CmpE 321
Introduction to Database Systems

Spring 2014

Project 1

Student Name: Evin Pinar Ornek

Student ID: 2012400057

March 16, 2015

Contents

List of Tables

1 Introduction

A database management system(DBMS) is a computer software application that allows user to interact with the database to analyse and modify the data. A DBMS is usually used for creating, defining, querying, updating, and managing the databases. Databases are kept in physical materialisations of various storage structures. These structures are administered by storage management systems. Generally, those systems deal with the organisation of databases in storage and the techniques for accessing them using several algorithms. There are many types of storage and database models for different purposes. This project describes a design for a simple storage management system. It is constructed by using one of the most common storage approaches. Briefly, there are 2 types of files: the system catalog file and the data files. System catalog file includes the summary of the data files. The data files store the records of different types. The files and records are kept in an order for an easier and faster manipulation and retrieval.

1.1 Assumptions

The storage management system is designed by assuming the following rules.

1. The characters are a member of ASCII.
2. File and field names consist alphanumeric chacarters and cannot be an empty space.
3. The key values of the records are unique.
4. The logical deletion is used.
5. The sizes of following structures are:

   Data files are 4 Kbytes.

   File names are 64 bytes.

   Field names are 64 bytes.

   Fields are 64 bytes.

   File records are 4 Kbytes.

   The maximum number of fields is 10.

   The representation of number of fields in the page header is 1 byte.

   The representation of a field size in system catalog is 2 bytes.

   The status of a record in the data file is 1 byte.

2 Data Structures

2.1 The System Catalog File

The system catalog file includes the summary of all types or files in the database. It is unique and necessary for the systems. It does not have a header. Page format illustrates the general idea of a system catalog. The record format shows the informations needed to keep in the records of a system catalog file.

2.1.1 The Page Format

Simply, it keeps the files(or types) as records.

File # 1
File # 2
File # 3
...

Table 1: Page Format of System Catalog File

2.1.2 The Record Format

The files(or types) are kept in the system catalog as records. These records have the form and shown in Table 2 .

- Field names are the given by the user to the created files. Their size cannot exceed 64 bytes.
- The number of fields is represented by 1 byte.
- The size is represented by 2 bytes.
- The maximum number of fields is 10.
- The field size can be at most 29 bytes. (2 x 10 + 9)
- The field names can be at most 649 bytes. (64 x 10 + 9)
- Primary key is unique for each record and it is one of the fields of each type.

File Name	Number of Fields	Field size	Field names	Primary Key
(64 bytes)	(1 bytes)	(29 bytes)	(649 bytes)	(1 byte)

Table 2: Record Format of System Catalog

2.2 Data Files

There is one data file for each type. The data files consists of pages. Generally, it can be demonstrated by the following table 3.

Page # 1
Page # 2
Page # 3
...

Table 3: Data File

2.2.1 Page Format

The page have a header which holds the number of empty records. Rest of the page includes all the records.

of empty records
Record # 1
Record # 2
Record # 3
...

Table 4: Page Format of Data File

2.2.2 Record Format

The actual data is kept in the records which have the following form and it is shown on the Table 5 .

- Each records have a header of 1 byte which represents whether the record is empty of full. The representative status is 1 if it is full, otherwise it is 0.
- Each record keeps 10 fields. One of these fields is primary key.
- Each field is 64 byte.

Record Status (1 byte)	Field #1	Field #2	...	Field #10

Table 5: Record Format of Data File

3 Algorithms

The overall system runs a menu-driven application. The program obtains input from a user by displaying the menu, which consists of options, from which the user indicates her choice.

3.1 The Main Algorithm

```
/* Asks user to choose one of the options.                    */
while TRUE do
    show the menu of options;
    ask user to choose DDL or DML;
    if DDL then
        go to DDL function;
    else
        go to DML function;
    end
end
```

3.2 Data Definition Language

A data definition(or description) language(DDL) is a syntax to define data structures.

3.2.1 Create a Type of File

```
/* Creates a type of file.The details are asked from user.The initial
   file is null.                                                      */
open system catalog file;
find the next empty record space;
ask file name or type;
create a data file with that name;
ask # of fields;
for $i = 1$ to # of fields do
    ask the field name;
    ask the field size;
end
ask the primary key value;
update the system catalog file;
set the # of empty records to maximum;
```

3.2.2 Delete a File

```
/* Deletes a type of file by its name.                                */
open system catalog file;
ask the file name to be deleted;
iterate in the system catalog;
if recordName == fileName then
    remove the record from the system catalog file;
    delete the record;
end
update the system catalog file;
close the system catalog file;
```

3.2.3 Display a File

```
/* Displays a type of file and its format.                            */
open system catalog file;
ask the file name to be displayed;
iterate in the system catalog;
if recordName == fileName then
    print the file name;
    print number of fields;
    print field names and sizes;
    print the primary key;
end
close the system catalog file;
```

3.2.4 Display All Files

```
/* Displays all types of files and their formats.                    */
open system catalog file;
iterate in the system catalog;
foreach type of file in the system catalog do
    print the file name;
    print number of fields;
    print field names and sizes;
    print the primary key;
end
close the system catalog file;
```

3.3 Data Manipulation Language

A data manipulation language(DML) is used for selecting, inserting, deleting and updating data in a database.

3.3.1 Insert a Record

```
/* Creates a type of record.The field informations are asked from user.
    */
open system catalog file;
ask file name or type to insert the record;
open data file with that name;
foreach page in the data file do
    if the page has empty record space then
        get the next empty record to memory;
    else
        create another page;
        get the next empty record to memory;
    end
end
for $i = 1$ to # of fields do
    ask the field information;
end
ask the key value for the new record;
set record status 1;
update the record;
update the page header;
close the data file;
close the system catalog file;
```

3.3.2 Retrieve a Record

```
/* Retrieves and displays a record with a given primary key value and
   file type.                                                          */
open system catalog file;
ask file name or type of the record;
open data file with that name;
ask the primary key value of the record;
foreach page in the data file do
    if nextPrimaryKey == myPrimaryKey && myStatus ==1 then
        for $i = 1$ to # of fields do
            print field information;
            nextLine;
        end
    end
end
close the data file;
close the system catalog file;
```

3.3.3 Retrieve All Record

```
/* Retrieves and displays all record in a given file type.             */
open system catalog file;
ask file name or type of the record;
open data file with that name;
foreach page in the data file do
    foreach record in the page do
        print primary key value;
        for $i = 1$ to # of fields do
            print field information;
        end
        nextLine;
    end
end
close the data file;
close the system catalog file;
```

3.3.4 Delete A Record

```
/* Deletes a record with a given primary key value and file type.     */
open system catalog file;
ask file name or type of the record;
open data file with that name;
ask the primary key value of the record;
foreach page in the data file do
    if nextPrimaryKey == myPrimaryKey && myStatus ==1 then
        set myStatus = 0;
    end
end
update the page header; close the data file;
close the system catalog file;
```

3.3.5 Delete All Records

```
/* Deletes all records with in the given file type.                   */
open system catalog file;
ask file name or type of the record;
open data file with that name;
foreach page in the data file do
    foreach record in the page do
        set recordStatus = 0;
    end
end
update the page header; close the data file;
close the system catalog file;
```

4 Conclusion

This project deals with the storage management of a database system. The design offers a basic approach to solve the problem of storage. It uses a system catalog file and data files in a straightforward manner. The main storage structures are simple and logical. The assumptions related to structure sizes may be modified for any purpose of usage. The main algorithms and retrieval techniques are sufficient and easy to implement. All in all, the system is flexible and simple but not suitable for more complex databases.