

DESIGN OF A SIMPLE STORAGE MANAGER

CMPE 321

Bekir Burak ASLAN

2016400330

PROJECT #1

CONTENT

1. INTRODUCTION

1.1. What is DBMS?

1.2. What is our aim?

2. ASSUPMTION

3.DATA STRUCTURES

3.1.SystemCatalog

3.2.File Structure

3.3.Page Structure

3.4.Record Structure

4.ALGORITHMS

4.1.FILE OPERATION

4.1.1.Create a Type

4.1.2.Delete a Types

4.1.3.List all Types

4.2.RECORD OPERATIONS

4.2.1.Create a Record

4.2.2.Delete a Record

4.2.3.Update a Record

4.2.4.Search a Record

4.2.5.List all Records

5.CONCLUSION

1.INTRODUCTION

1.1.What is DBMS?

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is hence a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.(Fundamentals.of.Database.Systems.6th.Edition)

1.2.What is our aim?

The aim is to keep the records sequentially. A system between the record enables us to find the record more easily and quickly. In this project, I have designed a simple storage manager. This project can do the these operations which I show them below.

FILE Operations:

- Create a type (Field of the Type is determined by user).
- Delete a type .
- List all files.

Record Operations:

- Create a record in a specified type.
- Delete a record according to the primary key from a specified type.
- Update a record with a given primary key.
- Search for a record according to the primary key field.
- List all records in a specified type.

2.ASSUMPTION

- Page Size is 1Kbytes.
- All records cannot exceed a page size.
- File names and the Field name can be at most 10 characters and at least 2 characters
- Each string characters are 1 byte.
- Each page has a unique ID.
- If a page has been filled, I need to create a new page.
- Every record has a unique ID.
- The first field of a record is the primary key of the record which is given automatically. There is a unique primary key for each type.
- Primary Key field cannot have any repeated value (Every Primary key is different).

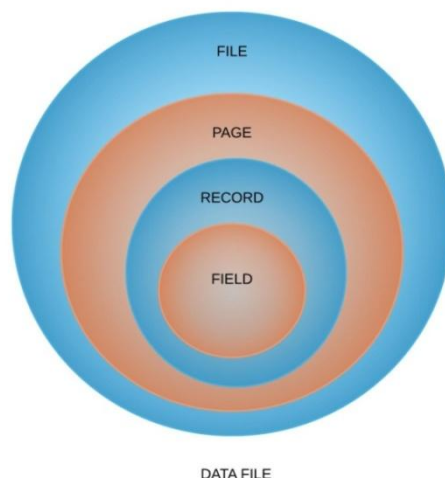
- There can be more than one records according to their sizes in one page, but one record can be in only one page.
- Type names will be shorter than 10 characters.
- System gives the keys automatically but user don't know the key. So Primary Keys will be showed automatically before updating and searching a record
- Fixed length record is used.
- Each record have 10 fields and each fields have 20 character length.
- Each record name length is between 2 and 10 characters.
- The most important assumption is System assume that user always enter valid data.
- There is an islast flag which used for page is empty or not.
- There is an isempty flag which used for record is empty or not. Generally this flag useful when user delete a record in the middle of the page.

3.DATA STRUCTURE

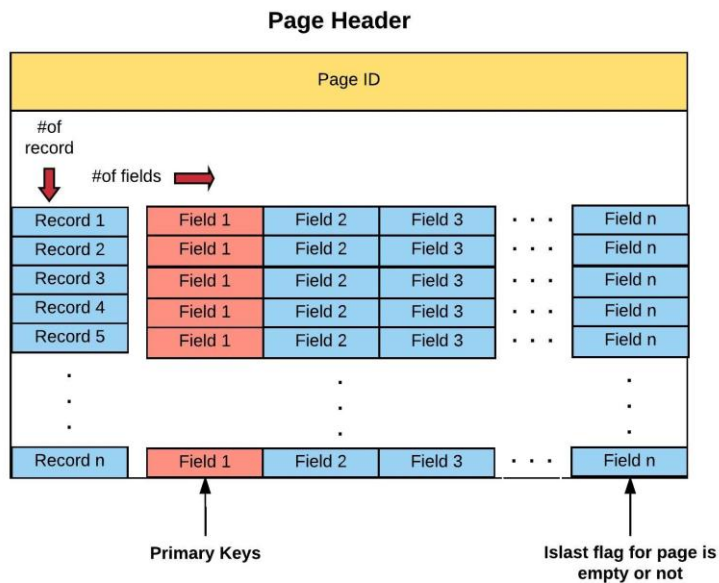
3.1.System Catalog

This part is the one of the important part of the DB management system because all general DB system and the pages are in this file. Another important thing for System Catalog is reachable for anytime. System Catalog keeps the information about files which show them below.

- File names
- Number of pages
- Number of records
- Number of fields
- Total size of records(fixed size)
- Primary key

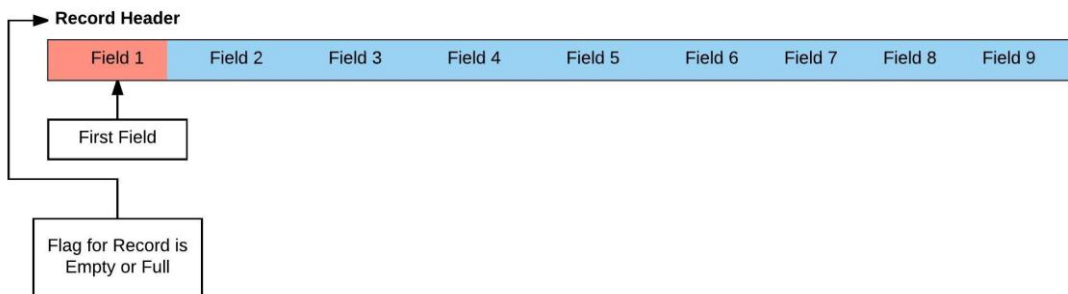


- Number of records and fields.
- First field is primary key.
- Flag for empty records.
- There is a number of record.



3.4.Record Structure

- Each record have maximum 10 fields and each field can be filled with maximum 20 characters.
- First field(field 0) is primary key.
- There is a flag for record is full or empty.
- Each record have 10 fields and each fields have 20 character length.



4.ALGORITHMS

1.File Operations

4.1.1.Create a new Type

```
CreateType{
get.TypeName //user insert Type Name
open(System.Catalog.dat);
    CreateNewType(TypeName) //User create a new Type
    increase #OfType; //System will use # as a counter
    get.FieldNumber;
    //user determine how many fields will be inserted
    for(int i;i<=FieldNumber;i++)
    //System can reach and save fields with for loop
    {
        CreateNewType(FieldName);
    }
    close(System.Catalog.dat);
}
```

4.1.2.Delete a Type

```
deletetype{
get NameOfType;
get #OfType;
open(System.Cat.dat);
    integer i=0;
    for(i=0;i<=#OfType;i++) //search the typeName with for loop
    {
        if(find(TypeOfName)) //System find the TypeName which is inserted by user
        {
            delete Type(NameOfType);
            decrease #of Type; //this is important for listing all types
        }
        }endif
    close(System.Cat.dat);
}
```

4.1.3.List all Types

```
ListAllRecords{
open(System.catalog.dat);
get # ofTypesFrom System.catalog.dat;
integer counter=#ofTypes;
//System count the created or deleted types with a counter;
    integer i=0;
    //System listed all types with loop
    for(i=0;i<counter;i++){
        Write NameOfTypes[i];
    }
    close(System.cat.dat);
}
```

2.Record Operations

4.2.1.Create a Record

```
CreateNewRecord{
gettypeName;
open(System.Catalog.txt);
    open(typeName)
    while(ID!=LastPageID)//System search page by page
    {if(check islast=true)
    //System check the page is completely filled or not with flag
    //If there is a deleted record in the page before the insertion,I will add my record in this
page
        integer i;
        while(!=EOF)
        {
            if(check(isEmpty=true))
            //check which record is deleted and its space is free
            {
                add NewrecordToThatAddress;
                set isEmpty=false;//I set my flag false because this space is filled
                if(page is filled completely){
                    //After Creating a record if page is full,I need to change islast flag
                    islast=false;
                }endif
            }endif
        }endif
    }endif
}
```



```

else
{
createNewPage(PageID) //If all pages are filled, System create a new page.
get NewPageName; //user insert a valid pageName
set NewPage ID; //ID is automatically assigned to prevent giving same ID;
add NewrecordToThatAddress(recordID);
}
close(TypeName);
close(System.Cat.txt);
}

```

4.2.2.Delete a Record

```

DeleteRecord{
show(all primary keys); //System show the all keys
//Primary keys are given automatically but user need primary key for update a record
open(System.Catalog.dat);
while(!=EOF) //Search Type in the File
{
Get PrimaryKey //User enter the Primary Key which given automatically by the system
if(check(Primary Key))
//System purpose is in there finding the record with primary key
binary search
{
get record in the data;
delete record;
flag isEmpty=true;
//Space of the deleted record doesn't filled with next record. So system doesn't need any
sort algorithm
}
else
{
errorMessage("Not Found");
}
}
}endwhile
close(System.Cat.dat);
}

```

4.2.3.Update a Record

```
UpdateRecord{
get TypeName
open(System.Catalog.dat);
show(all primary keys);//System show the all keys
//Primary keys are given automatically but user need primary key for update a record
Get PrimaryKey//If Type is valid,user enter the Primary Key
Get #OfRecord;
Get newRecord;
while(!=EOF)//Search Type in the File
{
    open (page);
    binary search
    if(check(PrimaryKey))//If System find the inserted Key ,system continues to operations
    {
        update record; //Flag doesn't change and new record added to same address.
    }
    else
    {
        errorMessage("Not Found");
    }endif
    close(page);
}endwhile
close(System.Cat.dat);
}
```

4.2.4.Search a Record

```
SearchRecord{
get #OfPages;
show(all primary keys);//System show the all keys
//Primary keys are given automatically but user need primary key for update a record
Get Primary Key;//user entered the primary key which is given automatically
open(system.catalog.dat);
integer i=0;
for(i=0;i<#OfPages;i++)
//System start to search from first page until the last page
{
    open (Page [i]);
    binary search
    //In the Type[i],System start to search with binary tree algorithm
    if(check(Primary key is found))//Primary key which is inserted by user
    {
        go to record;
        Print(record name and its fields);
    }endif
}endfor
closeFile(TypeName.txt);
Errormessage("This Primary Key doesn't exist");
close(system.cat.dat);
}
```

4.2.5.List all Records

```
ListAllRecord{
get #OfPages;
open(system.catalog.dat);
integer i=0;
  for(i=0;i<#OfPages;i++)
    //System start to list all record from first page until the last page
    {
      open(Page[i]);
      Print(all records name and its fields);
      closePage[i];
    }
  }endfor
  Errormessage("This Primary Key doesn't exist");
  close(system.catalog.dat);
}
```

5.CONCLUSION

Throughout this project, my goal is designing a basic but efficient database management system.This Project was very useful for understanding how the database system is working.

Disadvantages of my project is using fixed length record.For example ; I allocate full size for one short data.However using fixed size record can be used for more quick algorithms.So using fixed size record can be changed for your usage(what you expect from program). Another disadvantage is using one key field which is restricted our relations between the pages and relation types.Additionally I can change showing all Primary Keys before seraching and updating algoririthms.

Using flags in the pages and records prevent to implement extra algorithms.For Example;I deleted a record ,system just change its flag.If I don't use flag,I need to push up remaining records and pushing up used with extra algorithms.

Finally,I have tried to design the system as much as being faster and useful. Additionally, I use some comment lines for explain what I did.