CMU - SCS
15-415/15-615 Database Applications
Spring 2013, C. Faloutsos
Homework 1: E.R. + Formal Q.L.
Deadline: 1:30pm on Tuesday, 2/5/2013

Reminders - IMPORTANT:

- Like all homeworks, it has to be done **individually**.
- Please **typeset** your answers. Illegible handwriting may get no points, at the discretion of the graders. Only drawings may be hand-drawn, as long as they are neat and legible.
- Please submit your answers in **hard copy, in class**, 1:30pm, on Tuesday, 02/05/2013 .
- For ease of grading, please solve each of the five questions on a **separate** page, i.e., five pages in total for this homework. If you need more pages for one problem, please staple them together. Type your **name and andrew ID on each** of the five pages. FYI, we will have five labeled piles at the front of the classroom, one for each problem.

Reminders - FYI:

- Weight: 5% of homework grade.
- The points of this homework add up to 100.
- Rough time estimates: 3-6 hours (1-2 hours for each of the five questions).

Q1. Gym Database, 30 pts [Danai] - SUBMIT ON SEPARATE PAGE

Suppose that Allegheny County wants to centrally control the gyms of the area, and towards this goal you are hired to design a database to store information about the gyms, and their employees. Below we describe the entities and the relations that should be stored in the database.

- Each `gym` has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An `employee` is uniquely defined by her SSN. Moreover, we store her name.
- An `employee` may `work at` several gyms of the Allegheny County. For instance, Smith is working in the mornings at "X Shadyside" and in the evenings at "Fitness Factory".
- For every `employee` we record the percentage of time he or she works at each `gym`. Thus, employee 'Smith' above, would be recorded as working at 50% at "X Shadyside" and 50% at "Fitness Factory".
- Some employees may specialize in one of the following specialties: manager, receptionist, or personal trainer. Each `employee` has zero or one specialization.
- Every manager `manages` one or more gyms.
- Each `gym` has exactly one manager, that is, it can not be without a manager.
- For a personal trainer we also store the type(s) of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition. A trainer may have zero or more certifications.

Q1.1 Ignoring for now the entities of the database and focusing on the relationship `works at`, correct the errors (if any) in the following E.R. diagram. You may use the thin/thick 'arrow' notation of the book, or the M:N notation from the foils.[2 pts]

★ **SOLUTION:** "works-at" is a many-to-many relationship (M-to-N instead of 1-to-N).

Q1.2 Give the whole E.R. diagram for the small database described above. Make sure to indicate the primary and partial keys, cardinality constraints, weak entities (if applicable), and participation constraints. If you make any assumptions, state them clearly. [8 pts]

★ **SOLUTION:** The entities `gym`, `employee` (and its types), as well as the relationships `works at` and `manages` are required in this question. All the attributes should be included too.

Q1.3 Say whether the `employee` IS-A hierarchy has (a) covering and (b) overlapping constraints. Give a short explanation for your answer. [5 pts]

★ **SOLUTION:** (a) Not overlapping constraints: each employee has only one specialization.
(b) Covering constraints: there are no more specializations than the ones mentioned (manager, receptionist, personal trainer).

Q1.4 Turn the whole E.R. diagram into tables. Give SQL statements to create tables. Make sure to indicate primary keys, and foreign keys (if any). No need to specify `on delete` clauses, nor `check` constraints. [15 pts]

★ **SOLUTION:** The SQL statements are given below.

```
-- [4 pts]
create table gym (
    name char(30),
    strNo integer,
    strName char(30),
    ZIP integer,
    employee_SSN char(11) not null,
    primary key (name),
    foreign key (employee_SSN) references employee,
        on delete no action
);

-- [3 pts]
create table phoneNo (
```

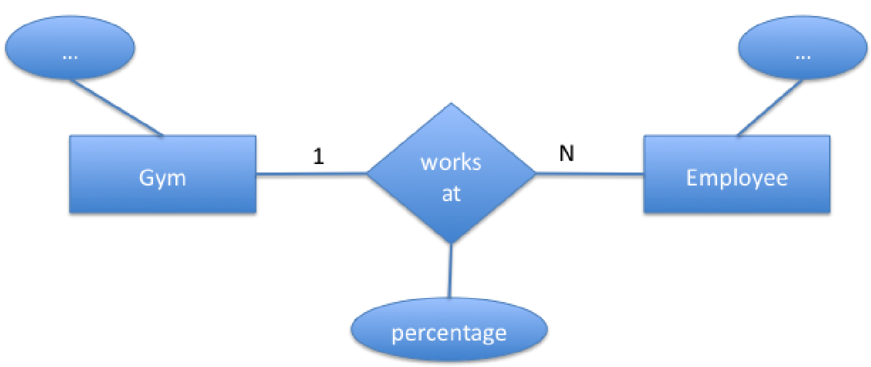

Figure 1: Finding the errors in the “works at” relationship.

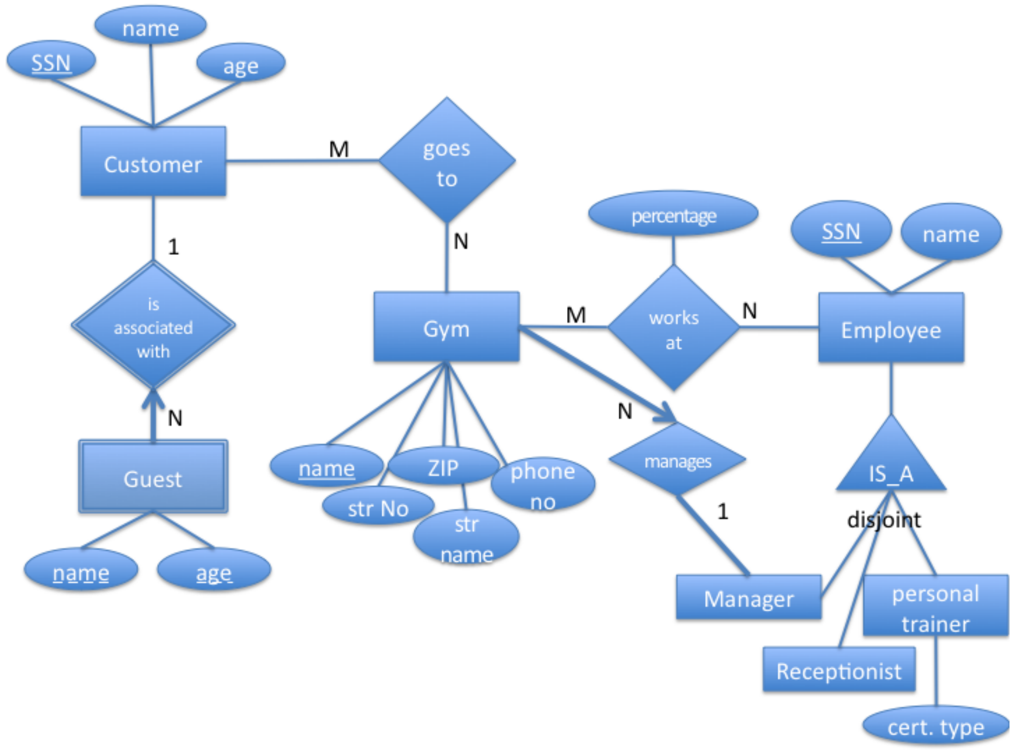

Figure 2: E.R. diagram of database described in exercises 1 and 2.

```
    gym_name char(30),
    phoneNo long integer;
    primary key (phoneNo);
    foreign key (gym_name) references gym
        on delete cascade
);

-- [3 pts]
create table works_at (
    gym_name char(30),
    employee_SSN char(11),
    percentage integer, not null,
    primary key (gym_name, employee_SSN, percentage),
    foreign key (gym_name) references gym,
    foreign key (employee_SSN) references employee
);
```

```sql
-- [2 pts]
create table employee (
    SSN char(11),
    name char(30),
    specialization char(30),
    primary key (SSN)
);

-- [3 pts]
create table certification_personal_trainer (
    employee_SSN char(11),
    certification_type char(30),
    primary key (employee_SSN, certification_type),
    foreign key (employee_SSN) references employee,
        on delete cascade
);
```

Q2. Extending the Gym Database, 15pts [Danai] - SUBMIT ON SEPARATE PAGE

Assume that we want to extend the gym database of exercise 1 to include information about the customers.

- The information stored for a `customer` is: SSN (unique), name, age.
- Each `customer` may be going to more than one `gyms`. For example, Alice attends group exercise classes both in "Fitness Planet" and "X Shadyside", while Bob has always been going to "Club One Fitness".

Q2.1 Complete the E.R. diagram below. Again, indicate the primary and partial keys, cardinality constraints, weak entities (if applicable), and participation constraints. If you make any assumptions, state them clearly. [4 pts]

★ **SOLUTION:** The entity `customer` and its attributes should be added to the E.R. diagram of the previous exercise. Also the cardinality of the relationship `goes to` should be annotated.

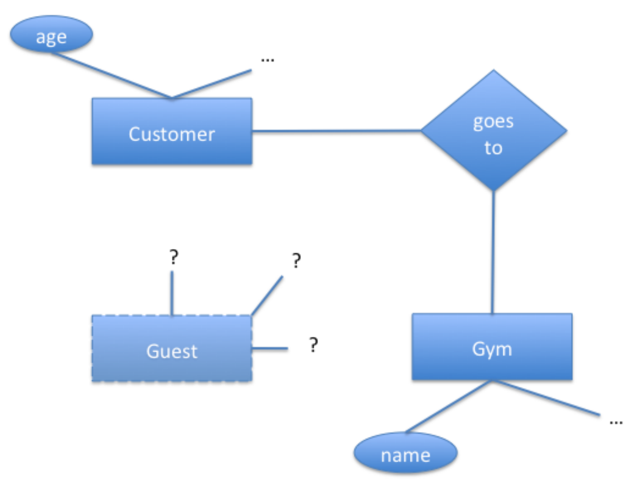

Figure 3: Complete the E.R. diagram of the extended database.

Q2.2 Allegheny County `gyms` also allow each `customer` to have `guests`/friends `associated with` him. These `guests` can use the facilities of the `gyms` their host goes. The `guests` are *not* considered customers of the gym. Only the name and age of the `guests` are stored in the database, and we assume that for each `customer` the pair `(guest-name,guest-age)` is unique. Extend the E.R. diagram further to also include the entity `Guest`. [5 points]

★ **SOLUTION:** The weak entity `guest`, its attributes, and the relationship `is associated with` should be added to the E.R. diagram of the previous exercise. Also the cardinality of the relationship `goes to` should be annotated.

Q2.3 Turn the new entities of the E.R. diagram into tables. Give SQL statements to create tables. Make sure to indicate primary keys, and foreign keys (if any). No need to specify `on delete` nor `check` constraints [6 pts].

★ **SOLUTION:** The SQL statements are given below.

```
-- [2 pts]
create table goes_to (
    customer_SSN char(11),
    gym_name char(30),
    primary key (customer_SSN, gym_name),
    foreign key (gym_name) references gym,
    foreign key (customer_SSN) references customer
);

-- [2 pts]
create table customer (
    SSN char(11),
    name char(30),
    age integer,
    goal char(50),
    primary key (SSN);
);

-- [2 pts]
create table guest (
    name char(30),
    age integer,
    customer_SSN char(11),
    primary key (name, age, customer_SSN),
```

```
    foreign key (customer_SSN) references customer,
        on delete cascade
);
```

Q3. Relational Algebra, 15 pts [Kate] - SUBMIT ON SEPARATE PAGE

We are creating a "CMU recommendation system" which will contain the preferences of current students for various recreational places on campus. The ultimate goal is to recommend venues to newer students based on the similarity of their preferences to more senior students. We start with the following schema:

Table STUDENT_LIKES (e.g. see Table 2):

- `sname` - Student's first and last name, e.g. Bilbo Baggins. Assume that every name in the database is unique.
- `category` - Name of category of interest, e.g. "Cafe," "Library."

It is not needed for this question, but, for your information, the primary key of the STUDENT_LIKES table is (`sname` , `category`).

Table PLACES - a place is determined by its category and building name and has a popularity score, e.g. see Table 1:

- `category` - Category name, e.g. "Cafe," "Library."
- `bname` - Building name, e.g. "Skibo," "UC."
- `popularity` - The popularity of a place, as determined by the building name (`bname`) and the category. An integer score between 1 and 10.

Again, for your information, the primary key of the PLACES table is (`category` , `bname`).

Q3..1 Select which option describes what the following relational algebra query outputs [2 points]:

$$\pi_{bname}(\sigma_{popularity=10}(PLACES))$$

A. Names of buildings in which all places have a popularity of 10.

B. Names of buildings which have at least one place with a popularity of 10.

C. Names of buildings that have only one place/venue/category, but it has popularity = 10.

D. None of the above - *write-in your answer*.

★ **SOLUTION:** B.

Q3..2 Write the relational algebra expression which gives all the tuples from STUDENT_LIKES that have "Cafe" as the `category`. See Table 2 later on. [3 points]

Table 1: PLACES table instance. A place is determined by its category and building name and has a popularity score.

PLACES	category	bname	popularity
	Cafe	Skibo	10
	Gym	Skibo	10
	Cafe	UC	7
	Gym	UC	9
	Movies	UC	7
	Cafe	Resnik	6
	Library	Hunt	9
	Cafe	Hunt	10
	Gym	Gesling	6

★ **SOLUTION:** $\sigma_{category='Cafe'}(STUDENT\_LIKES)$

Q3..3 Consider the PLACES table instance shown in Table 1:

Consider the following expression:

$$\pi_{category,bname}(PLACES) \div \pi_{category}[\sigma_{bname='Skibo'}(PLACES)]$$

Q3..3.1 Describe in English what the query tries to do. A (possibly correct) answer is: "Selects all (building, category) combinations, that are not located in Skibo." [2 points]

★ **SOLUTION:** Returns the building names that superseed Skibo, that is, they contain at least the facilities/categories of Skibo, and possibly, more. That is, at least a cafe and a gym. This includes Skibo! The right-hand side of the division uses 'Skibo' only to select 'Gym' and 'Cafe,' but the left-hand side never removes 'Skibo' from the results.

Q3..3.2 How many columns are returned in the output? [2 points]

★ **SOLUTION:** One.

Q3..3.3 What are the column headers of the output? [2 points]

★ **SOLUTION:** *bname*

Q3..3.4 How many rows are returned in the output? [2 points]

★ **SOLUTION:** Two.

Q3..3.5 Give the contents of the returned rows. [2 points]

★ **SOLUTION:** The rows are:

bname
UC
Skibo

Q4. Relational Tuple Calculus, 20 pts [Kate] - SUBMIT ON SEPARATE PAGE

We use the same schemas STUDENT_LIKES and PLACES as in Question Q3..

Q4..1 For the task of Q3..2, write the relational tuple calculus expression. [5 points]

★ **SOLUTION:** $\{S | S \in \text{STUDENT\_LIKES} \wedge S._{\text{category}} =' Cafe'\}$

Q4..2 Consider the following STUDENT_LIKES table instance[1]:

Table 2: STUDENT_LIKES table instance.

STUDENT_LIKES	sname	category
	Anna Abbott	Cafe
	Anna Abbott	Gym
	Anna Abbott	Movies
	Colin Creevey	Cafe
	Colin Creevey	Library
	Colin Creevey	Movies
	Gregory Goyle	Movies
	Luna Lovegood	Library

Consider the following relational tuple calculus expression:

$$\begin{aligned}\{S | S \in \ & \text{STUDENT\_LIKES} \ \wedge \ \exists S' \in \text{STUDENT\_LIKES}(\\ & S'.\text{category} = S.\text{category} \ \wedge \\ & S'.\text{sname} = '\text{Colin Creevey}' \ \wedge \\ & S.\text{sname} \neq '\text{Colin Creevey}')\}\end{aligned}$$

Q4..2.1 Describe in English what tuples this query selects. A (possibly correct) answer is: "The query selects all tuples from STUDENT_LIKES in which the student has no common preferences with Colin Creevey." [3 points]

[1] Irrelevant side note: Anna's real name is Hannah, but we chose "Anna" for the alliteration effect.

★ **SOLUTION:** The query selects tuples from STUDENT_LIKES in which the students have at least one category of interest in common with Colin Creevey, excluding Colin Creevy himself.

Q4..2.2 How many columns are returned in the output? [3 points]

★ **SOLUTION:** Two.

Q4..2.3 What are the column headers of the output? [3 points]

★ **SOLUTION:** *sname* and *category*.

Q4..2.4 How many rows are returned in the output? [3 points]

★ **SOLUTION:** Four.

Q4..2.5 Give the contents of the returned rows. [3 points]

★ **SOLUTION:** The rows are:

sname	category
Anna Abbott	Cafe
Anna Abbott	Movies
Gregory Goyle	Movies
Luna Lovegood	Library

Q5. Relational Domain Calculus, 20 pts [Kate] - SUBMIT ON SEPARATE PAGE

We use the same schemas STUDENT_LIKES and PLACES as in Question Q3.. Consider the same PLACES table instance as in Table 1 and the same STUDENT_LIKES table instance as in Table 2

Q5..1 Write the relational domain calculus expression equivalent to the relational algebra expression from Question Q3..1. Fill in the blanks: [5 points]

$\{< B > | \exists C, P(\_\_\_\_\_\_ \in \_\_\_\_\_\_ \wedge \_\_\_\_\_\_)\}$

★ **SOLUTION:** $\{< B > | \exists C, P(< C, B, P > \in \text{PLACES} \wedge P = 10)\}$

Q5..2 For the task of Question Q3..2, write the relational domain calculus expression. [5 points]

★ **SOLUTION:** $\{< S, C > | < S, C > \in \text{STUDENT\_LIKES} \wedge C =' Cafe'\}$

Q5..3 We want to find the students who will be super-happy, in the sense that they will find a super-popular venue (popularity=10), for at least one of their hobbies (= interests = categories). For example, since Anna Abbott likes Cafes, and since Skibo-Cafe is super popular (=10), Anna will be one of the students that your query should report.

Consider the following relational domain calculus expression:

$$\begin{aligned}\{< S > | \quad \exists I \quad & (< S, I > \in \text{STUDENT\_LIKES} \wedge \\ & \exists C, V, P(< C, V, P > \in \text{PLACES} \wedge \\ & P = 10)))\}\end{aligned}$$

Q5..3.1 Although syntactically correct, this query has a (very subtle) logical error and does not return what we intend it to. Give the corrected expression. *Hint:* it takes less than six keystrokes to correct it. [6 points]

★ **SOLUTION:** The problem is that the expression does not request a join of the two tables - we need to specify that the interests in both tables should match. The correct expression is:
$\{< S > | \exists \underline{C}(\underline{< S, C >} \in \text{STUDENT\_LIKES} \wedge \exists \underline{V, P}(\underline{< C, V, P >} \in \text{PLACES} \wedge P = 10)))\}$
(or equivalently, replace C with I).

Q5..3.2 How many rows are returned in the **original, incorrect** query? How many rows are returned by **your** query? [2 points]

★ **SOLUTION:** Four and two, respectively.

Q5..3.3 Give the contents of the returned rows for the **original, incorrect** query. Give the contents of the returned rows for **your** query. [2 points]

★ **SOLUTION:** The rows from the **incorrect** expression are:

Anna Abbott
Colin Creevey
Gregory Goyle
Luna Lovegood

The rows from the **correct** expression are:

Anna Abbott
Colin Creevey