# Project Toy

## A. Description

In this project, you design and implement a program, let's just call it "toy", that manages a table. Specifically, it supports the following functionalities.

### 1. (30 points) Toy create xyz.tb

This command creates a table called xyz.tb. It prompts the user to enter the information of attributes one by one. For each attribute, it does the following tasks:

1) Display "Attribute name: ", then read in the user input.
2) Display "Select a type: 1. integer; 2. double; 3. boolean; 4. string". Read in the number. In this project, an integer is java int, a double is java double, a boolean is java char of either "F" or "T", and a string is a java String.
3) Display "More attribute? (y/n)". If the answer is "y", repeat the above steps.
4) Otherwise, save the header of the newly created table, named as xyz.tb. For simplicity, save the header information block by block, each starting with "[" and ending with "]", as follows. The first block records the number of attributes. The next block records the information for the first attribute in the format of "[name:type]". Then the next block records the information for the second attribute, and so on so forth, until all attributes are recorded. The last block records the number of records in the table. Of course, this is 0 for a newly created table.

   As an example, consider this header for a table called abc.tb

   ```
   [4][SSN:4][Name:4][Salary:2][Married:3][98]
   ```

   The above header tells:

   - The table abc.tb has 4 attributes.
   - The first attribute is named "SSN", type being "String".
   - The second attribute is named "Name", type being "String".
   - The third attribute is named "Salary", type being "double".
   - The forth attribute is named "Married", type being "boolean".
   - The table has 98 records.

Note that characters "[" and "]"are used as special tokens (for purposes such delimiters, etc). A few other characters, including "{", "}", and "|", are also reserved as special tokens.

### 2. (10 points) Toy header xyz.tb

This code reads xyz.tb and display its the header information. For example, in the case of abc.tb, it displays:

- 4 attributes

- Attribute 1: SSN, String
- Attribute 2: Name, String
- Attribute 3: Salary, double
- Attribute 4: Married, boolean
- 98 records

**3. (20 points) Toy insert xyz.tb**
This code appends a record to table xyz.tb. It first opens the file of xyz.tb. If the file doesn't exist, display an error. Otherwise, read in the table's header and for each attribute, ask and read in the value entered by the user. In the example of abc.tb, it will display "SSN:" and read in user input; then display "Name: " and read in user input; then display "Salary:" and read in user input; finally display "Married: " and read in user input.

When reading an input, make sure to check its validity. For examples, if the type is integer, the input must be an integer; if the type is boolean, the input must be either "F" or "T". If an input is wrong, display an error and quit.

After reading all attribute values, append the record to the end of the file and save the file to the disk. Be sure to update the number of record in the header. For simplicity, each record is stored in the format of "{attribute value 1 | attribute value 2|...|attribute value 1}", where the value for each attribute is saved as text. Suppose the record entered is

SSN: 123456789
Name: Scott John
Salary: 89076
Married: T

Then the record is stored as {123456789|Scott John|89076|T}. Note that it starts with "{" and ends with "}", and each attribute value is delimited by "|". As mentioned earlier, these characters are reserved and users are NOT supposed to use in a string. (Think about how you would allow users to use these special tokens!)

**4. (20 points) Toy display rid xyz.tb**
This code prints out the record in xyz.tb whose index is equal to rid. The index of the first record is 0. Print out the value of this record's all attributes, one by one. For example:

- SSN: 123456789
- Name: Scott John
- Salary: 89076
- Married: T

**5. (10 points) Toy delete rid xyz.tb**

This code deletes a record from xyz.tb whose index is equal to rid. Specifically, it opens the file of xyz.tb. If the file doesn't exist, display an error. Otherwise, read in the table's header and then read in all records, delete the record, and write the rest to the xyz.tb. Then close xyz.tb. Be sure to update the header for the record number.

**6. (10 points) Toy search "condition" xyz.tb**
This code searches and displays the records in xyz.tb that satisfy the specified condition. For this project, your code needs to support only equality search on a single attribute. For example, the condition can be "Salary = 69080.34", in which case your code searches the table and prints out all records with a salary value equal to 69080:

- Record 10
- SSN: 123456789
- Name: John Scott
- Salary:  69080
- Married: F

- Record 23
- SSN: 223456789
- Name: Mary Hall
- Salary:  69080
- Married: T

## B. Suggestions

1) Use java language. If you have to use another language, you will be asked to do a demo, since we may not be able to run your code on our machine.

2) Below are some major classes for your reference. A table basically consists of a header and an array of records, and each record has a number of attribute values.

```
class Table
{
       int        numberOfAttributes;
       Attribute  attribute[]; // attribute[i] records the information
                               // of the ith attribute
       Record     record[];
}

class Attribute
{
       String     name;
       int        type;
}

class Record
```

```
{
       String value[];   // value[i] stores the value of the ith
                         // attribute, in text
                         // NOTE: you may need to convert the String
                         // into the type specified by attribute[i]
                         // in order to perform some operations.
                         // For example, if value[0] = "1234" is
                         // originally an integer, you need to convert
                         // into a number before doing an "increment".
                         // You can add some methods here to do such
                         // conversions (e.g., toInt(), toDouble())
}
```

## C. Submission

Zip your source code in one file and submit through Canvas.

## D. Follow-up thoughts

If we want to make Toy not a toy but a real DBMS that can handle large-scale data, what must be done?