



Atividade de banco de dados

Uma loja de sapatos precisa de um sistema para controlar todas as suas ações. O sistema será desenvolvido de forma modular, e, inicialmente, o gerente deseja cadastrar os vendedores e clientes, além de registrar cada venda em que ambos estão envolvidos. O objetivo é permitir que o gerente obtenha informações estratégicas sobre o faturamento da loja.

Crie o banco de dados:

```
CREATE DATABASE SAPATOSENAC;
```

Crie as tabelas:

```
CREATE TABLE vendedor (  
    id_vendedor INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50),  
    salario DECIMAL(10, 2)  
);  
  
CREATE TABLE cliente (  
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50),  
    endereco VARCHAR(100)  
);  
  
CREATE TABLE venda (  
    id_venda INT AUTO_INCREMENT PRIMARY KEY,  
    id_vendedor INT,  
    id_cliente INT,  
    valor DECIMAL(10, 2),  
    data DATE,  
    FOREIGN KEY (id_vendedor) REFERENCES vendedor(id_vendedor),  
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente)  
);
```

Crie um script para preencher as tabelas automaticamente com dados fictícios (**CONFIGURE ACESSO AO BANCO**):

```
import random  
import mysql.connector  
from mysql.connector import errorcode  
  
# Configurações do banco de dados  
db_config = {  
    "host": "localhost",  
    "user": "root",  
    "password": "admin",  
    "database": "senac"  
}  
  
# Função para gerar nomes aleatórios  
def generate_random_name():  
    first_names = ["John", "Jane", "Michael", "Emily", "William", "Olivia", "Sophia", "James", "Emma", "Daniel"]  
    last_names = ["Smith", "Johnson", "Williams", "Jones", "Brown", "Davis", "Miller", "Wilson", "Moore", "Taylor"]  
    return f"{random.choice(first_names)} {random.choice(last_names)}"  
  
# Função para gerar endereços aleatórios  
def generate_random_address():  
    streets = ["Main St", "Oak St", "Elm St", "Maple Ave", "Cedar Rd", "Pine Ln", "Birch Dr", "Willow Way", "Ash St"]  
    cities = ["New York", "Los Angeles", "Chicago", "Houston", "Phoenix", "Philadelphia", "San Antonio", "San Diego", "Dallas"]  
    return f"{random.randint(100, 999)} {random.choice(streets)}, {random.choice(cities)}"  
  
# Função para inserir dados de vendedor, cliente e venda  
def insert_data(cursor, num_rows=50000):  
    for i in range(1, num_rows + 1):  
        # Gerar dados aleatórios  
        vendedor_nome = generate_random_name()  
        vendedor_salario = round(random.uniform(2000, 8000), 2)  
  
        cliente_nome = generate_random_name()  
        cliente_endereco = generate_random_address()
```

```

valor_venda = round(random.uniform(100, 1000), 2)
data_venda = f"2023-{random.randint(1, 12):02d}-{random.randint(1, 28):02d}"

try:
    # Inserir dados nas tabelas
    cursor.execute("INSERT INTO vendedor (nome, salario) VALUES (%s, %s)", (vendedor_nome, vendedor_salario))
    cursor.execute("INSERT INTO cliente (nome, endereco) VALUES (%s, %s)", (cliente_nome, cliente_endereco))
    cursor.execute("INSERT INTO venda (id_vendedor, id_cliente, valor, data) VALUES (LAST_INSERT_ID(), LAST_INSERT_ID(), %s, %s)", (valor_venda, data_venda))

except mysql.connector.Error as err:
    print(f"Erro ao inserir dados: {err}")
    exit(1)

# Função principal
def main():
    try:
        # Conectar ao banco de dados
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()

        # Inserir dados nas tabelas
        insert_data(cursor)

        # Efetivar as alterações no banco de dados
        conn.commit()

    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Acesso negado ao banco de dados.")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("Banco de dados não existe.")
        else:
            print(err)

    finally:
        # Fechar conexão e cursor
        cursor.close()
        conn.close()

if __name__ == "__main__":
    main()

```

▼ Exercícios Práticos

Exercício 1:

Retorne todos os dados da tabela "vendedor".

Exercício 2:

Insira um novo cliente na tabela "cliente".

Exercício 3:

Atualize o salário do vendedor com o ID 2 para 3500.00.

Exercício 4:

Liste o nome e endereço de todos os clientes.

Exercício 5:

Atualize o nome do cliente com o ID 3 para "Lucas Fernandes".

Exercício 6:

Calcule o total de vendas de cada vendedor por mês.

Exercício 7:

Encontre o nome do vendedor que realizou a maior venda.

Exercício 8:

Liste o nome dos vendedores e a média de seus salários.

Exercício 9:

Calcule o total de vendas realizadas por mês para todos os vendedores.

Exercício 10:

Retorne todas as vendas realizadas pelo cliente com o ID 2.