--A. Adicionando dados a tabela de categoria

set search\_path = exp;

INSERT INTO categoria VALUES (1, 'Hatch compacto', 'true', 'Hatch');

INSERT INTO categoria VALUES (2, 'Sedã compacto', 'true', 'Sedã');

INSERT INTO categoria VALUES (3, 'Sedã médio', 'false', 'Sedã');

INSERT INTO categoria VALUES (4, 'Picape grande', 'false', 'Picape');

INSERT INTO categoria VALUES (5, 'SUV compacto', 'true', 'SUVs');

INSERT INTO categoria VALUES (6, 'SUV compacto', 'true', 'SUVs');

--adicionando dados a tabela de cliente

set search\_path = exp;

INSERT INTO cliente VALUES (1, 'Paulo', 'NH', '99999-9999', '1989-02-01');

INSERT INTO cliente VALUES (2, 'Solange', 'IV', '99999-9999', '1990-03-02');

INSERT INTO cliente VALUES (3, 'Pedro', 'SL', '99999-9999', '1991-04-11');

INSERT INTO cliente VALUES (4, 'Jorge', 'POA', '99999-9999', '1992-03-06');

INSERT INTO cliente VALUES (5, 'Miguel', 'NH', '99999-9999', '1993-06-03');

INSERT INTO cliente VALUES (6, 'Fernando', 'NH', '99999-9999', '2000-07-11');

set search\_path = exp;

--adicionando dados a tabela de montadora

INSERT INTO montadora VALUES (1, 'Renault', 'Brasil', '2022-01-31');

INSERT INTO montadora VALUES (2, 'Fiat', 'Itália', '2023-02-11');

INSERT INTO montadora VALUES (3, 'Chevrolet', 'EUA', '2024-01-10');

INSERT INTO montadora VALUES (4, 'Jeep', 'EUA', '2020-04-03');

INSERT INTO montadora VALUES (5, 'Citroen', 'França', '2022-02-17');

INSERT INTO montadora VALUES (6, 'Citroen', 'França', '2022-02-17');

set search\_path = exp;

--adicionando dados a tabela de vendedor

INSERT INTO vendedor VALUES (1, 'Ricardo', 'ricardo01@gmail.com', '2012-02-01');

INSERT INTO vendedor VALUES (2, 'Rogério', 'rogerio01@gmail.com', '2000-01-01');

INSERT INTO vendedor VALUES (3, 'Maria', 'maria01@gmail.com', '2020-05-01');

INSERT INTO vendedor VALUES (4, 'Pedro', 'pedro01@gmail.com', '2020-05-01');

INSERT INTO vendedor VALUES (5, 'Marcos', 'marcos01@gmail.com', '2019-10-01');

INSERT INTO vendedor VALUES (6, 'Vinícius', 'vinicius01@gmail.com', '2010-10-23');

set search\_path = exp;

--adicionando dados a tabela de locacao

INSERT INTO locacao VALUES (1, 'Paulo', 209.00, 1);

INSERT INTO locacao VALUES (2, 'Solange', 328.00, 2);

INSERT INTO locacao VALUES (3, 'Pedro', 120.00, 3);

INSERT INTO locacao VALUES (4, 'Jorge', 623.00, 4);

INSERT INTO locacao VALUES (5, 'Miguel', 328.00, 5);

INSERT INTO locacao VALUES (6, 'Fernando', 428.00, 6);

set search\_path = exp;

--adicionando valores a tabela de compra

INSERT INTO compra VALUES (1, '2019-05-23', 69990.00, 1, 1);

INSERT INTO compra VALUES (2, '2019-05-23', 79990.00, 2, 2);

INSERT INTO compra VALUES (3, '2019-05-23', 80990.00, 3, 3);

INSERT INTO compra VALUES (4, '2019-05-23', 157000.00, 4, 4);

INSERT INTO compra VALUES (5, '2019-05-23', 83000.00, 5, 5);

INSERT INTO compra VALUES (6, '2020-01-03', 83000.00, 6, 6);

set search\_path = exp;

--adicionando valores a tabela de carro

INSERT INTO carro VALUES (1, 'Mobi Like', 2018, 69990.00, 1, 1, 1, 1, 1);

INSERT INTO carro VALUES (2, 'Argo', 2018, 79990.00, 2, 2, 2, 2, 2);

INSERT INTO carro VALUES (3, 'Onix Plus', 2019, 80990.00, 3, 3, 3, 3, 3);

INSERT INTO carro VALUES (4, 'Gladiator', 2017, 157000.00, 4, 4, 4, 4, 4);

INSERT INTO carro VALUES (5, 'C4 Cactus', 2019, 83000.00, 5, 5, 5, 5, 5);

INSERT INTO carro VALUES (6, 'C4 Cactus', 2019, 83000.00, 6, 6, 6, 6, 6);

-- B. Consultando os dados da uma tabela

SELECT \*

FROM carro;

-- C. Consultando os dados das tabelas carro e categoria usando o equi-join

SELECT

carro.id\_carro,

carro.modelo,

carro.ano,

carro.preco,

carro.id\_categoria,

categoria.id\_categoria,

categoria.nome,

categoria.cambio,

categoria.tipo

FROM carro

INNER JOIN categoria ON carro.id\_categoria = categoria.id\_categoria;

-- D. Consultando os dados das tabelas carro e montadora usando o inner-join

SELECT \*

FROM carro

INNER JOIN montadora

ON carro.id\_montadora = montadora.id\_montadora;

--E. Consultando os dados das tabelas compra, carro e montadora usando o equi-join

SELECT

compra.id\_compra,

compra.data\_compra,

compra.valor,

carro.modelo,

carro.ano,

montadora.nome AS montadora\_nome

FROM

compra

INNER JOIN carro ON compra.id\_carro = carro.id\_carro

INNER JOIN montadora ON carro.id\_montadora = montadora.id\_montadora;

--F. Consultando os dados das tabelas compra, carro e vendedor inner-join

SELECT

compra.id\_compra,

compra.data\_compra,

compra.valor,

carro.modelo,

carro.ano AS ano\_fabricação,

vendedor.nome AS nome\_vendedor

FROM

compra

INNER JOIN carro ON compra.id\_carro = carro.id\_carro

INNER JOIN vendedor ON compra.id\_vendedor = vendedor.id\_vendedor;

--G. Consultando os dados das tabelas, carro, categoria, montadora e cliente

SELECT

carro.modelo,

carro.ano AS ano\_fabricação,

categoria.nome AS nome\_categoria,

montadora.nome AS nome\_montadora,

cliente.nome AS quem\_comprou

FROM

carro

INNER JOIN categoria ON carro.id\_categoria = categoria.id\_categoria

INNER JOIN montadora ON carro.id\_montadora = montadora.id\_montadora

INNER JOIN cliente ON carro.id\_cliente = cliente.id\_cliente;

--H. Consultando se os precos registrados nas tabelas compra e carro que não são nulos

SELECT

compra.id\_compra,

compra.data\_compra,

compra.valor AS valor\_venda,

carro.preco AS preco\_carro

FROM

compra

INNER JOIN carro ON compra.id\_carro = carro.id\_carro

WHERE

compra.valor IS NOT NULL AND carro.preco IS NOT NULL;

--I. Consultando o valor da venda de carros cujo modelo terminam com a letra s - envolvendo 2 tabelas

SELECT

carro.modelo,

compra.valor AS valor\_da\_venda

FROM

carro

INNER JOIN compra ON compra.id\_carro = carro.id\_carro

WHERE

carro.modelo LIKE '%s';

--J. Consultando o valor de venda de carros que tenham apenas 4 letras em seu nome(modelo)

SELECT

carro.modelo,

compra.valor AS valor\_da\_venda

FROM

carro

INNER JOIN compra ON compra.id\_carro = carro.id\_carro

WHERE

carro.modelo LIKE '\_\_\_\_';

-- K. Consultando modelo, ano, preco e id da montadora dos carros que estejam dentro

-- da tabela de compra e que tenham o id\_compra diferente de 2

SELECT

carro.modelo,

carro.ano,

carro.preco,

carro.id\_montadora

FROM

carro

WHERE

carro.id\_carro IN (

SELECT

compra.id\_carro

FROM

compra

WHERE id\_compra <> 2

);

--L. Consultando qual carro possui valor maior ou igual a maior compra realizada

SELECT

carro.modelo,

carro.ano,

carro.preco

FROM

carro

WHERE

carro.preco >= ALL (

SELECT

compra.valor

FROM

compra

);

--M. Consultando se para cada carro existe uma compra relacionada

SELECT

carro.modelo,

carro.ano,

carro.preco

FROM

carro

WHERE

EXISTS (

SELECT 1

FROM

compra

WHERE

compra.id\_carro = carro.id\_carro

);

--N. Consultando o nome de todos os clientes e vendedores

SELECT

nome

FROM

cliente

UNION

SELECT

nome

FROM

vendedor;

--O. Soma o valor total de vendas de cada modelo de carro

SELECT

c.modelo,

SUM(co.valor) AS total\_valor\_compras

FROM

carro c

LEFT JOIN compra co ON c.id\_carro = co.id\_carro

GROUP BY

c.modelo;

--P. Consultando todas as vendas por modelo que tiveram valor acima de 100000

SELECT

c.modelo,

SUM(co.valor) AS total\_valor\_compras

FROM

carro c

LEFT JOIN compra co ON c.id\_carro = co.id\_carro

GROUP BY

c.modelo

HAVING

SUM(co.valor) > 100000;

--Q. Deletando uma compra / não rodar a query que vai dar ruim

--devido a forma que foi criado as tabela

SELECT \*

FROM compra

WHERE id\_cliente = 5;

DELETE FROM compra

WHERE id\_cliente = 5;

--R. Atualizando valores e ano na tabela carro

UPDATE carro

SET

preco = 75000.00,

ano = 2022

WHERE

id\_carro = 1;

--S. Excluindo a tabela locacao

DROP TABLE locacao;

--T. Criando uma view para as tabelas compra, carro e montadora

CREATE VIEW vw\_compra\_carro\_montadora AS

SELECT

c.id\_compra,

c.data\_compra,

c.valor,

ca.id\_carro,

ca.modelo,

ca.ano,

ca.preco,

ca.id\_montadora,

m.nome AS nome\_montadora

FROM

compra c

JOIN

carro ca ON c.id\_carro = ca.id\_carro

JOIN

montadora m ON ca.id\_montadora = m.id\_montadora;

SELECT \* FROM vw\_compra\_carro\_montadora;