


Conserto do Banco de Dados

Já que os dados estariam extraídos, o primeiro objetivo para correção dos problemas e dar andamento ao ETL seria transformá-los e corrigidos, como parte da database veio com certos caracteres incorretos seria necessário antes fazer a correção deles usando JavaScript conforme solicitado no case. Segue abaixo o código com os comentários do que foi feito para poder ser realizada a transformação dos caracteres errôneos para os corretos e a alteração de algumas numerações que estavam no tipo string para o tipo int.

O arquivo com o código também se encontra na pasta do drive: [Case Data Analytics - Monks - Google Drive](#)

Este documento foi feito utilizando o documentos google e pode ser acessado também por esse link:

 [Case_Data_Analytics - Monks - Joao_Victor_Pereira_Sobreira_Vieira](#)

Importações

Primeiramente é feito o Import do fs com o objetivo de exportar o arquivo corrigido em JSON no final.

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

Logo após o Import tanto da **broken_database_1.json** quanto da **broken_database_2.json** definindo então essas variáveis.

```
const broken_data = require('./broken_database_1.json');  
const broken_data2 = require('./broken_database_2.json');
```

Conserto dos Caracteres

É feito então a criação de uma função chamada characterReplace que tem como objetivo realizar a alteração dos caracteres errados substituindo então “ø” por “o” e “æ” por “a”, para isso pensei primeiramente que seria necessário o uso de um for com o objetivo de criar um loop do código por uma quantidade limitada de vezes, passando globalmente por cada linha do banco de dados substituindo os caracteres incorretos pelos corretos. O `i = 0` seria responsável por definir o início como a primeira linha do array, `i < broken_data.lenght` serviria como o limite da repetição para o for, indo até a última linha do array, o `i++` para sempre adicionar mais 1 no `i` e assim ir passando por cada linha. Já para poder consertar o caractere o método `replace` seria ideal para isso, com ele é possível fazer a substituição, para isso apenas chamei o método conectando ele ao database que foi chamado pela função

criada, o g após ambos os caracteres incorretos seria para determinar como uma leitura global, assim se tiver o mesmo caractere incorreto mais de uma vez na mesma palavra ou frase ele não vai substituir só o primeiro, mas sim ambos.

```
function characterReplace(database){  
  
  if (Object.keys(database[0]).includes('nome')){  
    for(var i = 0; i < database.length; i++){  
      database[i].nome = database[i].nome.replace(/ø/g, 'o').replace(/æ/g,'a')  
    }  
  }  
  else{  
    for(var i = 0; i < database.length; i++){  
      database[i].marca = database[i].marca.replace(/ø/g, 'o').replace(/æ/g,'a')  
    }  
  }  
}  
  
characterReplace(broken_data)  
characterReplace(broken_data2)
```

Mudança dos Tipos de Dados

Novamente o uso do for porém desta vez o objetivo seria transformar todo número que estiver como string em um int, foi usado o mesmo conceito do i para o for passar por cada linha, o prop é usado apenas como uma denominação para cada propriedade do objeto, ela poderia ser substituída por qualquer outra palavra, no if é verificado se está sendo analisada a propriedade diretamente do objeto, verifica-se também se ela não é nula e se não é um caractere que não possa ser convertido para um número, com isso é possível então no final fazer a conversão usando o +broken_data, o + na frente serve para fazer a conversão de string para número.

```
for(var i = 0; i < broken_data.length; i++){  
  for(var prop in broken_data[i]){  
    if(broken_data[i].hasOwnProperty(prop) && broken_data[i][prop] !== null && !isNaN(broken_data[i][prop])){  
      broken_data[i][prop] = +broken_data[i][prop];  
    }  
  }  
}
```

Exportação das Databases Corrigidas

No final é feita a exportação dos arquivos corrigidos como JSON usando o método `writeFileSync`, nomeando então o nome do novo arquivo, e o convertendo para json usando o `stringify` seguido de um `null` tendo como objetivo garantir garantir que não haverá uma alteração nos dados de saída e o `2` para dar um certo espaço e deixar o arquivo em JSON mais legível.

```
fs.writeFileSync('corrected_database_1.json', JSON.stringify(broken_data,  
null, 2));  
fs.writeFileSync('corrected_database_2.json', JSON.stringify(broken_data2,  
null, 2));
```

União das Tabelas

Tendo sido feita a correção e exportação das novas databases o próximo passo seria combiná-las para poder começar o processo de análise dos dados, foi usado conforme sugerido no case o site <https://sqliteonline.com/>, nele foi possível dar upload em ambas as databases corretas, corrigir o nome das colunas para os nomes devidos e depois com um simples comando em SQL juntar as duas.

```
CREATE TABLE complete_database AS  
SELECT cd1.data, cd1.id_marca, cd1.vendas, cd1.valor_do_veiculo, cd1.nome,  
cd2.marca  
FROM corrected_database_1 AS cd1  
FULL JOIN corrected_database_2 AS cd2  
ON cd1.id_marca = cd2.id_marca;
```

O comando CREATE TABLE foi usado para conseguir criar uma nova tabela que seria denominada "complete_database", o SELECT veio logo em seguida com o objetivo de selecionar todas as colunas de ambas as corrected_database utilizadas, a alcunha de cd1 foi determinada para a corrected_database_1 e cd2 para correct_database_2, após isso foi escolhida uma das duas tabelas utilizando o FROM para então fazer um FULL JOIN e juntar todos os dados das duas tabelas, usando como ponte para isso o "id_marca" já que ele seria um dado compartilhado de forma igualitária por ambas as tabelas, concluindo esse processo foi possível baixar a nova tabela que seria a junção das duas como um arquivo de CSV.

Tratamento final e interpretação dos dados

Após conseguir exportar o novo banco de dados criado em formato CSV foi usado o Google Planilhas para poder olhar os dados e responder aos questionamentos feitos no case, para conseguir visualizar e apresentar os dados de uma forma mais completa utilizei tanto o Looker Studio quanto o próprio Google Planilhas fazendo nele o uso do recurso tabela dinâmica. Segue abaixo link de ambos.

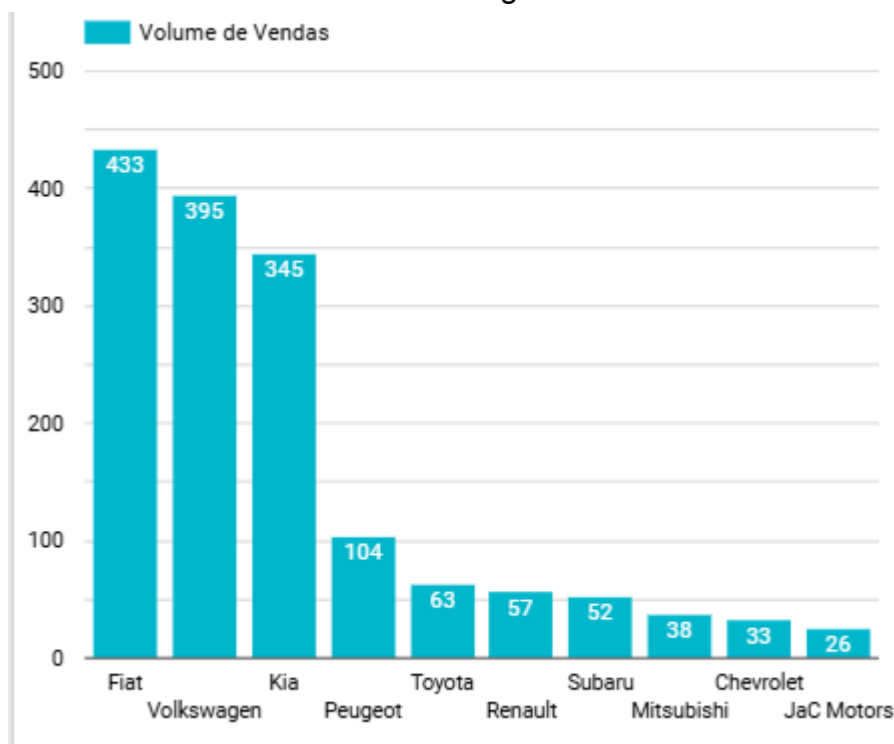
Google Sheets: [📊 Case Data Analytics - Monks](#)

Looker Studio: [Case Data Analytics - Monks](#)

Resolução das Questões

Questão 1 - Qual marca teve o maior volume de vendas?

De acordo com os dados foi feita a contagem de quantidade de vendas e a marca que teve o maior volume seria a Fiat tendo um total de 433 vendas realizadas conforme mostrado na tabela e no gráfico abaixo.







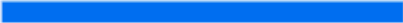

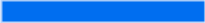






MARCA	NOME	Volume de Vendas
+ Fiat Total		433
+ Volkswagen Total		395
+ Kia Total		345
+ Peugeot Total		104
+ Toyota Total		63
+ Renault Total		57
+ Subaru Total		52
+ Mitsubishi Total		38
+ Chevrolet Total		33
+ JaC Motors Total		26
+ Nissan Total		23
Total geral		1569

Questão 2 - Qual veículo gerou a maior e menor receita?














Para saber a resposta dessa questão foi necessário criar uma nova coluna que foi denominada “Valor Total de Venda”, nela seria multiplicado o valor do veículo(coluna D) pela quantidade de veículos vendidos(coluna C), resultando assim na receita

total, essa função foi feita também diretamente no Looker Studio utilizando o recurso “Adicionar um campo calculado”, resultando assim no Mobi como o veículo que gerou a maior receita e o 307 como o que gerou a menor.

NOME	Valor Total Vendido
Mobi	R\$ 14.747.000,00
Up	R\$ 14.368.000,00
Picanto	R\$ 13.243.000,00
Forester	R\$ 9.800.000,00
208	R\$ 7.425.000,00
Corolla	R\$ 4.368.000,00
WRX	R\$ 3.750.000,00
Pajero	R\$ 2.680.000,00
L200	R\$ 2.580.000,00
onix	R\$ 1.418.400,00
XV	R\$ 1.280.000,00
Brz	R\$ 1.200.000,00
Yaris	R\$ 1.125.000,00
Lancer	R\$ 990.000,00
Clio	R\$ 979.000,00
E-JS1	R\$ 870.000,00
March	R\$ 601.000,00
2008	R\$ 570.000,00
Duster	R\$ 460.000,00
Sandero	R\$ 374.000,00
Saveiro	R\$ 360.000,00
Captur	R\$ 350.000,00
Cronos	R\$ 320.000,00
Gol	R\$ 288.000,00
E-J7	R\$ 270.000,00
Cerato	R\$ 254.000,00
Sandero RS	R\$ 240.000,00
Uno	R\$ 191.000,00
T-Cross	R\$ 180.000,00
J2	R\$ 165.000,00
argo	R\$ 154.000,00
Kombi	R\$ 125.000,00
J5	R\$ 125.000,00
Eclipse	R\$ 124.000,00
Jetta	R\$ 120.000,00
Polo	R\$ 99.000,00
Rio	R\$ 89.000,00
206	R\$ 72.000,00
Palio	R\$ 35.000,00
307	R\$ 19.000,00
Total geral	R\$ 86.408.400,00

NOME	Receita Total ▾
Mobi	14,7 mi 
Up	14,4 mi 
Picanto	13,2 mi 
Forester	9,8 mi 
208	7,4 mi 
Corolla	4,4 mi 
WRX	3,8 mi 
Pajero	2,7 mi 
L200	2,6 mi 
onix	1,4 mi 
XV	1,3 mi 
Brz	1,2 mi 
Yaris	1,1 mi 

1 - 40 / 40 < >

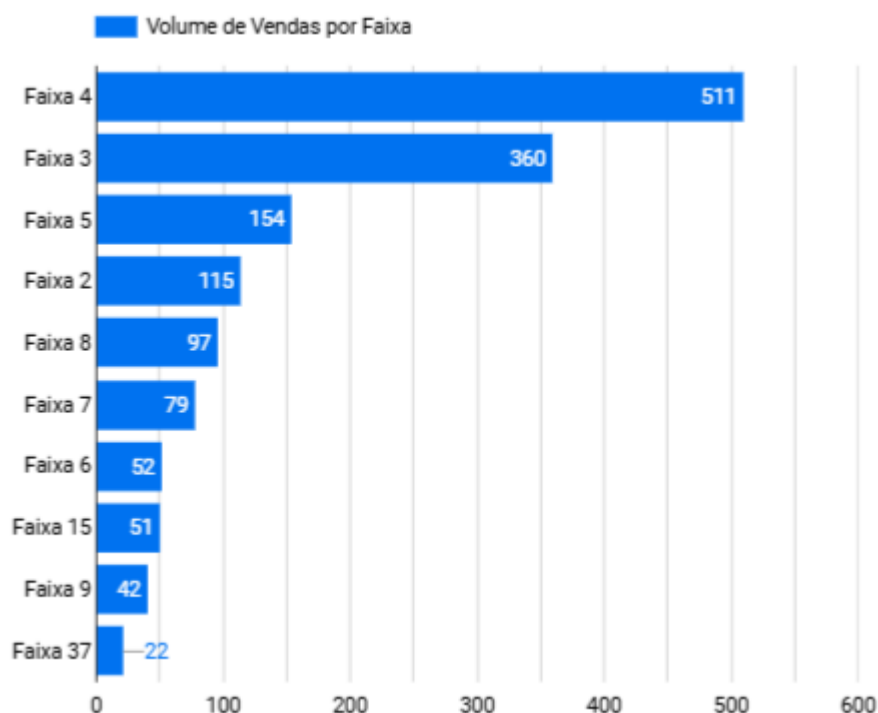
NOME	Receita Total ▾
Uno	191 mil 
T-Cross	180 mil 
J2	165 mil 
argo	154 mil 
Kombi	125 mil 
J5	125 mil 
Eclipse	124 mil 
Jetta	120 mil 
Polo	99 mil 
Rio	89 mil 
206	72 mil 
Palio	35 mil 
307	19 mil 

1 - 40 / 40 < >

Questão 3 - Considere faixas de preço de venda dos carros a cada 10 mil reais. Qual faixa mais vendeu carros? Quantos?

Essa foi uma das mais difíceis de ser feita, tentei pesquisar uma forma de fazer diretamente pela tabela dinâmica ou Looker Studio sem ter que escrever uma fórmula gigante mas não encontrei este resultado, sendo assim fiz um grande if que foi contando a cada R\$ 10.000 e classificando o valor em uma faixa que iria de 1 até 37, a faixa que mais vendeu carros foi a de número 4 tendo vendido no total 511 carros.

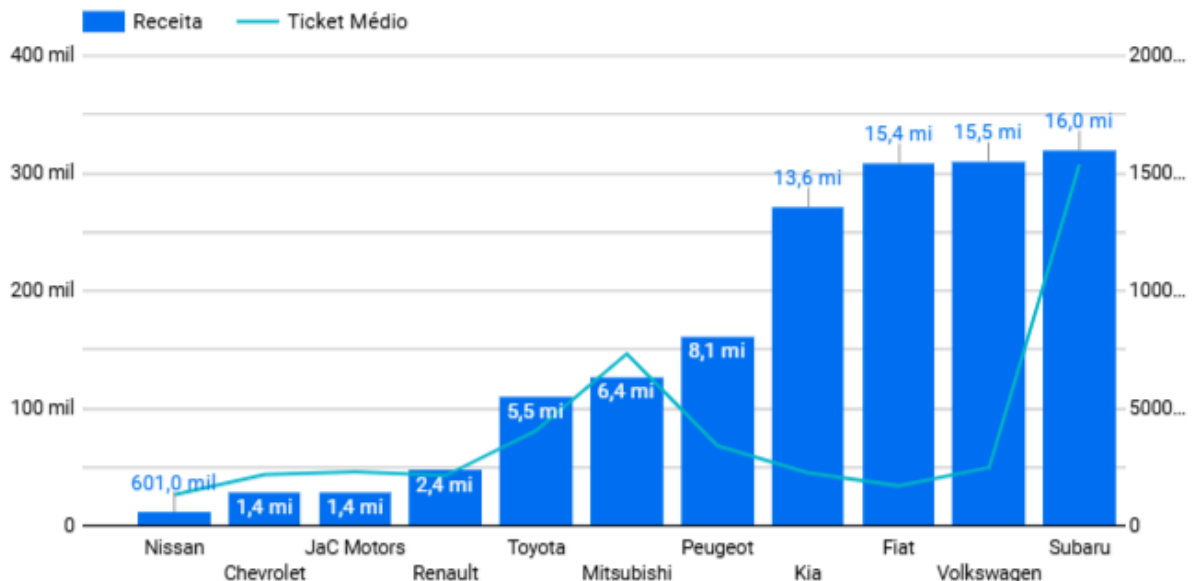
FAIXA VALOR	Vendas por Faixa
Faixa 4	511
Faixa 3	360
Faixa 5	154
Faixa 2	115
Faixa 8	97
Faixa 7	79
Faixa 6	52
Faixa 15	51
Faixa 9	42
Faixa 37	22
Faixa 10	19
Faixa 26	15
Faixa 25	12
Faixa 33	8
Faixa 17	8
Faixa 16	6
Faixa 13	6
Faixa 31	5
Faixa 29	5
Faixa 28	1
Faixa 1	1
Total geral	1569



Questão 4 - Qual a receita das 3 marcas que têm os menores tickets médios?

Para verificar a receita teria que observar o valor total vendido que se encontra na coluna E, já o ticket médio teria que fazer a média referente ao valor do veículo com relação às marcas, resultando assim na Nissan com um ticket médio de R\$ 26.428,57 e uma receita de R\$ 601.000,00, em segundo lugar a Fiat com um Ticket Médio de R\$ 33.954,55 e uma receita de R\$ R\$ 15.447.000,00 e em terceiro lugar estaria a Renault com um Ticket Médio de R\$ 42.666,67 e uma receita de R\$ 2.403.000,00.

MARCA	Receita por Marca	Ticket Médio
Nissan	R\$ 601.000,00	R\$ 26.428,57
Fiat	R\$ 15.447.000,00	R\$ 33.954,55
Renault	R\$ 2.403.000,00	R\$ 42.666,67
Chevrolet	R\$ 1.418.400,00	R\$ 43.355,56
Kia	R\$ 13.586.000,00	R\$ 45.066,67
JaC Motors	R\$ 1.430.000,00	R\$ 45.750,00
Volkswagen	R\$ 15.540.000,00	R\$ 49.476,19
Peugeot	R\$ 8.086.000,00	R\$ 68.111,11
Toyota	R\$ 5.493.000,00	R\$ 80.875,00
Mitsubishi	R\$ 6.374.000,00	R\$ 146.400,00
Subaru	R\$ 16.030.000,00	R\$ 307.142,86
Total geral	R\$ 86.408.400,00	R\$ 67.971,21



Questão 5 - Existe alguma relação entre os veículos mais vendidos?

Observando apenas os dados é possível perceber que os veículos mais vendidos seriam o Mobi da Fiat possuindo 414 vendas, o Up da Volkswagen com 373 vendas e o Picanto da Kia com 338 vendas, o que eles possuem em comum seria o preço e a faixa em que se encontram, ambos os três carros se encontram na faixa 3 ou 4 e custo em torno de R\$ 40.000, poderia também ser feita uma análise mais profunda para validar o motivo do porque esses carros em específicos de marcas discrepantes estariam sendo os mais vendidos tendo alguns outros carros mais baratos, poderia ser devido a ser um carro popular e ter uma alta demanda ou um melhor custo benefício porém não é algo que dá para confirmar apenas com os dados que possuímos no momento.