EECS-3421A Test #1

Design

Sur / Last Name:
Given / First Name:
Student ID:

- **Instructor:** Parke Godfrey
- **Exam Duration:** 75 minutes
- **Term:** Fall 2015

Answer the following questions to the best of your knowledge. Your answers may be brief, but be precise and be careful. The exam is closed-book and closed-notes. Calculators, etc., are fine to use. Write any assumptions you need to make along with your answers, whenever necessary.

There are five major questions, each with parts. Points for each question and sub-question are as indicated. In total, the test is out of 50 points.

If you need additional space for an answer, just indicate clearly where you are continuing.

Marking Box	
1.	/10
2.	/10
3.	/10
4.	/10
5.	/10
Total	/50

1. **Entity-Relationship Modelling.** *Draw on the count of three!* (10 points) [Exercise]

Nova Scotia Board of Fisheries: Licencing & Hauls.

The Nova Scotia Board of Fisheries (NSBF) oversees fishing stocks and *licences* fishing *companies* for how much they can *haul*; that is, catch of a given type of *fish* in a given *region* (a given area of the sea). They also keep track of companies's hauls, so that they can ensure companies are in compliance with their licences.

A *company* has a number of *boats*. A boat is owned by one company. We know a company by its chartered *company* name. For a company, we also keep the *harbour*, the town in which the company is chartered. A boat has a unique registration number (b#); and we need to know its *capacity* (how many tonnes of fish it can haul) and its usual number of crew (#crew).

NSBF licences a company for a certain type of fish to be hauled from a given region. A licence records *when* the license was issued, the *agent* who gave the licence, and a *limit* (the number of tonnes in total the licence allows the company to haul of that type of fish from that region). A company can have more than one licence for a given region, allowing them to fish for more than one type of fish there. They can have licences to fish for a given type of fish in more than one region. And, of course, more than one company may have licences for a given type of fish in a given region. NSBF only keeps track of current licences issued, not old ones that have expired.

NSBF tracks whenever a boat makes a haul of a type of fish in a region. For a haul, we record the *date* the *weight* (in tonnes), and, of course, the type of fish. (A given haul only involves one type of fish.)

NSBF also keeps track of how much fish of a given type that they estimate to be in each region. This is to be recorded in a field called *stock* (the value of which measures the estimated tonnage of fish of that type in the region).

In Question 1a on the next page, you are to devise an entity-relationship (E-R) diagram to model NSBF's Licencing & Hauls as described above. Then answer Question 1b below.

In your E-R, do not add any entity unless absolutely needed. You may broaden keys as needed to accommodate the necessary design constraints. Do not forget attributes, and show all keys.

b. (2 points) Does your design make it possible to check for *over-haulage*; that is, if a company takes more fish of a given type from a region than for which they are licenced?

If so, explain briefly how.

If not, explain why it is not possible to capture this in an E-R diagram that follows the requirements.

---

a. (8 points) Present your E-R diagram.

2. **Relational Schema.** *Meet my relations.* (10 points) [Short Answer]

Scientist(s#, name∗, office, phone#, email)
Grant(g#, agency∗, amount∗, started∗)
Funding(s#, g#, amount∗)
    FK (s#) refs **Scientist**
    FK (g#) refs **Grant**
Project(pname, g#, s#∗)
    FK (g#) refs **Grant**
    FK (s#, g#) refs **Funding** — *project leader*
Journal(journal)
Publication(title, year, journal, #pages)
    FK (journal) refs **Journal**
Author(s#, title, year, journal, rank)
    FK (s#) refs **Scientist**
    FK (title, year, journal) refs **Publication**

Figure 1: Grants.

A relational schema for tracking scientists, grants, projects, and publications is shown in Figure 1. The underlined attributes indicate a table's primary key (and are, hence, not nullable). Attributes that are appended by an '∗' are not nullable (for example, name∗ in **Scientist**). Foreign keys are indicated by FK.

For Questions 2a–2c, consider an E-R diagram from which this relational schema might have derived.

a. (3 points) Draw the part of the E-R diagram that shows how *project leader* is modelled (so, involving **Project**, **Grant**, **Funding**, and whatever else is needed for this).

b. (2 points) What are the many-many relationships?

c. (3 points) What would be weak entities?

d. (2 points) We want to know how many scientists are funded by a given grant. Should the *relational* schema be modified to support this?

If so, suggest a change to the relational schema for this.

If not, explain why not.

3. **General.** *Luck of the draw.* (10 points) [Multiple Choice]

Choose one best answer for each. There is no additional penalty for a wrong answer. If you feel clarification for your answer is needed, write a brief clarification next to the question.

---

a. The `SQL` statement "`DELETE from R;`"

A. is guaranteed to remove all the tuples from **R**.
B. may remove just some tuples from **R**.
C. may also remove tuples in tables other than **R**.
D. will drop table **R** from the database.
E. will do nothing because it is missing a `where` clause.

---

b. NULL values can be used

A. to opt a tuple out of enforcement of the primary key.
B. to opt a tuple out of enforcement of a foreign key.
C. to delete a tuple from the table.
D. to make a tuple to be non-updatable.
E. to add extra columns for a tuple.

---

c. A relational database systems does all *except* which of the following?

A. It ensures that no update to the database violates any of its integrity constraints.
B. It provides permanent storage, protecting against loss of data.
C. It handles queries over the data in the database via a query language.
D. It translates E-R models into correct relational schema.
E. It provides concurrency control via transaction management, allowing multiple people and applications to use the database at the same time.

---

d. It is *not* possible to capture which E-R concept in a relational schema?

A. One-to-one relationships.
B. One-many mandatory participation.
C. Many-many mandatory participation.
D. Ternary relationships.
E. Everything in E-R can be captured in a relational schema.

---

e. In an E-R diagram, if one sees a bold line with no arrow between an entity set and relationship set, this means

A. every entity in the entity set must appear exactly once in the relationship set.
B. every entity in the entity set must participate in the relationship set.
C. a relationship in the relationship set need not involve an entity from the entity set.
D. the entity set *is an* instance of the relationship set.
E. the entity set is weak.

---

f. *Ternary relationships*

A. cannot be used in aggregation.
B. can always be equivalently replaced by several binary relationships.
C. have keys like entities.
D. relate more than two entities.
E. are used to relate weak entities.

---

g. A *weak entity*

A. inherits part of its key from the "parent" entities to which it is related.
B. is an entity with no key.
C. is an entity with no attributes besides its key.
D. is never mapped to a table in conversion to a relational schema.
E. is the same thing as ISA in E-R.

---

h. Relational schema differ from E-R models in that

A. the concept of entity cannot be expressed.
B. the concept of relationship cannot be expressed.
C. the concept of ternary relationship cannot be expressed.
D. attributes / columns are somtimes repeated between tables, unlike entities.
E. all tables are inherently equivalent to weak entities.

---

i. When creating a table in a relational database, one must

A. declare a primary key.
B. declare at least one foreign key.
C. declare at least two columns.
D. declare at least one tuple.
E. *None of the above.*

---

j. Which of the following is *not* a relational schema? (The underlined attributes are to indicate the key.)

A. **Procedure**(name: a unique name for this piece of code,
description: a description of what the code does,
language: which computer language it is written in (references **language**),
code: the code itself

B. **Student**(name: name of the individual,
birthdate: when he or she was born,
from: where the individual lives,
clubs: a list of that student's club memberships (references **Club**))

C. **Marriage**(wife: name of the wife,
husband: name of the husband,
when: the date when they were married)

D. **Disease**(name: medical name of the disease,
symptom_1: boolean, whether the disease has symptom #1,
⋮
symptom_1729: boolean, whether the disease has symptom #1729)

E. **Book**(title: the title of the book,
year: the year it was published,
author: who wrote it,
publisher: who published the book,
text: the entire text of the book)

Extra space.

4. **Conceptual to Schema.** *Tables and chairs.* (10 points) [Exercise]

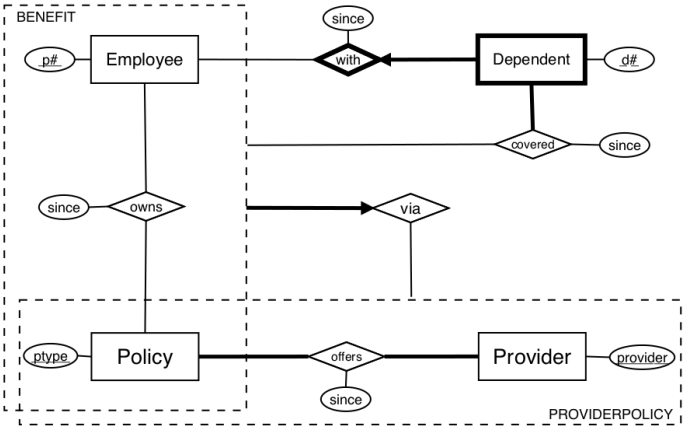

Figure 2: Dependents & Coverage E-R.

For Question 4a on the next page, translate the E-R diagram from Figure 2 into a reasonable relational database schema. Use the notation from Figure 1 in Question 2. Take a *minimalist* interpretation for keys.

Then answer Questions 4b–4d below.

b. (1 point) Can a dependent be associated with more than one employee in this model?

c. (1 point) Can a dependent be covered by more than one policy in this model?

d. (1 point) Can an employee have a given policy (i.e., ptype) via more than one provider in this model?

---

a. (7 points) Your relational schema.

5. **Design & Schema Policies.** *Keep it simple, er, simple.* (10 points) [Short Answer]

a. (3 points) Name three different reasons that an *attribute* in an E-R model might need to be made into an *entity* instead.

b. (2 points) The mechanism of *aggregation* in E-R modelling exists for what purpose?

c. (3 points) Consider the following schema.

```
create table T (
    c integer primary key,
    d integer);
create table S (
    b integer primary key,
    c integer references T(c) on delete cascade);
create table R (
    a integer primary key,
    b integer references S(b) on delete set null);
```

Assume that each of the tables **R**, **S**, and **T** each contain some tuples.

Consider the `SQL` statement "`DELETE FROM T;`".

Explain how many tuples are deleted from eah of the tables **R**, **S**, and **T**—*all*, *some*, or *none*—by this statement.

d. (2 points) Is it possible to capture any E-R model correctly in a relational schema without needing any composite keys?

Why or why not?

Extra space.

Relax. Turn in your test. You reached the end.