

CMPE 321 : INTRODUCTION TO DATABASE SYSTEMS

ASSIGNMENT 2

2017-2018 (2)SPRING

Implementing Storage Manager System

YUSUF KALAYCI

2016400357

Contents

1	Introduction	2
1.1	About The Project	2
1.2	Implementation Details	2
2	Changes From The Initial Design	4
3	Sample Usage & Outputs	5
4	Conclusions & Assessment	8

1 Introduction

1.1 About The Project

This project is about the implementation of a basic simple storage manager that is supposed to access database efficiently, conveniently and securely (DB Lecture Notes). The storage manager should have relation with file manager and database. User can create new types, delete and list types in a database and also do some record operations like insertion, deletion, list and search.

1.2 Implementation Details

In my project, I select Java language. I used Eclipse environment and the final product is a terminal application.

There are 2 classes:

- sysCatalogue.java
- Types_and_Records.java

sysCatalogue.java has methods:

- main() : This method is our project main and the program starts with this method. We can think the main is our Dbms.
- operationsMenu() : This prints whole operations and inside this method we can see another methods and these are calls from others. User can select whatever he/she wants in the operation menu and according to the choice calls this method for the specific one.
- listAllType() : This method scans the whole sysCatalogue file and list the information about the whole types. After list the type you can also call operationsMenu() and can some changes.
- createType() : This method creates data(.dat) and also index file(.idx). After the creation the data(.dat) stores in the myDataFile folder and also the .idx files stores in the directory. This method has also sub methods inside it. So it can call the createType() method from the TypesandRecords.java class. First created thing is the Primary key type.

- `deleteType()` : This method delete the data from the sysCatalogue. First of all this method wants from user the name of the table and after find the table in the sysCAtalogue, deletes it and deletes the data(.dat) from the myDataFile folder permanently but the idx. file can be found after deletion. My main goal is here, the user can find the data with idx file help. And also you can see after list the type the deleted file but you cant do any operation about it.
- `listAllRecords()` : This method scan the type and list all the records inside the type. Also it calls the corresponding method from the TypesandRecords class.
- `createRecord()` : This method ask the user about which type record insertion. User selects the type name and after this, the method scan the system catalogue and after find the name it allows the user create record in the type with calls the Types_and_Records class.
- `deleteRecord()` : This method allows the user delete a record from the type in the System catalogue. First of all same as the create part this ask the user about the name of type. Then goes the corresponding type delete the record inside it.
- `searchForRecord()` : This method search the records with the primary keys. And the user can see the records which includes the same primary key.
- `typeExist(String nameOfType)` : This method search the nameOfType which is given by the user in the system catalogue and after search if it finds return true, else return false and give some messages the user about the existion of type.

[Types_and_Records.java methods:](#)

- The whole methods inside the sysCatalogue.java class are calls from this class methods and these are works together.
- Search() : Searches the primary key in the index file. It works like binary search logic and calls from corresponding methods.

2 Changes From The Initial Design

The initial design is as follows and I made some changes in my first assignment when doing the implementation. I tried to stick to my first assignment but there is some changes in the implementation. The changes are;

- First of all I used special methods supplied from the Java. For example: readInt(), writeInt(), readChar(), writeChar()
- For file I have used RandomAccessFile.
- I also changed the idea of everything is fixed for some condition. For example user can select the each field length when the type creation but there is a limit with 10 characters.
- After deletion we can see the .dat file is deleted correctly but the idx. file is still there. When you tried some record addition or want to make sth with deleted data file, you cant. But when you list the types you can see the data but deleted version. I made this because I tried to see deletion history, I mean all created data in my database. I can find the better way for this but I made like that.
- I tried to stick to the first assignment logically but there is some changes about the design when writing the code in my implementation. I used some simplicity which is coming from the java languages and some methods.

3 Sample Usage & Outputs

In this part the terminal application will be shown with the screenshots and the way of how the program works. You can easily understand when looking the pictures about the general idea behind the implementation. These are the operations in the Dbms. The other figures will be for each operation screenshots.

1. Create Type;

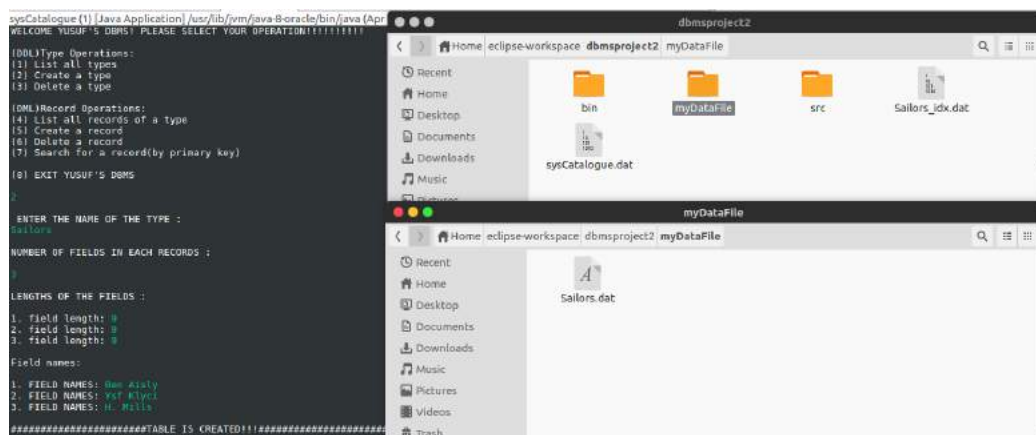


Figure 1: createType()

2. Delete Type;

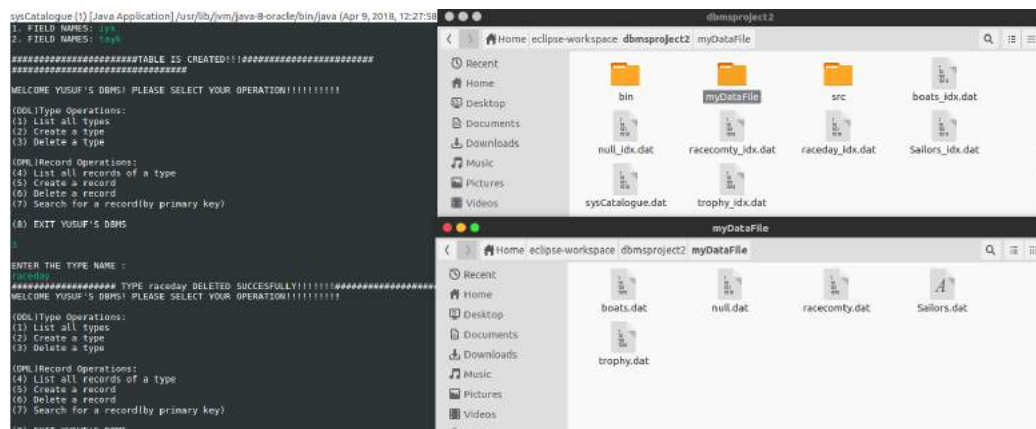


Figure 2: deleteType()

3.List All Types;

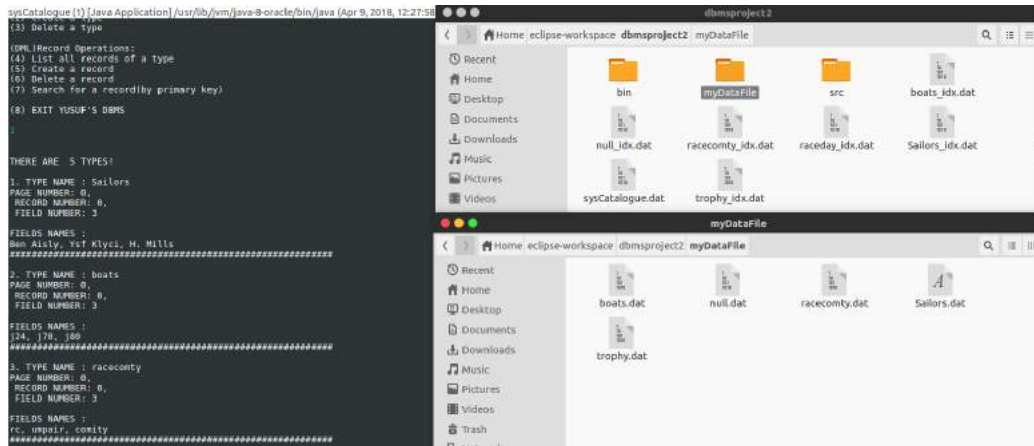


Figure 3: listAllType()

The types operations are like above. Then now, we can see the record operations and the record operations are as follows:

4.Create Record;

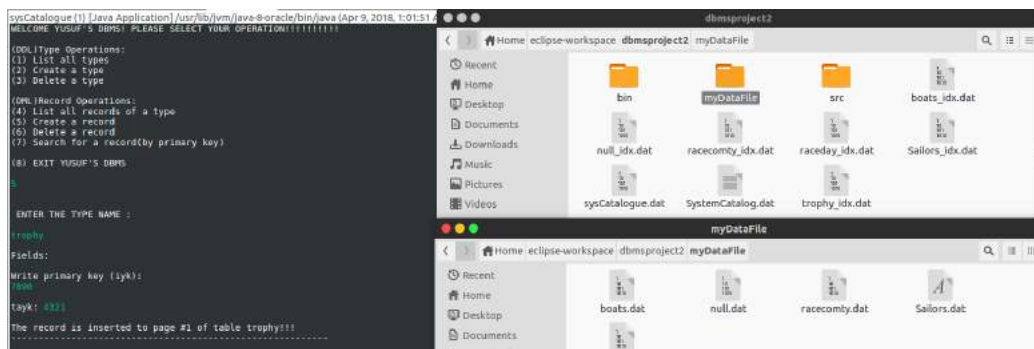


Figure 4: createRecord()

5.List All Records;

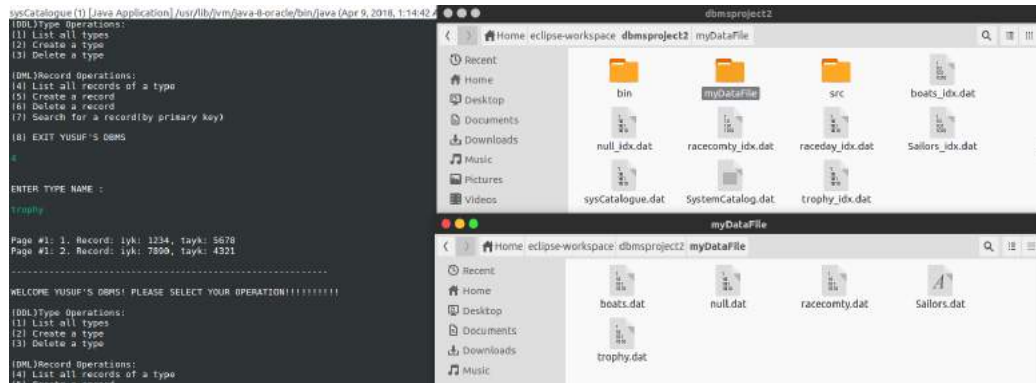


Figure 5: listAllRecords()

6.Search for Record;

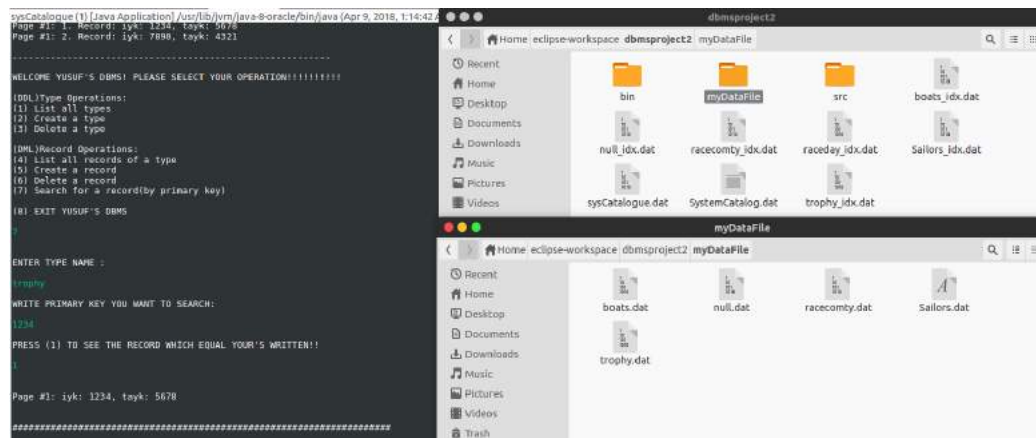


Figure 6: searchForRecord()

Delete record doesn't work properly I take an error which is::
 "at java.io.RandomAccessFile.readInt(RandomAccessFile.java:803)"
 but I do not resolve the problem.

7.Delete Record;

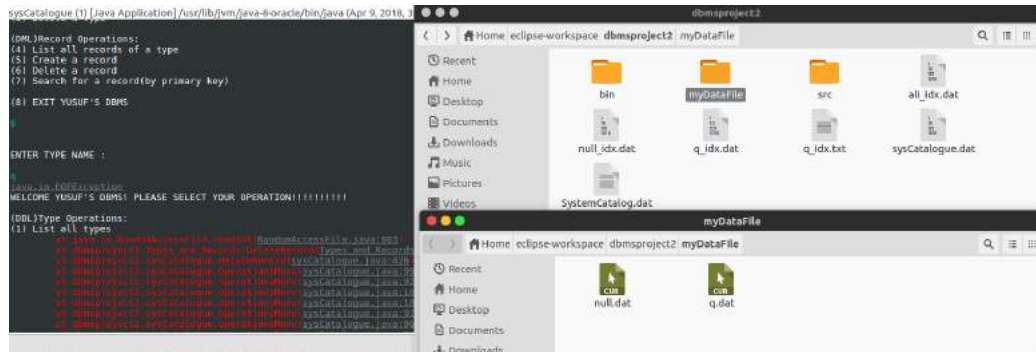


Figure 7: deleteRecord()

4 Conclusions & Assessment

This project is very helpful for me to understand DBMS and database structures. At the beginning of the project I searched about the architecture of Storage Manager. I tried to keep to my first assignment generally. I understood exactly after these projects the first plan then implementation is so difficult. So these projects are very helpful for me to get used to this style because the real world is like that.

Pros and Cons about the system;

- I used java language and Eclipse Oxygen IDE.
- Java offers some methods, some simplicity and increase the writability for the developers
- In the system I used fixed size for something, this is not a good way for storage allocation. Not all of them.
- Dynamic memory allocation for the records size can be more efficient.

- In the system about the page size; less page size system will be better.
- It is a good thing to use primary key just once and also first field.
- I seperated my implementation just two classes and it decrease the readability.
- Create type is so fast because it add new data file at the end of the page orderly.
- Delete record method has problem and didnt work properly.

The system can have some deficiency as well as every system has but I try to make a simple storage manager and I understood exactly the function of storage manager, file manager, database and their relations.

My implementation is the result of my first project the design of the simple storage manager. Especially I took the feedbacks from my first project then I use Eclipse IDE for Java language. I watch some videos about the java tutorials. These tutorials are about the RandomAccessFile operations and also the though part for my design. I seperated my project two class. My .dat files store in the myDataFile folder and sysCatalogue.dat, idx are store in the root directory of the project folder. .dat files is unsorted and idx is sorted file. When user create program finds the primary key and then create the data one by one with the order.

As a conclusion; it is hard work to design before the implementation. Because we always make implementation before the design. But the real world is not like that so I really need to thank you about this situation. I struggle with these condition before the geting a job.



Figure 8: THANKS.