**Processo Seletivo "Data Engineer"**

**Etapa Desafio Prático**

GITHub: https://github.com/erysonc/PD

**1ª Parte**

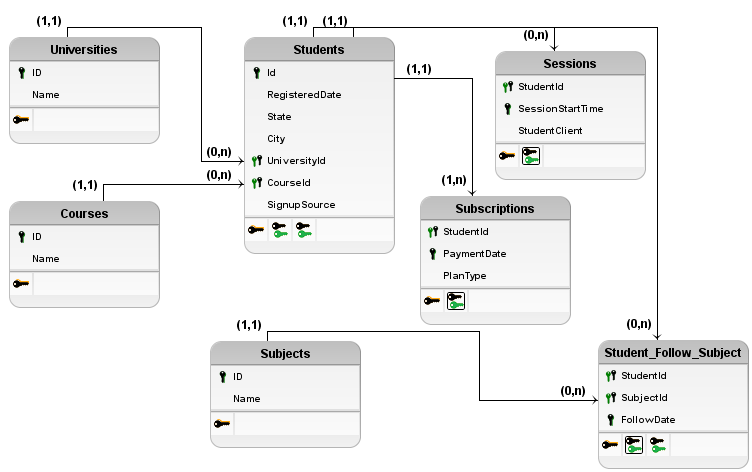
**Situação problema:**

1. Definir a modelagem de uma base analítica para que seja otimizada a análise do perfil de nossos usuários e suas segmentações relevantes.
2. Incluir os dados que você julgar importantes
3. Definir um pipeline dos dados. Estamos fornecendo uma base estática como exemplo, mas no mundo real esses dados são populados constantemente. Como você estruturaria a solução para que a base analítica seja mantida atualizada?
4. Para essa etapa o uso de tecnologias é livre. Pode utilizar qualquer linguagem de programação e banco de dados que achar melhor. Tenha em mente apenas uma busca por simplicidade para a solução do problema

**Solução**:

**1- e 2-**

Como primeiro passo, montei o modelo ER da base enviada como referência, para assim eu conseguir ter uma melhor visualização do banco de dados proposto.



Em seguida, realizei um estudo visando identificar possíveis KPIs que pudessem proporcionar uma otimização na análise do perfil dos usuário do PD. Como fruto do estudo, surgiu a base com a modelagem dimenssional onde identifiquei em 4 tipos de dimensões:

1. Dimensões Hierarquicas
2. Dimensões de tempo
3. Junk Dimension
4. Dimensão Degenerada
5. Dimensões Hierarquicas

Verifiquei que dentre os dados existem 2 dimensões que seriam mais bem aproveitadas se fossem encaradas como hierárquicas, o que facilitaria a análise, uma vez que existe apossibilidade da realização de Drill-Down e Drill-Up. Essas dimensões são uma dimensão geográfica com os atrubutos de STATE e CITY e a dimensão de disciplina com os atributos SUBJECTS, COURSES e UNIVERSITIES.

1. Dimensões de tempo

Como temos dados de tempo com granularidade de segundos, acredito que as análises serão bem mais precisas se criarmos uma dimensão de hora além da habitual dimensão de data. Com essa solução, consigo ter mais opções de análises aos dados de acesso dos usuários.

1. Junk Dimension

Temos 3 outros dados que não possuem relação com os demais dados além de terem a volumetria muito pequena. São eles PlanType (tipo de plano), SigSource (canal de inscrição), StdClient (canal de acesso). Com essas características, não justificaria a criação de uma dimensão para cada um desses tipos de dados isoladamente. Sendo assim, a melhor opção é a criação de uma dimensão junk, onde consigo ter as descrições de cada tipo de dado, além dos valores apresentados, sem a necessidade de ter diversas dimensões pequenas o que pode vir a comprometer o desempenho no momento da consulta aos dados.

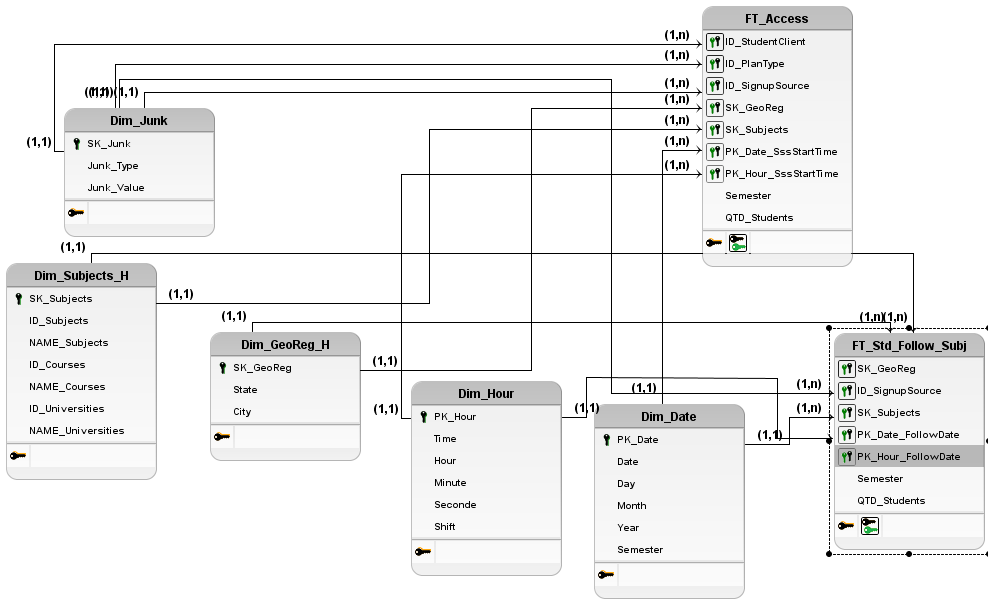
1. Dimensão Degenerada

Além das que citei acima, temos um dado que é obtido através do calculo da quantidade de semestre que o estudante está inscrito no PD. Como esse dado tem por origem a data que o estudante realizou sua inscrição, acredito não ser viável um relacionamento desta com a Dimensão Data, visto que o dado não seria bem aproveitado com funcionalidades do tipo Drill-Down ou Drill-Up que esta dimensão propcia. Acredito que o dado é bastante relevante quando trabalhamos com ele na forma de número de semestres que o estudante está cadastrado. Sendo assim, prefiro incluir ele diretamente nas tabelas fato, para que seja possível a criação de filtros durante a análise.

Além da definição das dimensões, verifiquei 4 KPIs relevantes e, dentre eles, 2 que ajudam na análise do perfil dos usuários.

1. Frequência na utilização da ferramenta;
2. Quantidade de estudantes matriculados.

Sendo assim, nosso modelo dimenssional passa a possuir 4 tabelas dimensão e 2 tabelas fato, como mostra a figura abaixo:



**3-**

**3.1**- Definir um pipeline dos dados

Como a modelagem proposta apresenta transfomações de baixa complexidade, o pipeline seria estabelecido com as camadas mais simples:

* 1. Staging Area: Cópia dos dados da origem, sem chave primária e sem relacionamentos. A camada deve ser limpa a cada início de execução.
  2. Dimensões: Criação das tabelas Dimensão, desconsiderando os dados duplicados e gerando as Surrogates Keys.
  3. Fatos: Criação das tabelas Fato, buscando as Surrugate Keys criadas na camada anterior nas dimensões e realizando os devidos agrupamentos para cálculo dos KPIs.

**3.2**- Dados populados constantemente. Como estruturar?

Como solução, temos a ferramenta Spark Streaming onde podemos configurar o monitoramento de uma pasta, realizando o consumo de dados novos automaticamente e em intervalos de tempo curto. Assim sendo, todo o fluxo de importação de dados novos pode ser executado praticamente em realtime tornando a base analítica atualizada constantemente.

**4-**

Premissas:

Para que o pipeline seja executado com sucesso, algumas ações precisam ser tomadas.

Segue o script "A1\_Script\_PRE.py" com código **PYSPARK**, a ser executado antes da execução do Pipeline:

Para gerar a camada de **Staging**, foram gerados os scripts **A2\_Script\_STG.py** e **A3\_Script\_TMPDIM.py**

Para gerar a camada de **Dimensões**, foi gerado o script **A4\_Script\_DIM.py**.

Para gerar a camada de **Fatos**, foi gerado o script **A5\_Script\_FT.py**

**2ª Parte**

**Situação problema:**

Def

1. Fazer o processamento da BASE B em conjunto com a BASE A
2. Criar um pipeline para extrair informações importantes que permitam analisar o que os usuários do PD acessam na nossa plataforma e como fazem isso. Analise os dados para extrair as informações que você julgue serem relevantes para entender o comportamento do usuário
3. Em relação à tecnologia, o único pré-requisito é a utilização do framework Spark para o processamento dos dados

**Solução**:

**1- e 2-**

Para realizar a importação dos dados da BASE B e sua execução do pipeline, foram gerados os Scripts **B1\_Script\_STG.py** e **B2\_Script\_FT.py**