

CSV Analyzer – Line-by-Line Explanation

1. `import csv`
 - Imports Python's built-in CSV module for reading and writing CSV files.
2. `def load_csv(file_path):`
 - Defines a function to load a CSV file given its path.
 - Uses `csv.DictReader` to convert each row into a dictionary using the header row as keys.
3. `def filter_data(data, column_name, condition, value):`
 - Filters a list of dictionaries based on the specified condition (`equals`, `greater_than`, `less_than`).
 - Converts numeric strings to float when possible to allow numerical comparison.
4. `def display_data(data):`
 - Prints the filtered data in a table-like format.
 - Shows column headers and each row's values.
5. `def save_to_csv(data, output_path):`
 - (Optional) Writes filtered data back into a new CSV file.
 - Uses `csv.DictWriter` to preserve header structure.
6. `if __name__ == '__main__':`
 - Main script entry point. Runs only when script is executed directly.
 - Loads 'employees.csv', filters rows where Salary > 50000, displays them, and optionally saves output.
7. `print(...)` statements:
 - Provide feedback in terminal about loading, filtering, and saving data.

Common issues handled:

- File not found errors.
- Extra spaces in headers.
- String vs numeric comparison.