Explicação dos Códigos

Código JS

Neste meu código JS fiz alguns comentários onde achei que seria necessário e explicativo, fiz cada função de acordo com o que foi pedido, uma função para ler os arquivos, outra função para percorrer os bancos e corrigir os erros nos caracteres, outra para converter os valores que estavam em string em número e por último uma para exportar os dois arquivos novos dos bancos com tudo corrigido.

Fiz pequenos tratamentos de bug, com confirmação se a operação deu certo ou aviso que deu errado.

Quis fazer o código todo em inglês pois prefiro sempre testar e aperfeiçoar meu inglês no dia a dia.

```js
const fs = require('fs');

// função para ler os dois arquivos json (tratamento de bug caso erro na leitura)
function readFiles(fileOne, fileTwo) {
    try {
        const dataOne = fs.readFileSync(fileOne, 'utf8');
        const dataTwo = fs.readFileSync(fileTwo, 'utf8');
        return { jsonOne: JSON.parse(dataOne), jsonTwo: JSON.parse(dataTwo) };
    } catch (error) {
        console.error('Erro na leitura dos arquivos: ', error.message);
        return null;
    }
}

// função de correção dos caracteres específicos nos nomes
function fixNames(data) {
    const replaceCharacters = (str) => str.replace(/æ/g, 'a').replace(/ø/g, 'o');
    const fixAllNames = (obj) => {
        for (let key in obj) {
            if (typeof obj[key] === 'string') obj[key] = replaceCharacters(obj[key]);
            else if (typeof obj[key] === 'object') fixAllNames(obj[key]);
        }
    };
    if (data) fixAllNames(data);
    return data;
}

// função de conversão de strings para valores numéricos
function fixSales(data) {
    const fixAllSales = (obj) => {
        for (let key in obj) {
            if (typeof obj[key] === 'string' && !isNaN(obj[key])) obj[key] = parseFloat(obj[key]);
            else if (typeof obj[key] === 'object') fixAllSales(obj[key]);
        }
    };
    if (data) fixAllSales(data);
    return data;
}

// função para exportar os dados para arquivo json (tratamento de bug caso erro ao exportar e nos dados)
function exportFiles(destinationPath, fileName, data) {
    try {
        if (!data) {
            console.error(`Erro ao exportar o arquivo ${fileName}: Os dados são indefinidos ou nulos.`);
            return;
        }
        const filePath = `${destinationPath}/${fileName}`;
        const jsonData = typeof data === 'object' ? JSON.stringify(data, null, 2) : data.toString();
        fs.writeFileSync(filePath, jsonData);
        console.log(`O arquivo ${filePath} foi exportado com sucesso!`);
    } catch (error) {
        console.error(`Erro ao exportar o arquivo ${fileName}:`, error.message);
    }
}

const destinationPath = 'C:/Users/Giovana/Downloads/Projeto/Database';

// leitura dos arquivos, correção dos nomes e dados de vendas, exporta os arquivos novos corrigidos
const { jsonOne, jsonTwo } = readFiles('C:/Users/Giovana/Downloads/Projeto/Database/broken_database_1.json', 'C:/Users/Giovana/Downloads/Projeto/Database/broken_database_2.json');

if (jsonOne && jsonTwo) {
    const fixed_database_1 = fixSales(fixNames(jsonOne));
    const fixed_database_2 = fixSales(fixNames(jsonTwo));

    exportFiles(destinationPath, 'fixed_database_1.json', fixed_database_1);
    exportFiles(destinationPath, 'fixed_database_2.json', fixed_database_2);
}
```

Código SQL

Já meu código SQL foi menor, fiz comentários também onde achei necessário, criei uma nova tabela chamada database, defini as colunas da nova tabela, após isso inseri os dados da primeira tabela corrigida, faço a mesma coisa pra segunda tabela corrigida atualizando a coluna marca com sua correspondência e por último verifico os novos dados da tabela.

O meu código SQL não teve muita complexidade, não fiz em inglês por conta dos nomes das colunas onde eu não achei certo mexer. Usei o site SQL Online IDE para fazer tudo isso e para exportar a tabela database como um arquivo .CSV, que posteriormente foi usado para fazer os gráficos e o relatório de vendas.

```sql
-- criação a nova tabela com nome 'database'
CREATE TABLE database (
    data DATE,                          -- coluna para armazenar a data
    vendas INT,                         -- coluna para armazenar as vendas
    valor_do_veiculo DECIMAL(10, 2),    -- coluna para armazenar o valor do veículo
    nome TEXT,                          -- coluna para armazenar o nome
    id_marca INT,                       -- coluna para armazenar o id da marca
    marca TEXT                          -- coluna para armazenar a marca
);

-- adiciona os dados da tabela fixed_database_1 na nova tabela 'database'
INSERT INTO database (data, vendas, valor_do_veiculo, nome, id_marca)
SELECT data, vendas, valor_do_veiculo, nome, id_marca_
FROM fixed_database_1;

-- adiciona os dados da tabela fixed_database_2 na nova tabela 'database'
UPDATE database
SET marca = (
    SELECT fixed_database_2.marca
    FROM fixed_database_2
    WHERE database.id_marca = fixed_database_2.id_marca
);

-- verifica todos os dados na nova tabela
SELECT * FROM database;
```