

PROGRAM 7

Write the command to create a light weight git tag named "V1.0" for a commit in your local repository.

STEP 1: `git tag V1.0 <commit_id>`:

* This command creates a tag named "V1.0" and associates it with the specified commit ID. This tag marks a specific point in your project's history, often representing a release version.

STEP 2: `git show V1.0`:

* This command displays detailed information about the commit tagged as "V1.0". It shows the commit message, author, date, and the changes introduced in that commit.

STEP 3: `git tag V1.1 <commit_id>`:

* This command creates another tag named "V1.1" and associates it with a different commit ID. This might represent a subsequent release or a major update.

STEP 4: `git show V1.1`:

* Similar to the previous command, this displays detailed information about the commit tagged as "V1.1". Why Use Git Tags?

- * Release Management: Tags help identify specific releases and their corresponding code.
- * Historical Reference: You can easily reference past versions of your project.
- * Code Review: Tags can be used to mark specific points for review or comparison.
- * Rollback: In case of issues, you can revert to a tagged version. Additional Notes:
- * Lightweight Tags: The commands above create lightweight tags, which are simply pointers to commits.
- * Annotated Tags: You can create annotated tags using the `-a` flag, which allows you to add additional metadata, such as a tagger, date, and a message.
- * Tagging the Latest Commit: To tag the latest commit, you can use the HEAD reference:
`git tag V1.0 HEAD`