CSE255 Introduction to Databases – Midterm Exam

Name: _____

Problem	Points	Score
1	15	
2	15	
3	15	
4	15	
5	15	
Total	75	

Please show all work to receive ANY credit!!!!
Roughly equate 15 points to fifteen (15) minutes of time/effort.

Consider the BBALL relational schema, with assumptions, below for tracking UConn's basketball program and statistics over the years, that will be used in Problems 1 and 2.

PLAYER(PLName, PFName, StartYear, NumYears, UniformNumber);
COACH(CLName, CFName, StartYear, EndYear);
TEAM(TeamID, Year, Squad);
ROSTERS(TeamID,PLName, CLName);
RSRECORD(TeamID, Wins, Losses);
PORECORD(TeamID, Wins, Losses);
STATISTICS(PLName, TeamID, PPG, RPG, APG);
TITLES(TeamID, TitleType);

- Last names of coaches and players are unique.
- The Year, StartYear, and EndYear attributes have values such as, 1968, 1971, 1994, 1999, 2001, etc. In the TEAM table, a year such as 1997 means the season that finishes in 1997 but started in November 1996. In all other tables, it represents when they started in the program, e.g., Tony Roberterson, a senior, started in 1999. EndYear is null for active coahes.
- NumYears represents the number of complete years a player is in the program, e.g., Tony Robertson would have NumYears = 3 since he hasn't completed his fourth year.
- STATISTICS contains values for each player ONLY after the season is complete, e.g., Tony Robertson will have just three tuples in STATISTICS.
- ROSTSERS is a ternary relationship among PLAYER, COACH, TEAM.
- There is one COACH per year, i.e., coaches are not fired during the season.
- RSRECORD is for the regular seasonrecord; PORECORD is for the playoffs.
- PPG, RPG, and APG are Points, Rebounds, and Assists Per Game, respectively.
- The attribute Squad has the values Mens and Womens, and the attribute TitleType has the values BigEastRS, BigEastCC, NCAA, NIT.

1. **(15 points) Relational Algebra - Only do 1(c) if you finish entire exam!**

   This question requires you to write relational algebra queries. Note that you are allowed to use variables (relations) to hold intermediate results if your answer is in multiple steps.

   (a) **(5 points)** List the full names of all mens players that have/had the uniform number 5.

   (b) **(5 points)** List the full names and points per game of all mens players that averaged more than 20 points per game and at least 5 rebounds per game in one year on teams with winning regular season records.

(c) **(5 points)** List the full names and uniform numbers of all womens players that played for Geno Auriemma in 1996.

2. **(15 points) SQL Queries - Only do 2(c) if you finish entire exam!**

   Specify the following SQL Queries on the BBALL schema given previously.

   (a) **(5 points)** Find the coaches (full names) of all teams that won a NCAA title without winning a BigEastRS (Regular Season) title.

(b) **(5 points)** Find the average PPG, RPG, and APG for their entire careers, for all mens players (full names) who started their careers during the 1980s.

(c) **(5 points)** Find the total wins of all coaches (full names) for their entire careers.

Problems 3 and 4 are based on a revised version of the BBALL Schema as given below, with all indication of keys removed. Notice that: PLAYER and COACH have been combined using boolean flags (PFlag, CFLag) to distinguish unique attributes; TEAM, ROSTERS, and TITLES have been combined into a single relation; and, RSRECORD and PSRECORD have been combined into a RECORD relation, with the attributes changed to represent wins and losses for the regular season (RSWins and RSLosses) and the entire season (TTLWins and TTLLosses), requiring playoff wins and losses to now be calculated. Other than this change, all other assumptions as given for the original BBALL Schema still apply.

PLAYERCOACH(LName, FName, StartYear, PFlag, NumYears, UniformNumber, CFlag, EndYear);
TEAMROSTERSTITLES(TeamID, Year, Squad, PLName, CLName, TitleType);
RECORD(TeamID, RSWins, RSLosses, TTLWins, TTLLosses);
STATISTICS(PLName, TeamID, PPG, RPG, APG);

3. **(15 points) Update Anomalies**

   For each relation, discuss the anamolies that exist in the revised BBALL Schema. If anomalies do not exist in a relation, explain why (1 or 2 sentences). If anomalies do exist in a relation, identify which type(s) exist, namely, insertion, deletion, or modification. For each type that does exist, describe an example that causes the anomaly. Focus on anomalies within a single relation rather than across multiple relations.

Continue Problem 3 solution on this page.

4. **(15 points) Functional Dependencies**

   (a) **(15 points)** Define all functional dependencies (FDs) for the revised BBALL schema. List your results on a relation-by-relation basis. Be very specific - do not simply specify that a single attribute determines all others. **Make sure that you use arrow notation for FDs, i.e., X Y $\rightarrow$ Z, SSN $\rightarrow$ EmpName EmpAddr.**

(b) **BONUS (5 points)** Multi-valued dependencies occur when one attribute can determine multiple values of another attribute. For example, in the relation

EMPLOYEE(EmpName, ProjName, DependentName)

there are multi-valued dependencies EmpName $\Longrightarrow$ ProjName (an Employee works on multiple projects) and EmpName $\Longrightarrow$ DependentName (an Employee has multiple dependents). Identify all multi-valued dependencies for the revised BBALL schema.

5. **(15 points) Reverse Engineering: Relational to ER Diagram**

One of the tasks often associated with database maintenance is documenting the structure of a relational schema that has been in use for a long period of time (prior to EER and UML design techniques). In such a situation, an EER diagram can be reverse engineered from a set of relational tables. Consider the schema below:

BOOK(BookId, Title, PublisherName);
PUBLISHER(Name, Addrress, Phone);
BOOKAUTHORS(BookId, SSN, LastName, FirstName, Email);
BORROWER(CardNo, SSN, LastName, FirstName, Address, Phone);
BOOKCOPIES(BookId, BranhId, NoOfCopies);
BOOKLOANS(BookId, BranhId, CardNo, DateOut, DueDate);
LIBRARYBRANCH(BranchId, BranchName, Address);

Reverse engineer this schema into a EER diagram. Notice that I have been very careful to utilize the same attribute names to indicate foreign keys. Note also that all primary keys are underlined. **You must have inheritance in your solution!**

Continue Problem 5 solution on this page.