Benefit Cloud

Engineering Project Assignment

This document outlines a self-contained engineering project to be completed prior to our next meeting. The goal is to evaluate problem-solving ability, data reasoning, software structure, and clarity of thought.

You will build a small Python application that:

1. Generates an inbound HRIS data file.
2. Processes, validates, and normalizes the data.
3. Produces a clean outbound carrier file.

This project is intentionally open-ended in its implementation. You may use any libraries, tools, or patterns you feel appropriate — as long as you can clearly explain your decisions.

1. Inbound HRIS File (You Generate This)

Create a CSV file named `employees_inbound.csv` with approximately 200 synthetic rows.

The file must contain the following columns exactly:

- **employee_id**
- **first_name**
- **last_name**
- **dob**
- **ssn**

- **hire_date**
- **term_date**
- **plan_name**
- **coverage_level**
- **effective_date**

Important: All plan names in your generated data must originate from the list of 15 messy plan names provided below.

2. Allowed Incoming Plan Names (15 Messy Options)

Your inbound data must use only variations of the following 15 messy plan names:

1. VIS-BUYUP
2. Dental – BASIC
3. PPO–Med Plan
4. Med_PPO
5. fsa-variation-1
6. Premium Dental Coverage
7. VISION Core Option
8. Medical HDHP
9. Basic Dental Plan
10. FSA_VOYA
11. vision-buy-up
12. Dental – Premium Plan
13. HDHP – Medical
14. Dental Premium
15. PPO Medical
16. High Deductible Medical
17. Medical PPO Core
18. Dental Basic
19. MED PPO – Core

3. Final Clean Plan Names (6 Options)

Your application must map the messy input plan names to exactly one of the following clean output plan names, if possible:

1. Medical PPO
2. Medical HDHP
3. Dental Basic
4. Dental Premium
5. Vision Core
6. Vision Premium

4. Required Data Normalization

Your application must enforce the following data-cleaning rules:

SSN Normalization

The SSN field must be output as:

- exactly 9 digits,
- no dashes,
- no spaces.

Date Normalization

All date fields (`dob`, `hire_date`, `term_date`, `effective_date`) must be normalized to the format:

`YYYY-MM-DD`

Plan Name Mapping

Map each messy plan name into one of the 6 clean plan names listed above.

5. Outbound Carrier File Requirements

Produce a CSV file named `carrier_export.csv` with the following fields:

- **MemberID** (from `employee_id`)
- **Name** (first and last name concatenated)
- **DOB** (normalized)
- **SSN** (clean 9-digit format)
- **Plan** (clean plan name)
- **Coverage** (from `coverage_level`)
- **EffectiveDate** (normalized)

6. Deliverables

Please provide:

- A GitHub repository link or zipped project folder.
- The generated `employees_inbound.csv`.
- The resulting `carrier_export.csv`.
- A **README** describing:
  - Your plan-name mapping approach.
  - Your data-cleaning logic.
  - How you handled unmapped or invalid rows.
  - Any assumptions you made.

7. Follow-Up Discussion

At our meeting, be prepared to walk through:

- Your mapping logic and why you chose it.
- Your validation and normalization design.
- How your structure would scale if many more plan types or carriers were added.
- What you would improve with more time.

The objective is not perfection but clarity of reasoning, structure, and problem-solving.