

Código 1 – JavaScript

Este código é um script em JavaScript que lê dois arquivos JSON de bancos de dados de veículos e marcas, faz algumas correções nos dados e exporta os arquivos corrigidos em formato JSON.

```
1 //função para corrigir o texto
2 function corrigirTexto(array) {
3   array = array.replace(/ø/g, "o").replace(/æ/g, "a");
4   return array;
5 }
```

- A função utilizada para fazer a correção do texto, recebe uma String como parâmetro e utiliza o método "replace" para substituir os caracteres ø por o e æ por a, retornando no final a String corrigida.

```
7 //função para corrigir numero
8 function corrigirNumero(texto) {
9   if (typeof texto === "string")
10     return Number(texto);
11   return texto;
12 }
```

- Esta função recebe um parâmetro de texto e verifica se ele é uma string. Se for, a função converte o texto em um número e o retorna. Caso contrário, o texto é retornado sem alterações.

```
13 //função para exportar texto corrigido
14 function exportar(database, nome){
15   const fs = require('fs');
16
17   fs.writeFileSync(nome, JSON.stringify(database));
18 };
19
```

- Esta função recebe dois parâmetros: um objeto JSON contendo um banco de dados e o nome do arquivo a ser exportado. A função usa o módulo "fs" do Node.js para escrever o objeto JSON em um arquivo com o nome especificado

```
19
20 let jsonData1 = require('./data_base1.json');
21
```

- Lê o arquivo "data_base1.json" e armazena seu conteúdo em uma variável chamada jsonData1.

```

22  jsonData1.forEach(veiculo =>
23  {
24    veiculo.nome = corrigirTexto(veiculo.nome);
25    veiculo.vendas = corrigirNumero(veiculo.vendas)
26  });
27

```

- Esta parte do código percorre cada objeto no array jsonData1 e faz algumas correções nos dados. Para cada objeto veiculo, a função corrigirTexto é usada para corrigir o nome do veículo e a função corrigirNumero é usada para corrigir o número de vendas.

```

29  let jsonData2 = require('./data_base2.json');
30
31  jsonData2.forEach(marca =>
32  {
33    marca.marca = corrigirTexto(marca.marca);
34  });

```

- Esta parte do código lê o arquivo "data_base2.json" e armazena seu conteúdo em uma variável chamada jsonData2., depois ela percorre cada objeto no array jsonData2 e corrige o nome da marca usando a função corrigirTexto.

```

36  exportar(jsonData1, "novoData1Corrigido.json");
37  exportar(jsonData2, "novoData2Corrigido.json");
38
39  console.log("FIM");
40
41  process.exit();

```

- Esta linha usa a função "exportar" para criar os novos arquivos corrigidos em formato .json, coloquei o console.log para indicar que o programa já tinha se finalizado e os arquivos gerados.

Código 2 – SQL

```
1  CREATE TABLE relatorio (  
2      id INTEGER PRIMARY KEY,  
3      data DATE,  
4      id_marca INTEGER,  
5      vendas INTEGER,  
6      valor_do_veiculo NUMERIC,  
7      nome TEXT,  
8      marca TEXT  
9  );  
10  
11  INSERT into relatorio (data, id_marca, vendas, valor_do_veiculo, nome)  
12  SELECT c1, c2, c3, c4, c5  
13  FROM novoData1Corrigido;  
14  
15  CREATE TABLE relatorio2 AS  
16  SELECT a.data, a.vendas, a.valor_do_veiculo, a.nome, a.id_marca, b.c2  
17  FROM relatorio AS a  
18  JOIN novoData2Corrigido AS b  
19  ON a.id_marca = b.c1;
```

Eu importei os arquivos json corrigidos para o sql server online, lá eles já geraram as tabelas e consegui utilizadas para poder criar a tabela principal presente neste código.

A primeira instrução presente nesse código cria uma tabela chamada "relatorio" com seis colunas: "id" (primary key), "data", "id_marca", "vendas", "valor_do_veiculo" e "nome". A tabela é criada com o comando "CREATE TABLE".

A segunda instrução insere dados na tabela "relatorio" usando o comando "INSERT INTO". Esses dados serão selecionados da tabela "novoData1Corrigido" que foi criada pelo arquivo json exportado do código anterior, eles então serão colocados nas colunas correspondentes na tabela "relatorio".

A terceira instrução cria uma tabela chamada "relatorio2" usando o comando "CREATE TABLE AS". A nova tabela é preenchida com dados selecionados de duas tabelas: "relatorio" e "novoData2Corrigido". A instrução usa o comando "SELECT" para definir as colunas e as condições de correspondência (JOIN) entre as duas tabelas. A nova tabela "relatorio2" tem cinco colunas: "data", "vendas", "valor_do_veiculo", "nome", "id_marca" e "c2"(representando o nome da marca).

Sendo assim foi gerada a tabela unificada, com todos os dados necessário para eu pode responder o relatório.