CS342 DDL and Modification Lab

Reference:

Murach, Chapters 7, 8 and 11
MS Reference: https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-table-transact-sql?view=sql-server-ver15
MS Reference: https://docs.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver15

*PLEASE NOTE THAT YOU WILL NEED TO RESEARCH RELATIONAL AND CHECK CONSTRAINTS IN THE MURACH TEXT AND THE MICROSOFT DOCUMENTATION REFERENCES ABOVE

Goal: In this lab you will work with DDL commands, INSERT, UPDATE and DELETE DML commands and a few simple examples of RESTRICT and PROJECT operators with the SQL SELECT statement. **THIS IS AN INDIVIDUAL ASSIGNMENT. You are expected to do research on your own to complete portions of the assignment. So, please make sure you read Chapter 11 from the text.**

POINTS Awarded: 10

Deliverable: You must provide two scripts. The first will contain all the commands you used to complete the steps in the first part of the exercise below. (You must key in the commands through the query editor in SQL Server Management Studio and not use SSMS' menu system.) The second is an export of the database and table definitions you created which will be generated by SQL Server Management Studio.

For your commands (not those generated by SSMS) everything you do needs to be placed into a single script file. Keep in mind, that due to the software on the workstations, you will lose all your work when the server you are using is rebooted. So, make sure you save your work in the form of a script (a flat text file) if you take a break.

The Problem: Let's keep track of some sheep. First, review the relation definitions and business rules below. Note, the table name is outside the parenthesis, the column names are listed with comma separation, the underlined column names are the primary key, and "FK" indicates the foreign key. Then, follow the instructions.

sheep(IdNumber, SheepName, BreedCategory, Gender, ShepherdId)
FK BreedCategory->breed
FK ShepherdId->shepherd

breed(BreedCategory, BreedDescription)

sheepShots(IdNumber, ShotType, ShotDate, InjectionType)
FK IdNumber->sheep
FK ShotType->shotList
FK InjectionType->injectionList

shotList(ShotType, shotDescription, dayCycle)

injectionList(InjectionType, injectionDescription)

shepherd(ShepherdId , LastName, FirstName, ShepherdCertification)

You can assume that the table design does not need to be expanded or revised for this exercise. The focus of this exercise is mostly on the physical implementation of new database objects.

Here are some business rules to apply when creating tables.

- Sheep are added to a pool of livestock only if their breed and gender are known.
- Sheep must be assigned a unique ID number which will be placed on a tag before being added to the flock. The ID numbers will be an ongoing series of numbers – a counter that is incremented with each new sheep. (Hint, research the use of the IDENTITY in the text and at the Microsoft site.)
- Sheep must be assigned to one (and only one) shepherd. A shepherd can watch over zero or more.
- Shepherds are classified as having or not having certification.
- Shepherds cannot be added if certification status is unknown.

Here are the shot types you can use: Parvo, CDT, Pasteurella. Here are three injection types you should include: Oral Injection, Subcutaneous, IV.

Follow the steps below and create a single script that will have all the supporting commands in the order created. **You are to "create" the commands manually. Except for the query editor window, do not use SQL Server Management Studio tools unless told to do so (i.e. use only SQL Commands to create objects).** Each SQL statement should end with a semi-colon. Even though this is in a single script, it is highly recommended that after each SQL statement that CREATES or ALTERS an object you enter a "GO" statement on a line immediately beneath it. Generally, SELECT, INSERT and DELETE statements do not need a GO after each one.:

1. Start SQL Server Management Studio (SSMS), and after you login open two new query windows. You will use one window to form and run the commands needed for the steps below. As you complete each step, copy and paste it to the other query window. That second query window is your ongoing list of the commands you used. Again, it may be best to ensure each step ends with a "GO".
2. Create a database named "sheep"
3. Change your database context to "sheep"
4. Create a schema named "ActiveHerd"
5. Unless specified otherwise, create all the database objects using the ActiveHerd schema.
6. Use the appropriate CREATE TABLE statements to set up the relations/tables defined above. Each table must be in the ActiveHerd schema. Use the data types you think are best. As you create each table, specify the PRIMARY KEY constraint and NULL vs NOT NULL. Do not specify other constraints in this step.
7. Except for the use of a primary key, use ALTER TABLE statements to add the appropriate constraints to support appropriate business rules and FOREIGN KEY constraints.
8. Research the SQL INSERT statement and, except for sheepShots, populate all the tables with at least three entries each. (Note: string literals require a single quote.) Make sure one entry in injectionList reflects "Oral Injection" in the description.
9. Using a combination of the INSERT command with a sub-select, insert into sheepShots so that it will show every shot is given to every sheep on the current date for the injection type with the description of "Oral Injection". For example, if there are five shot types, each sheep ID from the sheep table will show up five times in sheepShots.
10. Modify the date of all the sheepShots rows so that the date is set to February 3, 2023.

11. This step is to verify the data you have in the table. Write a SELECT statement that will list the contents of each table. For instance, to list all the data for the sheep table, you would enter:

    ```
    SELECT * FROM ActiveHerd.sheep
    ```

12. Now research the use of the DELETE command and use it to delete all the sheep. Note if the DELETE command works or does not work. If it worked, note how many rows were deleted. If it didn't work, explain why with a comment? (remember that explanations and other responses need to be placed in your script as comments).
13. If the DELETE command did not work, add to the script the SQL commands that will eventually let you get rid of the sheep.
14. Now, we've decided to fire all the shepherds. Enter the SQL command to drop their records.
15. **At this point, we will use SQL Server Management Studio (SSMS) menus to generate a script of your database structures and constraints. This will create a separate file that you are to hand in. Do the following:**
    a. **Right click on the database name "sheep" in the SSM object explorer.**
    b. **Select the following menu path: Tasks->Generate Scripts**
    c. **You should see an Introduction message. Click on Next**
    d. **In the next prompt, select the radio button labeled "Script entire database and all database objects"**
    e. **Click on Next**
    f. **Select "Open in new query window."**
    g. **Click on Next**
    h. **Note the "Review your selections" list**
    i. **Click on Next. You should see the progress list with green check marksd.**
    j. **Click on "Finish>>"**
    k. **The script will be placed in a new query window. Take time to compare that script to what you did.**
    l. **Save that generated query script as AUTO-GENERATED.sql. You will need to submit it to Moodle in addition to the script you created with all the commands.**
16. After all this work, now research the DROP table command and get rid of all the tables in the appropriate order.
17. Finally, research the DROP database command and get rid of the database.

As mentioned, have all the commands you used (except for those generated by SSMS) placed in a single script named CREATESCRIPT.sql. The commands must be in the same order as the instructions they support. You are encouraged to have someone test your script. Note: writing the SQL commands is an individual assignment.

NOW submit your work in Moodle.

THANKS!