

CSE255 Introduction to Databases – Midterm Exam

Name: _____

Problem	Points	Score
1	20	
2	15	
3	10	
4	10	
5	20	
Total	75	

Please show all work to receive ANY credit!!!!

Roughly equate 15 points to fifteen (15) minutes of time/effort.

1. (20 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 modified BBALL schema below:

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

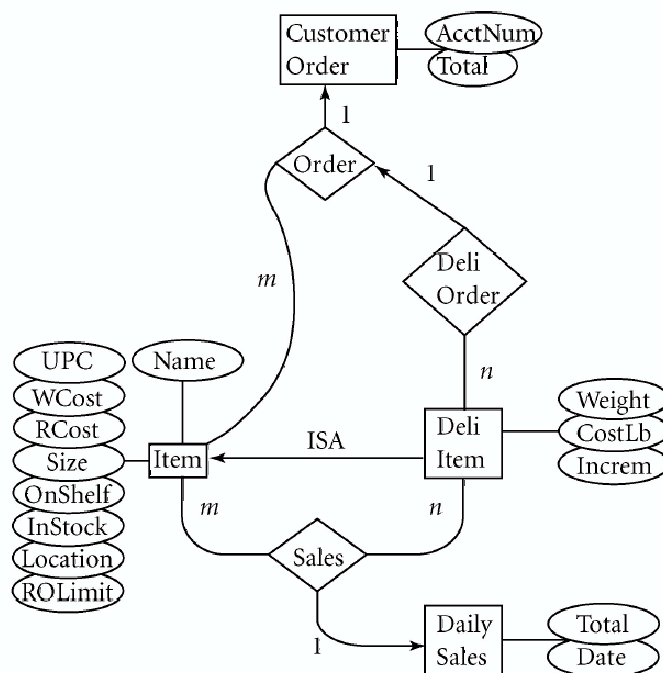
Note that a PersonID has been added to players and coaches, and is unique across the two relations. This changes the assumptions as follows: Players and Coaches can have the same name; a player one year can be a coach in a later year (but not in the same year).

Reverse engineer this schema into a EER diagram. **You must have one example of disjoint inheritance and one example of overlapping inheritance in your solution!**

Problem 1 solution on this page.

2. (15 points) EER to Relational Conversion

Consider the EER diagram shown below, with all of the various entities, relationships, attributes, keys, etc. This diagram is used to track the items at a supermarket, the sales that occur daily, and the orders for customers. Convert this diagram using the algorithm from Chapter 9 into a set of equivalent tables. Make sure that you indicate the step of the algorithm that you are utilizing as you perform the conversion.



Keys:
UPC for Item
AcctNum for Customer Order
Date for Daily Sales

Problem 2 solution on this page.

3. (10 points) Relational Algebra

This question uses a slightly modified version of the TV/Movie schema, namely:

TVRoles(PersonID, RoleID, ShowID, EpisodeID, EmmyFlag, FirstRole);

MovieRoles(PersonID, RoleID, ShowID, OscarFlag, FirstRole);

where FirstRole is True if this is the persons first role ever in a TV Show/Movie and false for all other subsequent roles. Note that you are allowed to use variables (relations) to hold intermediate results if your answer is in multiple steps.

- (a) **(5 points)** Find the names of all TV directors (first and last) who were (are) also movie actors.

- (b) **(5 points)** Find the Movie names and gross revenues, and names of all Movie directors (first and last) who had roles in at least 3 TV episodes.

4. **(10 points) SQL Queries**

This question uses the same modified version of the TV/Movie schema, namely:

TVRoles(PersonID, RoleID, ShowID, EpisodeID, EmmyFlag, FirstRole);

MovieRoles(PersonID, RoleID, ShowID, OscarFlag, FirstRole);

- (a) **(5 points)** Find the names (first and last) and role names (first and last) for all actors that had their first movie role that was before their first TV role.

- (b) **(5 points)** Find the show names and names of all TV directors (first and last) who have directed at least 4 different TV episodes for which they have won emmys, sorted by TV show name.

5. **(20 points) Functional Dependencies** Consider the revised version of the BOOK Schema from the Spring 2003 sample exam:

BOOKS(BookId, Title, PubName, PubAddress, PubPhone);
BOOKAUTHORS(BookId, SSN, LastName, FirstName, Email);
BORROWER(CardNo, SSN, LastName, FirstName, Address, Phone);
LIBRARYBRANCH(BranchId, BranchName, Address);
BOOKCOPIES(BookId, BranchId, NoOfCopies);
BOOKLOANS(BookId, BranchId, CardNo, DateOut, DueDate);

The content of this schema is self explanatory - dealing with books in the BOOKS table including the title and publisher information, their authors in the BOOKAUTHORS table, the people that borrow books in teh BORROWERS table, the different library branches, the copies of each book at each branch, and the books that are loaned to customers by branch.

- (a) **(11 points)** Define functional dependencies (FDs) for ONLY the tables **BOOKAUTHORS**, **BORROWER**, and **LIBRARYBRACH**. 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) **(9 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 $\text{EmpName} \twoheadrightarrow \text{ProjName}$ (an Employee works on multiple projects) and $\text{EmpName} \twoheadrightarrow \text{DependentName}$ (an Employee has multiple dependents). Identify all multi-valued dependencies in **ONLY** the tables **BOOKS**, **BOOKCOPIES**, and **BOOKLOANS**.