**3.3 Catalog Manager**

**3.3.1模块简介**

Catalog Manager负责管理数据库的所有模式信息，包括：

1. 数据库中所有表的定义信息，包括表的名称、表中字段（列）数、主键、定义在该表上的索引。
2. 表中每个字段的定义信息，包括字段类型、是否唯一等。
3. 数据库中所有索引的定义，包括所属表、索引建立在那个字段上等。

Catalog Manager还必需提供访问及操作上述信息的接口，供Interpreter和API模块使用。

在写模块的过程中，我将函数CreateIndex、CreateTable、DropIndex、DropIndex作为接口和其他模块相连接。在创建索引或者表之前我们需要先给它们各自分配ID，以便于区分调用。所以当创建索引和表时，要先确定它们的名称是独一无二的，再给它们分配ID。创建成功后，则将相关信息写入“tables.bin”、“tableID.bin”、“indices.bin”、“indexID.bin”，当删除索引和表时，则需要删除这些文件内的信息。此处还用到了buffermanager和intlistmanip中的某些函数。

**3.3.2声明**

头文件：

#ifndef CATALOGMANAGER\_H

#define CATALOGMANAGER\_H

#include "structs.h"

C文件

#include "catalogmanager.h"

#include "intlistmanip.h"

#include "buffermanager.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

**3.3.3函数调用**

所有的函数都在头文件内声明

|  |
| --- |
| #ifndef CATALOGMANAGER\_H  #define CATALOGMANAGER\_H  #include "structs.h"  int GetIID(char\* name);  int AllocateIID(char\* name);  void WriteIMeta(int IID, short TID, short AID, char fixed);  char GetAID(int TID, char\* name);  int DeleteIMeta(int IID);  void DeleteIIDMeta(int IID);  void DropIndex(char\* in);  int ifexist(char\* in, char\* tn, char\* an, char force);  void CreateIndex(char\* in, char\* tn, char\* an, char force);  int FindIID(int TID, int AID);  int GetTID(char\* name);  int AllocateTID(char\* name);  void WriteTMeta(short TID, MetaTable T);  void CreateTable(MetaTable T);  void DeleteTIDMeta(short TID);  void DeleteTMeta(short TID);  void DropTable(char\* tn);  void DroptheTable(char\* tn,int TID);  void GetTable(short TID, char\* table); |

各函数说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类型 | 名称 | 输入 | 输出 | 调用 | 被调用 | 作用 |
| int | GetIID | Char\* name | -1/相同名称索引 |  | DropIndex  CreateIndex  ifexist | 是否有此索引，没有则返回-1 |
| Int | AllocateIID | Char\* name | 最小可用IID | Read  Append  （buffermanager.h） | CreateIndex | 找出最小可用的IID |
| Void | WriteIMeta | int IID,  short TID, short AID, char fixed |  | Append | CreateIndex | 将索引ID，表ID，属性ID，写入indices.bin最高位为1则不可删除 |
| char | GetAID | int TID,  char\* name | -1或者 AID | Read | CreateIndex  ifexist | 确保TID正确，传入属性名称，查找。找到返回AID，没找到返回-1 |
| int | DeleteIMeta | int IID | 删除成功返回1，失败返回-1，如果试图删除\_PK\_开头的时候提示 | Read  DeleteLine  （buffermanager.h）  GetFileSize11（filemanip.c） | DropIndex | indices.bin中删除给定IID对应的索引信息 |
| void | DeleteIIDMeta | int IID |  | Read  （buffermanager.h）  DeleteLine  （filemanip.c） | DropIndex | 在indexID.bin中删除给定IID对应的索引的名称-ID映射 |
| void | DropIndex | char\* in |  | GetIID  DeleteIMeta  DeleteIIDMeta |  | 删除索引 |
| Int | ifexist | char\* in, char\* tn, char\* an, char force | 1/-1 | GetIID  GetTID  GetAID  FindIID | CreateIndex |  |
| void | CreateIndex | char\* in, char\* tn, char\* an, char force |  | ifexist  GetIID  GetTID  GetAID  AllocateIID  WriteIMeta  GetTable  IntListInsert  ReWrite  PVoidListInsert  BpBuild  IntListDestroy  PVoidListDestroy | CreateTable | 依照输入的信息创建索引 |
| Int | FindIID | int TID  int AID | 索引ID或者-1 | Read | ifexist | 传入表ID、属性ID，判断此属性是否存在索引，若存在返回索引ID。否则返回-1。 |
| Int | GetTID | char\* name | 表ID或者-1 |  | CreateIndex  DropTable  CreateTable  ifexist | 传入表名称，查找是否存在此表，如果存在返回表ID。否则返回-1。 |
| int | AllocateTID | char\* name | 最小可用TID或者-1 | Append  Read | CreateTable | 找出最小可用表的ID |
| Void | WriteTMeta | short TID, MetaTable T |  |  | CreateTable | 将表的TID、属性、类型等相关信息写入"tables.bin" |
| Void | CreateTable | MetaTable T |  | GetTID  AllocateTID  WriteTMeta  CreateIndex |  | 根据输入的相关信息创建一张表，并创建完后立即在tables.bin中写入，然后对主键创建索引 |
| Void | DeleteTIDMeta | short TID |  | Read  DeleteLine | DropTable | 根据TID在tableID.bin中删除表格数据 |
| void | DeleteTMeta | short TID |  | Read  DeleteLine | DropTable | 根据TID在tables.bin中删除数据 |
| void | DropTable | char\* tn |  | GetTID  DeleteTMeta  DeleteTIDMeta  DroptheTable |  | 判断是否符合删除表的条件，若符合则调用删除表的函数 |
| void | DroptheTable |  |  | Read  IntListInsert  DeleteLines  IntListDestroy | DropTable | 删除表以及相关联系 |
| Void | GetTable | short TID, char\* table |  | Read | CreateIndex | 在“tables.bin”中找到TID与表名所代表的行 |

注：WriteTMeta的前四个项的大小：TID-4bit；pk还是unique- 2bit； type-2bit;；size-1B； name-30B。

函数调用图：

