Bootcamp: Arquiteto(a) de Big Data

Trabalho Prático

Módulo 2: Coleta e Obtenção de Dados

Objetivos de Ensino

Exercitar os seguintes conceitos trabalhados no Módulo:

1. Realizar coleta de dados em arquivos.
2. Manipulação e visualização de dados.
3. Criar modelo entidade e relacionamento para armazenamento de dados.
4. Realizar carga de dados no banco de dados MySQL.
5. Tratamento de dados.
6. Realizar consultas na linguagem SQL.
7. Conhecimento teórico ministrado nas videoaulas.

Enunciado

Como arquiteto de Big Data, sua missão é projetar um sistema de armazenamento robusto para um marketplace online. Este sistema deve gerenciar eficientemente dados de vendas, garantindo flexibilidade e escalabilidade para lidar com volumes crescentes.

As vendas incluem os seguintes campos:

- Data da venda;
- Produto vendido;

- Vendedor;
- Quantidade vendida;
- Valor unitário;
- Valor total da compra;
- Estado onde a compra foi realizada.

Um aspecto crucial a considerar é que o valor unitário dos produtos pode variar ao longo do tempo. Essa flutuação é comum no ambiente de comércio eletrônico e precisa ser refletida no design do sistema.

O desafio é criar uma estrutura que permita o fácil acesso e análise desses dados, ao mesmo tempo em que acomoda mudanças frequentes nos preços e outras variações dinâmicas do mercado. Além disso, o sistema deve ser capaz de integrar dados de diferentes fontes e proporcionar análises rápidas para apoiar decisões de negócios.

Tarefas:

Coleta de Dados: inicialmente, você deverá identificar as fontes de dados relevantes e coletar informações.

Modelo de Entidade e Relacionamento (MER): com os dados coletados, você deverá criar um modelo de entidade e relacionamento que represente as relações entre as diferentes entidades. Certifique-se de incluir todos os atributos relevantes e estabelecer as relações apropriadas entre as entidades.

Armazenamento de Dados: implemente o modelo de entidade e relacionamento em um sistema de gerenciamento de banco de dados para armazenar os dados de forma eficiente.

Pré-processamento de Dados: realize o pré-processamento necessário nos dados, incluindo limpeza, transformação e tratamento de valores ausentes, para garantir a qualidade dos dados armazenados.

Análise e Geração de Insights: use técnicas de análise de dados e visualização para explorar os dados e gerar insights relevantes.

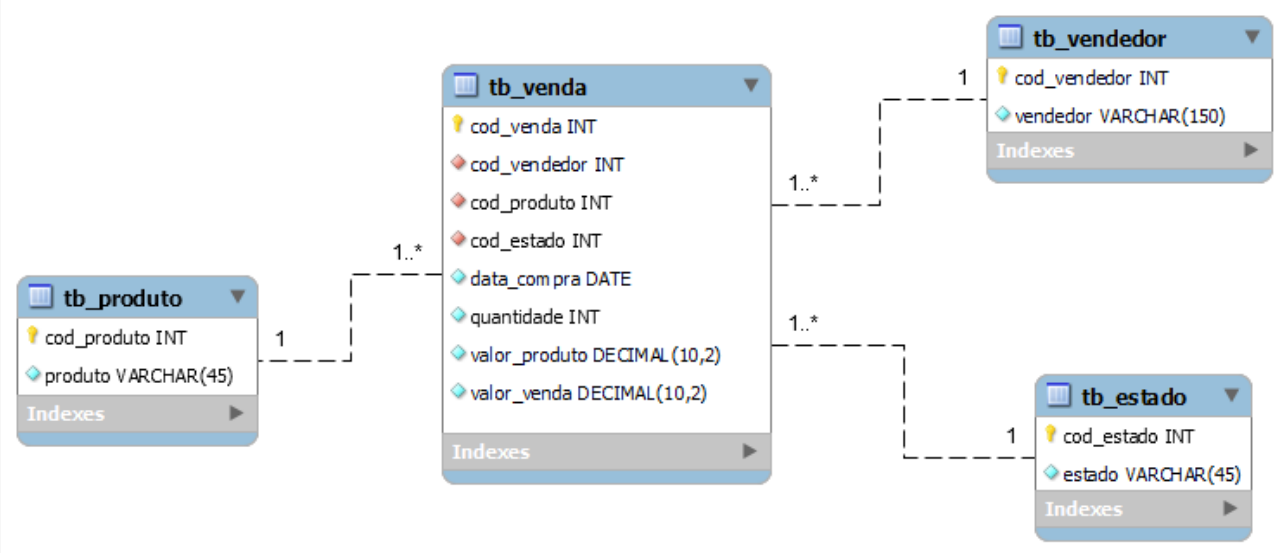
Atenção! Para garantir a obtenção dos mesmos resultados do projeto, é recomendável o uso das mesmas versões das bibliotecas.

```
VERSÕES BIBLIOTECAS UTILZIADAS
pandas: 1.5.2
sqlalchemy: 1.4.44
```

É crucial reconhecer que a linguagem de programação Python e suas bibliotecas associadas estão em constante evolução. Como resultado, pode ocorrer que funções ou métodos específicos, que costumavam estar disponíveis em versões anteriores, deixem de existir ou passem a ser implementados de maneira diferente em versões mais recentes.

Essas atualizações são realizadas para melhorar a eficiência, corrigir erros e fornecer novos recursos aos desenvolvedores. No entanto, essa dinâmica de mudança também pode criar desafios, especialmente quando se trabalha com código legado ou ao compartilhar código com outros membros da equipe. Portanto, é de extrema importância que os alunos estejam cientes dessas mudanças e estejam dispostos a se adaptar a elas.

Modelo de entidade e relacionamento que deverá ser criado.



Quando você insere dados em tabelas que têm relações de dependência com outras tabelas, como ocorre com a tabela 'tb_venda', que depende das tabelas 'tb_produto', 'tb_vendedor' e 'tb_estado', é crucial seguir uma ordem lógica para garantir que a operação de inserção seja bem-sucedida.

Por exemplo, para adicionar um registro à tabela 'tb_venda', você deve ter certeza de que as tabelas 'tb_produto', 'tb_vendedor' e 'tb_estado' já estejam preenchidas com os dados necessários. Assim, o processo de inserção deve seguir uma sequência que respeite essas dependências para evitar erros de referência.

Portanto, ao trabalhar com tabelas interconectadas, é recomendável começar populando as tabelas principais ou de referência, antes de inserir dados nas tabelas dependentes. Dessa forma, você reduz o risco de violações de integridade referencial e assegura que os relacionamentos entre as tabelas sejam mantidos corretamente.

Utilize a tabela de 'stage' para fazer um processo parecido com o PROCV do Excel para inserir os dados. Abaixo um exemplo de código.

```
insert into tb_venda (cod_vendedor, cod_produto, cod_estado,
                      data_compra, quantidade, valor_produto, valor_venda)
(
    select
        vend.cod_vendedor,
        prod.cod_produto,
        est.cod_estado,
        stg.`Data da compra`,
        stg.`Quantidade unitária`,
        stg.`Valor do produto`,
        stg.`Valor da venda`
    from stg_vendas AS stg
    inner join tb_produto as prod on prod.produto = stg.`Produto vendido`
    inner join tb_vendedor as vend on vend.vendedor = stg.`Nome do vendedor`
    inner join tb_estado as est on est.estado = stg.Localização

)
```

ATENÇÃO PARA TRATAMENTO DE DADOS

Avalie se será necessário realizar tratamento de dados ausentes nos datasets disponibilizados.

Instruções para correção de dados ausentes

1. Eliminação de dados para variáveis categóricas.
2. Para os dados numéricos, utilize:
    a. Regra do negócio para corrigi-los.
    b. Atenção para a correção baseada na regra.
        i. Ex.: valor total da venda = Quantidade unitária * Preço do produto
        ii. Utilize os próprios dados para fazer a correção.

Atividades

Para esta atividade, os alunos deverão realizar as seguintes tarefas:

1. Coletar os dados fornecidos através da lista de arquivos;
2. Criar estrutura de tabelas no banco de dados MySQL;
3. Inserir dados coletados na estrutura criada;
4. Realizar comandos SQL para extrair informações da base de dados.

Dicas do professor:

1. Antes de enviar as respostas, verifique se o gabarito está correto.
2. Analise se existem dados duplicados e elimine-os se necessário.
3. Siga fielmente todos os passos contidos no enunciado das questões.
4. É fundamental observar a configuração de autoincremento ao criar tabelas que requerem a geração automática de códigos para representar os dados.
5. Os dados disponibilizados no dataset são fictícios. Ou seja, não têm relação com o mundo real.
6. Realize a conversão da data antes de enviar para o banco de dados
    a. df['Data da compra'] = pd.to_datetime(df['Data da compra'], format='%d/%m/%Y')
7. Siga os procedimentos realizados nas videoaulas. O sucesso do experimento depende de seguir a mesma estratégia.

8. O dataset utilizado no trabalho pode ser obtido no link:

   https://leandrolessa.com.br/datasets/

   (Dados de Vendas de produtos e-commerce)