

|  |
| --- |
| Case para projeto final do Soul Code  Data Migration SQL to PySpark |

|  |
| --- |
| Document title | Introduction |

Case Sugerido



A proposta

Migração e conversão de dados é parte do nosso dia a dia. Mas nem sempre temos o luxo de escolher qual o formato de origem ou o de destino.

Nessa nota, gostariamos de sugerir um cenário baseado em um case real que envolve um cliente ter um ambiente transacional em produção e precisa ingestar esse ambiente em um Data Lake.

# O ambiente de origem

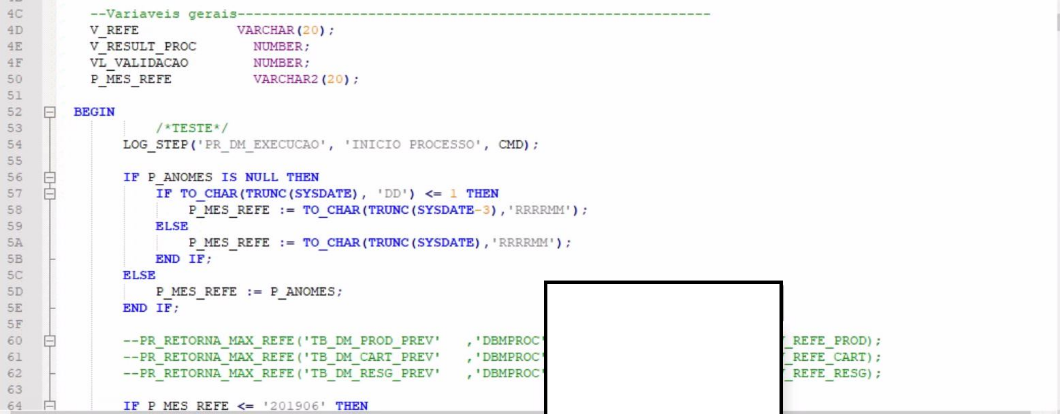
O ambiente de origem é um grupo 5 fontes(Tabelas) existentes em um banco relacional. No cliente original o banco é um Oracle, mas um Postgre funcionará igual. Usando colunas nessas 5 tabelas, o cliente tem um PL/SQL escrito com 4 procedures dentro.

Cada procedure cria uma tabela temporária que ao final do codigo é consumida para criar o Data mart em Oracle. Essas procedures não são tão elementares quanto copiar colunas. Muitas delas usam lógicas de negócio para formar suas tabelas temporárias. Ex: determinados valores estarem acima ou abaixo de um dado parâmetro, se tem certa condição encontrada... vários Ifs e JOINs são utilizados aqui.

Também é importante aqui que trabalhos de normalização dos campos das tabelas sejam realizados. Ex: padronização de formato de String para Numerico, checar e corrigir valores que estão em maiuscula ou minuscula... etc, para que as colunas possam ser trabalhadas ao final.

Uma vez formada essa tabela temporária, a parte MAIN do PL/SQL trabalha informações das 4 tabelas temporárias geradas pelas procedures e cria um Data Mart baseado no resultado.

Exemplo:



|  |
| --- |
| Sonhando com um código comentado |

### Resolvendo o cenário

O objetivo principal é criar uma ráplica (possivelmente automatizada, mas não sei se isso fará parte do escopo) do ambiente produtivo só que em uma estrutura de Data lake.

Os dados de origem devem ser ingestados no BUCKET de nuvem idealmente em formato Parquet, ORC ou Avro, mas caso esteja muito avançado no escopo, pode ser um CSV.

Os alunos devem construir um código Pyspark que se conecte ao BUCKET onde os dados estão. Em seguinda, remontar o mesmo algoritmo visto em SQL só que usando PySpark. Ele deve entregar como Output uma tabela que apresente os mesmo valores que estejam na tabela criada pelo ETL de origem.

Alunos devem explicar qual o caminho que tomaram, e porque decidiram por aquela escolha.

Idealmente o codigo PySpark deve levar em consideração performance.

Desafio

- Carregue seus dados em um DB MySQL, utilize triggers e procedures para realizar a normalização

- Crie um código em python que leia a DB MySQL trate os dados e insira em uma Lake Cassandra em nuvem

- Utilize Spark para ler a DB Cassandra e gerar seus resultados

- Utilize formato parquet onde aplicável

- Os códigos em python devem utilizar o paradigma orientado a objetos

- MySQL ou Postgres

|  |
| --- |
|  |