GIT - JENKINS - AGILE- SQL COMMANDS- MANUAL TESTING

JENKINS

JENKINS:

Jenkins is an leading open source continuous integration server built with Java. It is used to build and test software projects continuously making it easier to integrate changes to the project.

Below are the advantages of Jenkins :

1. Its an open source tool with great community support.
2. Easy to install and It has a simple configuration through a web-based GUI, which speeds up the Job
3. It has around 900+ plugins to ease your work. If a plugin does not exist, just code it up and share with the community.
4. Its built with Java and hence, it is portable on all major platforms.
5. Good documentation and enriched support articles/information available on internet which will help beginners to start easy

What can Jenkins Do?

1. Can associate jenkins with a version control server.
   2. Can trigger builds by polling, Periodic etc.
   3. Can execute bash scripts, shell scripts, ANT and Maven Targets
   4. Can Publish results and send email notifications

2. Continuous Integration (CI) is a practice that requires developers to integrate their code into a shared repository frequently at least daily, which leads to multiple integrations per day by different developers. Each commit made by the developers are then verified by an automated build, which detects problems early and communicate the team members.

Default port : 8080

Create Free- Style Job:

Plugins - Git plugin / Maven Plugin / Cucumber report plugin / or any report plugin

‘

Jenkins workflow:

Create job → Git repository link → Build Plugin - Maven goals → Post Build – Cucumber Report

Build Triggers:

Build periodically - It will run build continously for specific interval of time.

Poll SCM – Same as Build periodically but it will run only if developers commits.

Plugins are downloaded from Manage plugin.

Configure Java Home and Maven Home in configure system tool.

JENKINS WORKFLOW:

Developers check out code into their private workspaces.

When they are done with it they commit the changes to the shared repository (Version Control Repository).

The CI server monitors the repository and checks out changes when they occur.

The CI server then pulls these changes and builds the system and also runs unit and integration tests.

The CI server will now inform the team of the successful build.

If the build or tests fails, the CI server will alert the team.

The team will try to fix the issue at the earliest opportunity.

This process keeps on repeating.

. **Why do you need a Continuous Integration of Dev & Testing?**

For this answer, you should focus on the need of Continuous Integration. My suggestion would be to mention the below explanation in your answer:
Continuous Integration of Dev and Testing improves the quality of software, and reduces the time taken to deliver it, by replacing the traditional practice of testing after completing all development. It allows Dev team to easily detect and locate problems early because developers need to integrate code into a shared repository several times a day (more frequently). Each check-in is then automatically tested.

Faces any Issues in Maven Project basic things to do:

1. In Eclipsce Right click on Project Name go to Maven and click update project in context menu and check force/update release in displayed pop up and apply
2. Right Click maven project and run as maven clean'
3. Go to windows in main menu and click perferences and click java and go to installed JRE's and edit and give JDK Path and apply , then right click on project go to build path and configure build path and libraies and click java libries and edit and check in workspace default jres and apply.
4. Go to windows on main and perferences and click on maven and then user settings and check settings.xml path conffigured correctly. (c:apache maven/config/settings.xml)

Testing and Manual Testing

Testing
A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Manual Testing
•Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug.
•There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.
•Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing.

Automation Testing
•Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process.
•Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly

Software Testing Life Cycle (STLC) *is the testing process which is executed in systematic and planned manner.*

Requirement Analysis
Test Planning
Test Case Development
Environment Setup
Test Execution
Test Cycle Closure

Quality Assurance
QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements
Process-oriented activities.
Preventive activities.
It is a subset of Software Test Life Cycle (STLC).

Quality Control
It includes activities that ensure the verification of a developed software with respect to documented (or not in some cases) requirements.
Product-oriented activities.
It is a corrective process
QC can be considered as the subset of Quality Assurance.

Sanity Testing: Testing technique which determines if a new build is performing well enough to accept it for a major testing effort. It is performed by the testing teams.

Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing
It is called 'build verfication' or high level testing.

Acceptance Testing: Formal testing conducted to determine whether or not a software application satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the software.

Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units.It is often done by programmer by using sample input and observing its corresponding outputs.

Integration Testing

Integration testing is testing in which a group of components/modules are combined to produce output.

Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system.It is used to test speed and effectiveness of program.

Blackbox Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

Whitebox Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Testing:
It involves identifying bug/error/defect in a software without correcting it. Testing is performed in the testing phase.

Debugging:
It involves identifying, isolating, and fixing the problems/bugs. Developers who code the software conduct debugging upon encountering an error in the code. Debugging is a part of White Box Testing or Unit Testing. Debugging can be performed in the development phase.

Security Testing
Security testing involves testing a software in order to identify any flaws and gaps from security and vulnerability point of view.
Listed below are the main aspects that security testing should ensure:
Confidentiality
Integrity
Authentication
Authorization

Traceability Matrix Traceability Matrix (also known as Requirement Traceability Matrix - RTM) is a table that is used to trace the requirements during the Software Development Life Cycle.
It can be used for forward tracing (i.e. from Requirements to Design or Coding) or backward (i.e. from Coding to Requirements).
There are many user-defined templates for RTM.

How to write Test Case
Test cases involve a set of steps, conditions, and inputs that can be used while performing testing tasks.
The main intent of this activity is to ensure whether a software passes or fails in terms of its functionality and other aspects.
There are many types of test cases such as functional, negative, error, logical test cases, physical test cases, UI test cases, etc.
Test case ID
Product module.
Product version
Purpose
Pre-conditions
Steps
Expected outcome.
Actual outcome
Post-conditions

Why should Selenium be selected as a test tool?

Selenium

1.is a free and open source

2.have a large user base and helping communities

3.have cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)

4.have great platform compatibility (Windows, Mac OS, Linux etc.)

5.supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)

6.has fresh and regular repository developments

7.supports distributed testing

What is Automation Testing?

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves the use of a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn't require any manual intervention.

What are the benefits of Automation Testing?

Benefits of Automation testing are:

Supports execution of repeated test cases

Aids in testing a large test matrix

Enables parallel execution

Encourages unattended execution

Improves accuracy thereby reducing human-generated errors

Saves time and money

Agile

Agile:
Agile is a framework of approaches and behaviors that encourage “just-in-time” production that enables customers to receive quality software sooner.

Differences between Agile and traditional project management (Waterfall)?
Agile encourages that a little of everything, including design, development, and testing is done at the same time. Conversely, the traditional approach to projects closes and completes one phase before the next begins. Agile encourages short, frequent feedback loops and embraces changes to requirements. In Waterfall, feedback is usually not collected until the very end of the project, and changes are discouraged.

Scrum prescribes only three roles: the Product Owner, Scrum Master, and the Delivery Team.

.What is the Daily Stand-Up?
the team meets for no more than 15 minutes to answer three questions:
What did you do yesterday?
What do you plan on doing today?
Are there any blocks or impediments that keep you from doing your work?

Describe what happens in the Sprint planning meeting.
In Sprint planning, the Product Owner presents the goal of the sprint and discusses the high priority product backlog items. The Delivery team then chooses the amount of work for the next sprint.

Is there a difference between Agile and Scrum?
Agile is the broader umbrella which Scrum falls under. Agile has four main values and twelve principles. Scrum has its own set of values and principles and provides a lightweight “framework” to help teams become Agile.

Scrum is a type of iterative model only but it is iterative + incremental.

There are 3 major ceremonies performed in Scrum:-
Planning Meeting – Where the entire scrum teams along with the scrum master and product owner meets and discuss each item from the product backlog that they can work on the sprint. When the story is estimated and is well understood by the team, the story then moves into the Sprint Backlog.
Review Meeting – Where the scrum team demonstrates their work done to the stake holders
Retrospective meeting – Where the scrum teams along with the scrum master and product owner meets and retrospect the last sprint they worked on. They majorly discuss about 3 things:
What went well?
What could be done better?
Action Items

There are two artifacts maintained in Scrum:
•Product Backlog – Containing the prioritized list of business requirements
•Sprint Backlog – Contains the user stories to be done by the scrum team for a sprint.

What is the "time Boxing" of a scrum process called?
It's called "Sprint"

What should be an ideal duration of a sprint?
It is recommended to have 2 – 4 weeks of sprint cycle.

The user stories are defined in the format of
As a <User / type of user>
I want to <action / feature to implement>
So that < objective>

Scrum Master – Acts as a servant Leader for the scrum team. He presides over all the scrum ceremonies and coaches the team to understand and implement scrum values and principals.
Product Owner – Is the Point of contact for a scrum team. He/she is the one who work closest to the business. The main responsibility of a product owner is to identify and refine the product backlog items

Velocity is the sum of story points that a scrum team completes (meets the definition of done) over a sprint.

Burn down chart is a graph which shows the estimated v/s actual effort of the scrum tasks.

During **Sprint review** we walkthrough and demonstrate the feature or story implemented by the scrum team to the stake holders.

During **retrospective**, we try to identify in a collaborative way what went well, what could be done better and action items to have continuous improvement.

A **Story point** is calculated by taking into the consideration the development effort+ testing effort + resolving dependencies and other factors that would require to complete a story.

In scrum, user stories are short, one sentence definitions of a feature or functionality.

Scrum project is developed in a series of "sprint". It is a repeatable and regular work cycle in scrum methodology during which work is accomplished and kept ready for review.

Each feature in scrum is Story. Story point is an arbitrary measure used by Scrum teams, and it is a metric used by agile teams to determine the difficulty of implementing a given story.

Sprint Member **Capacity** Planning allows members and teams to capture detailed hourly **capacity** targets so they can define the amount of workload on a **sprint**-by-**sprint** basis, and understand when they have exceeded their hourly **capacity**
 Fibonacci Sequence (1, 2, 3, 5, 8, 13, 21, 34, etc.)
.

Database and SQL Commands

A Database

A database is a collection of data that is organized in a manner that facilitates ease of access, as well as efficient management and updating.

A database is made up of tables that store relevant information.

A table stores and displays data in a structured format consisting of columns and rows

A primary key is a field in the table that uniquely identifies the table records.

SQL is case **insensitive**.

SQL stands for Structured Query Language.

SQL is used to access and manipulate a database.

Fully Qualified Names

In SQL, you can provide the table name prior to the column name, by separating them with a dot.

SELECT customers.City FROM customers

Order By

SQL can:

- insert, update, or delete records in a database.
- create new databases, table, stored procedures, views.
- retrieve data from a database, etc.

Commands	Description	Sytax
1. **SHOW DATABASE**	the SHOW DATABASES command lists the databases managed by the server.	SHOW DATABASES
2.**SHOW TABLES**	The SHOW TABLES command is used to display all of the tables in the currently selected MySQL database.	**SHOW TABLES**
3.**SHOW COLUMNS FROM TABLE**	SHOW COLUMNS displays information about the columns in a given table.	**SHOW** column_list **FROM** table_name
4.**SELECT Statement**	The SELECT statement is used to select data from a database.	**SELECT** column_list **FROM** table_name
5.**Multiple Queries**	SQL allows to run multiple queries or commands at the same time.	SELECT column_name1, column_name2 FROM table_name;
6.**The DISTINCT Keyword**	The SQL DISTINCT keyword is used in conjunction with SELECT to eliminate all duplicate records and return only unique ones.	SELECT DISTINCT column_name1, column_name2 FROM table_name;
7.**The LIMIT Keyword**	to retrieve just a subset of records	SELECT column list FROM table_name LIMIT [number of records];
Order By	ORDER BY is used with SELECT to sort the returned data.	SELECT * FROM customers ORDER BY FirstName;

Commands	Description	Sytax
The WHERE Statement	The WHERE clause is used to extract only those records that fulfill a specified criterion.	SELECT * FROM customers WHERE ID = 7;
The BETWEEN Operator	The BETWEEN operator selects values within a range. The first value must be lower bound and the second value, the upper bound.	SELECT * FROM customers WHERE ID BETWEEN 3 AND 7; (or) SELECT ID, FirstName, LastName, City FROM customers WHERE City = 'New York';
Logical Operators	Logical operators can be used to combine two Boolean values and return a result of true, false, or null.	WHERE Age >= 30 AND Age <= 40; (or) WHERE state = 'CA' OR city = 'Boston';
The IN Operator	The IN operator is used when you want to compare a column with more than one value.	WHERE City IN ('New York', 'Los Angeles', 'Chicago');
The NOT IN Operator	The NOT IN operator allows you to exclude a list of specific values from the result set.	WHERE City NOT IN ('New York', 'Los Angeles', 'Chicago');
The CONCAT Function	The CONCAT function is used to concatenate two or more text values and returns the concatenating string.	SELECT CONCAT(FirstName, ', ' , City) FROM customers;
The AS Keyword	A concatenation results in a new column. The default column name will be the CONCAT function.	SELECT CONCAT(FirstName,', ', City) AS new_column

Commands	Description	Sytax
Subqueries	A subquery is a query within another query.	SELECT FirstName, Salary FROM employees WHERE Salary > (SELECT AVG(Salary) FROM employees)
The Like Operator	The LIKE keyword is useful when specifying a search condition within your WHERE clause.	LIKE 'A%'; (or) LIKE '%s';
The MIN Function		SELECT MIN(Salary) AS Salary FROM employees;
Joining Tables	One of the most beneficial features of SQL is the ability to combine data from two or more tables.	
- INNER JOIN	INNER JOIN is equivalent to JOIN. It returns rows when there is a match between the tables. ON keyword for specifying the inner join condition.	SELECT column_name(s) FROM table1 INNER JOIN table2 ON table1.column_name=table2.column_name;
- LEFT JOIN	The LEFT JOIN returns all rows from the left table, even if there are no matches in the right table.	SELECT table1.column1, table2.column2... FROM table1 LEFT OUTER JOIN table2 ON table1.column_name = table2.column_name;
- RIGHT JOIN	The RIGHT JOIN returns all rows from the right table, even if there are no matches in the left table.	SELECT table1.column1, table2.column2... FROM table1 RIGHT OUTER JOIN table2 ON table1.column_name = table2.column_name;

Commands	Description	Sytax
Set Operation **UNION**	The UNION operator is used to combine the result-sets of two or more SELECT statements.	SELECT column_name(s) FROM table1 UNION SELECT column_name(s) FROM table2;
UNION ALL	UNION ALL selects all rows from each table and combines them into a single table.	UNION ALL selects all rows from each table and combines them into a single table.
Inserting Data	SQL tables store data in rows, one row after another. The INSERT INTO statement is used to add new rows of data to a table in the database.	INSERT INTO table_name VALUES (value1, value2, value3,...);
Updating Data	The UPDATE statement allows us to alter data in the table.	UPDATE table_name SET column1=value1, column2=value2, ... WHERE condition;
Deleting Data	The DELETE statement is used to remove data from your table. DELETE queries work much like UPDATE queries.	DELETE FROM table_name WHERE condition;
Create Table	CREATE TABLE statement is used to create a new table.	CREATE TABLE Users (UserID int, FirstName varchar(100), LastName varchar(100), City varchar(100));
Dropping	The following SQL code demonstrates how to delete the column named DateOfBirth in the People table.	ALTER TABLE People DROP COLUMN DateOfBirth;

Commands	Description	Sytax
Renaming	The ALTER TABLE command is also used to rename columns:	ALTER TABLE People CHANGE FirstName name varchar(100);
VIEW	a VIEW is a virtual table that is based on the result-set of an SQL statement. A view contains rows and columns, just like a real table.	CREATE VIEW view_name AS SELECT column_name(s) FROM table_name WHERE condition;
Creating Views	Let's create a view that displays each employee's FirstName and Salary.	CREATE VIEW List AS SELECT FirstName, Salary FROM Employees;
Updating a View	You can update a view by using the following syntax:	CREATE OR REPLACE VIEW view_name AS SELECT column_name(s) FROM table_name WHERE condition;

Assume one employee table , in this table name, salary and role column is there. We have to retrieve salary by ascending order and group the role. Write the SQL query

A **trigger** is a type of stored procedure, but it runs based off of an event on a table instead of just being a set of instructions to be executed repeatedly.
A trigger is defined to activate when an INSERT, DELETE, or UPDATE statement executes for the associated table.

A **stored procedure** is a group of Transact-SQL statements compiled into a single execution plan.

GIT :

GIT

Version control systems consist of a central shared repository where teammates can commit changes to a file or set of file. .

Version control allows you to:

- Revert files back to a previous state.
- Revert the entire project back to a previous state.
- Compare changes over time.
- See who last modified something that might be causing a problem. Who introduced an issue and when.

With Version Control System (VCS), all the team members are allowed to work freely on any file at any time. VCS will later allow you to merge all the changes into a common version.

Advantages of Git
Free and open source
Fast and small
Implicit backup
Security
No need of powerful hardware
Easier branching

Git is a **Distributed Version Control** system (DVCS). It can track changes to a file and allows you to revert back to any particular change.

Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN one major advantage is that it does not rely on a central server to store all the versions of a project's files. Instead, every developer "clones" a copy of a repository I have shown in the diagram below with "Local repository" and has the full history of the project on his hard drive so that when there is a server outage, all you need for recovery is one of your teammate's local Git repository.

There is a central cloud repository as well where developers can commit changes and share it with other teammates as you can see in the diagram where all collaborators are commiting changes "Remote repository".

GIT WorkFlow:

Config – init – add – commit – add repo url – push

Description	GIT Commands
Configure user name / user email	git config --global user.name "username“ /// user.email “usermail”
Initating git in particular folder	git init
Clone operation creates the instance of the repository.	git clone “url”
Add the files to staging environment	git add . // git add filename
"records changes to the repository"	git commit -m "any message”
Create connection to repository	git remote add anyname url_of_empty_repository
operation copies changes from a local repository instance to a remote one.	git push anyname master
copies the changes from a remote repository instance to a local one.	git pull origin master
Branches are used to create another line of development a branch is created to work on a new feature	git checkout branch branchname
Switiching into another branch or master	Git checkout branchname
Taking the changes from one branch and adding them into another (traditionally master) branch.	git merge branchname1 branchname2
Reset to particular commit Head is updated with this	git reset commitid
Revert to previous commit	git revert commitid
Shows difference	git diff
Deleting branch	git branch -d branchname1
To check status	git status
To see log	git log
Operation combines merge and checkpout	git rebase branchname1 branchname2
Tags are used to define which portions of a project’s Git history is most important	git tag -a 'Release_1_0' -m 'Tagged basic string operation code'
The git fetch command imports commits from a remote repository into your local repo	Git fetch