NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO

Consider this snippet of Python code that implements a music directory by storing information about song titles, artist names, album names, and release dates in a MySQL database. The user is asked to enter the genre of music for which they are searching. This input is then passed to a database, which uses this information to select songs and returns them to the user. Consider the snippet of code below and answer the following questions, assuming that the code has all required permissions to execute.

```
01  # Import whatever is needed
02
03  genre = input("Please enter your desired genre>")
04
05  # Open database connection
06  db = MySQLdb.connect("localhost","testuser","test123","TESTDB")
07
08  '''
09  Prepares a cursor object that will hold the
10  information on the rows accessed by our query
11  '''
12  cursor = db.cursor()
13  cursor.execute("SELECT songName, albumName, releaseDate "
14      "FROM Songs WHERE genre = '%s'" % genre)
15
16  # Get all the rows
17  results = cursor.fetchall()
18  for row in results:
19     print(row)
20
21  db.close()
```

Questions:

1. What will the program do when executed?

2. What type of information could a user retrieve from this program?

a. The song name
b. The album name
c. The artist name
d. A and B
e. All attributes from the database

[Other statistical questions will be imported here while creating the survey.}

NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.

ANSWERS

1. It will return records from the database that match the specified genre.

2. e

A user can extract information about each of the attributes with a well-crafted string that will lead to SQL injection. For example, if a user entered "`'; select * From Songs--`" they would be able to receive all the records from the table.

NOTE: THE REST OF THE DOCUMENT CONTAINS EXTRA INFORMATION FOR THE PROJECT RESEARCHERS. IT IS NOT TO BE SHOWN TO THE PUZZLE TAKERS.

TAGS:

python, sql, sql-injection, mysql, database

CATEGORIES:

Blindspot - YES
Type - Injection
Number of distinct functions - 8
Number of total functions - 8
Blindspot function - `execute()`
Function call omitted - NO
Blindspot type - Function misuse
Number of parameters in the blindspot function - 1 parameter
Cyclomatic complexity - 2

NAME:

cursor.execute(query_string, params=None, multi=False)

DESCRIPTION:

This method executes the given database operation (query or command). The parameters found in the tuple or dictionary parameters are bound to the variables in the operation. Variables are specified using %s or %(name)s parameter style (that is, using format or pyformat style). The function execute() returns an iterator if multi is True.

BLINDSPOT:

String interpolation for SQL queries can be very dangerous because it does not escape the input parameters correctly and will leave the application open to SQL injection attacks.

CORRECT USE EXAMPLE:

The following procedure queries all the records from a table of EMPLOYEES earning a salary of more than $1000.

```
import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )

# prepare a cursor object using cursor() method
cursor = db.cursor()

# Prepare SQL query to INSERT a record into the database.
sql = "SELECT * FROM EMPLOYEE \
       WHERE INCOME > '%d'" % (1000)
try:
   # Execute the SQL command
   cursor.execute(sql)
   # Fetch all the rows in a list of lists.
   results = cursor.fetchall()
   for row in results:
      fname = row[0]
      lname = row[1]
      age = row[2]
      sex = row[3]
      income = row[4]
      # Now print fetched result
      print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \
             (fname, lname, age, sex, income )
except:
   print "Error: unable to fetch data"

# disconnect from server
db.close()
```

MORE INFORMATION:
#N/A

REFERENCES:
1. http://www.tutorialspoint.com/python/python_database_access.htm
2. http://stackoverflow.com/questions/775296/python-mysql-with-variables