Swiss Re

Disclaimer

Contents of this document are confidential and should be used for recruitment purposes only. Disclosing it (or any of its parts) to any third party, publishing on the internet, etc. is strictly forbidden.

Task Description

You receive requirements from your business peers which outline a new data transformation use case to be implemented.

Use **Python** as your language, store and transform your data using **Spark**. Your implementation should be delivered as a small code base which would pass a code review from your software engineering peers. The code review could check for example that:

- The code is **self-contained**: It is runnable and produces the required output format
- The code is **covered by tests**: Feel free to use any additional libraries you see fit
- The code follows production **coding best practices that you deem important**

If some requirements need clarification, feel free to make assumptions and explain your choices in the interview. (Note that some ambiguities and errors in the specs are deliberate.)

If you feel like something is too unclear to be able to implement at all then get back to us for clarification.

Requirements

We receive contracts and claims as an input from our "Europe 3" source and need to build transactions out of them that adhere to our common internal data model.

Input Data

Expected schema:

CONTRACT

Column Name	Type
(PK) SOURCE_SYSTEM	Text
(PK) CONTRACT_ID	Number
CONTRACT_TYPE	Text
INSURED_PERIOD_FROM	Date
INSURED_PERIOD_TO	Date
CREATION_DATE	Date

CLAIM

Column Name	Type
(PK) SOURCE_SYSTEM	Text
(PK) CLAIM_ID	Number
(PK) CONTRACT_SOURCE_SYSTEM	Text
(PK) CONTRACT_ID	Number
CLAIM_TYPE	Text
DATE_OF_LOSS	Date
AMOUNT	Decimal (16, 5)
CREATION_DATE	Date

* Rows marked with (PK) depict composite primary keys

Swiss Re

Data extract which can be used to base the implementation on:

CONTRACT

```
SOURCE_SYSTEM,CONTRACT_ID,"CONTRACT_TYPE",INSURED_PERIOD_FROM,INSUDRED_PERIOD_TO,CREATION_DATE
Contract_SR_Europa_3,408124123,"Direct",01.01.2015,01.01.2099,17.01.2022 13:42
Contract_SR_Europa_3,46784575,"Direct",01.01.2015,01.01.2099,17.01.2022 13:42
Contract_SR_Europa_3,97563756,"",01.01.2015,01.01.2099,17.01.2022 13:42
Contract_SR_Europa_3,13767503,"Reinsurance",01.01.2015,01.01.2099,17.01.2022 13:42
Contract_SR_Europa_3,656948536,"",01.01.2015,01.01.2099,17.01.2022 13:42
```

CLAIM

```
SOURCE_SYSTEM,CLAIM_ID,CONTRACT_SOURCE_SYSTEM,CONTRAT_ID,CLAIM_TYPE,DATE_OF_LOSS,AMOUNT,CREATION_DATE
Claim_SR_Europa_3,CL_68545123,Contract_SR_Europa_3,97563756,2,14.02.2021,523.21,17.01.2022 14:45
Claim_SR_Europa_3,CL_962234,Contract_SR_Europa_4,408124123,1,30.01.2021,52369.0,17.01.2022 14:46
Claim_SR_Europa_3,CL_895168,Contract_SR_Europa_3,13767503,,02.09.2020,98465,17.01.2022 14:45
Claim_SR_Europa_3,CX_12066501,Contract_SR_Europa_3,656948536,2,04.01.2022,9000,17.01.2022 14:45
Claim_SR_Europa_3,RX_9845163,Contract_SR_Europa_3,656948536,2,04.06.2015,11000,17.01.2022 14:45
Claim_SR_Europa_3,CL_39904634,Contract_SR_Europa_3,656948536,2,04.11.2020,11000,17.01.2022 14:46
Claim_SR_Europa_3,U_7065313,Contract_SR_Europa_3,46589516,1,29.09.2021,11000,17.01.2022 14:46
```

Transformation

Mapping to create TRANSACTIONS:

Target Column	**Mapping from Input Columns**
CONTRACT_SOURCE_SYSTEM	Default to “Europe 3”
CONTRACT_SOURCE_SYSTEM_ID	CONTRACT.CONTRACT_ID
SOURCE_SYSTEM_ID	CLAIM_ID without prefix (e.g. A_123 --> 123)
TRANSACTION_TYPE	"Corporate" if CLAIM_TYPE = 2 "Private" if CLAIM_TYPE = 1 "Unknown" if CLAIM_TYPE is empty
TRANSACTION_DIRECTION	"COINSURANCE" when CLAIM_ID contains "CL" prefix "REINSURANCE" when CLAIM_ID contains "RX" prefix
CONFORMED_VALUE	CLAIM.AMOUNT
BUSINESS_DATE	CLAIM.DATEOFLOSS mapped to format YYYY-MM-DD
CREATION_DATE	CLAIM.CREATION_DATE mapped to format YYYY-MM-DD HH:mm:ss
SYSTEM_TIMESTAMP	System timestamp of creating the transaction
NSE_ID	Map CLAIM_ID to a unique id via rest API call for every input row: https://api.hashify.net/hash/md4/hex?value= CLAIM_ID (Returns a JSON, take field "Digest" as result)

Expected Output

Save the output of the transformation in a file called TRANSACTIONS. Choose a reasonable file format that you think makes sense.

Swiss Re

Output schema provided by the data architect:

```
{
    "fields": [
        {
            "metadata": { "primary_key": true },
            "name": "CONTRACT_SOURCE_SYSTEM",
            "nullable": true,
            "type": "string"
        },
        {
            "metadata": { "primary_key": true },
            "name": "CONTRACT_SOURCE_SYSTEM_ID",
            "nullable": true,
            "type": "long"
        },
        {
            "name": "SOURCE_SYSTEM_ID",
            "nullable": true,
            "type": "integer"
        },
        {
            "name": "TRANSACTION_TYPE",
            "nullable": false,
            "type": "string"
        },
        {
            "name": "TRANSACTION_DIRECTION",
            "nullable": true,
            "type": "string"
        },
        {
            "name": "CONFORMED_VALUE",
            "nullable": true,
            "type": "decimal(16, 5)"
        },
        {
            "name": "BUSINESS_DATE",
            "nullable": true,
            "type": "date"
        },
        {
            "name": "CREATION_DATE",
            "nullable": true,
            "type": "timestamp"
        },
        {
            "name": "SYSTEM_TIMESTAMP",
            "nullable": true,
            "type": "timestamp"
        },
        {
            "metadata": { "primary_key": true },
            "name": "NSE_ID",
            "nullable": false,
            "type": "string"
        }
    ]
}
```

Conceptual Questions

Create a proposal on how to handle new batches of input data, so that existing transactions created in previous batches are not overridden. You don’t need to provide an implementation, just write down a short proposal which can be discussed at the interview.