

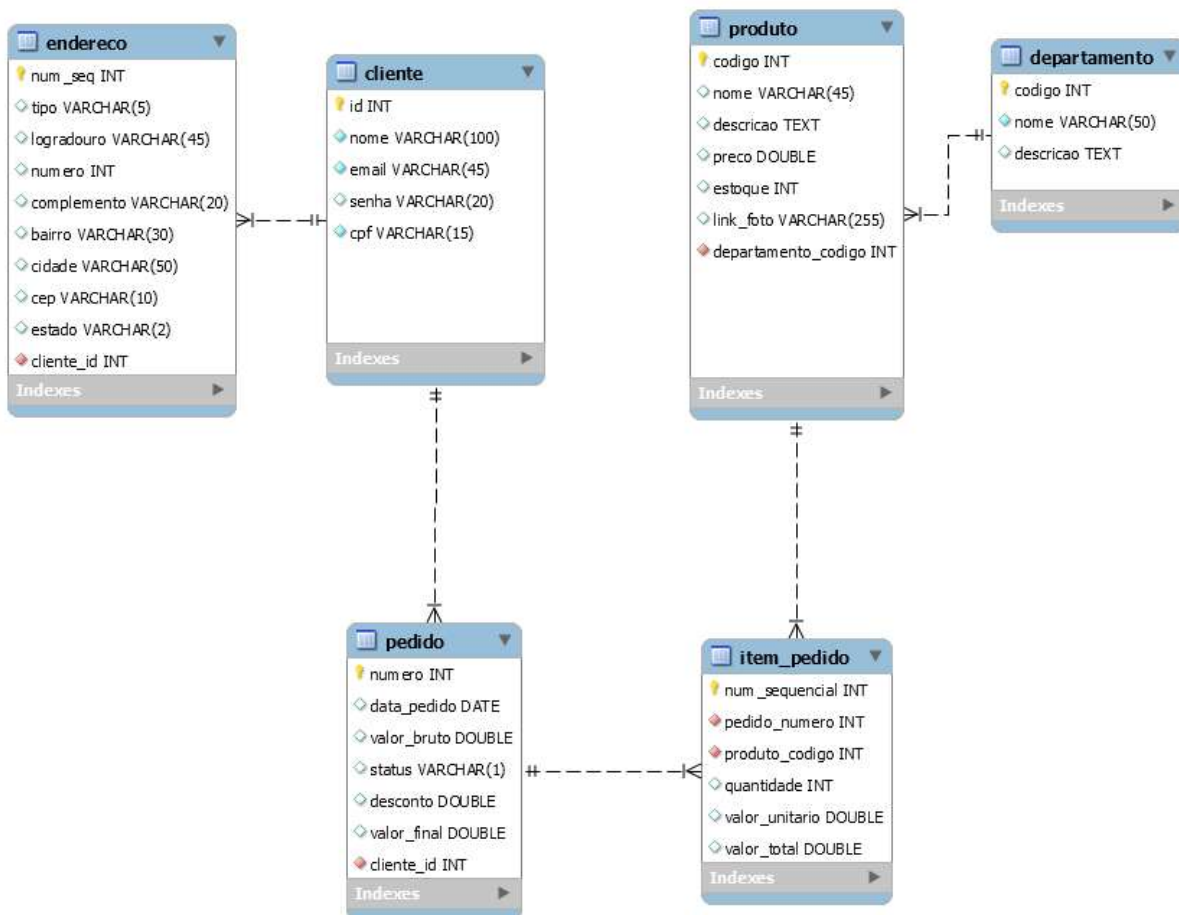
Hacker - Desafio Hands On - em grupo

Olá Hackers!

Muito bem! Agora é a hora mergulhar no Back End!

Como vocês já são quase experts em lógica de programação, vamos começar a trabalhar em trios para fazer a nossa primeira estrutura de banco de dados em SQL! Agora em grupos, teremos ainda mais energia para poder concluir esse Hands On Week.

Temos um cenário de Comércio Eletrônico conforme a figura abaixo:



O conjunto de instruções SQL que implementam este modelo está a seguir:

```

CREATE TABLE cliente(
  id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(70) NOT NULL UNIQUE,

```

```
senha VARCHAR(20) NOT NULL,  
cpf VARCHAR(15) NOT NULL UNIQUE  
);
```

```
CREATE TABLE departamento(  
codigo INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(50) NOT NULL,  
descricao TEXT  
);
```

```
CREATE TABLE endereco(  
num_seq INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
tipo VARCHAR(5) NOT NULL,  
logradouro VARCHAR(45) NOT NULL,  
numero INTEGER,  
complemento VARCHAR(20),  
bairro VARCHAR(30),  
cidade VARCHAR(50),  
estado VARCHAR(2),  
cep VARCHAR(10),  
cliente_id INTEGER NOT NULL,  
CONSTRAINT endereco_cliente FOREIGN KEY  
(cliente_id) REFERENCES cliente(id)  
);
```

```
CREATE TABLE pedido(  
numero INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
status VARCHAR(1) NOT NULL,  
data_pedido DATE,  
valor_bruto DOUBLE,  
desconto DOUBLE,  
valor_final DOUBLE,
```

```
cliente_id INTEGER NOT NULL,  
  
CONSTRAINT cliente_pedido FOREIGN KEY  
  
    (cliente_id) REFERENCES cliente(id)  
  
);
```

```
CREATE TABLE produto(  
  
codigo INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  
nome VARCHAR(45) NOT NULL,  
  
descricao TEXT,  
  
preco DOUBLE,  
  
estoque INTEGER,  
  
link_foto VARCHAR(255),  
  
    departamento_codigo INTEGER NOT NULL,  
  
CONSTRAINT produto_depto FOREIGN KEY  
  
(departamento_codigo) REFERENCES departamento(codigo)  
  
);
```

```
CREATE TABLE item_pedido(  
  
num_sequencial INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  
quantidade INTEGER,  
  
valor_unitario DOUBLE,  
  
valor_total DOUBLE,  
  
produto_codigo INTEGER NOT NULL,  
  
pedido_numero INTEGER NOT NULL,  
  
CONSTRAINT item_produto FOREIGN KEY  
  
(produto_codigo) REFERENCES produto(codigo),  
  
CONSTRAINT item_pedido FOREIGN KEY  
  
    (pedido_numero) REFERENCES pedido(numero)  
  
);
```

Com base nestas informações, realize os seguintes exercícios, gerando instruções SQL para responder às questões.

1. Este exercício é livre para você inserir dados nas tabelas. Adicione vários dados em todas as tabelas. Crie vários clientes, com vários endereços. Insira muitos produtos em vários departamentos. Crie pedidos em várias datas com meses diferentes (serão necessários para os próximos exercícios).
2. Quantos clientes estão cadastrados na sua base?
3. Qual o produto mais caro?
4. Qual o produto mais barato?
5. Mostre todos os produtos com seus respectivos departamentos.
6. Quantos produtos há em cada departamento? Exiba o nome do departamento e a quantidade de produtos que há em cada um. (pense em SUM e GROUP BY)
7. Mostre os dados dos pedidos, incluindo nomes dos clientes e nomes dos produtos que foram vendidos.
8. Mostre quantos pedidos foram feitos por mês no ano de 2022 (caso você tenha registros neste ano, senão escolha um ano que você tenha cadastrado - Novamente pense em COUNT e GROUP BY).
9. Mostre quanto foi faturado por mês (leve em conta o valor total de cada pedido - novamente pense em GROUP BY e SUM).
10. Mostre o valor total do estoque por departamento.

Entrega: 11 de fevereiro de 2022

Até às 11h59 [antes do meio dia]

*** (campo obrigatório) Link da entrega**

Grupo 49 - Grupo 4 - Hacker e Hipster

ENVIAR