**CS 122 Final Exam Fall 2015**

Read the questions carefully. Many questions are similar but not identical to questions you have seen before.

During the exam, periodically save your work and submit it on CSNS. I will only look at the version with the latest timestamp.

**I. Short Answers, 3 points each. Answer each question in no more than 3 sentences.**

1. Although an aggregate function directly in a WHERE clause causes an error, you can avoid the error by putting the aggregate function in a subquery, as in the query SELECT SalesID, Lastname, Base FROM Salespeople WHERE Base < (SELECT AVG(Base) FROM Salespeople);

How does using a subquery avoid the error?

2. Consider a database which tracks students and the courses in which they enroll. A student has a last name, a first name, and a CIN. A course has a unique course ID, a department id, and a course name. A course many enroll any number of students, and a student may take any number of classes. How many tables would you use for this database, and what fields would each one contain?

3. Consider the query:

SELECT (NULL IS NULL) IS NULL = (NULL = (NULL IS NULL));

Why does this query return NULL? Explain how the expression is evaluated step by step.

4. What language is used to insert data into database tables? (Hint: the answer is not "SQL.")

**II. SQL to English, 7 points each. Describe in simple terms what these queries or commands do.**

5. SELECT T.TitleID, T.Title FROM Titles T WHERE T.ArtistID IN (SELECT ArtistID FROM Artists WHERE Region NOT IN (SELECT Region FROM Studios));

6. SELECT SA.SalesID, SA.Lastname FROM SalesPeople SA WHERE SA.SalesID IN (SELECT SalesID FROM Studios ST WHERE ST.StudioID IN(SELECT StudioID FROM Titles WHERE Genre = "Alternative"));

**III. English To SQL, 12 points each. Write queries or commands that find the information specified or make the changes requested to the database. Unless otherwise specified, answer with one query or command per question. Do not assume you already know any of the data; for example, if a question refers to a member's name, do not assume you already know the memberID.**

7. Show the names of all salespeople who do not represent studios where The Neurotics have recorded titles. Use one or more joins, do not use any subqueries.

8. Show the same information requested in the previous question, but

use subqueries instead of joins.

9. Find the age in years of the oldest female member. Assume each year contains 365.25 days.

10. (Use a transaction so you can restore the original data when you are testing this.) Roberto Alvarez has quit Word and joined the Bullets. Update the database accordingly.

**IV. Multi-Step Problem, 26 points**

This problem involves a series of sequential steps. Make sure to test

each step carefully before proceeding to the next.

Parts a through h, 3 points each:

a. Create a table called Courses. Each record should contain a CourseID, CourseName, InstructorCIN, and NumberOfUnits. Give each field an appropriate data type. Define the appropriate field as the primary key and make sure it automatically receives a unique value when each record is added to the table.

b. Add six records (you can use six separate commands). Give exactly two of them the InstructorCIN 98765 and exactly two the InstructorCIN 54321. Use whatever values you like for the other data.

c. Write a query to show all InstructorCIN with the number of courses taught by each instructor. Write your query so that the instructor CINs will appear in numeric order irrespective of whether the records happen to be in this order.

d. Add a new field called StartDate and give it an appropriate data type.

e. Supply the existing records with values in the StartDate field without changing any other data. You may set all the records to show the same value in order to make this step easier.

f. Change the table definition to change the name of the InstructorCIN field to InstructorID and to ensure that every record must have a value in this field (ie, the field can not be null for any record.)

g. Drop the column NumberOfUnits from the table.

h. Drop the entire table from the database.

i. 1 point: insert SELECT \* FROM COURSES on the table you create after parts b, e, f, and g in your solution to this part.

j. 1 point: Show the names of all tables in the database.