DATA WITH SIMPLEDB

Amazon's SimpleDB is a distributed document-oriented database. SimpleDB shares many characteristics of relational databases, but it also has some significant dissimilarities. For instance, like a relational database, SimpleDB is designed for storing tuples of related information. Unlike a relational database, SimpleDB does not provide data ordering services—that is left to the programmer.

Instead of tables, SimpleDB offers **domains**, which are schema-less. A domain is filled with **items—** similar to a row. Each item must have a unique name (which is not generated by SimpleDB) and may contain up to 256 **attributes**. Each attribute can have multiple values.

The following table is an example domain for a Person. Note that the "Phones" attribute contains more than one value for Bob and Joe.

Item Name	Age	Phones
Bob	21	(555)555-1212 (555)555-7777
Alice	22	(555)555-1213
Joe	32	(555)555-1432 (555)555-5432

TABLE 1 - TABULAR REPRESENTATION OF A SAMPLE DOMAIN.

All item names and attribute values in SimpleDB are strings. All comparisons during queries are string-based comparisons. This means that non-string data types (such as numbers), need to be converted into a single, consistent lexigraphic representation for storage (zero padding for numbers.) Queries are performed in a custom query language.

Queries are formed from **predicates**. A predicate is a simple comparison expression, surrounded by square brackets. For instance: *['Age' = '22']*. A query with that predicate will return *Alice* from TABLE 1. Predicates can be combined with the **union**, **intersection**, or **not** keyword. For instance: *['Age' = '22'] union ['Age' = '32']* will return *Bob* and *Joe* from TABLE 1.

The SimpleDB reference documentation at http://docs.amazonwebservices.com/AmazonSimpleDB/2007-11-07/DeveloperGuide/ contains more details of the query syntax.

PROBLEMS

Follow the instructions in the document "Getting Started with Amazon's Web Services for Lessons Plans" for getting an Amazon Web Services account and setting up your system.

1. The file "Internet_Users.txt" includes Internet usage information for UN member countries for the past 10 years in comma delimited format:

```
"Country or Area";"Year";"Value"
"Afghanistan";"2004";"25000"
"Afghanistan";"2003";"20000"
"Afghanistan";"2002";"1000"
```

Write a utility function to upload the values in the file into a single SimpleDB domain.

BONUS: for extra points, use Java Future http://java.sun.com/j2se/1.5.0/docs/api/java/util/concurrent/Future.html to perform the upload asynchronously.

2. Using the now-uploaded data, query the domain for all records after the year 2004. Print out the results.

3. Bonus: When SimpleDB processes a query and returns a result, it includes a *usage report* call **BoxUsage**. Attempt various ad-hoc queries against the data in the domain, such as intersections, nots, etc. Does the BoxUsage vary? Explain what may or may not cause BoxUsage changes.