

CMPE 321

PROJECT 2 REPORT

DESIGN OF A SIMPLE STORAGE MANAGER FOR A DATABASE MANAGEMENT SYSTEM

MEHMET SEFA BALIK

2016400372

1-INTRODUCTION

1.1 PURPOSE

Purpose of this project is to design a simple storage manager for a database management system. The storage manager system should access the database in a secure and efficient way.

1.2 What is DBMS?

DBMS stands for database management system. Database management system is a system software which lets us create and manage databases. Users and programmers can create, retrieve, update and manage data using DBMS with a systematic way. Actually, database management system is an interface between database and application programs.

1.3 Overview

There are three main parts in this project:

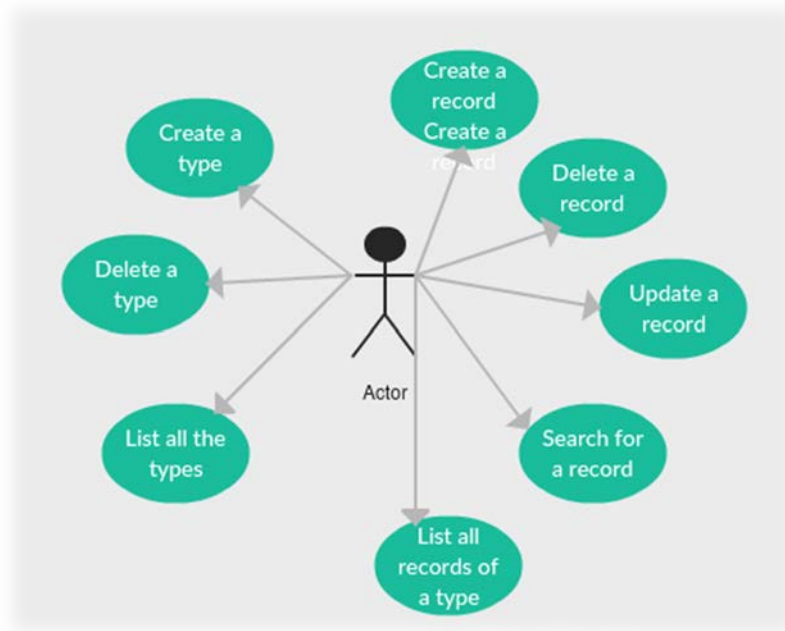
- Assumptions,
 - In this part, I made some constraints and assumptions.
- Data structures,
 - In this part, contents and the functionalities of the record format, system catalog, file format and page format are explained with simple tables and drawings.
- Algorithms that is written in pseudo code
 - In this part, I simply wrote DDL and DML operations in pseudo code.

DDL Operations stands for Data Definition Language, and will work on the structure of data, namely:

1. Create a type
2. Delete a type
3. List all the types

DML Operations stands for Data Manipulation Language, and will work on the data, manipulates the data itself, namely:

1. Create a record
2. Delete a record
3. Update a record
4. Search for a record by primary key
5. List all records of a type



2. ASSUMPTIONS AND CONSTRAINTS

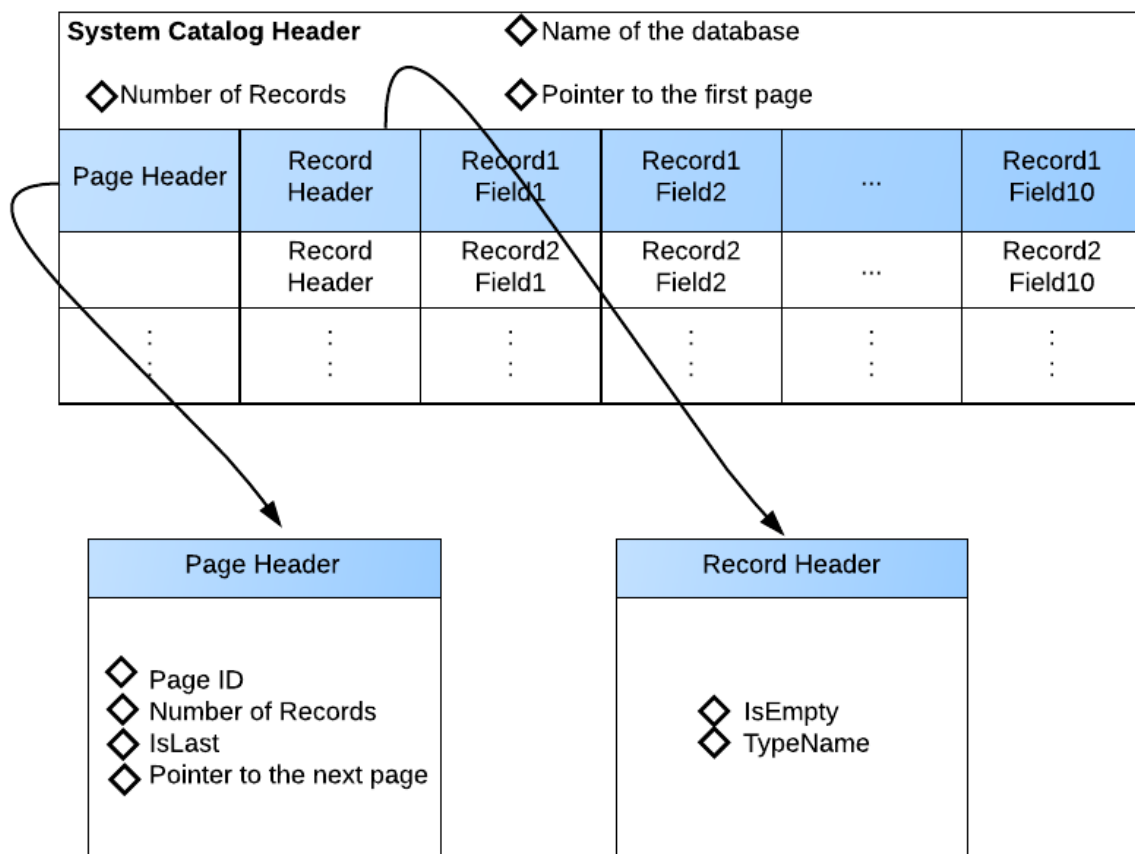
- Each page is 908 bytes.
- Each string character is 1 byte.
- The number of fields for a certain type is 5.
- The number of characters for a field is 15.
- Sizes of all records is fixed, I will use fix size records.
- There are 2 type of file construction in the project: System Catalog(sys.cat) file and Data File(% .dat).
- For all the types, program will assign an id as the primary key field.
- Type names must be unique.
- System catalog must store each type data in a separate page.
- Maximum number of records in a page is 5.
- Algorithms are written assuming that user will always input valid data as described in project description.

3.DATA STRUCTURES

3.1 System Catalog

System Catalog file is an important component of the storage manager because it contains the metadata of types. In my design System Catalog file contains a **system catalog header** that holds the **number of pages data, pointer to the first page** and **name of the database**. Since my design is based on pages, system catalog consists of pages. Each pages has a page header. **Page header** holds the **page id, number of records, islast** and **pointer to the next page** data. Each page can hold maximum 10 records. Each record has a record header. **Record header** has the **isempty and typename** data. Each record can hold maximum 10 fields.

To clarify which data system catalog will hold, **when a new type is created, a new record will be created in the system catalog file**. The size of the records will be fixed so each record will contain 10 fields. First field of the record will be the primary key.



3.2Data File

Data file is the file that can contain multiple pages inside, and pages holds records up to 5. Every record holds 5 fields, it is fixed. Records holds fields of a type, User must provide all the fields.

➤ **File Header holds:**

- the number of records
- Pointer to the first page

➤ **Page Header holds:**

- PageID
- Number of records
- IsLast
- Pointer to the next page

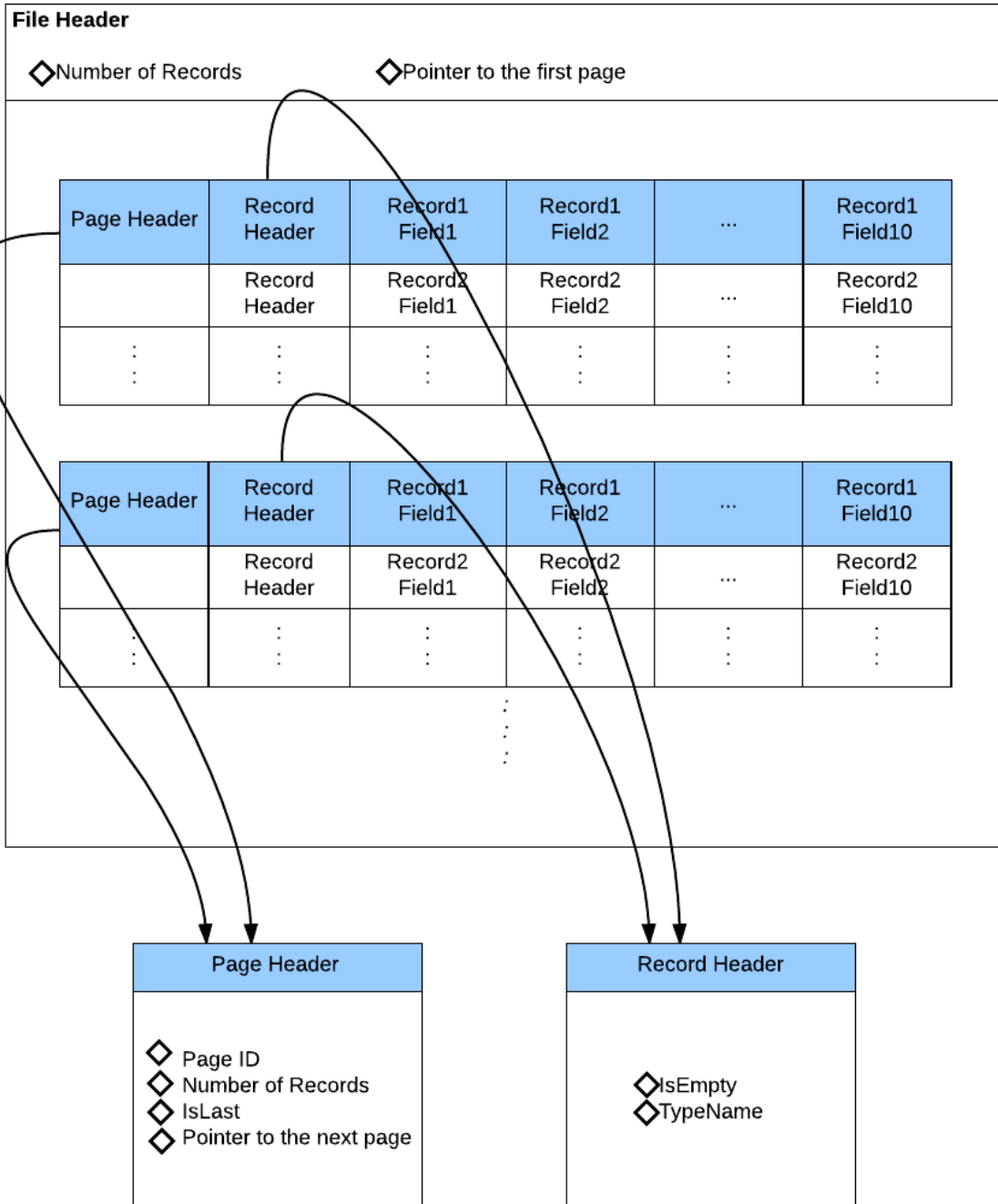
➤ **Record Header holds:**

- IsEmpty
- TypeName

Data file has a file header that contains the number of records in the data file and pointer to the first page. We will be able to know how many records there are in the file with number of records attribute in file header.

Page also has a page header that contains the pageid, number of records, islast and pointer to the next page. With the help of number of records in the page header, we can know how many records in the page. And islast attribute lets us to know if the page is the last page. Pointer to the next page lets us to reach next page.

Each record also has a header that contains isempty and typename. Isempy lets us to know whether the record is empty or not. Typename attribute has the name of the type of the record.



4.ALGORITHMS

4.1 Create a Type

```
function createType()
  open(sys.cat)    //opens System Catalog
  name=getTypeName() //get type name from user
  write(name into record header in sys.cat)
  j = getFieldCount()    //get field count of type from user
  i = 1
  while i <= j do
    content = getFieldContent()    //get field content from the user
    addField(name, i, content)    //set record of the type
    increment # of records in pageheader and system catalog header
    i++
  endwhile
  close(sys.cat)
end function
```

4.2 Delete a Type

```
function deleteType()
  name = getName()
  deleteRecords = ask(name)    //ask whether the user wants to delete all records of that type or
                                //just delete type
  if deleteRecords is true then    //if user wants to delete all records of the type
    open(data.dat) //open data file
    tobeDeleted = search(name)    //search for the records that has the type name and store
                                //the ids of that records in a list
    foreach item in tobeDeleted do
      deleteRecordById(item)
      decrement # of records in pageheader and fileheader
    endforeach
    close(data.dat)
  endif
  open(sys.cat)
  while existsRecord(name) do
    deleteRecordByName(name)
    decrement # of records in pageheader and system catalog header
  endwhile
  close(sys.cat)
endfunction
```

4.3 List All Types

```
function listTypes()
  open(sys.cat)
  if not recordExists() then
    print there is no type name
  else
    list = binarySearchByName(name)
    foreach item in list
      print item
    endforeach
  endif
  close(sys.cat)
endfunction
```

4.4 Create a Record

```
function createRecord()
  name = getNameFromUser()
  open(sys.cat)
  open(data.dat)
  fieldlist = getFieldListByName(name)
  fieldlength = length(fieldlist)
  while true do
    i = 1
    while i <= fieldlength do
      content = getContentFromUser()
      pageID = getAvailablePageID()
      write(name in typename, pageID)
      write(content in ith field of record)
      i++
    endwhile
    if askUserToQuit == true then
      break loop
    endif
  endwhile
  increment # of records in fileheader
  increment # of records in pageheader
  close(sys.cat)
  close(data.cat)
endfunction
```


4.5 Delete a Record

```
function deleteRecord()
    recordID = getRecordID()
    open(data.dat)
    record = binarySearchById(recordID)
    delete record
    record.isEmpty = true
    decrement # of records in pageheader and fileheader
    close(data.dat)
endfunction
```

4.6 Update a Record

```
function updateRecord()
    id = getRecordId()
    name = getRecordName()
    open(sys.cat)
    open(data.dat)
    fieldlist = getFieldListByName(name)
    fieldlength = length(fieldlist)
    while true do
        i = 1
        while i <= fieldlength do
            content = getContentFromUser()
            pageID = getAvailablePageID()
            write(name in typename, pageID)
            write(content in ith field of record)
            i++
        endwhile
        if askUserToQuit == true then
            break loop
        endif
    endwhile
    increment # of records in fileheader
    increment # of records in pageheader
    close(sys.cat)
    close(data.cat)
endfunction
```

4.7 Search For a Record

```
function searchRecord()
    open(data.dat)
    name = getNameFromUser()
    primary = getPrimaryByName(name)
    fields = binarySearchByKey(primary)
    write(name, fields)
    close(data.dat)
endfunction
```

4.8 List All Records of a Type

```
function listAllRecords()
    open(data.dat)
    type = getTypeNameFromUser()
    recordIds = binarySearchByTypeName(data.dat)
    foreach record in recordIds do
        write(record)
    endforeach
    close(data.dat)
endfunction
```

5 CONCLUSION

By doing this project, my primary goal is to design a simple but efficient storage manager system for DBMS. My storage manager has two types of files: System Catalog and Data File. In assumptions section, I explained the assumptions, and in data structures part, I explained the data structures that is used throughout project.

System catalog file has the metadata of the database, such as type names and field names of the types. Actual data is in data file. Data file and system catalog file is based on pages. Both files can contain multiple pages. Each pages has 10 records and each record has 10 fields. I designed the records to have fixed number of fields. So one record can have less than 10 fields, but its complement of ten fields have null field.

I tried to design my storage manager fast and useful.