CSCI585 Summer '18 Midterm Exam

June 11th, 2018

CLOSED book and notes. No electronic devices. DO YOUR OWN WORK. Duration: 2 hours. If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS. If you continue working on the exam after time is up you will get a 0.

Solutions are displayed in red font!

Signature: ___________________________________

Problem Set	**Number of Points**
Q1	5
Q2	5
Q3	5
Q4	5
Q5	5
Q6	5
Q7	5
Total	**35**

Q1. (5 points total) ER MODELING

A. (2 points) Explain the difference between weak and strong entity and provide example.

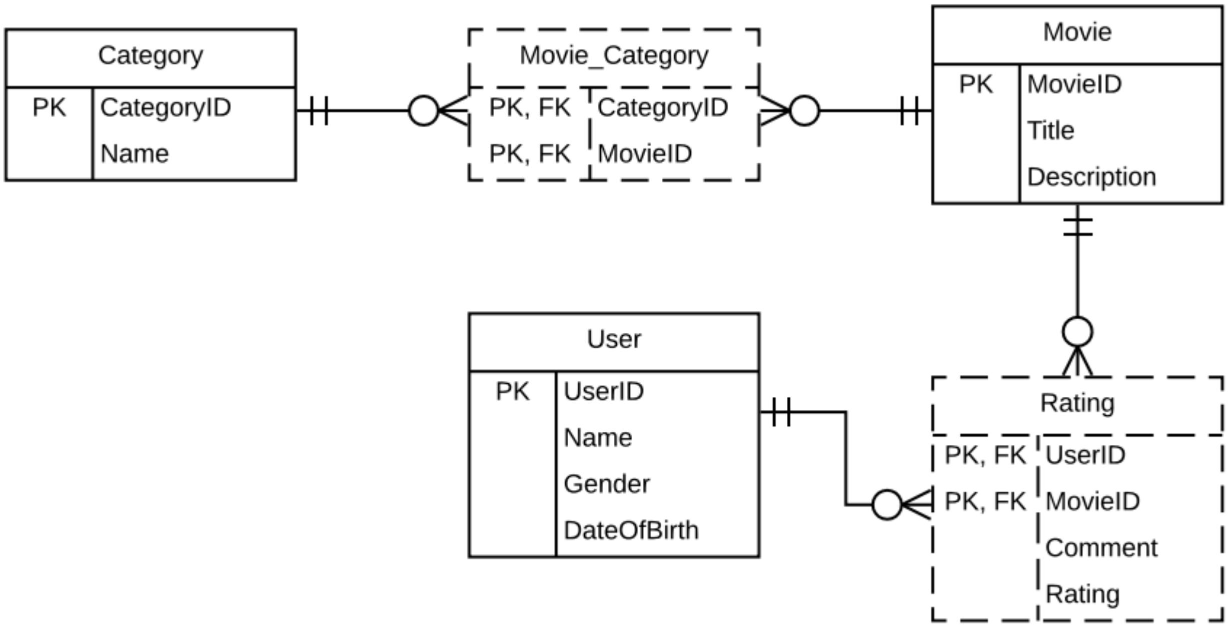
Weak entity: Entity that depends on another entity to exist. Its primary key contains the primary key of another entity.
Strong entity: Entity that **may** exist independently with other entities.

Example:

Department(DeptID, Name, Address, Phone)
Employee(DeptID,EmplID, Name, Age, Gender)

Department is a strong entity, while Employee is a weak entity.

B. (3 points) Design ERD using Crow's foot notation for the following description:
A movie-rating site like the Internet Movie Database (IMDB) has multiple movie categories. Each category has zero, one or more movies. One movie may belong to multiple categories. A person must create an account to rate a movie. To rate a movie, he/she provides a comment along with a rating star ranging from 1-5. A person can rate any numbers of movies he/she likes, but cannot have multiple ratings for the same movie (hence, he/she may provide either no rating or one rating per movie).



Q2. (5 points total) SQL

Write the following queries for an online store that sells books. Below are the tables for the store, primary keys are underlined, foreign keys are *italic*. AvailableCount attribute in Book table records the number of copies of a book that are available for sale. Quantity attribute in Order table records the number of copies of a book that is ordered by a customer. For simplicity, we assume a customer can only order a book in one order (he can order more copies of it, however).
Category(CategoryID, Name)
Book(ISBN, *CategoryID*, Title, Author, Description, PublishDate, AvailableCount, Price)
Customer(CustomerID, Name, Age, Gender, Balance)
Order(OrderID, *CustomerID*, *ISBN*, Quantity, Total, OrderDate)

A. (1 point) List up to 10 books in categories “Science fiction” or “Romance”. For each book, show all its attributes.

```
SELECT ISBN, BOOK.CAT_ID, TITLE, AUTHOR, DESCRIPTION, PUBLISH_DATE, AVAILABLE_COUNT, PRICE
FROM CATEGORY, BOOK WHERE BOOK.CAT_ID = CATEGORY.CAT_ID AND CATEGORY.NAME IN ('SCIENCE
FICTION',’ROMANCE’) LIMIT 10;
```

B. (2 points) List all books customer id 5 ordered. For each book, show the ISBN, title, the number of copies he/she bought and the total amount he/she spent on that book.

```
SELECT BOOK.ISBN, TITLE,  SUM(QUANTITY), SUM(TOTAL) FROM BOOK, BOOK_ORDER WHERE
BOOK.ISBN = BOOK_ORDER.ISBN AND CUSTOMER_ID = 5 GROUP BY ISBN;
```

C. (2 points) List up to five customers that ordered “Horror” books the most (order counts).

```
SELECT CUSTOMER_ID FROM CATEGORY, BOOK, BOOK_ORDER WHERE BOOK.CAT_ID =
CATEGORY.CAT_ID AND CATEGORY.NAME = 'HORROR' GROUP BY CUSTOMER_ID ORDER BY
COUNT(ORDER_ID) DESC LIMIT 5;
```

Q3. (5 points total) NORMALIZATION

A. (1 point) Explain the difference between 2NF and 3NF by providing example of 2NF but not 3NF.

While 2NF is converted from 1NF by removing the partial dependencies, 3NF is converted from 2NF by removing the transitive dependencies.

Example:

(Department, Employee, JobType, Salary)

There is no partial dependency, so the table is 2NF.

However, there is transitive dependency: (JobType → Salary), to the table is not 3NF.

B. (4 points) Convert the following table to 3NF.

Show dependency diagram for each form and identify the primary key for each table.

Season	Day	Home Team	Home Team City	Away Team	Away Team City	Result	Scored By	Of Team	Born Year
2016	June 7	MU	Manchester	Chelsea	London	2-1	A. Herrera	Manchester	1989
2016	June 7	MU	Manchester	Chelsea	London	2-1	Rashford	Manchester	1997
2016	June 7	MU	Manchester	Chelsea	London	2-1	Hazard	Chelsea	1991
2016	Oct 8	Arsenal	London	Liverpool	Liverpool	1-0	O. Giroud	Arsenal	1986
2015	Apr 15	Everton	Liverpool	MU	Manchester	1-1	Pogba	Manchester	1993
2015	Apr 15	Everton	Liverpool	MU	Manchester	1-1	W. Rooney	Everton	1985

1NF(Season, Day, HomeTeam, HomeTeamCity, AwayTeam, AwayTeamCity, Result, ScoredBy, OfTeam, BornYear)

Partial Dependencies: (Season, Day → HomeTeam, AwayTeam, Result)

Transitive Dependencies:

(HomeTeam → HomeTeamCity)

(AwayTeam → AwayTeamCity)

(ScoredBy → OfTeam, BornYear)

3NF:

TEAM	**CITY**

DAY	**SEASON**	**HOME_TEAM**	**AWAY_TEAM**	**RESULT**

DAY	**SEASON**	**PLAYER**

PLAYER	**OF_TEAM**	**BORN_YEAR**

Q4. (5 points) TRANSACTION MANAGEMENT

A. (4 points) Assume PRODUCT table has a record for a notebook, whose product ID is 996 and its quantity on hand is 10. A developer executes a SQL statement "UPDATE product SET quantity = 6 WHERE id = 996" through the program. However, after the operation, the developer runs SELECT query and finds out that this notebook's quantity is still set to 10. The developer makes sure the update statement was executed and he retrieves the data from the table directly. What could be two possible reasons for this situation?

1. That statement is rollbacked after the execution
2. Lost updates: some other transaction may update the notebook's quantity at the same time.

B. (1 point) Locking is widely used in concurrency control in databases, but we need to make sure there are no deadlocks between transactions. Briefly explain (or give example) how deadlock occurs in transaction management.

A deadlock occurs when two transactions wait indefinitely for each other to unlock data.

Q5. (5 points) QUERY OPTIMIZATION

A. (2 points) The following two queries are performing the same function. Which one do you think is more efficient? Why? (2 points)

SELECT id FROM users WHERE DATEDIFF(MONTH, registerDate, '2015-04-28') < 0;

SELECT id FROM users WHERE registerDate > '2015-04-28';

The second one is more efficient. The first one uses a runtime function, and the database has to visit all rows to retrieve the required data.

B. (2 points) Consider the query discussed during class and pertaining to the schema discussed in class. List at least two ways to optimize this query.

```
SELECT      CUS_CODE, MAX(LINE_UNITS*LINE_PRICE)
FROM        CUSTOMER NATURAL JOIN INVOICE NATURAL JOIN LINE
WHERE       CUS_AREACODE = ‘615’
GROUP BY    CUS_CODE;
```

1. Filter by area code 615 before joining with customer
2. Store line_units*line_price in a column called line_total instead of deriving it.

C. (1 point) In order to retrieve the orders that are placed by the residents of cities whose name starts with “Cha”, the developer writes a query:
SELECT * FROM orders WHERE city LIKE ‘%Cha%’.
Is there a problem with this query? If yes, write down your optimized version.

Yes, the query will also pull unexpected results, cities that contain “Cha” but not starts with “Cha”.

The optimized query is SELECT * FROM orders WHERE city LIKE ‘Cha%’

Q6. (5 points) DISTRIBUTED DATABASES

A nationwide commercial bank has many branches in each state. At the end of each month, the bank's audit department wants to know the deposits and loans for each state. All the data is stored in a distributed database.

A. (2 points) In order to meet the requirement of the audit department, which data fragmentation technique should be used, and how to do that?

Horizontal fragmentation. Each sub-table stores its state's records (rows).

B.(2 points) For each branch, the headquarters will establish a monthly goal according to the operational data (ie. deposits and loans). Assume that there is a table Branch which contains the following attributes:

branchId, address, manager, deposit_amount, loan_amount, audit_Time

If we only consider meeting the requirements of the headquarters, which data fragmentation technique should be used, and how?

Vertical fragmentation.

One fragmented table contains branchId, deposit_amount, loan_amount, audit_Time.

BranchId, address, manager are stored in another fragmented table.

C. (1 point) What are the two phases in two-phase commit protocol (2PC)?

1. Preparation
2. The final COMMIT.

Q7. (5 points) INTRODUCTION AND DATA MODELING

A (1 point) Explain how division (one of relational algebra operations) works. Feel free to use example.

It answers queries about one set of data being associated with all values of data in another set (looks for commonality).

B (1 point) What does DDL stand for?

Data Definition Language. The language that allows a database administrator to define the database structure, schema, and subschema.

C (1 point) Provide an example of discipline specific database.

Any database that contains data focused on specific subject areas (answers will vary).

D. (1 point) In which database (operational database or data warehouse) data doesn't get modified?

Data warehouse (analytical database).

E. (1 point) Explain one of the many functions that DBMS must support.

Answers will vary.

BONUS!!! (1 point) What was your favorite part of Science documentary shown in class? If you have seen the entire movie, feel free to reference the part not displayed in class.
Your mileage may vary. ☺