

## EXPLICAÇÃO DOS CODIGOS

### Princiapal.js

Importação de Módulos:

```
//IMPORTA OS ARQUIVOS JSON
const data1 = require("./db/broken_database_1.json")
const data2 = require("./db/broken_database_2.json")
const fs = require("fs")
```

Os módulos necessários são importados: fs para manipulação de arquivos e os dados dos bancos de dados (JSON) são importados de seus respectivos arquivos.

Tratamento de Nomes no BD2:

```
//tira erro dos nomes do BD2
for (const key in data2) {

    data2[key].marca = data2[key].marca.replace(/æ/g,"a")
    data2[key].marca = data2[key].marca.replace(/ø/g,"o")

}
```

Substitui caracteres especiais nas propriedades de marca do banco de dados 2.

Tratamento de Nomes e Vendas no BD1:

```
for (const key in data1) {
    //TIRA OS ERROS DE NOMES DO BD1
    data1[key].nome = data1[key].nome.replace(/æ/g,"a")
    data1[key].nome = data1[key].nome.replace(/ø/g,"o")
    //TIRA OS ERROS DE VENDAS DO BD1
    if (typeof(data1[key].vendas == "string")) {
        data1[key].vendas = parseInt(data1[key].vendas)
    }
}
```

Substitui caracteres especiais nas propriedades de nome do banco de dados 1.

Converte a propriedade vendas para número, se for uma string.

Exportação dos Dados Corrigidos:

```
//EXPORTA O JSON USANDO fs
fs.writeFile("./src/db/broken_database_1_corrigid.json",JSON.stringify(data1),"utf-8", (erro)=> {
    if (erro) {
        console.log("ERRO ESCRITA DB 1");
    }
});
fs.writeFile("./src/db/broken_database_2_corrigid.json",JSON.stringify(data2),"utf-8", (erro)=> {
    if (erro) {
        console.log("ERRO ESCRITA DB 2");
    }
});
```

Exporta os dados corrigidos para novos arquivos JSON.

Caso ocorra algum erro durante a escrita, exibe uma mensagem de erro.

## Consulta\_db.sql

```
1 SELECT
2     t1.c1 AS data,
3     t1.c2 AS id_marca,
4     t1.c3 AS vendas,
5     t1.c4 AS valor_do_veiculo,
6     t1.c5 AS nome,
7     t2.c2 AS nome_marca
8 FROM broken_database_1_corrigid t1
9 INNER JOIN broken_database_2_corrigid t2 ON t1.c2 = t2.c1;
```

Este código SQL é uma consulta que combina dados de duas tabelas, `broken_database_1_corrigid` e `broken_database_2_corrigid`, usando a cláusula `INNER JOIN`. A consulta seleciona colunas específicas de ambas as tabelas, renomeando-as conforme necessário.

**SELECT:** Indica as colunas que serão incluídas no resultado da consulta.

`t1.c1 AS data, t1.c2 AS id_marca, t1.c3 AS vendas, t1.c4 AS valor_do_veiculo, t1.c5 AS nome:` Seleciona colunas da tabela `broken_database_1_corrigid` e as renomeia para `data`, `id_marca`, `vendas`, `valor_do_veiculo` e `nome`, respectivamente.

`t2.c2 AS nome_marca:` Seleciona a coluna `c2` da tabela `broken_database_2_corrigid` e a renomeia como `nome_marca`.

`FROM broken_database_1_corrigid t1:` Especifica a tabela principal como `broken_database_1_corrigid` e a renomeia como `t1`.

`INNER JOIN broken_database_2_corrigid t2 ON t1.c2 = t2.c1:` Realiza uma junção interna (`INNER JOIN`) entre `t1` e `t2` com base nas colunas `c2` e `c1`, respectivamente. Isso significa que apenas as linhas com valores correspondentes nessas colunas serão incluídas no resultado.