Day 05

1)**Find the row of data which has the word "Engineering"**

Command:

grep "Engineering" data.csv

Output:

{ ~ } » grep "Engineering" data.csv

102,Bob,25,50000,Engineering

105,Eve,28,60000,Engineering

108,Hank,32,68000,Engineering

Explanation of Command:

grep => A command-line tool used to search for a pattern in a file.

"Engineering" => The pattern (keyword) to search for.

data.csv => The file where grep will search for the pattern.

2)**find the number of columns in first row**

Command:

awk -F, '{print NF; exit}' data.csv

Output:

{ ~ } » awk -F, '{print NF; exit}' data.csv

5

Explanation of Command:

awk => A powerful text-processing command in Linux.

-F, => Specifies the field separator as a comma (,) (since it's a CSV file).

{print NF; exit} =>

NF => Number of fields (columns) in the current row.

print NF => Prints the number of columns in the row.

exit => Stops awk after processing the first row only (prevents reading the entire file).

3)find the number of columns in each row

Command:

awk -F ',' {print NF}'

Output:

{ ~ } » awk -F ',' '{print NF}' data.csv

5

5

5

5

5

5

5

5

5

5

5

Explanation of Command:

awk => A powerful text-processing command in Linux.

-F ',' => Specifies comma (,) as the field separator, which is necessary for CSV files.

{print NF} =>

NF (Number of Fields) => Represents the number of fields (columns) in each row.

print NF => Prints the number of fields for every row in the file.

4)**Sorting CSV by Salary in Reverse Order Using the sort Command**

Command:

sort -t',' -k4,4r data.csv

Output:

```
{ ~ }  » sort -t',' -k4,4r data.csv
ID,Name,Age,Salary,Department
104,David,40,90000,HR
103,Charlie,5,80000,Data Science
106,Frank,38,75000,HR
107,Grace,27,72000,Data Science
110,Jack,31,71000,HR
101,Alice,30,70000,Data Science
108,Hank,32,68000,Engineering
109,Ivy,29,62000,Data Science
105,Eve,28,60000,Engineering
102,Bob,25,50000,Engineering
```

Explanation of Command:

1. sort:

This is the command that sorts the contents of a file. It reads input, processes it,

and returns a sorted version of that input based on the options provided.

2. -t',':

The -t option specifies the delimiter to be used for separating fields (columns) in the file.

',' specifies that the delimiter is a comma, which is typical for CSV (Comma Separated Values) files.

3. -k4,4:

The -k option specifies which columns to sort by. It uses a column number to indicate where the sort should start and end.

The syntax is -kstart_column,end_column. For example, -k4,4 means:

start_column: Start sorting at the third column.

end_column: Stop sorting at the third column as well.

So -k4,4 tells sort to only sort based on the forth column.

r:

This option modifies the sort order:

r stands for reverse sorting. This means the sorting will be in descending order (largest to smallest).

5) **sort a file by the 4th column in numerical order.**

Command:

sort -t',' -k4,4n data.csv

Output:

{ ~ } » sort -t',' -k4,4n data.csv

ID,Name,Age,Salary,Department

102,Bob,25,50000,Engineering

105,Eve,28,60000,Engineering

109,Ivy,29,62000,Data Science

108,Hank,32,68000,Engineering

101,Alice,30,70000,Data Science

110,Jack,31,71000,HR

107,Grace,27,72000,Data Science

106,Frank,38,75000,HR

103,Charlie,5,80000,Data Science

104,David,40,90000,HR

Explanation of Command:

1.sort:

Sorts the contents of the file.

2.-t',':

Specifies that the columns are separated by commas (CSV format).

3.-k4,4:

Sorts by the 4th column (Salary).

4.-n:

Sorts numerically, from smallest to largest.

Note:

The sort command treats the first row (header) as a string, so it is sorted before numeric values.

This is why the header appears at the top when sorting.

6**) sort in reverse order based on the 4th column**

Command:

sort -t',' -k4,4 -r data.csv

Output:

sort -t',' -k4,4 -r data.csv

ID,Name,Age,Salary,Department

104,David,40,90000,HR

103,Charlie,5,80000,Data Science

106,Frank,38,75000,HR

107,Grace,27,72000,Data Science

110,Jack,31,71000,HR

101,Alice,30,70000,Data Science

108,Hank,32,68000,Engineering

109,Ivy,29,62000,Data Science

105,Eve,28,60000,Engineering

102,Bob,25,50000,Engineering

Explanation of Command:

1. sort:

This command sorts the rows in a file.

2. -t',':

Sets the comma , as the separator for columns (since it's a CSV file).

3. -k4,4:

Sorts the data based on the 4th column (which is Salary).

4. -r:

Sorts in reverse order (highest to lowest for numbers, or Z to A for text).

Note:

Without -n: If I don't use the -n flag, the sort will be lexicographical (alphabetical). This can cause problems when salaries have different lengths.

For example:

If I have "50000" (5 characters) and "9000" (4 characters), the command would compare them as text.

"50000" would come before "9000" because '5' comes before '9' alphabetically.

With -n: Using -n ensures that the sort is numeric. So, the command will correctly compare the values. For instance:

"50000" will correctly come after "9000" because 50000 is a higher number than 9000.

7) **sort a CSV file by the 4th column in descending numerical order**

Command:

sort -t',' -k4,4 -n -r data.csv

Output:

{ ~ } » sort -t',' -k4,4 -n -r data.csv

104,David,40,90000,HR

103,Charlie,5,80000,Data Science

106,Frank,38,75000,HR

107,Grace,27,72000,Data Science

110,Jack,31,71000,HR

101,Alice,30,70000,Data Science

108,Hank,32,68000,Engineering

109,Ivy,29,62000,Data Science

105,Eve,28,60000,Engineering

102,Bob,25,50000,Engineering

ID,Name,Age,Salary,Department

Explanation of Command:

1. sort:

This is the command used to sort the rows of a file.

2. -t',':

Specifies that the columns in the file are separated by commas (,), which is typical for CSV files.

3. -k4,4:

Sorts based on the 4th column only, which is Salary in this case.

4. -n:

Sorts numerically (instead of alphabetically), ensuring that numbers are compared based on their value (e.g., 90000 > 50000).

5. -r:

Sorts in reverse order (from highest to lowest).

Note:

The header (ID,Name,Age,...) is at the bottom because:

sort treats the header like a normal row.

It tries to compare the word "Salary" to the numbers like 90000, 80000

Since "Salary" is not a number, it's treated as lowest in a numeric sort.

8) **Multi-Level Reverse Sorting in CSV using sort Command**

Command:

sort -t',' -k5,5 -k2,2 -r data.csv

Output:

{ ~ } » sort -t',' -k5,5 -k2,2 -r data.csv

110,Jack,31,71000,HR

106,Frank,38,75000,HR

104,David,40,90000,HR

108,Hank,32,68000,Engineering

105,Eve,28,60000,Engineering

102,Bob,25,50000,Engineering

ID,Name,Age,Salary,Department

109,Ivy,29,62000,Data Science

107,Grace,27,72000,Data Science

103,Charlie,5,80000,Data Science

101,Alice,30,70000,Data Science

Explanation of Command:

1. sort:

This is the command used to sort the lines in a file.

2. -t',':

Tells sort that the columns in the file are separated by commas (,), which is common in CSV files.

3. -k5,5:

Sort the rows by the 5th column (which is the Department column).

4. -k2,2:

If two or more rows have the same department, then sort those rows by the 2nd column (Name).

5. -r:

Sort everything in reverse order (Z to A, or highest to lowest)