

CatalogManager

蔡灿宇

主要功能

- 负责管理数据库的所有模式信息，包括：
 - 负责管理数据库的所有模式信息，包括表的名称、表中字段（列）数、主键、定义在该表上的索引。
 - 表中每个字段的定义信息，包括字段类型、是否唯一等。
 - 数据库中所有索引的定义，包括所属表、索引建立在哪个字段上等。
- 提供访问及操作上述信息的接口。

接口说明

API:

```
void createTable(Table& table);           //创建表格
void createIndex(Index index);           //创建索引
void dropTable(Table table);             //删除表格
void dropIndex(Index index);             //删除索引
```

- 本模块提供接口给表格、索引的建立和删除的信息存储。
 - createTable(Table& table);**
首先判断该表格是否已存在，若存在则返回重名错误信息，若不存在则将interpreter输入的table信息存入catalog。
 - createIndex(Index index);**
首先判断索引所在表格是否存在，若不存在则返回表格不存在错误信息；然后判断该索引是否已存在，若存在则返回重名错误信息。一切正常后将interpreter输入的index信息存入catalog。
 - dropTable(Table table);**
首先判断是否存在改表格，若不存在则返回表格不存在错误信息；若存在则将该表格从catalog信息中删除，顺便删除在这个表格上定义的所有索引。
 - dropIndex(Index index);**
首先判断是否存在改表格，若不存在则返回表格不存在错误信息；随后判断该表格中是否已存在该索引，若不存在则返回索引不存在错误信息；若存在则将该索引从catalog信息中删除。

模块说明

- 类定义：

```
class CatalogManager
{
```

```

private:

    int tableNum;
    vector<Table> Tables;           //用于存储table信息
    int indexNum;
    vector<Index> Indexes;         //用于存储index信息

    void readTable();              //从文件中读取信息存入Tables容器
    void readIndex();              //从文件中读取信息存入Indexed容器
    void writeTable();              //将Tables容器里的信息存入文件
    void writeIndex();              //将Indexes容器里的信息存入文件

public:

    CatalogManager()                //构造函数，将文件内容读取进容器
    ~CatalogManager()              //析构函数，将容器内容写入文件

    void clear();                   //清空容器
    void reload();                  //重新刷新容器内容
    void createTable(Table& table); //创建表格
    void createIndex(Index index);  //创建索引
    void dropTable(Table table);    //删除表格
    void dropIndex(Index index);    //删除索引
    bool ExistTable(string table_name); //判断是否已有该表格
    bool ExistIndex(string indexname); //判断是否已有该索引
    Table getTable_info(string table_name); //返回表格信息
    Index getIndex_info(string index_name); //返回索引信息
    int GetColumnNumber(Table& table, string columnname); //返回属性所在位置
    int GetColumnAmount(Table& table); //返回表格的属性数量
};

```

- 工作原理：
 - 构造CatalogManager时，将文件中保存的表格和索引信息读入容器中，方便操作。
 - 直接操作对象是保存信息的容器。
 - 析构函数容器中的信息重新写回磁盘文件，达到保存文件的目的。
 - reload()函数用于可选择的随时更新磁盘文件，方便select等操作。
- 函数说明：
 - **ExistTable(string table_name)**
以表格名为依据判断该表格是否已经存在。
 - **ExistIndex(string indexname)**
以索引名为依据判断该索引是否已经存在
 - **Table getTable_info(string table_name)**
以表格名为参数，返回保存在Tables容器中的表格信息，以便后续条件查找。
 - **Index getIndex_info(string index_name)**
以索引名为参数，返回保存在Indexes容器中的索引信息，以便后续条件查找。
 - **int GetColumnNumber(Table& table, string columnname)**
查询该属性在表格中所在位置，以便将位置存入Condition中，方便后续select等其他接口的操作。
 - **int GetColumnAmount(Table& table)**
返回该表格中的属性数量。

- **void reload()**

将现有容器里的信息存入硬盘，然后将容器清空，再将硬盘中的信息重新读入容器中，达到可选择地更新磁盘与容器中的信息的功能，避免出现create table之后table信息还没存入磁盘的情况。