

Computer Science 597A
Fall 2008
First Take-home Exam
Out: 4:20PM Monday November 10, 2008
Due: 3:00PM SHARP Wednesday, November 12, 2008

Instructions:

This exam must be entirely your own work. Do not consult with anyone else regarding this exam. If you have questions about what is being asked, contact Professor LaPaugh. You are allowed to use the following reference materials while taking the exam:

- class notes posted on the “Schedule and Readings” page of the Web site for the course.
- the problem set solutions posted on the course Web site,
- your personal notes and graded problem set submissions,
- the course textbook: Abraham Silberschatz, Henry Korth, and S. Sudarshan, *Database System Concepts, Fifth Edition*, McGraw-Hill, 2006.
- the slides at the textbook web site: <http://www.db-book.com/>
- the errata at the textbook web site: <http://www.db-book.com/>

No other materials are allowed. Also, you are **NOT allowed to use online database interfaces.**

There are 5 problems, with point values indicated, totaling 100 points.
Be sure to give explanations for your answers.

If you wish to submit your work electronically, send Professor LaPaugh an email with your submission in a PDF or Word attachment by 3:00pm on Wednesday, November 12, 2008.

The following database will be used in Problems 1 and 2. This database holds information about doctors and patients in a large practice.

- For each doctor working in the practice, the doctor's name, license number, and specialty are recorded. A doctor's license number uniquely identifies him or her. Some doctors don't really specialize; their specialty is recorded as "general practice". A doctor is added to the database as soon as the doctor is hired.
- Each patient who visits the practice is recorded. The SS#, name, address, and primary insurance co. of the patient are recorded; the SS# uniquely identifies a patient. A patient is added to the database when he or she makes his or her first appointment with a doctor in the practice. Also recorded for a patient are the doctors the patient has seen. For each doctor a patient has seen, the date of the most recent past visit is recorded - unless the patient has not seen any doctor yet, in which case the date of the pending first appointment is recorded with the doctor to be seen.
- Some patients have specified a "primary care physician" from among the doctors in the practice to satisfy insurance requirements. No patient can designate more than one "primary care physician". For each patient who has selected a "primary care physician", the selected doctor is noted.

Problem 1 (20 points)

Part a. Draw an entity-relationship (ER) diagram representing the database. Represent the constraints on data and the constraints on relationships between data.

Part b. Are there any constraints you cannot represent? If so, why?

Problem 2 (20 points)

Part a. Translate your ER diagram from Problem 1 into a relational schema. Describe the relational schema using SQL statements to create the tables for the relations. Choose reasonable data types. Also indicate any primary key, candidate key and foreign key constraints. (Minor SQL syntax errors will be ignored.)

Part b. Are there any constraints you cannot represent? If so, why?

Problem 3 (30 points)

This problem refers to the movie database used in Problems 3 and 5 of Problem Set 1. The relational definition is taken from the posted solution to Problem 5 of that problem set (two minor changes: attribute `text` has been dropped as part of the primary key of `review` and the use of ‘`_`’ in attribute names has been omitted):

movie: (name, producer, reldate, rating)

theater: (name, loc, #screens, manager)

distributor: (name, proprietor, addr, tele)

review: (reviewer, publisher, date, text, name, producer, reldate)

name **not null**

producer **not null**

reldate **not null**

foreign key (name, producer, reldate) **references** *movie*

distributes: (name, producer, reldate, distribname)

foreign key (name, producer, reldate) **references** *movie*

foreign key (distribname) **references** *distributor*

showing: (name, producer, reldate, tname, tloc, distribname, start, end)

distribname **not null**

foreign key (name, producer, reldate, distribname) **references** *distributes*

foreign key (tname, tloc) **references** *theater*

Recall that attribute `reldate` is the release date of the movie and that relation ***showing*** records which movies are currently being shown at which theaters.

Your solutions for this problem will be graded primarily on correctness. However, you may lose some points for the inefficiency of a solution – that is, if the solution uses more operations than necessary, and it is fairly obvious that some operations are unnecessary. DO NOT spend time trying to optimize your expressions, but do ask yourself if all the operations serve a purpose.

Part a. Express the following queries with ***relational algebra expressions***. You may use any relational algebra operations, including joins and division.

- i:** Find the reviewer, movie name and text of all reviews for movies with rating ‘R’.
- ii:** Find the names and ratings of those movies currently being shown at every theater.
That is, each movie in the result is being shown at every theater.
- iii:** Find the movie names and distributors’ names for the most recently released movie or movies (all released on same date). Assume that attribute `reldate` is of type *date* and this type allows temporal comparison using $<$, \leq , $>$, \geq , $=$.

Part b: Express the following query in the *tuple relational calculus*:

Find the proprietors and telephone numbers of all distributors that are distributing only movies rated 'G'.

Part c: Express the constraint "The number of movies showing in a theater is equal to the number of screens" as a *SQL assertion*. (Minor SQL syntax errors will be ignored.)

Problem 4 (20 points)

Consider relational scheme $R = (A, B, C, D, E, F)$ with functional dependencies:

$A \rightarrow BC$

$D \rightarrow EF$

$F \rightarrow C$

Part a: What are the candidate keys for R ? Give a *brief* justification for your answer.

Part b: Is R in Third Normal Form (3NF)? If so, is it also in Boyce-Codd Normal Form (BCNF)? Give a *brief* justification for your answer.

Part c: For each of the two decompositions of R below, state whether or not the decomposition is lossless *and* whether or not the decomposition is dependency preserving. Give *brief* justifications for your answers.

i. $R_1 = (A, B, C)$; $R_2 = (A, D, E, F)$

ii. $R_1 = (A, B, C)$; $R_2 = (C, D, E, F)$

Problem 5 (10 points)

This problem refers to the XML specification of database *books* that we examined in class. The tree model of the schema for *books* is reproduced on the next page.

Write an XQuery query to find the name and date of birth of each author who has published exactly one book. The tag structure of the result should be:

```
<OneBookAuthor><name> ... </name><dob> ... </dob></OneBookAuthor>
```

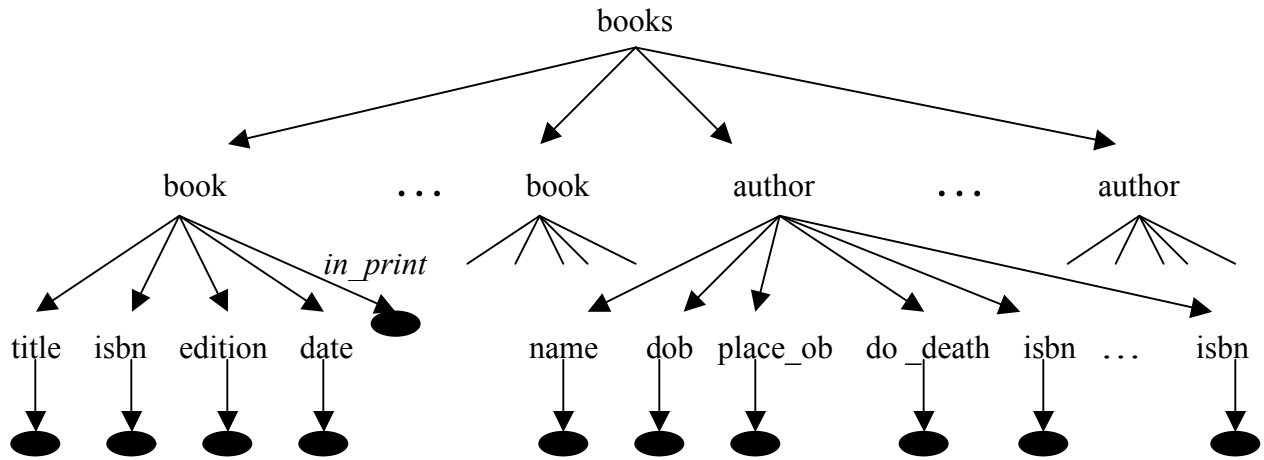
```
<OneBookAuthor><name> ... </name><dob> ... </dob></OneBookAuthor>
```

```
...
```

```
<OneBookAuthor><name> ... </name><dob> ... </dob></OneBookAuthor>
```

(Minor XQuery syntax errors will be ignored.)

Schema for tree model:



● Indicates content of type “xs:string”