Explicação dos códigos

Explicação do código JavaScript (**index.js**) utilizado para a recuperação dos dados:

readJsonFile(filename): Esta função lê um arquivo JSON do sistema de arquivos. Ela usa o método fs.readFileSync para ler o conteúdo do arquivo e então usa JSON.parse para converter os dados brutos em um objeto JavaScript.

correctNames(data): Esta função percorre cada item nos dados fornecidos e corrige os nomes e marcas substituindo os caracteres ‘æ’ por ‘a’ e ‘ø’ por ‘o’. Antes de tentar substituir os caracteres, ela verifica se a propriedade ‘nome’ e ‘marca’ existe para evitar um erro TypeError.

correctSales(data): Esta função percorre cada item nos dados fornecidos e corrige a propriedade ‘vendas’. Se ‘vendas’ for uma string, ela a converte para um número usando a função Number.

exportToJsonFile(data, filename): Esta função exporta os dados corrigidos para um arquivo JSON. Ela usa JSON.stringify para converter o objeto JavaScript em uma string JSON e então usa fs.writeFileSync para escrever a string JSON em um arquivo.

Alguns tratamentos feitos no código e pontos a compartilhar:

No código, antes de tentar substituir caracteres na propriedade ‘nome’ e ‘marca’, o código verifica se essas propriedades existem. Isso é feito para evitar um erro **TypeError** que ocorreria se tentássemos acessar uma propriedade que não existe.

Na função correctSales(data), o código verifica se a propriedade ‘vendas’ é uma string antes de tentar convertê-la em um número. Isso é feito para evitar erros que poderiam ocorrer ao tentar converter um valor que já é um número.

O código usa o módulo ‘fs’ do Node.js para ler e escrever arquivos.

Explicação do código SQL (**tabela_unificada.sql**) usado para criar a tabela única com os dados corrigidos:

Criação da tabela unificada: CREATE TABLE tabela_unificada AS

Esta parte do código cria uma nova tabela chamada "tabela_unificada". O AS é usado para renomear a tabela resultante.

Seleção dos campos:
SELECT
db1.c1 AS data,
db1.c2 AS id_marca_,
db1.c3 AS vendas,
...
Aqui, estão sendo selecionados os campos específicos de ambas as tabelas. Os alias (como AS data, AS id_marca_, etc.) são usados para renomear os campos na tabela resultante.

Origem dos Dados (FROM):
A cláusula FROM indica a fonte dos dados. Está pegando dados da tabela "corrected_database_1" (renomeada como db1) e da tabela "corrected_database_2" (renomeada como db2). A cláusula JOIN é usada para combinar os registros com base na condição especificada, que é ON db1.c2 = db2.c1.

Tratamentos e explicações adicionais:

Ao importar os arquivos json na plataforma SQL Online IDE, são criadas as tabelas com os nomes das colunas c1, c2, c3 etc., dessa forma, foi usado um alias (AS) para renomear os dados para seus nomes originais na nova tabela unificada.

A condição de junção ON db1.c2 = db2.c1 assume que as colunas "c2" em "corrected_database_1" e "c1" em "corrected_database_2" são equivalentes