AUTOMATED SOFTWARE TESTING
(502072)

Lab 9 – Database testing
Prepared by Nguyen Thanh Quan (MEng)

1. **GOALS:** This lab helps students to
   - Practise automation testing with database.
   - Be able to write test cases with Database Library – Robot Framework.
   - Understand various functionalities of Database Library.
   - Be able to work with database testing.
2. **OBJECTIVES**
   - Install automated testing environment with Database/Robot Framework.
   - Install web drivers.
   - Write and execute test cases.
   - Work with database.
   - Work with database testing.
3. **CONTENT**

3.1 PREREQUISITES

To practise and complete this assignment, students must prepare to install the required libraries below which are also presented in the previous labs. More importantly, students should have a basic understanding of testing concepts, database concept (both relational and non-relational database).

To practise and complete this assignment, students must prepare to install the following libraries and packages.

- **Python**
  Go to python official site – https://www.python.org/downloads/ and download the latest version or the prior version of python as per your operating system.
  Remember to set PATH correctly to use Python after installation.

- **Pip**
  PIP gets installed along with python. Run **`*pip --version*`** to check pip version

- **Robot Framework**
  Use pip – python package manager to install the robot framework and the command for it is as follows

  `pip install robotframework`
  `robot --version` to check robot framework version.

- **wxPython for Ride IDE**
  wxPython is needed for Robot Framework Ride, which is an IDE for Robot Framework.
  This package should be included when you install **`*ride*`.**

  Windows:
  https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/

Linux: Install wxPython with the package manager of OS.

- **Selenium library** (https://github.com/robotframework/SeleniumLibrary/)
  `*pip install --upgrade robotframework-seleniumlibrary*`

- **Selenium webdriver**
  (https://www.selenium.dev/documentation/webdriver/)
  `*pip install --upgrade robotframework-seleniumlibrary*`

- Robot Framework Ride
  Use pip command to install Ride IDE.

  `pip install robotframework-ride`.

To open Ride IDE, run **`*ride.py*`**

3.2 THEORY

- Student must review the theory of Robot Framework and Selenium Library which are presented in the previous labs.

- Database concept.

3.3 PRACTICE

For testing, it becomes important to understand how to interact with the browser and locate the html elements. It is very easy to work with input fields with robot framework. In this lab, we will learn how to work with form fields namely textbox, radio button, dropdown, checkbox, text area etc using Selenium Library. To work with form field elements, we need the locator, which is the main unique identifier for that the fields and it can be id, name, class, etc.

Let create a new project and import Selenium Library properly.

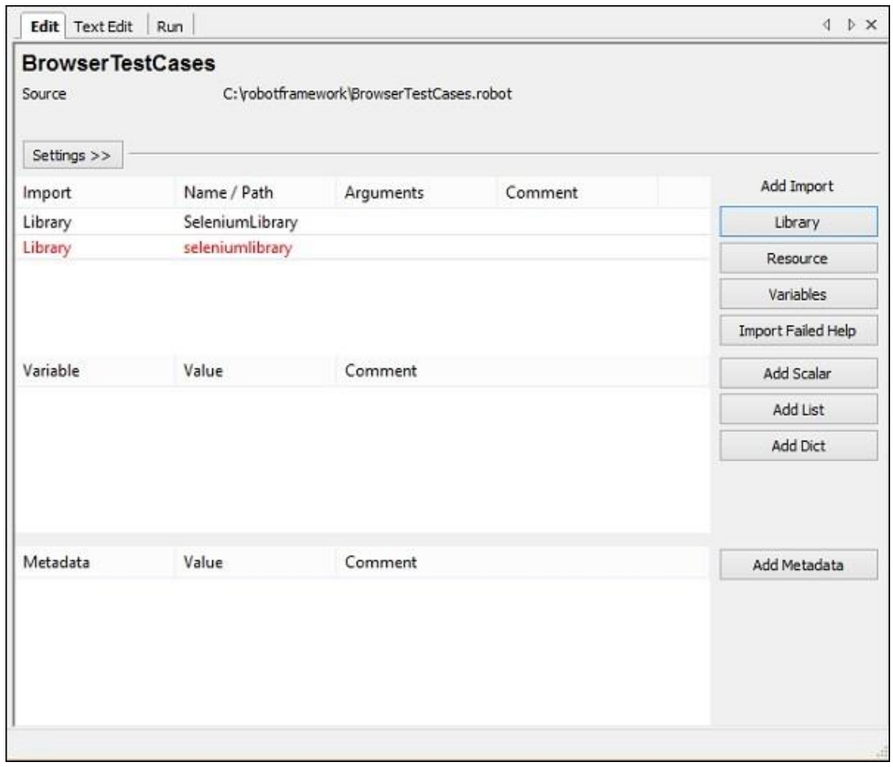

Figure 1 - Valid/invalid library import

The name given has to match with the name of the folder installed in site-packages. In case the names do not match, the library name will be in red as shown above. Library import in red is as good as the library does not exist inside python. Now, we have

completed selenium library import.

The official document page of Robot Framework provides users with detalied explanation and demo, students should refer to the page during the time working with this lab.

https://docs.robotframework.org/docs

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#introduction

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

3.4 Database Library

- Install: **`*pip install robotframework-databaselibrary*`**

- Reference: https://docs.robotframework.org/docs/different_libraries/database

4. EXERCISES

Read the database description below carefully, and then complete the requirements.

The sample database represents employees in an organization. It contains three types of records: **employees**, **departments**, and **historical records of employees**.

Each employee has **an identification number, name, hire date, salary, and manager**. Some employees earn a commission in addition to their salary. All employee-related information is stored in the emp table.

The sample company is regionally diverse, so the database keeps track of the location of the departments. Each company employee is assigned to a department. Each department is identified by a unique department number and a short name. Each department is associated with one location. All department-related information is stored in the dept table.

The company also tracks information about jobs held by the employees. Some employees have been with the company for a long time and have held different positions, received raises, switched departments, and so on. When a change in employee status occurs, the company records the end date of the former position. A new job record is added with the start date and the new job title, department, salary, and the reason for the status change. All employee history is maintained in the jobhist table.

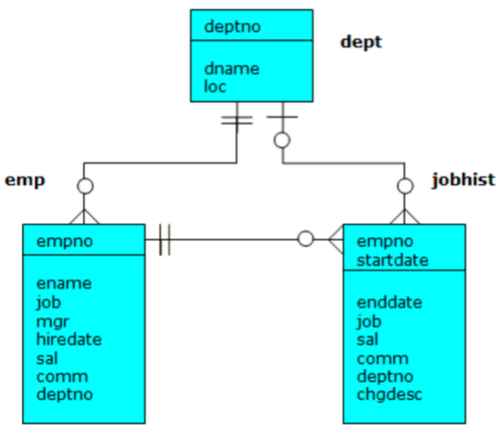

Figure 2 – Database sample

The above is an entity relationship diagram of the sample database tables.

- **Assignment 1 – Create database**

  Using robotframework database library to create tables and their contraints, and then verify if tables are created successfully.

- **Assignment 2 – Insert data**

  Insert 10 samples for each table, and then verify if data is inserted successfully.

- **Assignment 3 – Update data**

  Randomly choose several employees to update their information, and then verify if data is updated successfully.

- **Assignment 4 – Verify that an employee can be deleted**

  Randomly choose an employee to delete his information, and then verify if data is deleted successfully.

- **Assignment 5 – Verify a certain record is present or not**

  Write an SQL statement with SELECT query to get a certain of record, and then check it is present or not, and then verify if data is selected (it depends on your SELECT query).

- **Assignment 6 – Verify if a table is present or not**

  Verify if the three tables exist or not.

- **Assignment 7 – Verify that the row count is 1**

  Write an SQL statement with SELECT FROM WHERE to get a certain of record, and then check the number of rows selected is 1.

- **Assignment 8 – Verify that a table can be deleted**

  Choose a table from the three tables to verify that if it can be deleted (Remember CASCADE).

4. REFERENCES

[1] Daich, G., Price, G., Ragland, B., Dawood, M. "Software Test Technologies Report." STSC, Hill Air Force Base, Utah, August 1994.

[2] https://robotframework.org/

5. REVISION HISTORY

Revision	Date	Author(s)	Description
1.0	Dec 2023	Nguyen Thanh Quan (MEng)	Created
1.1	April 2024	Nguyen Thanh Quan (MEng)	Updated database