数据库系统概论–大作业报告

熊天翼　　李宣成

计 95　　计 92

2019011303　　2018011133

1　系统架构设计

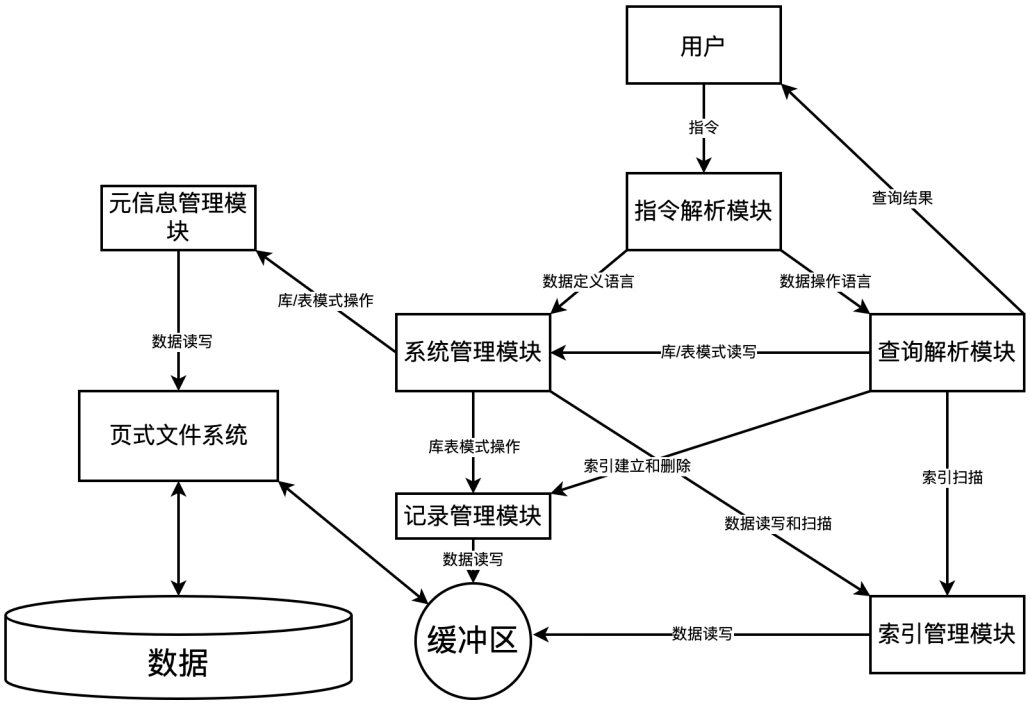

图 1: 系统架构设计

我们主要参考了实验指导书的框架，自底向上构建我们数据库。底层方面，为了存储大量的原始数据和索引数据，我们采用页式文件系统进行存储管理，而对于表、库级别的元信息，考虑其文件大小较小，我们使用 python 的 pickle 包直接进行读写。在中层，我们代码的核心为系统管理模块，其中针对不同的数据操作调用文件管理、记录管理、索引管理、元信息管理模块的各接口进行具体的数据库更新操作。上层方面，我们利用 antlr 生成了 SQL 语言的 parser，结合所学的编译知识继承了其中的 Visitor 类，通过调用系统管理模块的对应接口函数将解析好的 SQL 语句转化为对数据库的实际操作。

2 各模块详细设计与接口说明

2.1 页式文件系统

2.1.1 详细设计

页式文件设计中我们主要参考了实验指导书上的 C++ 代码并且在此基础上进行了修改。我们设计了 FileManager 模块来管理数据库和底层文件系统的读写；在 FileManager 的基础上引入 BufManager 来管理缓冲区。缓存更新依靠双向环状链表的实现 LRU 算法进行。我们系统中页面大小为 8192 字节，缓存大小为 8k 页。

2.1.2 接口说明

文件基本操作：创建、打开、关闭、删除、替换。

页操作：新建、读取、写入文件，标记脏页、写回内存。

2.2 记录管理系统

2.2.1 详细设计

记录管理系统负责对存储记录的文件进行管理，为每张表生成一个记录文件。为此我们实现了记录类 Record 以及作为 Record 唯一标识符的 RID 类（由页号和槽号组成）。每条记录由一个首页和多个数据页构成，采用定长方式记录。首页存储表长、单页可容纳最大记录数、总页数、总记录数、bitmap 长度，当前文件空闲页链表的入口页号等信息。bitmap 是记录本页有空位的位置。数据页除了存储记录之外，还存储了下一空闲页号，本页的 bitmap，用以标识和快速查找空闲槽情况。

此外，我们使用 RecordManager 来对页式文件系统的操作进行了封装，并且定义了 FilScan 类用于上层模块扫描所有记录。

2.2.2 接口说明

对记录的插入、更新、删除等操作并且为上层提供了扫描全部记录的接口。

2.3 索引管理系统

2.3.1 详细设计

索引模块中我们为文件中的记录建立了 B+ 树索引以提高查询速度，在叶节点处进行 RID 的存储。关于 B+ 树的原理，我们主要参照了实验指导书的说明。在实现中，我们定义了抽象类树结点并且派生出叶结点和内部结点两种结点，每个节点会动态计算节点大小来保证可以存储于文件系统的一页中，若超过 page size 则进行分裂。内部结点记录下层节点以及其对应的最大索引 key，叶子结点记录索引 key 和对应的 RID。结点内部采用二分法查找边界。在我们的实现下，验收中的数据表格对应的 B+ 树的树高不会超过 3。

处理约束的逻辑如下：该列主键：自动创建索引; 该列为外键：依次检查引用列、该列是否有索引，如果没有则自动创建; 该列为 unique，则自动为该列创建索引。我们为 NULL 值设定了一个足够小的负数，以保证 B+ 树的高效性。

2.3.2 接口说明

B+ 树的对外接口主要包括：插入、删除、单个 key 搜索、范围 [low, high] 搜索、序列化（写入文件）、反序列化（从文件读取）.....

索引相关操作：创建索引、删除索引、写入索引文件、从文件读取索引、用索引查询等等

2.4 元信息管理系统

2.4.1 详细设计

元信息管理系统在系统管理的下层，为系统管理服务，主要管理数据库、数据表、列的相关元信息。在此模块中，我们使用了"pickle" 函数将这些信息直接存到文件系统中。上层模块可以通过对应的数据库名称、数据表名称、列名称获取元信息。

2.4.2 接口说明

数据库元信息相关操作：新增/删除数据表，新增/删除外键、索引

数据表元信息相关操作：新增/删除列，新增/删除约束，格式化展示表 schema 等。

2.5 系统管理

2.5.1 详细设计

系统管理模块主要实现了接受从命令解析器传来的指令，调用对应的接口从而对下层的模块进行读、写、扫描等操作。在 SystemManager 中我们首先实现了有关查询解析的相关功能，大致流程如下：1. 根据查询推导表名 2. 判断是否使用索引 3. 根据查询条件构造对应的 lambda 用于删选 4. 扫描并筛选数据 5. 返回。然后，我们实现了数据库/表 Schema 增删，索引增删，以及嵌套、聚合、模糊查询，分组分页查询、双表连接查询等附加功能。

此外，我们还定义了输出结果的 Printer 类用于输出查询结果，使用 prettytable 库模仿 MYSQL 的 CLI 界面。

2.5.2 接口说明

查询解析：解析不同类型的查询构造过滤函数，索引/遍历查找，对记录数据的序列化和反序列化。

系统管理：对于数据库的新建切换删除，对于数据表的新建删除和查看，对于索引约束的新建和删除等。

2.6 命令解析

2.6.1 详细设计

命令解析模块利用 antlr 通过 SQL.g4 文法文件自动生成。用于解析用户指令之后调用系统管理的方法执行指令。

2.7 异常处理

2.7.1 详细设计

异常处理模块主要用于处理用户的非法输入以及数据库运行中出现的各种异常，输出报错信息，防止程序崩溃。

3 实验结果

3.1 项目运行方式

项目依赖：`numpy, prettytable, antlr4-python3-runtime==4.9.2`

运行方式如下：

- **命令行交互**: `python main.py`
- **导入 SQL 文件执行**：`python main.py -f <sql-path>`
- **CSV 批量导入数据**：`python main.py -f <csv-path> -t <table-name> <db-name>`

3.2 结果与展示

在验收过程中，我们的基础与拓展功能绝大部分验收正确，个别地方存在系统报错情况。此外，我们 python 代码的查询和执行效率很高，得到了助教的肯定（据说甚至比一些 C 语言的实现更快）。这体现了我们功能实现与优化的有效性。

以下是一份简要展示部分拓展功能的测试代码。

```
CREATE TABLE T1(A INT, B VARCHAR(10));
INSERT INTO T1 VALUES(1,'ABECF'), (2,'BCD'), (3,'CDBD'), (4,'CDE') ;
-- 聚集查询:
SELECT COUNT (A) FROM T1;
SELECT AVG(A) FROM T1;
SELECT MAX(A) FROM T1;
SELECT MIN(A) FROM T1;
SELECT SUM(A) FROM T1;
-- 模糊查询:
SELECT * FROM T1 WHERE B LIKE '%C_';
SELECT * FROM T1 WHERE B LIKE '%B%';
SELECT * FROM T1 WHERE B LIKE '_D_';
-- 嵌套查询:
CREATE TABLE T2(A INT, B INT);
INSERT INTO T2 VALUES(1,2), (2,3), (3,3);
```

```
SELECT * FROM T2 WHERE A IN (SELECT B FROM T2);
-- 分组查询:
INSERT INTO T2 VALUES(1,3), (2,4), (1,1), (2,6);
SELECT A, SUM(B) FROM T2 GROUP BY A;
SELECT A, AVG(B) FROM T2 GROUP BY A;
-- 分页查询:
SELECT * FROM T2 LIMIT 1 OFFSET 1;
-- UNIQUE约束:
CREATE TABLE T3 (A INT);
ALTER TABLE T3 ADD UNIQUE A(A);
INSERT INTO T3 VALUES(1);
INSERT INTO T3 VALUES(1); --ERROR
-- 日期:
CREATE TABLE T4(D DATE);
INSERT INTO T4 VALUES('2023-1-7');
SELECT * FROM T4;
```

4 分工、感想与致谢

在作业的完成过程中，我们进行了如下分工：

- **熊天翼**：文件管理、索引管理、解析模块（适配 antlr 生成的编译器）、部分系统管理
- **李宣成**：记录管理、元信息管理、部分系统管理、查询

时光飞逝，一转眼一学期又过去了。对于我们两个大四学生，这也意味着本科生活即将走向终点。回顾一学期的数据库课程，从最开始奋力起床赶周一的早八，到后面因为疫情不得不线上学习，再到后来封校、新冠、放开……从某种角度而言，这或许也是我们在疫情时代完成的最后的课程学习了，因此这段学习经历更应当得到珍藏。

说回课程本身，从无到有的搭建数据库，对我们而言是全新的挑战和收获。在这个过程中，我们巩固了数据库知识，提升了自己的项目管理、代码书写和 debug 的能力。但与此同时，我们认为课程也存在改进空间。一方面是，大作业可以拆成不同的模块，或提供基本的框架和接口，这样同学同样可以在实现功能过程中掌握数据库的相关知识，同时可以平均工作量并进行更多探索；另一方面，课堂的趣味性和生动性的提升，也一定能让同学们更专注于课堂本身，在课堂上学习。

最后，很高兴可以选学“数据库系统概论”课程。衷心感谢负责任的老师和助教们！

5 参考资料

我们除作业文档（https://thu-db.github.io/dbs-tutorial/）之外主要参考了如下文献：

- **课程幻灯片**。学习了有关数据库架构设计、记录管理、SQL 语言等方面的基本知识
- **Stanford CS346 课程网站**。对照文档一起阅读，帮助我们理解数据库各模块的分工和接口。
- **往年的 Python 实现代码**。我们参考了https://github.com/rcy17/pybase。这份代码在项目框架、模块接口等方面为我们提供了较大帮助。