CS 348: Introduction to Database Systems
(Winter 2015)
Assignment 2 (due 5pm on Thursday, March 5th)

1 Overview

You are to use your accounts and DB2 to implement the **BIBsearch** system. **BIBsearch** consists of two or three application programs with simple command line interfaces. The requirements for the programs are given in the following sections. A specification of the underlying schema for the database is also given and includes a schema diagram that indicates primary and foreign key constraints. Note that all projects must use this schema, and that projects themselves can be done by an individual alone or as part of a team consisting of two students. Either C or C++ must be used together with SQL and the static embedded SQL protocol to implement **BIBsearch**. NOTE: *As an aid to reducing the scope of the assignment, you are free to assume that none of the application programs is required to check for possible errors in input.*

2 Applications Programs

2.1 Program `bibauthor` (by individuals and teams)

This application is to print a list of publication records for each publication by an author with a name supplied as a string argument on the command line. In the case of articles, the publication identified by the pubid for the publication in which the article appears should be added to the list of requested publications so that a record for this publication is also printed. Note that a record for any publication should appear at most once in the output. Also, the publication information records should be printed to standard output.

The publications should appear according to a major sort on the year of the publication (latest first) and a minor sort on the name of the *first* author for the publication. In the case of publications without an author (i.e., journals and proceedings), the name of the first author should be assumed to be the zero-length string for sorting purposes.

The output format for each publication record is as follows.

```
Pubid:   pubid
Type:    { book | journal | proceedings | article }
Title:   title
```

In addition, for each particlar type of publication, the following information should also be printed.

- For books:

```
Authors:   author_1, ..., author_k
Publisher:   publisher
Year:   year
```

- For journals:

```
Volume:   volume
Number:   number
Year:   year
```

- For proceedings:

  ```
  Year:    year
  ```

- For articles:

  ```
  Authors:   author₁, ..., authorₖ
  In:    appearsin
  Pages:   startpage--endpage
  ```

2.2   Program `bibcontent` (by teams only)

This application is to print a list of publication records for each article appearing in a proceedings or journal identified by a pubid supplied as a string argument on the command line. The articles should appear according to a major sort on their starting page numbers, and should be printed with an output format matching that of `bibauthor` above.

2.3   Program `bibmaint` (by individuals and teams)

This application inserts new publications into the database or modifies existing publications already in the database. The input is given to the application as a list of insertion or update requests on the standard input in the following format (depending on the publication type):

```
author(aid#name)
authorurl(aid#url)
book(pubid#title#aid₁;...;aidₖ#publisher#year)
journal(pubid#title#volume#number#year)
proceedings(pubid#title#year)
article(pubid#title#aid₁;...;aidₖ#appearsin#startpage#endpage)
```

An update request is distinguished from an input request when the primary key values for the respective table already occur in the table. Note that, in the case of an update request for articles or books, the new list of authors completely replaces the old list of authors.

Sample input data for an execution of `bibmaint` is as follows:

```
proceedings(SIGMOD02#Proc. ACM SIGMOD Conference on the Management of Data#2002)
author(1#Peter Bumbulis)
author(2#Ivan T. Bowman)
authorurl(2#http://db.uwaterloo.ca/itbowman)
article(BB02#A Compact B-Tree#1;2#SIGMOD02#533#541)
```
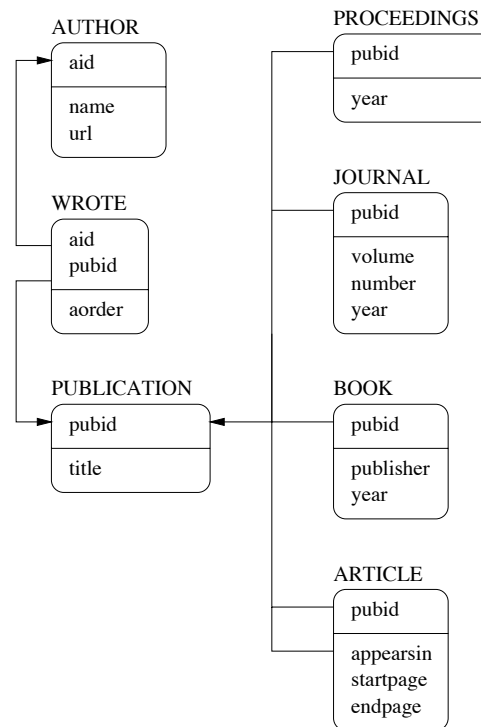
You can assume that all string constants in the input will not contain any of the characters "#", ";", "(" or ")". (HINT: read "man `strtok(3)`" for a way to help with parsing input commands.)

When an insertion or update succeeds, the application must print out the newly inserted or updated data in the same format used in the reporting applications. Also, recall that you are free to assume that error checking of input data is unnecessary.

3 Database Schema

(*attribute domains and class diagram*)

Attribute	SQL Domain
aid	integer not null
name	char(22) not null
url	char(42)
pubid	char(10) not null
aorder	integer not null
title	char(70) not null
publisher	char(50) not null
year	integer not null
volume	integer not null
number	integer not null
appearsin	char(10) not null
startpage	integer not null
endpage	integer not null

AUTHOR
aid
name
url

WROTE
aid
pubid
aorder

PUBLICATION
pubid
title

PROCEEDINGS
pubid
year

JOURNAL
pubid
volume
number
year

BOOK
pubid
publisher
year

ARTICLE
pubid
appearsin
startpage
endpage

(*additional constraints*)

There are a number of additional constraints that are always satisfied by a given bibliography. (Your programs may assume these constraints are always true, including `bibmaint`.) Let Pu, B, J, Pr and A denote the set of values for attribute "pubid" in tables PUBLICATION, BOOK, JOURNAL, PROCEEDINGS and ARTICLE, respectively, and let In denote the set of values for attribute "appearsin" in table ARTICLE. Then each of the following conditions is always true:

- $Pu \subseteq (B \cup J \cup Pr \cup A)$;

- $In \subseteq (B \cup J \cup Pr)$; and

- $|Pu| = (|B| + |J| + |Pr| + |A|)$,

where $|x|$ denotes a count of the number of elements in x. The last constraint ensures, for example, that B and J are disjoint. Finally, note that "pubid" values occurring in table WROTE will always refer to publications that are either books or articles.

4 Assignment Submission

A submission of the following items should be made by only one user ID in the case of teams.

1. Source listings of each of the two or three application programs comprising `bibsearch`. These should be submitted online in two or three files with names beginning with `bibauthor`, `bibcontent` and `bibmaint`, and ending with the appropriate suffix (i.e., *.sqc).

2. A shell script named `compile` should be submitted online. Typing "`./compile`" should suffice to compile all three application programs.

3. A short design document that outlines the design of your **BIBsearch** system consisting of no more than 2 pages. For teams, the document must include an extra page that outlines the division of tasks among the two team members as well as the user ID used for the electronic submission.