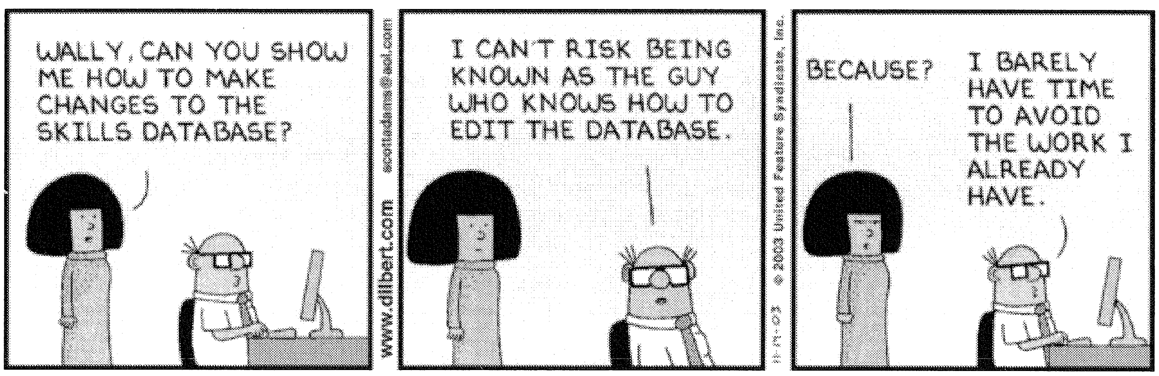
Midterm Exam
CS 122A
Spring 2014

Max. Points: 100

(Please read the instructions carefully!)

Instructions:
- The total time for the exam is 75 minutes; be sure to budget your time accordingly.
- The exam is closed book and closed notes (but "open cheat sheet").
- Be sure to answer each part of each question after reading them carefully.
- If you don't understand something, ask the instructor for clarification.
- If you still find ambiguities in a question, note the interpretation you are taking.
- The last page of this exam is blank; you can use it as scratch paper.



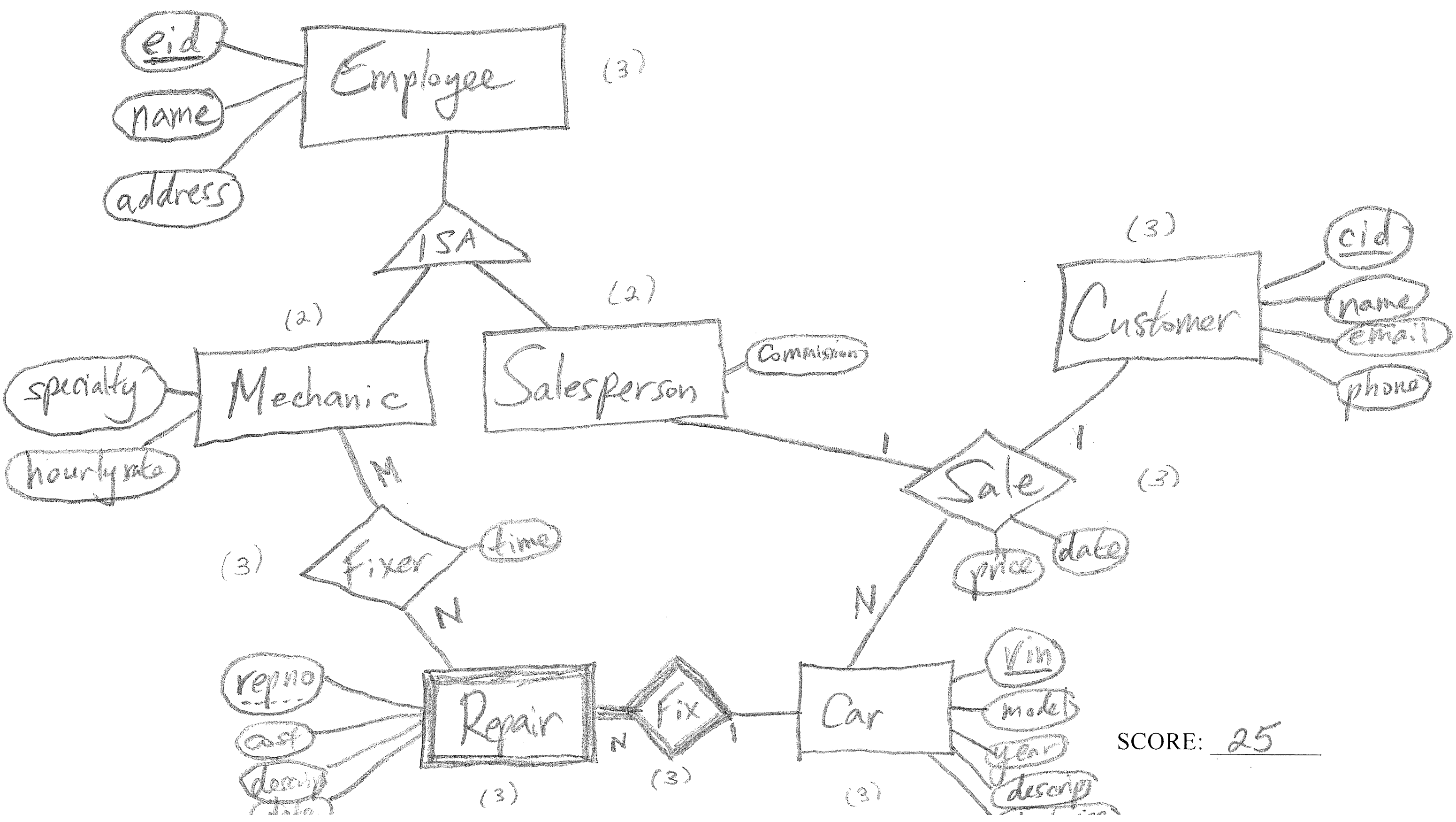
NAME: P. R. Fecto **STUDENT ID:** /

QUESTION	***POINTS***	***SCORE***
1	**25**	25
2	**25**	25
3	**50**	50
TOTAL	**100**	100

Question 1: E-R Modeling (25 points)

(a) (25 pts) You've been hired as a consultant by a friend who's about to open Orange County's greenest car dealership, WaterWorld! WaterWorld will be selling and maintaining the new Aqua brand of cars based on water-powered engine technology. Available Aqua models will range from the compact entry-level Squirt all the way up to the luxurious 8-passenger Cruise Liner. Your job is to help WaterWorld set up a database for keeping track of their employees, cars, customers, etc. You are to design and draw an E-R schema for their collection of data. Be sure to capture all of the relevant entities, relationships, and attributes. Label each relationship to indicate what kind it is (e.g., 1:1, 1:N, etc.), what its participation constraints are, and so on. Identify the primary key for each entity. Note your additional assumptions (if any) next to your E-R diagram. Here's what you're told about WaterWorld's data by your friend:

- Each WaterWorld employee has a unique employee number, a name, and a mailing address.
- Some more specialized employees will be either a mechanic or a salesperson (but not both). In addition to the basic employee information, mechanics will have a specialty and an hourly billing rate. Similarly, a salesperson will have a commission rate that sets the percentage of their car sales' profits that they get to keep.
- Cars have a vehicle identification number, a model, a year, a detailed description, and a list price.
- Customers have a customer id, a name, an e-mail address, and a phone number.
- When a sale takes place, a given sale is of one particular car to one particular customer by one particular salesperson on a given date and at an agreed-upon selling price.
- Customers aren't limited to just buying one car. WaterWorld hopes their customers will love this new technology and that many will buy multiple cars – leading, of course, to a flood of sales. ☺
- When an Aqua car has a problem, a repair job will be scheduled to fix it. A repair job is associated with the car that it is a repair for, and the job has a cost, a description, a date, and a sequence number (e.g., this might be the 3rd repair job for the associated car).
- A given repair job will be performed by one or more mechanics, each of whom spends some number of hours on their part of the repair job.



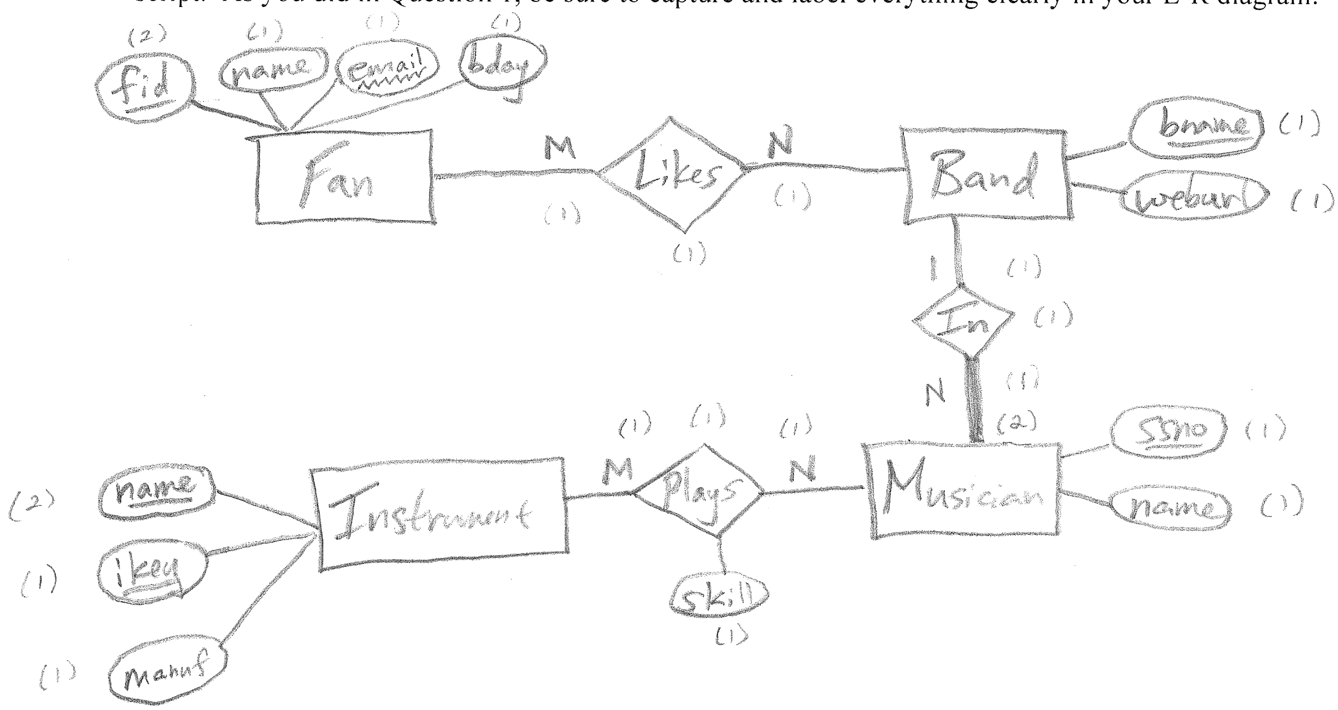
SCORE: 25

Question 2: E-R Translation (25 points)

(a) (25 pts) Another friend of yours – what a great alumni network Anteaters have! – is a huge music fan. She has just purchased a failing music-related Web startup after inheriting $250,000 from a long-lost relative that she hadn't known about. Unfortunately, the company had already laid-off most of its IT staff, including all of its application software developers and its database administrator. Knowing that you're now a CS122a-trained database expert, your friend has hired you as a consultant to help her restart their database-related efforts. The good news is that the laid-off staff didn't have sufficient time to shred all their documents, so there is some record of what they did. The bad news is that the all-important E-R diagram is nowhere to be found – in fact, all that your friend found was a cryptic SQL script that says:

```
CREATE TABLE Fan (fid INT, name VARCHAR(60), email VARCHAR(40), birthday DATE,
    PRIMARY KEY (fid),
    UNIQUE (email));
CREATE TABLE Band (bname VARCHAR(40), weburl VARCHAR(120),
    PRIMARY KEY (bname));
CREATE TABLE Musician (ssno CHAR(9), name VARCHAR(60), band VARCHAR(40) NOT NULL,
    PRIMARY KEY (ssno),
    FOREIGN KEY (band) REFERENCES Band (bname) ON DELETE CASCADE);
CREATE TABLE Instrument (name VARCHAR(20), ikey CHAR(2), manufacturer VARCHAR(30),
     PRIMARY KEY (name, ikey));
CREATE TABLE Likes (fan INT NOT NULL, band VARCHAR(40) NOT NULL,
    PRIMARY KEY (fan, band),
    FOREIGN KEY (fan) REFERENCES Fan (fid),
    FOREIGN KEY (band) REFERENCES Band (bname));
CREATE TABLE Plays (player CHAR(9), axe VARCHAR(20), ikey CHAR(2), skill FLOAT,
     PRIMARY KEY (player, axe, ikey),
     FOREIGN KEY (player) REFERENCES Musician (ssno),
     FOREIGN KEY (axe, ikey) REFERENCES Instrument (name, ikey));
```

Help your friend by reverse-engineering the E-R diagram that most likely led to the creation of this SQL script. As you did in Question 1, be sure to capture and label everything clearly in your E-R diagram.



SCORE: 25

Question 3: Relational Languages and Queries (50 points)

Consider the following relational schema for storing information about UCI students and their major departments. The primary key for Student is *sid*, the primary key for Dept is *dno*, and there are two foreign keys in Student: *major*, which references Dept.dno, and *mentor*, which references Student.sid. (A student can have another student assigned to help mentor them during their studies.)

Relations: Student(*sid*, sname, age, year, major, mentor) Dept(*dno*, dname, school, chair)

Given this schema, write each of the following queries in the indicated relational language. Pay careful attention to the language being asked for, as ***no*** partial credit will be given if you answer a question in the wrong language (!). As a quick syntax reminder, to print out the names and chairs of the departments in ICS using the relational algebra or the relational calculus, you might write the following

$$\pi_{\text{dname}\rightarrow\text{deptname, chair}\rightarrow\text{chairname}} (\sigma_{\text{school= 'ICS'}} (\text{Dept}))$$

$$\{ t\ (\text{deptname, chairname}) \mid \exists\, d \in \text{Dept}\ (t.\text{deptname} = d.\text{dname} \wedge t.\text{chairname} = d.\text{chair} \wedge d.\text{school} = \text{'ICS'}) $$

Student

sid	sname	age	year	major	mentor
1	Sally	22	Junior	10	3
2	John	18	Freshman	*null*	3
3	Rahul	22	Grad	30	*null*
4	Jake	17	Senior	20	1
5	Leo	13	Grad	30	3
6	Emily	69	Freshman	10	*null*
…	…	…	…	…	…

Dept

dno	dname	school	chair
10	CS	ICS	Turing
20	EE	Eng	Ohm
30	Physics	PhySci	Einstein
40	Stats	ICS	Poisson
…	…	…	…

(a) (10 pts) Print the department names and school names of departments that have at least one retired student (assuming that retirement occurs at/after age 65) as a major. Write this query in the *relational calculus.*

(1) (0) (1) (1)

$\{ t(\text{dname}, \text{school}) \mid \exists\, d \in \text{Dept} (d.\text{dname} = t.\text{dname}$

(1) (1) (2)

$\wedge\ d.\text{school} = t.\text{school} \wedge \exists\, s \in \text{Student} (s.\text{age} \geq 65$

(3)

$\wedge\ s.\text{major} = d.\text{dno})) \}$

SCORE: 10

Relations: Student(*sid*, sname, age, year, major, mentor) Dept(*dno*, dname, school, chair)

(b) (10 pts) Print the department number and chair name of the department (if there is one) that all of the 30-year-old graduate students at UCI are majoring in. Write this query in the *relational algebra.*

$$\underset{(1)}{\pi_{dno,chair}}\left(\left(\pi_{\underset{(1)}{major},\underset{(1)}{sid}}(Student) \overset{(2)}{\div} \underset{(1)}{\pi_{sid}}\left(\sigma_{\underset{(1)}{age=30} \wedge \underset{(1)}{year=\text{"Grad"}}}(Student)\right)\right) \underset{\underset{(2)}{major=dno}}{\bowtie} Dept\right)$$

(c) (10 pts) For each department in the Engineering (Eng) school that has at least 10 majors, print the department's total number of majors as well as their majors' average age. The result should have four columns containing the department's department number, department name, major count, and average major age, and it should have a result row for each qualifying department. Write this query in *SQL.*

```
        (1)       (1)         (1)          (1)
SELECT d.dno, d.dname, count(*), avg(s.age)
FROM Dept d, Student s                   (1)
WHERE d.school = 'Eng'                   (1)
   AND d.dno = s.major                   (1)
GROUP BY d.dno, d.dname                  (2)
HAVING count(*) >= 10                    (1)
```

SCORE: 20

Relations: Student(*sid*, sname, age, year, major, mentor) Dept(*dno*, dname, school, chair)

(d) (10 pts) What does the following SQL query do? Answer in three parts, as requested below.

```
SELECT d1.dname, d2.name AS otherdname
FROM Dept d1, Dept d2
WHERE (SELECT COUNT(*) FROM Student WHERE major = d1.dno)
      = (SELECT COUNT(*) FROM Student WHERE major = d2.dno)
AND d1.dno != d2.dno
```

(i) (4 pts) Show the query's output for the sample data given at the start of the problem (on p. 3).

dname	otherdname
CS	Physics
Physics	CS
⋮	⋮

(1)
(2)
(1)

(ii) (4 pts) State what this query does in clear English terms.

It prints department name (2) pairs for departments with the same number (2) of majors.

(iii) (2 pts) A clever colleague suggested changing the last line of the query to "... AND d1.dno < d2.dno". Explain why changing the comparison in this way would be a good idea.

It will eliminate (1) redundant pair printing by printing each pair once (in dno order). (1)

(h) (4 pts) Which of the following statements are actual advantages of using a declarative query language? Answer by putting an **X** in the square next to each such advantage.

- ☒ It leaves the selection of an appropriate query evaluation strategy up to the system. (1)
- ☐ It allows developers to avoid dealing with SQL by using the relational algebra instead. (1)
- ☒ It shortens the time that would otherwise be required to develop a new database application. (1)
- ☐ It avoids possible race conditions related to concurrent database access by multiple users. (1)

(i) (6 pts) Indicate whether each of the following statements is a TRUE (**T**) or FALSE (**F**) statement.

T **F** The relational algebra is more expressive than the safe subset of the relational calculus. (2)

T **F** SQL is more expressive than the relational algebra. (2)

T **F** The union operator (**U**) in relational algebra is unnecessary since a join can do the same thing. (2)

SCORE: 20