--1. Qual a quantidade de endereços por estado?

-- Seleciona a coluna 'uf' e usa a função COUNT para contar quantas linhas estão no resultado

-- da tabela estados

SELECT estados.uf, COUNT(\*)

FROM estados

-- junção entre as tabelas estados e cidades usando a coluna uf

JOIN cidades ON estados.uf = cidades.uf

-- junção entre as tabelas cidades e enderecos usando a coluna cod\_cidade

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Agrupa os dados com base nos valores únicos na coluna uf

GROUP BY estados.uf;

--2. Qual a quantidade de clientes de cada estado?

-- Seleciona a coluna 'uf' e usa a função COUNT para contas quantos clientes distintos

-- da tabela estados

SELECT estados.uf, COUNT(DISTINCT clientes.cod\_cliente)

FROM estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e clientes usando a colua 'cod\_cliente'

-- Relaciona cada cliente ao seu endereço correspondente

JOIN clientes ON clientes\_enderecos.cod\_cliente = clientes.cod\_cliente

-- Agrupa os resultados com base nos valores exclusivos da coluna 'uf' na tabela estados

GROUP BY estados.uf;

--3. Qual o ranking de estados por quantidade de clientes, ou seja, em ordem decrescente de quantidade de clientes?

-- Seleciona a coluna 'uf' e usa a função COUNT para contas quantos clientes distintos

-- da tabela estados

SELECT estados.uf, COUNT(DISTINCT clientes.cod\_cliente)

FROM estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e clientes usando a colua 'cod\_cliente'

-- Relaciona cada cliente ao seu endereço correspondente

JOIN clientes ON clientes\_enderecos.cod\_cliente = clientes.cod\_cliente

-- Agrupa os resultados com base nos valores exclusivos da coluna 'uf' na tabela estados

GROUP BY estados.uf

-- Usada para ordenar os resultados em ordem decrescente com base no número de clientes

-- distintos em cada estado.

ORDER BY COUNT(DISTINCT clientes.cod\_cliente) DESC;

--4. Qual o ranking de regiões por quantidade de clientes, ou seja, em ordem decrescente de quantidade de clientes?

-- Seleciona a coluna região e conta quantos clientes distintos estão associados a cada região

-- da tabela estados

SELECT estados.regiao, COUNT(DISTINCT clientes.cod\_cliente)

FROM estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e clientes usando a colua 'cod\_cliente'

-- Relaciona cada cliente ao seu endereço correspondente

JOIN clientes ON clientes\_enderecos.cod\_cliente = clientes.cod\_cliente

-- Agrupa os resultados com base nos valores exclusivos da coluna 'região' na tabela estados

GROUP BY estados.regiao

-- Usada para ordenar os resultados em ordem decrescente com base no número de

-- clientes distintos em cada região

ORDER BY COUNT(DISTINCT clientes.cod\_cliente) DESC;

--5. Qual a quantidade de pedidos por região?

-- Seleciona a coluna região e usa a função COUNT para contar quantos pedidos existem

-- em cada região da tabela estados

SELECT regiao, COUNT(pedidos.num\_pedido)

FROM estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e pedidos usando as colunas 'cod\_cliente'

-- e 'cod\_endereco'. Relaciona cada pedido aos seus clientes e endereços correspondentes

JOIN pedidos ON clientes\_enderecos.cod\_cliente = pedidos.cod\_cliente AND clientes\_enderecos.cod\_endereco = pedidos.cod\_endereco

-- Agrupa os resultados com base nos valores exclusivos da coluna região na tabela estados

GROUP BY regiao

-- Usada para ordenar os resultados em ordem decrescente com base no número de pedidos

-- em cada região

ORDER BY COUNT(pedidos.num\_pedido) DESC;

--6. Qual a quantidade de pedidos por ano e por região, considerando apenas os pedidos feitos nos anos de 2000 até 2004?

-- Seleciona TO\_CHAR que formata a data de emissão dos pedidos no formato de ano

-- Seleciona a coluna região e a função COUNT para contar quantas fezes o número de pedidos

-- aparece para cada combinação de ano e região da tabela estados

SELECT TO\_CHAR(pedidos.data\_emissao, 'yyyy'), estados.regiao, COUNT(pedidos.num\_pedido)

FROM estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e pedidos usando as colunas 'cod\_cliente'

-- e 'cod\_endereco'. Relaciona cada pedido aos seus clientes e endereços correspondentes

JOIN pedidos ON clientes\_enderecos.cod\_cliente = pedidos.cod\_cliente AND clientes\_enderecos.cod\_endereco = pedidos.cod\_endereco

-- Filtra os resultados os resultados apenas para pedidos emitidos entre os anodes de 2000 e 2004

-- Converte a data de emissão para o formato de ano

WHERE TO\_CHAR(pedidos.data\_emissao, 'yyyy') BETWEEN '2000' AND '2004'

-- Agrupa os resultados com base nos valores exclusivos da data de emissão no formato de ano

-- e da coluna regiao na tabela estados.

-- Conta quantos pedidos foram emitidos em cada ano e região

GROUP BY TO\_CHAR(pedidos.data\_emissao, 'yyyy'), estados.regiao

-- Ordena os resultados primeiro por ano (em ordem crescente) e

-- por região (em ordem crescente)

ORDER BY TO\_CHAR(pedidos.data\_emissao, 'yyyy'), estados.regiao;

--7. Qual o valor total gasto por cliente, ordenado em ordem decrescente de valor total?

-- Seleciona a coluna cod\_cliente e calcula a somo total dos produtos para cada cliente

-- da tabela estados

select cod\_cliente, sum(quantidade \* valor\_unitario)

from estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e pedidos usando as colunas 'cod\_cliente'

-- e 'cod\_endereco'. Relaciona cada pedido aos seus clientes e endereços correspondentes

JOIN pedidos ON clientes\_enderecos.cod\_cliente = pedidos.cod\_cliente AND clientes\_enderecos.cod\_endereco = pedidos.cod\_endereco

-- Faz uma junção entre as tabelas pedidos e pedidos\_produtos usando a coluna 'num\_pedido'

-- Relaciona cada pedido aos produtos associados a ele.

JOIN pedidos\_produtos ON pedidos.num\_pedido = pedidos\_produtos.num\_pedido

-- Ordena os resultados em ordem decrescente com base no valor total gasto por cada cliente.

group by cod\_cliente

order by sum(quantidade \* valor\_unitario) desc;

--8. Qual o valor total gasto por cliente, ordenado em ordem decrescente de valor total, considerando apenas os clientes do Rio Grande do Sul?

-- Seleciona a coluna cod\_cliente e p valor total gasto por cada clientes em seus pedidos

select cod\_cliente, sum(quantidade \* valor\_unitario)

from estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e pedidos usando as colunas 'cod\_cliente'

-- e 'cod\_endereco'. Relaciona cada pedido aos seus clientes e endereços correspondentes

JOIN pedidos ON clientes\_enderecos.cod\_cliente = pedidos.cod\_cliente AND clientes\_enderecos.cod\_endereco = pedidos.cod\_endereco

-- Faz uma junção entre as tabelas pedidos e pedidos\_produtos usando a coluna 'num\_pedido'

-- Relaciona cada pedido aos produtos associados a ele.

JOIN pedidos\_produtos ON pedidos.num\_pedido = pedidos\_produtos.num\_pedido

-- Filtra os resultados apenas para clientes que estão no estado do RS.

where uf = 'RS'

-- Agrupa os resultados com base nos valores exclusivos da coluna cod\_cliente da tabela clientes

group by cod\_cliente

-- Ordena os resultados em ordem decrescente com base no valor total gasto por cada cliente

order by sum(quantidade \* valor\_unitario) desc;

--9. Qual o valor total vendido por autor?

-- Seleciona a coluna cod\_autor e calcula o valor total das vendas de produtos associados a cada autor

select cod\_autor, sum( quantidade \* valor\_unitario )

from autores

-- Faz uma junção entre as tabelas autores e autores\_produtos usando a coluna cod\_autor

-- Relaciona cada autor aos produtos associados a ele

JOIN autores\_produtos ON autores.cod\_autor = autores\_produtos.cod\_autor

-- Faz uma junção entre as tabelas autores\_produtos e produtos usando a coluna cod\_produto

-- Relaciona cada produto aos seus autores associados

JOIN produtos ON autores\_produtos.cod\_produto = produtos.cod\_produto

-- Faz uma junção entre as tabelas produtos e pedidos\_produtos usando a coluna cod\_produto

-- Relaciona cada produto às vendas associadas a ele

JOIN pedidos\_produtos ON produtos.cod\_produto = pedidos\_produtos.cod\_produto

-- Agrupar os resultados com base nos valores exclusivos da coluna cod\_autor da tabela autores

group by cod\_autor;

--10. Qual o valor médio faturado com as vendas por produto?

-- Seleciona a coluna cod\_produto e a média do valor total de vendas de cada produto

select cod\_produto, avg( quantidade \* valor\_unitario )

from produtos

-- Faz uma junção entre as tabelas produtos e pedidos\_produtos usando a coluna cod\_produto

-- Relaciona cada produto às vendas associadas a ele

JOIN pedidos\_produtos ON produtos.cod\_produto = pedidos\_produtos.cod\_produto

-- Agrupa os resultados com base nos valores exclusivos da coluna cod\_produto da tabela produtos

group by cod\_produto;

--11. Qual o valor total de cada pedido?

-- Seleciona a coluna num\_pedido e o valor total de cada pedido

select num\_pedido, sum( quantidade \* valor\_unitario )

from pedidos

-- Faz uma junção entre as tabelas pedidos e pedidos\_produtos usando a coluna num\_pedido

-- Relaciona cada pedido aos produtos associados a ele

JOIN pedidos\_produtos ON pedidos.num\_pedido = pedidos\_produtos.num\_pedido

-- Agrupa os resultados com base nos valores exclusivos da coluna num\_pedido da tabela pedidos

group by num\_pedido;

--12. Qual o valor médio dos pedidos por estado?

select uf, sum(quantidade \* valor\_unitario) / count(distinct num\_pedido)

from estados

-- Faz uma junção entre as tabelas cidades e estados usando a coluna 'uf'

-- Relaciona cada estado às suas cidades correspondentes

JOIN cidades ON estados.uf = cidades.uf

-- Faz uma junção entre as tabelas cidades e endereços usando a coluna 'cod\_cidade'

-- Relaciona cada cidade aos seus endereços correspondentes

JOIN enderecos ON cidades.cod\_cidade = enderecos.cod\_cidade

-- Faz uma junção entre as tabelas endereços e clientes\_endereços usando a coluna 'cod\_endereco'

-- Relaciona cada cidade aos seus endereços

JOIN clientes\_enderecos ON enderecos.cod\_endereco = clientes\_enderecos.cod\_endereco

-- Faz uma junção entre as tabelas clientes\_endereços e pedidos usando as colunas 'cod\_cliente'

-- e 'cod\_endereco'. Relaciona cada pedido aos seus clientes e endereços correspondentes

JOIN pedidos ON clientes\_enderecos.cod\_cliente = pedidos.cod\_cliente AND clientes\_enderecos.cod\_endereco = pedidos.cod\_endereco

-- Faz uma junção entre as tabelas pedidos e pedidos\_produtos usando a coluna 'num\_pedido'

-- Relaciona cada pedido aos produtos associados a ele

JOIN pedidos\_produtos ON pedidos.num\_pedido = pedidos\_produtos.num\_pedido

-- Agrupa os resultados com base nos valores exclusivos da coluna uf da tabela estados

group by uf;