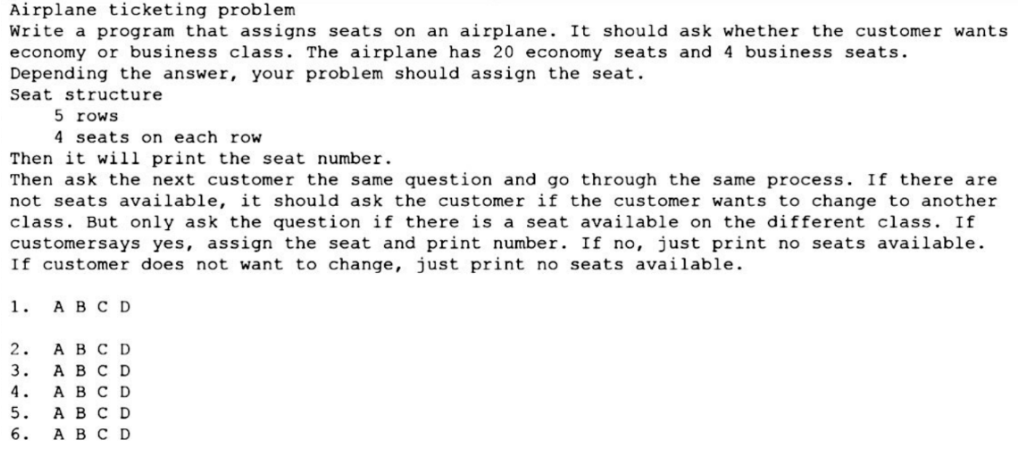
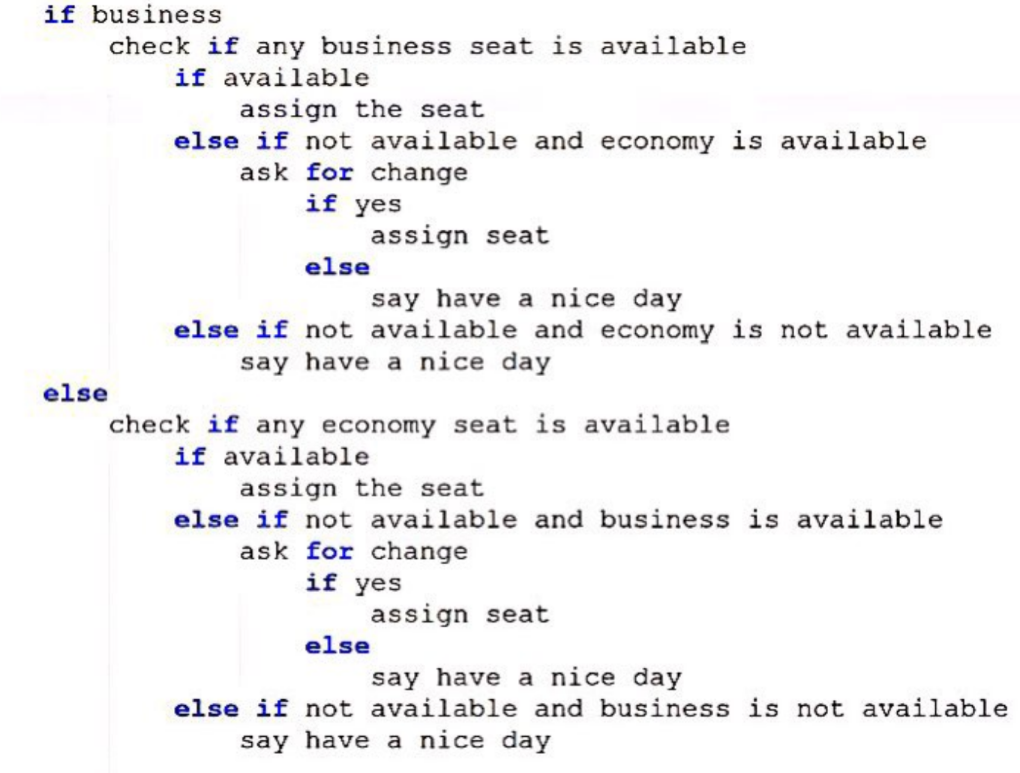
**Java**





**Selenium**

1. **For Database Testing in Selenium Webdriver what API is required?**

For Database Testing in Selenium Webdriver, you need JDBC (Java Database Connectivity) API. It allows you to execute SQL statements.

1. **When should I use Selenium IDE?**

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

1. **When should I use Selenium Grid?**

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

1. **How can you find if an element is displayed on the screen?**

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()
2. isSelected()
3. isEnabled()

**Syntax:**

* **isDisplayed():**

***boolean****buttonPresence = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

* **isSelected():**

***boolean****buttonSelected = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

* **isEnabled():**

***boolean****searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();*

**5. How can we get a text of a web element?**

Get command is used to retrieve the inner text of the specified web element. The command doesn’t require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

**Syntax:**  
*String Text = driver.findElement(By.id(“Text”)).getText();*

**6. How to select value in a dropdown?**

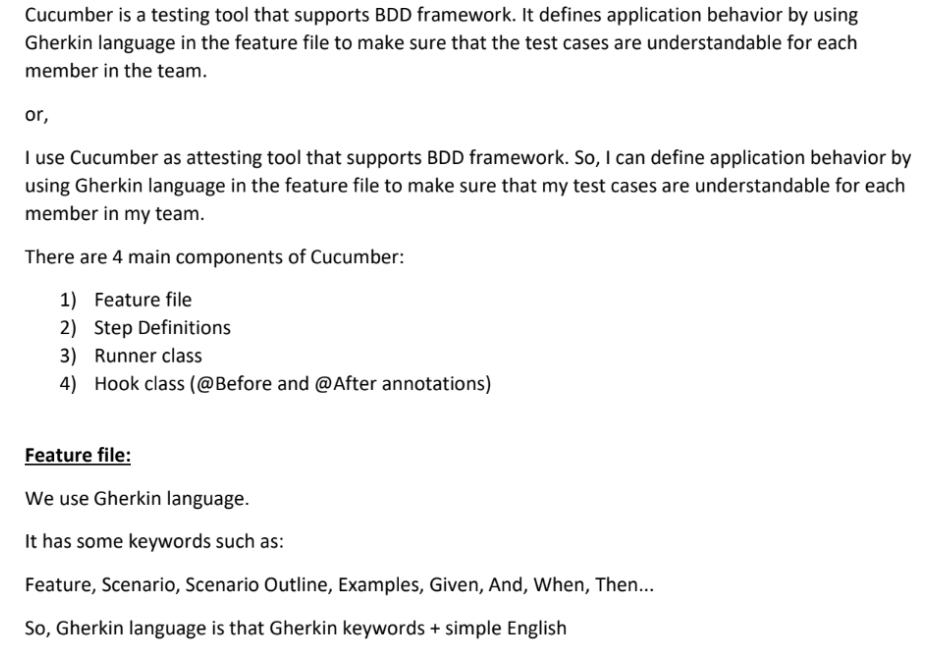
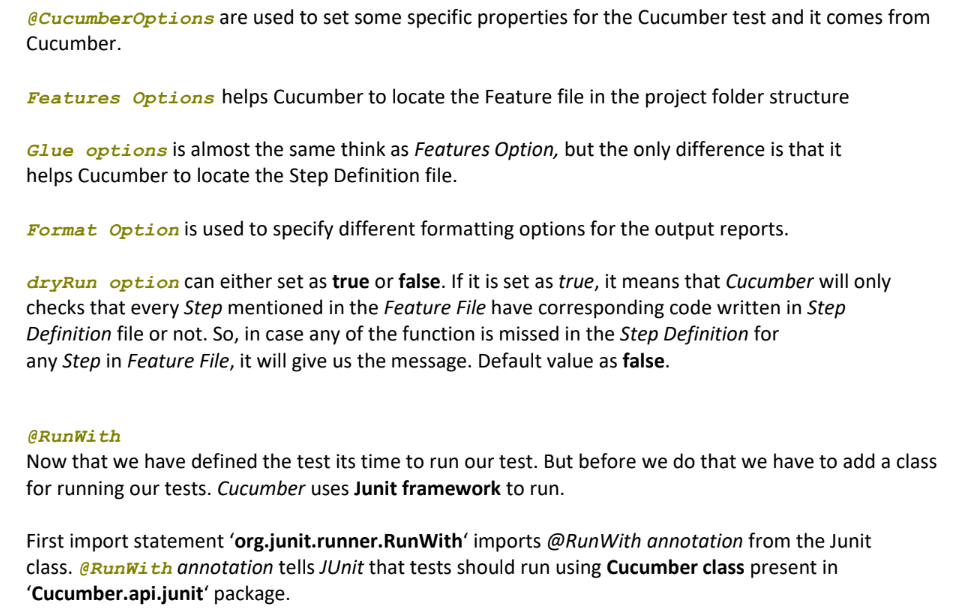
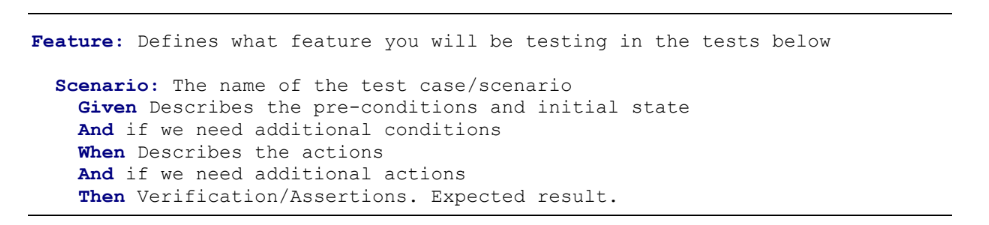
Value in the drop down can be selected using WebDriver’s Select class.

**Syntax:**

* **selectByValue:**  
  *Select selectByValue =****new****Select(driver.findElement(By.id(“SelectID\_One”)));*  
  *selectByValue.selectByValue(“greenvalue”);*
* **selectByVisibleText:**  
  *Select selectByVisibleText =****new****Select (driver.findElement(By.id(“SelectID\_Two”)));*  
  *selectByVisibleText.selectByVisibleText(“Lime”);*
* **selectByIndex:**  
  *Select selectByIndex =****new****Select(driver.findElement(By.id(“SelectID\_Three”)));*  
  *selectByIndex.selectByIndex(2);*

**7. What is the difference between driver.close() and driver.quit command?**

* **close()**: WebDriver’s close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver.
* **quit()**: Unlike close() method, quit() method closes down all the windows that the program has opened.

**CUCUMBER********

# GIT

Git is a version control system.

* **Version Control**

It is making flags/check point whenever your code is changed. Those flags or check

points help to go back to the version is working properly or to the version you want

to use at the moment.

* **Git vs. GitHub**

Git is a version control system

GitHub is a company that provides a platform to do version controlling

* **Version Control System Actions**
* **Commit**: when saving the initial source code
* **Push**: when modifications are made to the source code
* **Pull** request/clone: when a copy of the current source code is made, to work on
* **Push**: when a third part makes some edits. It can be accepted or not.
* **Master** is the original copy
* **Merge request** is a copy of the edits along with the master
* **Create a GIT Repository**

Right Click on Java project / Team / Share Project / Configure Git Repository /

Create Repository / finish (Note: we cannot see the Git Repository because it is running in the background)

**Open Git Perspective**

* **Working Directory** is the area where we make changes.
* **Staging Area** is where we are ready to commit (it is also called ‘the index’: “when we are ready to commit, we add files to the index).
* **Commit and Push / Commit Message**: ANY\_MESSAGE / Destination Git Repository

o Whenever we change something we add and Commit Message

o It is not possible to make changes without recording the change

* **Git view to see changes**

Whoever make the first changes, the other developers are obligated to pull (update) / Right Click / Pull

* You can only update when you are working in the same head version as the remote
* When there is a conflict, the person doing the pull need to solve it

Fetch, Merge, Check Out, Feature, and Command Line Git are related to the way Git works internally.

# **JENKINS**

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early. Continuous integration is cheap. Not integration continuously is expensive because it takes longer time to detect the failure and fix the problem.

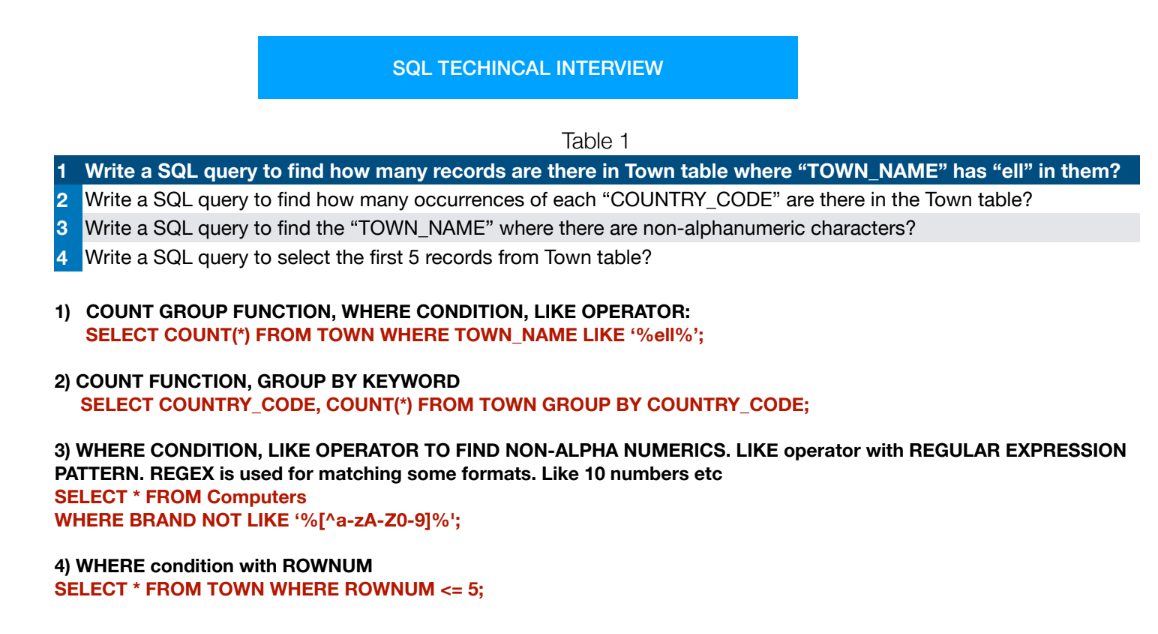
Jenkins is a Built Integration Tool, open source. Jenkins, with the help of different plugins, automates building, testing, packing, staging and deploying the application.

Jenkins takes the changed source code and, by schedule or trigger, it runs and tests it if required. Then, the build output is available in the Jenkins’ dashboard, but automatic notification can also be sent to the tester ready for deployment.

# 

# 

# SQL



# Real Interview Questions

1. **Can you tell me about your current project?**

In my current project we are using a hybrid framework. We have lots of components that need to be

updated on the current project, So because of that we have a lot of new functionalities for old components created before by previous people not working on the project any more. Either I have to learn their script and fix it or I have to recreate the script.

Whenever I get a chance I support the manual testing, I learn the functionalities, and go through the requirements document and then I test the test case manually first before I begin to automate it. I also

support regression. Whenever we have a major release I run my regression script for the functionalities that I developed. I execute 100s of test scripts. If any script fails, I will analyze the results to see if the fail

is due an application issue, environment issue or an issue with my script. If it's due to a defect I reproduce and recreate the defect.

Once in a while every 3 months we have production support. I execute my read only test cases in

the production environment to make sure the environment is stable and ready to go.

1. **Challenge using Keyword driven framework?**

When Multiple people and multiple machines are trying to access excel at the same time the excel file

gets corrupted and throws errors. So for the solution we can put the keywords in the function library instead, so if the test case id says yes then the script will read the keyword from the function library instead of the excel file and we can do this using datatable to export/import to excel.

1. **How do you run your regression?**

We usually schedule ahead of time, a week ahead, we make sure test data and environments are ready. I execute my scripts and monitor the process, and once I get the result I analyze the results. It could be

caused by script, environment or application issue. If its an application issue then I will have to reproduce the defect multiple times to ensure its and application issue.