

COSC 6340—Database Systems

First Project: A Toy Relational Database

(due February 25, 2015 at 11:59:59pm)

For this project, you will develop a simple relational database system. The input to your program will consist of pseudo-SQL commands whose formats are illustrated below:

```
CREATE table (field type{, field type})
INSERT INTO table (value{, value})
SELECT * FROM table1, table2 WHERE field1 = field2
SELECT field {, field} FROM table
SELECT * FROM table WHERE field = constant
EXIT
```

where field is the *name* of a field and type is either INT or STRING. All queries will start on a new line and never exceed that line. Keywords will always be in upper case while table-names and field names will always be lower case. Constant values will always be integer values or strings delimited by apostrophes (single quotes). String sizes will *never* exceed 64 bytes.

Your program is to perform very rudimentary error checking before attempting to execute an operation and print out the resulting table. For instance, it should detect syntactic errors, such as missing/extra keywords. It should also detect some semantic errors, such as the fact that a given field does not occur in the specified table or that a comparison is attempted between a field value and a constant of a different type. Once an error is detected, your program should print an appropriate error message, skip the remainder of the command line and continue.

THE DATA DICTIONARY

You are to store the structure of your database in a data dictionary. This dictionary should have one entry per attribute and contain:

1. the name of the table containing the attribute,
2. the name of the attribute,
3. the attribute type (I or S), and
4. the name of file in which the table is stored.

Hence if your data dictionary contains:

book	author	S	BF
book	title	S	BF
book	lid	I	BF
patron	pid	I	PF
patron	name	S	PF

the relation **book** will be stored in the file BF in the following manner:

Homer	Iliad	0
H. Melville	Moby Dick	0
G. Vidal	Julian	7
...

IMPORTANT

1. Your program cannot read whole relations into the main memory but should process them one tuple at a time.
2. The data dictionary will always be small enough to be stored in main memory but must be written to disk at the end of the session.
3. You may choose any reasonable implementation for your data dictionary and the files containing your relations but you *must document them* in some detail in the comments of your program.

This document was updated last on *Wednesday, February 4, 2015*.