

## Cmpe 321 Project-2

Emre Hoşer  
2016400366

July 30, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What is a DMBS? . . . . .	3
1.2	What is the goal? . . . . .	3
<b>2</b>	<b>Assumptions &amp; Constraints</b>	<b>3</b>
2.1	Assumptions . . . . .	3
2.2	Constraints . . . . .	3
<b>3</b>	<b>Storage Structures</b>	<b>4</b>
3.1	System catalog(Data dictionary) . . . . .	4
3.2	Page Header . . . . .	5
3.3	Record Header . . . . .	5
<b>4</b>	<b>Operations</b>	<b>6</b>
4.1	DDL Operations . . . . .	6
4.1.1	Create a type . . . . .	6
4.1.2	Delete a type . . . . .	6
4.1.3	List all types . . . . .	7
4.2	DML Operations . . . . .	7
4.2.1	Create a record . . . . .	7
4.2.2	Delete a record . . . . .	7
4.2.3	Search a record . . . . .	8
4.2.4	Update a record . . . . .	8
4.2.5	List all record of type . . . . .	8
<b>5</b>	<b>Conclusion &amp; Assessment</b>	<b>9</b>

# 1 Introduction

## 1.1 What is a DMBS?

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation of the data in the database or the management of the database structure itself. Database Management Systems (DBMS) are categorized according to their data structures or types.

## 1.2 What is the goal?

This assignment is about designing simple database management (storage) system. The desired system requires DDL operations (creating, deleting types and listing all types with itself) and DML operations (creating, deleting, searching and updating for a record, and listing all records of a type with itself). The document includes all the assumptions and pseudo code for the project. As a result, the main purpose of this Project is designing a storage manager for consistent, efficient and ease to use systems.

# 2 Assumptions & Constraints

## 2.1 Assumptions

- Each page has 1KB size.
- Type names must be unique.
- User always enters valid input.
- All fields shall be integers. However, type and field names shall be alphanumeric.
- A disk manager already exists that is able to fetch the necessary pages when addressed.

## 2.2 Constraints

1. Maximum length of a name type name is 8 characters.
2. Maximum length of a name field name is 8 characters.
3. Each type is not able to contain more than 10 fields.
4. Each file is not able to contain more than 4 pages.
5. The data must be organized in pages and pages must contain records.

6. It is not allowed to store all pages in the same file and a file must contain multiple pages.
7. Although a file contains multiple pages, it must read page by page when it is needed. Loading the whole file to RAM is not allowed.

### 3 Storage Structures

#### 3.1 System catalog(Data dictionary)

Data dictionary(system catalog) is the one of essential parts of DBMS because metadata and all data storage units are kept in it. Data dictionary is able to be reachable for anytime.Each file has a unique type.Each type is belong to a file not another.This system catalog shall include following:

- Number of files
- Pages header
- Number of records
- Number of fields
- Total size of records
- Primary key
- Whether a file full or not
- last page ID

System Catalog(Disk directory) *number of files *pointer of the first file *name of DB					
Type(file) name	Number of Pages	Number of fields	isFull	Primary key	Last pageID
FileA	20	9	1	a	201
FileB	17	8	0	b	171
FileC	15	7	0	c	151
FileD	20	6	1	d	202
FileE	16	9	0	e	161
...	...	...	...	...	...

Figure 1: system catalog

### 3.2 Page Header

Page Header *number of records *Pointer of first records *pageID				
Records	Primary key(field1)	Field2	Field3	...
Record1	<u>1</u>	123	456	...
Record2	<u>2</u>	789	321	...
Record3	<u>3</u>	654	987	...
Record4	<u>4</u>	876	432	...
Record5	<u>5</u>	65	32	...
...	...	...	...	...

Figure 2: page header

### 3.3 Record Header

Record Header *Pointer of first field			
Primary key(field1)	Field2	Field3	...

Figure 3: record header

## 4 Operations

### 4.1 DDL Operations

#### 4.1.1 Create a type

```
//Create a type

declare table
createType{
  typeName <-- input
  numberOfField <-- input
  table.push(typeName,numberOfField);
  for 0 to NumberOfField do
    nameOfTheField <-- input;
    table.push(nameOfTheField);
  end
  primaryKey <-- choose(nameOfTheField)
  file <-- open(dataDictionary)
  file.push(table)
```

#### 4.1.2 Delete a type

```
nameOfType <-- input
deleteFile(nameOfType){
  file <-- open(dataDictionary)
  for each type in file do
    if(nameOfType == type.name) then
      type.delete;
    end
  end
end
```

#### 4.1.3 List all types

```
file <-- open(dataDictionary).getType
for each type in file do
    print(type);|
end
```

## 4.2 DML Operations

#### 4.2.1 Create a record

```
newRecord <-- input
thatType <-- getType(newRecord.type,dataDictionary)
if thatType.isFull == 1 then
    createExtension(thatType);
    extentionType.push(newRecord);
else
    if thatType.lastPage.isFull == 1 then
        createNewPage(thatType);
        thatType.lastPage.push(newRecord);
    else
        thatType.lastPage.push(newRecord);
    end
end
end
```

#### 4.2.2 Delete a record

```
theRecord <-- input
thatType <-- getType(theRecord.type,dataDictionary)
for each page in thatType do
    for each record in page do
        if theRecord.primaryKey == record.primaryKey then
            deleteRecord(record);
        end
    end
end
end
```

#### 4.2.3 Search a record

```
theRecord <-- input
thatType <-- getType(theRecord.type,dataDictionary)
for each page in thatType do
  for each record in page do
    if theRecord.primaryKey == record.primaryKey then
      print(record);
    else
      print(no record);
    end
  end
end
end
```

#### 4.2.4 Update a record

```
theRecord <-- input
thatType <-- getType(theRecord.type,dataDictionary)
for each page in thatType do
  for each record in page do
    if theRecord.primaryKey == record.primaryKey then
      for each field of record do
        field <-- input
      end
    end
  end
end
end
```

#### 4.2.5 List all record of type

```
theType <-- input
thatType <-- getType(theType,dataDictionary)
for each page in thatType do
  for each record in page do
    for each field of record do
      print(record);
    end
  end
end
end
```



## 5 Conclusion & Assessment

Throughout this project, my goal is programming a basic but efficient database management system. This Project was very useful for understanding how the database system is working. I cannot extend file and page number. I cannot handle with that problem. However, I think rest of constraints is supported in my project.