Simple Storage Manager

Cmpe321 - Introduction to Database System
Semester: 3 (Summer Class)
Name: Barın Özmen
Id: 2012400045

1.Introduction

The project is about learning database systems and design how can a storage manager be. The problem is about having lots of data and where to store them nicely to do finding that data back easy and fast. The better design we have, the most usable it is. This storage manager will also implement some features:

DDL Operations:

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

DML Operations:

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

2. Assumptions & Constraints

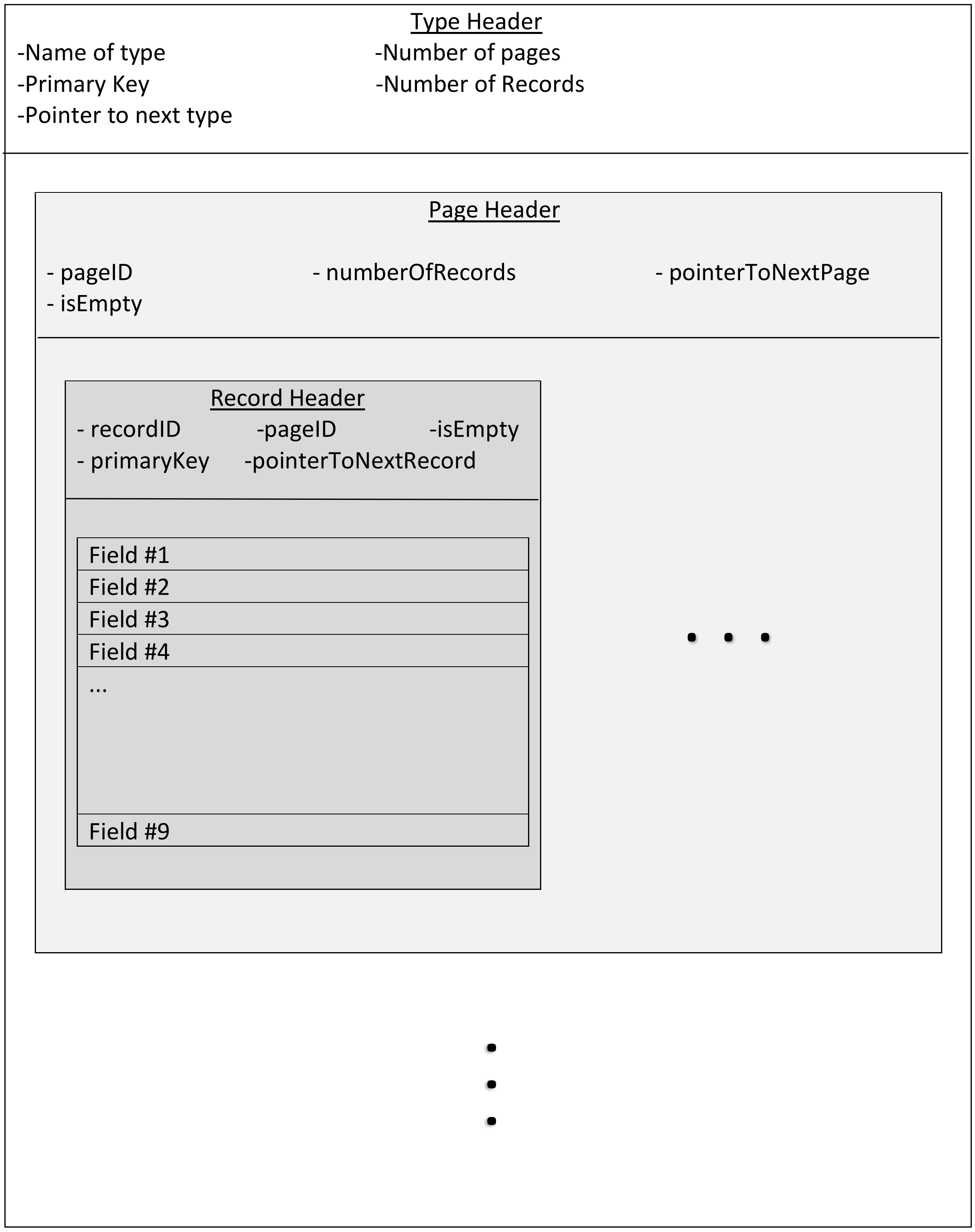
This is where this design project draw limits. It helps to decrease complexity. By giving assumptions and constraints first will make it simpler.

- Page size : 1KB per each
- Type header includes; name of types, primaryKey, numberOfPages, numberOfRecords, and pointer to next type
- Page header includes; pageID, isEmpty, numberOfRecords, pointerToNextPage
- Record header includes; recordID, pageID, isEmpty, pointerToNextRecord, primaryKey
- Max number of fields a type can have: 9 fields for each record.
- Max length of a type name: 10 bytes
- Max length of a field name: 10 bytes
- These assumptions mean 9x10+10 = approximately 100bytes for each record and maximum record number in each page is 10. 100x10 = approximately 1KB.
- Fixed number of fields. 9 for each record starting with null.

3. Data Structures

3.1 System Catalog

System Catalog is a file "sys_cat.data" to store meta data inside of it including type names, type numbers, page numbers and record Numbers. It also keeps pages are empty or not to decide and operate fast.



3.2 Type Structure

This part where the types are differ in each other and locates under that file. Number of fields are fixed to 9 it doesn't need to keep that record. Beside of that every type has its own primary key and ID to keep and fill records easily. It also keeps hown many records and pages are in that type. Also pointer to next type to operate fast.

3.3 Page Structure

Pages are under types and maximum size of 1KB per each. 1 page will has approximately 10 records at most if we count 100bytes for each record. Page header will have its own "pageID", look if this page is empty or not by checking "isEmpty" and it will help to operate fast in this structure, keeps "numberOfRecords", and has a pointer to go fast to next page "pointerToNextPage".

3.4 Record Structure

The data will come records' fields. First it will look for the records capacity by checking isEmpty. If it's still empty than the data record goes one of the "Null" Fields and replace with it. Every record has its own primary Key and ID to track it easily.

5. Operations (Algorithms)

5.1 DDL Operations

5.1.1 – Create a type

```
function createType(name)
        for (i in typeNumber(sys_cat.data)) do
                if(name == typeName[i]){
                        uniqueName = false
                end if
        end loop
        if(uniqeName == false ) then
                errorMessage: "Type already exist"
        end if
        else begin
                if(name.length > 10) than
                        errorMessage: "Long type name!"
                end if
                else begin
                        create(<name>.data)
                        write(sys_cat.data (typeNumber+=1))
                        write(sys_cat.data(typeNames.add(name)))
                        return(<name>+"added to your DataBase")
                end
        end
```

```
end function
```

5.1.2 – Delete a type

```
function deleteType(name)
    for(i in typeNumber(sys_cat.data)) do
        if(name == typeName[i]) than
            write(sys_cat.data(typeName[i].delete))
            write(sys_cat.data (typeNumber-=1))
            remove(<name>.data)
            return(<name> is deleted from you DataBase)
            isDeleted = true
            break
        end if
        else begin
            continue
        end
    end loop
    if ( ~ isDeleted ) than
        errorMessage: "Type does not exist!"
    end if
end function
```

5.1.3 – List all types

```
function listTypes()
    if(typeNumber(sys_cat.data) == 0) than
        errorMessage: "There are no types recorded in your DataBase!"
    end if
    else begin
        for (i in typeNumber(sys_cat.data)) do
            log(typeName[i](sys_cat.data))
        end loop
    end else
end function
```

5.2 DML Operations

5.2.1 – Create a record

```
function createRecord(variable)
    getTypeName
    if (getTypeName == null) than
        errorMessage: "Type does not exists"
    end if
    else begin
        while (page.isEmpty == false) do
            go pointerToNextPage
        end while

        primaryKey = variable
        insert record(primaryKey)
        page.setNumberOfRecords(+1)
        if(numberOfRecords > 9) than
```

```
                page.isEmpty(true)
            end if
        end else
end function
```

5.2.2 – Delete a record

```
function deleteRecord(primaryKey)
    getTypeName
    if (getTypeName == null) than
        errorMessage: “Type does not exists”
    end if
    else begin
        while( isFound == false && page.isEmpty== false) do
            for (i in recordNumbers)
                if (primaryKey == record[i].primaryKey) than
                    isFound == true
                    delete(record[i])
                    page.NumberOfRecords—
                end if
            end loop
            go pointerToNextPage
        end loop
    end else
    if (isFound == false) than
        errorMessage: “Record does not exists”
    end if
end function
```

5.2.3 – Seach for a record (by primary key)

```
function searchRecord(primaryKey)
    getTypeName
    if (getTypeName == null) than
        errorMessage: “Type does not exists”
    end if
    else begin
        while( isFound == false && page.isEmpty== false) do
            for (i in recordNumbers)
                if (primaryKey == record[i].primaryKey) than
                    isFound == true
                    display (record[i])
                end if
            end loop
            go pointerToNextPage
        end loop
    end else
    if (isFound == false) than
        errorMessage: “Record does not exists”
    end if

end function
```

5.2.4 – List all records of a type

```
function listAllRecords(typeName)
    getTypeName
    if (getTypeName == null) than
        errorMessage: “Type does not exists”
    end if
    else begin
        for i in typeName.pageNumbers do
            for j in page.RecordNumbers do
                display(record.primaryKey)
                go to: pointerToNextRecord
            end loop
            go to: pointerToNextPage
        end loop
    end
end function
```

6. Conclusion

This project aims to design proper dataBase yet, from the beginning it is a little bit hard for me to compherense because of not running and seeing on the moniter of what we think. Altough having trouble while designing, it makes me understand some of the features and it will help us to go on 2nd project much more easy. I am pretty sure that in second project i will change most part of this report, yet i believe it will nice draft for me to come up with something more efficient.