**DATABASE FOR BALTIMORE CITY COMMUNITY COLLEGE**

**TUTORIAL / REFERENCE MANUAL**



**DATABASE MANAGEMENT SYSTEMS PROJECT PREPARED BY:**

Dalton Brown | Matt Clark | Jacob Della | Christian Sauls

Towson University

Fall 2020

|**Table of** **Contents**

[| **1 – Introducing the BCCC College Management System** 2](#_Toc57160661)

[| **1.1 – Who should use this application?** 2](#_Toc57160662)

[| **1.2 – Who should use this guide?** 3](#_Toc57160663)

[| **2 – Key Features** 3](#_Toc57160664)

[| **2.1 – Features** 3](#_Toc57160665)

[| **2.2 – Limitations** 3](#_Toc57160666)

[| **2.3 – Design decisions** 3](#_Toc57160667)

[| **3 – Guided Tour / Tutorial Manual** 4](#_Toc57160668)

[| **4 – Functions / Reference Manual** 15](#_Toc57160669)

# | **1 – Introducing the BCCC College Management System**

# | **1.1 – Who should use this application?**

* Staff members and instructors often face difficulties managing their courses, grades and advisors using PeopleSoft because of how slow and clunky it is.
* Today we are here to show you a powerful, modular alternative to PeopleSoft.
* Introducing BCCC CMS or Baltimore City Community College Management System.
* It is designed specifically so that it easy to pickup and use as a drop-in replacement for PeopleSoft.
* Going forward BCCC CMS provides a powerful and scalable solution to manage many facets of an education system.

# | **1.2 – Who should use this guide?**

* Staff members and instructors
* Staff members and instructors in training
* IT Support

# | **2 – Key Features**

# | **2.1 – Features**

* Highly customizable with GUI based on built-in Tkinter window manager library for Python.
* Adding new features and queries is easy to do and the application can easily have additional features added.
* Debugging output via the command line.
* Scrollable Tkinter frame to display results.
* Lightweight and simple to navigate application.

# | **2.2 – Limitations**

* Per user authentication is not implemented in the current release.
* Python MySQL library can be slower than equivalent Java or C++ libraries.
* In current version authentication information stored in executable file.
* Must be launched via command line.
* Python library must be installed prior to using the application.
* Custom query error handling not fully implemented yet.

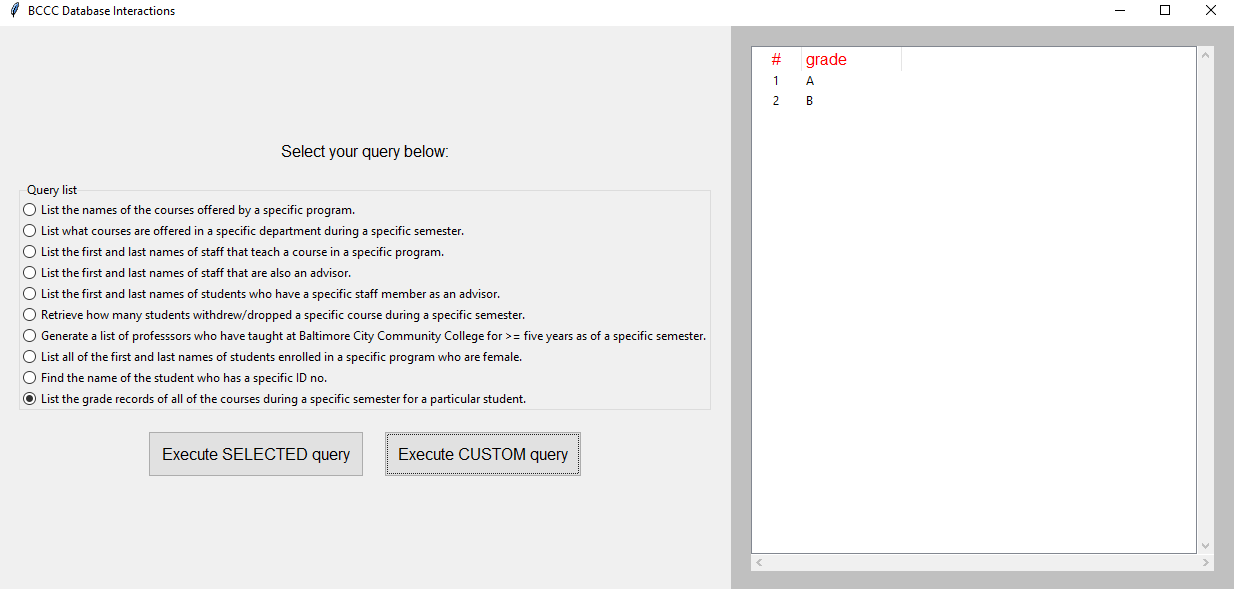
# | **2.3 – Design decisions**

* Database was constructed using the follow specifications:
  + Data Model Requirements for BCCC
    - Allow users to input information for the department, students, staff, courses, programs, and grade records. A few examples of some user requirements that this database will meet is listed as the following:
      * Allow user to update all the above information.
      * Allow user to access specific information for students, staff, courses, etc.
      * Allow user to generate a list of the number of students currently assigned under a specific department.
      * Allow user to generate a list of staff assigned to a specific course.
      * Allow user to generate a list of courses offered by a program during a specific semester.
      * Allow user to retrieve grade records for a specific course.
      * Allow user to retrieve grade records for each student.
      * Allow user to retrieve a list of visitors on a specific day.
      * Allow user to separate information according to identification status (student, staff, or
      * visitor).
      * Allow user to verify all students have an advisor.

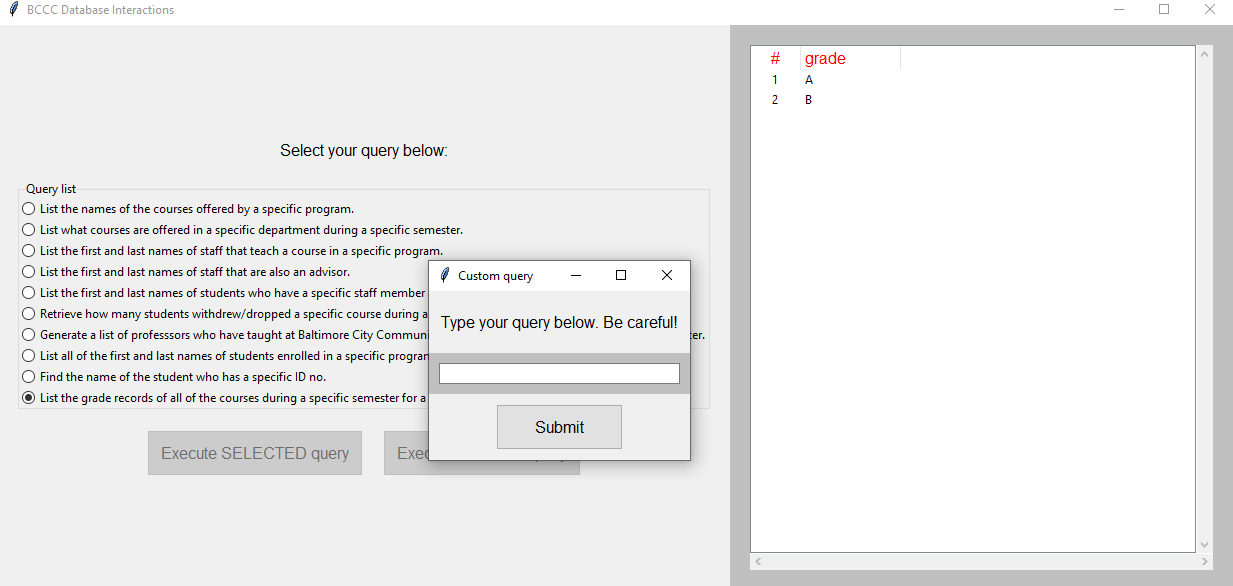
# | **3 – Guided Tour / Tutorial Manual**

* General usage of the program’s numerous functions is shown below.
* The program ships with several built-in queries for the user to choose from.
* If a query is not included the user can still execute their own SQL commands via the CUSTOM query option.

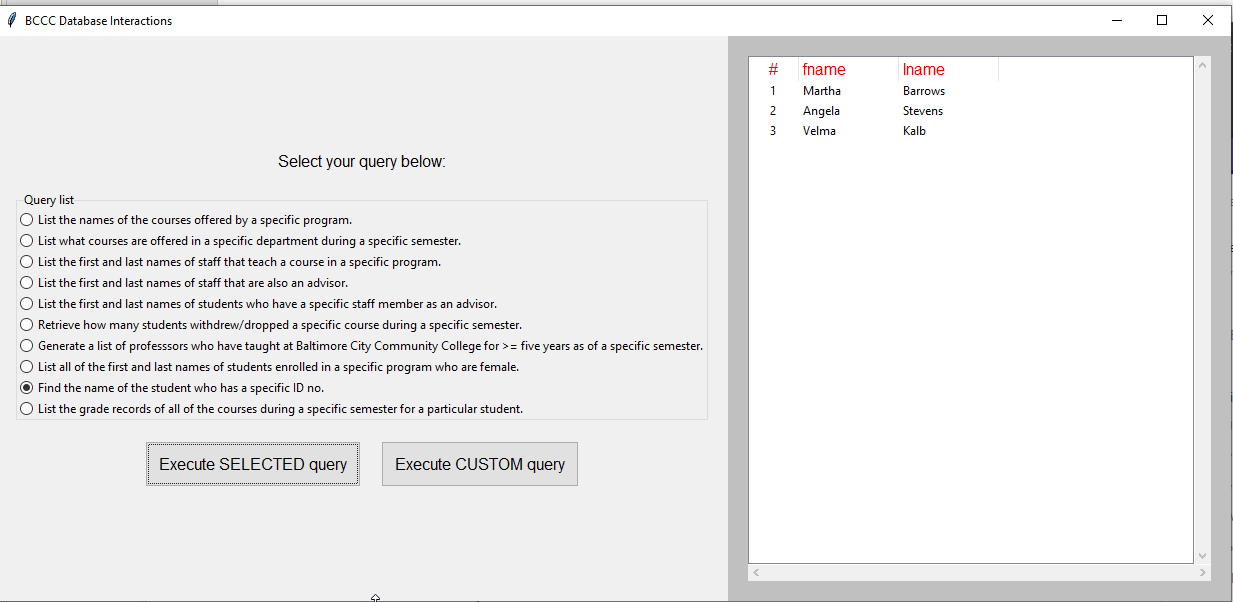
1. Execute CUSTOM query.
   1. We can create our own custom query; user must be careful in this mode.



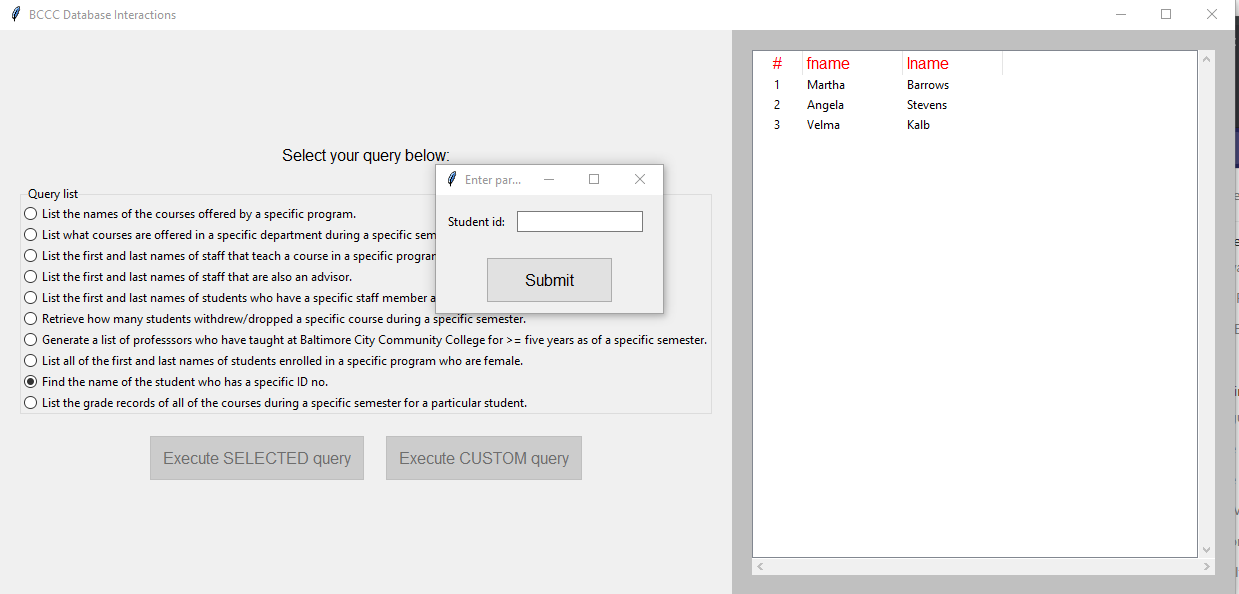
* 1. Enter query and execute.



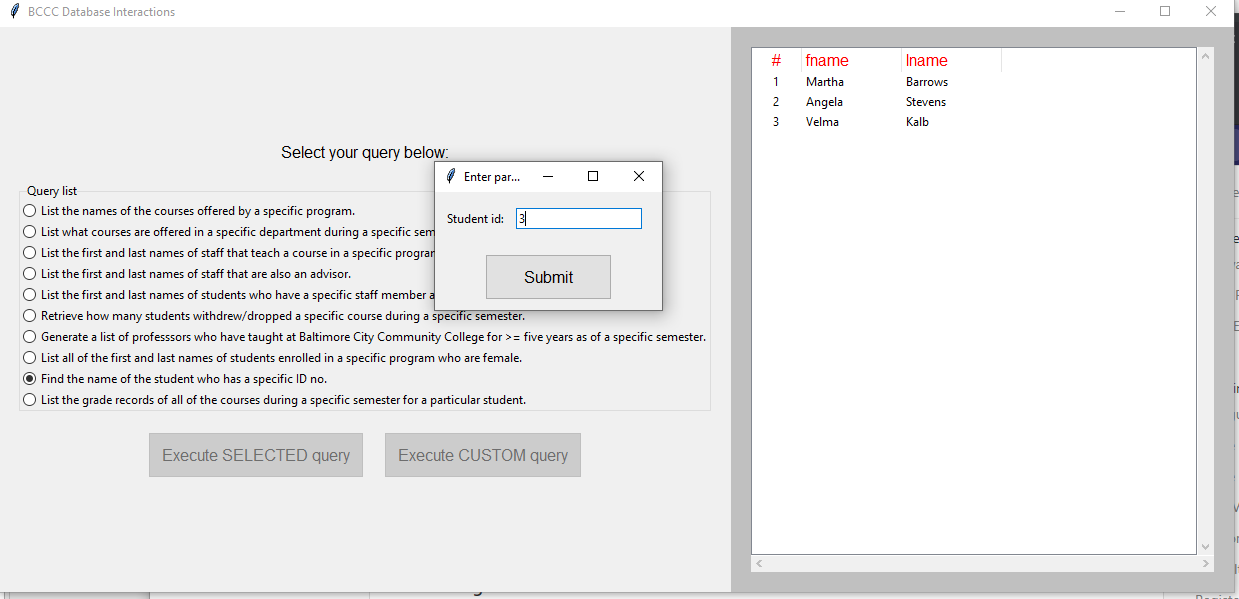
1. Find the name of the student who has a specific ID no.
   1. We select the query we want to use, in this case we want to find the name of the student who has a specific ID no.



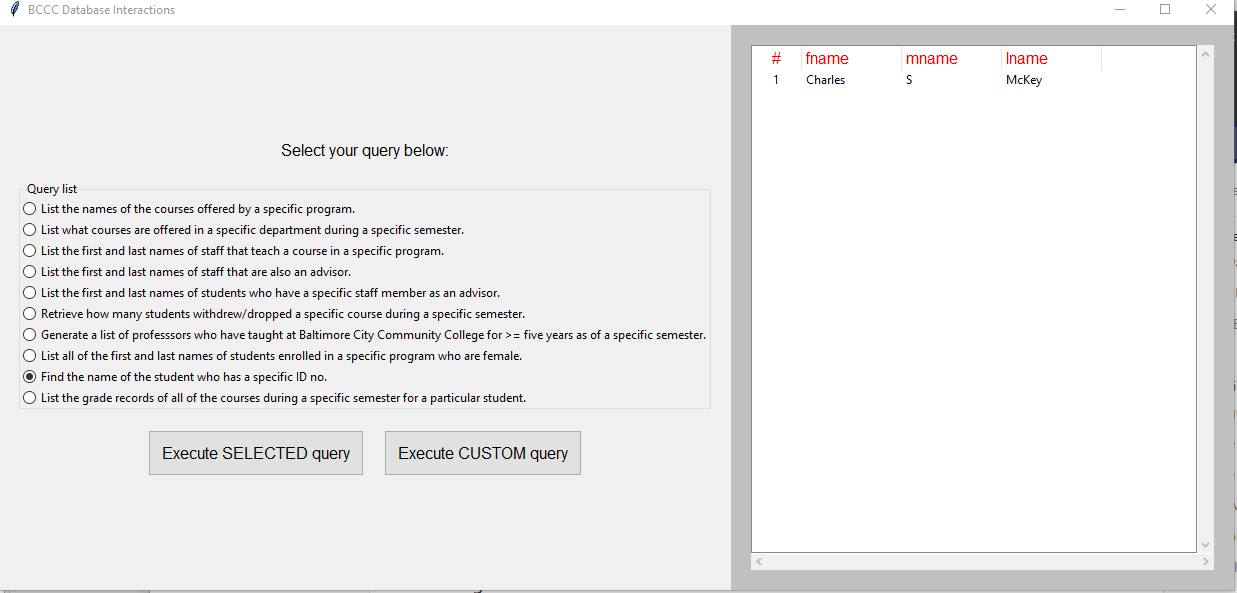
* 1. We hit the “Execute SELECTED query”.



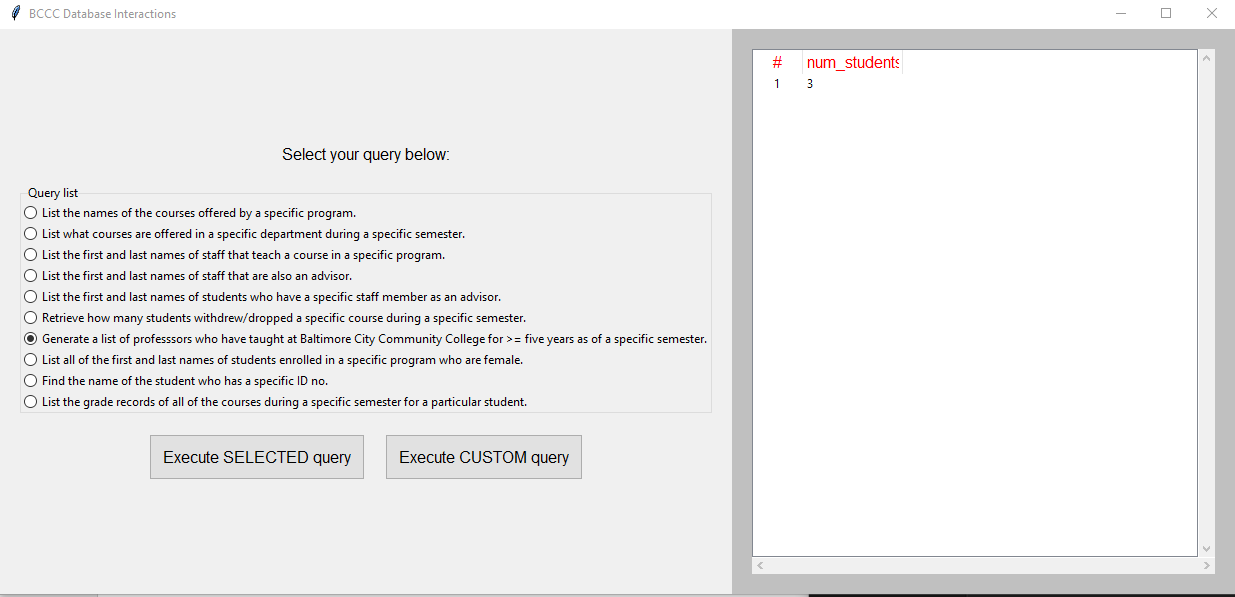
* 1. We type in our parameter we want to search for.



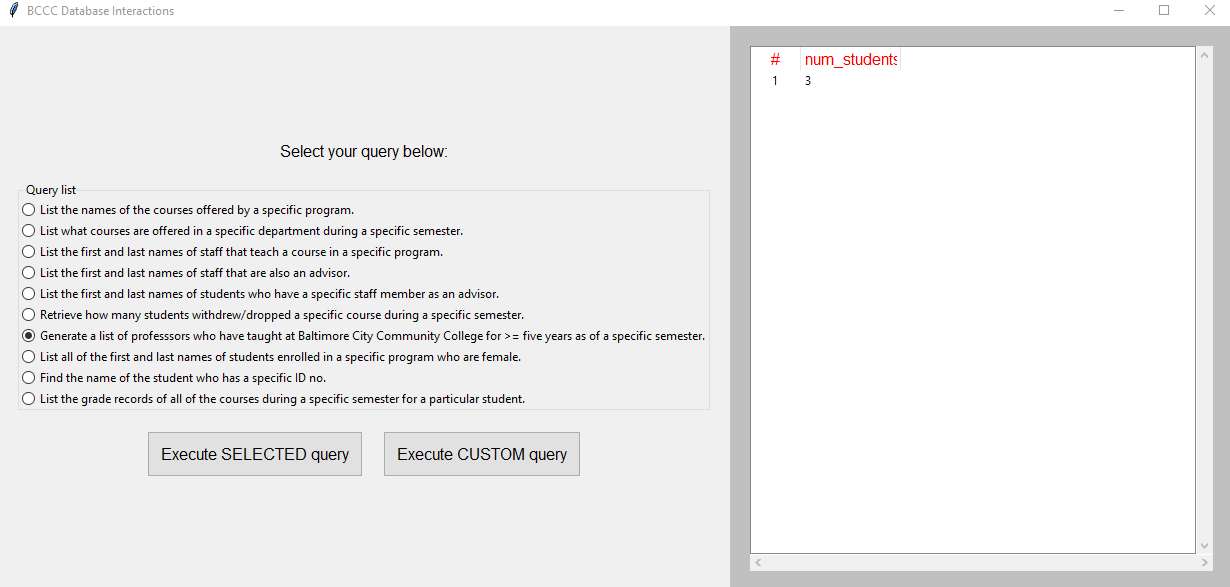
* 1. We hit submit and our results are displayed.



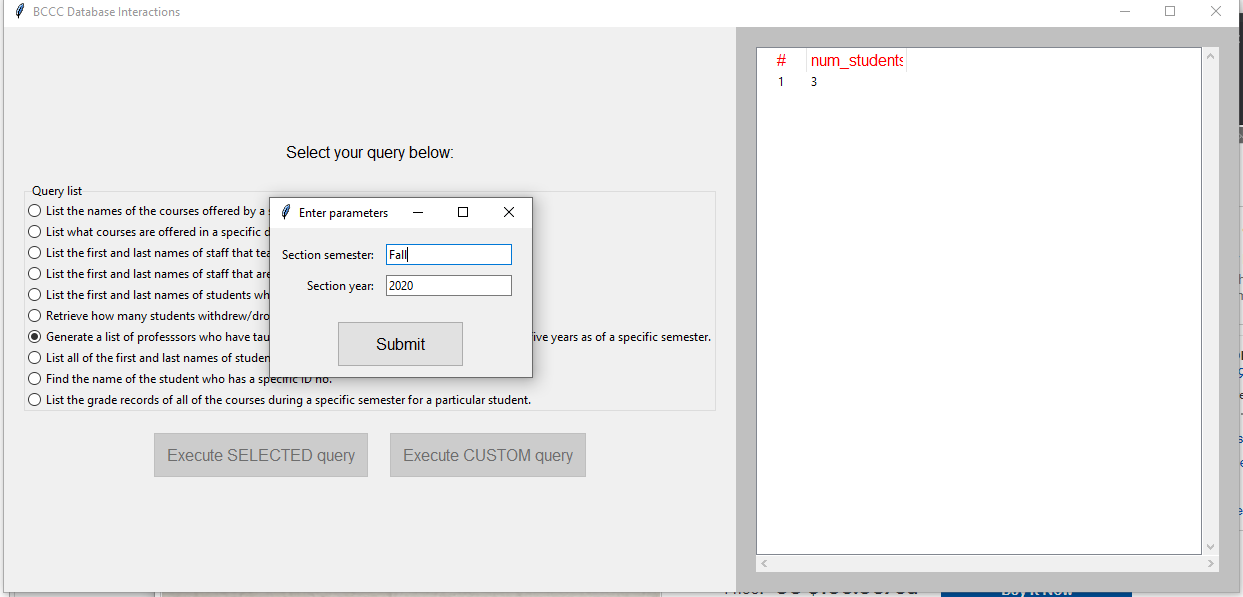
1. Generate a list of professors who have taught at Baltimore City Community College for more than or equal to five years as of a specific semester.
   1. We select the query we want to use, in this case we want to list of professors who have taught at Baltimore City Community College for more than or equal to five years as of a specific semester.



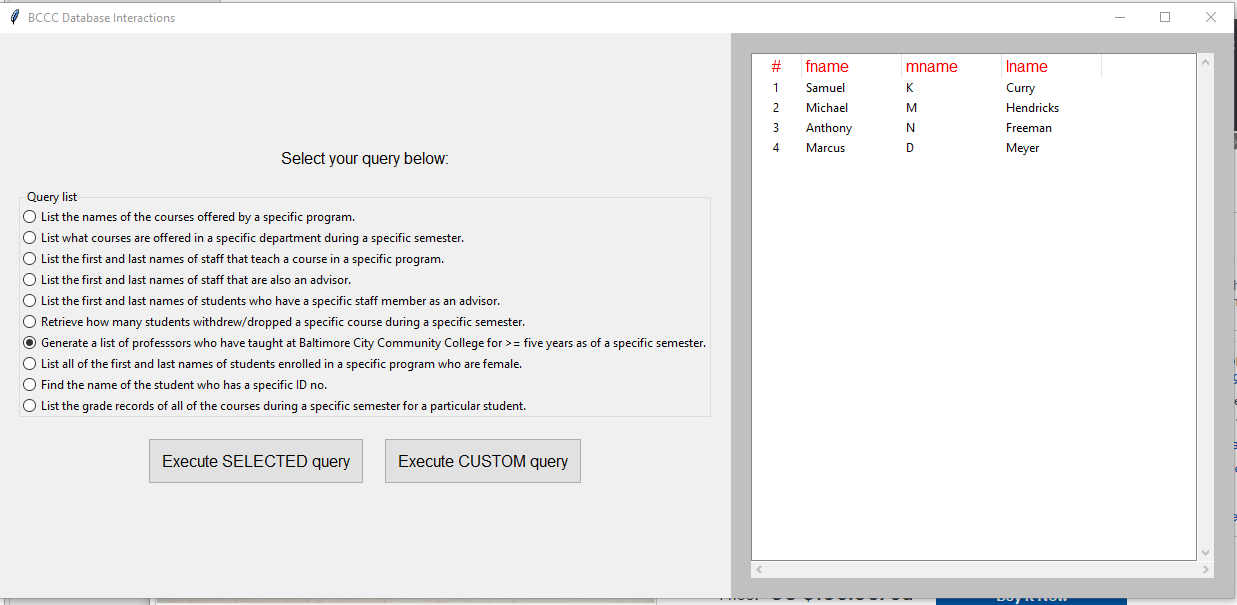
* 1. We hit the “Execute SELECTED query”.



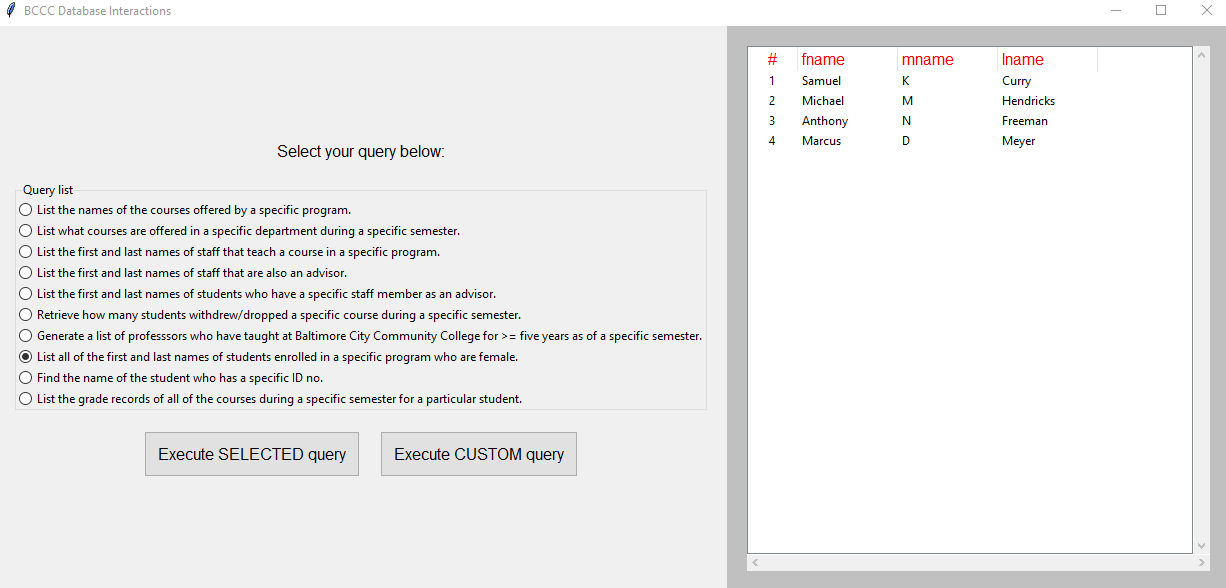
* 1. We type in our parameter we want to search for.



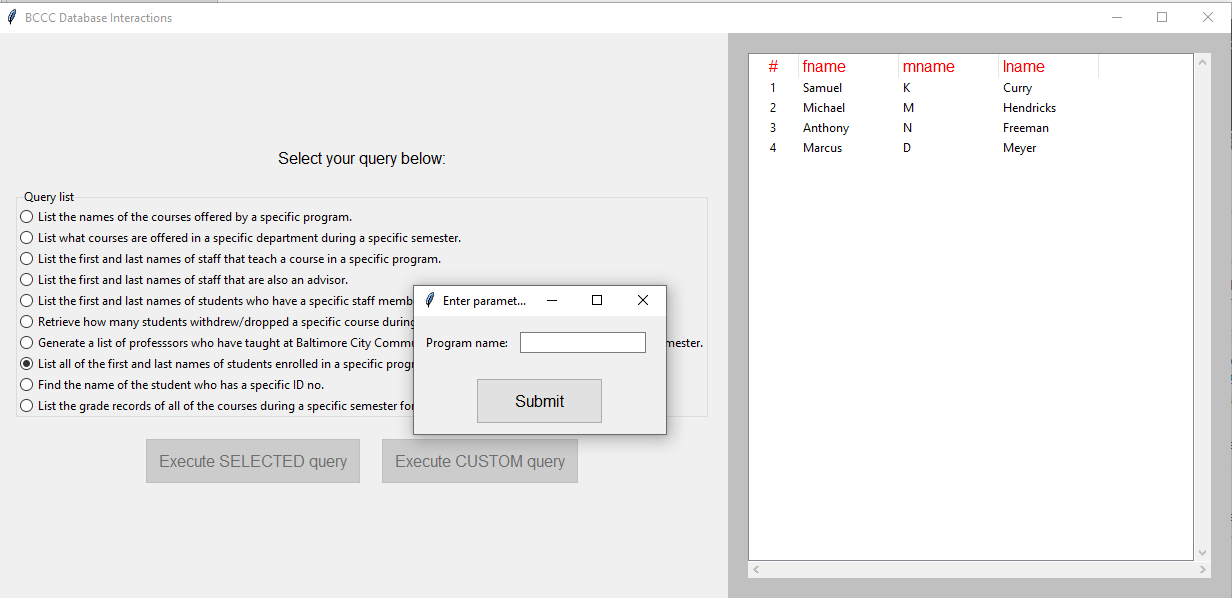
* 1. We hit submit and our results are displayed.



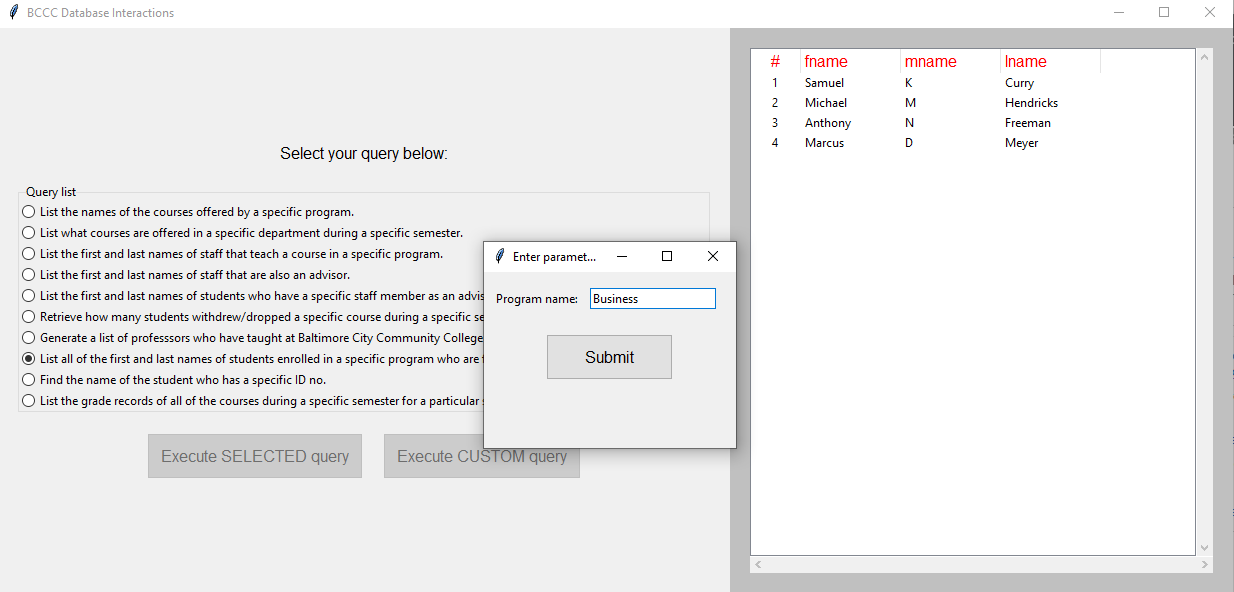
1. List all the first and last names of students enrolled in a specific program who are female.
   1. We select the query we want to use, in this case we want to list of all the first and last names of students enrolled in a specific program who are female.



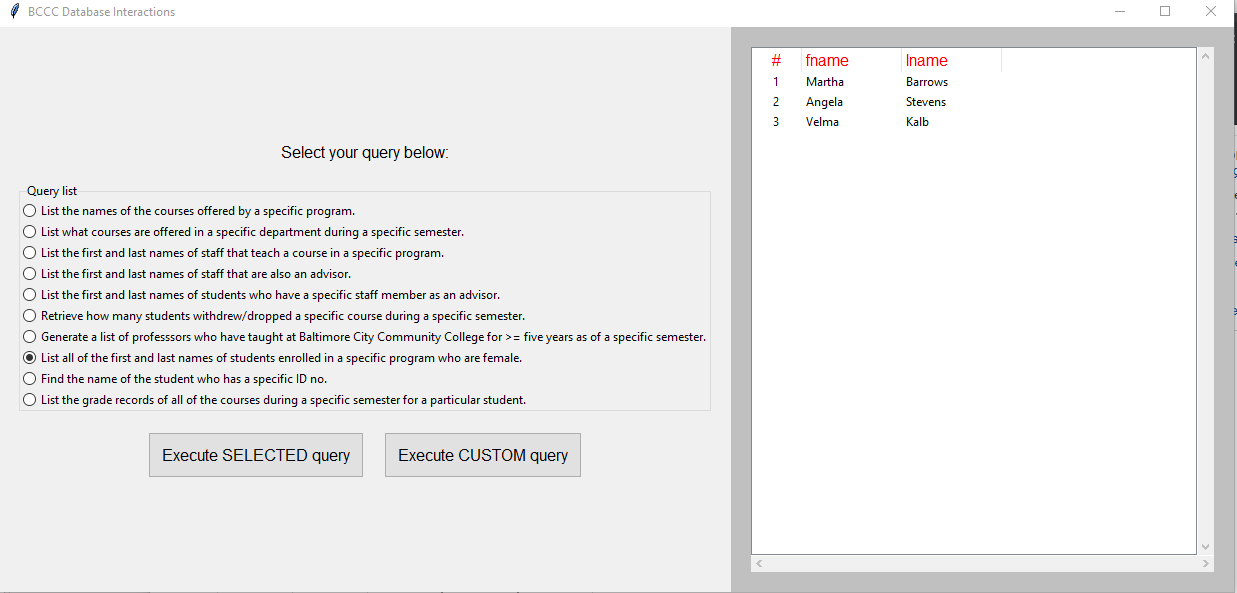
* 1. We hit the “Execute SELECTED query”.



* 1. We type in our parameter we want to search for.



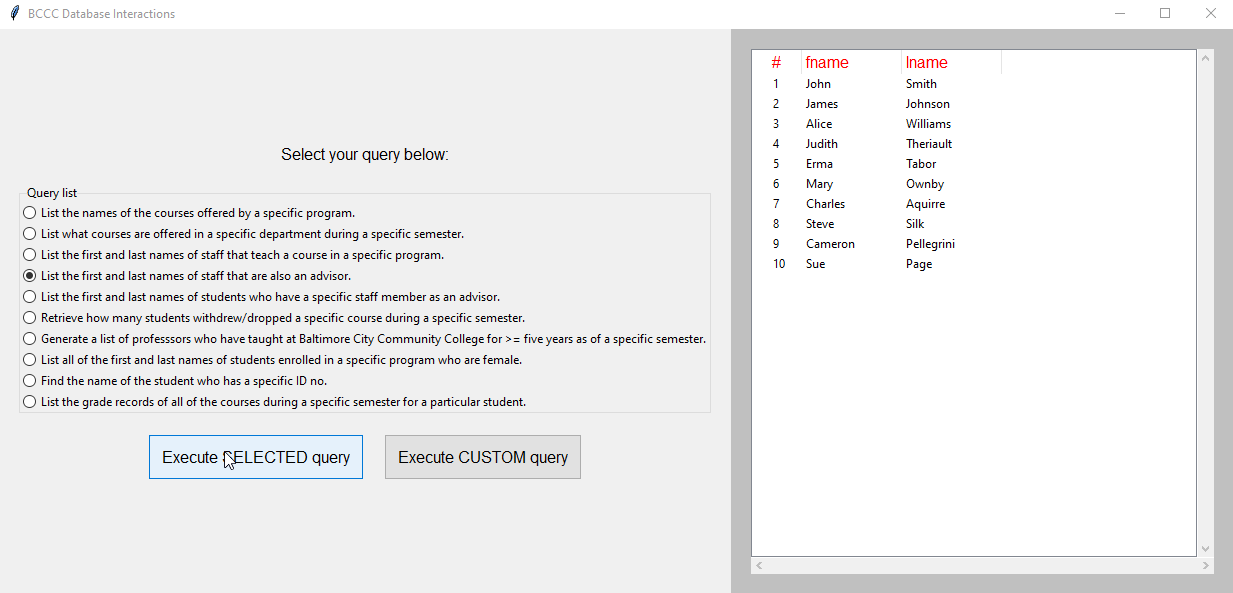
* 1. We hit submit and our results are displayed.



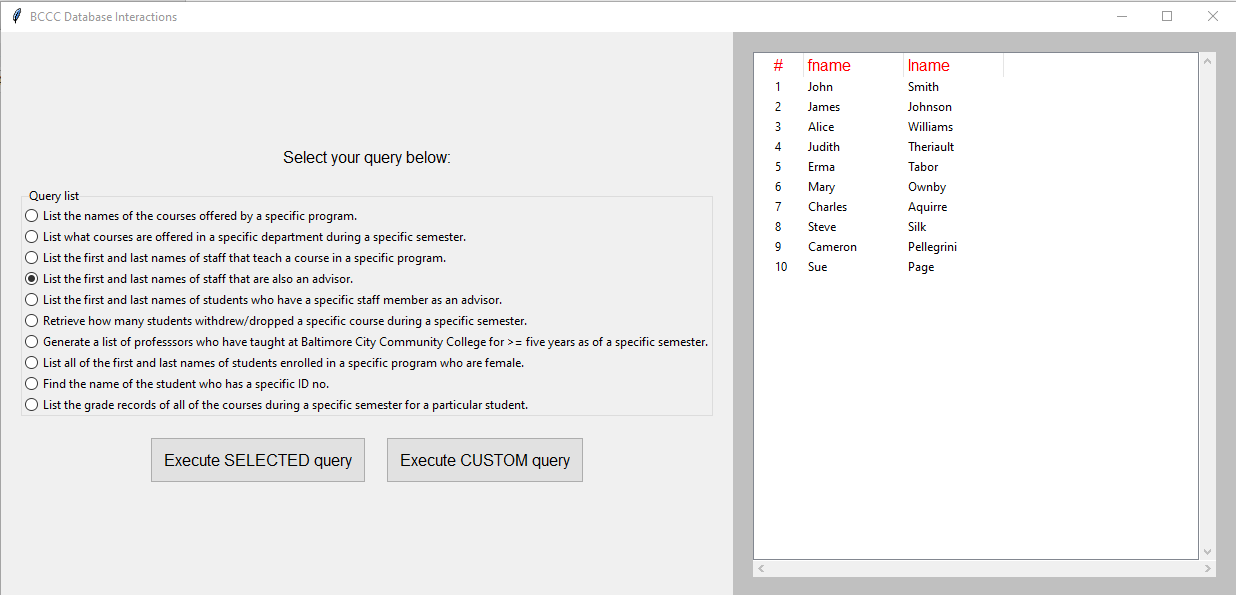
1. List the first and last names of staff that are also an advisor.
   1. We select the query we want to use, in this case we want to list the first and last names of staff that are also an advisor.



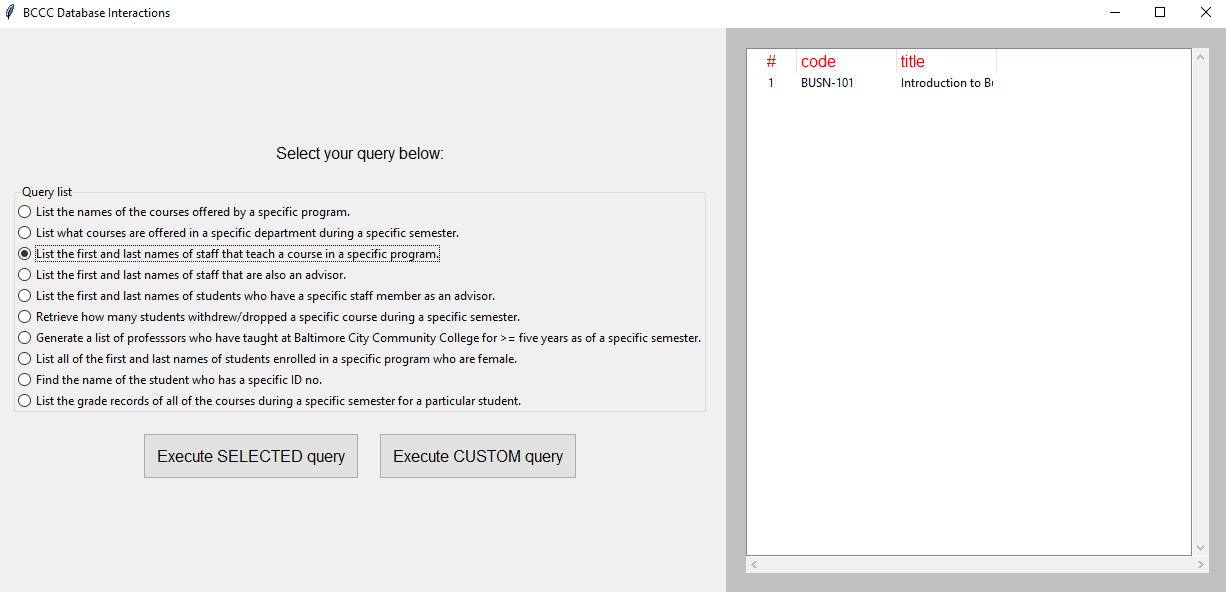
* 1. We hit the “Execute SELECTED query”.

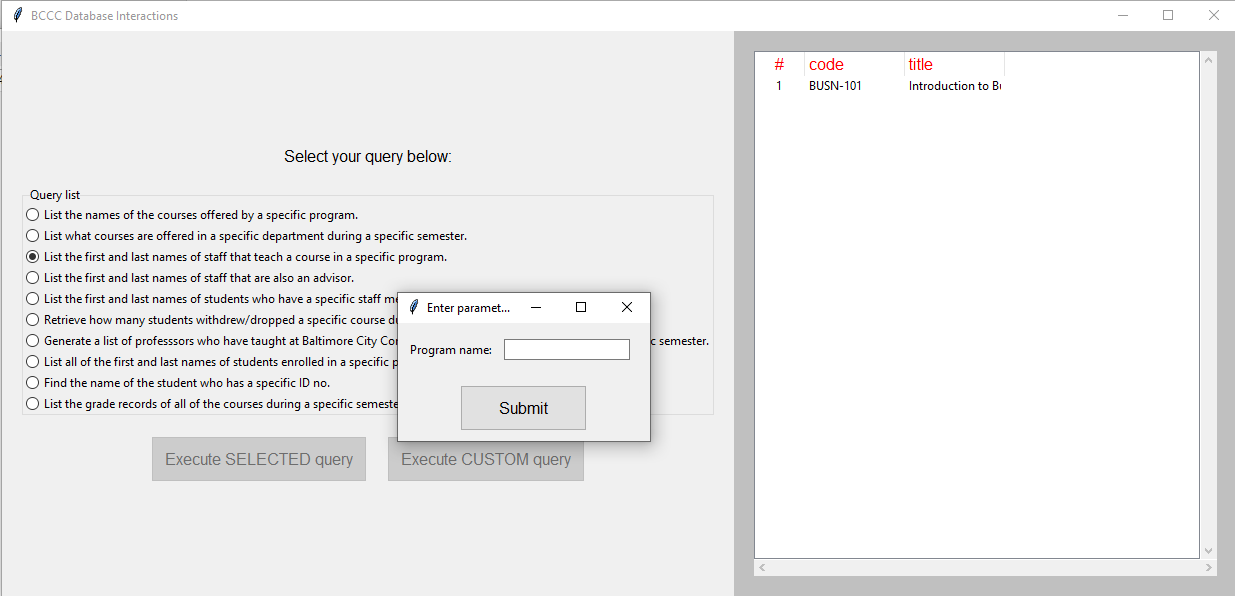


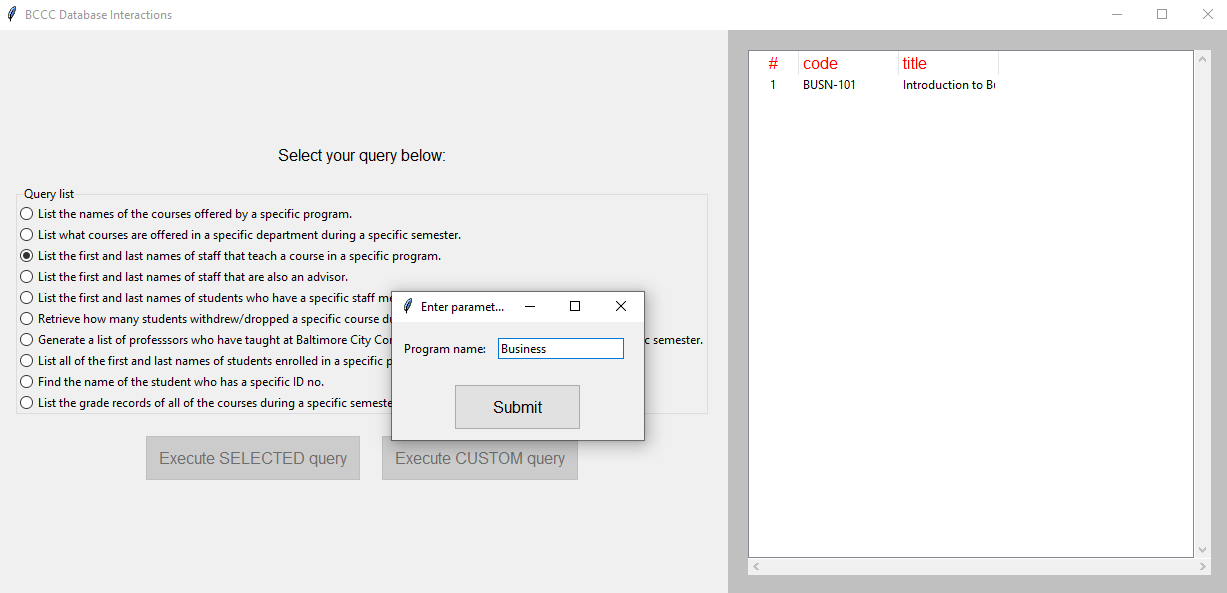
* 1. Our results are displayed.



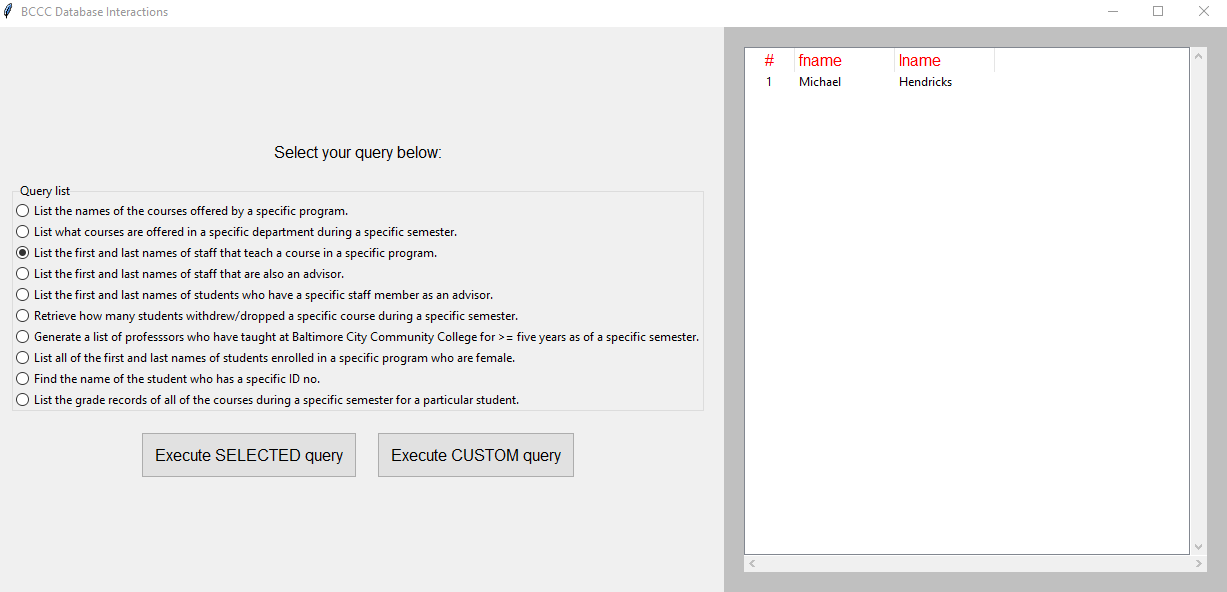
1. List the first and last names of staff that teach a course in a specific program.
   1. We select the query we want to use, in this case we want to list the first and last names of staff that teach a course in a specific program.



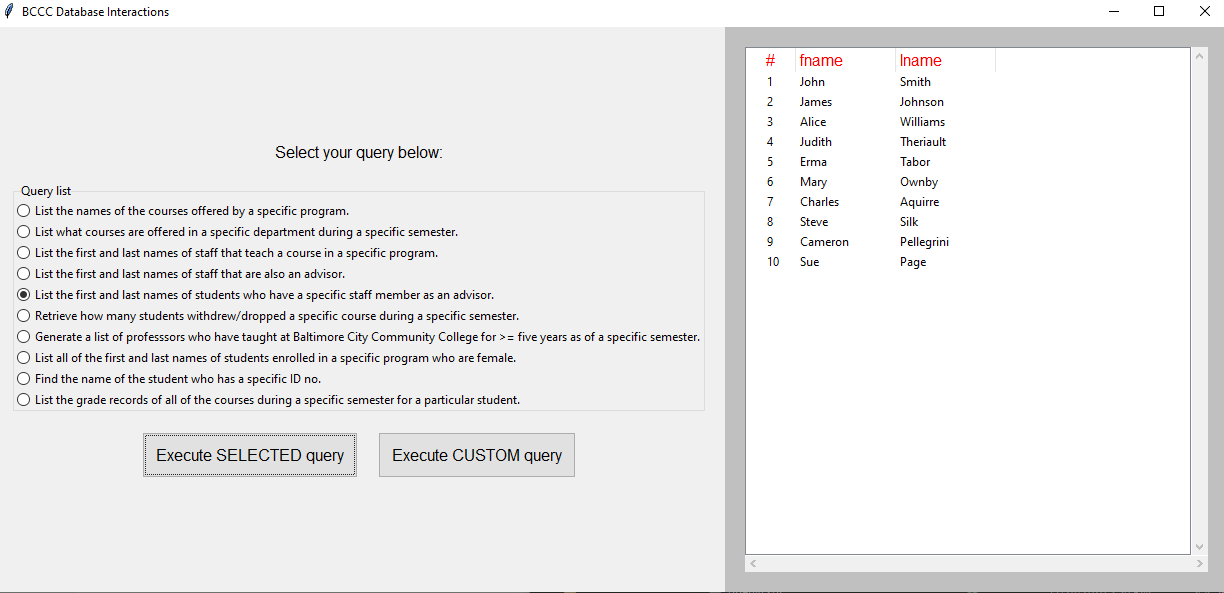
* 1. We hit the “Execute SELECTED query”. 
  2. We type in our parameter we want to search for.



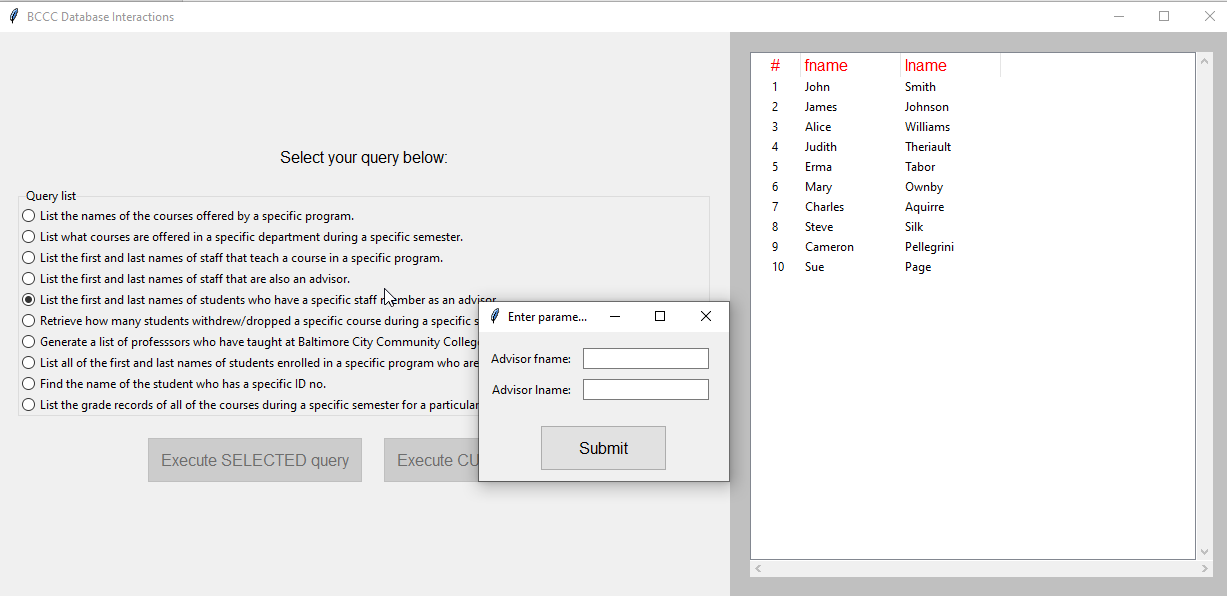
* 1. We hit submit and our results are displayed.



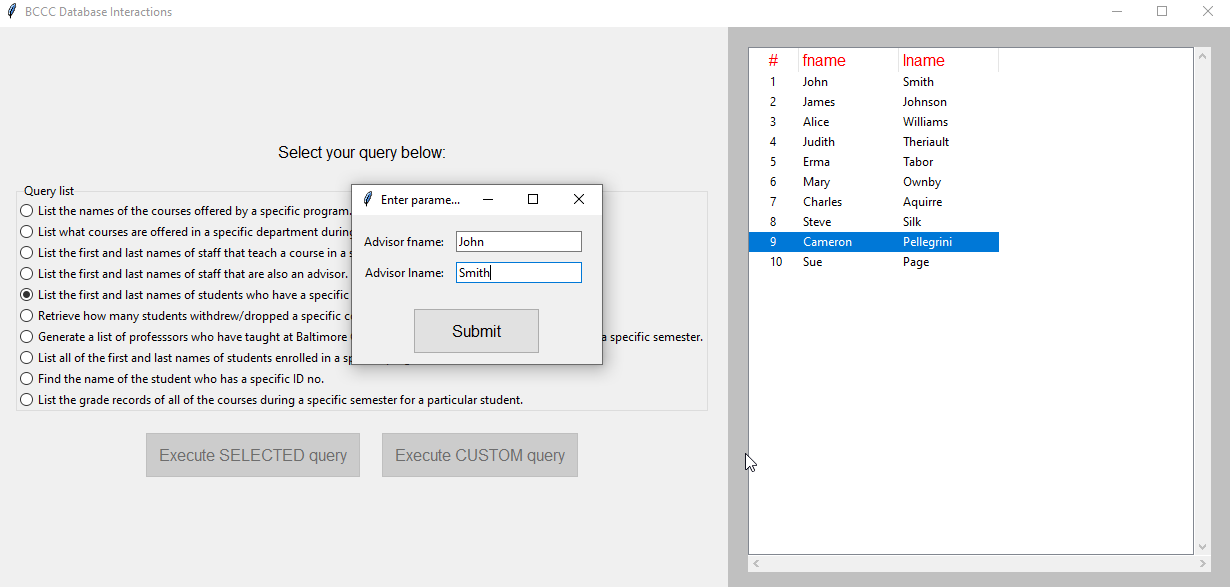
1. List the first and last names of students who has a specific staff member as an advisor.
   1. We select the query we want to use, in this case we want to list the first and last names of students who has a specific staff member as an advisor.



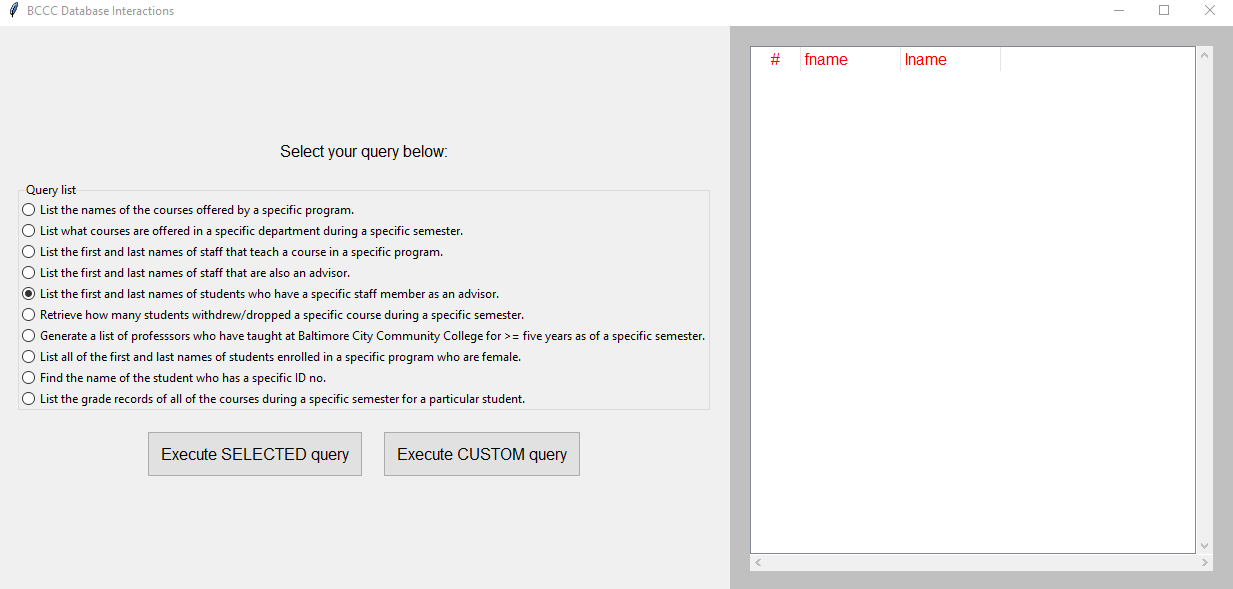
* 1. We hit the “Execute SELECTED query”.



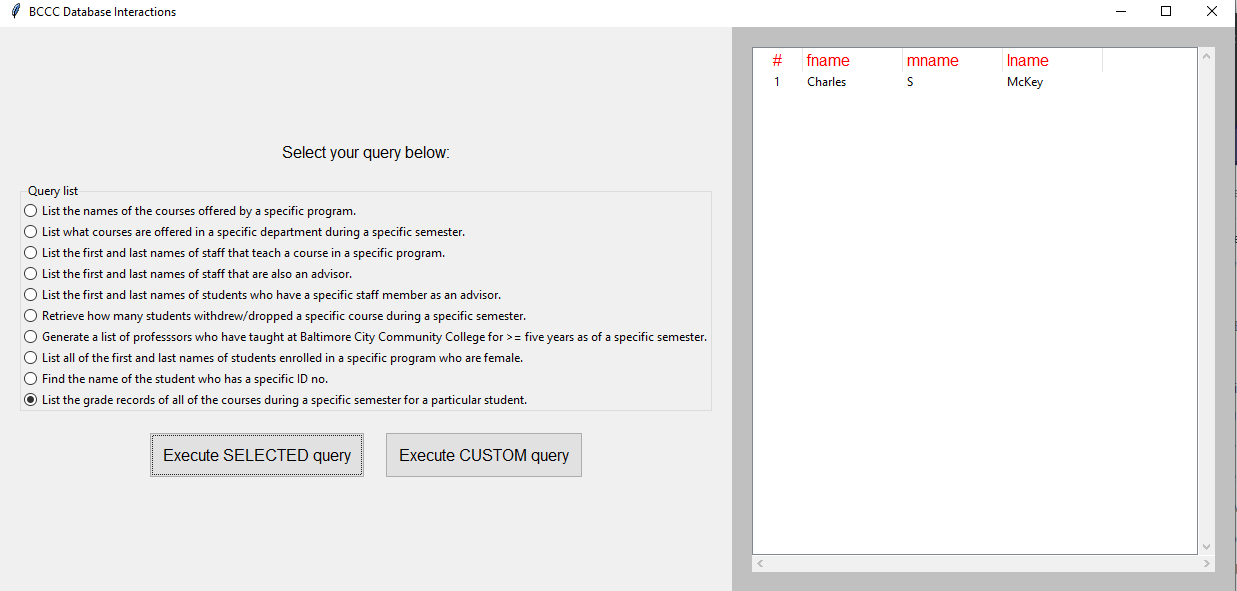
* 1. We hit submit and our results are displayed.



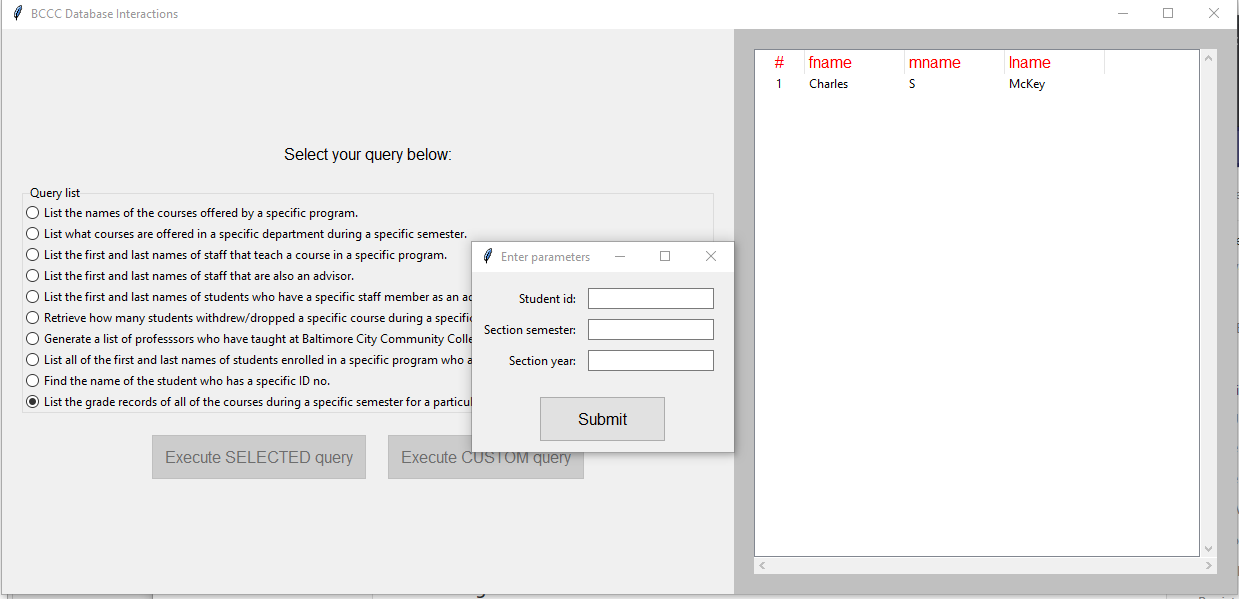
* 1. We hit submit and our results are displayed.



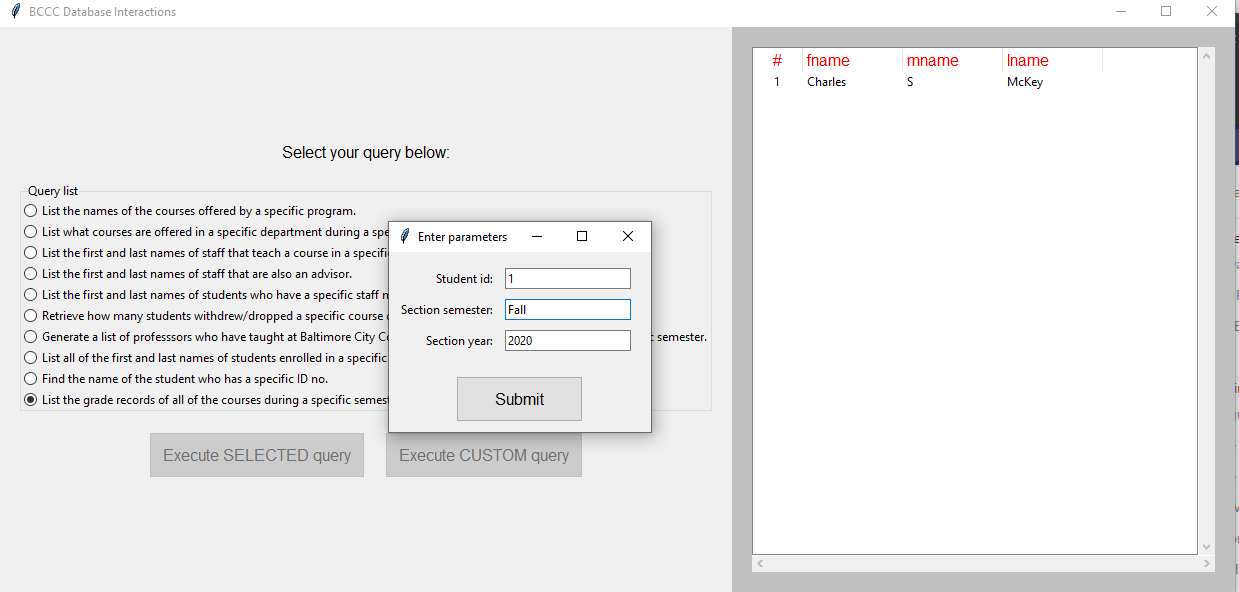
1. List the grade records all the courses during a specific semester for a particular student.
   1. We select the query we want to use, in this case we want to list the grade records all the courses during a specific semester for a particular student.



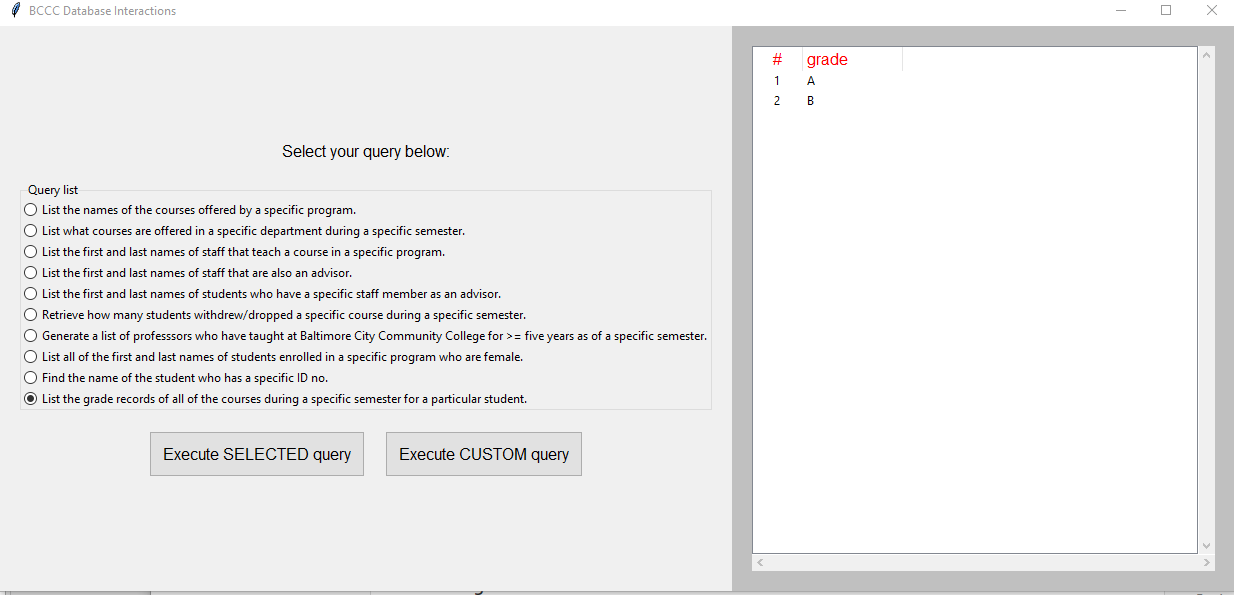
* 1. We hit the “Execute SELECTED query”.

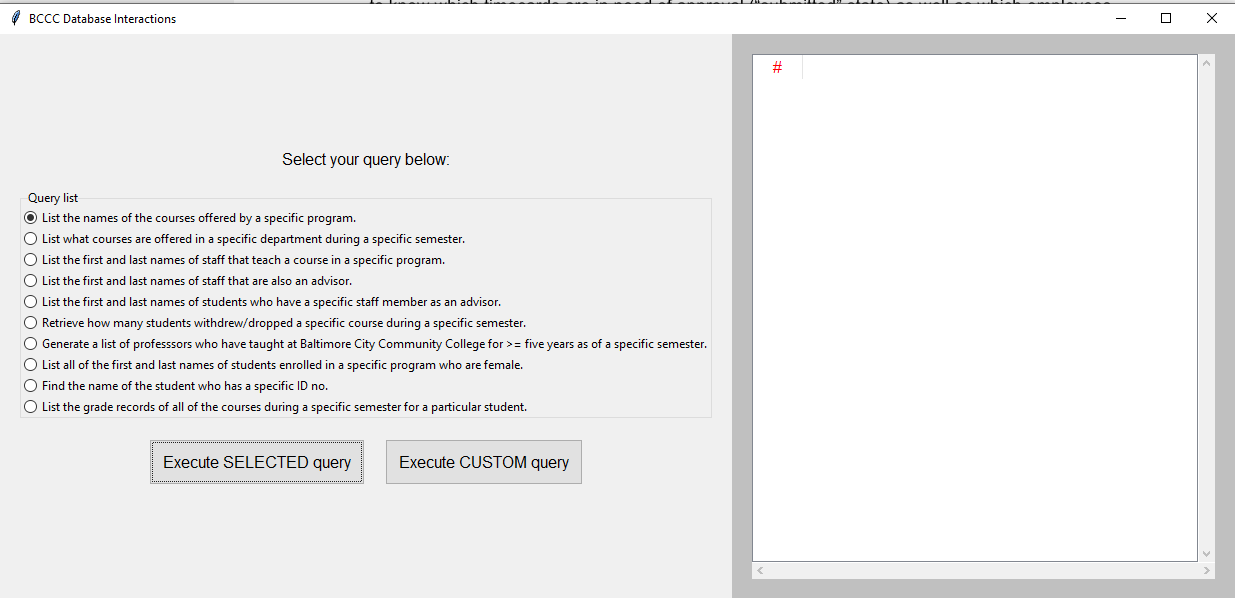


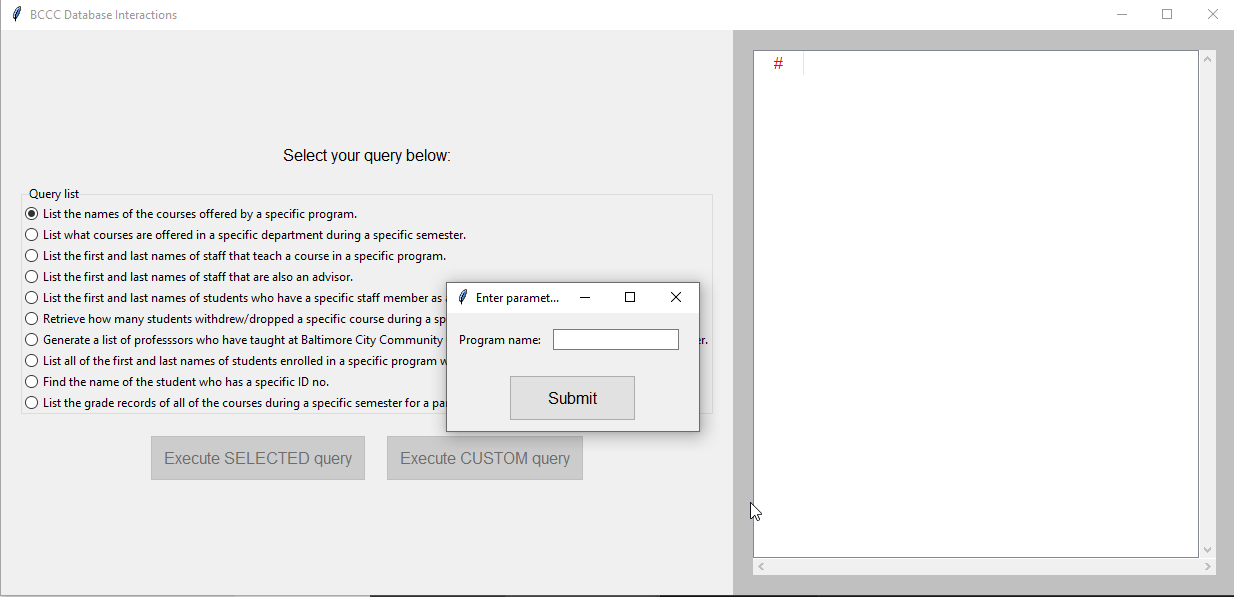
* 1. We type in our parameter we want to search for.



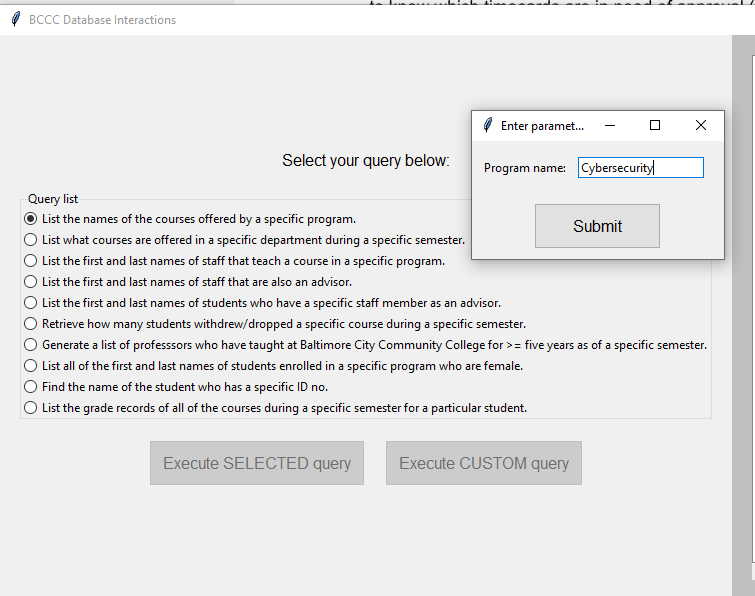
* 1. We type in our parameter we want to search for.



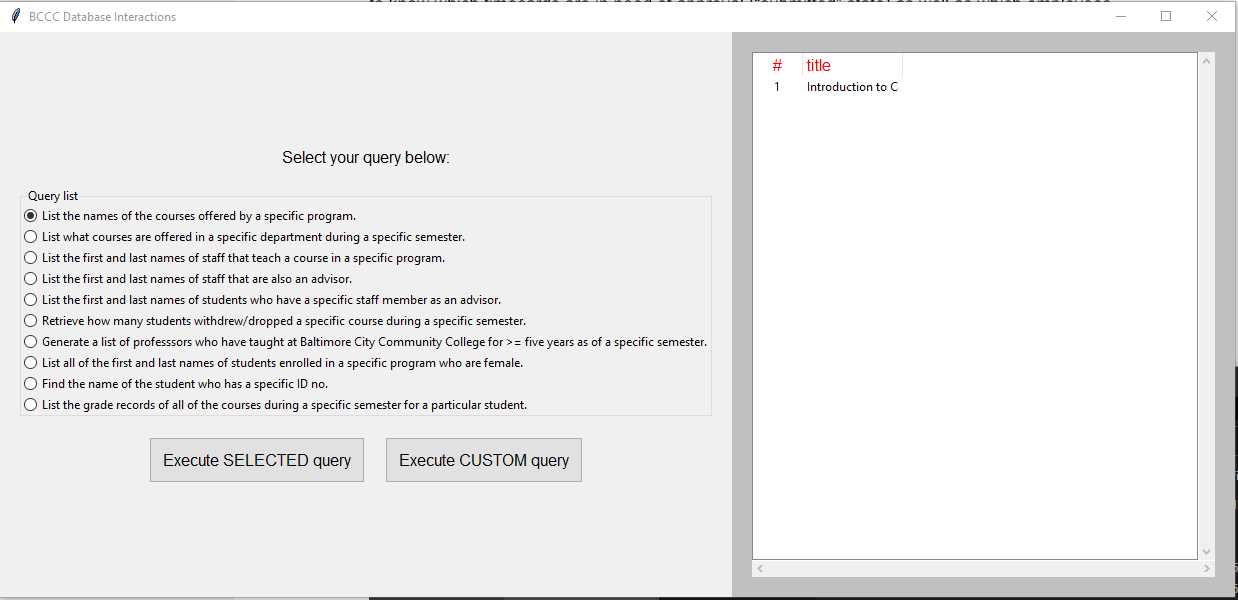
1. List the names of the courses offered by a specific program.
   1. We select the query we want to use, in in this case we want to list the names of the courses offered by a specific program. 
   2. We hit the “Execute SELECTED query”.



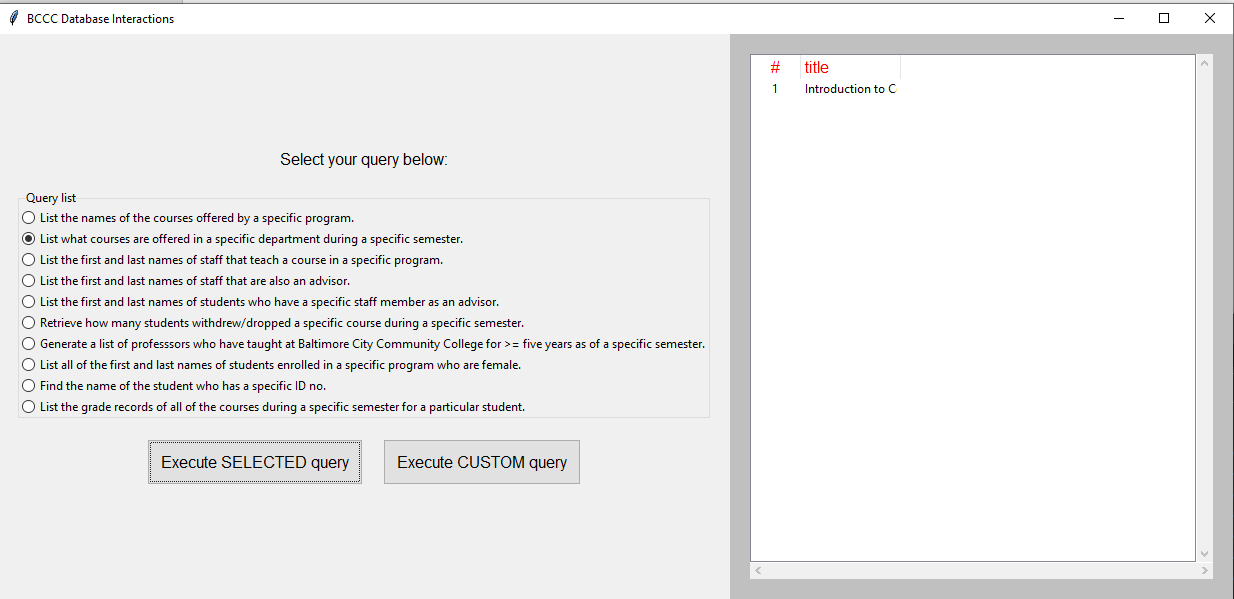
* 1. We type in our parameter we want to search for.



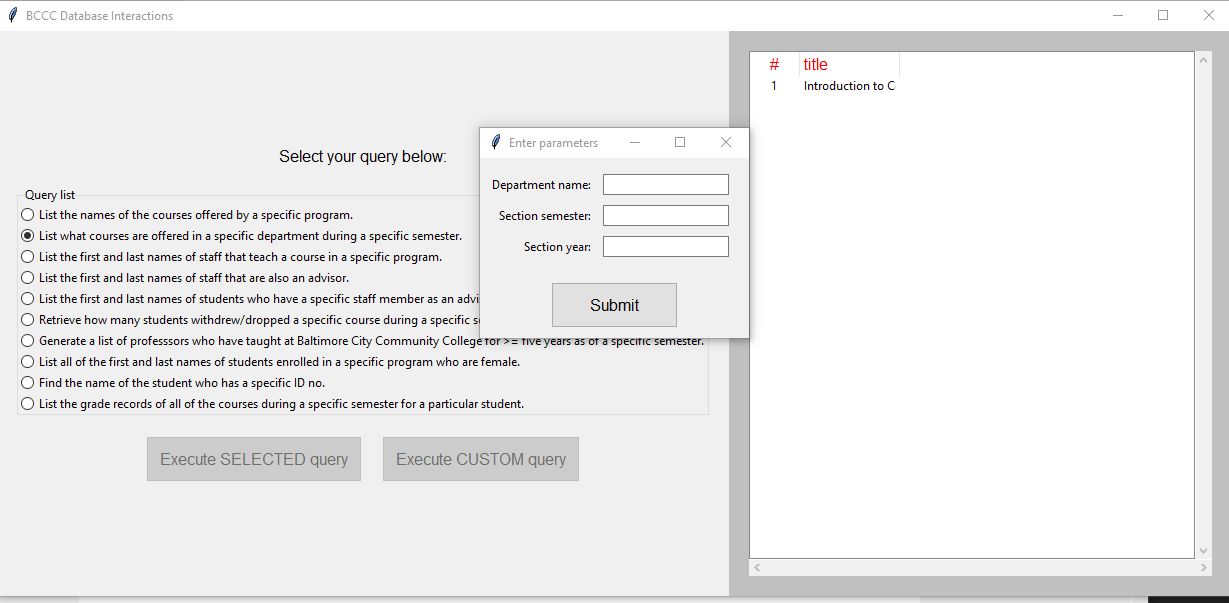
* 1. We hit submit and our results are displayed.



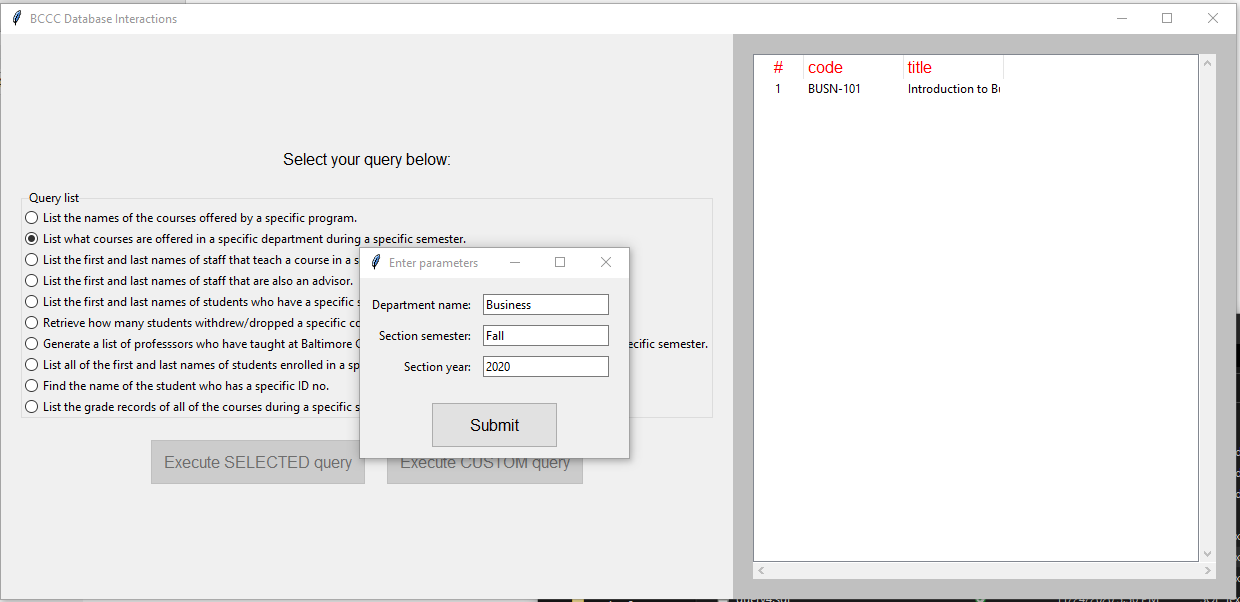
1. List what courses are offered in a specific department during a specific semester.
   1. We select the query we want to use, in this case we want to list what courses are offered in a specific department during a specific semester.



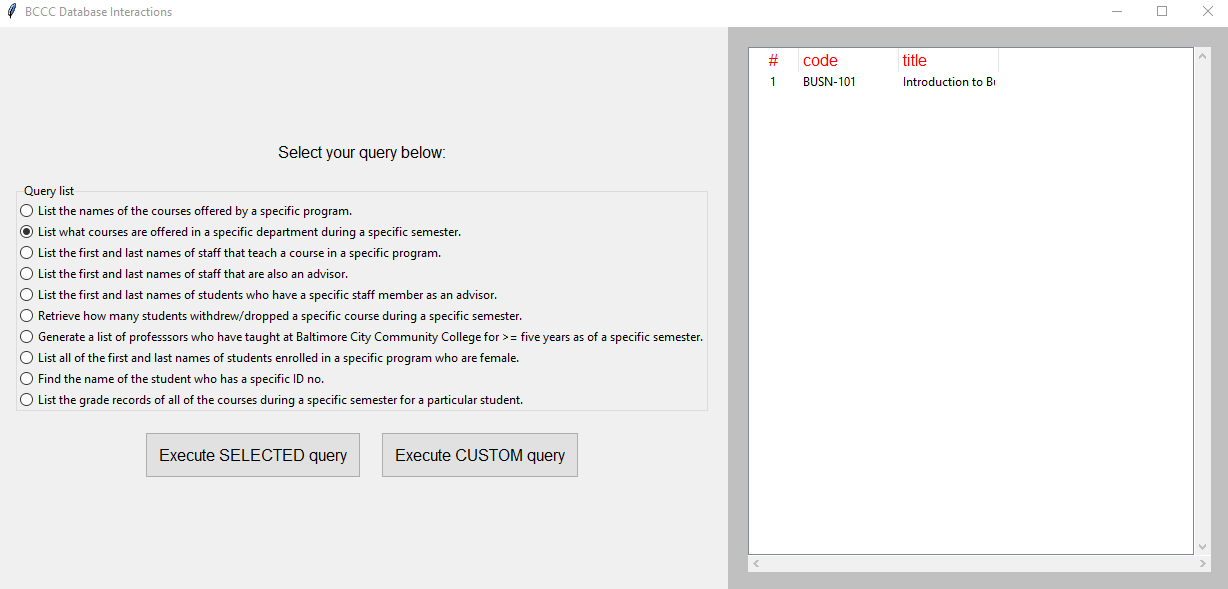
* 1. We hit the “Execute SELECTED query”.



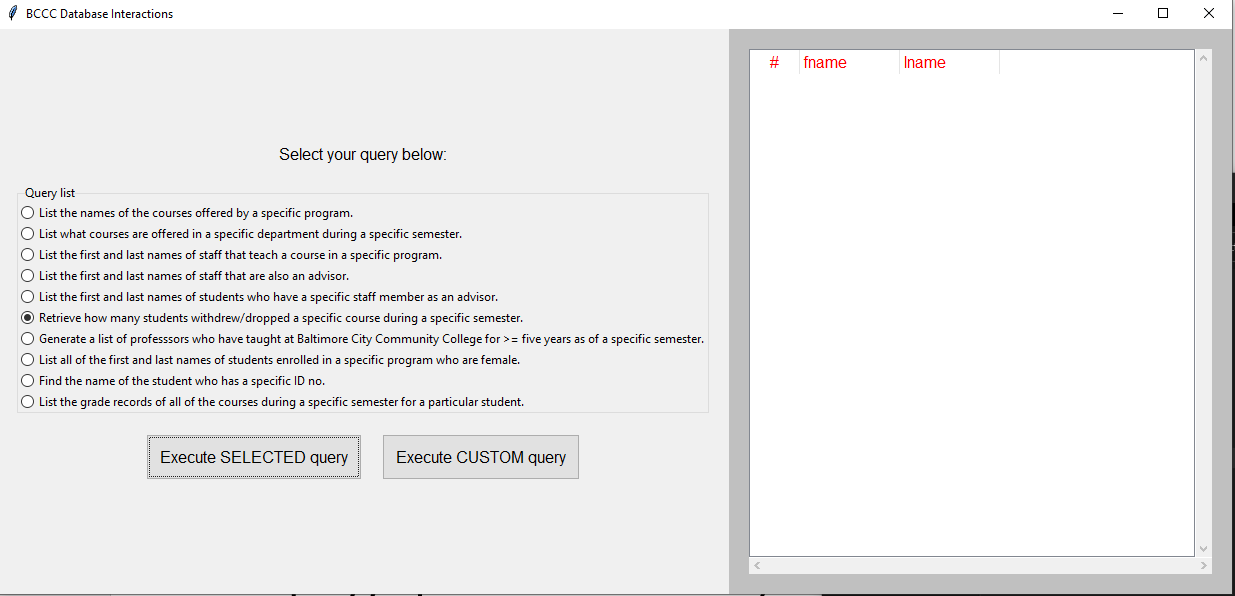
* 1. We type in our parameter we want to search for.



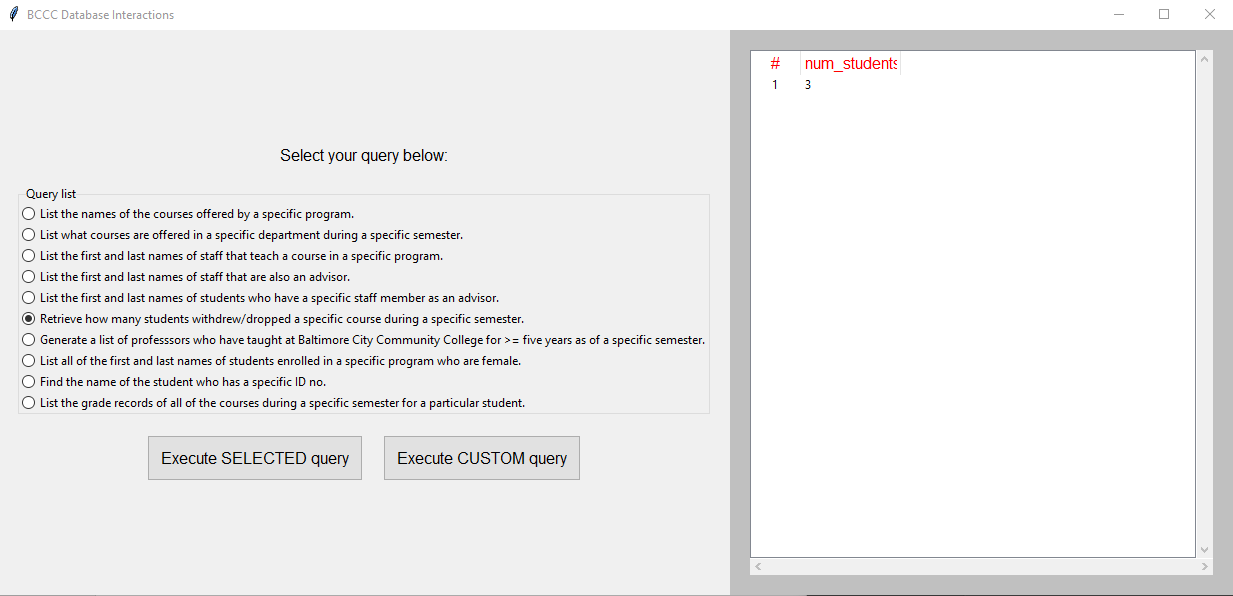
* 1. We hit submit and our results are displayed.



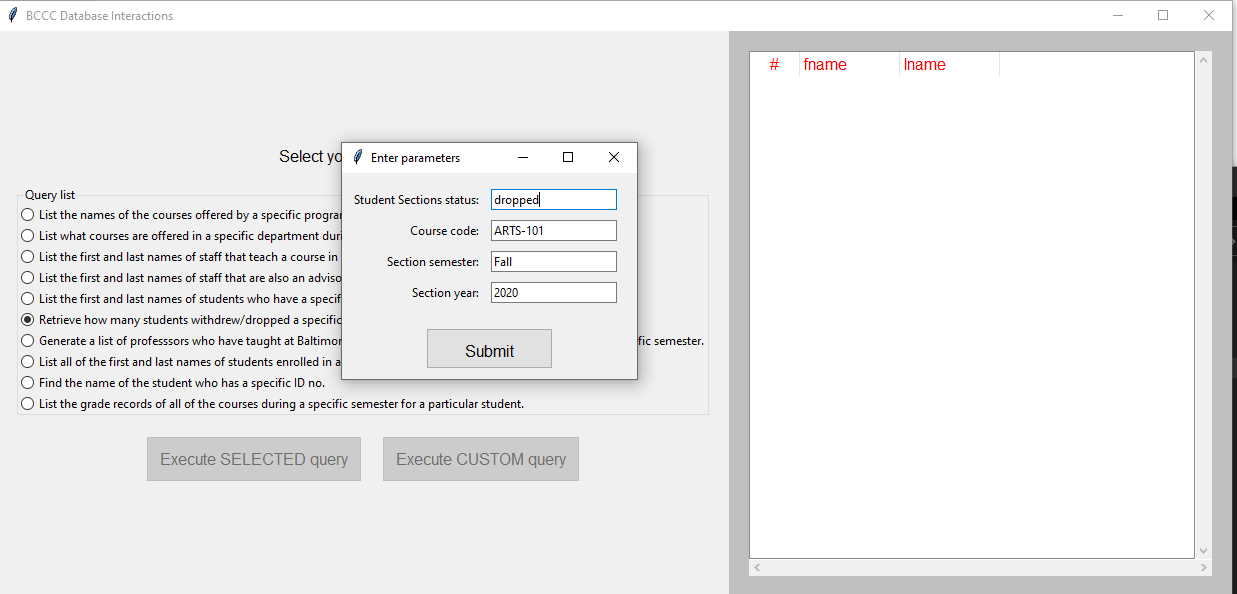
1. Retrieve how many students withdrew/dropped a specific course during a specific semester.
   1. We select the query we want to use, in this case we want to list how many students withdrew/dropped a specific course during a specific semester.



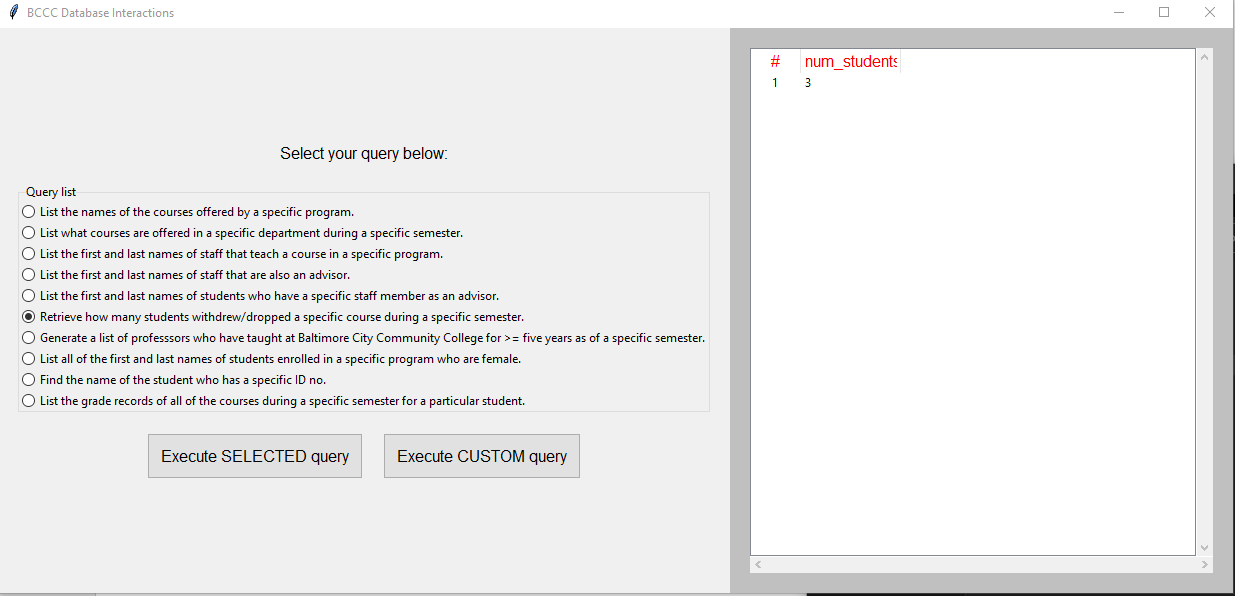
* 1. We hit the “Execute SELECTED query”.



* 1. We type in our parameter we want to search for.

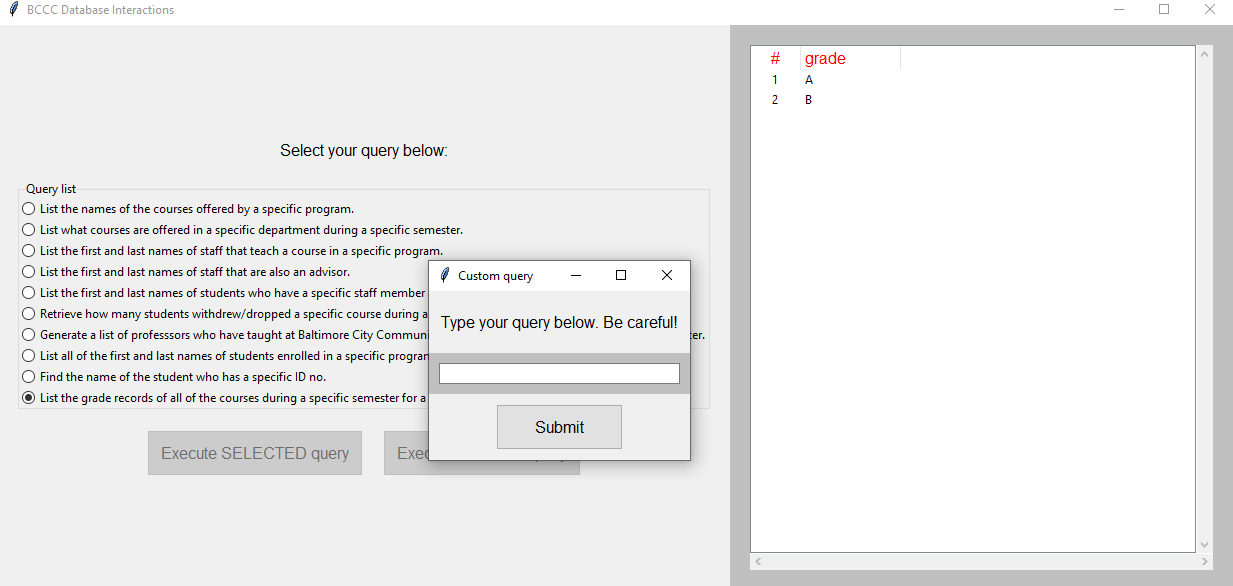


* 1. We hit submit and our results are displayed.

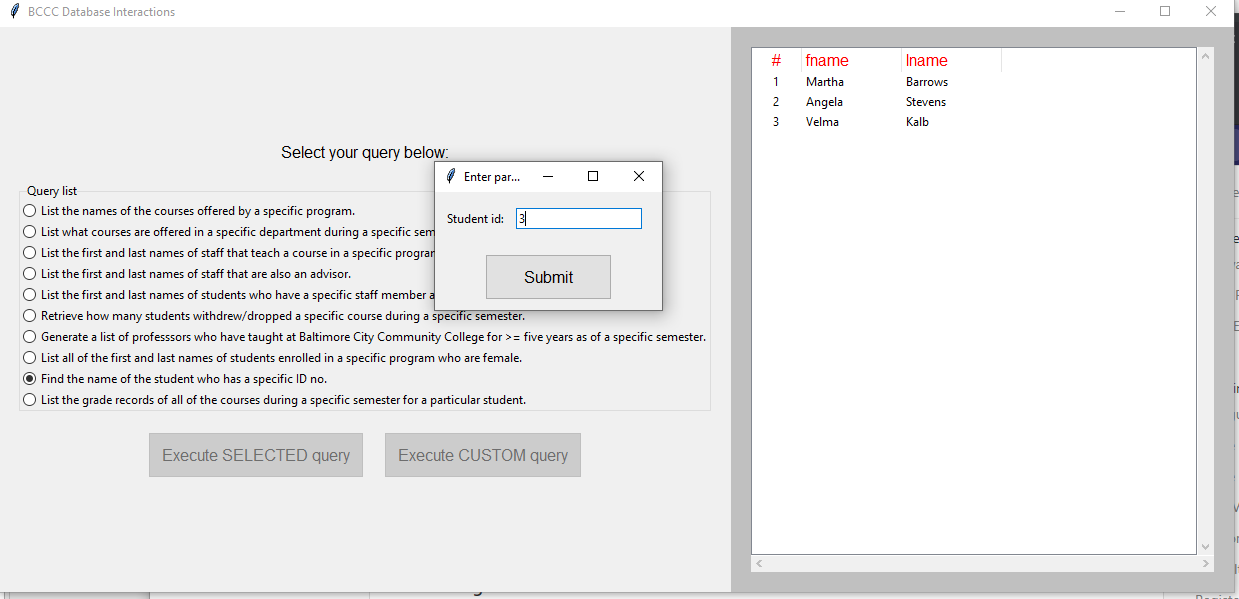


# | **4 – Functions / Reference Manual**

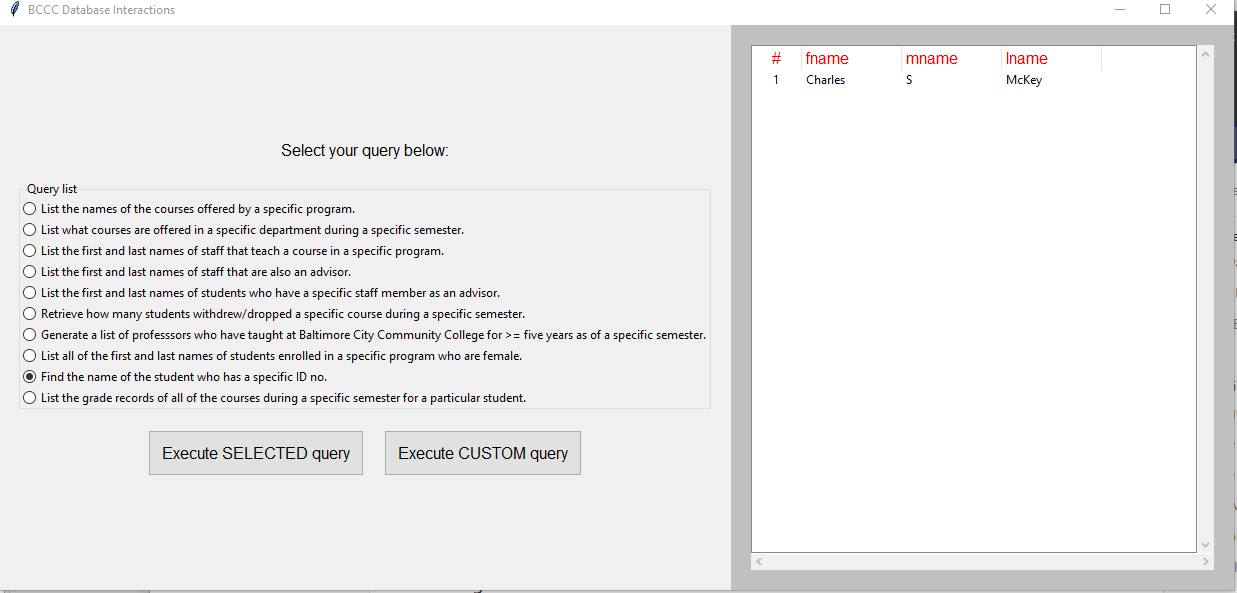
1. Execute CUSTOM query.
   1. Function: This function allows the user to create their own custom queries. Caution and care must be used when using this function as failing to do so can damage the database or delete data by accident.



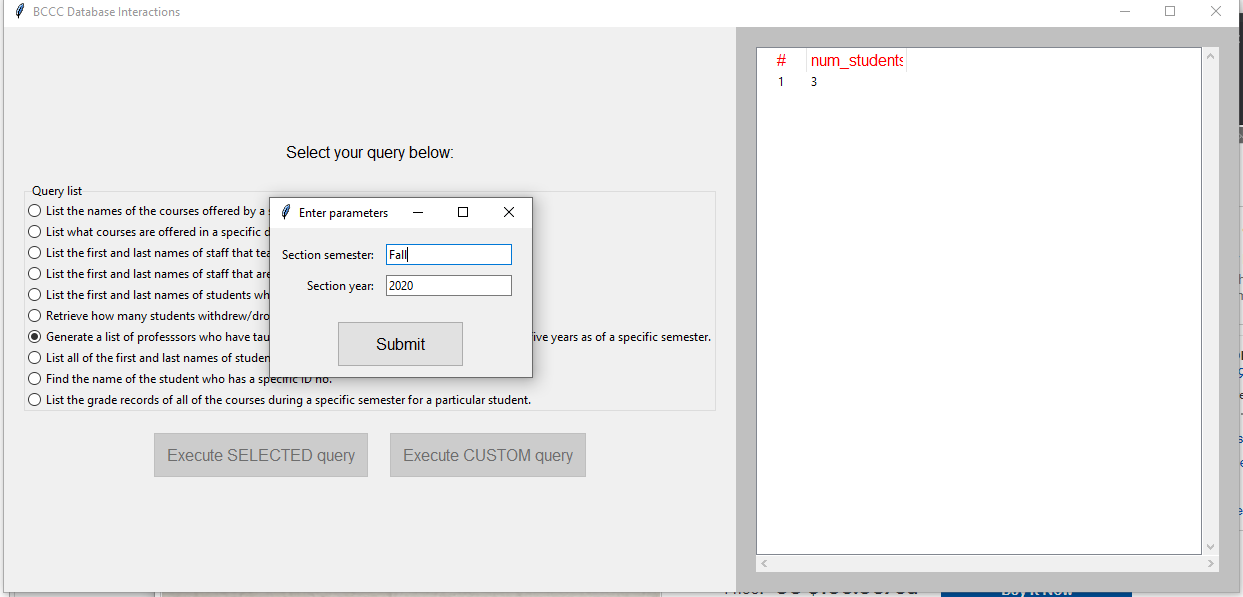
1. Function: Find the name of the student who has a specific ID no.
   1. This function allows the user to find a student with a specific ID no. This function takes a single parameter, the student’s ID no. This is a safe query to execute and has no risk of damaging the database.



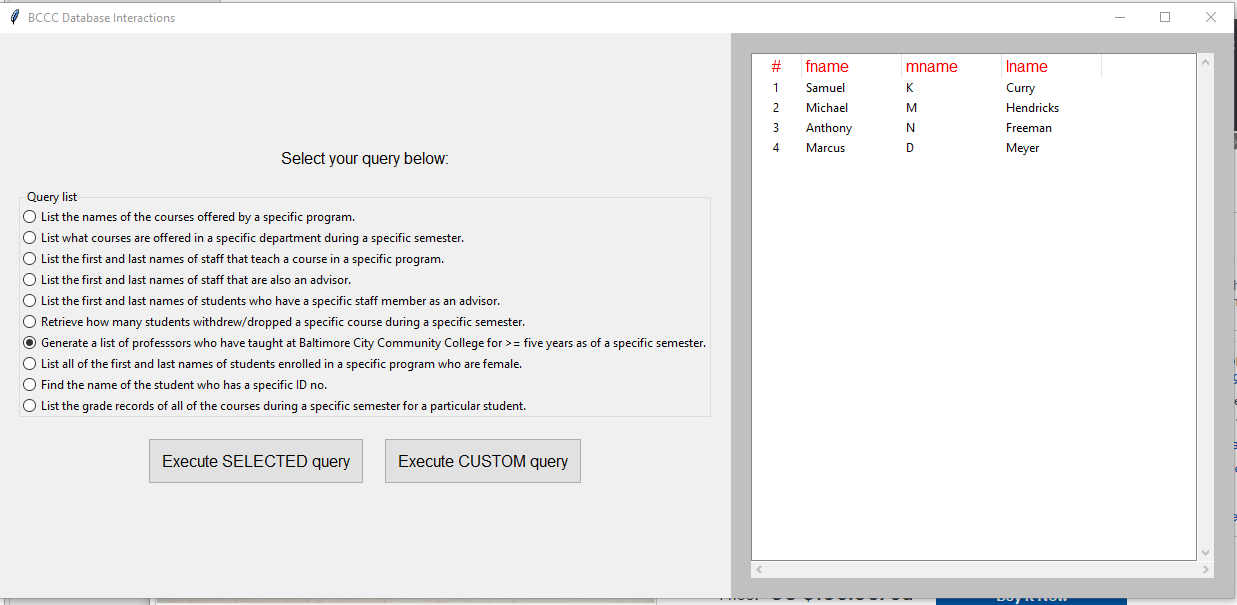
* 1. We hit submit and our results are displayed.



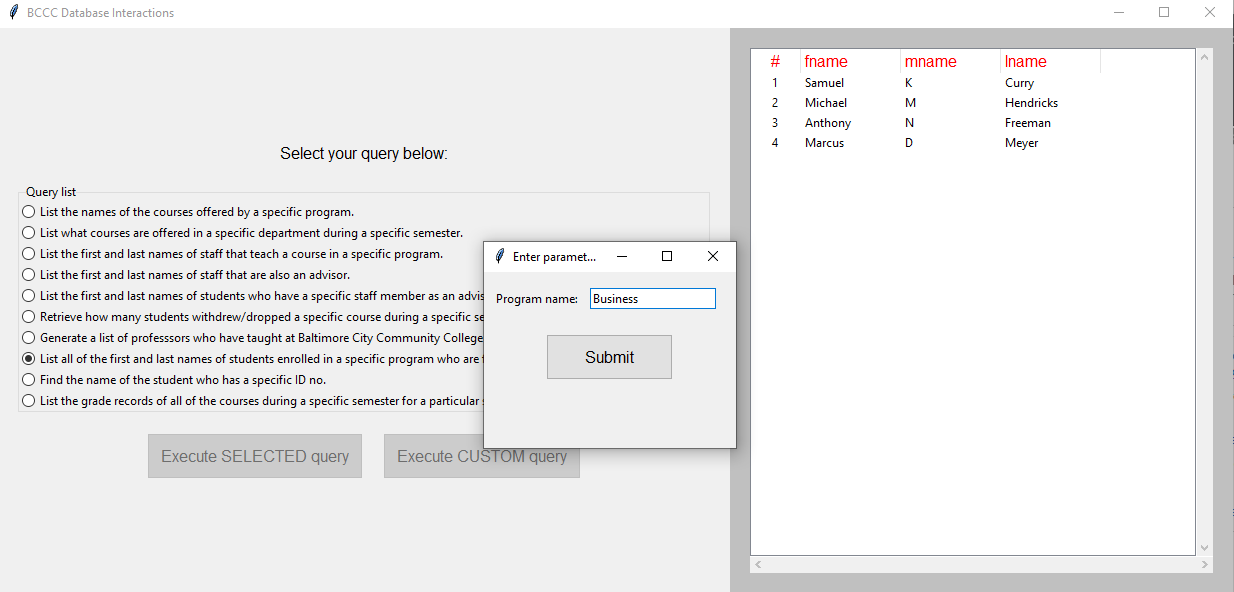
1. Function: Generate a list of professors who have taught at Baltimore City Community College for more than or equal to five years as of a specific semester.
   1. This function allows the user to create a list of professors that have taught for five years and greater as of a specific semester. This function takes two parameters, the Semester and the year. This is a safe query to execute and has no risk of damaging the database.



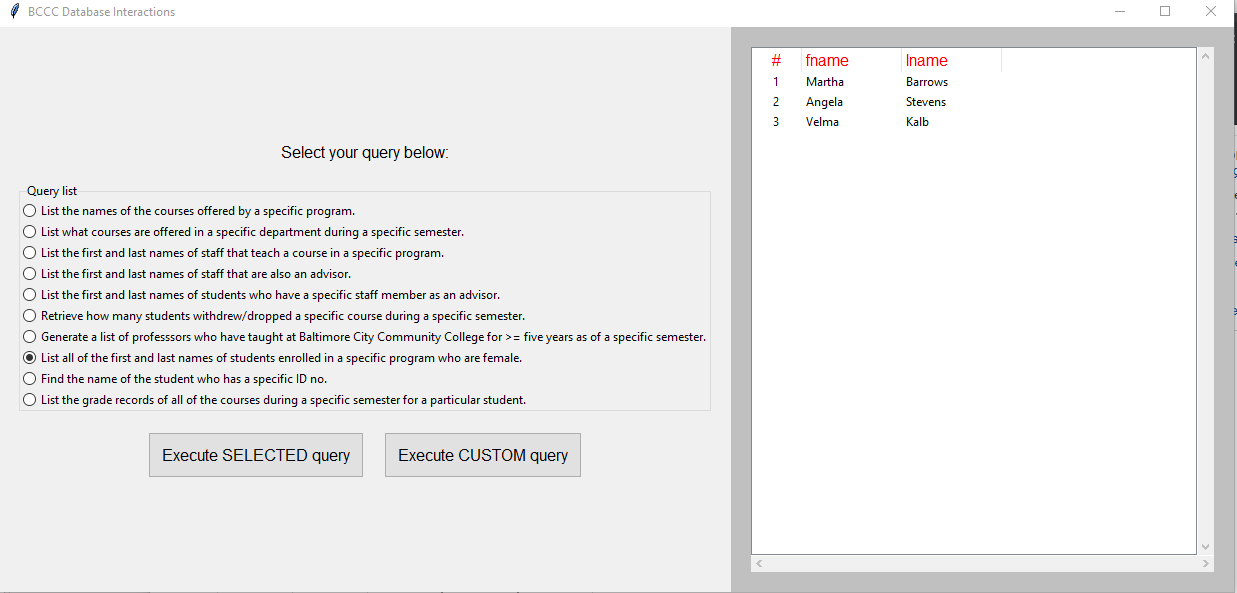
* 1. We hit submit and our results are displayed.



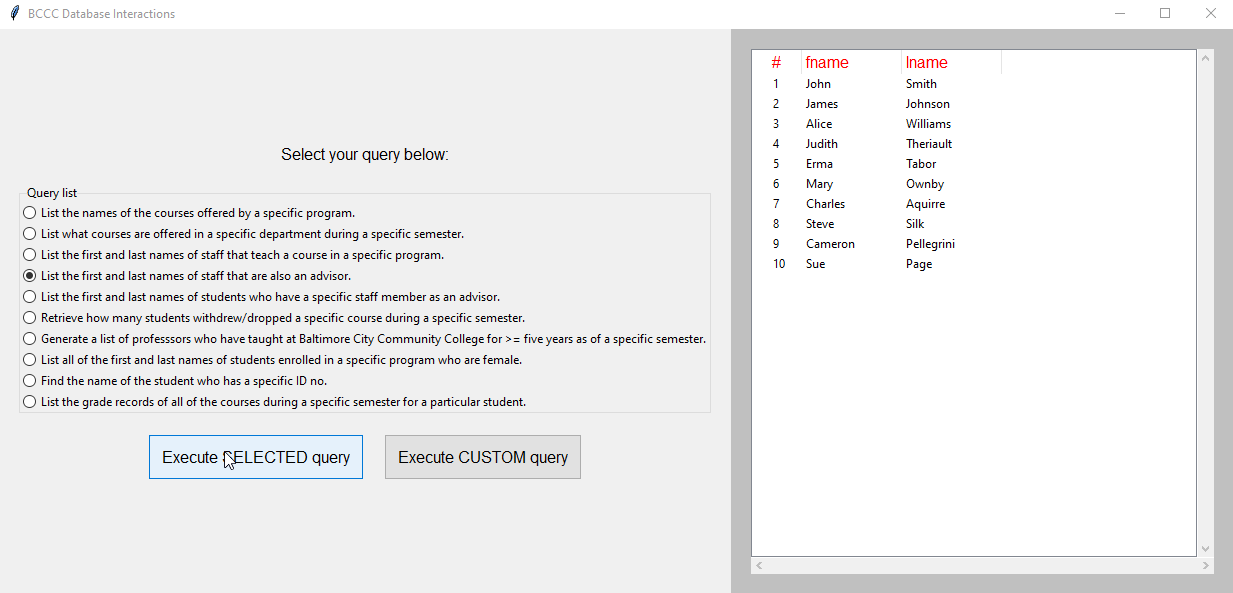
1. Function: List all the first and last names of students enrolled in a specific program who are female.
   1. This function allows the user to create a list of the first and last names of students enrolled in a specific program who are female. This function takes one parameter, the program name. This is a safe query to execute and has no risk of damaging the database.



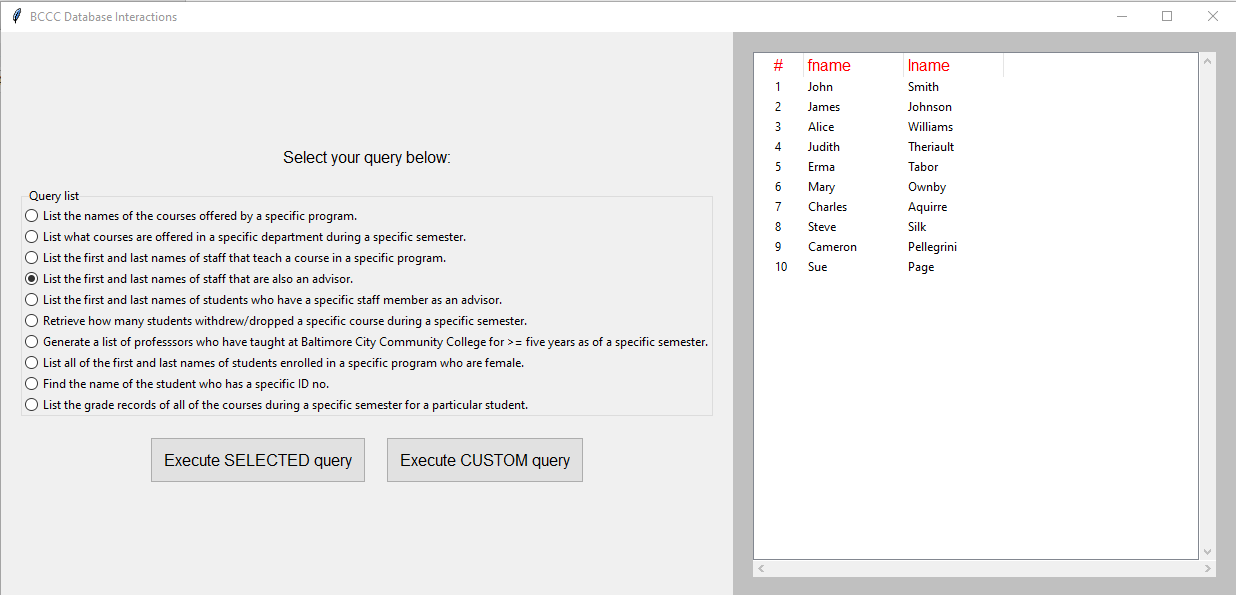
* 1. We hit submit and our results are displayed.



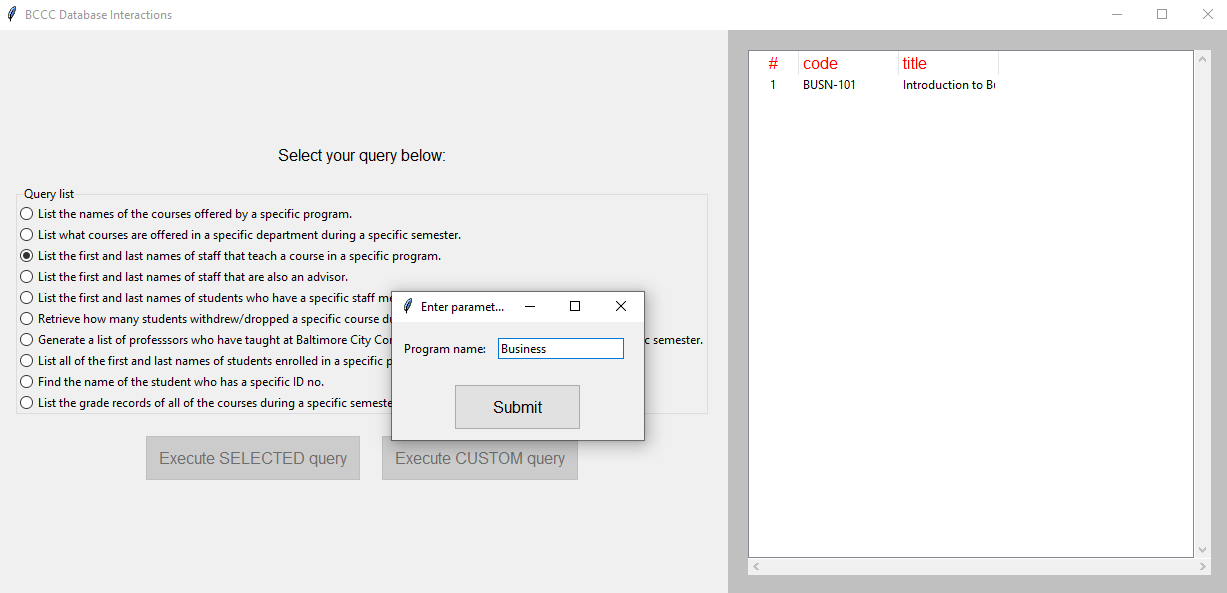
1. Function: List the first and last names of staff that are also an advisor.
   1. This function allows the user to create a list of first and last names of staff that are also an advisor. This function takes no parameters. This is a safe query to execute and has no risk of damaging the database.



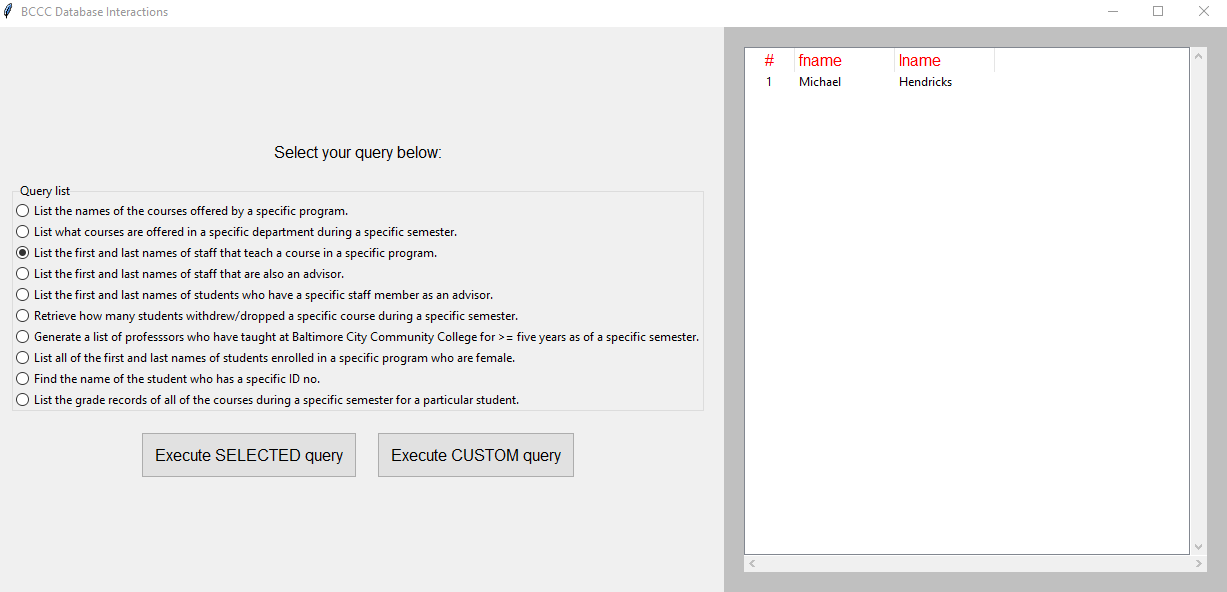
* 1. Our results are displayed.



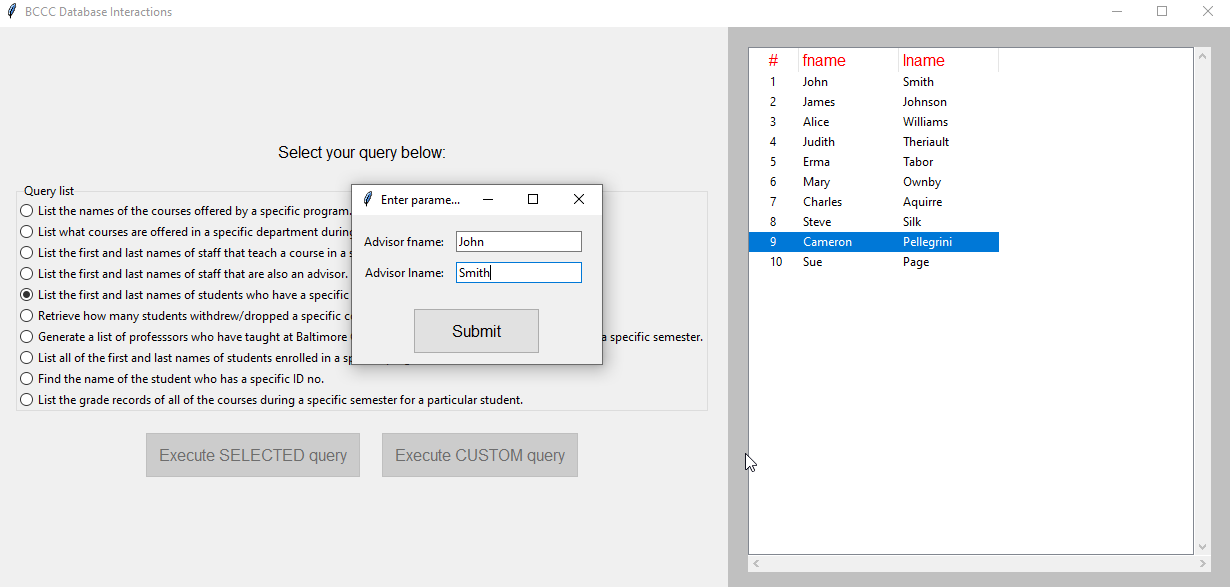
1. Function: List the first and last names of staff that teach a course in a specific program.
   1. This function allows the user to create a list of the first and last names of staff that teach a course in a specific program. This function takes one parameter, the program name. This is a safe query to execute and has no risk of damaging the database.



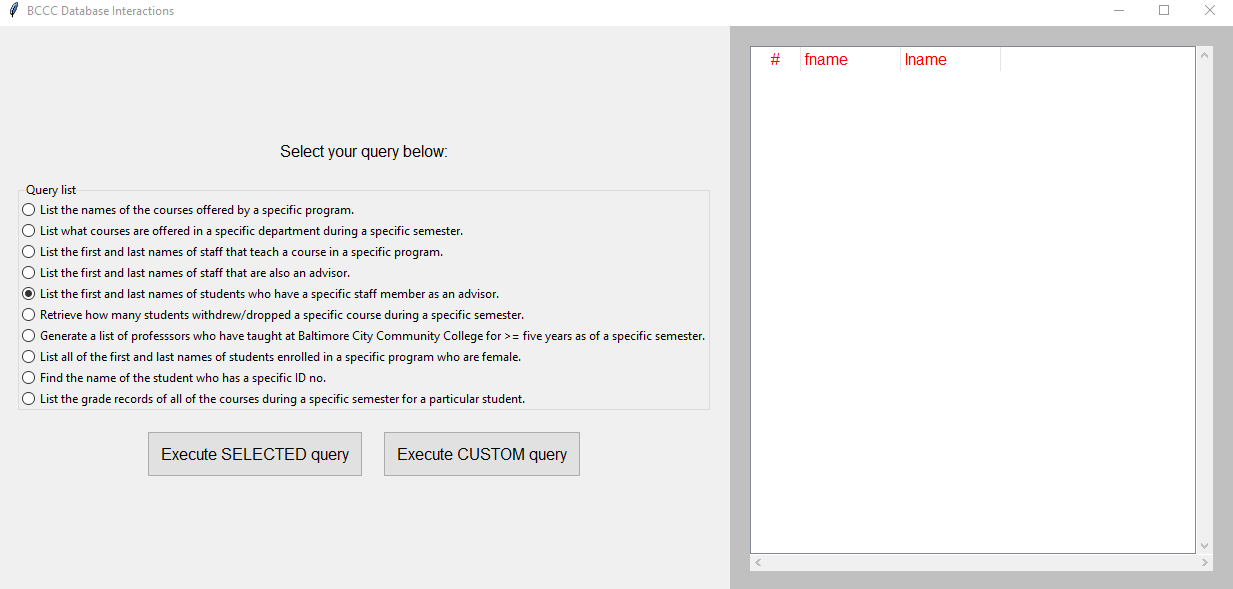
* 1. We hit submit and our results are displayed.



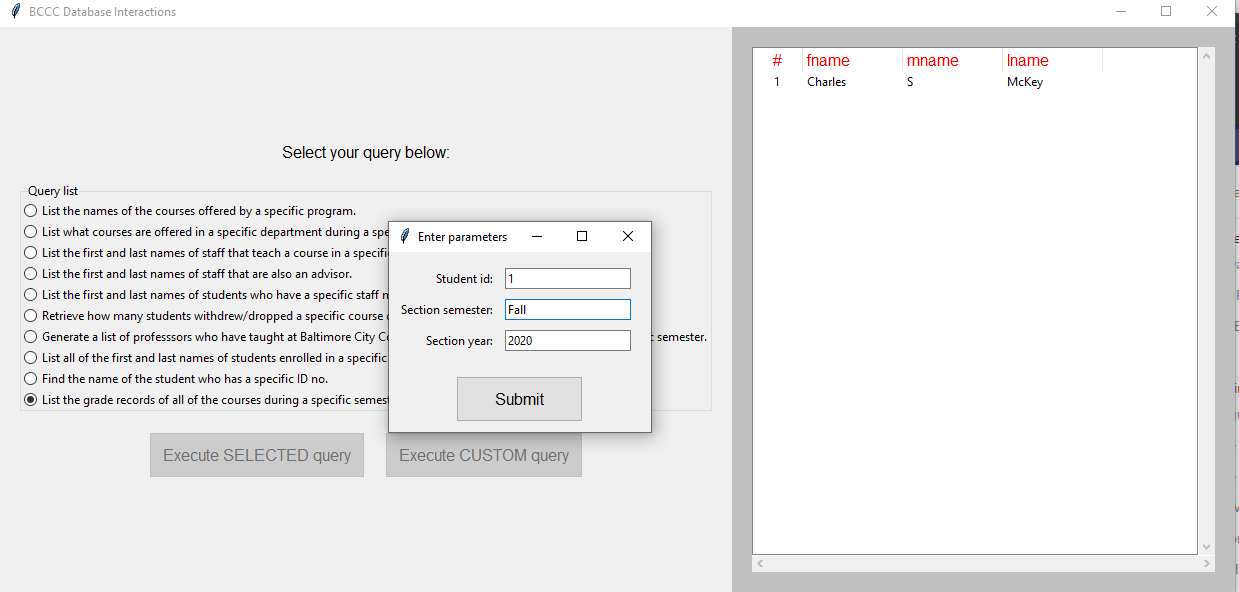
1. Function: List the first and last names of students who has a specific staff member as an advisor.
   1. This function allows the user to create a list of the first and last names of students who has a specific staff member as an advisor. This function takes two parameters, the advisor first name (fname) and advisor last name (lname). This is a safe query to execute and has no risk of damaging the database.



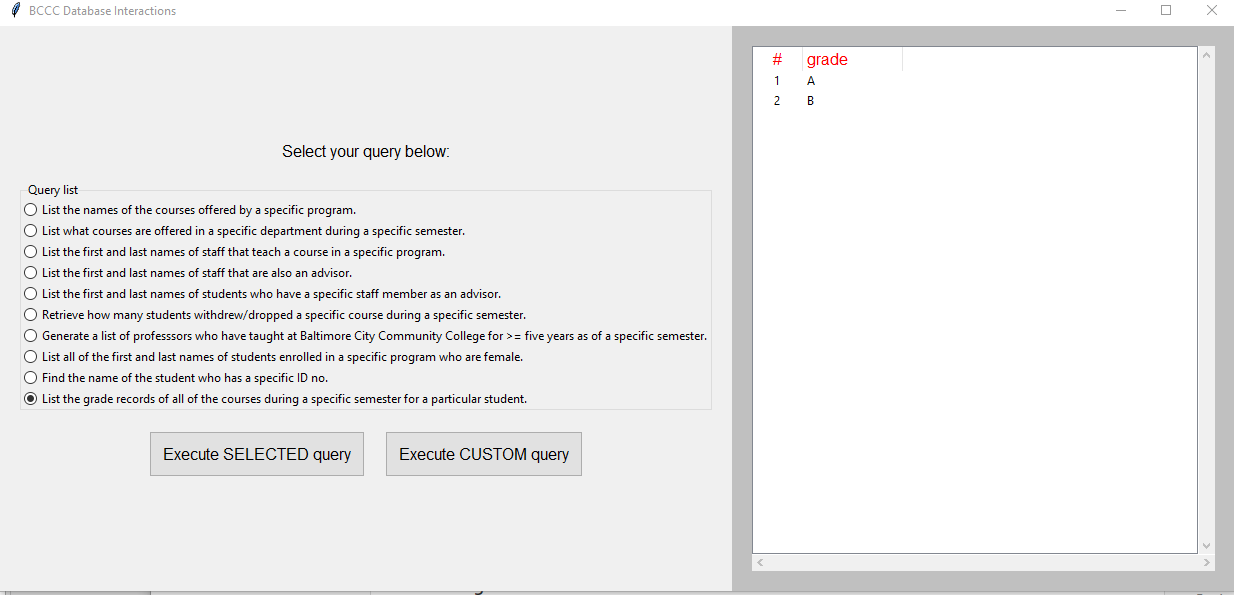
* 1. We hit submit and our results are displayed.



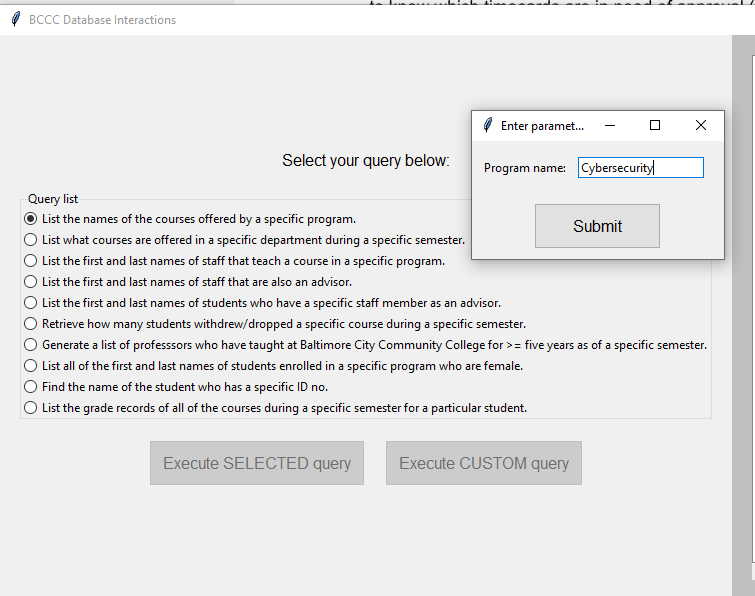
1. Function: List the grade records all the courses during a specific semester for a particular student.
   1. This function allows the user to create a list of the grade records all the courses during a specific semester for a particular student. This function takes three parameters, the Student id, Section Semester and Section Year. This is a safe query to execute and has no risk of damaging the database.



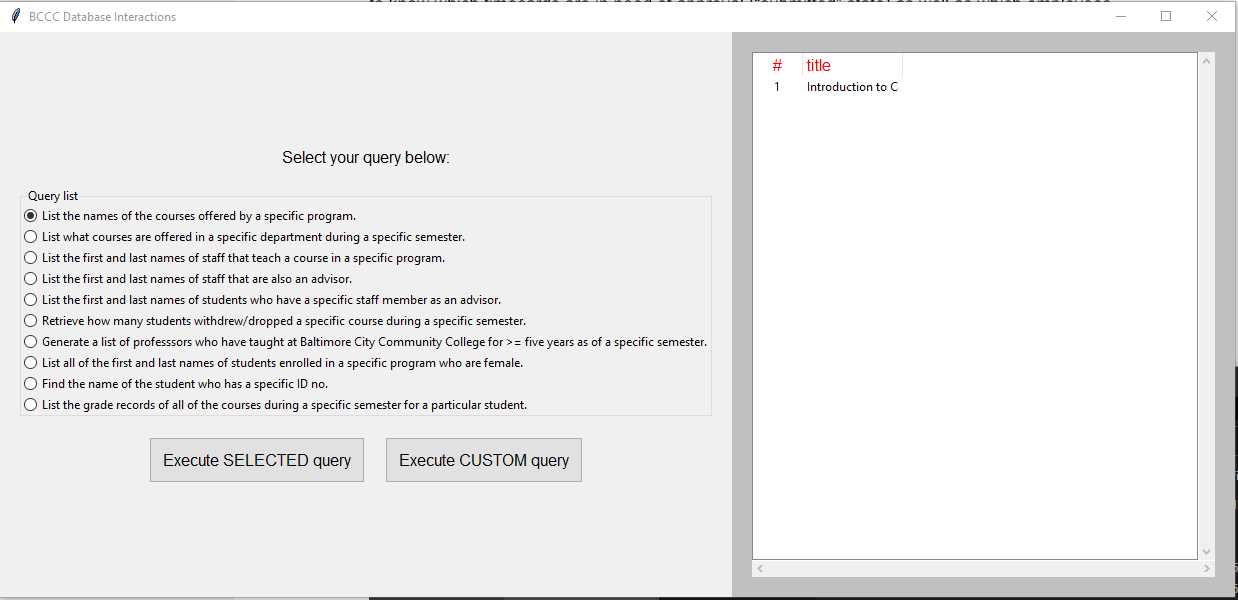
* 1. We type in our parameter we want to search for.



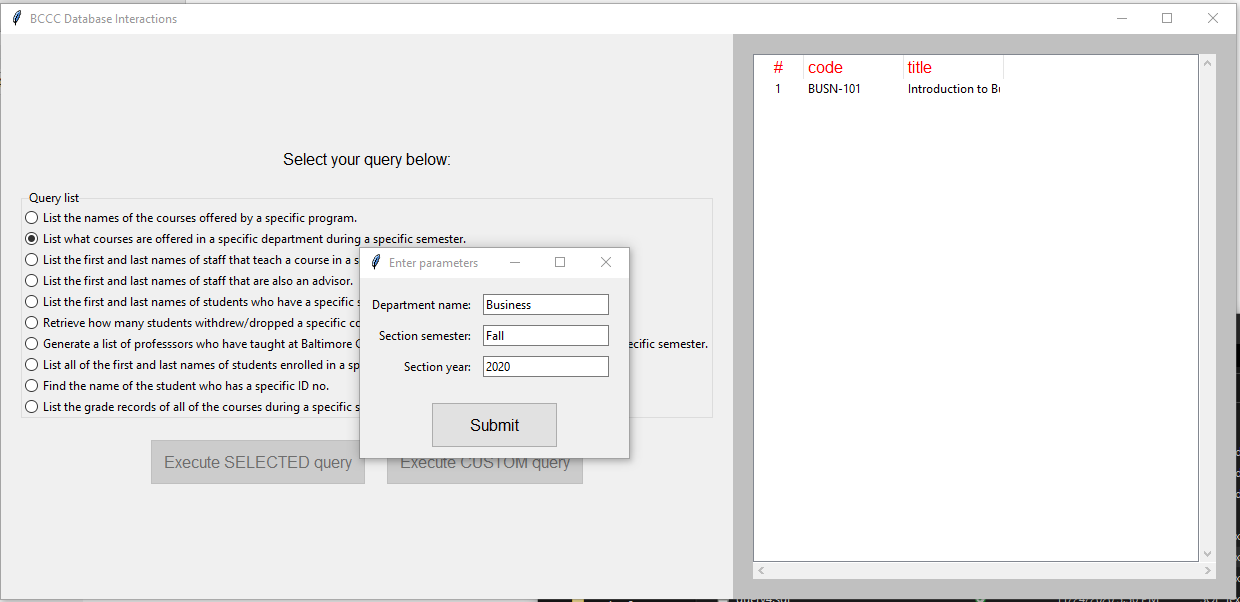
1. Function: List the names of the courses offered by a specific program.
   1. This function allows the user to create a list of the names of the courses offered by a specific program. This function takes one parameter, the Program name. This is a safe query to execute and has no risk of damaging the database.



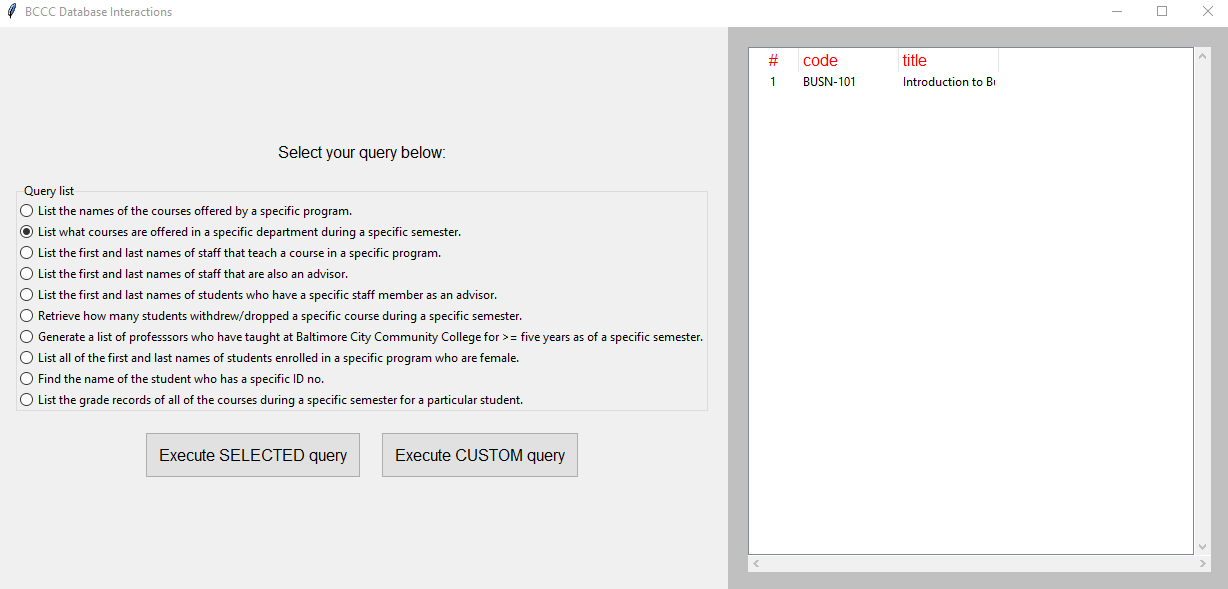
* 1. We hit submit and our results are displayed.

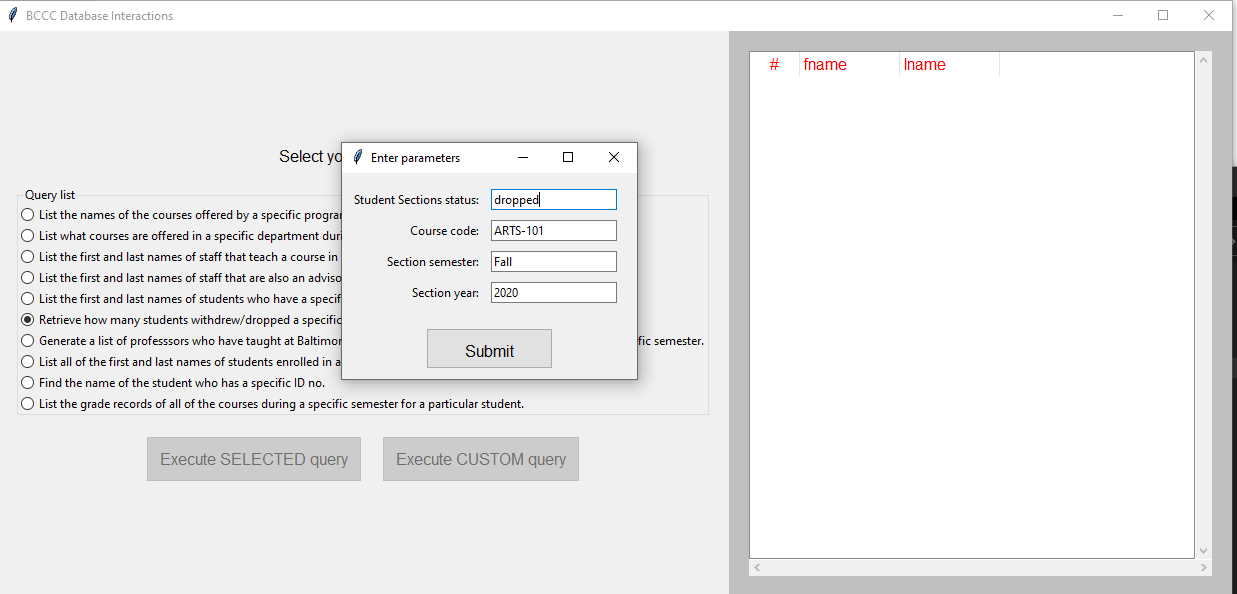


1. Function: List what courses are offered in a specific department during a specific semester.
   1. This function allows the user to create a list what courses are offered in a specific department during a specific semester This function takes one parameter, the Program name. This is a safe query to execute and has no risk of damaging the database



* 1. We hit submit and our results are displayed.



1. Retrieve how many students withdrew/dropped a specific course during a specific semester.
   1. This function allows the user to create a list of how many students withdrew/dropped a specific course during a specific semester. This function takes four parameters, Student Section status, Course code, Section Semester and Section Year. This is a safe query to execute and has no risk of damaging the database
   2. 
   3. We hit submit and our results are displayed.

