

# EXPLICAÇÃO DOS CÓDIGOS

EXPLICAÇÃO DO COD.JS

EXPLICAÇÃO DO TABELA.SQL

```

const { Console } = require("console");
const fs = require("fs");

fs.readFile("./broken_database_1.json", "utf-8", (error, arq)=>{
  try{
    let lista = [];
    const user = JSON.parse(arq);
    var i=0;
    while(user[i]!=null){
      //AJEITANDO VENDAS
      vendas = user[i].vendas;
      vendass = parseInt(vendas);
      //AJEITANDO VENDAS

      //AJEITANDO O NOME
      var palavra = "";
      var name = user[i].nome;
      for(var j=0; j<name.length; j++){
        if(name[j]=="ø"){
          palavra += "o";
        }
        else if(name[j]=="æ"){
          palavra += "a";
        }
        else{
          palavra += name[j];
        }
      }
      const data={
        data:user[i].data,
        id_marca_:user[i].id_marca_,
        vendas:vendass,
        valor_do_veiculo:user[i].valor_do_veiculo,
        nome:palavra,
      };
      lista[i]=data;
      i++;
      //AJEITANDO O NOME
    }
    const dataString = JSON.stringify(lista, null);
    fs.writeFileSync("./database1.json", dataString, "utf-8");
  }catch(e){
    console.log(e);
  }
});

fs.readFile("./broken_database_2.json", "utf-8", (error, arq)=>{
  try{
    let lista = [];

```

```

const user = JSON.parse(arq);
var i=0;
while(user[i]!=null){
    //AJEITANDO A MARCA
    var palavra = "";
    var name = user[i].marca;
    for(var j=0; j<name.length; j++){
        if(name[j]=="ø"){
            palavra += "o";
        }
        else if(name[j]=="æ"){
            palavra += "a";
        }
        else{
            palavra += name[j];
        }
    }
    const data={
        id_marca:user[i].id_marca,
        marca:palavra,
    };
    lista[i]=data;
    i++;
    //AJEITANDO A MARCA
}
const dataString = JSON.stringify(lista, null);
fs.writeFileSync("./database2.json", dataString, "utf-8");
}catch(e){
    console.log(e);
}
});

```

Código em javascript que faz a leitura dos arquivos broken\_database\_1.json e broken\_database\_2.json através da função fs.readFile, após isso temos um try catch que serve para capturar algum erro como ter digitado o nome do arquivo de forma errada. Para o broken\_database\_1.json foi utilizado a função parseInt nas vendas para mudar as que estiverem em string, além disso para realizar a manutenção das letras "æ" por "a" e "ø" por "o" foi utilizado uma estrutura de repetição (for) que no nome verifica cada letra e caso for "æ" ou "ø" faz a devida mudança (o mesmo acontece para o broken\_database\_2.json) após as mudanças os objetos são guardados em uma lista e após isso é escrito no arquivo correto (database1.json e database2.json) através da função fs.writeFileSync.

```

/*PARA ALTERAR OS NOMES REMOVA OS COMENTARIOS*/

/*
ALTER TABLE database1 RENAME COLUMN c1 TO data;
ALTER TABLE database1 RENAME COLUMN c2 TO id_marca;
ALTER TABLE database1 RENAME COLUMN c3 TO vendas;
ALTER TABLE database1 RENAME COLUMN c4 TO valor;
ALTER TABLE database1 RENAME COLUMN c5 TO nome;
ALTER TABLE database2 RENAME COLUMN c1 TO id_marca;
ALTER TABLE database2 RENAME COLUMN c2 TO marca;
*/

SELECT marca AS database2, SUM(VENDAS) AS database1
FROM database2, database1
WHERE database1.id_marca = database2.id_marca
GROUP by marca
ORDER BY SUM(VENDAS) DESC;

SELECT nome as database1, SUM((vendas * valor)) as database1
from database1
GROUP by nome
ORDER BY SUM((vendas * valor)) DESC;

SELECT marca AS database2, AVG(VENDAS) AS database1
FROM database2, database1
WHERE database1.id_marca = database2.id_marca
GROUP by marca
ORDER BY SUM(VENDAS) DESC;

SELECT marca AS database2, SUM(vendas) AS database1, SUM((vendas *
valor)) as database1
FROM database2, database1
WHERE database1.id_marca = database2.id_marca
GROUP by marca
ORDER BY SUM((vendas * valor)) DESC;

SELECT nome as database1, valor as database1, sum(vendas) as database1
FROM database1
GROUP by nome
ORDER BY SUM(vendas) DESC;

```

**Código em SQL que lê os arquivos database1 e database2 e realiza 5 selects que fazem as seguintes operações:**

- 1- Retorna uma tabela com as marcas e seus respectivos volumes de vendas (do maior volume de vendas para o menor).**
- 2- Retorna uma tabela com todos os veículos e suas receitas geradas (da maior para a menor).**

- 3- Retorna uma tabela com as marcas e suas médias de vendas no ano (da maior média para a menor).
- 4- Retorna uma tabela com o nome das marcas, as receitas e seus números de vendas (da maior receita para a menor).
- 5- Retorna os veículos, seus valores individuais e os seus totais de vendas (do maior número de vendas para o menor).

A função SUM no sql realiza a soma de todos os elementos na tabela e a função AVG realiza a média dos elementos na tabela.