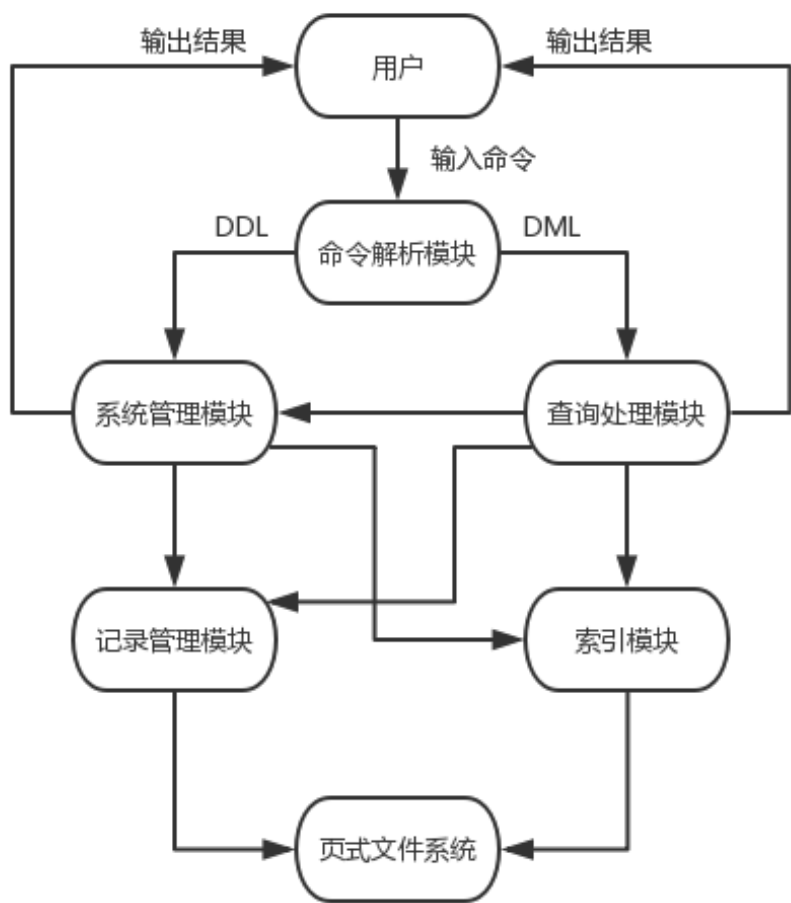


# 数据库系统概论 项目报告

计54 徐毅 (2015011334)  
计54 阎世宏 (2015011316)

## 概述与系统架构

我们基于 Stanford CS 346 Redbase 项目的框架 (以下简称“框架”), 实现了一个单用户的关系数据库管理系统 RippleDB, 系统架构如下:



其中, 页式文件系统完全由框架提供, 我们主要基于框架实现了记录管理模块、索引模块、系统管理模块、查询处理模块和命令解析模块, 从而实现了全部基本功能。

此外, 我们还实现了以下附加功能: 扩展数据类型、外键约束、聚集查询、分组聚集查询、模糊查询、多表连接与查询优化。

## 功能说明与实验结果

---

下面举例描述系统支持的功能与相应的 SQL 语句，同时给出了运行截图。

### DDL

- 创建名为 orderDB 的数据库：

```
RIPPLEDB >> CREATE DATABASE orderDB;  
[CreateDB]  
dbName=orderDB
```

- 删除名为 orderDB 的数据库：

```
RIPPLEDB >> DROP DATABASE orderDB;  
[DropDB]  
dbName=orderDB
```

- 列出当前所有的数据库：

```
RIPPLEDB >> SHOW DATABASES;  
orderDB  
[ShowDBs]
```

- 当前数据库切换为 orderDB：

```
RIPPLEDB >> USE orderDB;  
[OpenDB]  
dbName=orderDB
```

- 列出当前数据库包含的所有表：

```
RIPPLEDB >> SHOW TABLES;  
relcat  
attrcat  
[ShowTables]
```

此时看到了两个系统表 relcat 与 attrcat。

- 创建名为 price 的表，它包含三个字段 website\_id, book\_id 和 price，其中 website\_id 与 book\_id 是外键，二者共同构成主键，这三个字段的数据类型分别为整型、整型和实型，并且都不允许为空：

```
RIPPLEDB >> CREATE TABLE price(
    website_id INT(10) NOT NULL,
    book_id INT(10) NOT NULL,
    price FLOAT NOT NULL,
    PRIMARY KEY (website_id,book_id),
    FOREIGN KEY (website_id) REFERENCES website(id),
    FOREIGN KEY (book_id) REFERENCES book(id)
);
[CreateTable]
relName=price
fieldCount=6
attributes=
website_id INT NOT NULL PRIMARY KEY FOREIGN KEY REFERENCES website(id)
book_id INT NOT NULL PRIMARY KEY FOREIGN KEY REFERENCES book(id)
price FLOAT NOT NULL
```

除此以外，还支持日期类型 DATE 和变长字符串类型 VARCHAR(MAX\_LENGTH)。

- 删除名为 orders 的表：

```
RIPPLEDB >> DROP TABLE orders;
[DropTable]
relName=orders
```

- 列出 price 表的模式信息：

```
RIPPLEDB >> DESC price;
website_id INT 4 NOT NULL PRIMARY KEY FOREIGN KEY REFERENCES website(id)
book_id INT 4 NOT NULL PRIMARY KEY FOREIGN KEY REFERENCES book(id)
price FLOAT 4 NOT NULL
[DescTable]
relName=price
```

- 打印系统表 relcat 的全部记录：

```
RIPPLEDB >> PRINT relcat;
relName      tupleLength attrCount  indexCount
-----
relcat       40          4          0
attrcat      130         10         0
customer     35          3          2
book         416         5          2
website      134         3          2
price        15          3          5
orders       37          6          2

7 tuple(s).
[Print]
relName=relcat
```

此时已经创建了 customer, book, website, price, orders 五个表。

- 为 customer 表的 name 字段创建 / 删除索引:

```
RIPPLEDB >> CREATE INDEX customer(name);
[CreateIndex]
relName=customer
attrName=name

RIPPLEDB >> DROP INDEX customer(name);
[DropIndex]
relName=customer
attrName=name
```

## DML

操作前建立了一个新的表 test:

```
RIPPLEDB >> CREATE TABLE test(
    id INT(10) NOT NULL,
    name VARCHAR(25) NOT NULL,
    gender VARCHAR(1) NOT NULL,
    university VARCHAR(20),
    PRIMARY KEY(id)
);
[CreateTable]
relName=test
fieldCount=5
attributes=
id INT NOT NULL PRIMARY KEY
name STRING NOT NULL
gender STRING NOT NULL
university STRING
```

## 插入

正常插入如下:

```
RIPPLEDB >> INSERT INTO test VALUES(1, 'ysh', 'M', 'thu');
```

```
RIPPLEDB >> PRINT test;
id          name          gender university
-----
1          ysh          M          thu

1 tuple(s).
[Print]
relName=test
```

异常插入如下：

```
RIPPLEDB >> INSERT INTO test VALUES(1, 'ysh', 'M', 'thu');
QL warning: QL_PRIMARYKEYREPEAT

RIPPLEDB >> INSERT INTO test VALUES(2, 1, 'M', 'thu');
QL warning: attr type not right

RIPPLEDB >> INSERT INTO test VALUES(2, NULL, 'M', 'gg');
QL warning: attr is null

RIPPLEDB >> INSERT INTO test VALUES(2, 'ysh', 'M', 'thu', 'gg');
QL warning: attr number not right

RIPPLEDB >> INSERT INTO test VALUES(2, 'ysh', 'MA', 'thu');
QL warning: QL_STRINGLENGTHWRONG
```

分别提示了主键重复、值与属性类型不符、非空属性出现空值、值与属性数量不符、字符串超过设定的长度限制。

## 删除

正常删除如下：

```
RIPPLEDB >> DELETE FROM test WHERE id = 1;

RIPPLEDB >> PRINT test;
id          name          gender university
-----
0 tuple(s).
[Print]
relName=test
```

异常删除如下：

```
RIPPLEDB >> DELETE FROM test WHERE gg = 1;
QL warning: attribute not found

RIPPLEDB >> DELETE FROM test WHERE id = 'M';
QL warning: attr type not right
```

分别提示了属性未找到、值与属性数量不符。

更新

正常更新如下：

```
RIPPLEDB >> PRINT test;
id      name      gender university
-----
1       ysh       M       thu
2       xyi       M       thu

2 tuple(s).
[Print]
relName=test

RIPPLEDB >> UPDATE test SET name = 'xy' WHERE id = 2;

RIPPLEDB >> PRINT test;
id      name      gender university
-----
1       ysh       M       thu
2       xy        M       thu
```

```
RIPPLEDB >> UPDATE test SET university = 'Tsinghua' WHERE university IS NOT NULL;
```

```
RIPPLEDB >> PRINT test;
id      name      gender university
-----
1       ysh       M       Tsinghua
2       xy        M       Tsinghua

2 tuple(s).
[Print]
relName=test
```

外键约束如下：

```
RIPPLEDB >> UPDATE orders SET customer_id = 316001 WHERE id= 100001;
QL warning: QL_FOREIGNKEYNOTEXIST
```

查询

正常查询如下：

```
RIPPLEDB >> SELECT * FROM book WHERE publisher = 'Monday Morning Books' AND id > 270000;
id      title      authors      copies
-----
279385   My Life as a Fifth-Grade Comedian      Joan Sweeney      2368
283473   Hilda Boswell\'s Treasury of Poetry      Peter Cannon      7204
286736   Time for Tea (Helen Exley Giftbook)      Ita Buttrose      6859
286987   Tap d\'escopeta (M?nima de butxaca)      Joseph C. Pearce      7123

4 tuple(s).
```

```
RIPPLEDB >> SELECT title FROM book WHERE authors IS NULL;
title
-----

0 tuple(s).
```

聚集查询如下：

```
RIPPLEDB >> SELECT SUM(quantity) FROM orders;
quantity
-----
799547

1 tuple(s).

RIPPLEDB >> SELECT AVG(quantity) FROM orders;
quantity
-----
7

1 tuple(s).

RIPPLEDB >> SELECT MAX(quantity) FROM orders;
quantity
-----
15

1 tuple(s).

RIPPLEDB >> SELECT MIN(quantity) FROM orders;
quantity
-----
1

1 tuple(s).
System Settings
```

分组聚集查询如下 (同时体现了扩展类型 FLOAT)：

```
RIPPLEDB >> SELECT website_id, AVG(price) FROM price GROUP BY website_id;
website_id  price
-----
1001        222.733841
1002        225.369232
1003        222.105621
1004        221.412720
1005        219.985306
1006        221.388992
1007        219.788177
1008        220.924667
1009        223.622879
1010        219.660645
1011        223.821228
1012        224.550293
1013        219.261429
1014        220.920883
1015        219.891174
1016        225.680481
1017        219.142792
1018        219.613815

18 tuple(s).
```

模糊查询如下：

```

RIPPLEDB >> SELECT title, authors FROM book WHERE authors LIKE '____David%';
title
-----
Bathroom Sports Quiz Book          Kay David
Don Juan (Penguin Classics)        Cay David
Fairy Tales of Charles Perrault    Joe David Brown
The Concise Encyclopedia of Hockey Joy David
Vergiss nie, da?? ich Dich liebe.  Jay David
Punch With Judy (An Avon Flare Book) Max Davidson
Don Juan (Penguin Classics)        Cay David
Fairy Tales of Charles Perrault    Joe David Brown
The Concise Encyclopedia of Hockey Joy David
Vergiss nie, da?? ich Dich liebe.  Jay David
Punch With Judy (An Avon Flare Book) Max Davidson

11 tuple(s).
```

多表连接如下 (同时体现了扩展类型 DATE)：

```

RIPPLEDB >> SELECT customer.name, website.name, book.title, orders.quantity FROM website, customer, book,orders WHERE website.id=orders.
website_id AND customer.id=orders.customer_id AND book.id=orders.book_id and orders.odate='2017-10-01';
title      customer.name      quantity      website.name
-----
Money (Eyewitness Books)      LOLITA SCHANK      11      jiushu
A Corner of the Universe      JERROD VALERIO     11      buybook
The Lake of Dead Languages    CATHEY CATLIN       3      tmall
Understanding the Constitution AFTON SKILLMAN     13      ucgo5
The Awakening (Vampire Diaries) DANA RANCOURT       3      winxuan
Intuition: Knowing Beyond Logic MIKEL BUSHNELL     11      suning
50 Below Zero (Munsch for Kids) VAL RIO             15      yhd
Betty Crocker\'s Bisquick Cookbook TATUM LACKEY        4      suning
Curries & Rice (Chefs Special) DOYLE DAILY         1      tmall
Aman Story of a Somali Girl  LUPE DELCID         4      winxuan
Night Embrace (Dark-Hunter)  KYLIE HUMBLE        10     gg
You\'re Missin\' a Great Game KARLEEN EBY         11     sz
Jonathan Livingston Seagull   PATIENCE DUMMER      8      booyee
The American Century Cookbook ELSA CURROLA        9      buybook
The Blessing of a Broken Heart GABRIELLE NICHOL    3      jd
Serendipities: Language and Lunacy THADDEUS BYNUM       2      jd

16 tuple(s).
```

# 主要模块设计

首先说明一下全局定义部分 ( `global.h` , `attr.cc` , `rid.cc` ), 包括了各模块所需的常量, 数据类型及其比较函数, 运算符表, RID 类 (RID 包括页号和槽号, 是一个记录的唯一标识) 等。

为处理 NULL, 在一条记录的数据中, 并非由各属性数据直接拼接而成, 而是在各属性数据前增加 1 字节表示该属性是否为 NULL。对于字面量的解析也使用了同样的处理方式。

一个数据库对应一个目录, 一个表则对应该目录下的一个文件, 一个表的某个属性的索引对应该目录下的两个文件 (一个存 NULL 值, 一个存非 NULL 值)。

下面依次介绍各主要模块的设计。

## 记录管理模块

记录管理模块主要用于管理存储记录的文件, 其依赖于框架提供的页式文件系统 ( `pf.h` , `pf_internal.h` , `pf_error.cc` , `pf_manager.cc` , `pf_filehandle.cc` , `pf_pagehandle.cc` , `pf_buffermgr.h` , `pf_buffermgr.cc` , `pf_hashtable.h` , `pf_hashtable.cc` )。

## 本模块代码说明

- `rm.h` : 记录管理模块声明。
- `rm_error.cc` : 记录管理模块错误输出。
- `rm_manager.cc` : 记录文件管理器, 提供创建指定文件名和单条记录大小的记录文件、删除指定文件名的记录文



件、打开指定文件名的记录文件并返回记录文件操作接口、关闭记录文件操作接口的接口。

- `rm_filehandle.cc`：记录文件操作接口，提供根据 RID 查找 / 删除 / 更新记录、插入新记录并返回 RID 的接口。
- `rm_filescaan.cc`：记录文件扫描器，绑定于一个记录文件操作接口，提供创建扫描、返回下一个满足条件的记录、关闭扫描的接口。
- `rm_record.cc`：记录类，包括数据指针和 RID。

## 记录文件存储结构

记录文件分页存储，对于空闲页使用链表记录，对于页内各槽是否空闲使用 bitmap 标记，故有了下面的存储结构。

记录文件的首页存储文件头，包括单条记录大小 (定长)、每页记录条数、首个空闲页号 (即链表头)、每页标记各槽是否空闲的 bitmap 大小。

记录文件的其余页首先包括一个页头，包括下个空闲页号 (即链表的 next 指针)、标记各槽是否空闲的 bitmap，接下来就是定长的各条记录数据。

## 索引模块

索引模块主要用于为记录文件的某个属性建立 B+ 树索引以加快查询速度，其同样依赖于框架提供的页式文件系统。

### 本模块代码说明

- `ix.h`：索引模块声明。
- `ix_error.cc`：索引模块错误输出。
- `ix_internal.cc`：索引模块公用全局函数。
- `ix_bplustree.cc`：B+ 树结构，封装了页式文件系统，为索引模块提供插入、删除、查找等接口。
- `ix_manager.cc`：索引管理器，提供创建指定属性的索引、删除指定属性的索引、打开指定属性的索引并返回索引操作对象、关闭索引操作对象的接口。
- `ix_indexhandle.cc`：索引操作对象，提供插入索引记录、删除索引记录的接口。
- `ix_indexscan.cc`：索引扫描器，绑定于一个索引操作对象，提供创建扫描、返回下一个满足条件的索引记录、关闭扫描的接口。

## 索引存储结构

索引分页存储。

索引文件的首页存储 B+ 树的基本信息与被索引属性的基本信息。

索引文件的其余页头部记录 B+ 树结点的基本信息，其后存储关键字数组与子结点数组。

## 系统管理模块

系统管理模块主要用于完成 DDL 语句对应的操作，核心是维护系统表，其中打印表操作使用了框架提供的打印辅助类 (`printer.h`, `printer.cc`)。

### 本模块代码说明

- `sm.h`：系统管理模块声明。

- `sm_error.cc`：系统管理模块错误输出。
- `sm_manager.cc`：系统管理器，提供创建 / 删除 / 打开指定名字数据库、显示数据库列表、关闭当前数据库、显示当前数据库内表的列表、创建 / 删除 / 描述 / 打印指定名字表、创建 / 删除指定名字表和属性的索引的接口。
- `sm_internal.cc`：系统表 RelCat 与 AttrCat 对应的类。

## 系统表结构

### 表之表 RelCat

- relName：表名。
- tupleLength：单条记录大小。
- attrCount：属性个数。
- indexCount：下一新建索引编号。

### 属性之表 AttrCat

- relName：表名。
- attrName：属性名。
- offset：属性相对于记录数据指针的偏移。
- attrType：属性类型。
- indexNo：属性对应的索引编号 (无索引时为 -1，否则 indexNo 对应非 NULL 值的索引编号，indexNo + 1 对应 NULL 值的索引编号)。
- isNotNull：属性是否要求非空。
- primaryKey：属性是第几个主键 (非主键则为 0)。
- refrel：外键表名 (非外键则为 "")。
- refattr：外键属性名 (非外键则为 "")。

## 查询处理模块

查询处理模块主要用于完成 DML 语句对应的操作，核心是基于记录管理模块与索引模块实现数据表的增删查改功能。

### 本模块代码说明

- `ql.h`：查询处理模块声明。
- `ql_error.cc`：查询处理模块错误输出。
- `ql_manager.cc`：查询处理器，提供查询、聚集查询、分组聚集查询、插入、删除、更新的接口。

## 命令解析模块

命令解析模块主要用于解析用户在命令行界面输入的 DDL 与 DML 语句，并调用对应的系统管理模块与查询处理模块的接口完成操作。

在框架中提供了 Redbase 项目的命令解析模块，但由于本项目的 SQL 语句与 Redbase 项目差别较大，我们在沿用其基本组织方式与一些通用工具函数的基础上，进行了较大的改动。

### 本模块代码说明

- `scanhelp.c`：词法分析工具函数，被词法分析器直接包含，用于为字面量分配内存、标识符转小写、区分关键字

等。

- `scan.l`：词法分析描述文件，供 flex 生成词法分析器，主要任务是解析出所有终结符 (字面量、关键字、标识符)。
- `parser_internal.h`：抽象语法树结点声明。
- `nodes.c`：抽象语法树结点定义，隐去了语法分析中对语义无影响的部分。
- `parse.y`：语法分析描述文件，供 bison 生成语法分析器，主要任务是建立抽象语法树。
- `parser.h`：解析结果类型声明，作为系统管理模块与查询处理模块的接口参数使用。
- `interp.c`：语义分析器，遍历抽象语法树产生解析结果类型，调用系统管理模块与查询处理模块的接口完成实际的语义动作。

## 附加功能实现

### 扩展数据类型

为统一起见，直接对所支持的 4 种数据类型进行介绍。

- `int`：定长为 4 字节。
- `varchar`：定长为建表时规定的长度上限，内部以 `\0` 作为字符串真正的结尾。
- `float`：定长为 4 字节，需要注意的是 `float` 允许接收 `int` 字面量。
- `date`：定长为 11 字节，以字符串形式存储，需要注意的是 `date` 一定接收的是 `varchar` 字面量，并需要对日期合法性进行判断。

### 外键约束

实现外键约束主要是进行以下两个判断：

- 在插入记录时，对于插入的外键，判断其对应的表的属性中是否存在该值，如不存在则报错。
- 在更新记录时，对于更新的外键，判断其对应的表的属性中是否存在该值，如不存在则报错。

### 聚集查询

除命令解析模块的支持外，利用与查询处理相似的方法获得查询 (连接) 结果，然后按照聚集函数的需求计算出聚集查询结果。

### 分组聚集查询

除命令解析模块的支持外，利用与查询处理相似的方法获得查询 (连接) 结果，依据分组条件分组，然后按照与聚集查询类似的方式计算出分组聚集查询结果。

### 模糊查询

除命令解析模块的支持外，还增加了模糊比较运算符，其内部实现为一个动态规划。

对于原始模式串 *originalPattern* 和文本串 *text*，首先将原始模式串中的 `%` 变为特殊功能码 `\1`，`_` 变为特殊功能码 `\2`，转义字符 `\<char>` 变为真实字符 `<char>`，从而产生真正的模式串 *pattern*，各字符只占一个字节以方便动态规划的进行。

令  $possible(i, j)$  表示  $pattern[1..i]$  与  $text[1..j]$  是否可能匹配，则初始条件为

$$possible(0, j) = \begin{cases} true & j = 0 \\ false & others \end{cases}$$

$i > 0$  时,

$$possible(i, 0) = \begin{cases} possible(i-1, 0) & pattern[i] = 1 \\ false & others \end{cases}$$

$i > 0, j > 0$  时,

$$possible(i, j) = \begin{cases} possible(i-1, j) \vee possible(i-1, j-1) \vee possible(i, j-1) & pattern[i] = 1 \\ possible(i-1, j-1) & pattern[i] = 2 \vee pattern[i] = text[j] \\ false & others \end{cases}$$

## 多表连接与查询优化

查询优化分为两步：获得单表查询结果、获得多表连接结果。

### 单表查询

首先遍历所有查询条件，查找具有如下特征的条件：

- 条件右侧为字面值。
- 条件左侧属性已经建立索引。
- 条件操作符适合索引查询 (比如“相等”)。

如果没有发现满足上述要求的条件，使用记录管理模块提供的扫描器扫描整个数据表获取单表查询结果。

如果发现存在满足上述要求的查询条件，选择最适合的条件进行索引扫描，并且使用其他条件对扫描结果筛选得到单表查询结果。

### 多表连接

优先连接具有连接条件的单表查询结果，降低临时数据集的规模。然后暴力连接其余单表查询结果。

## 小组分工

这里仅标明相应模块的主要负责人。

1. 记录管理模块：徐毅
2. 索引模块：阎世宏
3. 系统管理模块：徐毅
4. 查询处理模块：阎世宏
5. 命令解析模块：徐毅

## 总结收获

1. 除去框架文件，整个项目代码量近 9000 行，大小近 300 KB，伴随了我们整整一个学期，对我们的代码能力是一种

极大的锻炼。

2. 最初只是依次实现一个个模块，感觉还不是太明显，但当最后合在一起成功跑起来这个数据库的时候，成就感还是爆棚的。
3. 构建一个鲁棒的系统是相当不容易的，需要进行各种类型的大量的测试。
4. 总体收获颇丰，希望这门课越办越好。

## 参考文献

---

[1] Stanford CS 346 课程主页. <https://web.stanford.edu/class/cs346/2015/>

[2] Stanford CS 346 Redbase 项目的框架文件来源 (切换到 initial commit 等可获取). <https://github.com/junkumar/redbase>