```
CREATE database ecommerce;

use ecommerce;
show databases;


-- Recuperações simples com SELECT
-- Lista todos os produtos cadastrados
SELECT idproduto, nomeProduto, precoProduto
FROM produto;


-- Filtros com WHERE
-- Seleciona clientes do tipo jurídico
SELECT nome, sobreNome, email
FROM cliente
WHERE tipoCliente = 'juridico';

-- Produtos com preço acima de R$ 500
SELECT nomeProduto, precoProduto
FROM produto
WHERE precoProduto > 500;


-- Expressões para gerar atributos derivados
-- Calcula o valor total por item no pedido (preço × quantidade)
SELECT
   pp.idpedido,
   p.nomeProduto,
   p.precoProduto,
   pp.quantidade,
   (p.precoProduto * pp.quantidade) AS valor_item
FROM
   pedido_produto pp
JOIN
   produto p ON pp.idproduto = p.idproduto;


-- Ordenações com ORDER BY
-- Lista os pedidos ordenados pelo valor total decrescente
SELECT idpedido, descrição, valorTotal
FROM pedido
ORDER BY valorTotal DESC;

-- Lista produtos ordenados por nome
SELECT nomeProduto, precoProduto
FROM produto
ORDER BY nomeProduto ASC;

-- Filtros por grupo com HAVING
-- Total de produtos por estoque, mostrando apenas estoques com mais de 100 unidades no total
SELECT
   pe.idestoque,
   SUM(pe.quantidade) AS total_estoque
FROM
   produto_estoque pe
GROUP BY
```

```
    pe.idestoque
HAVING
    SUM(pe.quantidade) > 100;

-- Junções entre tabelas para visão mais complexa
-- Consulta que mostra pedidos com nome do cliente e status da entrega
SELECT
    c.nome,
    c.sobreNome,
    p.idpedido,
    p.descrição,
    e.status_entrega,
    e.data_entrega
FROM
    pedido p
JOIN
    cliente c ON p.idcliente = c.idcliente
LEFT JOIN
    entrega e ON p.idpedido = e.idpedido;

-- Produtos vendidos por vendedores terceiros com preço e quantidade
SELECT
    vt.razao_social AS vendedor,
    pvt.nomeProduto,
    pvt.preco,
    pvt.quantidade
FROM
    produto_vendedor_terceiro pvt
JOIN
    vendedor_terceiro vt ON pvt.idvendedor_terceiro = vt.idvendedor_terceiro;


CREATE TABLE Cliente (
    idCliente INT AUTO_INCREMENT,
    nome VARCHAR(20) NOT NULL,
    sobreNome VARCHAR(40) NOT NULL,
    email VARCHAR(50) NOT NULL,
    tipoCliente ENUM('fisico', 'juridico') NOT NULL,
    telefone VARCHAR(20) NOT NULL,
    rua VARCHAR(40) NOT NULL,
    bairro VARCHAR(40) NOT NULL,
    estado VARCHAR(40) NOT NULL,
    pais VARCHAR(40) NOT NULL,
    cep INT NOT NULL,
            unique (email),
  PRIMARY KEY (idCliente)
);



CREATE TABLE pessoa_fisica (
    idpessoa_fisica INT PRIMARY KEY,
    cpf VARCHAR(14) UNIQUE NOT NULL,
    data_nascimento DATE,
    CONSTRAINT fk_cliente_pf FOREIGN KEY (idpessoa_fisica) REFERENCES cliente(idcliente)
);
```

```
CREATE TABLE pessoa_juridica(
	idpessoa_juridica int primary key,
	razao_social varchar (45),
	cnpj varchar (20)not null,
	unique (cnpj),
	constraint fk_cliente_pj foreign key (idpessoa_juridica) references cliente(idcliente)
);

CREATE TABLE pedido (
	idpedido int auto_increment primary key,
	idcliente int not null,
	descrição varchar (100) not null,
  valorTotal decimal (10,2) not null,
  data_pedido date not null,
	status_pedido enum('Confirmado','Em processamanto','cancelado') default ('Em
processamento'),
	constraint fk_pedido_cliente foreign key (idcliente) references cliente(idCliente)
	on delete cascade
  on update cascade
);

CREATE TABLE entrega(
	identrega int auto_increment primary key,
  idpedido int not null,
  tipo_entrega enum('Correios', 'transportadora') not null,
	data_envio date not null,
	data_prevista_entrega date not null,
	data_entrega date not null,
  codigo_rastreio varchar (45),
  status_entrega enum ('Enviado', 'Em transito', 'Saiu para entrega', 'Entregue') not null,
  unique (idpedido),
  constraint fk_entrega_pedido foreign key (idpedido) references pedido(idpedido)
);

CREATE TABLE pagamento(
	idpagamento int auto_increment primary key,
  data_pagamento date not null,
  status_pagamento ENUM('pago','aguardando') default ('aguardando')
  );


CREATE TABLE cartao_credito(
	idpagamento INT PRIMARY KEY,
  numero_cartao VARCHAR(20) NOT NULL,
  nome_titular VARCHAR(100) NOT NULL,
  validade DATE NOT NULL,
  constraint fk_pg_cartao foreign key (idpagamento) references pagamento(idpagamento)
);

CREATE TABLE pix(
	idpagamento INT PRIMARY KEY,
  chave_pix VARCHAR(100) NOT NULL,
  banco VARCHAR(50) NOT NULL,
  constraint fk_pg_pix foreign key (idpagamento) references pagamento(idpagamento)
);
```

```
CREATE TABLE boleto(
	idpagamento INT PRIMARY KEY,
  codigo_barras VARCHAR(100) NOT NULL,
  vencimento DATE NOT NULL,
  constraint fk_pg_bol foreign key (idpagamento) references pagamento(idpagamento)
);


CREATE TABLE produto(
	idproduto int auto_increment primary key,
  nomeProduto varchar (40) not null,
  descricaoProduto varchar (300) not null,
  precoProduto decimal (10,2) not null

);


CREATE TABLE pedido_produto(
	idproduto int not null,
	idpedido int not null,
	quantidade int,
  primary key (idproduto, idpedido),

  constraint fk_prd foreign key (idproduto) references produto(idproduto)
  on delete cascade
  on update cascade,

  constraint fk_pd foreign key (idpedido) references pedido(idpedido)
	on delete cascade
  on update cascade
);

CREATE TABLE produto_estoque(
	idproduto int not null,
  idestoque int not null,
  quantidade int default 0,
  primary key (idproduto, idestoque),
	constraint fk_prd_estq foreign key (idproduto) references produto(idproduto),
  constraint fk_estq foreign key (idestoque) references estoque(idestoque)
);

CREATE TABLE estoque(
	idestoque int auto_increment primary key,
  local_ varchar (100) not null
);

 insert into fornecedor (razao_social, cnpj, telefone, email)
 values
 ('GlobalTech Importações', '33.333.333/0001-33', '31-77777-4444', 'suporte@globaltech.com'),
  ('Distribuidora Aurora', '44.444.444/0001-44', '41-66666-5555', 'aurora@distrib.com');
  select * from fornecedor;
CREATE TABLE fornecedor(
	idfornecedor int auto_increment primary key,
  razao_social varchar (450) not null,
  cnpj varchar (14) not null,
  telefone varchar (20) not null,
  email varchar (50) not null,
```

```
   unique (cnpj, email)
);

CREATE TABLE distribui_produto(
		idproduto int not null,
   idfornecedor int not null,
   constraint fk_prd_fncdor foreign key (idproduto) references produto (idproduto),
   constraint fk_fncdor_prd foreign key (idfornecedor) references fornecedor (idfornecedor)
   on delete cascade
   on update cascade
);

CREATE TABLE vendedor_terceiro(
		idvendedor_terceiro int auto_increment primary key,
   razao_social varchar (45) not null,
   cnpj varchar (14) not null,
   email varchar (50) not null,
   telefone varchar (20) not null,
   local_ varchar (100) not null,
   unique (cnpj, email)
);


CREATE TABLE produto_vendedor_terceiro(
		idproduto int not null,
   idvendedor_terceiro int not null,
   nomeProduto varchar (100) not null,
   quantidade int default 0,
   descricao_produto_terceiro varchar (300) not null,
   preco decimal (10,2) not null,
   primary key (idproduto, idvendedor_terceiro),
   constraint fk_prod foreign key (idproduto) references produto (idproduto),
   constraint fk_prd_vndor_terc foreign key (idvendedor_terceiro) references
vendedor_terceiro(Idvendedor_terceiro)
   on delete cascade
   on update cascade
);
```