**Test task Zero LLM**

In the financial sector, one of the routine tasks is mapping data from various sources in Excel tables. For example, a company may have a target format for employee health insurance tables (Template) and may receive tables from other departments with essentially the same information, but with different column names, different value formats, and duplicate or irrelevant columns.

Your task is to devise an approach using LLM for mapping tables A and B to the template by transferring values and transforming values into the target format of the Template table (example below)

Example tables are attached as .**csv** files.

**Possible solution approach:**

• Extract information about the columns of the Template table and tables A and B in the format of a text description.

• For each of the candidate tables (A and B), ask the LLM to find similar columns in the Template.

• In case of ambiguous mapping, ask the user to choose the most suitable column from the candidates.

• Automatically generate data mapping code for each column display in the final Template format. For example, for date columns, they may be in different formats, and it is necessary to change the order from dd.mm.yyyy to mm.dd.yyyy. The person can check the code (or pseudocode) that will perform the mapping.

• Check that all data has been transferred correctly; if any errors occur, issue an alert to the person.

**There is also an additional challenge\* (Not required but would be a big plus):**

• Since such operations can be repeated quite often, and a person will edit the transformation logic, it is desirable to save this data and have the ability to retrain on it. Propose an approach for retraining and try to implement such retraining on synthetic examples (you can come up with them using GPT =))

**Expected result:**

• Code on GitHub using OpenAI API, Langchain, and other LLMOps tools of your choice.

• An interface in which you can perform such an operation. Place the interface in the public domain for testing work

Example user journey:

• The user uploads the Template table.

• The user uploads table A.

• For each column in the Template, the system suggests columns from column A (1 or more relevant candidates), showing the basis for the decision (formats, distributions, and other features that are highlighted in the backend).

• The person confirms the mapping.

• Next, the data transformation stage begins. The system generates and displays the code with which it will perform the transformation. The user can edit it and run it, checking the correctness of the mapping.

• At the output, the user receives a table in the Template format (columns, value formats) but with values from table A.

• The same for table B.

3. Your thoughts on edge cases and how they can be overcome