Course Name: Cmpe 321 – Introduction to Database Systems
Semester: 2017/2018 Summer Term
Student Name: Burak Çuhadar
Student Number: 2015400153

# Implementing Storage Manager System

23 July 2018

## Introduction

In this project, we are expected to implement the design we had in the first project. I implemented the project using Java. To read the files by pages I have used RandomAccessFile class in Java. With this class we can read files with any page size we want and write data at any location in the file. Since we cannot delete data at a specific location, I have used the next empty record field of the records as an indicator whether the record is full or empty.

All the fields of a record are 32 bit integers. The pointers to pages are 64 bit long numbers and pointers to records are integers. Names are encoded using UTF 8 standard.

There are seven classes:

- Main:  Implements the part that interacts with the user. It parses the given command and calls the appropriate methods.
- SystemCatalog: This is responsible for listing types and getting next empty page for a type.
- Type: Create type and delete type operations are implemented in this class. Some helper methods like getting number of fields of a type and finding location of the type in system catalog are also defined in this class.
- Record: Every operation about records(search, create, delete, list) is implemented here. It also has fields to fields and address of next empty record to store them when reading a page from a file.
- Page: Creating and reading page operations are implemented here. Also it includes a class named PageHeader which stores the data in page header.
- ByteUtility: This class consists of helper functions for the other functions defined in the classes. Converting data types like integer, long and string into byte arrays are implemented here.
- C: This class includes C(onstants) like page size, maximum name length, file name of system catalog etc.

### Changes from the Initial Design

Next empty record fields of records can have the value -1 meaning null and -2 meaning full. Null means that there isn't any empty record left in the page and full means that the record is not deleted and list record operation can list it. Next empty page field of pages can now have the value -1 representing null.

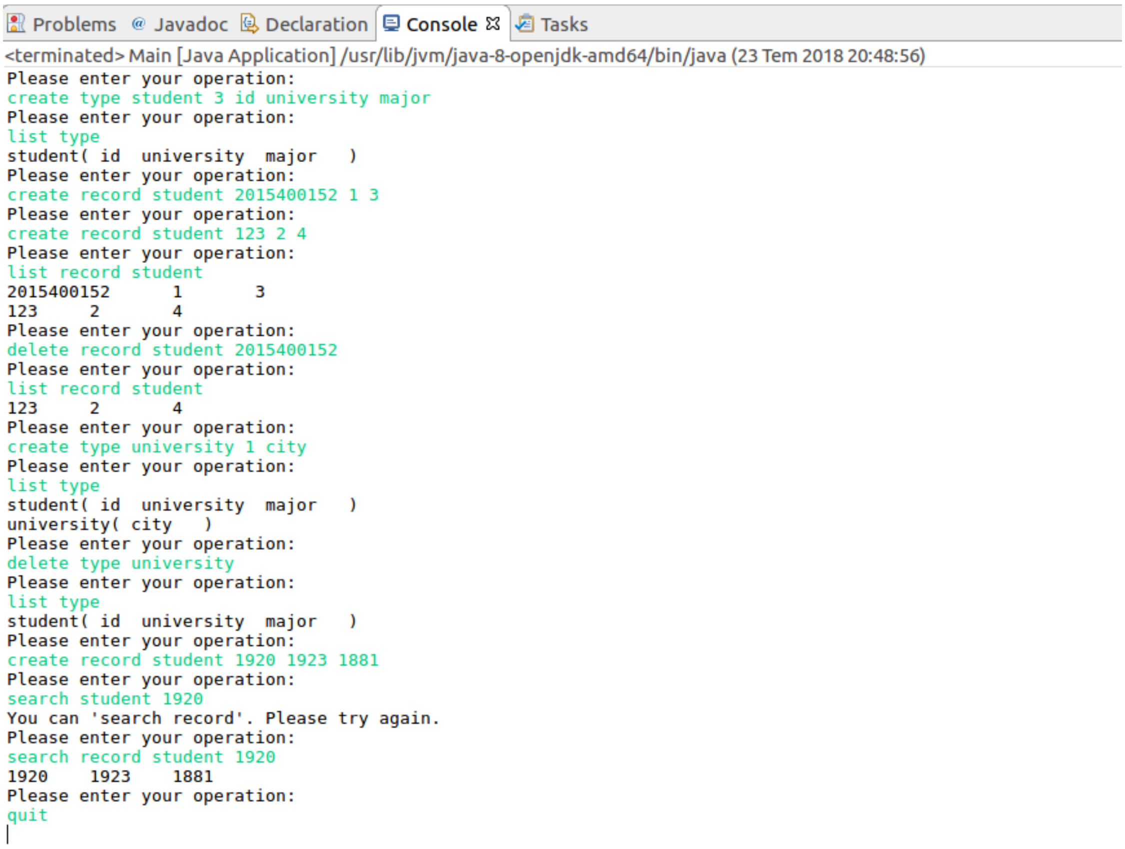
When deleting a type from system catalog instead of making its name and field names null, it is completely deleted. To do it system catalog from the beginning until the type definition and after the type until the end of file is copied into a new file and renamed as system catalog again.

When creating record if the page becomes fulls and there is at another empty page in the file, its next empty page field is set to -1(null) which tells that the page is full.

If there is additional space left which cannot be filled with records, this space is filled with 0 bytes.

### Sample Usage & Outputs

- creating record:
  **create record type-name value1 value2 ....**
- deleting record:
  **delete record type-name key-value**
- searching record:
  **search record type-name key-value**
- listing all records of a type:
  **list record type-name**
- creating type:
  **create type type-name number-of-fields field1-name field2-name ....**
- deleting type:
  **delete type type-name**
- listing types:
  **list type**



```
<terminated> Main [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (23 Tem 2018 20:48:56)
Please enter your operation:
create type student 3 id university major
Please enter your operation:
list type
student( id  university  major   )
Please enter your operation:
create record student 2015400152 1 3
Please enter your operation:
create record student 123 2 4
Please enter your operation:
list record student
2015400152      1       3
123     2       4
Please enter your operation:
delete record student 2015400152
Please enter your operation:
list record student
123     2       4
Please enter your operation:
create type university 1 city
Please enter your operation:
list type
student( id  university  major   )
university( city   )
Please enter your operation:
delete type university
Please enter your operation:
list type
student( id  university  major   )
Please enter your operation:
create record student 1920 1923 1881
Please enter your operation:
search student 1920
You can 'search record'. Please try again.
Please enter your operation:
search record student 1920
1920    1923    1881
Please enter your operation:
quit
```

## Conclusions & Assessment

I have successfully implemented my design with changes stated above. To improve it, when deleting types instead of copying system catalog I could have flags for each type which denotes whether it is empty or full. That way the performance would increase if system catalog is huge and if there are many delete type commands.