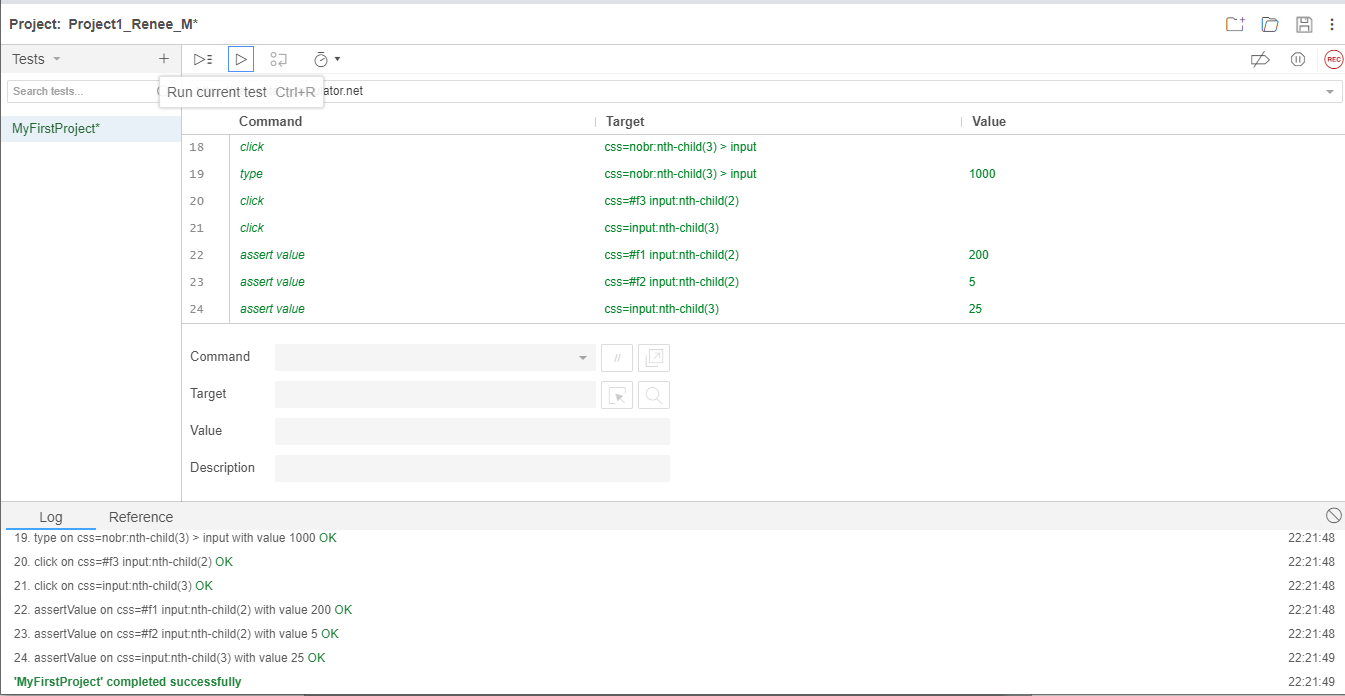
|  |  |
| --- | --- |
| **Name** | Renée Manigo |
| **Date** | 14 September 2020 |
| **Assignment 1001** | Verifying three rows in Selenium on <https://percentagecalculator.net/> |

1. **Test Cases and Java Code**



|  |
| --- |
| **Java Code** |
| // Generated by Selenium IDE  import org.junit.Test;  import org.junit.Before;  import org.junit.After;  import static org.junit.Assert.\*;  import static org.hamcrest.CoreMatchers.is;  import static org.hamcrest.core.IsNot.not;  import org.openqa.selenium.By;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.openqa.selenium.chrome.ChromeDriver;  import org.openqa.selenium.remote.RemoteWebDriver;  import org.openqa.selenium.remote.DesiredCapabilities;  import org.openqa.selenium.Dimension;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.interactions.Actions;  import org.openqa.selenium.support.ui.ExpectedConditions;  import org.openqa.selenium.support.ui.WebDriverWait;  import org.openqa.selenium.JavascriptExecutor;  import org.openqa.selenium.Alert;  import org.openqa.selenium.Keys;  import java.util.\*;  import java.net.MalformedURLException;  import java.net.URL;  public class MyFirstProjectTest {  private WebDriver driver;  private Map<String, Object> vars;  JavascriptExecutor js;  @Before  public void setUp() {  driver = new ChromeDriver();  js = (JavascriptExecutor) driver;  vars = new HashMap<String, Object>();  }  @After  public void tearDown() {  driver.quit();  }  @Test  public void myFirstProject() {  driver.get("https://percentagecalculator.net/");  driver.manage().window().setSize(new Dimension(1015, 708));  driver.findElement(By.cssSelector("body")).click();  driver.findElement(By.cssSelector("#f1 > .values > input")).click();  driver.findElement(By.cssSelector("#f1 > .values > input")).sendKeys("50");  driver.findElement(By.cssSelector("#f1 nobr > input")).click();  driver.findElement(By.cssSelector("#f1 nobr > input")).sendKeys("400");  driver.findElement(By.cssSelector("#f1 > .results > input:nth-child(1)")).click();  driver.findElement(By.cssSelector("#f1 input:nth-child(2)")).click();  driver.findElement(By.cssSelector("#f2 > .values > input")).click();  driver.findElement(By.cssSelector("#f2 > .values > input")).sendKeys("300");  driver.findElement(By.cssSelector("#f2 nobr > input")).click();  driver.findElement(By.cssSelector("#f2 nobr > input")).sendKeys("6000");  driver.findElement(By.cssSelector("#f2 > .results > input:nth-child(1)")).click();  driver.findElement(By.cssSelector("#f2 input:nth-child(2)")).click();  driver.findElement(By.cssSelector("#f3 nobr:nth-child(2) > input")).click();  driver.findElement(By.cssSelector("#f3 nobr:nth-child(2) > input")).sendKeys("800");  driver.findElement(By.cssSelector("nobr:nth-child(3) > input")).click();  driver.findElement(By.cssSelector("nobr:nth-child(3) > input")).sendKeys("1000");  driver.findElement(By.cssSelector("#f3 input:nth-child(2)")).click();  driver.findElement(By.cssSelector("input:nth-child(3)")).click();  {  String value = driver.findElement(By.cssSelector("#f1 input:nth-child(2)")).getAttribute("value");  assertThat(value, is("200"));  }  {  String value = driver.findElement(By.cssSelector("#f2 input:nth-child(2)")).getAttribute("value");  assertThat(value, is("5"));  }  {  String value = driver.findElement(By.cssSelector("input:nth-child(3)")).getAttribute("value");  assertThat(value, is("25"));  }  }  } |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **COMMANDS** | | | |
| **DDL** | Create | Alter | Drop | Truncate |
| **DML** | Insert | Update | Delete | Select |
| **DCL** | Grant access | Revoke |  |  |
| **TCL** | Commit | Rollback |  |  |

**2. Write the 4 types of commands in SQL**