

O objetivo do teste foi editar arquivos JSON utilizando a linguagem JavaScript. Posteriormente houve o acesso à plataforma Sqliteonline para a geração de uma tabela onde suas células foram preenchidas pelo consumo dos arquivos JSON. Por fim, o Google Sheets foi necessário para a elaboração de tabelas mais elaboradas e conteúdos visuais como cores e gráficos.

## 1. Explicação sobre o código em JavaScript

### a) Ler os arquivos Json:

```
// 1. Ler os arquivos json:

// Acessar o módulo fs do Node.js
const fs = require('fs');

const dataFile1 = fs.readFileSync('../db/broken_database_1.json', 'utf8');
const dataFile2 = fs.readFileSync('../db/broken_database_2.json', 'utf8');

const jsonData1 = JSON.parse(dataFile1);
const jsonData2 = JSON.parse(dataFile2);

// test
// console.log(jsonData1, jsonData2);
```

Houve a necessidade de acessar funções especiais pelo módulo FS do Node.js. Posteriormente as constantes “dataFile1” e “dataFile2” armazenaram os respectivos conteúdos dos arquivos JSON1 e JSON2.

Além do que foi citado, também foi necessário transformar o conteúdo textual (json) em algo que pudesse ser manipulado pelo JavaScript. O array de objetos com seus pares de chaves-valores foram editados pela função JSON.parse().

### b) Corrigir nomes de marca e veículo:

```
// 2. Corrigir nomes de marca e veículo --> substituir caracteres indesejáveis por vogais:

function changeCharacters(database, key, replacement) {
  for (let i = 0; i < database.length; i++) {
    if (database[i][key] && typeof database[i][key] === 'string') {
      for (let substitution of replacement) {
        database[i][key] = database[i][key].replace(substitution[0], substitution[1]);
      }
    }
  }
}

const replacement = [
  [new RegExp('ø', 'g'), 'o'],
  [new RegExp('æ', 'g'), 'a']
];

changeCharacters(jsonData1, 'nome', replacement);
changeCharacters(jsonData2, 'marca', replacement);
```

Acima houve a criação da função “changeCharacters()” para acessar, por meio de um laço for, os caracteres que apresentaram problema de modo a serem trocados por suas respectivas vogais.

No final, a função é chamada para que seja executada, como parâmetros as constantes inicialmente criadas e as chaves de objetos (“nome” e “marca”) sofreram um “replacement”, onde houve a substituição requisitada pelo teste.

c) Corrigir vendas:

```
// 3. Corrigir vendas --> passar tipo "string" para "number":

// o último objeto contém a chave "vendas" com valor do tipo string
// console.log(jsonData1[(jsonData1.length - 1)]);

function fixSalesType(database, key) {
  for (let i = 0; i < database.length; i++) {
    if (database[i][key] && typeof database[i][key] === 'string') {
      database[i][key] = parseFloat(database[i][key]);
    }
  }
}

fixSalesType(jsonData1, 'vendas');

// test
// console.log(jsonData1);

// teste do último objeto alterando o tipo da chave "vendas" "para number"
// console.log(jsonData1[(jsonData1.length - 1)]);
```

Alguns dos valores das chaves “vendas” eram do tipo “string”, o que era indevido visto que são números inteiros. O código acima alterou o tipo e atualizou a constante “jsonData1”.

d) Exportar um arquivo JSON com o banco corrigido:

```
// 4. Exportar os arquivos JSON após correções:

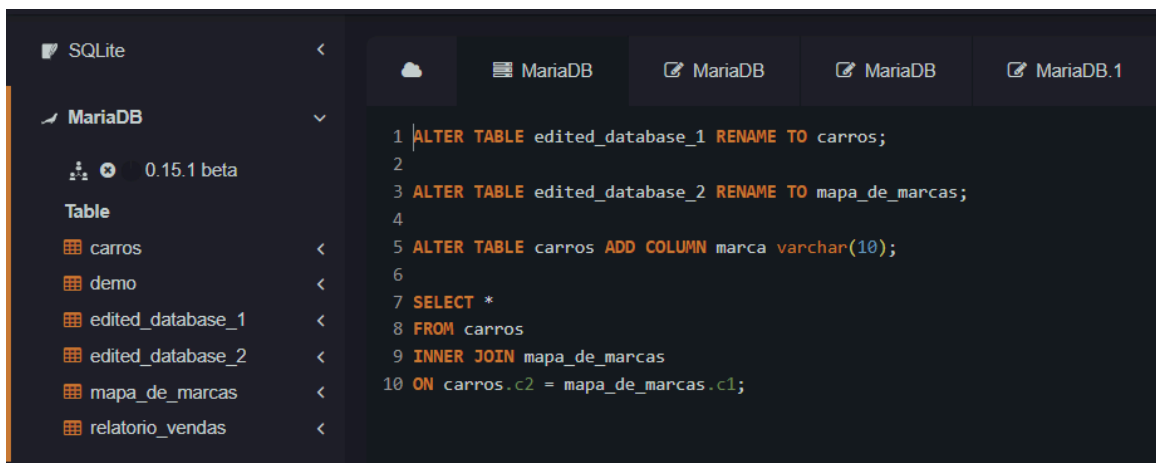
function exportToFile(database, nomeArquivo) {
    const jsonCorrigido = JSON.stringify(database, null, 2);
    fs.writeFileSync(nomeArquivo, jsonCorrigido, 'utf-8');
}

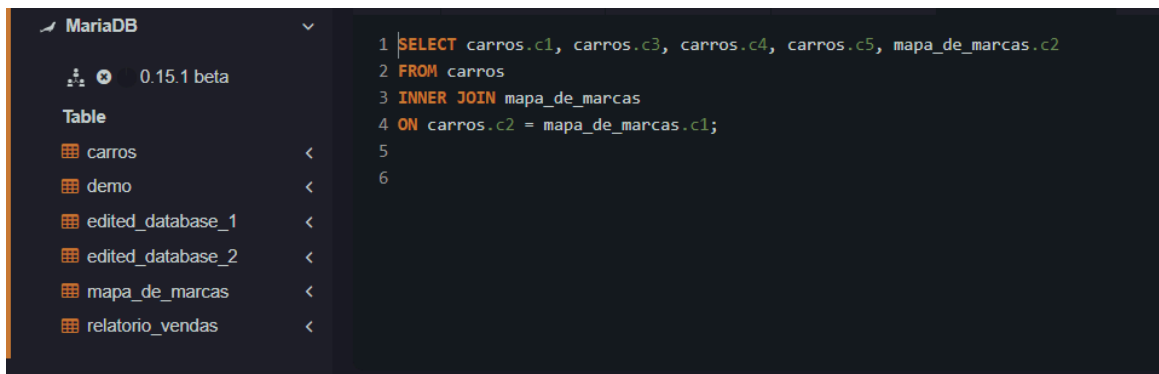
exportToFile(jsonData1, 'edited_database_1.json');
exportToFile(jsonData2, 'edited_database_2.json');

// test
// console.log(jsonData1, jsonData2);
```

Por fim, o array de objetos foi exportado novamente como arquivo JSON.

## 2. Utilizando o SQLITE





The screenshot shows the MariaDB command-line interface. On the left, there is a sidebar with the following elements: the MariaDB logo, the version '0.15.1 beta', and a section titled 'Table' containing a list of tables: 'carros', 'demo', 'edited\_database\_1', 'edited\_database\_2', 'mapa\_de\_marcas', and 'relatorio\_vendas'. Each table name is preceded by a small icon and followed by a right-pointing arrow. The main area on the right displays a SQL query across six lines, numbered 1 to 6. The query is: 1 SELECT carros.c1, carros.c3, carros.c4, carros.c5, mapa\_de\_marcas.c2; 2 FROM carros; 3 INNER JOIN mapa\_de\_marcas; 4 ON carros.c2 = mapa\_de\_marcas.c1; 5 6

```
1 SELECT carros.c1, carros.c3, carros.c4, carros.c5, mapa_de_marcas.c2;
2 FROM carros;
3 INNER JOIN mapa_de_marcas;
4 ON carros.c2 = mapa_de_marcas.c1;
5
6
```

De forma resumida, toda a syntax vista representa a criação de uma tabela que consumiu dados das “tabelas json”. Posteriormente foi realizado a exportação como arquivo do tipo csv. Este foi carregado no Google Sheets para a criação do relatório (ver Relatório de Vendas - 2022).