Bootcamp: Arquiteto de Big Data

Trabalho Prático

Módulo 2: Coleta e Obtenção de Dados

Objetivos de Ensino

Exercitar os seguintes conceitos trabalhados no Módulo:

- Realizar coleta de dados em arquivos.
- Manipular e visualizar dados.
- Criar modelo entidade e relacionamento para armazenamento de dados.
- Realizar carga de dados no banco de dados MySQL.
- Tratar dados.
- Realizar consultas na linguagem SQL.
- Explorar o conhecimento teórico ministrado nas videoaulas.

Enunciado

Você foi contratado como arquiteto de Big Data por uma multinacional para coletar, tratar e armazenar dados relacionados a licenças médicas processadas de seus colaboradores. O objetivo principal deste trabalho prático é criar um modelo de entidade e relacionamento (MER) que permita o armazenamento eficiente desses dados e possibilite a geração de insights valiosos para a empresa no futuro.

Tarefas:

Coleta de Dados: inicialmente, você deverá identificar as fontes de dados relevantes e coletar informações de licenças médicas, médicos e colaboradores. Isso pode incluir dados como datas de licenças médicas, diagnósticos, duração das licenças, nomes dos médicos, informações de identificação dos colaboradores, entre outros.

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, como "Licença Médica", "Médico" e "Colaborador". 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. Isso pode incluir identificar tendências nas licenças médicas, padrões de utilização de médicos, análise de custos relacionados a licenças médicas, entre outros.

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

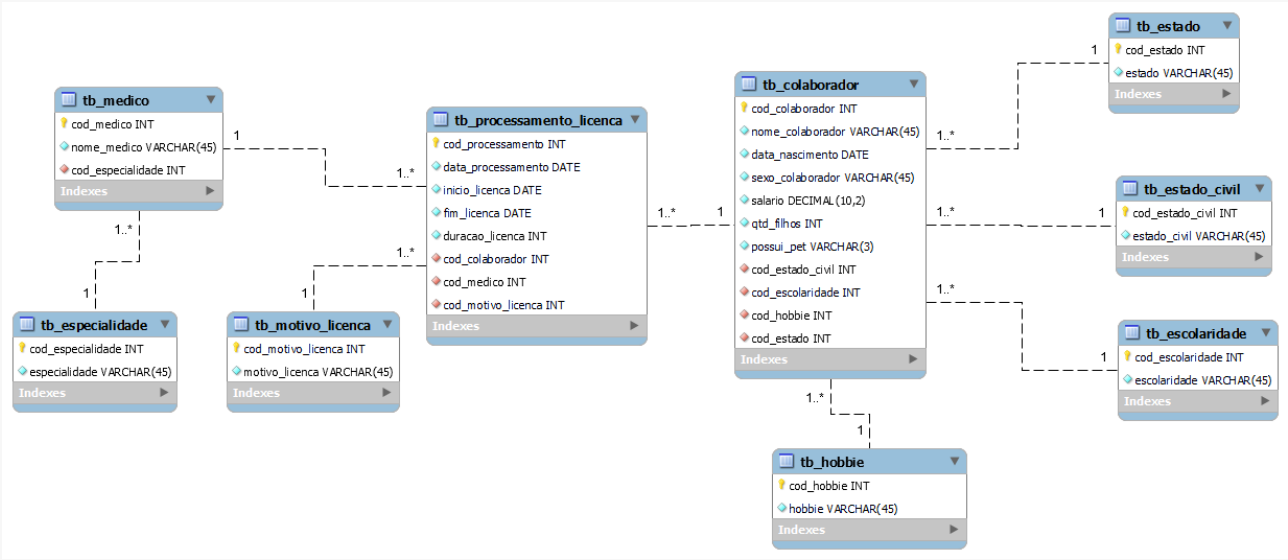
```
VERSÕES BIBLIOTECAS UTILZIADAS
pandas: 1.5.2
sqlalchemy: 1.4.44
```

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.



É importante observar que ao inserir dados em tabelas que dependem de informações de outras tabelas para concluir com sucesso a operação de inserção, como o caso da tabela 'tb_medico' que requer que a tabela 'tb_especialidade' já esteja populada, é necessário seguir uma ordem estratégica de inserção.

Além disso, 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_medico (nome_medico, cod_especialidade)
(
      SELECT distinct nome_medico, esp.cod_especialidade
      FROM stg_licenca stg
```

```
        INNER JOIN tb_especialidade esp on esp.especialidade =
        stg.especialidade
);
```

Certifiquem-se de que estamos buscando o nome da especialidade tanto na tabela 'stage' quanto na tabela 'tb_especialidade', retornando apenas o código correspondente. Repitam esse processo para todas as tabelas que se encontrem nessa mesma situação.

ATENÇÃO PARA TRATAMENTO DE DADOS

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

Instruções para correção de dados ausentes

1. Média arredondada para 2 casas decimais para as variáveis do tipo numéricas:

   Exceção para qtd_filhos utilize round().

2. Moda para as variáveis categóricas.

Atividades

Para esta atividade, os alunos deverão criar um modelo de entidade e relacionamento (MER) que permita o armazenamento eficiente de dados e possibilite a geração de insights valiosos.

- Coletar os dados fornecidos através da lista de arquivos;
- Criar estrutura de tabelas no banco de dados MySQL;
- Inserir dados coletados na estrutura criada;

- 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.
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 tem relação com o mundo real.
6. Siga os procedimentos realizados nas videoaulas. O sucesso do experimento depende seguir a mesma estratégia.

   Por exemplo cálculo das idades.
7. O dataset utilizado no trabalho pode ser obtido no link:

   https://github.com/ProfLeandroLessa/classroom-datasets/blob/master/CDD/TP/processamento_licencas_medicas.zip