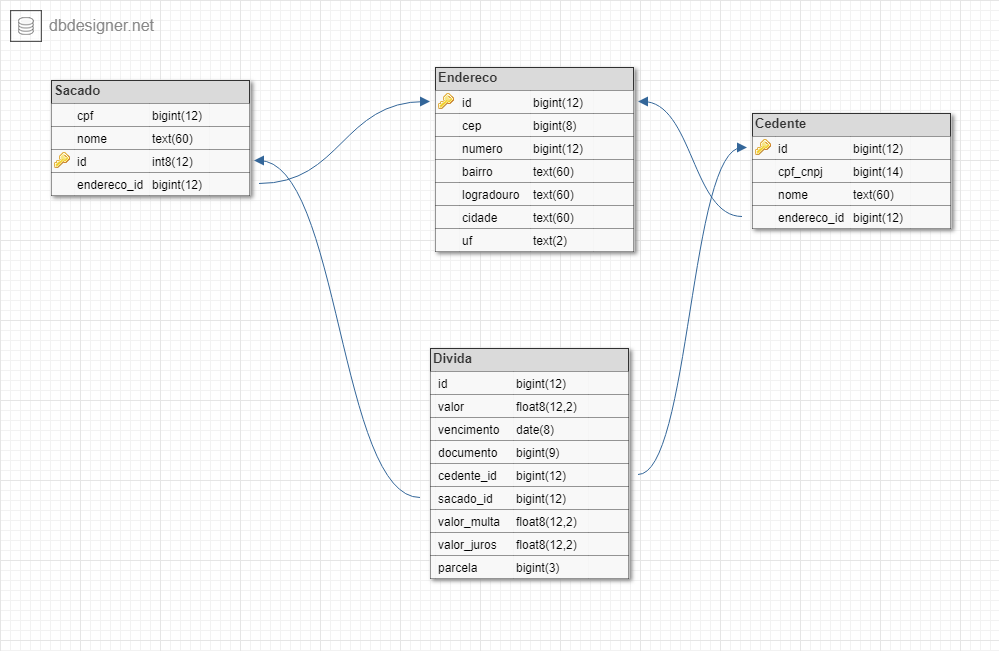
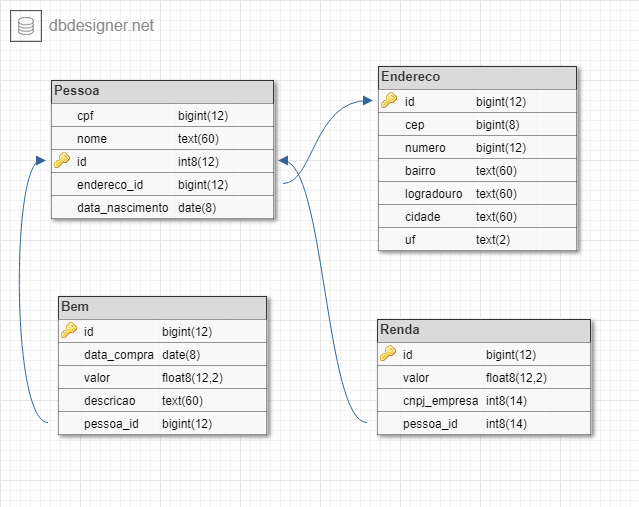
**Solução Consulta de Crédito por CPF**

Para a base A que é sensível e protegida e não tão performática eu escolhi o PostgreSQL como base de dados, as informações conforme abaixo:



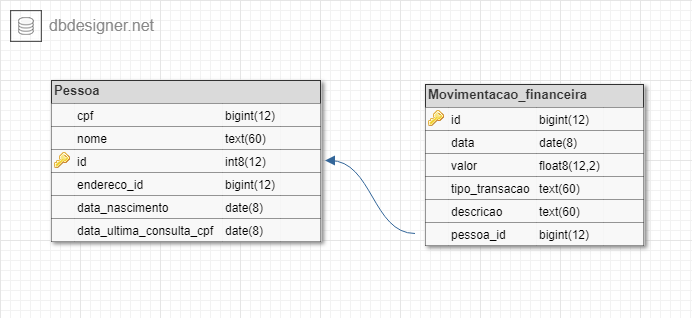
Para implementar a API para servir as informações eu escolhi utilizar o Python através do framework Django REST retornando as informações da pessoa como sacado e suas dívidas com as empresas através do cedente. Também o endereço desta pessoa e da empresa.

Para a base B que é sensível e protegida, porém mais rápida e permite a extração de dados com machine learning eu escolhi o MongoDB como base de dados, as informações conforme abaixo:



Para implementar a API para servir as informações eu escolhi utilizar o Python através do framework Django REST retornando as informações da pessoa com o endereço, a lista de bens e as fontes de renda.

Para a base C que é em dados críticos e acesso extremamente rápido eu escolhi o ElasticSearch como base de dados, as informações conforme abaixo:



Para implementar a API para servir as informações eu escolhi utilizar o Python através do framework Django REST retornando as informações da pessoa com a movimentação financeira, permitindo extrair pelo tipo de transação se é cartão de crédito e pela data buscar a última movimentação.

Definida as bases de dados e os meios de consulta através de API Django REST o CPF vai permitir conectar as três bases de dados e as informações que nelas contém e todas estas bases e APIs devem estar no AWS.

Para a interface de consulta eu escolhi usar o Flask para consumir os serviços e usar o Bootstrap como tema para consultar as informações da base C.

O prazo que foi me passado foram 48 horas, porém úteis mesmo eu tenho mais ou menos 8 horas, acredito que em 8 horas o que posso fazer é analisar as informações solicitadas e retornar o que eu poderia fazer para armazenar e prover as informações com as tecnologias utilizadas por este motivo não escrevi uma só linha de código, este é o próximo após definir a solução.