COSC 3P32 Winter 2025
Lab Activity 1 – PostgreSQL

This lab activity is worth 2% of your final grade. To obtain full marks for this lab activity, it must be demonstrated to one of the TAs, Michael or Nujitha in Week 8 (week of 3rd March). You must attend the lab in which you are registered. All labs take place in D205.

The main point of this activity is to get started working with the database tools you will be using for your project. In this lab activity you will become familiar with PostgreSQL and use it to create some simple tables and manipulate data in them. The version of PostgreSQL installed in the labs is 12. Documentation on this version can be obtained at https://www.postgresql.org/docs/12/index.html

You can access PostgreSQL by logging in to your Sandcastle account using (for example) PuTTY. Each student has an individual account set up with user name = their login and password = their student id#. Once you have logged in to Sandcastle, type `psql` at the command prompt. You will be asked for your password. Although you must use your individual account for this lab, see note below concerning setup of group accounts for your project.

The first thing you should do is change your password, as follows:
`ALTER USER <username> WITH PASSWORD 'newpass';`
where `newpass` is your chosen new password.

The tables used for this lab activity are taken from the case study in your textbook. This case study follows through several chapters but for the purposes of this lab activity you need only consider the tables in section 19.9.

1. Use PostgreSQL to create tables for each of the following relations. To do so, first determine the appropriate `CREATE TABLE` statements (not forgetting primary keys and foreign keys). Then type each statement at the prompt, followed by a semi-colon(;). Note that the names of tables and fields *are* case-sensitive.
   See the following for more information on constraints (including primary and foreign keys) in PostgreSQL:
   https://www.postgresql.org/docs/12/ddl-constraints.html

   customer (cid: `INTEGER`, cname: `CHAR(80)`, address: `CHAR(200)`)
   - After creating the table, you can see how it is set up in PostgreSQL by typing (at the prompt):
     `\d customer`
   - You can also see a list of all your tables by typing: `\d`

   book (isbn: `CHAR(13)`, title: `CHAR(80)`, author: `CHAR(80)`, qtyinstock: `INTEGER`, price: `REAL`)
   orders (ordernum: `INTEGER`, cid: `INTEGER` , order_date: `DATE`, cardnum: `CHAR(16)`)
   - cid is a foreign key that references customer

   orderlist (ordernum: `INTEGER`, isbn: `CHAR(13)`, qty: `INTEGER`, ship_date: `DATE`)
   - ordernum is a foreign key that references orders
   - isbn is a foreign key that references book

2. To enter data into a table, type `INSERT INTO` statements at the prompt, followed by a semi-colon (;). Enter the following information into book:
   '0000136006329', 'Operating Systems: Internals and Design Principles', 'Stallings, William', 1, 89.99

   Entering the following data into book should violate the primary key constraint – see what happens when you insert it:
   '0000136006329', 'Some other random book', 'Stahl, Willa', 1, 9.99

   Entering the following data into orders should violate the foreign key constraint (since no customer info has yet been entered) – see what happens when you try to insert it.
   Note that the format of a `DATE` value is `yyyy-mm-dd`. See the following for more information on `DATE`:
   https://www.postgresql.org/docs/12/functions-datetime.html
   123, 101, '2011-05-01', '4505123412344321'

3. At this point you can see how painful it is to enter data one record at a time. You will find it very helpful to populate your tables using a saved file (especially when you later accidentally delete stuff…) Use the electronic file `populatelabA.sql` provided on Brightspace to populate your tables. Make sure this is in the directory you were in when you started PostgreSQL. Run it by typing `\i filename` (for the given file name) at the PostgreSQL prompt. Take note of what happens.

4. At any time you can check on the data currently in a table by writing an SQL query, for example:
   ```
   SELECT * FROM book;
   ```

5. Now see what happens when you delete some information. For example, try deleting the customer with cid=101. See the effect due to foreign key constraints (what foreign key option did you choose?) Think about what would be the difference for different foreign key options (e.g. `NO ACTION` instead of `CASCADE`, or vice-versa).

6. Update some information. For example, try updating the customer with cid=102 by changing the cid and see what happens. Again, think about what would be the difference for different foreign key options for update.

7. Write some queries (inline) – pick your own if you like, or use the following suggestions:
   (a) Information on customers with cname='Luke Skywalker'.
   (b) Names of books ordered by 'John Doe'.
   (c) cid's of customers who have ordered the most books.
   (d) Names of customers who have not ordered any books.

Exit PostgreSQL by typing `\q`
Log off Sandcastle by typing `logout` or `exit`.

Important note about group accounts:
For your group project you need to have a joint account that all members of your group can access. These will be created based on the groups created in Brightspace, after all group allocations have taken place.