**PROJECT 1**

CS 687, Fall 2015

**Due:**October 21st

**General Information**

    You need to get a linux username and password from the system administrator. The PostgreSQL is installed on the server voyager.cs.uah.edu. If you work on a Windows machine, you need to use an SSH program (e.g. SSH Client on desktop of Windows machines in the labs). For host name, enter *voyager.cs.uah.edu*. For user name, enter your user name for linux machines. It will ask your password, and then enter the password you got from the system administrator for linux machines.

    You may need to use IP address of voyager, if you connect to voyager off-campus. IP address of voyager is ---------------

    If you directly work on a linux system, start a terminal (e.g. xterm) and then type *ssh voyager.* It will ask your password for linux machines, and then enter your password.

    Do not try to use a browser to access the server.

    When you logon to *voyager*, type *psql cs687 username* (you need to use the username for the PostgreSQL). Your user name might be the same as your linux username. Then it will ask your password for the database. Enter the password for the Postgres to access the database (not the linux password). Please contact me for your Postgres password.

    If you change your database password, make sure that it starts with a character.

**Design & Implementation**

Using PostgreSQL interactive terminal (not the web user interface), implement the company database schema given in Figure 3.5 of the textbook. You first need to create corresponding tables and then insert the tuples in Figure 3.6. *Domain*, key, entity, and referential integrity constraints should be preserved.

Using Empedded SQL in C (ecpg), write a program to implement the following:

         Your program should input the SSN of employee, project number, hours for the project, and salary increase for the employee.

         Firstly, list the employee’s name, employee’s department name, and salary.

         Then for each project the employee is working, provide a list of <the project number, the project name, the number of hours the employee is working on the project, the number of employees working on the project, the total number of hours for the project (by all employees)>.

         After providing the current status related with the employee, update the number of hours worked by the employee for the project as follows.

         (Update/Insert/Delete) If the employee is already working on that project, the number of hours will be updated; otherwise a new tuple will be inserted. The program should input the project number, each employee’s SSN and the number of hours from the command line. If the number of hours is 0, the tuple is deleted.

         (Update) If the hours are updated, your program should print an update message (e.g., “The number of hours for employee ‘**John B Smith**’ on project ‘**ProductX**’ is *updated* from **32.5** to **30**.”).

         (Insert) Otherwise an insert message should be printed (e.g., “Employee ‘**John B Smith**’ *started to work* on **5** hours on project ‘**ProductZ**’.”).

         (Delete) Note that if the number of hours is set to 0, it means the corresponding tuple should be deleted if it already exists (e.g.,  “Employee ‘**John B Smith**’ who was working on **5** hours on project ‘**ProductZ**’ stopped working on this project.”). The parameters will be read from the command line. The message must include the employee name, project name, number of hours, and previous number of hours if applicable.

         After this you should again provide the information about the employee and projects the employee is working after this insert/update/delete as done previously.

         Your executable file name is YourLoginCompany.

The following is a sample call for a program.

YourLoginCompany –ssn 123456789 -pno 1 -hours 30

**Testing**

During testing, I will only enter a valid Employee SSN, valid project number, and hours. Your database should be exactly the same as in the textbook when you have submitted your assignment.

**Note**

      Create a schema using the command “CREATE SCHEMA <schema\_name>”. Schema\_name should be your user name. You should create tables or access tables in your schema by placing “<schema\_name>.” in front of your table.

    To compile, type the following:

      ecpg     filename.x

*This command generates a file that has the same header but having extension .c. If there are errors in your code, it may not generate the .c file.*

  cc filename.c –o YourLoginCompany –I/usr/include/postgresql –L/usr/lib –lecpg

*This command compiles your code and generates an executable named as output.*

  ./YourLoginCompany

*This command executes your program. Depending on which shell you use on linux, you may not need ./ at the beginning of the command.*

    To see the definition of a table in psql, type

\d table\_name

The following additional comments also apply:

1. Good programming style must be observed. This includes using meaningful variable names, descriptive comments, and readable formatting.
2. All your filenames must start with your login name. The main program should be your login + Company (i.e. YourLoginCompany.x). For all other files, your file names should start with your login name.
3. If you submit multiple files, you should either prepare a make file or a script file.
   * If you prepare a make file, you should name it as YourLoginp1.mk
   * If you prepare a script file, you should name it as YourLoginp1Run.sh. Make sure that your file only processes your files. For example, “cc \*.c” is not acceptable, since it is compiling all files.
   * Make sure that output executable file name is generated correctly (i.e., YourLoginCompany)
4. The program source code file must contain a header comment at the beginning of the file. This comment must include the following items:
   * Source code file name, e.g., "YourLoginCompany.x."
   * Student Name
   * Date
5. I will run your programs with *cc* compiler under linux. So make sure that you do not use any additional libraries and your program is runnable under linux.
6. Students should work independently. Each student is responsible for handing in an original program.

**Submitting Materials**

  You should submit the following materials (hard copy) for the grading.

* **Database:** This includes the schema and inserted tuples.

    Provide create table statements

    Provide the inserted tuples by “Select \* from <table\_name>”. No need to provide insert statements.

* **Code:** ecpg code (both online and hard copy)