

Record Manager 详细设计报告

张小爽(3110104779)

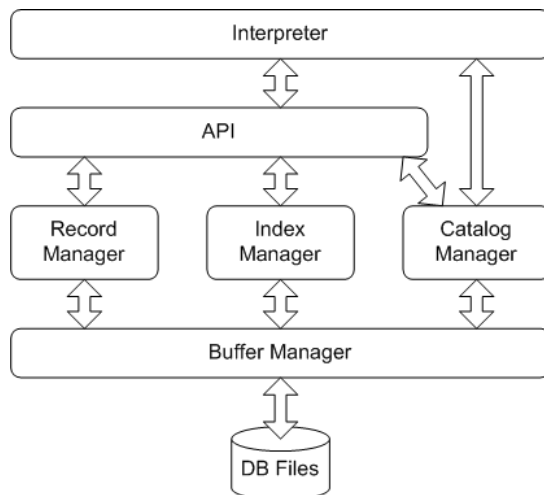
一、模块概述

Record Manager 负责管理记录表中后缀为“.table”的数据文件。数据文件由一个或多个数据块组成，块大小应与缓冲区块大小相同。一个块中包含一条至多条记录，为简单起见，只支持定长记录的存储，且不支持记录的跨块存储。

1. 主要功能如下：

- 实现数据文件的创建与删除
- 记录的插入
- 记录的删除。分为有 index 和没有 index（遍历）两种删除模式。
- 记录的查找。分为有 index 和没有 index（遍历）两种查找模式，能够支持不带条件的查找和多个条件的查找（包括等值查找、不等值查找和区间查找）。

2. 模块在整体架构中的位置



Record Manager 向上对 API 提供相应的调用接口，向下通过调用 Buffer Manager 的函数，实现与 Buffer 中块信息的交互。

二、实现原理

通过 API 调用以下模块，实现数据管理器对数据的不同处理功能，分别如下：

1. 数据文件创建/删除

- 创建：bool createTable(const string& tableName, const vector<Attribute>& vtAttr);
传入表名以及所有属性的信息，根据属性信息得到每行记录的大小；函数中调用 buffer 的函数，写入一个空块，设置块中每行记录的标记位为 NOT_VALID，表示可写（或已删除）。
- 删除：bool RecordMan::dropTable(const std::string& tableName);

传入表名，调用buffer中remove()函数，在DB files中删除该表格数据、索引、目录信息。

2. 记录插入: `int insertTuple(Table table, std::vector<std::string>& tuple);`
传入表信息和一行记录（按属性排列顺序构建的VECTOR<STRING>），判断表中是否包含UNIQUE的属性，若包含，则调用函数`bool RecordMan::hasExisted(Table table, string content, int num, int blockNum)`判断是否冲突，不冲突则按次序遍历读取文件的块，找到块中第一个标记位为NOT_VALID的位置，修改信息标记位为VALID，表示数据有效，并调用buffer中的write()函数，将新的块数据信息写入buffer。
返回该条记录在文件中的偏移，供API判断是否修改INDEX等其他信息。
3. 记录删除:
 - 有 index 的删除
`vector<pair<Tuple,int>> deleteTuple(const Table& table,const vector<Condition>& condition, vector<int> m_offset)`
传入表信息和选择判断条件的vector，以及index做好的数据偏移量。遍历所有offset，得到一行记录的内容，调用`isValid(size,substring)`判断数据是否为有效数据，调用`tmp->isSatisfied(table, condition)`判断是否满足参数condition。若满足，则将该条记录信息和其偏移量做成pair返回。
 - 无 index 的删除
`vector<pair<Tuple,int>> deleteTuple(const Table& table,const vector<Condition>& condition, int blockNum);`
传入表信息和选择判断条件的vector，以及块号。遍历块中所有记录，调用`isValid(size,substring)`判断每行记录是否为有效数据，调用`tmp->isSatisfied(table, condition)`判断是否满足参数condition。若满足，则将该条记录信息和其偏移量做成pair返回。
4. 记录查找
 - 有 index 的查找
`vector<Tuple> selectTuple(const Table& table,const vector<Condition>& condition, vector<int> m_offset);`
传入表信息和选择判断条件的vector，以及index做好的数据偏移量。遍历所有offset，得到一行记录的内容，调用`isValid(size,substring)`判断数据是否为有效数据，调用`tmp->isSatisfied(table, condition)`判断是否满足参数condition。若满足，则将该条记录信息压入VECTOR<TUPLE>，最后返回。
 - 无 index 的查找
`vector<Tuple> selectTuple(const Table& table,const vector<Condition>& condition, int blockNum);`
传入表信息和选择判断条件的vector，以及块号。遍历块中所有记录，调用`isValid(size,substring)`判断每行记录是否为有效数据，调用`tmp->isSatisfied(table, condition)`判断是否满足参数condition。若满足，则将该条记录信息压入VECTOR<TUPLE>，最后返回。
5. 获得表所有记录的偏移量

传入table信息, attribute名, 以及块号, 通过buffer取得块内容, 遍历块中所有记录, 调用isValid(size, substring)判断每行记录是否为有效数据, 若有效, 则将属性和其偏移量做成pair返回。

Record files（数据文件）是 DB files 的一部分。Record 数据记录存放的规则如下：

- 例如，下图所示为表 **student** 信息在一个 **block** 中的存放示意图，中括号里的数字表示该属性所占空间：

Stu1.name[20]	Stu1.age[4]	Stu1.score[4]	Dirty[1]	Stu2.name[20]	Stu2.age[4]	Stu2.score[4]
Dirty[1]	Stu3.name[20]	Stu3.age[4]	Stu3.score[4]	Dirty[1]
.....						