Midterm Examination

- Please read all instructions (including these) carefully.
- There are 5 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: ______________________________

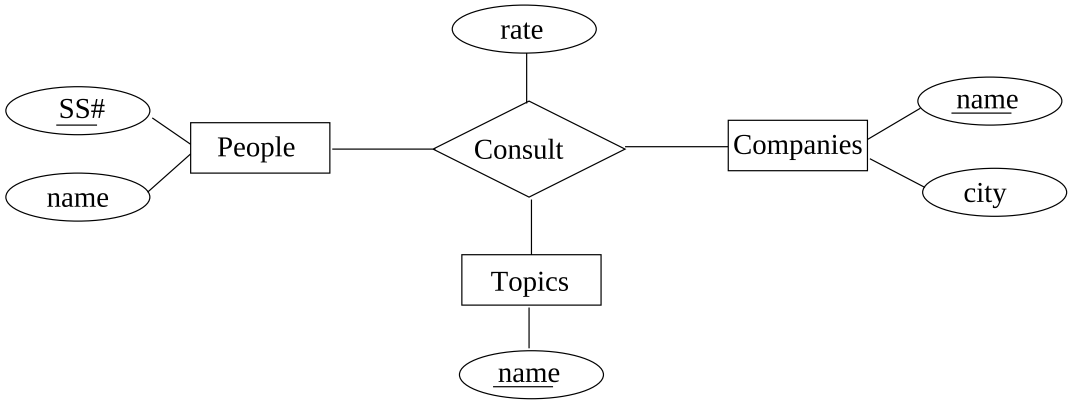
In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

SIGNATURE: ______________________________

Problem	Max points	Points
1	18	
2	18	
3	12	
4	15	
5	12	
TOTAL	75	

1. **Entity-Relationship Model and ODL** (18 points)

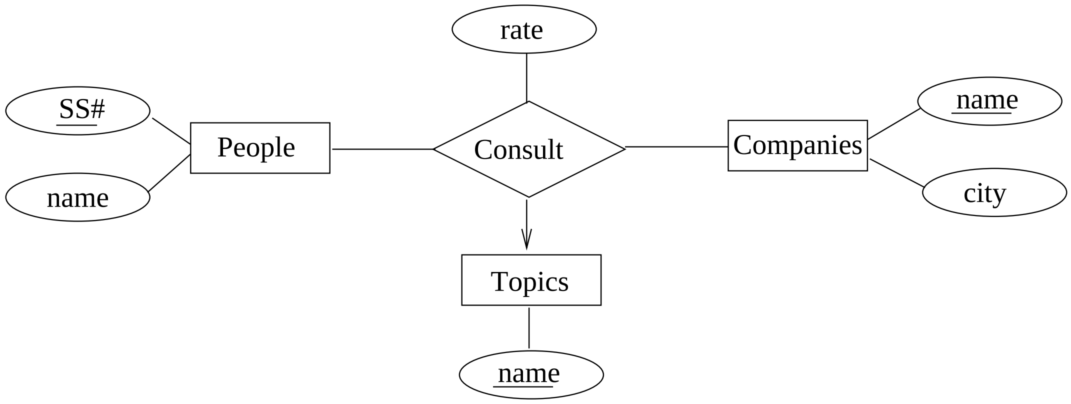
   Consider the following entity-relationship diagram:



   Three-way relationships in *Consult* represent that certain people consult for certain companies on certain topics, e.g., Joe consults for IBM on database design, and Mary consults for Microsoft on SQL. The relationships also record the hourly rate for each consulting agreement.

   (a) (10 points) Specify an ODL design for the same information. Your design should use four classes: one corresponding to each entity set in the E/R diagram above and one corresponding to the relationship set.

   (b) (2 points) Now suppose we add an arrow into the *Topics* entity set:



   Briefly state in English the real-world assumption that is encoded when we add this arrow.

   (c) (2 points) Without adding any new classes or relationships, can you modify your ODL design in part (a) to include the constraint added in part (b)? Circle either **Yes** or **No**. If Yes, briefly describe the modifications you need to make.

   (d) (4 points) Using the standard translation from E/R diagrams to relations, specify the relation for the *Consult* relationship set. Assuming the constraint added in part (b), also specify a key for the relation.

2. **Dependencies and Normal Forms** (18 points – 3 for each subpart)

   Consider the following two relational schemas:

   Schema 1: $R(A, B, C)$

   Schema 2: $R_1(A, B)$, $R_2(A, C)$

   (a) Suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $A \to BC$ and all dependencies that follow from this one.

      i. Schema 1 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these
      ii. Schema 2 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

   (b) Now suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $BC \to A$, $A \to C$, and all dependencies that follow from these two.

      i. Schema 1 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these
      ii. Schema 2 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

   (c) Now suppose that the only dependencies (functional or multivalued) that hold on the relations in these schemas are $A \twoheadrightarrow B$, $A \twoheadrightarrow C$, and all dependencies that follow from these two.

      i. Schema 1 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these
      ii. Schema 2 is in (circle the strongest one): 3NF, BCNF, 4NF, none of these

3. **Relational Database Design** (12 points)

   Consider the following poorly designed relational schema:

   `UnivInfo(studID, studName, course, profID, profOffice)`

   Each tuple in relation `UnivInfo` encodes the fact that the student with the given ID and name took the given course from the professor with the given ID and office. Assume that students have unique ID's but not necessarily unique names, and professors have unique ID's but not necessarily unique offices. Each student has one name; each professor has one office.

   (a) (4 points) Specify a set of completely nontrivial functional dependencies for relation `UnivInfo` that encodes the assumptions described above but no additional assumptions.

(b) (4 points) Decompose relation `UnivInfo` into BCNF according to your functional dependencies in part (a).

(c) (4 points) Now add the following two assumptions: (1) No student takes two different courses from the same professor; (2) No course is taught by more than one professor. Modify your set of functional dependencies from part (a) to take these new assumptions into account.

4. **Relational Algebra and SQL** (15 points – 3 for each part)

Consider two relations, $R(A, B)$ and $S(A, B)$. For each of the following pairs of queries, circle whether or not the two queries are *equivalent*, i.e., whether or not they are guaranteed to give the same answer over all possible databases.

(a) Query 1: $\pi_A(R \cup S)$
Query 2: $\pi_A(R) \cup \pi_A(S)$

Equivalent or **Not equivalent**

(b) Query 1: $\pi_A(R - S)$
Query 2: $\pi_A(R) - \pi_A(S)$

Equivalent or **Not equivalent**

(c) Query 1: $\sigma_{A=5}(R)$
Query 2: $\sigma_{A=5}(\pi_{R.A,R.B}(R \bowtie S))$

Equivalent or **Not equivalent**

(d) Query 1: $\pi_{R.A}(\sigma_{R.B=S.B}(R \times S))$
Query 2: `select R.A from R, S where R.B = S.B`

Equivalent or **Not equivalent**

(e) Query 1: `select R.A from R, S where R.B = S.B`
Query 2: `select R.A from R where R.B in (select B from S)`

Equivalent or **Not equivalent**

5. **Queries in Relational Algebra** (12 points)

Consider the following relational schema:

```
student(ID, major)          // <ID,major> is key
advisor(studID, profName)   // studID is key
prof(name, dept)            // name is key
```

Note that a student may have many majors but only one advisor, and a professor has only one department.

(a) (6 points) Write a relational algebra expression to find the ID's of all students whose advisor's department is one of the student's majors.

(b) (6 points) Write a relational algebra expression to find the ID's of all students whose advisor's department is not one of the student's majors.