CS34800 Fall 2016 Midterm 1, 26 September, 2016
Prof. Chris Clifton

Turn Off Your Cell Phone. Use of any electronic device during the test is prohibited. As previously noted, you are allowed notes: Up to two sheets of 8.5x11 or A4 paper, single-sided (or one sheet double-sided).

Time will be tight. If you spend more than the recommended time on any question, **go on to the next one**. If you can't answer it in the recommended time, you are either giving too much detail or the question is material you don't know well. You can skip one or two parts and still demonstrate what I believe to be an A-level understanding of the material.

Note: It is okay to abbreviate in your answers, as long as the abbreviations are unambiguous and reasonably obvious.

I would put the A/B borderline around 28, and a B/C borderline around 22-23. If your score is below 20, we should talk.

1 Query Result (10 minutes, 8 points)

A.
```
SELECT DISTINCT MovieID FROM Ratings
WHERE Rating > (SELECT AVG(Rating) FROM Ratings);
```

```
MovieID
----------
   7
   3
   10
```

B.
```
SELECT Age, Count(*)
FROM Movies, Users, Ratings
WHERE Movies.MovieID = Ratings.MovieID
      AND Ratings.UserID = Users.UserID
      AND Year=1998
GROUP BY Age;
```

```
Age   Count(*)
--------------------
25       3
35       1
50       1
```

C. $\Pi_{MovieTitle}(Movies) - \Pi_{MovieTitle}(Movies \bowtie Ratings)$

```
MovieTitle
----------
Toy Story
Jumanji
Doctor Zhivago
The Jungle Book
It Takes Two
A Streetcar Named Desire
```

D. $\Pi_{UserID,Age}(\sigma_{Rating>3}(Users \bowtie Ratings))$

```
UserID   Age
--------------------
1        50
2        25
3        35
4        45
6        25
```

Scoring: 1 for getting correct result schema, 1 for getting right tuples.

2 Query Equivalence (10 minutes, 8 points)

For each pair of queries below, are the queries equivalent, i.e., is **1** $\equiv$ **2** ? Answer **True** or **False**. In other words, do the queries *always* return the same result (regardless of data), or might there be some data on which they return different results?

You get one point (1 pt) if your answer is correct for the data given in the example database, but not correct for ALL data following that schema. Two points (2 pts) if your answer hold for any data (e.g., if the queries give the same result on the example database, but there could be a different database where they wouldn't give the same result, you get 1 pt for answering true (holds on this data), but 2 pt for answering false (queries aren't really equivalent.)

1	**2**	**1 ≡ 2 ?**
$\sigma_{Year=1995}(\ Movies \bowtie Ratings)$	$\sigma_{Year=1995}(Movies) \bowtie Ratings$	**T**
$\gamma_{Year,count(Genre)}(Movies)$	$\gamma_{Year,count(Genre)}(\Pi_{Year,Genre}(Movies))$	**F**

`SELECT MovieTitle, AVG(rating)`
`FROM Movies, Ratings, Users`
`WHERE Ratings.UserID = Users.UserID`
`AND Movies.MovieID = Ratings.MovieID`
`AND Age < 30`
`GROUP BY MovieTitle`
	`SELECT MovieTitle, AVG(rating)`
	`FROM Movies, Ratings`
	`WHERE Ratings.UserID in`
	`(SELECT UserID FROM Users`
	`WHERE Age < 30)`
		F

`SELECT DISTINCT MovieTitle, Genre`
`FROM Movies`
`WHERE Movies.MovieID in`
`(SELECT MovieID`
`FROM Ratings,Users`
`WHERE Ratings.UserID = Users.UserID`
`AND Gender="F")`
	$\Pi_{MovieTitle,Genre}(\sigma_{Gender='F'}($
	$\Pi_{MovieTitle,Genre,Gender}($
	$(Movies \bowtie Ratings) \bowtie Users)))$
		T

3 Query story problem (10 minutes, 6 points)

We have been asked to do a gender-based comparison of different movie genres. Give a query that allows us to easily compare the average rating and number of people who've rated movies, broken down by genre, with separate averages and numbers for each gender.

A. **In SQL:**

```
SELECT Genre, Gender, AVG(Rating), Count(*)
FROM Movies, Users, Ratings
WHERE Movies.MovieID = Ratings.MovieID AND Users.UserID = Ratings.UserID
GROUP BY Gender, Genre;
```

B. **In Relational Algebra:**

$$\gamma_{Genre,Gender,Avg(Rating),Count(*)}(Movies \bowtie Ratings \bowtie Users)$$

Scoring: 1 point for join, 1 for group by, 1 for count, 1 for reasonable SQL attempt, 1 for reasonable Relational algebra attempt, 1 for getting one of them completely correct. Note: Another solution computed each gender and (outer)joined the results, giving a single row for each genre with different columns for count/average of each gender. Which only works if you assume you know in advance all possible values for gender.

4 Query Execution (5 minutes, 4 points)

Given two tables **T1** (columns: a, c) and **T2** (columns: b, c), briefly explain the result of the following query execution:

```
SELECT T1.a, AVG(T2.b)
FROM T1 LEFT OUTER JOIN T2 on T1.c = T2.c
```

The query won't work since the GROUP BY clause is missing for an attribute T1.a which is selected. To make this query work you would need to add GROUP BY T1.a at the end.

Scoring: 2 for recognizing it won't work, 2 for explanation why it won't work, 1 for showing understanding of join, 1 for understanding of aggregation (average of b*), 1 for solid description of left outerjoin (keeps all values of T1, inserts nulls for T2.b where necessary.)*

5 Levels of Data Abstraction (5 minutes, 4 points)

Briefly describe the three levels of data abstraction. At what level would you consider the **Movies** table given on page 4? Explain why.

View: Access to the data in a manner appropriate for a particular user/application.

Logical: Captures the entire dataset, relationships, and constraints in a manner independent of the physical representation of the data.

Physical: Reflects actual storage of the data.

Movies could be considered View level (reflects how someone would access the data) or logical. However, a view level for just "movies" would likely omit MovieID (it really isn't important to the movie, only relating it to ratings), so I would consider it logical level.

Scoring: 1 for naming the three levels, 1 for description of each, 1 for equating movie to a level, 1 for explanation that matches the answer.

6 Data Independence (5 minutes, 3 points)

We defined *Physical Data Independence* as the ability to modify the physical schema without changing the logical schema. Assume I change the data type of an attribute. Would this change satisfy physical data independence? Explain your answer.

Achieving physical data independence should mean that use of the data is not affected by changes to the physical schema. Changing datatypes is likely to break applications. If the change were compatible (e.g., extending the length of a varchar field), and applications were written independent of the length of the field, then this would satisfy physical data independence, but cases where the datatypes are incompatible (e.g., date vs. string) would not.

Scoring: 1 for showing understanding of distinction between physical and logical schema, 1 for good explanation, 1 for explanation matching answer. For example, if you got it backwards (changing datatype doesn't change physical storage, which generally isn't true, as different datatypes are represented differently), but your answer and explanation matched well, this would be 2 points.

Example tables for the exam

Consider a database with tables **Movies**, **Users** and **Ratings**. **Movies** contains the id and name for a movie along with its year of release and genre. **Users** has the id for each user and information on their demographic e.g., age and gender. **Ratings** contains information on user ratings (on a scale of 1 to 5) on particular movies.

Movies

MovieID	*MovieTitle*	*Year*	*Genre*
1	Toy Story	1995	Animation
2	Jumanji	1995	Fantasy
3	Everest	1998	Documentary
4	Hush	1998	Thriller
5	Doctor Zhivago	1965	Drama
6	The Jungle Book	1967	Animation
7	Sabrina	1954	Romance
8	It Takes Two	1995	Comedy
9	A Streetcar Named Desire	1951	Drama
10	The Bicycle Thief	1948	Drama

Users

UserID	*Gender*	*Age*
1	F	50
2	F	25
3	F	35
4	F	45
5	M	18
6	M	25
7	M	35

Ratings

UserID	*MovieID*	*Rating*
1	10	4
5	10	3
2	10	4
3	10	5
6	10	4
4	7	5
7	7	3
1	3	4
2	3	5
6	3	4
7	4	1
6	4	2