Linux and PostgreSQL Lab

Preliminaries (just READ)

- You will be logging on to a Linux server. You will have a log-on account on that server, AND an account for the DB server (postgresql). Those accounts have the same name but they are DIFFERENT; however theirpasswords are synchronized. There will be one account per person. The account name is your e-mail handle
- Also, keep in mind you will be working FROM your workstation, but the work will be done on the server. You need to keep straight what are you doing locally and what on the server.
- It is assumed you will be using Moba XTERM (which you can get from http://mobaxterm.mobatek.net/en/ if you don't already have it from our previous lab) . You can also boot with a LiveCD Linux or BSD distro, like Knoppix, or boot on Linux on dual boot machines, or use other ssh client and Xwindows server of your choice.
- A basic familiarity with Linux or Unix is helpful, but not required.
- Each person has a different account, same as your official username for your SPSU e-mail
- To paste into an Xwindows window, press the middle mouse button, or the left and right buttons at the same time.

Logging on

- **Type** `ssh -X yourusername@okaram.spsu.edu` (remember to change yourusername for your username). Notice there is a space after the ssh and another after -X; the X is capitalized, and your username is your spsu email handle.
- You may see a weird message, **type** `yes`.
- You will be prompted for your password. **Type** the password (remember you may have changed your password in a previous lab).
- If you don't see any error messages, you are logged on. Also, the prompt (the stuff displayed every line by the shell) should change, to reflect the fact you're a different user on a different server.

Connecting to PostgreSQL

Bluefish

The DBMS client we will be using, psql, is a text-based client; although it has some basic editing functions, it is probably a whole lot easier to open a text editor, edit your code there, and then copy-paste into the DBMS client; also, that way you can save those commands into a file and reuse them later. On the Linux system, there is a simple text editor called bluefish; to start it, type:

```
bluefish &
```

Notice the & at the end; this tells bash, the shell, to open the program in the background, and to let you keep typing commands while the program runs. When you're running bluefish you can switch to another program by pressing alt-tab (on Windows); you can copy by highlighting, or pressing Ctrl-C. To paste from bluefish into the shell you can click the middle mouse button (or both left and right buttons at the same type) or press Ctrl-Insert.

psql

1. **Type** `psql`
2. You should be now connected to postgreSQL, to a database with the same name as your account. Notice how the prompt changes; psql displays a different set of characters indicating the database you're connected to (normally the same as your username), and ends in a #, while the normal bash prompt ends in a $.
3. Just in case, psql has help for most SQL commands, you can get a list by typing `\h`, and help on a particular command by typing the command after the `\h` . Typing `\?` gives you a list of the special psql commands (for listing tables etc)

Create a table

Create a small table, **type**:

```
CREATE TABLE Person (
    id Integer PRIMARY KEY,
    Name VARCHAR(20),
    Age Integer
);
```

Insert a few values

Use the INSERT sql command to insert a few values. For example, **type**:

```
INSERT INTO Person (id,name,age) values (1,'John',20);
```

Now try repeating the previous statement; what happens ? You should get an error message, since the id is repeated.

Insert at least 3 Persons. To make your life easier, you can use the arrow keys to edit (the up arrow gives you your previous commands, you can move with left/right arrows, delete etc). You can also cut and paste; to paste into an Xwindows program, you click the middle mouse button (or both left and right buttons at the same time).

Basic SELECT

Execute a select statement, to list all rows in the person table:

```
SELECT *
FROM Person;
```

You should see 4 or more rows, if your insert statements worked.

UPDATE, DELETE

Now do at least one UPDATE and one DELETE statements, to change data in the table; use SELECT

to verify the statements worked as intended.

More complicated SELECT

List all the Persons whose name start with J (we use the like operator, which does string comparison, but interprets % as meaning any characters).

```
SELECT * from Person where name like 'J%';
```

Now try doing a SELECT statement to get the id, name and age of people over 21.

Add another table

Now let's add another table, for keeping emails; type:

```
CREATE TABLE PersonEmails (
     PersonId integer REFERENCES Person(id),
     email VARCHAR(50)
);
```

Notice we're not specifying a primary key for this table; we'll later learn how to specify composite primary keys.

Now add a few rows to this table, specifying some emails for some of your people. Use SELECT to verify they're there. Practice UPDATE and DELETE , and make sure it all works as expected.

Get out of psql

Type `\q` to get out of psql. Again, notice how the prompt goes back to the bash prompt, which ends in $

Making your life easier (for later)

- If you're going to be creating lots of tables, you probably want to type the stuff in a file, and then make the interpreter execute it.
- An easy text editor is bluefish; to run it type `bluefish &` (the & tells the shell to start the process in the background). You will see a bunch of messages; ignore them)
- To load a file in psql you can connect to it (by typing psql on the command prompt) and then typing \i; to load a file called abc.sql you'd type `\i abc.sql` . You can also type: `psql < abc.sql` if you just want psql to interpret the file.
- If you try to create a table that already exists, you get an error message. You have to first drop the table with the DROP TABLE abc; command (substitute abc by your table name).

Get out of Linux

To get out of bluefish, you just click on the X button, like every other program (or do File/Quit on the menu); on the bash shell, you type `exit` to get out

And get back

Now log in again (ssh etc), to make sure you can do it. Go back to psql, and verify all your data is there (it should). Keep practicing !