1. [1] Which of the following is not true concerning relational algebra?
   a) Relational algebra contains a collection of operations that are used to manipulate entire relations.
   b) The result of each relational algebraic operation is a new relation that can be further manipulated.
   c) Results of relational algebraic operations can include multisets, which are collections of tuples where duplicate tuples can exist.
   d) Relational algebraic operations can combine tuples from several relations for the purpose of specifying a query.

2. [3] Relational algebraic operations are usually divided into two groups. One group includes set operations from mathematical set theory (set theoretic operations), and the other group includes operations developed specifically for relational databases (relational operations). Name three operations from each group in the table below.

<table>
<thead>
<tr>
<th>Relational Operations</th>
<th>Set Theoretic Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projection</td>
<td>(Any of the following): Union, Intersection, Difference</td>
</tr>
<tr>
<td>Selection</td>
<td></td>
</tr>
<tr>
<td>Join</td>
<td>Cartesian Product, Outer Union</td>
</tr>
</tbody>
</table>

3. [2] Assuming two relations, RA and RB, for which partial schemas are shown below, can RA \( \cap \) RB be performed? _________ Why or why not? (Answer below, under “Explanation.”)

```
<table>
<thead>
<tr>
<th>RA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EmpID</td>
<td>LastName</td>
</tr>
<tr>
<td>RB</td>
<td></td>
</tr>
<tr>
<td>DeptID</td>
<td>DeptName</td>
</tr>
</tbody>
</table>
```

Explanation:
RA and RB are not union compatible
4. In a, b & c, below, write out, in relational algebraic notation, the *relational algebraic expressions* (not SQL statements) that can be performed on the relations of the database shown below to yield the results required in a, b & c. You must use the proper symbols, and you must include all aspects of the operation(s). (It’s OK to name and re-use results if needed.)

a) [2] List the names (first and last) and phone numbers of all students in Massachusetts (‘MA’).

\[ \pi_{\text{FirstName}, \text{LastName}, \text{Phone}} (\sigma_{\text{State} = \text{‘MA’}} (\text{Student})) \]

b) [4] List the names of all courses in which students appearing in the results of question 4a, are enrolled.

\[
\begin{align*}
R1 & \leftarrow \sigma_{\text{State} = \text{‘MA’}} (\text{Student}) \\
R2 & \leftarrow (R1) \bowtie \sigma_{\text{StudentID} = \text{StudentID}} (\text{Enrollment}) \\
R3 & \leftarrow (\text{Course}) \bowtie \sigma_{\text{CourseID} = \text{CourseID}} (\text{Section}) \\
R4 & \leftarrow (R3) \bowtie \sigma_{\text{CourseID} = \text{CourseID}} (R2) \\
\text{Answer} & \leftarrow \pi_{\text{CourseName}} (R4)
\end{align*}
\]
This is a closed-book test. Write your name on every page in the space provided. You have 3 hours in which to complete it. For each of the multiple-choice questions, circle only one answer, unless otherwise instructed. For each of the True/False questions, circle one of the words True or False which corresponds to your answer. For each question regarding a written answer, write your answer in the space provided below the answer. If you require more space, please indicate with an arrow, continue on the back of the page, and identify the question you are answering in your continuation. Feel free to draw diagrams, etc., if needed. Bracketed numbers next to a question number indicates the points value of the question.

c) [4] List the names (first and last) and phone numbers of all instructors teaching either one or both of the following courses: “Finding Happiness in Database Design” and “Australia Travels.”

\[
\begin{align*}
R1 &\leftarrow (\text{Instructor}) \bowtie_{\text{InstructorID} = \text{InstructorID}} (\text{Section}) \\
R2 &\leftarrow (R1) \bowtie_{\text{CourseID} = \text{CourseID}} (\text{Course}) \\
R3 &\leftarrow \sigma_{\text{CourseName} = ‘Finding Happiness in Database Design’ \text{ OR } \text{CourseName} = ‘Australia Travels’}(R2) \\
\text{Answer} &\leftarrow \pi_{\text{LastName}, \text{FirstName}, \text{Phone}}(R4)
\end{align*}
\]

5. [1] Using the database shown in question 4, suppose the dean wishes to see a list of instructors and all of their sections, but also wishes to see which instructors are not teaching any sections. What is the name of a type of join that would be used for this? _______ Outer Join _______

6. [2] Using the database shown in question 4, what is a possible question in English that may have generated the following query?

\[
\pi_{\text{CourseName}}(\sigma_{\text{RoomNumber} = ‘510B’}(\text{Section}) \bowtie_{\text{CourseID} = \text{CourseID}} (\text{Course}))
\]

What are the names of all courses held in room 510B?

7. [1] True  False  
The result of an SQL query is always a valid relation state.
8. [4] Assuming that the Create Table command shown below is valid, which of the following statements cannot be said to be true based only on the Create Table command shown?
   (circle all that apply)
   a) There is at least one referential integrity constraint in the database schema to which this table belongs.
   b) There is at least one foreign key in the Instructor table.
   c) A referential integrity constraint can be removed from the Department table by issuing a statement that includes the clause DROP CONSTRAINT DEPTCHAIRFK
   d) If a row in the Instructor table is deleted, any associated row in the department table will not be deleted as a result.
   e) There is a N:M relationship between the Department and Instructor tables.
   f) Suppose I offered to give you $1,000 to correctly guess the name of the primary key in the Instructor table. Assuming I was serious about giving you the $1,000 and that you want the $1,000, your best guess would be “InstID.”

   CREATE TABLE Department (DepartmentID CHAR(10) NOT NULL,
   DepartmentName VARCHAR(20) NOT NULL,
   DepartmentChair VARCHAR(10) NOT NULL,
   CONSTRAINT DeptPK PRIMARY KEY (DepartmentID),
   CONSTRAINT DeptK1 UNIQUE(DepartmentName),
   CONSTRAINT DeptChairFK FOREIGN KEY (DepartmentChair) REFERENCES Instructor (InstructorID)
   ON DELETE CASCADE
   );

9. Write SQL query statements for each of the queries listed in question 4. (They have been repeated here for your convenience.)
   a) [2] List the names (first and last) and phone numbers of all students in Massachusetts (‘MA’).

   SELECT FirstName, LastName, Phone
   FROM Student
   WHERE State = ‘MA’;
b) [4] List the names of all courses in which students appearing in the results of a, above, are enrolled.

```
SELECT CourseName
FROM ((Course INNER JOIN Section ON Course.CourseID = Section.CourseID) INNER JOIN Enrollment ON Section.SectionID = Enrollment.SectionID) INNER JOIN Student ON Enrollment.StudentID = Student.StudentID
WHERE State = ‘MA’;
-OR-
SELECT CourseName
FROM Course, Section, Enrollment, Student
WHERE Course.CourseID = Section.CourseID AND Section.SectionID = Enrollment.SectionID AND Enrollment.StudentID = Student.StudentID AND State = ‘MA’;
```

c) [4] List the names (first and last) and phone numbers of all instructors teaching either one or both of the following courses: “Finding Happiness in Database Design” and “Australia Travels.”

```
SELECT FirstName, LastName, Phone
FROM (Course INNER JOIN Section ON Course.CourseID = Section.CourseID) INNER JOIN Instructor ON Section.InstructorID = Instructor.InstructorID
WHERE CourseName = ‘Finding Happiness in Database Design’ OR CourseName = ‘Australia Travels’;
-OR-
SELECT FirstName, LastName, Phone
FROM Course, Section, Instructor
WHERE Course.CourseID = Section.CourseID AND Section.InstructorID = Instructor.InstructorID AND CourseName = ‘Finding Happiness in Database Design’ OR CourseName = ‘Australia Travels’;
```

10. Using the database shown in question 4, write the following SQL queries:

a) [2] For each course, list the course id and the number of sections belonging to that course.

```
SELECT CourseID, Count(*)
FROM Section
GROUP BY CourseID;
```
b) [3] List only those departments in which the average units is less than 4.

```
SELECT DepartmentID, AVG(Units)
FROM Department INNER JOIN Course ON Department.DepartmentID = Course.DepartmentID
GROUP BY DepartmentID
HAVING AVG(Units) < 4;
-OR-
SELECT DepartmentID, AVG(Units)
FROM Department, Course
WHERE Department.DepartmentID = Course.DepartmentID
GROUP BY DepartmentID
HAVING AVG(Units) < 4;
```

11. Write a query for question 5 (project the instructors’ IDs and last names together with the section IDs) that will give the dean the information he/she desires, in the forms requested in a and b, below (question 5 has been repeated here for your convenience):

“…suppose the dean wishes to see a list of instructors and all of their sections, but also wishes to see which instructors are not teaching any sections.”

a) [2] Write the relational algebraic expression(s).

```
R1 ← (Instructor) ╕
      Instructor.InstructorID = Section.InstructorID(Section)
Answer ← π_Instructor.InstructorID, LastName, SectionID(R1)
```

b) [2] Write the SQL statement.

```
SELECT InstructorID, LastName, SectionID
FROM Instructor LEFT JOIN Section ON Instructor.InstructorID = Section.InstructorID;
-OR-
SELECT InstructorID, LastName, SectionID
FROM Section RIGHT JOIN Instructor ON Instructor.InstructorID = Section.InstructorID;
```

12. [1] True  False
A relation can only be said to be in a normal form if it is also in the previous normal form.

13. [1] True  False
While it is not enough to simply verify that a relation has no partial functional dependencies on a key to say that it is in 2NF (one must also verify that it is in 1NF), for BCNF it is not necessary to verify compliance with 2NF.
14. [2] Which of the following is not true?
   a) When normalizing a database, some relation schemas may be decomposed into smaller relation schemas.
   b) Normalization guarantees a good design for a database.
   c) Databases are not always normalized to the highest normal form.

15. [2] In the mineworld to which the relation schema shown below belongs, queries that show information for specific projects need to be performed. What, if any, is the lowest normal form violated in the following relation schema (Write “none” if none)? _________ (Explain your answer under “Explanation,” below.)

<table>
<thead>
<tr>
<th>EmployeeId</th>
<th>LastName</th>
<th>ProjectsAndHours</th>
</tr>
</thead>
</table>

Explanation:

Since queries that distinguish between projects are performed, it follows that the attribute, ProjectsAndHours, is not atomic. This alone is enough to conclude that there is a 1NF violation. Furthermore, since the name of the attribute is in the plural, this implies that the attribute is also multivalued, leading to the “table-inside-of-a-table” problem, again indicating a 1NF violation.

16. [0] The John Harvard Brewery
   a) is in Texas
   b) requires a boat to get there
   c) is just a quick walk from Harvard Yard down Dunster St on the right (next to the Au Bon Pain)
17. [4] For the relation schema shown below, each patient can have only one primary-care doctor. What is the lowest normal form violated, if any? _____ 3NF ___. Explain your answer under “Explanation,” below. Then, if you determine that a normal form is violated, describe or draw a solution that eliminates that normal form violation without data loss.

<table>
<thead>
<tr>
<th>PatientID</th>
<th>PrimaryCareDoctorID</th>
<th>PrimaryCareDoctorName</th>
<th>PatientName</th>
</tr>
</thead>
</table>

Explanation:

All attributes are simple attributes, i.e., they are atomic and are not multivalued, so the relation schema is in 1NF. Since the primary key is composed of only one attribute, there are no partial dependencies on it, so the relation schema is in 2NF. However, PrimaryCareDoctorID, which is functionally determined by the primary key (PatientID), itself determines PrimaryCareDoctorName, which describes a transitive dependency. Since neither PrimaryCareDoctorID or PrimaryCareDoctorName are prime attributes, this relation schema violates 3NF, which is the lowest normal form violated in this case.

Solution, if any:

<table>
<thead>
<tr>
<th>PatientID</th>
<th>PrimaryCareDoctorID</th>
<th>PatientName</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PrimaryCareDoctorID</th>
<th>PrimaryCareDoctorName</th>
</tr>
</thead>
</table>

(Any kind of diagram showing this kind of information is acceptable here. The arrow is not a requirement for the answer to this question.)
18. [4] In the following relation schema, each horse (HID) can participate in many races (RID). Each race yields different placements for the participating horses (PLA). Assuming that there cannot be any ties in any races, i.e., no two horses can occupy the same placement in the same race, does this relation schema violate any of the normal forms up to and including BCNF? _______ Explain your answer under “Explanation,” below. Include a discussion of this specific scenario in your answer.

<table>
<thead>
<tr>
<th>HID</th>
<th>RID</th>
<th>PLA</th>
</tr>
</thead>
</table>

Explanation:

All attributes are simple attributes, i.e., they are atomic and are not multivalued, so the relation schema is in 1NF. In addition, this relation schema has the following keys:

{HID, RID}
{RID, PLA}

and the following determinants:

{HID, RID} → PLA
{RID, PLA} → HID

Since both determinants are keys and the relation schema is in 1NF, it is also in BCNF.
BONUS [12] For the following description and queries, draw a relational database schema using the Enhanced Informal E-R diagram (the one with the crow’s feet) that is normalized to BCNF and show SQL statements for all of the queries listed.

**Description**
Fuzzy the squirrel has decided that it’s just stupid to put the usual effort into collecting and storing nuts, only to forget where all the nuts are or, what is worse, to have the nuts stolen by other squirrels. Fuzzy has organized his little squirrel friends into a cooperative community in which squirrels have responsibilities and, in turn, this maximizes their efforts and allows all of them more free time to do the things they enjoy (Fuzzy doesn’t yet realize that this is a myth; remember, he is just a squirrel). Here’s how Fuzzy has organized things:

First, all nuts are stored in holes in trees and all trees belong to one of many territories. Holes have a depth, which determines the number of nuts that can go into it. Trees are certain types, such as Maple, Oak, and so on, which squirrels like to know about for no particular reason, as well as a height. Each territory has a location. Locations are just descriptions, such as, “starts on the left bank of the creek and goes all the way up to the old barn” (after all, we’re talking about squirrels, here). Each territory has oversight responsibilities assigned to one squirrel. Fuzzy only assigns territorial responsibilities to squirrels that are above a certain age. Territorial responsibilities include many activities (described ahead) and so manager squirrels may recruit the help of other squirrels, who themselves may recruit the help of other squirrels in the performance of these activities. Squirrels are not too bright, and so they really don’t discriminate between manager squirrels or other squirrels, so anyone can get recruited to help with any part of any territory. (Fuzzy has recruited some cats to lend their more powerful brains to help smooth out this part of the system, but this is presenting it’s own problems—a few of Fuzzy’s friends haven’t been seen in some time. Anyway, Fuzzy is not going to deal with any of the stuff in these parenthesis right now. He’s just put the cats on unpaid leave.) Also, some squirrels don’t do anything because they are either lazy or don’t quite understand how the system works. That’s fine with Fuzzy, because at least they’re not stealing nuts as long as they are benefiting from the system. Every squirrel has collecting responsibilities in which they find nuts and bring them to the trees to be stored in the appropriate hole. Nothing is tracked regarding this last aspect.

Maintenance activities are always one of the following: building holes in the trees and labeling them for one kind of nut (because there are many types of nuts, and squirrels sometimes have cravings for a particular type of nut in the winter, squirrels need to know what kind of nut is stored in what holes—a particular type of nut can appear in many different holes, but may not appear in any holes in a particular season), cleaning holes, and counting the nuts in a hole. Fuzzy is always coming up with other maintenance tasks, so this list is dynamic. Also, it’s perfectly OK for different squirrels to perform different tasks on the same hole. When something happens to a hole, Fuzzy wants to know which one of the maintenance tasks happened, when, and by which squirrel; otherwise, squirrels (who, again, are not too bright to begin with) may perform the same maintenance on a hole before its time. The squirrel responsible simply looks over the maintenance records to determine what needs to be done when.
Because, as has been pointed out throughout, squirrels are not exactly related to Einstein, there are some problems. First, even though some squirrels are recruited by others, they sometimes get confused and end up working on some other tree either in the same territory or in another territory. Fuzzy is not too worried about this problem, because it doesn’t happen too often anyway, and, even when it does, the work is getting done. However, even when this happens, records of what squirrel worked on what hole of a tree doing what to it are properly kept.

A squirrel that has waken up to the obvious after thousands of years deserves help. Help Fuzzy with his records. Design a database that properly models his mini-world on the next page, then create SQL statements for the queries listed:

- What are the names of the squirrels that manage territories, the name of the territory they manage, and their locations?

```
SELECT Squirrel.Name, Territory.Name, Location
FROM Squirrel INNER JOIN Territory ON SquirrelID = MgrSquirrelID;
```

- Which trees have holes that have not had maintenance done in more than 5 days (you can assume calculations involving date values will yield days)?

```
SELECT DISTINCT TreeID
FROM Hole INNER JOIN Maintenance ON Hole.HoleID = Maintenance.HoleID
WHERE Date - MaintDate > 5;
```

- How many trees are in each territory?

```
SELECT TerritoryID, Count(*)
FROM Tree INNER JOIN Hole ON Hole.HoleID = Tree.TreeID
WHERE Date - MaintDate > 5
GROUP BY TerritoryID;
```

- Which squirrels have performed which maintenance activities?

```
SELECT MaintID, SquirrelID
FROM Maintenance
ORDER BY MaintID;
-OR-

...ORDER BY SquirrelID;
```
This is a closed-book test. Write your name on every page in the space provided. You have 3 hours in which to complete it. For each of the multiple-choice questions, circle only one answer, unless otherwise instructed. For each of the True/False questions, circle one of the words True or False which corresponds to your answer. For each question regarding a written answer, write your answer in the space provided below the answer. If you require more space, please indicate with an arrow, continue on the back of the page, and identify the question you are answering in your continuation. Feel free to draw diagrams, etc., if needed. Bracketed numbers next to a question number indicates the points value of the question.

Fuzzy’s database diagram:

(Names of entity-types and attributes may vary, of course!)