DENILSON GUILHERME DUDAS

OLIST SERVIÇOS DIGITAIS LTDA

ARQUITETURA DE DADOS - OLIST

CURITIBA

2020

SUMÁRIO

1 INTRODUÇÃO

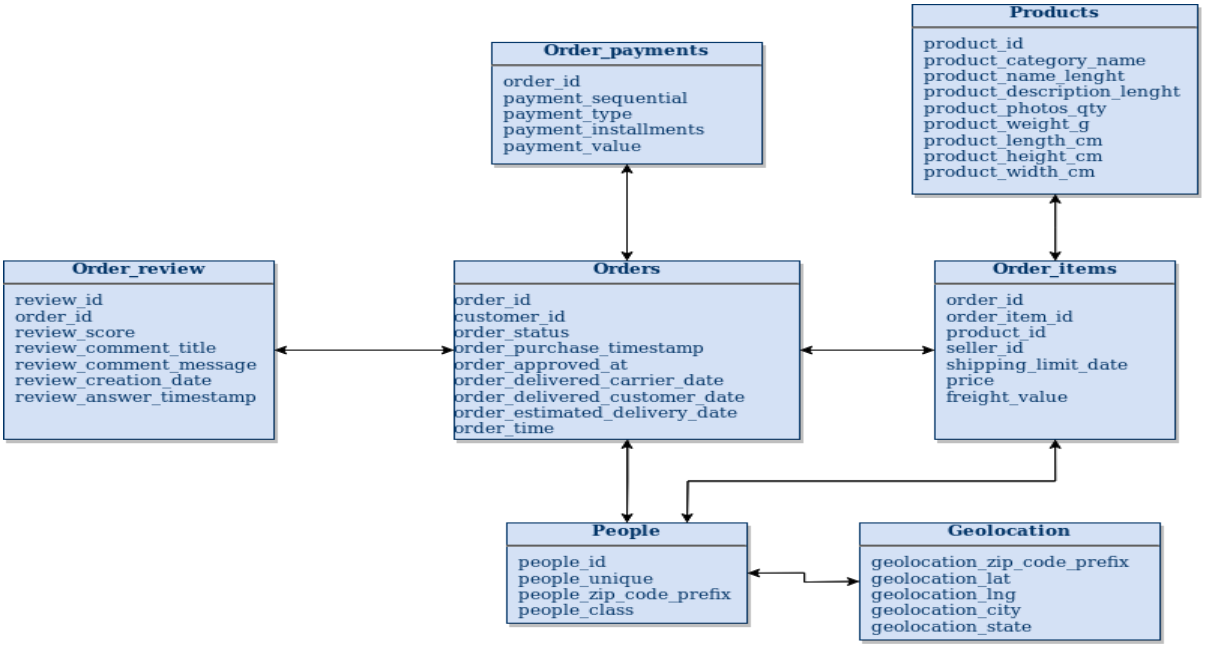
O presente documento foi desenvolvido com fim de explicar uma proposta de arquitetura a ser utilizada por analista e cientista de dados pela empresa Olist.

2 ARQUITETURA PROPOSTA

A projeto tomou como base a atual arquitetura da empresa. Dentre todas as tabelas, foram efetuadas modificações somente na customer, sellers e orders.

As tabelas customer e sellers foram unidas (people) devido possuírem campos semelhantes. Além disso recebeu mais um campo para identificação (people_class) da classificação da pessoa. As mesmas duplicavam dados de cidade e estado, além de possuir um campo com o zip-code que através dele é possível recuperar estas mesmas informações.

Considerando que a mesma se encontra ou venha ir para a nuvem, todo o armazenamento e as transações seriam cobrados, dessa forma haveria uma pequena economia.



Na tabela orders foi acrescentado o campo order_time, que efetua o cálculo entre o tempo de order_estimated_delivery_date e order_delivered_customer_date , e caso

order_estimated_delivery_date seja maior que ou igual a order_delivered_customer_date este campo recebe 'NO PRAZO', e se order_estimated_delivery_date for menor que order_delivered_customer_date então 'FORA DO PRAZO', caso contrário 'N/A'.

A tabela ORDERS foi replicada, e logo após recriada por meio da query abaixo:

```
create table orders as

SELECT *,

case

when o.order_estimated_delivery_date >= order_delivered_customer_date then 'NO PRAZO'

when o.order_estimated_delivery_date < order_delivered_customer_date then 'FORA DO PRAZO'

else 'N/A'

end order_time

FROM `projeto1-256320.SC.orders` o;
```

Todos os processos de criação de tabela, cópia de dados foram feitos em scripts python e se encontram disponíveis no repositório a seguir.

https://github.com/DudasDenilson/PETL

Os scripts foram criados para serem usados no Apache Airflow, onde esta ferramenta efetuaria o gerenciamento dos mesmos, tratando de suas execuções e replicação para um novo ambiente.

Foi em pensado em dividir em scripts para criação da estrutura, caso essa ainda não exista, assim como scripts para cópia, inserção dos dados, ou de acordo com a necessidade.

Foi utilizado nos scripts uma base postgres local para realização do teste. Porém se houvesse a possibilidade utilizaria as ferramentas de cloud do Google. Como por exemplo a base Google BigQuery. Com os dados na base do Google sua integração com a ferramenta de visualização de dados se tornaria mais fácil e rápida, pois poderia se usar o Google DataStudio. Esta é de fácil aprendizado e gratuita, além de possuir vários plugins da comunidade. Nos links abaixo pode ser visto um modelo bem simples criado para apresentação dos dados.

https://datastudio.google.com/s/r8MWHxC2Xqc

ttps://datastudio.google.com/s/ltmxHfWHKnE

Sobre o ponto de refatoração de script e criação de documentação, são dois pontos muito importantes, deixando até difícil a escolha para se priorizar. Para uma possível solução deste impasse, por que não refatorar os scripts e deixá los com a própria documentação. Isso faria com que além do refatoramento eles ainda estivessem documentados. Como exemplo podemos citar os scripts python, pode ser usado a documentação por meio do “””, ou também chamados docstrings . Após esta documentação pode se usar um utilitário da comunidade, pydoc, para geração de uma documentação HTML. Pensando até de uma forma mais genérica, por que não criar uma ferramenta que verifique os scripts alterados e gere o arquivo html de forma automática, e venha a disponibilizar estes HTMLs em um ambiente comum disponível para consulta de todos.

As KPIs em tempo real, eu concordo com o time em ser criado, pois quanto antes houver a possibilidade em se tomar uma decisão melhor, visto que empresas que trabalham com vendas acabam saindo na frente de seus concorrentes.

Em relação a empresa utilizar base de dados na nuvem e ferramentas de análise on premisses, seria um dos pontos a serem discutidos de qual seria o melhor abordagem, pois pois estando em lugares diferentes, há uma limitação, baseada na velocidade de conexão. Outro ponto a ser considerado, seria a estrutura on premisses onde a ferramenta se encontra, se está satisfatória ou necessita algum tipo de melhoria, como por exemplo um Tableu server hospedado em uma máquina virtualizada, com 1 gb de ram, processador de 1 core e 10 gb de espaço em disco. Por que não efetuar um upgrade na máquina antes de uma possível migração?. O upgrade foi realizado mas mesmo assim o problema de dashboard lento continua? Por que não revisar os dashboards, garantir que as queries estão escritas da melhor forma ou verificar se as estruturas usadas estão legais, necessitam de algum ajuste, como criação de índice, particionamento, expurgo de dados muito antigos.

Voltando ao assunto de estruturas, pode ser constatado que os dados não possuem um padrão, como por exemplo o preenchimento de cidade:

- sao paulo
- são paulo
- sãopaulo

Uma automatização do preenchimento dos campos facilitaria para o usuário, além de melhorar a qualidade dos dados.