Midterm Examination

- Please read all instructions (including these) carefully.
- There are seven problems on the exam, with a varying number of points for each problem. You should look through the entire exam before getting started, in order to plan your strategy. You have one hour and fifteen 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	15	
2	15	
3	20	
4	20	
5	15	
6	10	
7	5	
TOTAL	100	

1. **Relational schema to entity-relationship schema** (15 points)

   Consider the following relational database schema that combines both current and historical information about students and their courses. A course is uniquely identified by its code (e.g., "CS145"), and a student is uniquely identified by his or her ID#. You may assume that students are enrolled only in courses that are currently offered (although this assumption is not enforced by the relational schema). Don't make any other assumptions except for those implied by the key attributes.

   ```
   course(code, #units)                 // all courses
   student(ID#, name)                   // all students
   offered(code, time, room)            // courses currently offered
   taking(ID#, code)                    // current enrollment information
   took(ID#, code, quarter, grade)      // past enrollment information
   ```

   Draw an entity-relationship diagram that captures the same information as this relational database schema. Underline key attributes for entity sets. (You need not indicate keys for relationship sets.) Do not use weak entity sets. Your schema may include repeated information if necessary. Indicate all mapping constraints. Note that the translation from your E-R diagram back into a relational schema need not yield exactly the relations above.

2. **Relational schema to object-oriented schema** (15 points)

   Consider again the relational database schema from Problem 1, with the same assumptions.

```
course(code, #units)                   // all courses
student(ID#, name)                     // all students
offered(code, time, room)              // courses currently offered
taking(ID#, code)                      // current enrollment information
took(ID#, code, quarter, grade)        // past enrollment information
```

   Design a simple object-oriented database schema that captures the same information as this relational database schema. Give class declarations and a class hierarchy diagram. Use at least one complex object type and at least one instance of inheritance. Do not include methods.

3. **Queries in relational algebra and calculus** (20 points, 5 per query)

Consider again the relational database schema from Problem 1, with the same assumptions.

```
course(code, #units)                 // all courses
student(ID#, name)                   // all students
offered(code, time, room)            // courses currently offered
taking(ID#, code)                    // current enrollment information
took(ID#, code, quarter, grade)      // past enrollment information
```

(a) Write an expression in relational algebra that produces the names of all students who are currently taking CS145 and have taken CS145 at least once in the past.

(b) Write an expression in domain relational calculus that produces the ID#'s of all students who have taken CS145 more than once in the past.

(c) Write an expression in relational algebra that produces the codes of all courses that are currently offered but have no students enrolled.

(d) (Same query as part (c)) Write an expression in domain relational calculus that produces the codes of all courses that are currently offered but have no students enrolled.

4. **Network data model** (20 points)

Consider the following schema declaration in the network (DBTG) model for a database of information about a library. Assume that a borrower may borrow many books, a book may be loaned to at most one borrower, a book may have many authors, and an author may have written many books.

```
Borrower = record                set name is Loaned
             ID#: integer          owner is Borrower
             name: string          member is Book

Book = record                    set name is WrittenBy
         call#: integer            owner is Book
         title: string             member is AuthorLink

Author = record                  set name is Wrote
           name: string            owner is Author
                                   member is AuthorLink
AuthorLink = record
               <no attributes>
```

(a) (5 points) Draw the data structure diagram corresponding to this schema. You may use the notation for data structure diagrams from the textbook or you may use the simplified notation from class.

(b) (5 points) Draw a simple database instance that conforms to this schema and illustrates data that is graph-structured, not just tree-structured.

(c) (10 points) Using the language for the network model covered in class and in the textbook, write a query/program to print the names of all authors of books currently on loan to a borrower named "JW." You may assume that there is only one borrower in the database named JW. If JW has borrowed several books by the same author, you may print the author's name several times.

5. **Hierarchical data model** (15 points)

   (a) (5 points) Using the hierarchical model, give a tree structure diagram that captures the same information that was captured by the network model schema in Problem 4. You may use the notation for tree structure diagrams from the textbook or you may use the simplified notation from class.

   (b) (10 points) Using your schema from part (1), and using the language for the hierarchical model covered in class and in the textbook, write a query/program to print the names of all authors of books currently on loan to a borrower named "JW." As in Problem 4, you may assume that there is only one borrower in the database named JW, and if JW has borrowed several books by the same author you may print the author's name several times.

6. **Relational algebra equivalences** (10 points)

   Consider a relational database schema with two relations: $R_1(A, B)$ and $R_2(B, C)$. All attributes are integers. Make no assumptions about keys. Consider the following three relational algebra queries.

   (a) $\pi_{A,C}(R_1 \bowtie \sigma_{B=1}(R_2))$

   (b) $\pi_A(\sigma_{B=1}(R_1)) \times \pi_C(\sigma_{B=1}(R_2))$

   (c) $\pi_{A,C}(\pi_A(R_1) \times \sigma_{B=1}(R_2))$

   Two of these three queries are equivalent, while one of them can produce a different answer. Which query can produce a different answer? Give the simplest database instance you can think of where a different answer is produced.

7. **SQL** (5 points)

   Consider for one last time the relational database schema from Problem 1, with the same assumptions.

```
course(code, #units)               // all courses
student(ID#, name)                 // all students
offered(code, time, room)          // courses currently offered
taking(ID#, code)                  // current enrollment information
took(ID#, code, quarter, grade)    // past enrollment information
```

   Using this schema, write the simplest SQL query you can think of for which an equivalent query cannot be expressed in relational algebra.