EECS 484: Sample Midterm Exam

Please ***print*** your name and uniqname below.

Name:
Uniqname:

Instructions:

1. This is a closed book exam. The use of electronic devices is not permitted on this exam. But you are allowed to use bring hand-written notes on one double-sided 8.5x11 sheet of paper with you.
2. You have 80 minutes to complete this exam. The points on this exam total to **100**.
3. Read the question carefully, and only answer what is asked in the question.
4. There are no syntax errors in the queries given in the exam. If you see any, ask us to correct them, if the correction is not obvious to you. Questions on the exam are about semantics, not syntax.
5. At the end of the exam, sign the Honor Pledge given below before turning in the exam.

Honor Code Pledge: I have neither given nor received assistance in this exam, nor have I concealed any violation of the Engineering Honor Code.

Signature:

(For Instructors Use)

Q1: Relational Calculus (5 points)	
Q2: Relational Model (15 points)	
Q3: E-R and Relational Modeling (20 points)	
Q4: Relational Algebra (10 points)	
Q5: SQL Queries (30 points)	
Q6: Functional Dependencies (20 points)	
TOTAL (100 points)	

Question 1. (5 points) Relational Calculus

Consider the following relational schema for a library (primary keys are underlined), and use it to answer the following questions.

```
Member (mid, name, dob)
Textbook (isbn, title, author, publisher)
Borrows (mid, isbn, date)
```

Write a tuple relational calculus expression to find the names (`name`) and date of births (`dob`) of members who have borrowed textbooks published by 'Prentice Hall'.

Question 2. (15 points) Relational Model

a) (5 points) Consider the following relation in a database:

```
Customer(name VARCHAR(24),
         email VARCHAR (32),
         phone VARCHAR (12),
         ssn INTEGER UNIQUE,
         PRIMARY KEY (email),
         FOREIGN KEY (ssn) REFERENCES Person(ssn)
         );

Person(ssn INTEGER PRIMARY KEY,
        name VARCHAR(32)
       );
```

For each instance of Customer shown below, select or circle if the instance is legal or illegal, just based on the information in the relation. NULL is the SQL null value. Assume that other values shown are of appropriate types.

name	email	phone	ssn
joe	joe@umich.edu	123	123
smith	NULL	123	345

☐ Legal ☐ Illegal

name	email	phone	ssn
joe	joe@umich.edu	123	123
jane	jane	123	123

☐ Legal ☐ Illegal

name	email	phone	ssn
joe	joe@umich.edu	123	123
NULL	jane@umich.edu	123	345

☐ Legal ☐ Illegal

name	email	phone	ssn
joe	joe@umich.edu	123	123
jane	jane@umich.edu	123	NULL

☐ Legal ☐ Illegal

name	email	phone	ssn
joe	joe	123	123
jane	jane@umich.edu	123	345

☐ Legal ☐ Illegal

b) (2 points). Is `ssn` a candidate key in the Customer relation? Justify your answer.

c) (6 points) The `Customer` relation has a single record whose `ssn` is 123. Also, the `Person` relation has a single record whose `ssn` is also 123. Describe the specific change you would need to either one or both schemas so that both the records in the two tables can be deleted using a single DELETE command. Also give that DELETE command. To save you some typing, we repeat the existing schemas below. You can mark your changes on the schemas.

-- Mark the edits in the schemas below. *If there is no change to a table, say so as well.*

```
Customer(name VARCHAR(24),
         email VARCHAR (32),
         phone VARCHAR (12),
         ssn INTEGER UNIQUE,
         PRIMARY KEY (email),
         FOREIGN KEY (ssn) REFERENCES Person(ssn)
         );


Person(ssn INTEGER PRIMARY KEY,
       name VARCHAR(32)
       );
```

-- The single DELETE command to delete both records:

d) (2 points) Following two options are suggested to you to remove `Customer` and `Person` relations from the system (not just empty them, but *remove* them).

Option 1:
```
DROP TABLE Customer;
DROP TABLE Person;
```

Option 2:
```
DROP TABLE Person;
DROP TABLE Customer;
```

Assume that the only constraints are those that are implied by the given schemas in part (a). Which of the above options will succeed in removing the two relations? Assume there are no additional triggers. Select **one** best answer from the choices below by circling the choice:

i. Option 1 only

ii. Option 2 only

iii. Both options 1 and 2 will work.

iv. Neither option 1 nor 2 will work.

Question 3. (20 points) E-R and Relational Modeling

A university database contains information about **professors**, identified by an ID number, an integer. A **professor's** other attributes are *lastname* and *firstname.* The database also contains information about **courses**, which are identified by a *courseid,* an integer. A **course's** other attributes are *courseName* and *departmentName*. The database also contains information about semesters, identified by an ID, *semid*. Semester's other attributes are *semesterName* and *year*. Here are the constraints that your design should assume:

- A course may or may not be offered in a given semester. For example, EECS 489 may or not be offered in Winter 2014.
- A course, when offered in a semester, can be taught by **exactly one** professor. For example, EECS 489 in Winter 2014, if offered, must be assigned to exactly one professor. Note: The same course can be taught in different semesters, possibly by same or different professors.
- A professor can teach any number of courses each semester, including not teaching at all. No restrictions.

Note: There are no sections for courses. A course, if offered in a semester, has only one offering.

a) (10 points) Give the ER diagram that best represents the given specs using the notation that we have been using in EECS 484.

b) (10 points) Give the best relational model by giving the schema for the relations. **Make sure you specify the primary key, and any additional integrity and key constraints unambiguously.** *If you wish, you can omit types such as INTEGER or VARCHAR for any attributes for brevity.* **Note:** we will grade this independently of part (a). If you get part (a) wrong, you must still try to give the right answer for this part that satisfies the specs.

Here is an example from an unrelated database to remind you of the format of CREATE TABLE with the types omitted for brevity:

```
products (product_id PRIMARY KEY, supplier_id NOT NULL, FOREIGN KEY (supplier_id)
REFERENCES supplier(supplier_id) );
```

Problem 2 also contains additional examples of schemas, though that contains types as well.

Question 4. Relational Algebra (10 Points)

You are given the following relations (primary keys are underlined):

```
Student (sid, name, age)
Take (sid, cid, grade)
Course (cid, title)
```

Write a relational algebra expression to find the `sid` and `name` of students who are taking both a course with title 'Databases' and a course with title 'Calculus'.

Question 5. SQL Queries (30 Points)

Consider the following relational schema (primary keys are underlined), and use it to answer the following questions.

```
Student (sid, name, age)
Take (sid, cid, grade)
Course (cid, title)
```

The `Take` relation gives information on courses taken by students and the grade received.

a) (5 points) Write a SQL query to find the `sid` of students who have taken a course titled 'Databases'.

Schema repeated for convenience:

```
Student(sid, name, age)
Take(sid, cid, grade)
Course(cid, title)
```

b) (15 points) Write a SQL query to find the `sid` of students who took the same course as any course taken by a student who took a course titled 'Databases'. You may assume that a student takes a course with himself or herself (thus, every student who has taken Databases automatically qualifies). For brevity, you can use the query from part a) in your query here, using the name DBSTUDENTS as a shorthand for that query. We will do a literal substitution of any use of DBSTUDENTS with your query from a). You can use views.

Example: Let's say a student s1 has taken both Databases and Compilers, student s2 has taken Compilers, s3 has taken only Databases, and s4 has taken only OS. In this instance, s1, s2, and s3 will be in the result set.

Schema repeated for convenience:

```
Student(sid, name, age)
Take(sid, cid, grade)
Course(cid, title)
```

c) (10 points) Write a SQL query to find the `cid` of courses for which the count of students over age 20 who took that course is more than 50.

Question 6. (20 points) Using Functional Dependencies for Database Design

Suppose you are given the following relation:

R: ABCDE

The following functional dependencies must hold:

BC → A
A → BCD
B → E
E → B

a. (3 points) Give *all* the possible candidate or primary key(s) for the above relation. Hint: There is more than one candidate key. Try to get this right. Most other parts of this question depend on this.

b. (4 points) Which of the functional dependencies violate BCNF? Circle the right choices.

BC → A violates does not violate

A → BCD violates does not violate

B → E violates does not violate

E → B violates does not violate

c. (4 points) Which of the functional dependencies violate 3NF? Circle the right choices.

BC → A violates does not violate

A → BCD violates does not violate

B → E violates does not violate

E → B violates does not violate

d. (3 points) Give a decomposition of R into BCNF.

e. (3 points) Is the following decomposition of the original relation R *lossless*? Justify your answer.

R1: ABE
R2: ACD

f. (3 points) Is the following decomposition of the original relation R *dependency preserving*? Justify your answer.

R1: ABE
R2: ACD