Lab #8.2: Set up database connectivity using Python

Python program: Library database management

Overview

Objective

Use Python programming language to connect to a database and manage data in the database.

We will develop a Python program that interfaces with the database in the following labs:

- Lab 7: Create a skeleton program for a database application.
- ***Lab 8****: Set up database connectivity and cursor setup.*
- Lab 9: Create and call functions that query the database.

Outcomes

- Connect to the database with Python database connector functions provided by a module.
- Test database connectivity via a Python program.

Knowledge expected

This is a review of previously acquired knowledge in Python programming.

Submission instructions

- **READ ALL THE WORDS**
- **You must follow ALL submission instructions**: Submission instructions are explained in the "Lab 8 submission details" document, posted on BrightSpace. ANY submission instructions below NOT followed result in a grade of zero.
  Note: This includes extra functionality that has not been requested.

Section A - Set up database connectivity.

In this section we will:

- Set up the Python program file.
- Install the Python database connector.
- Establish and close a database connection in the Python program.
- Test the database connection.
- Establish and close the cursor object.

Before proceeding, verify that both the library database (**`lib_your_networkID`**) and a database role (**`db_your_networkID`** or other) are created in PostgreSQL. If not, create them.

***Exercise #1*: Set up a Python program.**

- Create a Python program file and name it **`dblib_your_networkID.py`**.
  Example: `dbib_smit0001.py`
- Create program header comment.
- Verify that the program is executable.

***Syntax practice #2*: Install the Python database connector.**

Use the following commands to install the Python database connector module:

- `yum install postgresql-devel`
- `yum install python36-devel`
- `yum install python36-psycopg2`

***Syntax practice #3*: Establish and close a database connection in a Python program.**

- Import `sys` and `psycopg2`, one import per line.
- The code section below establishes a database connection. Provide the following information: your database name and the role that you use to access the database.
  - *Best practice*: Set up database and role name as constant variables.

```
try:
    connection=psycopg2.connect(database='name',
user='role')
except psycopg2.DatabaseError:
    print("Error: Connection to database not established.")
    sys.exit(1) # early exit if DB not available
print("Database connection established")
```

- Close the connection. The VERY LAST statement of the program has to be the closing of the database connection:
  `connection.close()`

***Exercise #4*: Test the database connection.**

- Execute the script to test database connectivity.
  - Test successful connectivity.
  - Test unsuccessful connectivity: use a wrong database name.

***Syntax practice #5*: Set up the cursor.**

To enable Python to perform database operations using SQL commands a "cursor" is required.

- Set up the cursor with the function call `cursor = connection.cursor()` *right after* the database connection is opened (after the `try`/`except`).

- Closed the cursor with the function call `cursor.close()` *just before* the database connection is closed in the program file.