

## Task: Create a Service class that can parse and query CSV files using SQL like syntax and return the results in JSON format

- The Service class should implement the following interface:

```
public interface ICSVService
{
    // Returns a JSON string that contains an array of objects, where
    // each object represents a row of the CSV file
    // The keys are the column names and the values are the column
    // values
    // The query parameter is a string that follows the SQL-like syntax
    // For example: "SELECT name, age FROM users.csv WHERE age > 20"
    string Query(string query);
}
```

- The SQL-like syntax should support the following clauses:
  - SELECT**: specifies which columns to return. Use \* to return all columns.
  - FROM**: specifies the file name or full path of the CSV file to query. The file will have a header row with the column names.
  - WHERE**: specifies a condition to filter the rows. The condition can use the following operators: =, !=.
  - Optional**: implement support for , <=, >=, AND, OR, NOT. The condition can also use parentheses to group expressions. The condition can also use the wildcard % to match any string of zero or more characters. For example: "SELECT \* FROM products.csv WHERE name LIKE '%book%'"
  - Optional AS**: assigns an alias to a column or an expression, which can make the query more readable and concise. For example: "SELECT name AS full\_name, age \* 12 AS months FROM sample.csv"
  - Optional ORDER BY**: sorts the results by one or more columns in ascending or descending order. Use **ASC** or **DESC** to specify the order. For example: "SELECT \* FROM employees.csv ORDER BY salary DESC, name ASC"
- The Service class should have a dependency injection of an interface that handles the file operations, such as reading the CSV files. Implement the following interface:

```
public interface IFileService
{
    IEnumerable<string> ReadFileLines(string fileName);

    string ReadFile(string fileName);
}
```

- Optional**: Modify the Service class to automatically detect the separator used in the CSV file, such as comma (,), semicolon (;), colon (:), etc. The Service class should not require the user to specify the separator in the query or in the constructor. Instead, the Service class should automatically detect the separator by analyzing the first line of the file.