**Parameters:** Test components – could be SQL command, table name, etc.

**Variables:**

**Purpose:** Make writing tSQLt easier

Automate boring, repetitive tests

**End user:** Semi-technical users with knowledge of SQL but not other coding knowledge

**Python:** Integrate with unittest so unittest library can handle executions/assertions of tests

Each test defined in the workbook 🡪 single unittest test case

For each possible test command, there would be a separate class – parameters to the class would be the same as the ones defined in the Excel spreadsheet

TestCommand (base class)

DataUnitTestCase(TestCase) \_\_init\_\_(self, [test commands])

LoadSheettotable(Test command) \_\_init\_\_(self,

TestLoader

**Flow:**

python -m dataunit <workbook>

Calls TestLoader.load(), which reads the Excel file (in memory) and builds the object model for TestCases/TestCommands (in memory) 🡪 output of TestLoader.load() = objects in Python for TestCases/TestCommands

When the loader returns the test suite, call the runner 🡪 Unittest.testRunner.Run(TestSuite)

TestSuite = list of test cases or test suites

**Get familiar with source code for unittest package**