1. Liste as informações de todas as categorias

SELECT \* FROM categorias;

1. Liste os nomes de todos os clientes.

SELECT nome FROM clientes;

1. Mostre o nome e o e-mail dos clientes que moram no estado "SP", no Brasil

SELECT nome, email FROM clientes WHERE estado = 'SP' AND pais = 'BR';

1. Quais as profissões de meus clientes?

SELECT DISTINCT profissao FROM clients;

1. Quais clientes têm "actor" como profissão?

SELECT id, nome, profissao FROM clientes WHERE lower(profissao) = 'actor';

1. Liste os produtos que estão descontinuados.

SELECT id, produto, descontinuado FROM produtos WHERE descontinuado = 1;

1. Quais produtos possuem precisam ser comprados, indique a quantidade que precisa ser comprada para ter o dobro da quantidade mínima?

SELECT id, produto, unidadesemestoque, unidadespedidas, nivelminimo, (2 \* nivelminimo) - (unidadesemestoque + unidadespedidas) AS quantidade\_a\_comprar FROM produtos WHERE (unidadesemestoque + unidadespedidas) < nivelminimo;

1. Liste os nomes e documentos dos vendedores do estado "RJ", no Brasil

SELECT nome, documento FROM vendedores WHERE estado = 'RJ' AND pais = 'BR';

1. Mostre os produtos com preço unitário superior a 100

SELECT produto, precounitario FROM produtos WHERE precounitario > 100;

1. Mostre os 5 produtos mais caros.
   * O limit não é padronizado sendo usazo no mysql use TOP 1 para SQL Server, ou FETCH FIRST 1 ROW ONLY para PostgreSQL/Oracle/DB2

SELECT produto, precounitario FROM produtos ORDER BY precounitario DESC LIMIT 5;

1. Liste os nomes dos clientes que nasceram depois de 01/01/2000

SELECT nome FROM clientes WHERE nascimento > '2000-01-01';

1. Quantos pedidos que foram pagos, mas ainda não enviados

SELECT COUNT(id) AS total\_pedidos\_pagos\_nao\_enviados FROM pedidos WHERE datadopagamento IS NOT NULL AND datadeenvio IS NULL;

1. Quantos pedidos foram entregues atrasados

SELECT COUNT(id) AS total\_pedidos\_atrasados FROM pedidos WHERE datadaentrega IS NOT NULL AND previsaoentrega IS NOT NULL AND datadaentrega > previsaoentrega;

1. Quantas venda foram feitas até o momento?

SELECT COUNT(id) AS total\_vendas\_pedidos FROM pedidos;

1. Qual a data do pedido mais antigo registrado?

SELECT MIN(datadopedido) AS data\_pedido\_mais\_antigo FROM pedidos;

1. Qual a data do cadastro de cliente mais novo?

SELECT MAX(nascimento) AS data\_cadastro\_cliente\_mais\_novo FROM clientes;

1. Qual o preço unitário médio de todos os produtos em nosso catálogo?

SELECT AVG(precounitario) AS preco\_medio\_produtos FROM produtos;

1. Qual o tempo médio de entrega (em dias) para todos os pedidos (da data de envio à data de entrega)
   * *O tratamento de datas não está bem padronizado nos diversos SGBs,* a função DATEDIFF é comum em MySQL e SQL Server. Para PostgreSQL, você pode usar EXTRACT(DAY FROM (datadaentrega - datadeenvio)). Para Oracle e DB2, datadaentrega - datadeenvio retorna a diferença em dias.

SELECT AVG(DATEDIFF(datadaentrega, datadeenvio)) AS tempo\_medio\_entrega\_dias FROM pedidos WHERE datadaentrega IS NOT NULL AND datadeenvio IS NOT NULL;

1. E o tempo médio de cada transportadora?

SELECT via,AVG(DATEDIFF(datadaentrega, datadeenvio)) AS tempo\_medio\_entrega\_dias FROM pedidos WHERE datadaentrega IS NOT NULL AND datadeenvio IS NOT NULL GROUP BY via;

1. Qual o preço médio de cada categoria?

SELECT idcategoria, AVG(p.precounitario) AS preco\_medio\_categoria FROM produtos GROUP BY idcategoria ORDER BY idcategoria;

1. Quantos clientes por identidade sexual?

SELECT sexo, COUNT(id) AS total\_clientes FROM clientes GROUP BY sexo ORDER BY sexo;

1. Quais são os e-mails de clientes que usam o domínio "gmail.com"?

SELECT id, nome, email FROM clientes WHERE lower(email) LIKE '%@gmail.com';

1. Liste todos os produtos "Smart"

SELECT id, produto, precounitario FROM produtos WHERE lower(produto) LIKE '%smart%';

1. Liste os produtos cujo nome tem exatamente 5 caracteres

SELECT produto FROM produtos WHERE produto LIKE '\_\_\_\_\_';

1. Liste os produtos cujo nome tem pelo menos 5 caracteres

SELECT produto FROM produtos WHERE produto LIKE '\_\_\_\_\_%';

1. Liste os produtos cujo nome tem no máximo 5 caracteres

SELECT produto FROM produtos WHERE produto not LIKE '\_\_\_\_\_\_%';

1. Mostre os clientes cujo CEP de moradia começa com "0" e o complemento do cep (3 últimos dígitos) também começe com "0".

SELECT id, nome,cep FROM clientes WHERE cep LIKE '0%0\_\_';

1. Quais clientes têm o sobrenome "Silva" em seu nome completo?

SELECT Count(\*) FROM clientes WHERE lower(nome) LIKE '% Silva %' OR lower(nome) LIKE '% Silva';

1. Liste as categorias e seus respectivos produtos ativos ordenado peno nome da categoria e pelo nome do produto

SELECT c.nomedacategoria,p.produto FROM categorias c LEFT JOIN produtos p ON c.id = p.idcategoria ORDER BY c.nomedacategoria, p.produto;

1. Considerando que todo cliente que fez pelo menos 10 pedidos é considerado VIP, list o nome e o E-mail de todos os clientes VIP

SELECT nome,email,COUNT(\*) FROM clientes c INNER JOIN pedidos p ON p.idcliente = c.id GROUP BY nome,email HAVING COUNT(\*) > 9;

1. Quais os clientes que só fizeram uma compra na nossa loja?

SELECT nome, email,COUNT(\*) FROM clientes c INNER JOIN pedidos p ON p.idcliente = c.id GROUP BY nome,email HAVING COUNT(\*) = 1;

1. Liste os fornecedores e os produtos que necessitam de serem comprados.

SELECT f.empresa AS nome\_fornecedor, p.produto AS nome\_produto, p.unidadesemestoque, p.unidadepedidas, p.nivelminimo FROM fornecedores AS f INNER JOIN produtos AS p ON f.id = p.idfornecedor WHERE (p.unidadesemestoque + unidadespedidas)< p.nivelminimo;

1. Liste os clientes e is produtos que ele comprou

SELECT nome,produto FROM cliente cl INNER JOIN pedidos pe ON pe.idcliente = cl.id INNER JOIN itens it ON ir.idpedido = pe.id INNER JOIN produtos pr ON it.idproduto = pr.id ORDER BY nome,produto

1. Quais os produtos com o preço unitário maior que o preço médio?

SELECT produto, precounitario FROM produtos WHERE precounitario > (SELECT AVG(precounitario) FROM produtos);

1. Quais os clientes que residem nos estados: 'Rio Grande do Sul', 'Santa Catarina' e 'Paraná'?

SELECT nome, email, cidade, estado FROM clients WHERE estado IN ('RS', 'SC', 'PR') AND PAIS = 'BR';

1. Quais itens foram incluídos em pedidos onde o frete foi superior a R$50 e a data do pedido foi entre '2020-01-15' e '2020-07-23'?

SELECT i.idproduto, pr.produto, i.quantidade, i.precounitario, p.datadopedido, p.frete FROM itens AS I INNER JOIN pedidos AS p ON i.idpedido = p.id INNER JOIN produtos AS pr ON i.idproduto = pr.id WHERE p.frete > 50 AND p.datadopedido BETWEEN '2020-01-15' AND '2020-07-23';

1. Liste os produtos que nunca foram pedidos.

SELECT produto, precounitario, unidadesemestoque FROM produtos WHERE id NOT IN (SELECT DISTINCT idproduto FROM itens);

1. Quais pedidos tiveram desconto aplicado e foram realizados por vendedores alemães?

SELECT ped.id AS id\_pedido, ped.datadopedido, ped.desconto, vend.nome AS nome\_vendedor, pais.pais AS pais\_vendedor FROM pedidos AS ped INNER JOIN vendedores AS vend ON ped.idvendedor = vend.id INNER JOIN paises AS pais ON vend.pais = pais.sigla WHERE ped.desconto > 0 AND pais.pais = 'Alemanha';

1. Liste o nome do supervisor e seus subordinados

SELECT Supervisor.nome AS NomeDoSupervisor, Subordinado.nome AS NomeDoSubordinado FROM vendedores AS Supervisor INNER JOIN vendedores AS Subordinado ON Supervisor.id = Subordinado.supervisor ORDER BY NomeDoSupervisor, NomeDoSubordinado;

1. Quais os produtos que têm o preço unitário superior ao preço médio de sua categoria?

SELEC produto,precounitario FROM produtos p WHERE precounitario > (SELECT AVG(precounitario) FROM produtos p1 where p1.idcategoria = p.idcategoria);

1. Qual o faturamento por país considerando os descontos?

select c.pais,sum(aux.valorbruto \* (1-pe.desconto)) from pedidos pe inner JOIN clientes c ON c.id = pe.idcliente INNER JOIN (SELECT idpedido,SUM(quantidade\*precounitario) as valorbruto FROM itens GROUP BY idpedido) aux ON pe.id = aux.idpedido GROUP BY c.pais;

1. Considerando o valor final do pedido qual vendedor tem a maior média de venda?

SELECT ve.nome,AVG(valorpedido) from vendedores ve INNER JOIN pedidos pe ON pe.idvendedor = ve.id INNER JOIN (SELECT pe.id,SUM(it.precounitario\*it.quantidade \* (1 - pe.desconto)) AS valorpedido FROM pedidos pe INNER JOIN itens it ON pe.id = it.idpedido GROUP BY pe.id) aux ON aux.id = pe.id GROUP BY ve.nome ORDER BY 2 DESC LIMIT 1;

1. Liste todas as pessoas que tem algum relacionamento com a empresa e qual é esse relacionamento

SELECT nome, 'Cliente' AS Tipo\_Contato FROM clientes UNION ALL SELECT nome, 'Vendedor' AS Tipo\_Contato FROM vendedores UNION ALL SELECT contato, 'Contato no Fornecedor' AS Tipo\_Contato FROM fornecedores WHERE contato IS NOT NULL AND contato != '' UNION ALL SELECT contato 'Contato na Transportadora' AS Tipo\_Contato FROM transportadoras WHERE contato IS NOT NULL AND contato != '';

1. Liste o produto e a evolução de seu preço ao longo do tempo

SELECT pr.produto, pr.precounitario, CURRENT\_DATE() as Data\_da\_utilização FROM produtos pr UNION SELECT pr.produto, it.precounitario, MIN(pe.datadopedido) FROM produtos pr INNER JOIN itens it ON pr.id = it.idproduto INNER JOIN pedidos pe ON it.idpedido = pe.id GROUP BY pr.produto,it.precounitario ORDER BY 1,3;

1. Quem é o melhor vendedor em valor de vendas de cada dategoria considerando os últimos 12 meses

SELECT v1.nomedacategoria, v1.nome, v1.total\_vendas FROM (SELECT ca.nomedacategoria, ve.nome, UM(it.quantidade \* it.precounitario \* (1-pe.desconto)) AS total FROM pedidos pe JOIN itens it ON pe.id = it.idpedido INNER JOIN produtos pr ON it.idproduto = pr.id INNER JOIN categorias ca ON pr.idcategoria = ca.id INNER JOIN vendedores ve ON pe.idvendedor = ve.id WHERE pe.datadopedido >= DATE\_SUB(CURRENT\_DATE(), INTERVAL 12 MONTH) GROUP BY ca.nomedacategoria, ve.nome) v1 INNER JOIN (SELECT nomedacategoria, MAX(total) AS maior FROM (SELECT ca.nomedacategoria, ve.nome, SUM(it.quantidade \* it.precounitario \* (1-pe.desconto)) AS total FROM pedidos pe INNER JOIN itens it ON pe.id = it.idpedido INNER JOIN produtos pr ON it.idproduto = pr.id INNER JOIN categorias ca ON pr.idcategoria = ca.id INNER JOIN vendedores ve ON pe.idvendedor = ve.id WHERE pe.datadopedido >= DATE\_SUB(CURRENT\_DATE(), INTERVAL 12 MONTH) GROUP BY ca.nomedacategoria, ve.nome) max GROUP BY nomedacategoria) v2 ON v1.nomedacategoria = v2.nomedacategoria AND v1.total = v2.maior ORDER BY 1, 2;