Esse código realiza um **INNER JOIN** entre a tabela proprietária e a tabela carro. A escolha por um **INNER JOIN** foi feita porque não cabe nulos nesse cenário porque não existem nulos de nenhum lado. Usei a cláusula Where para selecionar apenas os carros Sedan. Esse código seleciona o id e o nome dos donos de carros Sedan.

**select idProprietario, nome as NomeCompleto from proprietario**

**inner join carro on carro.proprietario\_idProprietario = proprietario.idProprietario**

**where Categoria = 'Sedan';**

Esse código faz um **INNER JOIN** também entre a tabela carro e a tabela modelo, usei um **INNER JOIN** por não ter nulos de nenhum lado. Utilizei a cláusula Where para selecionar apenas os carros com motor novo. Ele seleciona o ano e o motor dos carros novos.

**select ano, motor as Motor from carro**

**inner join modelo on modelo.carro\_idcarro = carro.idcarro**

**where Descrição = 'Novo';**

Esse código faz um **INNER JOIN** entre a tabela modelo e a tabela carro, usei um **INNER JOIN** por não ter nulos de nenhum lado. Utilizei a cláusula Where para selecionar apenas os carros com ano maior que 2010. Ele seleciona as características dos carros do ano maior que 2010.

**select ano, características as Caracteristicas from modelo**

**inner join carro on carro.idcarro = modelo.carro\_idCarro**

**where Ano > 2010;**

Esse código faz um **INNER JOIN** entre a tabela fabricante e a tabela modelo, usei um **INNER JOIN** por não ter nulos de nenhum lado. Utilizei a cláusula WHERE para selecionar apenas os carros seminovos ou novos. Ele seleciona a descrição e o nome do fabricante onde a descrição for nova ou semi.

**select Descrição, Nome\_do\_Fabricante as Fabricante from fabricante**

**inner join modelo on modelo.idmodelo = fabricante.modelo\_idmodelo**

**where Descrição like 'Semi' or Descrição like novo;**

Esse código faz um **INNER JOIN** entre a tabela carro e a tabela placa, usei um **INNER JOIN** por não ter nulos de nenhum lado. Utilizei a cláusula WHERE para selecionar apenas os carros com a placa específica. Ele seleciona a placa e o ano dos carros onde a placa for igual a 'FXJ-0618'.

**select placa, ano as Ano from carro**

**inner join placa on placa.idplaca = carro.placa\_idplaca**

**where placa like 'FXJ-0618';**

Neste código utilizei a cláusula **ORDER BY** na tabela proprietário. Este código seleciona o nome e o estado da tabela proprietário onde o nome termina com A, ordenando pelo estado e em ordem ascendente.

**select nome,estado from proprietario where nome like '%a' order by estado ASC;**

Neste código utilizei a cláusula **ORDER BY** na tabela chassi. Este código seleciona idchassi e o número do chassi da tabela chassi onde os números do chassi terminam com 6, ordenando pelo número e em ordem ascendente.

**select idchassi,numero\_do\_chassi from chassi where Numero\_do\_chassi like '%6' order by numero\_do\_chassi ASC;**

Neste código utilizei a cláusula **ORDER BY** na tabela carro. Este código seleciona idcarro,ano e categoria dos carros que está entre os anos de 2005 e 2010, ordenando pelo id em ordem descendente.

**select idcarro,ano, categoria from carro where ano between '2005' and '2010 'order by idcarro DESC;**

Neste código utilizei a cláusula **ORDER BY** na tabela opcionais. Este código seleciona os carros de cor azul ou preto e verifica se tem ar, ordenando pela pintura e em ordem ascendente.

**select distinct pintura,Ar\_condicionado as 'Tem Ar?' from opcionais where pintura like 'Azul' or pintura like 'Preto' order by pintura ASC;**

Neste código utilizei a cláusula **ORDER BY** na tabela placa. Este código seleciona id e placa que começam com I, ordenando pelo id e em ordem ascendente.

**select idplaca,placa as 'Placas RS' from placa where placa like 'I%' order by idplaca ASC;**

Neste código utilizei as cláusulas **GROUP BY e HAVING** na tabela proprietário. Este código conta os estados e só retorna qual a soma for maior que dois, agrupando pelo estado. Assim retorna os estados que aparecem mais de 2 vezes.

**select estado as 'Estado com mais pessoas', COUNT (estado) as 'Quantas vezes aparece'**

**FROM proprietario GROUP BY estado HAVING COUNT (estado) > 2;**

Neste código utilizei as cláusulas **GROUP BY** na tabela fabricante. Este código conta os idfabricante, agrupando pelo nome do fabricante. Assim retorna o número de concessionárias.

**select nome\_do\_fabricante, COUNT (idFabricante)**

**as 'Quantidade de Concessionarias'**

**from fabricante**

**GROUP BY nome\_do\_fabricante;**

Neste código utilizei as cláusulas **COUNT**, **GROUP BY e o HAVING** na tabela opcionais. Este código conta o número de pinturas, agrupando a pintura. Assim retorna a pintura que aparece mais de uma vez.

**select pintura as 'pintura que aparece mais de uma vez’, COUNT(pintura) as 'quantas vezes aparece' from opcionais**

**GROUP BY pintura having count(pintura) > 2;**

Neste código utilizei as cláusulas **GROUP BY e HAVING** na tabela descrição. Este código verifica a descrição e só retorna qual a descrição for novo e por fim retorna tudo dos carros novos agrupando pelo idModelo e em ordem ascendente.

**select \* from modelo GROUP BY idModelo having Descrição = 'Novo'**

**ORDER BY idModelo ASC;**

Neste código utilizei as cláusulas **ORDER BY**, **GROUP BY e o HAVING** na tabela carro. Este código verifica a categoria e só retorna qual a categoria for Sedan ou hatch e acima do ano de 2006 e por fim retorna Idcarro, ano, categoria dos carros Sedan agrupando pelo idCarro, ordenando pelo ano e em ordem ascendente.

**select idCarro,ano,categoria from carro GROUP BY idCarro having ano > 2006 and (Categoria = 'Sedan' or categoria = 'Hatch' ) ORDER BY ano ASC ;**

Esse código faz um **INNER JOIN**, **ORDER BY e GROUP BY** entre a tabela carro e a tabela opcionais, usei um **INNER JOIN** por não ter nulos de nenhum lado. Ele seleciona o ano, categoria dos carros onde o ano for maior ou igual que 2005, agrupando pela categoria e ordenando pelo ano.

**select ano,Categoria from carro**

**INNER JOIN opcionais ON opcionais.idOpcionais = carro.opcionais\_idopcionais where ano >= 2005**

**GROUP BY categoria**

**ORDER BY Ano**

Esse código faz um **INNER JOIN** e **ORDER BY** entre a tabela modelo e a tabela carro, usei um **INNER JOIN** por não ter nulos de nenhum lado. Ele seleciona a descrição, características dos carros onde a descrição for usada ou semi, ordenando pela descrição.

**select Descrição,caracteristicas from modelo**

**inner join carro on carro.idCarro = modelo.carro\_idCarro where Descrição = 'Usado' or Descrição = 'Semi'**

**ORDER BY Descricao;**

Esse código faz um **INNER JOIN** e **ORDER BY** entre a tabela placa e a tabela carro, usei um **INNER JOIN** por não ter nulos de nenhum lado. Ele seleciona tudo da tabela placa onde o estampador for Mercosul e o ano for 2011, ordenando pela placa ascendente.

**select \* from placa**

**inner join carro on carro.placa\_idPlaca = placa.idPlaca where Estampador like 'Mercosul' and ano like 2011**

**order by placa ASC;**

Esse código faz um **INNER JOIN** e **ORDER BY** entre a tabela chassi e a tabela carro, usei um **INNER JOIN** por não ter nulos de nenhum lado. Ele seleciona tudo da tabela chassi onde o número do chassi começa com 6 e o motor for turbo, ordenando pelo id ascendente.

**select \* from chassi**

**inner join carro on carro.Chassi\_idChassi = chassi.idChassi where Numero\_do\_chassi like '6%' and motor = 'Turbo'**

**order by idChassi ASC;**

Esse código faz um **INNER JOIN** e **ORDER BY** entre a tabela proprietário e a tabela carro, usei um **INNER JOIN** por não ter nulos de nenhum lado. Ele seleciona o idProprietario e o nome da tabela proprietário onde o nome não começa com a letra A, ordenando pelo nome descendente.

**select idProprietario,nome from proprietario**

**inner join carro on carro.proprietario\_idProprietario = proprietario.idProprietario where not nome like '%a'**

**order by nome DESC**

Neste código utilizei a cláusula **AVG** na tabela carro, usei um alias para facilitar o entendimento. A cláusula **AVG** retorna a média dos anos dos carros, então esse código retorna a média dos anos de todos os carros.

**SELECT AVG(Ano) AS MediaAno FROM carro;**

Neste código utilizei a cláusula **MAX** na tabela custo. A cláusula **MAX** retorna o valor máximo, logo esse código retorna o maior valor de custo.

**SELECT MAX(Valor\_de\_custo) 'maximo valor de custo' FROM custo;**

Neste código utilizei a cláusula **MIN** na tabela custo. A cláusula **MIN** é usada para retornar o valor mínimo. Então esse código retorna a retornar o valor mínimo.

**SELECT MIN(Valor\_de\_custo) 'minimo valor de custo' FROM custo;**

Neste código utilizei a cláusula **HEX** na tabela proprietário, usei um alias para facilitar o entendimento. A cláusula **HEX** é usada para retornar uma representação hexadecimal, então esse código retorna a representação hexadecimal do cpf onde o id é igual a 5.

**SELECT HEX(cpf) as 'representacao Hexadecimal ' from proprietario where idProprietario = 5;**

Neste código utilizei a cláusula **COUNT** na tabela custo, usei um alias para facilitar o entendimento. A cláusula **COUNT** é usada para contar a quantidade de itens, logo esse código retorna a quantidade de vendas.

**SELECT COUNT(idCusto) as 'numero de vendas' FROM custo;**

Neste código utilizei a cláusula **SUM** na tabela custo, usei um alias para facilitar o entendimento. A cláusula **SUM** retorna a soma, então esse código retorna a soma do custo total.

**select SUM(Valor\_de\_Custo) as 'Custo Total ' from custo ;**

Neste código vou realizar uma subtração entre duas colunas, usei um alias para facilitar o entendimento, logo esse código retorna o lucro onde o id é igual a 1.

**SELECT (Valor\_de\_Venda - Valor\_de\_custo) as 'lucro' from custo where idCusto = 1;**