**CSCI 4125/5125**

**Data Models and Database Systems**

**Spring 2023**

**Course Project**

**Phase 3: E-R Mapping (9/1)**

**Due: Sunday, 9/10 @ 11:59pm**

**Reading:** SilberschatzChapters 6 and 2

**Submission Guidelines:**

1. This assignment is worth 50 points for all students.

2. All answers in the form of images or screenshots must be readable. Any additional files must be clearly referenced and labeled.

3. It is your responsibility to make sure all files are readable and submitted on time.

4. If you work with a team member, make sure you state this at the top of your submission. Each group member is still required to submit all required work and files.

**Submission:**

- Task 1 requires you to submit a relational schema (containing primary keys and foreign keys) worth 25 points.

- Task 2 requires you to submit a Java source file and an output file named “animal.sql” worth 25 points.

**Task 1. E-R Mapping (25 pts)**

Your task is to generate a complete logical schema for your E-R diagram from Phase 2 of the project. You may submit a drawing using your favorite software (e.g., PowerPoint) or clearly handwritten. Remember that all primary keys are to be underlined with a solid line, all foreign keys use a dotted underline (or clearly stated), and foreign keys must point to the primary key they reference. While you may have generated a slightly different E-R diagram in Phase 2, use the E-R diagram below to generate your relational schema.



**Task 2. Generating SQL Insert Statements (25 points)**

Writing SQL insert statements is tedious if you have a few dozen records and unrealistic if you have thousands or millions of records to populate. Also, picture a user interface (e.g., a website) you may develop for a typical, non-CS end user. You would embed SQL statements in the code and automatically generate SQL based on their input; you don’t want the typical user writing SQL! You will now modify your Java code from Phase 1 to automatically generate SQL INSERT statements.

Your Java program should include the following functionality:

1. It should accept a command line argument, which will be the table name that the record will be inserted into. **Note:** in this project, the table names will match the names of the text files I will provide to you. For example, I provided the animal.txt file, which contains data for the “animal” table. When you run **$java Phase3 animal**, your code should read the file animal.txt. **Hint:** string placeholders (e.g., **“%s.txt”**) will make things simple for you!
2. Datatypes should be correctly formatted in your insert statement. Do not hardcode the positions of values in files. In Phase 4, I will give you input text files with varying schemas. Your Java program should work for any schema.
   1. Numbers and floats don’t use single quotes.
   2. Strings use single quotes
   3. Also, consider NULL values now. NULL values do not use single quotes.

As an example, if an input line for my table, Example, is: hello, world, 4125, NULL, 5125, my output line should be:

**INSERT INTO Example VALUES('hello','world',4125,NULL,5125);**

**Hint:** Again, string placeholders can really simplify this for you. I recommend using a placeholder for your table name and the values inside the parenthesis (e.g., **INSERT INTO %s VALUES (%s);** ). Then you can simply place your table name and formatted values into the placeholder values.

Along with the animal.txt file, I included the table schema in animal\_schema.sql, which you can use this to test the INSERT statements you generate.

**Submit:** Your Java source file and the output file containing (your properly formatted) INSERT statements, when you run pass “animal” to your Java program.