**SED – 15 Practical Use Cases (With Detailed Explanations)**

**🔧 sed stands for Stream Editor**

It's a powerful command-line tool used for:

* 🔄 Text transformation
* 🧹 Editing files automatically
* 🧼 Removing or replacing text
* 📄 Modifying content in files without opening them in an editor

**🧠 Think of sed as:**

**🔽** A tool that reads a file line-by-line and lets you make automated edits using patterns and commands.

**✅ Key Features of sed:**

| **Feature** | **Example Use** |
| --- | --- |
| **🔁 Replace text** | **Change "v1" to "v2" in code** |
| **🧽 Delete lines** | **Remove unwanted entries** |
| **➕ Insert/Append lines** | **Add headers or notes** |
| **✂ Strip or trim content** | **Remove white spaces or extra columns** |
| **💾 In-place file editing** | **Automate fixes in shell scripts/configs** |
| **🔍 Regex support** | **Pattern matching with precision** |

**🧪 Sample File: data.txt**

1 John Manager 5000 Active

2 Alice Developer 4200 Inactive

3 Bob Tester 3000 Active

4 Charlie Manager 5200 Active

5 David Developer 4500 Inactive

**✅ 1. Replace a word globally in all lines**

sed 's/Developer/Engineer/g' data.txt

📤 Output:

1 John Manager 5000 Active

2 Alice Engineer 4200 Inactive

...

🔍 Explanation:

* s: Substitution command.
* g: Global replacement (replaces all occurrences per line).
* No -i means output is shown, file is not modified.

🧠 Use Case: Updating job titles, terms, or labels across records.

**✅ 2. Replace only the first occurrence in each line**

sed 's/Manager/Lead/' data.txt

🧠 Replaces only the first Manager per line. Default behavior of sed.

**✅ 3. Replace on specific line (e.g., line 2 only)**

sed '2s/Developer/Engineer/' data.txt

📤 Output (Line 2 updated only):

2 Alice Engineer 4200 Inactive

🧠 Useful when you want targeted edits in automation scripts or config updates.

**✅ 4. Replace and modify the file directly**

sed -i 's/Inactive/Offline/g' data.txt

🔍 Explanation:

* -i: Edits the file **in-place**.
* No output to terminal – it silently updates the file.

🧠 Used in deployment scripts or automated corrections.

**✅ 5. Delete specific line**

sed '3d' data.txt

📤 Output: Line 3 removed

🧠 Great for cleaning corrupted entries or skipping headers/footers.

**✅ 6. Delete a range of lines (2 to 4)**

sed '2,4d' data.txt

🧠 Removes multiple lines. Used in logs, batch processing.

**✅ 7. Delete all blank lines**

🧪 Sample file with blanks:

John

Alice

Bob

bash

sed '/^$/d' file.txt

🧠 ^$ matches empty lines.

**✅ 8. Print only specific lines (lines 2–4)**

sed -n '2,4p' data.txt

🔍 Explanation:

* -n: Suppresses automatic output.
* p: Prints only selected lines.

🧠 Ideal for previewing log segments.

**✅ 9. Append a line after a specific line**

sed '3a\New employee joined' data.txt

📤 Output (Line 3 gets new line appended below it)

🧠 Used in dynamic changelogs, summaries.

**✅ 10. Insert a line before a specific line**

sed '2i\--- New Entry ---' data.txt

🧠 Adds headers, labels, or dividers before data blocks.

**✅ 11. Replace only if a line matches a pattern**

sed '/Charlie/s/Manager/Lead/' data.txt

🧠 Only replaces "Manager" on the line where "Charlie" is found.

**✅ 12. Remove everything after the 3rd word**

sed 's/\(\([^ ]\* \)\{3\}\).\*/\1/' data.txt

📤 Output:

1 John Manager

2 Alice Developer

...

🧠 Removes salary and status fields. Useful for cleanup or summarization.

**✅ 13. Remove first word (ID) from each line**

sed 's/^[^ ]\* //' data.txt

📤 Output:

John Manager 5000 Active

Alice Developer 4200 Inactive

...

🧠 Helps when unique IDs are no longer needed or for anonymization.

**✅ 14. Change lowercase to uppercase**

sed 's/.\*/\U&/' data.txt

📤 Output:

1 JOHN MANAGER 5000 ACTIVE

...

🧠 \U tells sed to convert the whole line (&) to uppercase.

**✅ 15. Find and replace using a variable in a script**

name="Alice"

sed "s/$name/Alicia/" data.txt

🧠 When using sed in scripts, always use double quotes if you're interpolating variables.

**💡 Bonus: Combine multiple sed commands**

sed -e 's/Developer/Dev/' -e 's/Inactive/Offline/' data.txt

🧠 Chain multiple changes in one pass—more efficient for large files.