

Explicação sobre o código utilizado para corrigir dados:

O código lê os arquivos JSON, um loop é usado para iterar sobre cada objeto em um Array chamado de dados, utilizando o método `replace` para substituir caracteres especiais, dentro do loop ele passa por cada objeto usando expressões regulares verificando se a um caractere incorreto, depois de substituídos o novo valor da String 'veiculos' é atribuído a propriedade 'nome' do objeto correspondente no array 'dados'. Trecho do código citado abaixo.

```
const dados = require("../dataBase/broken_database_1.json");

for (let i = 0; i < dados.length; i++) {
  let veiculo = dados[i].nome;
  veiculo = veiculo.replace(/æ/g, "a").replace(/ø/g, "o");
  dados[i].nome = veiculo;
}
```

O código a seguir é utilizado para corrigir o valor 'vendas' do arquivo json que em vez de estar em número inteiro estava em String, Utilizando o `forEach` o código percorre o array dados e verifica se a propriedade vendas é uma String por meio do método 'typeof' caso seja um String é utilizado o método `parseFloat()` para Passar para um tipo numérico. Trecho do código citado abaixo.

```
dados.forEach((obj) => {
  if (typeof obj.vendas === "string") {
    obj.vendas = parseFloat(obj.vendas);
  }
});
```

Esta parte faz a reescrita do arquivo já corrigido utilizando a função 'fs.writeFile()' para reescrever o arquivo JSON, caso ocorra algum erro durante a escrita do arquivo, o 'console.error(err)' é usado para imprimir o erro no console. Caso contrário uma mensagem avisando que o arquivo foi salvo com sucesso será impressa no console.

```
fs.writeFile("broken_database_1.json", jsonString, (err) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log("Arquivo salvo com sucesso!");
});
```

Estes mesmos códigos são executados uma segunda vez com os parâmetros do segundo arquivo a ser corrigido.