CS411 Database Systems

Fall 2007

Department of Computer Science
University of Illinois at Urbana-Champaign

Midterm Examination

October 17, 2007

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 **10** 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!*

Problem	1	2	3	4	5	6	Total
Points	14	11	15	16	28	16	100
Score							
Grader							

Turn over the page when instructed to do so.

Problem 1 Basics (14 points)

For each of the following statements, indicate whether it is TRUE or FALSE by circling your choice. If you change your mind, cross out both responses and write "True" or "False." You will get 1 point for each correct answer, 0 point for each incorrect answer.

(1) *True False*
In E/R model, only entities can have attributes.

(2) *True False*
Entity sets are weak when their key attributes come from other classes to which they are related.

(3) *True False*
All relations in BCNF also in 3NF.

(4) *True False*
BCNF decomposition is always dependency preserving.

(5) *True False*
In relational algebra, the project operator may change the number of tuples in set semantics.

(6) *True False*
The logic of conditions in SQL is 2-valued logic: TRUE, FALSE.

(7) *True False*
$< tuple >$ IN $< relation >$ is true if and only if $< relation >$ is not empty.

(8) *True False*
A view is updatable if it is defined by selecting some attributes from a single relation.

(9) *True False*
In SQL, an INSERT statement always requires all column values to be specified.

(10) *True False*
In SQL, deletes can be cascaded to enforce foreign key constrains.

(11) *True False*
SQL applies bag semantics for intersect, union, and differece operations by default.

(12) *True False*
By defining a trigger statement in SQL, we can always make a view updatable.

(13) *True False*
Authorization in SQL supports discretionary access control.

(14) *True False*
In SQL, we can define an authorization policy that limit access to a particlar tuple in a relation.

Problem 2 ER Diagram (11 points)

Consider the following information about a baseball legend:

- Teams have a TID and a name
- Players have a name and an age
- Pitchers are a type of Players. Each pitcher has attributes W (win games), L (loss games), and ERA (earned run average)
- Batters are a type of Players. Each batter has attributes AVG (batting average) and HR (home runs)
- Each player plays for exactly one team; a team can have many players
- Each player has exactly a contract relationship with his team, and a contract contains years and salary.
- Games have a date (on which the game was played) and a score (e.g., "3:2")
- A game has exactly a winning team and has exactly a losing team; A game can be uniquely identified by its winning team, losing team, and the date.

Referring to the following E/R diagram in Figure 1, we have identified all the entities and their attribues. Now please complete the rest of the diagram by adding all the relationships between entities. Don't forget to indicate multiplicity constraints and supporting relationships if there is any.

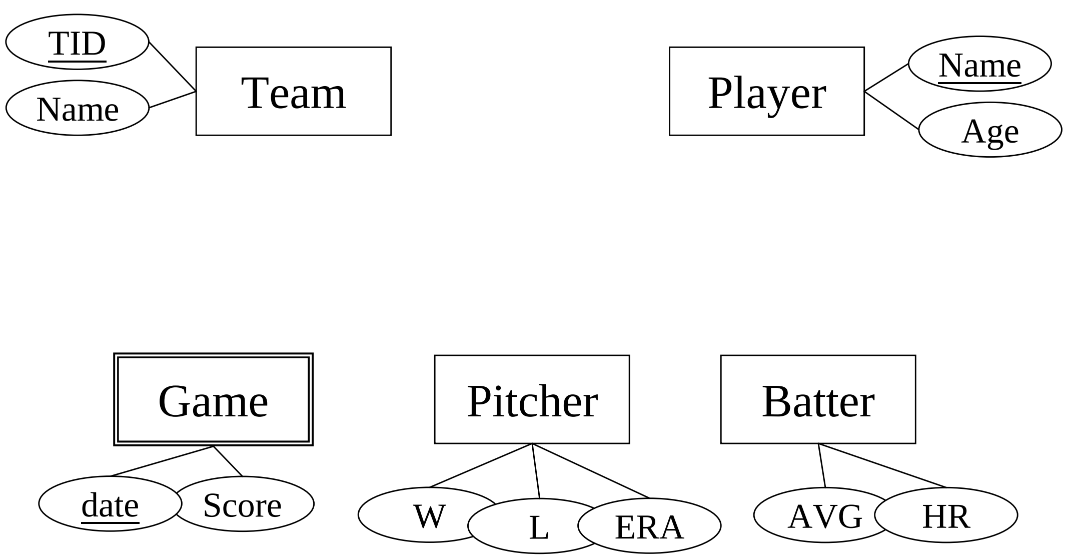

Figure 1: ER Diagram

<u>**Problem 3**</u> Functional Dependencies and Normal Forms (15 points)

Consider a relation $\mathsf{R}(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E})$, with given FD's $\mathsf{AB} \rightarrow \mathsf{C}$, $\mathsf{BC} \rightarrow \mathsf{D}$, $\mathsf{CD} \rightarrow \mathsf{E}$, $\mathsf{DE} \rightarrow \mathsf{A}$.

(i) Determine all the keys of R.
(Hint: There are three keys and you don't not need to list superkeys that are not keys)

(ii) List which FDs violate 3NF if any.

(iii) List which FDs violate BCNF if any.

(iv) Decompose R using BCNF decomposition. Indicate your working and summarize your final set of relations. Before applying a FD to decompose a relation, add to the right side as many attributes as are functionally determined by the attributes on the left.

Problem 4 Relational Algebra (16 points)

Consider the following schema that is used for CS411 course maintenance, including the students enrolled, their project teams and midterm grades:

Student(UIN,Name,Department)
ProjectTeam(TeamName,UIN)
Midterm(UIN,Grade)

The key fields are underlined. Write relational algebra expressions for the following queries. Use only operators join $\bowtie$, cross product $\times$, difference $-$, rename ρ, project π, and select σ. The Selection expression may evaluate a simple boolean expression for each tuple. e.g. $= < > \neq \vee \wedge$ but not group operations, e.g. *min*, *max*.

(i) Find the Names of students who are in the team with TeamName 'BEE'

(ii) Find the UIN of students who have not been in any team yet.

(iii) Find the TeamNames of teams whose member students are all from Department 'CS'

(iv) Find the Names of student(s) obtaining the highest midterm grade.

Problem 5 SQL Query (28 points)

Consider the following relations that the online bookstore, CS411.com, maintains. You can assume a reasonable data type for each attribute of those relations and that SQL works in set semantics in this problem; you do not have to worry about duplicate tuples in a relation. Underlined attributes are keys of those relations. The attributes *cid* and *isbn* in Buy relation are foreign keys referring to *cid* in Customer and *isbn* in Book respectively. The attribute *isbn* in Author is a foreign key referring to *isbn* in Book.

Book(<u>isbn</u>, title, publisher, price)
Author(<u>assn</u>, aname, isbn)
Customer(<u>cid</u>, cname, state, city, zipcode)
Buy(<u>tid</u>, cid, isbn, year, month, day)

(i) Make a list of the names of customers who live in Illinois and spent more than $5,000 in year 2000.

(ii) Find the name of the author who sold his or her books most in year 2006. Note that a single author could write more than one book, and we need to count all of the copies of his books. Also, if multiple authors coauthor the same book, we consider that each author of that book sold the number of copies of that book sold.

Book(isbn, title, publisher, price)
Author(assn, aname, isbn)
Customer(cid, cname, state, city, zipcode)
Buy(tid, cid, isbn, year, month, day)

(iii) Create a view *FriendsOfBob* that contains a list of people (i.e., a list of *cid* attribute values in Customer) who share a common interest with Bob whose cid = 12345. We consider that two persons share a common interest if they purchased more than 20 same books before.

(iv) Make a list of recommended books (i.e., a list of isbn attribute values in Book) for Bob using view *FriendsOfBob* in Problem 5(iii). (If you cannot answer the question in Problem 5(iii), you can assume that such a view exists and use that view to answer this question.) We recommend a book for Bob if his possible friend in *FriendsOfBob* bought that book before and Bob has not bought that book yet.

Problem 6 Datalog (16 points)

Consider two relations $Likes(name1, name2)$ and $Dislikes(name1, name2)$. Atom $Likes(Bob, Alice)$ means that Bob likes Alice and atom $Dislikes(Alice, Dave)$ means that Alice dislikes Dave. Note that $Likes(Bob, Alice)$ does not necessary imply the negation of $Dislikes(Bob, Alice)$ (i.e., $NOT\ Dislikes(Bob, Alice)$) in this problem.

Define relation $RemoteLikes(p, q)$ where person p likes person q by way of $Likes$ relation transitively without involving any pair of two people in a $Dislikes$ atom in that transitive chain of $Likes$ tuples. For example, if $Likes(Bob, Alice)$ and $Likes(Alice, Michael)$ hold and neither $Dislikes(Bob, Alice)$, $Dislikes(Alice, Michael)$, nor $Dislikes(Bob, Michael)$ hold, then $RemoteLikes(Bob, Michael)$ hold. More precisely, if we derive $RemoteLikes(p_0, p_n)$ from a transitive chain of atoms $Likes(p_0, p_1), Likes(p_1, p_2), \ldots, Likes(p_{n-2}, p_{n-1}), Likes(p_{n-1}, p_n)$, then there is no atom $Dislikes(p_i, p_j)$ where $0 \leq i < j \leq n$.

(i) Define the $RemoteLikes$ relation using Datalog rules.

(ii) Show that negation in the Datalog rules in Problem 6(i) is stratified or not.