

# 数据库系统概论：大作业

授课教师：冯建华 马冬哲

截至日期：2012.1.13

Group30 计 92 肖桐 毛佳昕 褥震 金旻俊

## 整体框架

整个工程主要由以下四部分组成，可以用图 1 来表示整体的框架。

### 1. Column 类

对于数据库的每个列信息的一个封装，包括列名、类型、索引等信息，以及根据索引做屏蔽等函数。

### 2. Table 类

对与数据库的每个表信息的一个封装，包括表名、每行的信息、每列的信息，以及给定条件做筛选等函数。

### 3. SQLParser 类

用于 SQL 语句解析的类，提取需要的信息。

### 4. Executor 类

实现逻辑控制的类，执行 client 里面应做的事情。

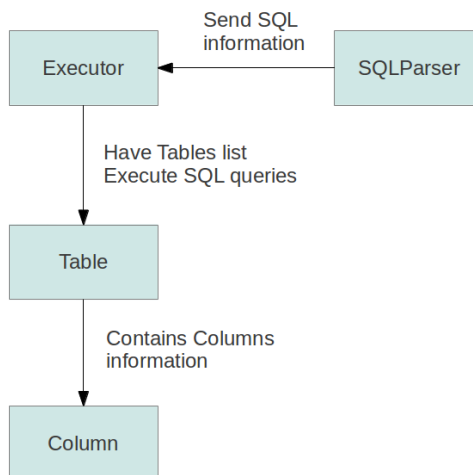


Figure 1: 工程整体框架示意图

## 数据存储机制

### 1. 行内容存储

我们组采用了 kyotocabinet 的 TreeDB 来管理数据库每行内容的存储，为方便定位查找，每行的信息以定长的字节串的形式存储。每个整型的列，用 4 字节来表示其值；每个字符串型的列，用其声明长度加 1 来记录（加 1 是为了在末尾加入字符串结束符），长度不足的话，其结束符之后的字节随意填充。例如，对于这样的一个表：

```
tableA
colA    INTEGER
colB    VARCHAR(5)
colC    INTEGER
```

若加入这样一行

```
255, 'abcde', 1
```

则存入的记录如下，其中整数是以小端的形式存储的

ff	00	00	00	'a'	'b'	'c'	'd'	'e'	'\0'	80	00	00	00
----	----	----	----	-----	-----	-----	-----	-----	------	----	----	----	----

### 2. 索引存储

此外，我们会对在 train 中出现在条件中的列做索引，原本索引是用 HashDB 来管理的，但由于其效率实在难以忍受，后来改为都存入内存中，用 multimap 作索引。我们对于每行用其行号作为唯一的隐藏主键。在 load 和 insert 时，即会对每一行需要做索引的列添加索引。

## 查询的算法

1. 解析 SQL 的 SELECT 语句，计算出所需要的行列信息。
2. 每个 SELECT 语句都可以表示为若干个表的 join，这可以用一棵树表示，同时对每个节点会有一些限制条件。根据上一步得出的行列信息构建这样一棵树。
3. 计算出一个 join 的顺序，即给出一个树的遍历顺序。这里用到了一个简单的估价，对于每个节点估计符合其限制条件的行数。对于形如 "colA = 1" 这样的条件，估价函数为  $\frac{totalRows}{diff\ keys}$ ；对于形如 "colA < 100" 这样的条件，估价函数为  $totalRows \frac{key - minKey}{maxKey - minKey}$ 。估价越低，join 的顺序越靠前。
4. 根据生成的 join 顺序依次做 join。join 的中间结果保存的是每个表符合条件的行号，每次 join 先对要 join 的表用其限制条件做筛选，返回一个集合。然后枚举中间结果，用值去找要 join 进来的表的索引，又得到一个集合。两集合的交即加入中间结果中。
5. 我们经过一些实验，对某个表做筛选采用了如下的方法。用上面提到过的估价函数对限制条件做从小到大的排序，然后取出第一个，利用索引得到满足条件的行号。接下来对于每行，从原表中取出行的内容，依次判断是否符合剩下的限制条件。如果都满足，则加入筛选结果中。



Figure 2: 查询的算法流程

## 其他

1. 参考书籍：数据库系统与实现，Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom。
2. 分工情况

小组成员贡献率	
肖桐	30%
毛佳昕	30%
褥震	20%
金旻俊	20%

3. 一些建议：如果可以的话，明年的大作业希望做成一个类似 oj 的形式，这样可能会给教学双方都减少些压力。