Simple Storage Manager Implementation

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

1.Introduction

This is the second part of the project which implementation takes apart. After designin the database, near %35 of the design is changed as expected. Yet, desiging before implementing was really helpful, only some names and values are changed, plan is still same.

In this project i have implemented a simple database management system in JAVA programming language. This program creates a file if there is no exist, else read and write on that specific file. It holds only 1 file with the storage capacity 1MB. This storage manager also makes these 7 operations successfuly.

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. Changes From Initial Design

2.1) Assumptions & Constraints

This is where the changes are shown from the initial design. Red characters represents the old design. Red character = “it is deleted!!” ... Blue character = “Created new!!”

- Page size : 1KB - 1240 Bytes per each
- Type header includes; name of types, primaryKey, numberOfPages, numberOfRecords, pointer to next type, isEmpty and newType<ID>
- Page header includes; pageID, isEmpty, numberOfRecords, pointerToNextPage and newPage
- Record header includes; recordID, pageID, isEmpty, pointerToNextRecord, primaryKey and newRecord<ID>
- 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 + 20 = approximately 120bytes for each record and maximum record number in each page is 10. 120x10 = approximately 1200Bytes + 40Bytes for page Header = 1240 Bytes per page as default.
- Fixed number of fields. 9 for each record starting with null.
- Max number of type that DBMS or SystemCatalog can have 10 types. 1 type has 75 pages with approximately 100KB. Because storage manager can have 1MB max. 1000/100 = 10 type maximum.
- System Catalog includes, Number of pages, types and records. Besides that it also keep the record of Storage information and type names that created. Sys Catalog keeps 210Bytes.

2.3) Data Structures

2.3.1 System Catalog

System Catalog is a file 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. System Catalog also keeps the records of Storage information to decide it exceed 1MB or not.

2.3.2 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. It also has newType<ID> to determine exact location of the database to be able to do Read&Write. There is also field at footer which is called "endRecord<ID> to decide if the type is finished or not.

2.3.3 Page Structure

Pages are under types and maximum size of 1240 per each. 1 page will has approximately 10 at most 9 records at most if we count 120bytes 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". newPage<ID> to decide position.

2.3.4 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. With primary key, records have also newRecord<ID> to check double to find exact location. With combination of PK and ID, surely the keyword can be found.

Type Header

-Name of type
-Primary Key
-Pointer to next type
-Number of pages
-Number of Records
-newType<ID>

Page Header

- pageID
- isEmpty
- numberOfRecords
- newPage<ID>
- pointerToNextPage

Record Header

- recordID
- primaryKey
-pageID
- *toNextRec
-isEmpty
-newRec<ID>

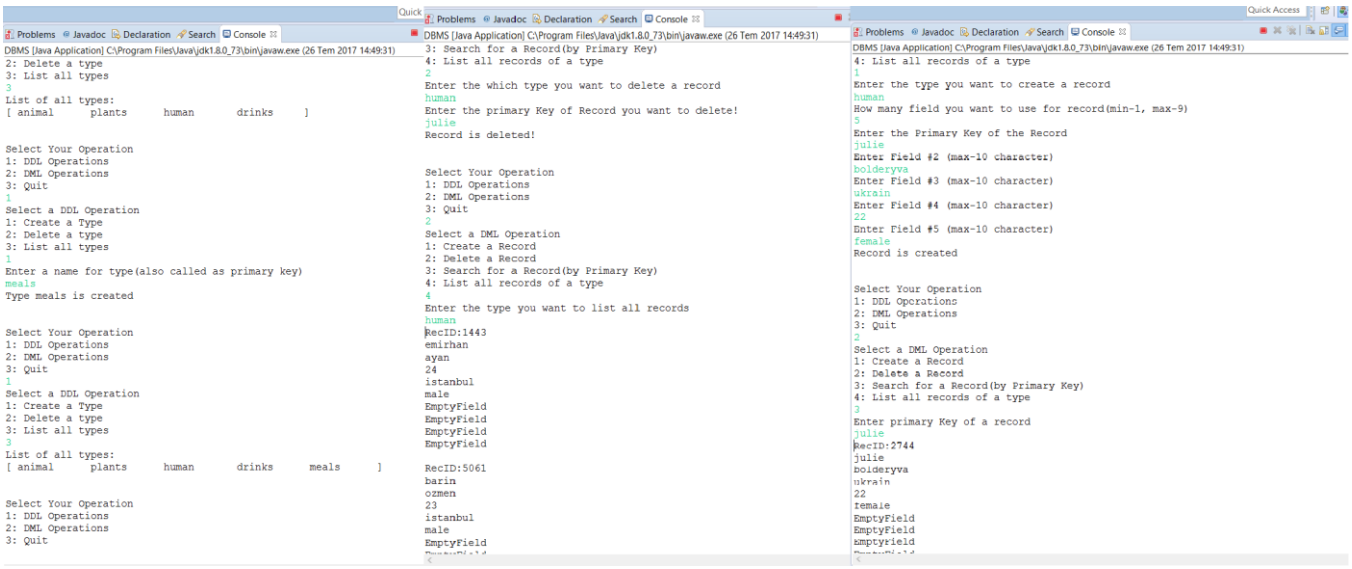
Field #1
Field #2
Field #3
Field #4
...
Field #9

Type Footer

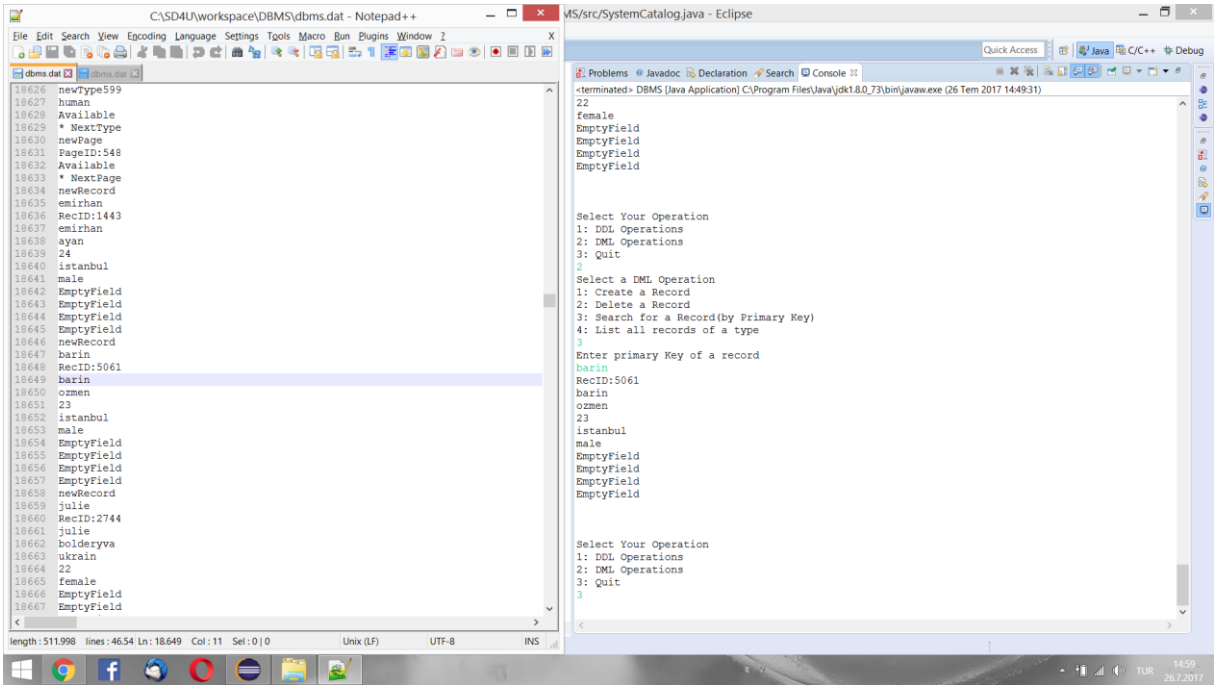
- endType<ID>

3. Sample Output

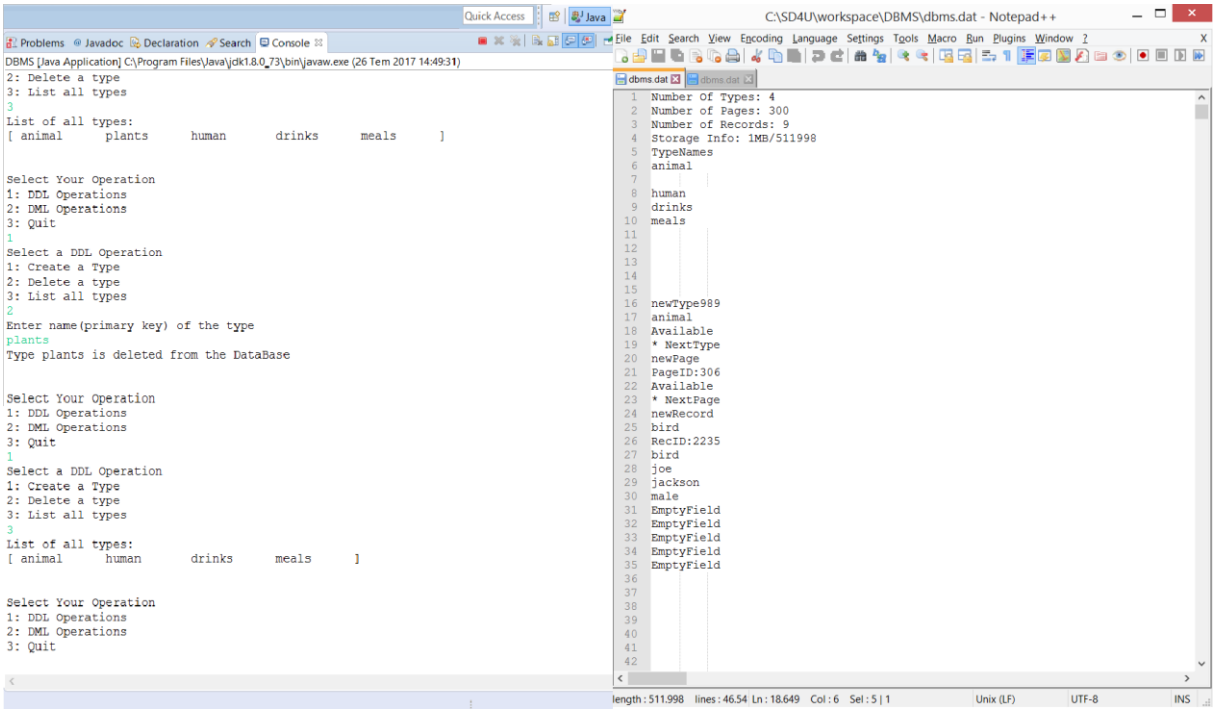
Here are some operation outpus of the program. It creates type, list the types, deletes a record, list all records and creates a record starting from the beginning in the first picture.



Search an operation for the spescific primary key of a record. And also shown in the database file (dbms.dat) line by line.



The last picture is about deleting a type and how it affects to database file.



4. Conclusion and Assessment

In this project, i learn how a database management systems' alghoritms work. From the beginning of it with designing and implementing onto a java code, most of the database concepts are learned well not only the techniques of it but also as a real time experiance. Whole system works well unless the user doesn't write wrong inputs. Yet, sentece written in the project description block that situation which is "Assume user always inputs valid data."