

EXPLICAÇÃO CÓDIGO JS

Primeiramente, na função “corrigirBancoDeDados” usei o map para aplicar a regra a todos os itens do banco de dados selecionado, ao passar nas estruturas condicionais o valor é corrigido por meio do replace, que substitui os caracteres “æ” e “ø” por “a” e “o”, corrigindo também, no caso do banco de dados 1, o tipo do valor de vendas de “String” para “Number”. Ainda nas estruturas condicionais, está o tratamento de erros, que retorna o texto “Item inválido” e qual item não está funcionando corretamente.

Na função “corrigirEExportar” utilizei a função “fs.readFile”, que vai ler o arquivo de entrada e caso ocorra um erro, exibirá a mensagem “Erro ao ler o arquivo de entrada: “, apontando a causa do erro. Caso não haja erros, a partir da função “JSON.parse” converterá o arquivo para ser lido pelo JavaScript, associa a função “corrigirBancoDeDados” ao nome “dadosCorrigidos” para usar como parâmetro na próxima linha, com base no tipo de banco especificado, usando o “JSON.stringify” converte os dados novamente para JSON. Através da função “fs.writeFile”, os dados corrigidos são escritos no arquivo de saída, e a mensagem de sucesso ou erro é retornada.

Por fim, a chamada da função “corrigirEExportar”, recebe como parâmetro o nome do banco de dados corrompido, o nome do banco de dados corrigido e o número do banco de dados, como retorno é criado o arquivo corrigido.

EXPLICAÇÃO CÓDIGO SQL

No código SQL, a primeira linha cria uma tabela chamada “database”. Em seguida, a instrução SELECT é utilizada para escolher todas as colunas da tabela fixed_database_1 e apenas a coluna “c2” da tabela fixed_database_2. O FROM indica a fonte dos dados recuperados, enquanto o JOIN une as tabelas e o ON especifica a relação entre as colunas “c2” e “c1” nas duas tabelas.