CS411 Database Systems

Spring 2010, Prof. Chang

Department of Computer Science
University of Illinois at Urbana-Champaign

Midterm Examination

March 2, 2010

Time Limit: 75 minutes

- Print your name and NetID below. In addition, print your NetID in the upper right corner of every page.

Name: _____ **NetID:** _____

- Including this cover page, this exam booklet contains **9** pages. Check if you have missing pages.
- The exam is closed book and closed notes. No calculators or other electronic devices are permitted. Any form of cheating on the examination will result in a zero grade.
- Please write your solutions in the spaces provided on the exam. You may use the blank areas and backs of the exam pages for scratch work. Please do not use any additional scratch paper.
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Simplicity does count!*
- Generally, we think one minute per point is a reasonable allocation of time; so plan your time accordingly. *You should look through the entire exam before getting started, to plan your strategy.*

Problem	1	2	3	4	Total
Points	36	28	18	18	100
Score					
Grader					

Problem 1 (*36 points*) Misc. Concepts

For each of the following statements:

- for true/false choices, indicate whether it is *TRUE* or *FALSE* by circling your choice, and provide an explanation to justify;
- for short answer questions, provide a brief answer with clear explanation.

You will get *3point* for each correct answer with correct explanations, and ***no penalty* (of negative points) for wrong answers**.

Notice: To get full credit, you need to provide both correct answer, and correct explanations. If you simply choose True or False without explaining, you will automatically lose 1 point for the question. Moreover, since the questions here are true/false choices, or short answer questions, no partial credit is given for incomplete answers.

(1) *False*

An E-R diagram with m entities and n relationships will translate to $m+n$ tables.

$\Rightarrow$ *Explain*: Many-to-one relationships can often be merged and therefore reduce the overall number of tables needed.

(2) *True*

A table with two attributes, such as $R(A, B)$, must be in BCNF.

$\Rightarrow$ *Explain*: The only two non-trivial FDs for this relation will be $A \rightarrow B$ or $B \rightarrow A$, and both does not violate BCNF.

(3) *True* or *False*

An SQL query can often be translated into *multiple* relational algebra expressions.

$\Rightarrow$ *Explain*: The answer to this question can be true or false based on different assumptions. Our grading takes your assumption into consideration, and grades accordingly.

(4) Give a relation that is in 3NF but not in BCNF.

$\Rightarrow$ *Explain*: A relation R(A,B,C) with two FDs: $AB \rightarrow C$ and $C \rightarrow A$. The second FD violates BCNF, since C is not a superkey. However, it does not violate 3NF as A is part of a key (AB).

(5) *True*
In relational algebra, join is a *derived* operator.

$\Rightarrow$ *Explain*: It can be derived from Cartesian product and selection.

(6) *False*
Schema normalization often contributes to enhancement of query processing efficiency.

⇒ *Explain*: Schema normalization is mainly for reducing redundancy, rather than for improving query efficiency. In fact, it often introduces more query processing, as decomposed tables have to be joined online for answering queries.

(7) *True*
For relation $R(A, B, C)$, if A is a key, then the decomposition into $R(A, B)$ and $S(A, C)$ is lossless.

⇒ *Explain*: You can always reconstruct the original table (A,B,C) by joining (A,B) with (A,C), since A is a key. (many of your wrong answers give examples where A is not a key)

(8) *True*
In the following relation $R(A, B, C, D)$, the F.D. $AC \rightarrow B$ is not violated.

A	B	C	D
1	3	2	2
2	3	2	4
3	1	3	6
3	1	1	12

⇒ *Explain*: In the 4 tuples given, $AC \rightarrow B$ is not violated. You will also get full credit, if you choose False and argue that FD dependence can not be inferred from a set of given tuples (and should rather be derived based on the property of the relation).

(9) In the above relation $R(A, B, C, D)$, what are *possible* keys of the table?

⇒ *Explain*: Possible keys are D, or AC. You will also get full credit, if you argue we can not inferred keys from the set of 4 tuples.

(10) Provide a lossless-join decomposition of the above relation $R(A, B, C, D)$ (into to two relations). If there is no such decomposition, explain why.

⇒ *Explain*: You can decompose it into (ABC) and (ACD), just based on the 4 tuples given in the table. You will also get full credit, if you argue no such decomposition is possible, since we do not know the FDs for the relation.

(11) For the above relation $R(A, B, C, D)$, write a relational algebra expression to return the lowest value of D.

⇒ *Explain*: $\pi_D R - \pi_{R1.D}(R1 \bowtie_{R1.D>R2.D} R2)$ where R1 and R2 are just renames of R.

(12) What is the result of the following query, for the above relation $R(A, B, C, D)$?

```
select A, B
from R
where C >
      (select D from R where A = 3)
```

⇒ *Explain*: This query will report error. The reason is the subquery returns a scalar, and therefore can not be compared against a single data value. You would need to add "ANY" or "ALL" before the subquery for the query to run.

Problem 2 (*28 points*) Database Design

You have been asked to design a database for the university administration, which records the following information:

1. All students necessarily have a unique student ID, a name, and a university email address. Each student is also either an undergraduate or a graduate student.
2. Each graduate student has an advisor.
3. Each undergraduate student has a major.
4. Students take courses. A student may take one course, multiple courses, or no courses.
5. Each course has a course number, course name, and days of the week the course is scheduled.
6. Each course has exactly one head TA, who is a graduate student.
7. Every head TA has an office where he or she holds office hours.

(a) Draw an ER diagram for this application. Be sure to mark the multiplicity of each relationship of the diagram. Decide the key attributes and identify them on the diagram. Please state all assumptions you make in your answers. (*16 points*)

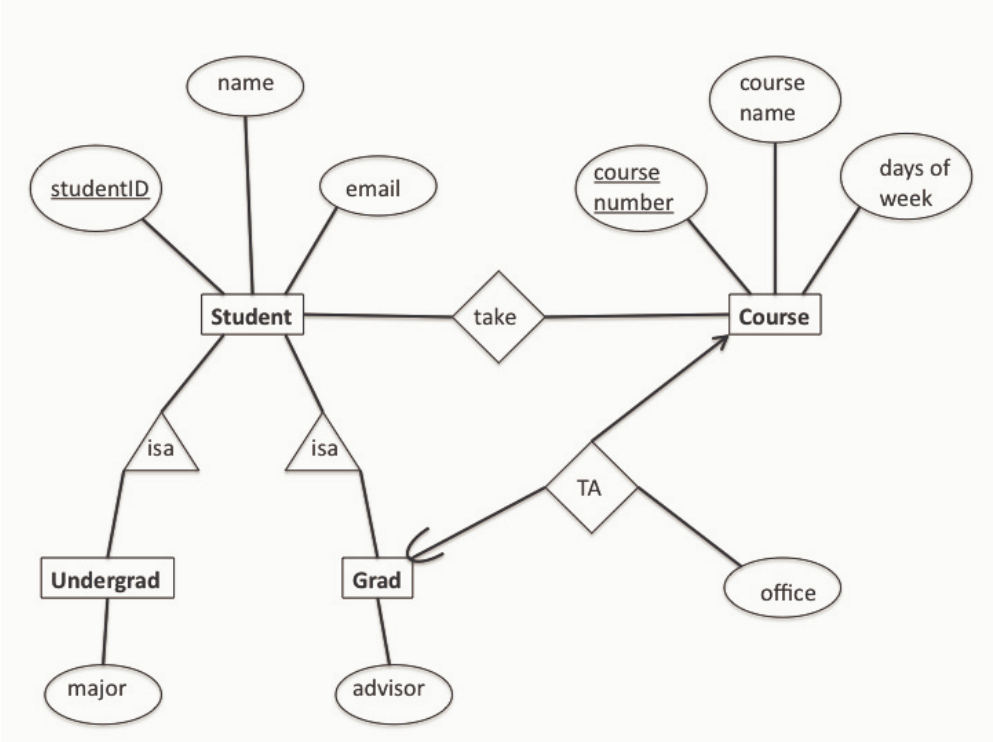

Figure 1: ER Diagram

(b) Translate your ER diagram into a relational schema. Select approaches that yield the fewest number of relations; merge relations where appropriate. Specify the key of each relation in your schema. (*12 points*)

Undergrad(<u>studentID</u>, name, email, major)
Grad(<u>studentID</u>, name, email, advisor)
TakeCourse(<u>studentID</u>, <u>course_number</u>)
Course(<u>course_number</u>, <u>studentID</u>, course_name, days_of_week, office)

Problem 3 (*18 points*) Relational Algebra

Nowadays, web search engine is widely used by people all over the world to find useful information. To analyze users' search behaviors, a *query log* stores the history of users' queries and clicked URLs.

There are two relations in a query log:

- $Request(Time, UserID, Query, Results)$, which lists the time, the userID, the query he/she requested, the number of search results for one search process.
- $Click(Time, UserID, Query, URL)$, which lists the time, the userID, the query he/she requested and the URL clicked on after request this query.

For both *Request* and *Click* relation, the attributes $(Time, UserID)$ forms the only key which implies one user can request only one query or click on only one URL each time.

Example instances of these two relations are given here:

Request Relation

Time	UserID	Query	Results
2010-02-12 19:00:00	U001	NBC	100
2010-02-12 19:01:00	U001	NBC Olympics	50
2010-02-12 18:54:00	U002	NBC 2010	80
2010-02-13 09:00:05	U001	Olympics wiki	70
2010-02-15 19:27:08	U003	Olympics 2010 medals	5
2010-02-16 13:24:45	U004	NBC Olympics	50
2010-02-16 13:25:55	U004	Vancouver 2010	95
2010-02-16 17:11:56	U002	NBC 2010	80
2010-02-20 20:13:45	U005	Olympics ski	54
2010-02-20 20:45:34	U005	Olympics Austria medals	7

Click Relation

QueryID	UserID	Query	URL
2010-02-12 19:01:06	U001	NBC Olympics	www.nbcolympics.com
2010-02-12 19:01:34	U001	NBC Olympics	www.nbcolympics.com/video
2010-02-12 18:54:01	U002	NBC 2010	www.nbcolympics.com
2010-02-15 19:27:22	U003	Olympics 2010 medals	www.vancouver2010.com
2010-02-15 19:29:01	U003	Olympics 2010 medals	www.vancouver2010.com/olympic-medals/
2010-02-16 13:25:58	U004	Vancouver 2010	www.vancouver2010.com
2010-02-16 17:12:56	U002	NBC 2010	www.nbcolympics.com
2010-02-20 20:14:00	U005	Olympics ski	www.usskiteam.com
2010-02-20 20:45:40	U005	Olympics Austria medals	en.wikipedia.org/wiki/Austria_at_the_Olympics
2010-02-20 20:45:50	U005	Olympics Austria medals	www.vancouver2010.com

Note that not all the queries have corresponding clicked URLs since a user might not be interested in any returned URL. Sometimes user may click on multiple URLs using one query.

Though different from the actual case, you can assume that the number of returned search results associated with the same query doesn't change no matter when and who request this query.

Answer the following queries using relational algebra. Your answer should work for any instance of the database, not just this one.

(a) List all the queries that do not have any corresponding URL. (*8 points*)

Solution:
$\pi_{Query}Request - \pi_{Query}Click$

(b) List all the queries that have been requested by *different* users. (*10 points*)
Solution:
$\pi_{Request_1.Query}(Request_1 \bowtie_C Request_2)$
where, $Request_1$ and $Request_2$ are both type of $Request$
$C = (Request_1.Query = Request_2.Query$ AND $Request_1.UserID <> Request_2.UserID)$

Problem 4 (*18 points*) SQL

Consider the same schema as Problem 3. Answer the following queries using SQL. Your statements should work for any instance of the database, not just this one. You do not need to remove duplications in your results.

(a) List all the users (userID) who either requested the query "NBC 2010" or clicked on the URL "www.nbcolympics.com." (*8 points*)

Solution:

```
select UserID from Request where Query = "NBC 2010"
union
select UserID from Click where URL = "www.nbcolympics.com"
```

(b) Return the query whose corresponding URLs contain "www.vancouver2010.com" and, among these queries, has the minimum number of returned search results. (*10 points*)

Solution:

```
select Request.Query from Request, Click
where Request.Query = Click.Query
and Click.URL = "www.vancouver2010.com"
and Results <= All
(select Results from Request, Click
where Request.Query = Click.Query
and Click.URL = "www.vancouver2010.com")
```