Midterm Examination

- Please read all instructions (including these) carefully.
- There are four problems on the exam, with a varying number of points for each problem and subproblem for a total of 75 points. You should look through the entire exam before getting started, in order to plan your strategy. You have 75 minutes to complete the exam.
- The exam is closed book and closed notes, but you may refer to your three pages of prepared notes.
- Please write your solutions in the spaces provided on the exam. Make sure your solutions are neat and clearly marked. You may use the backs of the exam pages as scratch paper. Please do not use any additional scratch paper.
- *Simplicity and clarity of solutions will count.* You may get as few as 0 points for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.

NAME: ____

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

SIGNATURE: ____

Problem	Max points	Points
1	30	
2	30	
3	8	
4	7	
TOTAL	75	

1. **Entity-Relationship model and ODL** (30 points)

   Consider the following entity-relationship diagram representing appointments that include one doctor, one nurse, and one patient:

   (a) (3 points) Translate the E/R diagram to a relational schema using the method described in class and in the course notes. Underline key attributes for the relations derived from entity sets only.

   (b) (4 points) Suppose that each doctor, nurse, and patient can be scheduled for at most one appointment at a given date and time. For the relation you derived in part (a) corresponding to the relationship set in the E/R diagram, list all of the possible minimal keys.

(c) (10 points) Specify an ODL schema that captures the database design in the above E/R diagram as closely as possible. Assume that names, dates, and times are represented as strings, while IDs and SSNs are represented as integers. It is not necessary to specify keys. You will be graded on clarity and simplicity as well as on correctness.

(d) (7 points) Translate your ODL schema from part (c) to a relational schema. Use the method described in class where ODL relationship pairs are translated to a separate relation, rather than the technique described in the course notes where relationship pairs are represented within the relations for the classes. For the translation you may need to make assumptions about keys for your ODL classes. Please state any assumptions you make about keys if (and only if) the assumptions differ from or are in addition to keys specified in the original E/R diagram. If you need to add any new attributes to your ODL schema for the translation to "work," please do so. It is not necessary to underline key attributes in the relational schema.

(e) (3 points) Ignoring the issue of keys, is there any database instance using your relational schema from part (a) that cannot be represented in your schema from part (d)? Assume that null values are not permitted in relations. If so, show the simplest such instance you can find.

(f) (3 points) Ignoring the issue of keys, is there any database instance using your relational schema from part (d) that cannot be represented in your schema from part (a)? Assume that null values are not permitted in relations. If so, show the simplest such instance you can find.

2. **Queries in relational algebra and SQL** (30 points)

   Consider the following relational schema:

```
course(course#, dept-name)      // course# is the key
enroll(studentID, course#)      // <studentID,course#> is the key
```

   (a) (7 points) Write a relational algebra expression to find the IDs of all students who are not enrolled in course# 145.

   (b) (8 points) Write a relational algebra expression to find the IDs of all students who are enrolled in courses in at least two different departments.

(c) (7 points) Write an SQL query to find the IDs of all students who are not enrolled in any courses in the EE department. Do not use the SQL **except** operator.

(d) (8 points) Write an SQL query to find the names of all departments that offer at least 20 courses.

3. **More relational algebra and SQL** (8 points)

There is a relational operator called "outerjoin" that we have not (yet) studied in class. The outerjoin of two relations $R(A, B)$ and $S(B, C)$ is defined as follows: We take the natural join $R \bowtie S$. We add to the result a tuple $(a, b, null)$ for every tuple (a, b) in R such that (a, b) does not participate in the result of $R \bowtie S$ (i.e., there is no joining value for b in S), and we add to the result a tuple $(null, b, c)$ for every tuple (b, c) in R such that (b, c) does not participate in the result of $R \bowtie S$ (i.e., there is no joining value for b in R). Write an SQL query that computes the outerjoin of relations $R(A, B)$ and $S(B, C)$, without using the SQL `outerjoin` operator of course. You may `select` the constant value `NULL` if you find it useful; for example, the following query returns all tuples in $Q(A, B, C)$ with all B values replaced by `NULL`:

```
select A,NULL,C from Q
```

Write the SQL query for outerjoin here:

4. **Indexes** (7 points)

Consider again the schema from Problem 2:

```
course(course#, dept-name)      // course# is the key
enroll(studentID, course#)      // <studentID,course#> is the key
```

Suppose there are three types of queries commonly asked on this schema:

(1) Given a course#, find the name of the department offering that course.

(2) Find all students together with all of the departments they are taking courses in; i.e., effectively take the natural join of `enroll` and `course`.

(3) Given a studentID, find all courses the student is enrolled in.

Here's the problem:

(a) What is the minimal number of indexes needed to speed up all three types of queries?

(b) On which attributes should these indexes be created?