1.

Question 1

Database connection pooling via MySQL Connector/Python API improves the performance of Python-based data-centric applications by avoiding the need to create a new connection each time a user wants to connect to the application.

1 / 1 point

False

True

Correct

Correct! Establishing a new connection for every request to access the MySQL database from the Python-based application is resource intensive and could significantly affect the performance of your application. Connection pooling decreases the required resources.

2.

Question 2

You use a connection from a connection pool to connect to a database. What happens to the connection when you invoke the close module on the connection object after completing the task?

1 / 1 point

The connection between MySQL and Python is closed permanently.

You cannot invoke the close module if you are using the connection for the pool.

The connection is returned to the pool and made available for the next user.

The connection is closed and becomes unavailable for users of the connection pool.

Correct

Correct! The close module puts the connection back into the pool for the next user.

3.

Question 3

You are using a connection from the connection pool to accomplish a certain task on the database using Python. Under what conditions could you expect to encounter the PoolError message? Select all that apply.

1 / 1 point

You request a connection from the pool. However, all the connections in the pool are in use and there is no other connection available.

Correct

Correct! If the connection is not available for use, it raises the error message PoolError: [Failed getting connection; poll exhausted].

You do not pass the pool size parameter when creating the connection pool.

You pass the wrong username or password while configuring the database.

You return the connection after finishing your task. However, the pool is already full and there is no space for another connection.

Correct

Correct! The pool size cannot be increased after it is created. You can only return the connection if the pool is not full. Otherwise, it raises the error message PoolError: [Failed adding connection; queue is full].

4.

Question 4

You want to create a connection pool with four connections using the following code. What syntax is missing to complete this task?

dbconfig = {

"database":"mysql\_database",

"user" : "my\_username",

"password" : "my\_password"

}

pool = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(pool\_name = "pool\_1",

pool\_size = 4,

\*\*dbconfig)

1 / 1 point

MySQLConnectionPool

Correct

Correct! MySQLConnectionPool is a class that defines a pool of MySQL connections using Python.

5.

Question 5

If all the connections from the pool are occupied, then Python automatically adds a new connection using the add\_connection() module to accommodate the new users.

1 / 1 point

True

False

Correct

Correct! You explicitly need to add a new connection using the add\_connection() module to accommodate new users.