CSE 560: Project #1 (due 10/13/08)

You are given the following relational schema (keys underlined):

```
Article(Title,JournalName,Year)
Author(AuthorName,Affiliation)
Authoring(AuthorName,Title)
Bibliography(Title,ReferencedTitle)
Journal(JournalName,Area)
```

The attribute `JournalName` in `Article` is a foreign key referencing `JournalName` in `Journal`. The attribute `AuthorName` in `Authoring` is a foreign key referencing `AuthorName` in `Author` The attribute `Title` in `Authoring` is a forein key referencing `Title` in `Article`. The attributes `Title` and `ReferencedTitle` in `Bibliography` are foreign keys referencing `Title` in `Article`.

Problem 1 (50 pts)

Write the following queries in SQL2, defining appropriate views if necessary:

1. Return the titles of all the papers in Biology published in 2008.
2. For each area and year, return the number of articles published in that area that year (if the number is greater than zero).
3. Return the titles of all articles written exclusively by authors affiliated with UB.
4. Find all the authors that had at least 10 papers, each of which was directly referenced by at least 10 other papers.
5. Return all the journals in the "Data Models" area if at least one is present in the database, otherwise return all the journals. Do not use IF-THEN-ELSE.

Create a test database, run the queries against it, and report the results.

Problem 2 (50 pts)

Implement the following transactions using Oracle JDBC:

1. List the bottom 5 journals in terms of the number of published articles (ties should be broken arbitrarily);
2. Find all the articles in the Biology area that reference, directly or indirectly, only articles in the Biology area. You may assume that there at most 50 tuples in the `Bibliography` relation.

Run the queries against the test database and report the results.

Problem 3 (Extra credit: 20 pts)

You are given a relation with N columns of the same type. Write an SQL2 query that returns the tuples having the *maximum number of different values*. The query should have size polynomial in N.