

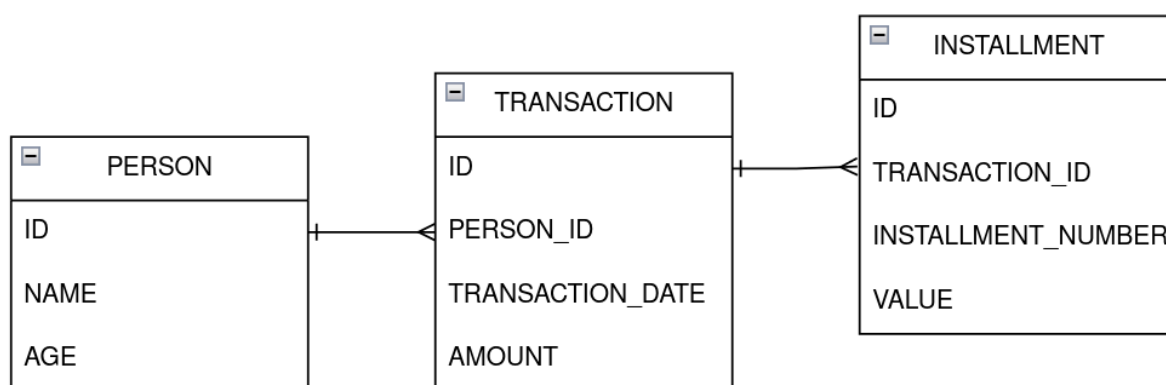
## Nível 1 - Integração entre camadas

O candidato receberá um arquivo input-data.csv, separado por ponto e vírgula (;) com muitas linhas no seguinte formato:

| ID da Transação | Data da Transação | Documento | Nome | Idade | Valor | Num. de Parcelas |

É exigido que o candidato codifique um programa capaz de ler este arquivo e publicar, para cada linha do arquivo, uma mensagem em formato JSON de acordo com um critério de parse.

Um outro programa deve ser construído para consumir as mensagens publicadas e persistir os dados em um modelo relacional com as seguintes tabelas e relacionamentos:



Regra de processamento:

- Cada linha do arquivo representa uma transação financeira feita por uma pessoa. O ID da tabela PERSON deve ser o documento da pessoa. Caso uma pessoa com aquele documento não exista no banco de dados, um registro na tabela PERSON deve ser criado.
- O campo “Valor” que vem na linha do arquivo representa o valor total da transação. Cada linha do arquivo deve ser inserida na tabela TRANSACTION, onde:
  - ID: UUID da transação que vem na primeira posição do arquivo
  - PERSON\_ID: o ID da tabela PERSON
  - AMOUNT: o valor que vem na linha do arquivo
  - TRANSACTION\_DATE: data da transação
- O campo “Num. de Parcelas” define um critério de divisão do campo “Valor”. Por exemplo, se o “Num. de Parcelas” foi igual a 3 e o “Valor” for igual a 300, o programa deverá criar três parcelas (parcela 1, parcela 2 e parcela 3) que devem ser inseridas na tabela INSTALLMENT, onde:
  - ID: um UUID gerado pelo programa (utilize uma biblioteca de geração de UUID para facilitar)
  - TRANSACTION\_ID: ID da transação na tabela TRANSACTION
  - INSTALLMENT\_NUMBER: número da parcela
  - VALUE: valor da parcela

## Nível 2 - Evolução do negócio

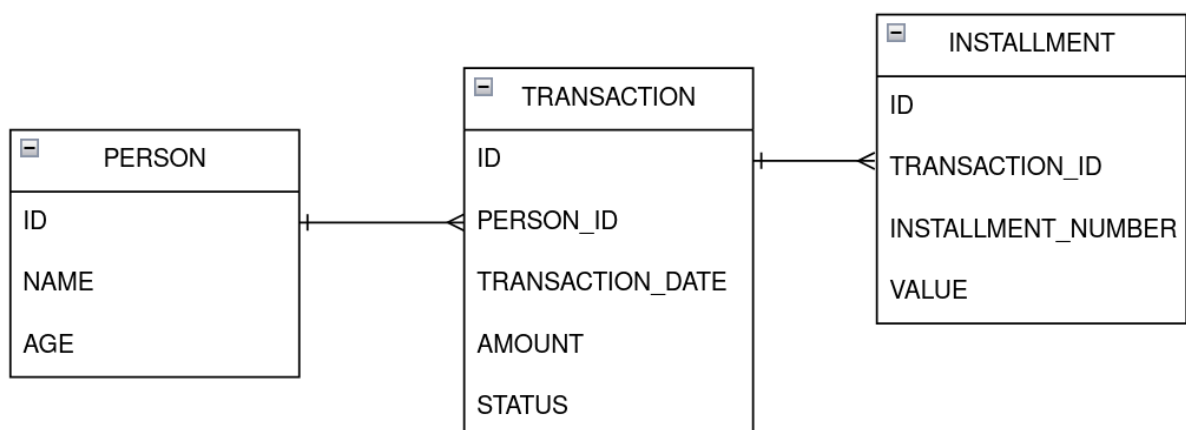
A área de negócio atualizou os requisitos e vamos ter que fazer algumas alterações no processo e evoluir o sistema.

O arquivo de transações input-data.csv processado é montado em um momento onde há uma possibilidade de que algumas dessas transações não sejam válidas. A empresa responsável por nos passar este arquivo (input-data.csv) disse que, em algum determinado momento do dia enviará um segundo arquivo em formato csv chamado conciliation-data.csv, com os identificadores das transações que chegaram no primeiro arquivo e um status de confirmação ou negativa de cada uma dessas transações. O arquivo virá neste formato:

| ID da Transação | Data da Transação | Documento | Status |

Regras de processamento:

- O campo do arquivo de nome Status é um campo char(1) que pode vir com os seguintes valores no arquivo:
  - C: Processamento confirmado
  - N: Processamento negado
- A coluna STATUS da tabela TRANSACTION deve armazenar o retorno do campo Status que vem no arquivo. A coluna deve ser NOT NULL. **Essa alteração deve ser feita com dados na tabela, como se já houvessem muitas transações já inseridas no sistema e esse requisito chegasse após o sistema já estar em uso.**
- O programa que processa o primeiro arquivo deve ser alterado, passando a inserir todas as transações com STATUS pendente (defina a melhor maneira para fazer isso).
- O programa que processará o segundo arquivo deverá modificar o valor de cada transação registrada na tabela TRANSACTION de acordo com o que chega no campo Status do arquivo (defina a melhor maneira para fazer isso).



### Nível 3 - Aplicação 24/7

Essa aplicação deve ficar em execução o tempo inteiro. Transforme este programa em uma aplicação que fica sempre em execução. Faça com que ela observe diretórios na sua máquina (ex: {HOME\_DIR}/hackathon-asapcard/INPUT e {HOME\_DIR}/hackathon-asapcard/CONCILIATION). Sempre que um arquivo for transferido para esse diretório, ele inicia o processamento.

### Nível 4 - Criação de Container

Executar todas as aplicações com Docker.