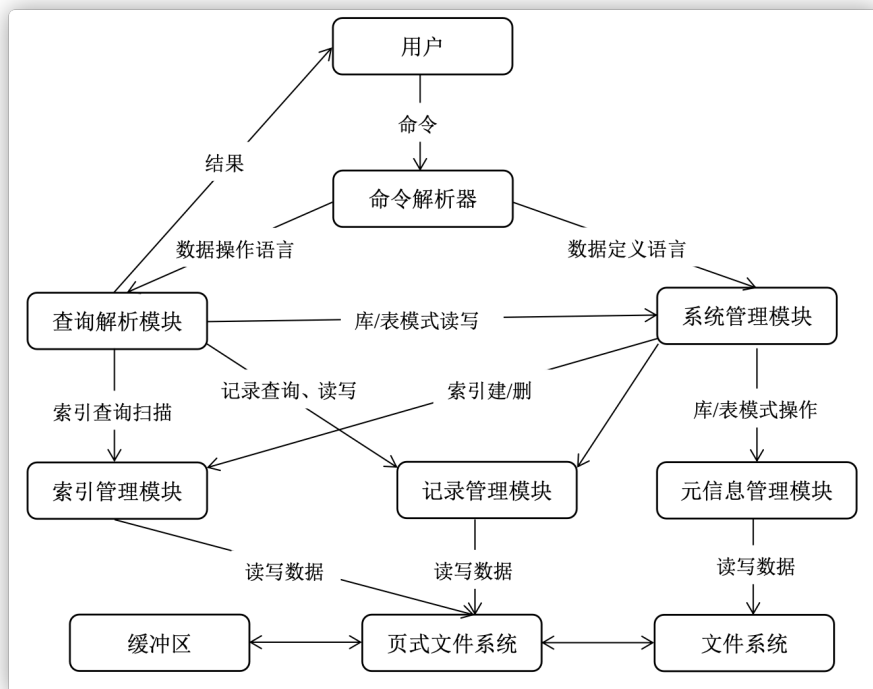
1	系统架构设计	1
2	各模块详细设计	2
2.1	页式文件系统（对应 FileSystem）	2
2.2	记录管理模块（对应 RecordSystem）	2
2.3	索引管理模块（对应 IndexSystem）	3
2.4	元信息管理模块（对应 MetaSystem）	3
2.5	查询解析模块 & 系统管理模块（对应 ManageSystem）	3
2.6	命令解析器（对应 SQL_Parser）	4
2.7	其他	4
3	主要接口说明	4
4	实验结果	5
5	小组分工	6
6	参考文献	6

1 系统架构设计

我们的系统架构大体上遵照实验指导，按照自下而上的顺序进行实现。

*2018011687,liuch18@mails.tsinghua.edu.cn

†2018012358,liuxh18@mails.tsinghua.edu.cn



底层方面，对于记录数据和索引数据，由于数据量庞大，我们采用页式文件系统存储管理；而对于库、表级别的数据（即一些数据库元信息），考虑到其空间占用不大，我们直接用 python 的 pickle 模块进行存储和读取，不使用页式文件系统。上层方面，我们使用 antlr 语法解析工具，结合本学期编译原理编写代码的经验，构造设计了能够解析大部分 SQL 语句的命令解析器，将命令解析为对相应模块接口的调用再输出结果，通过 CLI 与用户进行交互。

2 各模块详细设计

2.1 页式文件系统（对应 FileSystem）

参考实验指导提供的页式文件和缓冲区管理的“祖传实现”，我们设计了自己的页式文件系统。具体实现上，我们使用 FileManager 作为与底层文件系统进行页式文件读写的模块，每页大小为 8192 字节。在此基础上，我们用 BufManager 来实现缓冲区管理，与“祖传实现”相同，采用双向的环形链表 MyLinkedList 来执行 LRU 算法。

2.2 记录管理模块（对应 RecordSystem）

在记录管理模块中，我们定义了 Record 类，以及每个 Record 的唯一标识符 RID 类，以便后续用于索引。另外，我们在 RecordManager 中对所有下层的文件操作进行了进一步封装，在 FileHandler 中参考实验指导建议提供了对记录进行操作的各类接口，并且提供了用于扫描所有记录的类 FileScan，以便于上层进行查询操作。

记录文件由一个 Head Page 和若干 Record Page 组成（动态增加）。

每个 HeadPage 中存储了这张表的单条记录长度、单页最大能容纳的记录条数、目前的总页数、目前的总记录数、每页中 SlotBitmap 的长度等信息，并且存储了当前文件中有空余空间的页（称为“空闲页”）链表的入口页号。

每个 Record Page 在头部存储了下一空闲页表号，以及本页的 SlotBitmap，用以标识和快速查找空闲槽情况。随后整个页面由记录填充。尽管支持 varchar 类型数据，但我们采用的是定长记录形式，实际每条记录中的 varchar 长度都定义为其声明的最大长度。

2.3 索引管理模块（对应 IndexSystem）

在索引管理模块中，我们构造了 B 树来组织记录数据，用每条数据的唯一标识符 RID（由页号、槽号构成）作为索引键。对于各类约束对索引的要求，我们的处理是：如果某列被声明为主键，则自动在该列创建索引，且不允许重复和 NULL 值；如果某列被声明为外键，则首先检查其引用的列是否有索引，如果没有则自动创建，然后再检查被声明为外键的这列是否有索引，如果没有则创建；如果给某列加上唯一性约束 unique，则自动为该列创建索引。

2.4 元信息管理模块（对应 MetaSystem）

对每个数据库而言，元信息管理模块主要负责管理库级别的一些元信息，以及其中每个表的一些元信息。具体来说，库的元信息包括各表之间的外键映射关系，以及一些索引名称、外键名称和具体的列的映射关系。而表的元信息则包括了各列的数据长度、数据类型、有无对应索引以及有无约束信息等，这些信息可以在上层调用 DESC 语句时被一并展示出来。

2.5 查询解析模块 & 系统管理模块（对应 ManageSystem）

由于查询解析模块和系统管理模块的行为模式有一定相似之处：接受从命令解析器传来的接口调用，对下层的索引、记录、元信息管理模块进行读、写、扫描等操作，故我们把这两个模块合并在一起实现。

在 SystemVisitor 中，我们重载了 antlr 工具自动生成的 visitor 模式的各条 SQL 语句对应的函数，并针对每条语句解析的结果，从中获取我们所需要的参数，传给 SystemManager 进行操作。

在 SystemManager 中，我们实现了的功能有：基本查询解析（记录的增删改查），索引的创建/删除，数据表/数据库的 schema 增删（包括各类约束条件的增删，实现了比较完善的完整性约束，支持联合主键），双表连接查询，嵌套查询，聚合查询，模糊查询，分组查询，分页查询等。

除此之外，该模块还需要提供操作结果的输出，故我们定义了自己的 LookupOutput 类作为查询结果的标准，实现了 printer 类，其中利用 prettytable 库构建了模仿 Mysql 的 CLI 界面。

2.6 命令解析器 (对应 SQL_Parser)

这个模块的代码主要由 antlr 工具自动生成, 其功能为根据实验指导提供的 SQL.g4 文件, 按照 visitor 模式生成抽象语法树 AST, 完成对用户输入命令的解析 (包括词法、语法分析), 然后调用对应的 visitor 函数 (在查询解析模块中被重载), 随后交给查询解析模块和系统管理模块执行对应的操作。

2.7 其他

在 Exceptions 目录下, 我们定义了自己的异常类型 MyException, 在此基础上把程序执行中各种可能会产生的异常 (文件系统的文件打开关闭异常、记录的长度异常、类型异常、数据库/表元信息的异常、不满足完整性约束的异常、查询结果异常等等) 汇总到一起, 在触发这些异常时输出响应的报错信息, 在不导致程序崩溃, 不引起额外 bug 的基础上提示并引导用户进行正确的交互操作。

3 主要接口说明

1. 页式文件系统以及缓冲区:

创建、打开、关闭、删除、替换文件, 按页粒度新建、读取、写入文件, 标记脏页、写回内存等。

2. 记录管理模块:

插入、删除、更新、扫描记录, 以及由此产生的对空闲页链表的管理操作等。

3. 索引管理模块:

创建、删除、读取、写入索引文件, 用索引查询, 插入、删除 B 树节点等

4. 元信息管理模块:

修改数据库元信息 (新增/删除数据表, 新增/删除外键、索引), 修改数据表元信息 (新增/删除列, 新增/删除约束), 格式化展示表 schema 等。

5. 系统管理模块:

数据库增删换, 数据表 schema 增删看, 索引增删, 约束增删等。

6. 查询解析模块:

按查询条件/聚合条件/模糊查询构造过滤函数, 按索引查找/遍历式查找, 对记录数据的序列化和反序列化。

4 实验结果

我们在验收第一天就参与完成了统一验收，包括必做功能和若干选做功能，索引的性能提升也有所体现。值得一提的是，尽管我们使用了 Python 作为我们的语言，但我们的性能与助教 C++ 语言实现的性能相差较少，助教验收时对此十分惊讶。一份简单的附加功能测试代码如下：

聚集查询：

```
CREATE TABLE G(A INT(10), B VARCHAR(10));
INSERT INTO G VALUES(1, 'ABC');
INSERT INTO G VALUES(2, 'BCD');
INSERT INTO G VALUES(2, 'CDE');
SELECT COUNT (A) FROM G;
SELECT AVG(A) FROM G;
SELECT MAX(A) FROM G;
SELECT MIN(A) FROM G;
SELECT SUM(A) FROM G;
```

模糊查询：

```
SELECT * FROM G WHERE B LIKE '%C_';
SELECT * FROM G WHERE B LIKE '%B%';
SELECT * FROM G WHERE B LIKE '_D_';
```

嵌套查询：

```
CREATE TABLE H(A INT(10), B INT(10));
INSERT INTO H VALUES(1,2);
INSERT INTO H VALUES(2,3);
SELECT * FROM H WHERE A IN (1,2);
SELECT * FROM H WHERE A IN (SELECT B FROM H);
```

分页：

```
SELECT * FROM H LIMIT 1 OFFSET 1;
```

分组：

```
INSERT INTO H VALUES(1,3);
INSERT INTO H VALUES(2,4);
SELECT A, SUM(B) FROM H GROUP BY A;
SELECT A, AVG(B) FROM H GROUP BY A;
```

日期：

```
CREATE TABLE I(D DATE);
INSERT INTO I VALUES('2022-1-9');
SELECT * FROM I;
```

```
UNIQUE 约束:
CREATE TABLE K(K INT(10));
ALTER TABLE K ADD UNIQUE MYU(K);
INSERT INTO K VALUES(1);
INSERT INTO K VALUES(1); #ERROR
NULL 约束:
CREATE TABLE L(A INT(10));
INSERT INTO L VALUES(NULL);
SELECT * FROM L WHERE A IS NOT NULL;
```

5 小组分工

- 刘星翰：页式文件系统、记录管理模块、元信息管理模块、部分查询解析模块（基本查询、模糊查询等）、部分系统管理模块
- 刘程华：索引管理模块、命令解析器、部分查询解析模块（双表连接、聚合查询、索引查询等）、部分系统管理模块

6 参考文献

我们参考了 Stanford 的 RedBase 项目中对数据库实现的一些建议。