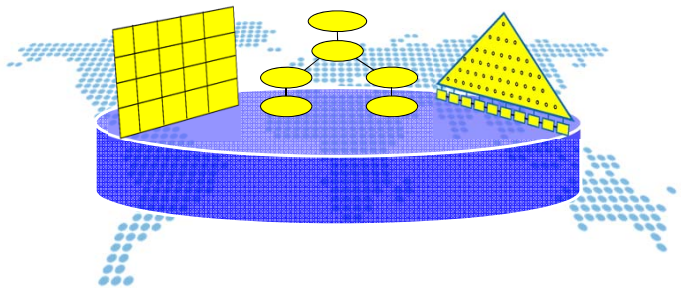


数据库系统

LAB3: 数据库原型实现

陈世敏

(中科院计算所)



实验3安排

- 内容: 在一个原型内存数据库系统AimDB上实现select查询处理功能
- 掌握
 - 多种关系运算的查询处理实现
- 本堂课: 11月29日
 - 实验3具体要求
 - AimDB介绍、编程环境说明 (刘刚)
- 1月10日: 验收实验3
- 1月17日: 期末考试

AimDB

- An implementation of main memory DataBase
- 我们自己实现的一个简单的内存数据库原型
 - C/C++实现
 - Database, Table, Index的定义
 - Table在内存中的存储
 - Hash index, B+-Tree index
 - 数据类型和操作的支持: 每个类型有一个class
- 预留了查询处理部分
 - 预留的调用接口
 - 定义了输入参数和返回值类型
 - 要求实验3实现查询处理部分

实验3: 内存数据库中查询处理实现

- 要求: 支持Select语句的基础功能
 - 选择: 过滤条件
 - 投影: 提取部分列
 - 连接: 找匹配的记录
 - 分组聚集: 进行统计运算
 - 排序
- 只实现Select的一个子集
- 不需要对SQL语句解析, Select的内容将会形成一个C的结构体, 是查询处理的调用参数
- 需要实现的是查询处理的功能

Select语句的子句

SELECT ...

FROM ...

WHERE ...

GROUP BY ...

HAVING ...

ORDER BY ...

每个子句中，最多出现4个子项

需求1: TableScan Operator

- 实现一个扫描Table的Operator
- 顺序访问给定Table的所有tuple

需求2: Where条件

- 最多有4个条件
- 条件之间是AND关系
 - (条件1) AND (条件2) ...
- 两种形式的条件
 - 列名 op 常量
 - 记录在单独某一行上的属性值与一个常量进行比较
 - 列名1 op 列名2
 - 连接条件
- 可以使用数据类型的相关操作计算单个条件

需求3: FilterProject Operator

- 下层输入是tuple
- 对tuple实现过滤
- 并对满足过滤条件的tuple, 进行投影
- 产生结果记录

需求4: From子句和2种Join Operator

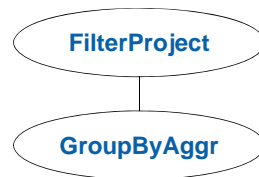
- 最多4个Table
- 连接条件在where子句中给定
- Sort Merge Join Operator
 - 当join key上没有index时
 - 采用Sort Merge Join实现
- IndexNestedLoopJoin Operator
 - 当至少一个表的join key上有index时
 - 直接用Index Nested Loop Join实现

需求5: Group By + Aggregation

- 实现GroupByAggr Operator
 - Group By的分组组别最多可以有4个列
 - Aggregation最多在4个列上计算
 - SUM, AVG, COUNT, MAX, MIN
 - 采用基于Hash的方式实现
 - 分组组别列→Hash table的key
 - Hash table的value是一个指针, 指向分组组别列和Aggregation的中间结果

需求6: 在Group By之后可以有Having

- 构造query operator tree时, 允许在GroupByAggr上有FilterProject



需求7: OrderBy Operator

- 排序的键最多包含4个列
- 全部是从小到大顺序
- 采用quick sort算法实现

需求8: Select子句输出

- 最多有4个列
- 有可能是Aggregation
 - sum, count, avg, min, max

需求9: 采用Tuple-at-a-time方式计算

- 每个Operator实现
 - open
 - getNext
 - close
- 根据Select的内容, 构造Operator Tree
- 确定具体的初始化参数

提示

- AimDB提供了hash table的class
- 可以用libc中的quicksort来实现排序
 - 也可以自己写quicksort算法

评分标准

- 总分: 20分
 - 代码和注释: 3分
 - 注释需要对每个函数的实现进行说明
 - 是否确实实现了上述规定的Operator
 - 采用doxygen规定的格式 (<https://en.wikipedia.org/wiki/Doxygen>)
 - 并用doxygen产生pdf文档, 包括调用关系图
 - select语句: 17分
 - 我们将产生17个测试, 每一个测试占1分
 - 选择投影: 4分
 - 连接: 7分
 - 分组聚集: 4分
 - 排序: 2分
 - 提交代码, 进行抄袭相似性检查
 - 如果发现抄袭, 那么0分, 并通知教务处
 - 我们验收时会给定输入, 检查输出是否正确