

# Assignment 3

## CIS 4301 - Spring 2009

### DB Programming and SQL

You can choose to complete this assignment together with a classmate. No more than two people can work together.

No need to submit a hardcopy of this assignment. Email submission is encouraged. Make sure that your email subject is the following:

#### ***CIS4301 - Assignment 3***

Also, make sure that the first two lines in your message are the names of the members of the team that worked together on this assignment.

**DUE Date: Thursday March 26, 5:10pm.** Your email submission must have been sent by that time. Late submissions will suffer a 15% penalty for every hour (or fraction) after the deadline. You will receive a confirmation message once your submission is received.

Use the following relational schema to complete this assignment. The schema below represents the ER design given in Assignment 2.

Key: PrimaryKey *ForeignKey*

**Donors**(donorID, taxID)

**Individuals**(donorID, lastName, firstName, notes, *companyDonorID*, login, dateEntered)

**Companies**(donorID, name, *primaryContactDonorID*, login, dateEntered)

**Addresses**(donorID, address, zip, login, dateEntered)

**Phones**(donorID, number, login, dateEntered)

**Cars**(VIN, make, model, year, color, mileage, seating, login, dateEntered)

**Maintenance**(VIN, date, cost, notes, login, dateEntered)

**Parts**(storeIdx, type, details, condition, VIN, login, dateEntered)

**Donations**(donationID, date, status, towCompany, towCost, towDate, *donorID*, VIN, *buyerID*, saleDate, salePrice, saleConditions, *custID*, giveawayDate, giveawayConditions, login, dateEntered)

**ActivityTypes**(name, description, *previousActivityType*, login, dateEntered)

**Activities**(donationID, activityTypeName, date, notes, login, dateEntered)

**Buyers**(buyerID, lastName, firstName, ssn, address, phone, login, dateEntered)

**Customers**(custID, lastName, firstName, ssn, homeAddress, homePhone, workAddress, workPhone, need, login, dateEntered)

**Modules**(name)

**Users**(login, password)

**Access**(moduleName, login, type)

1. Generate the SQL data definition language commands to create the above schema in an SQL database. Make sure to include ALL primary and foreign key constraints. As deliverable for this part of the assignment, submit a plain text file with all the SQL code.

2. Write an application that implements the *security module* from the schema above. The application should require the user to login to the system and once inside the system it will provide the user with access to the modules. Which of those modules that user has access to depends on the data stored in the database you will create based on the schema above. Note that because you will not implement all the modules (you will only implement the security module), when the user tries to access a module that was not implemented he/she should just get a message indicating that the module is yet to be implemented.

The security module must include:

- a) a login screen, where the user must be authenticated before accessing the system
- b) a password changing screen for all users that wish to change their password
- c) two maintenance screens: one for maintaining Users, and one for maintaining Modules. The maintenance screens allow the user to add, remove, or modify existing Users or Modules. Notice that only users with explicit access to the Security module can access the maintenance screens.

It is recommended that you use one of Oracle, MySQL, PostgreSQL, or Sybase to write your application and that your application be web-based. Your application must be available for grading at any time the week after the assignment is due. If you choose not to write your application using one of the DBMS listed or to implement your application as a desktop (not-web-based) application, then please contact the instructor prior to starting the implementation in order to get approval of your intended implementation architecture.

As deliverables for this part of the assignment you must provide:

- a) The URL where your application can be accessed.
- b) A **PDF or Plain Text** document describing:
  - Implementation architecture you chose (programming language, DBMS, etc).
  - Issues that you found most challenging to overcome while implementing the system.
  - Any suggestions on what would make the DB schema a better schema to work with.