

# CatalogManager 设计报告

刘圣源

我们的 CatalogManager 被整合在了 RecordManager.h 中。Catalog Manager 记录了表及其属性的信息和索引的信息，并提供了大量接口以方便其他模块使用。

本部分将介绍 CatalogManager 对外提供的接口：

- addIndex() 将在储存索引信息的文件中添加新的索引的信息（即函数参数列表）。同时会调用另一成员函数 setIndexOnAttribute()，在储存对应的 table 信息的文件中，修改相应属性的信息，声明该属性将获得一个索引。（尚未写入文件，内存中处理，但声明了该部分最近被修改过，需要写入文件）

```
int addIndex(string indexName, string tableName, string
attributeName, int type);
```

- dropIndex() 在储存索引信息的文件中，将特定的索引信息用其后的索引覆盖掉，亦即删除该索引。然后调用 revokeIndexOnAttribute() 成员函数，在储存 table 信息的文件中，将该属性关于该索引的信息删除。

（同理，这些操作都是在内存中处理的，文件内容还没有修改，但是已经通过一个参数告知程序，该部分内容需要重新写在文件中，之后函数均是同理，不再赘述）

```
int dropIndex(string index);
```

- setIndexOnAttribute()，正如之前提到，修改相应文件中的信息，声明了某个属性将获得一个索引。

revokeIndexOnAttribute() 则是 setIndexOnAttribute() 的反操作（由函数声明易知，两函数有相同的参数列表）。

```
int setIndexOnAttribute(string tableName, string AttributeName,
string indexName);
int revokeIndexOnAttribute(string tableName, string AttributeName,
string indexName);
```

- addTable() 与 addIndex() 类似，创建了储存 table 信息的文件，其中包括该 table 中 record 的数量、primarykey 的位置、属性数量和各属性的信息。

```
int addTable(string tableName, vector<Attribute>* attributeVector,  
string primaryKeyName, int primaryKeyLocation);
```

- dropTable() 即将储存特定 table 信息的文件删除。

```
int dropTable(string tableName);
```

- findTable() 和 findIndex() 作用类似，即判断相应的 table (index) 是否已经被建立，返回的 int 值可以修改为 bool 值。

```
int findTable(string tableName);  
int findIndex(string indexName);
```

- deleteValue() 和 insertRecord() 也是互为反操作，两个函数均是对一个表中关于 record 数量的信息进行修改，前者减后者增。（具体关于记录内容的增删不在 CatalogManager 的讨论范围内）

```
int deleteValue(string tableName, int deleteNum);  
int insertRecord(string tableName, int recordNum);
```

- attributeGet() 和 indexNameListGet() 两个函数，会将特定 table 关于属性的全部信息或该 table 相关的全部索引名写进函数参数中的 vector 容器指针所指的 vector 容器内。

```
int attributeGet(string tableName, vector<Attribute>*  
attributeVector);  
int indexNameListGet(string tableName, vector<string>*  
indexNameVector);
```

- getRecordNum() 将返回特定 table 中已经包含的 record 数量。

```
int getRecordNum(string tableName);
```

• getAllIndex() 则是将存储索引的文件中的全部关于索引的信息，而前面提到的 indexNameListGet() 函数只读取函数名。

```
int getAllIndex(vector<IndexInfo> * indexs);
```

• getIndexType() 返回该索引对应属性的类型，用特定数字代表了特定类型，因此返回值为 int 类型。

```
int getIndexType(string indexName);
```

• calcuteLenth() 接受一个 table 的名字为参数，返回该 table 的一个 record 将占的空间大小。

```
int calcuteLenth(string tableName);
```

• calcuteLenth2() 将返回某个类型的大小，即 sizeof(int)、sizeof(float) 等。函数的参数 type 为 int 类型，即前面提到的用特定数字代表属性类型。

```
int calcuteLenth2(int type);
```

• recordStringGet() 函数，将一个 record 的内容（即参数中的 vector 容器指针所指内容）连接成一个字符串，赋给 recordResult。方便写进文件中。

```
void recordStringGet(string tableName, vector<string>*  
recordContent, char* recordResult);
```