**Java**

1. **Difference between throw and throws in Java?**

* Both of them are two keywords related to Exception features of Java. The main difference between throw and throws is in their usage and functionality. throw keyword is used to throw an exception explicitly, on the other hand, throws keyword is used to declare an exception which means it works similar to the try--catch block.
* throw is followed by an instance of Exception class throws is followed by exception class names.
* throw new ArithmeticException (“Arithmetic Exception”); throws ArithmeticException;
* throw keyword is used to method body, while throws is used in method signature to declare the exception.
* throw keyword can also be used to break a switch statement without using break keyword
* throws is used in method signature to declare Exception possibly thrown by any method, for example

public void shutdown() throws IOException{

throw new IOException("Unable to shutdown"); }

But throw is actually used to throw Exception in Java code.

Throw new Exception(“is Not able to initialize”);

In other words; throws keyword cannot be used anywhere exception method signature while throw keyword can be used inside method or static initializer block provided sufficient exception handling.

1. **Static and dynamic polymorphism?**

Polymorphism in java is a concept by which we can perform a single action in different ways.

Polymorphism means many forms.

There are two types of polymorphism in java: compile time(static) polymorphism and runtime(dynamic) polymorphism. We can perform polymorphism in java by method overloading and method overriding.

* **Static (compile time)**

Static polymorphism in Java is achieved by method overloading. Method overloading means

there are several methods present in a class having the same name but different types/order/number of parameters.

The compiler looks at the method signature and decides which method to invoke for a particular method call at compile time. So, this is called compile time polymorphism or static binding.

* **Dynamic(runtime)**

Dynamic polymorphism in Java is achieved by method overriding. So, when a call to some method is made, Java waits until runtime to determine which object is actually being pointed to by the reference.

**Selenium**

1. **Do you use JavaScriptExecutor?**

JS has way more control than selenium. We can send JS commands to the browser with using this class

JavaScriptExecutor jsExecutor=(JavaScriptExecutor)driver;

executeScript(); performs the command

* Inside the parameter is where you put JS code
* jsExecutor.executeScript(“alert(‘WARNING: This is a useless message’);” =>This code will bring up a JS popup
* You can also put 2 parameters is .executeScript(“js code”,element);
* Used for scrolling (selenium is not good with scrolling, you can say a challenge is when I was working on a terms and conditions page, where you have to read the page before clicking on continue.
* When I tried using selenium and actions class it didn't work, so i used javaexecutor ) and clicking an element

1. **How to press ENTER key on text box in Selenium WebDriver?**

To press Enter key using Selenium WebDriver, we need to use Selenium Enum keys with its constant.

Driver.findElement(By.xpath(“xpath”)).sendKeys(Keys.ENTER);

**CUCUMBER**

1. **Tell me more about Cucumber, how did you guys decide to start using Cucumber ?**

In the past few years, more and more IT teams follow Agile methodology in their development process to adapt to the rapid changes of the market. This is also a challenge for the test team in managing test cases and test scripts which can be changed when the requirements are updated monthly. Finding a suitable testing method from the beginning is one of the keys to the success of an Agile software project. Many Agile teams have successfully applied Behavior Driven Development (or BDD) approach in testing process using the Cucumber tool.

1. **What is Cucumber? And why is it one of the good approaches in Agile projects?**

Cucumber is a tool for running automated acceptance tests written in a behavior driven development style. One of its wonderful main features is the ability to execute plain text functional description (written in language named Gherkin) as automated tests. Here is an example:

Feature: Update password

Scenario: Admin user can update user password

Given I am in the HR system with an Admin account

When I update password of another “user”

Then I receive a message for updating password successfully

And user’s password is updated to the new password

1. **Tell me about advantages of using BDD approach for testing?**

* Writing BDD tests in Ubiquitous language, a language structured around the domain model and used by all team members including developers, testers, BAs, etc.
* Building bridges between the technical and nontechnical members of a software team
* Allows interaction directly with the developers’ code, but written in a language that business stakeholders can understand
* Last but not least, Cucumber is an Automated Acceptance Test Tool which runs tests written in a behavior driven development (BDD) style.

**2. Tell me what are the most important things in Cucumber, what makes it unique ?**

Features file, Step Defs, Runner Classes, Hook Class, Tags

**3. How to see your reports in cucumber?**

My framework generates a cucumber reports folder in the target folder which contains the reports.

When we run the tests on Jenkins, Jenkins saves the report of every run.

Home page of the Jenkins job always points to the last run reports. All the reports for previous runs can be found under the build number.

Go to target folder

Open with system explorer

Go to target>cucumber report>index shows the tests you ran

**4. What is Gherkin?**

Language used by feature files

Feature, Scenario, Given, Then, When, And, But, BackGround, Scenario Outline

**5. What are the components of Cucumber BDD framework?**

* **Feature files**

Consists of scenarios that test a certain feature or functionality. Feature is main story while scenarios are the test cases to the story(feature)

* **Cukes Runner**

A class that strictly runs the tests, generates codes for step definition

* **Step definition**

A class that made of steps that starts with Gherkin language

Make sure the step definition is in the same package as cukes Runner, or child package (not parent or sibling)

**6. What are Hooks in cucumber?**

Cucumber hook allows us to better manage the code workflow and helps us to reduce the code redundancy. We can say that it is an unseen step, which allows us to perform our scenarios or tests.

Class that uses

@Before → runs before each cucumber scenario

@After → runs after each scenario (It will always run no matter if scenario passes or fails)

# 

# JENKINS

**1. What is JENKINS?**

Continuous Integration and Deployment tool. It is used to schedule and automated builds, deployments etc. It is used by developers and testers.

3 components of Jenkins:

Code change => Devs makes changes to the application code

Test => CI tool automatically picks up the changes and tests the application

Deploy => CI tool deploys the application with changes

**2. What is Jenkins job?**

A task that Jenkins performs based on its schedule. Made of several steps

Can have a trigger that determines when it runs. Reports the results of the run automatically

**3. What is continuous Integration?**

CI is a development practice that requires developers to integrate code into a shared repository several times a day. Every time the software’s code is changed, it is built and tested automatically

1. **What is continuous deployment?**

Code changes are automatically built, tested, deployed, and prepared for a release to production.Each check-in is then verified by an automated build, allowing teams to detect problems early.

1. **Do you maintain Jenkins?**

It is done by Special Team -DevOps Team- , environment team, architecture team. But I provide them information about my tests and configuration info and also the emails to send notifications.

# 

# SQL

1. **Do you know SQL?**

Yes, I am very comfortable with writing SQL Queries and DDL and DML commands.

● Currently working with Oracle database that is running in AMAZON CLOUD SERVER.

● DDL (Data definition language) : CREATE , ALTER, DROP, TRUNCATE

● DML(Data manipulation language): SELECT, DELETE, INSERT, UPDATE

● Structured Query Language. Used for managing and manipulating data in db.

● Provides statements for a variety of tasks:

○ Querying data

○ Inserting, updating ,deleting rows in a table

○ Creating, replacing, altering, and dropping objects

○ Controlling access to the database and its objects

○ Database consistency and integrity

1. **What kind of Database testing are you doing?**

● I am mostly doing Database validations.

● I make changes or insert data (create loan) in the front end and validate in the database. Data in front end matches the DB

● I also make changes using RESTapi and verify that changes are successful in Database as well.

1. **Do you know RDBMS**

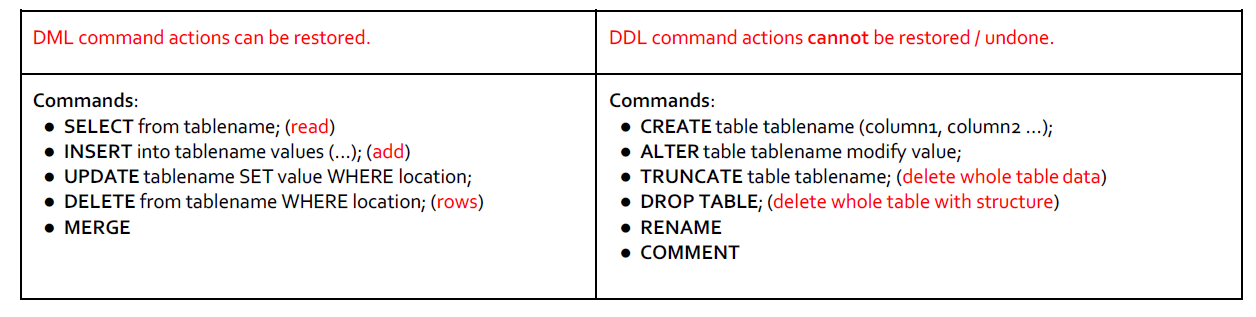
● Relational Database Management System

● Data is organized into tables that are related to each other

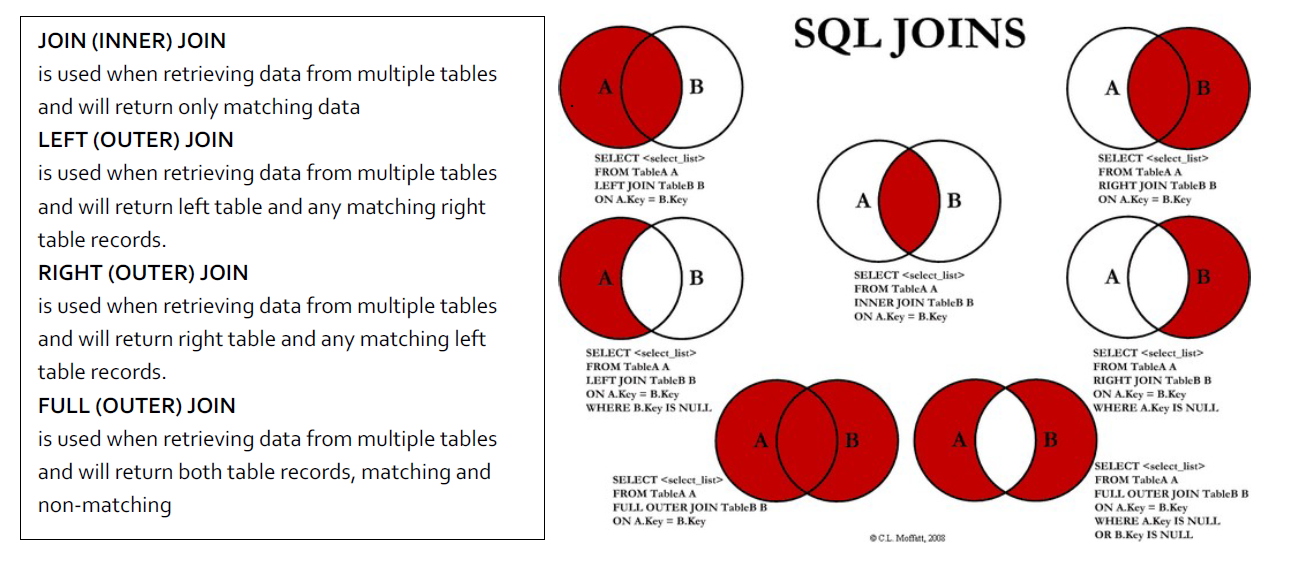
1. **How are they related?**

Primary Key (unique and not NULL) and Foreign Key (duplicate and NULL)

1. **Explain DML (Data Manipulation Language) vs DDL (Data Definition Language)**



1. **Explain SQL JOIN and JOINS**



# 

# Real Interview Questions

1. What is your 2 year plan?
2. You have experience in both manual and automation? which one you prefer and why?
3. Tell me how you use scenario outline in your application?
4. Tell me something you like and didn't like in the past work environments you worked for?
5. How do you find a dynamic element when an element changes id such as id name " button-x",button-y, button-z?
6. What will you do when you have many user stories, and you have two days before the sprint end and you know you will be unable to complete?
7. What is a base page and do you have a base page in your project?
8. What if your supervisor tells you to do something you do not agree with, what will you do?
9. How do you rate yourself in java?
10. Tell me about the key qualities for a quality engineer?
11. What made you become a QA?
12. Have you written a test plan?
13. How many people have you managed or have you had any leadership position?
14. What are your strengths and weaknesses?
15. How can you decide which user stories manual testers will test and which one you will automate?
16. Why are you leaving your company?
17. Tell me how you create a page object model for a login page?
18. How will you get all the links in a page and print all link text?
19. How do you estimate work on your team?
20. How will you respond if a manager comes up to and tells you to do something asap, that is very important and high priority?
21. What are some of the challenges QA face in their career?
22. What are some of the things you like about being a QA?
23. What does SDET mean?
24. How do you develop a traceability matrix? Whom do you share that information to?
25. How do you create reports and share the reports to your teammates?
26. Tell me about a technical challenge while creating Automation Script?
27. Tell us about your trophy BUG?
28. How do you select a Checkbox?
29. What does the Dev Ops do?
30. What would you do if the Developer does not agree with you that the defect you reported is not a Defect?
31. Tell me about Jira?
32. Do you create Test Cases? How do you create a test case?
33. How do you use SQL joins?
34. What is the difference between a Left and an Inner join?
35. Do you have any questions?
36. What is the difference between INTERFACE and CLASS?
37. What is FINAL?
38. Can classes be FINAL?
39. Give me an example of the DATA DRIVEN TEST you performed? Where was your data located to use in the test?
40. How does the FEATURE FILE WORK? Tell me the SYNTAX of an ENTIRE FEATURE FILE?
41. How do you get the snippets for implementing the test script in the STEP-DEFINITIONS file? Walk me through that?
42. How do you run your smoke test and regression test? Tell me the process with CI?
43. What is the entire flow of the code being developed by developers, test scripts being developed by the Automation engineers and then being pushed to GIThub? How often the developers and the Automation people commit the code?
44. How is the build in Jenkins updated after you commit the code?
45. What is the difference between an Error and an Exception and how do we handle them?
46. Who is reviewing your code and if there is any issue with your code what happens?
47. Two people changed the same line of the same code differently, what happens.
48. What kind of tests do you have in your smoke test other than log-in and log-out?
49. Do you use parallel testing, how many ways do you know? What are the disadvantages?
50. Where do you install the Jenkins, on company machine or another service