CSE-3421M Test #1

“Design”

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

- **Instructor:** Parke Godfrey
- **Exam Duration:** 75 minutes
- **Term:** Winter 2011

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 four 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.	/15
3.	/15
4.	/10
Total	/50

1. (10 points) **The Relational Model.** *Amazing Schemas, Inc.* [Short Answer]

You recently have been hired by Duey, Cheatem, & Howe, Inc.'s (DC&H's) Information Systems Division to do database design work for them. Congratulations!

Customer(cno, *cname*, *address*)
Author(ano, *aname*, birth, country)
Book(isbn, *title*, *year*, *publisher*, *language*)
Wrote(ano, isbn)
    FK (ano) refs **Author**
    FK (isbn) refs **Book**
Stock(isbn, from, *qnty*, *price*)
    FK (isbn) refs **Book**
Purchase(cno, isbn, when, *from*, *qnty*)
    FK (cno) refs **Customer**
    FK (isbn, from) refs **Stock**
Payment(cno, when, *amount*)
    FK (cno) refs **Customer**

Figure 1: **Amazing.com** Schema.

Your boss, Dr. Datta Bas, provides you the *relational schema* in Figure 1 which is the basic schema of the main database for the on-line booksellers **Amazing.com**

The underlined attributes indicate a table's primary key, and are not nullable. Attributes in *italics* are not nullable. Foreign keys are indicated by FK. The attribute price in **Stock** tells how much a customer pays for that book. The attribute from in **Purchase** indicates where the stock was pulled from.

The following questions refer to the schema in Figure 1.

---

a. (4 points) Show the *piece* of an E-R diagram that would result in this schema for **Purchase**, including *any* relationships and entities *directly* related to **Purchase**.

(*Do not* draw the diagram for the *whole* schema.)

b. (4 points) Show the *piece* of an E-R diagram that would result in this schema for **Stock**, including *any* relationships and entities *directly* related to **Stock**.

(Again, *do not* draw the diagram for the *whole* schema.)

c. (2 points) Are **Author** and **Book** related by a one-to-many or a many-to-many relationship? Explain briefly.

2. (15 points) **General.** *Dealer's choice.* [MULTIPLE CHOICE]

Choose *one* best answer for each of the following. Each is worth one point. There is no negative penalty for a wrong answer.

---

a. A relational database management system (RDBMS) does *not* do which of the following?
A. Provide mechanisms to help protect the integrity of the data.
B. Allow for concurrent transactions against the database.
C. Facilitate crash recovery of the database in case of hardware failure.
D. Optimize query evaluation for arbitrary SQL queries.
E. Ensure that relational schemas are in at least 3NF.

---

b. Which of the following is *not* a relational schema? (The underlined attributes indicate the key.)
A. **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)
B. **Personal**(name: name of the individual,
birthdate: when he or she was born,
from: where the individual lives,
hobbies: a list of that person's hobbies (references **hobby**))
C. **Procedure**(name: a unique name for this piece of code,
language: which computer language it is written in (references **language**),
code: the code itself,
description: a description of what the code does)
D. **Marriage**(wife: name of the wife,
husband: name of the husband,
when: the date when they were married)
E. **Disease**(name: medical name of the disease,
symptom_1: boolean, whether the disease has symptom #1,
⋮
symptom_1219: boolean, whether the disease has symptom #1219)

---

c. Codd's rule of *physical data independence* is that
A. all information in the database is to be represented in one and only one way, namely by values in column positions within rows of tables.
B. all views that are theoretically updatable must be updatable by the system.
C. changes that are made to the physical storage representations or access methods must not require changes be made to application programs.
D. changes that are made to tables that do not modify any of the data already stored in the tables must not require changes be made to application programs.
E. data in different tables must not be related.

d. Why are NULL values needed in the relational model? NULL values can be used for all *except* which one of the following?

A. To allow duplicate tuples in the table by filling the primary key column(s) with NULL.
B. To avoid confusion with actual legitimate data values like 0 for integer columns and ‘ ’ (the empty string) for string columns.
C. To leave columns in a tuple marked as “unknown” when the actual value is unknown.
D. To fill a column in tuple when that column does not really “exist” for that particular tuple.
E. To opt a tuple out of enforcement of a foreign key.

---

e. 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);
```

Suppose the current contents of **R**, **S**, and **T** are as follows.

R

A	B
1	1
2	2

S

B	C
1	1
2	1

T

C	D
1	1
2	1

After executing the command:

```
delete from T;
```

what tuples will **R** contain?

A. **R** will not be changed.
B. (1, NULL) and (2, 2).
C. (1, NULL) and (2, NULL).
D. (2,2) only.
E. **R** will contain no tuples.

---

f. Under a relational database system, in the most general case, if table **R** has a foreign key (FK) constraint referencing table **S**, then, via the FK,

A. each tuple in **R** is related to zero or more tuples in **S**.
B. each tuple in **R** is related to zero or one tuple in **S**.
C. each tuple in **R** is related to exactly one tuple in **S**.
D. each tuple in **R** is related to one or more tuples in **S**.
E. all tuples in **R** are related to the same one tuple in **S**.

g. In translating from an entity-relationship (E-R) diagram to a relational schema, one piece of E-R logic that *cannot* be captured by primary keys, uniques, and foreign keys is
A. the weak entity.
B. any ternary relationship.
C. mandatory participation for one-time occurrence (that is, with the arrow).
D. mandatory participation for many-time occurrence (that is, without the arrow).
E. aggregation.

---

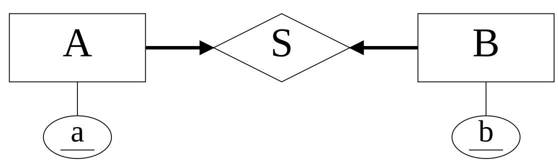
h. In an E-R diagram, if one sees a bold line with no arrow between an entity set and relationship set, this means
A. every entity in the entity set must participate in the relationship set.
B. every entity in the entity set must appear exactly once in the relationship set.
C. a relationship in the relationship set need not involve an entity from the entity set.
D. the entity set is weak.
E. the entity set *is an* instance of the relationship set.

---

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

---

j. Consider converting the following E-R diagram into a relational schema, and that we must have tables for both **A** and **B**.



A. To model **S**, we just need one foreign key from **A** to **B**.
B. To model **S**, we just need one foreign key from **B** to **A**.
C. To model **S**, we need a foreign key from **A** to **B** *and* a foreign key from **B** to **A**
D. It requires that we make a table for **S**.
E. Its logic cannot be captured properly by a relational schema.

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

---

l. Why are the normal forms useful?
  A. They are just a tool for checking whether our relational design makes sense or not.
  B. A schema in BCNF will always consist of fewer tables than an "equivalent" schema not in BCNF.
  C. They help us find anomalies in the data.
  D. They guarantee that certain types of data anomalies cannot occur.
  E. They are useless, but earn database consultants lots of money. (Don't tell anyone!)

---

m. If we know that table **Q** has only one candidate key, then which of the following is *true*?
  A. **Q** is in 2NF, but is not in 3NF.
  B. **Q** is in 2NF, but we cannot tell about 3NF.
  C. **Q** cannot be in BCNF.
  D. If **Q** is in 3NF, it is also in BCNF.
  E. None of the above.

---

n. Consider table **R** with attributes A, B, C, D, and E. How many possible candidate keys are there for **R**?
  A. 1
  B. 5
  C. 10
  D. 31
  E. 365

---

o. Consider table **R** with attributes A, B, C, D, and E. What is the largest number of candidate keys that **R** could have *at the same time*?
  A. 1
  B. 5
  C. 10
  D. 31
  E. 365

3. (15 points) **E/R Modeling.** *Revolving Schema.* [Exercise]

The Allegheny Medical Consortium has hired Duey, Cheatem, & Howe, Inc.'s (DC&H's) Information Systems Division to do a database design for them. They want to track medical interns through their rotations. DC&H has assigned Dr. Bas—and hence, you too—to the project.

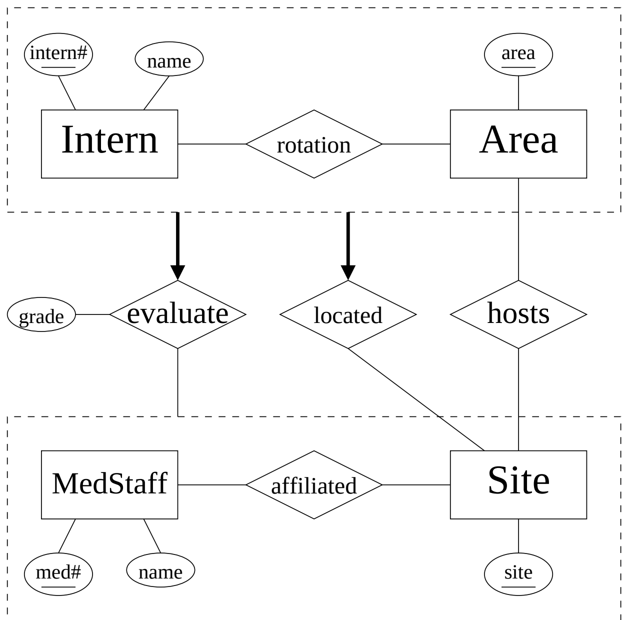


Figure 2: Dr. Bas's E-R for rotations.

Dr. Bas designed the E-R diagram in Figure 2 based on Allegheny's design requirements as below.

Medical *staff* members evaluate the medical *interns* who train at their *site*. An intern's stay at a site (for instance, a hospital, a medical research centre, or a clinic) is called a *rotation*. An intern is a medical student. Throughout the intern's training, he or she has half a dozen rotations. Each rotation is in a particular *area* (for instance, pediatrics, psychiatry, and internal medicine). The intern will have a rotation in each area, but *never* two rotations in the same area. Certain sites are equipped to *host* certain rotation areas. Ideally, the database should ensure that any rotation in a given area assigned to a site is, in fact, in an area that site hosts. Medical staff are *affiliated* with sites. A medical staff member can be affiliated with a number of sites, and of course, a site will have many medical staff.

For each intern's rotation, there is one medical staff member who oversees the rotation. At the end of an intern's rotation, this member of the medical staff evaluates the intern. Essentially, he or she gives a *grade* to the intern for the rotation.

a. (10 points) Write a relational schema for the rotation database based upon Dr. Bas's design in Figure 2 and the (English) specifications above.

- Use the abbreviated notation for schemas used in Figure 1 for Question 1 instead of writing out SQL CREATE statements.
- Do not create more tables than necessary. Be certain to account for mandatory participations.
- Ensure in your schema design that any **MedStaff** who evaluates an **Intern** for a **rotation** is, in fact, **affiliated** with the **Site** at which the **rotation** is **located**.

b. (3 points) A couple of days later, Allegheny Medical Consortium requests a change to the design: a panel of medical staff members—instead of just a single medical staff member—at the site evaluates an intern's rotation, still assigning the intern collectively a single grade for the rotation.

What changes do you make to your *relational design* for Question 3a for this?

Just discuss / show your changes. Do not entirely redo the design! And do not discuss changes to the E-R diagram! These are changes to your *relational schema*.

c. (2 points) What addition to your *relational design* for Question 3a could you make to ensure that, for any rotation in a given area located at a site, *that* site indeed hosts *that* area?

You are limited to relational design tools here: primary keys, unique, and foreign keys. (You may not use CHECK or ASSERTION constraints or other such techniques.)

Extra space.

4. (10 points) **Normal Forms.** *This just isn't normal!* [ANALYSIS]

Consider table **R** with attributes A, B, C, D, E, F, & G, and the set of functional dependencies

$$\begin{array}{rcl} \mathsf{BG} \mapsto \mathsf{AC} & \quad & \mathsf{A} \mapsto \mathsf{G} \\ \mathsf{ABG} \mapsto \mathsf{DF} & & \mathsf{D} \mapsto \mathsf{E} \\ & & \mathsf{E} \mapsto \mathsf{C} \end{array}$$

a. (3 points) Is AD a candidate key? Prove your answer.

b. (2 points) Is BDG a superkey? Prove your answer.

c. (2 points) Does $\mathsf{ADC} \mapsto \mathsf{F}$ logically follow from the set of functional dependencies? Prove your answer.

d. (3 points) Determine whether **R** is in 2NF, 3NF, and BCNF.

EXTRA SPACE.

1NF: Domain of each attribute is an *elementary* type; that is, not a *set* or a *record structure*.

2NF: Whenever $\mathcal{X} \mapsto A$ is a functional dependency that holds in relation $\mathbf{R}$ and $A \notin \mathcal{X}$, then either

- A is *prime*, or
- $\mathcal{X}$ is not a proper subset of any key for $\mathbf{R}$.

3NF: Whenever $\mathcal{X} \mapsto A$ is a functional dependency that holds in relation $\mathbf{R}$ and $A \notin \mathcal{X}$, then either

- A is *prime*, or
- $\mathcal{X}$ is a key or a super-key for $\mathbf{R}$.

BCNF: Whenever $\mathcal{X} \mapsto A$ is a functional dependency that holds in relation $\mathbf{R}$ and $A \notin \mathcal{X}$, then

- $\mathcal{X}$ is a key or a super-key for $\mathbf{R}$.

An attribute A is called *prime* if A is in any of the candidate keys.

Extra space.

Relax. Turn in your exam. Go home.