**Lista 4º bimestre**

**Enzo Dante Mícoli – 2º H**

**50210203**

**38.6 – Desafios arquivos texto**

**a)**

while True:

    try:

        nomearquivo = input("Insira o nome do arquivo a ser criado:  ")

        arquivo = open(f'{nomearquivo}.html', 'x')

        titulo = input('Digite um título para sua página:  ')

        texto = input("Digite seu texto:\n")

        if not texto.startswith('<p>') and not texto.endswith('</p>'):

            texto = f'<p>{texto}</p>'

        formatacao\_html = f'''

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>{titulo}</title>

</head>

<body>

    <header><h1>{titulo}</h1></header>

    <main>

        {texto}

    </main>

</body>

</html>

        '''

        arquivo.write(formatacao\_html)

        arquivo.close()

        print("Arquivo gravado com sucesso!")

    except:

        print("Ocorreu um erro ao criar o arquivo! verifique se o arquivo ja existe!\n")

    finally:

        if input('Deseja continuar? 1-sim qualquer tecla-não') != '1':

            break

**b)**

import mysql.connector

lista = list()

dicio = dict()

disciplinas\_por\_curso = list()

def banco():

    try:

        global conn

        conn = mysql.connector.Connect(

            host='localhost',

            database='univap',

            user='root',

            password=''

        )

        if conn.is\_connected():

            global sql

            sql = conn.cursor()

            return 1

        else:

            print('não foi possível conectar ao banco')

            return 0

    except Exception as erro:

        print(f'erro: {erro}')

#=====buscar por disciplinas de determinada relação==========

def disciplinas(c=0, curso=0):

    try:

        sql = conn.cursor()

        sql.execute(f'select \* from disciplinas where codigodisc={c}')

        discs = sql.fetchall()

        if sql.rowcount > 0:

            for d in discs:

                cod\_disc = d[0]

                nome\_disc = d[1]

                disciplinas\_por\_curso.append(f'{curso} {cod\_disc} | {nome\_disc}')

    except:

        print("Ocorreu um erro ao buscar dados na tabela disciplinas!\n")

#==========buscar por dados de determinado professor======

def disciplinasxprofs(prof=0):

    try:

        sql = conn.cursor()

        sql.execute(f'select \* from disciplinasxprofessores where codprofessor={prof} order by curso')

        discprofs = sql.fetchall()

        if sql.rowcount > 0:

            for dados in discprofs:

                disciplinas(dados[1], dados[3])

                dicio['curso'] = disciplinas\_por\_curso.copy()

            disciplinas\_por\_curso.clear()

            lista.append(dicio.copy())

            dicio.clear()

    except:

        print("Ocorreu um erro ao buscar pela relação do professor com disciplinas!\n")

#=========buscar por professor=======

def professores():

    try:

        sql = conn.cursor()

        sql.execute('select \* from professores;')

        tabela = sql.fetchall()

        if sql.rowcount > 0:

            for i in tabela:

                professor = i[0]

                global arquivo

                arquivo = open(f'{professor}.html', 'w')

                nome\_prof = i[1]

                dicio['id\_prof'] = professor

                dicio['nome\_prof'] = nome\_prof

                #print(dicio)

                disciplinasxprofs(professor)

                format\_html = f'''

<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>{professor}</title>

</head>

<body>

    <header><h1>Disciplinas do professor: {nome\_prof}</h1></header>

    <main>

        <h2>CÓDIGO DA DISCIPLINA | NOME DISCIPLINA</h2>

        '''

                '''print('=========')

                for i in lista:

                    print(i)'''

                for i in lista:

                    for indice, valores in i.items():

                        v = '0'

                        if indice == 'curso':

                            for lista\_cursos in valores:

                                curso = f'{lista\_cursos[0]}{lista\_cursos[1]}'

                                if curso != v:

                                    format\_html += f'''<h3>Curso: {lista\_cursos[0]}{lista\_cursos[1]}</h3>'''

                                    #print(f'Curso: {lista\_cursos[0]}{lista\_cursos[1]}')

                                    v = curso

                                format\_html += f'''<p>{lista\_cursos[2:]}</p>'''

                format\_html +='''</main></body></html>'''

                lista.clear()

                arquivo.write(format\_html)

                arquivo.close()

        else:

            print('Não existe professor cadastrado!\n')

    except:

        print('Ocorreu um erro ao procurar por determinado professor ou gerar o arquivo desse professor!')

#=========main ===========

if banco() == 1:

    if input('Deseja criar os arquivos? 1-sim qualquer tecla-não\n') == '1':

        print('Verificando dados do banco...\n')

        professores()

        print("criando arquivos...\n")

        print("arquivos criados com sucesso\n")

**39.2 DESAFIOS**

import mysql.connector

from reportlab.pdfgen import canvas

def banco():

    try:

        global conn

        conn = mysql.connector.Connect(

            host='localhost',

            database='univap',

            user='root',

            password=''

        )

        if conn.is\_connected():

            global sql

            sql = conn.cursor()

            return 1

        else:

            print('não foi possível conectar ao banco')

            return 0

    except Exception as erro:

        print(f'erro: {erro}')

#========opcao 1============

def registroprof():

    try:

        registro = int(input("Digite um registro: "))

        while registro < 0:

            registro = int(input("Digite um registro: "))

        sql = conn.cursor()

        sql.execute(f"select \* from professores where registro={registro}")

        pdf.setFillColor('black')

        pdf.setTitle('Dados de Professor')

        pdf.setFont("Helvetica-Oblique", 20)

        pdf.drawString(250, 800, 'Professor')

        pdf.setFont("Helvetica-Oblique", 15)

        profs = sql.fetchall()

        x = 700 #mm

        if sql.rowcount > 0:

            for i in profs:

                pdf.drawString(35, x, f"Nome: {i[1]} telefone: {i[2]} idade: {i[3]} Salário: {i[4]}")

                x -= 15

        else:

            print("Não existe esse registro")

    except:

        print("O registro informado não é um número ou ocorre um erro ao procurar pelo professor")

#===========opcao 2============

def caractereprof():

    try:

        #WHERE nome\_p LIKE '%$valor%'

        caractere = input("Digite um caractere: ")

        while caractere == '':

            caractere = input("Digite um caractere corretamente: ")

        sql = conn.cursor()

        sql.execute(f"select \* from professores where nomeprof like '%{caractere}%'")

        pdf.setFillColor('black')

        pdf.setTitle('Dados de Professor')

        pdf.setFont("Helvetica-Oblique", 20)

        pdf.drawString(250, 800, 'Professor')

        pdf.setFont("Helvetica-Oblique", 15)

        dados = sql.fetchall()

        x = 700

        if sql.rowcount > 0:

            for i in dados:

                pdf.drawString(25, x, f"Nome: {i[1]} telefone: {i[2]} idade: {i[3]} Salário: {i[4]}")

                x -= 30

    except:

        print("Erro, não foi possível achar esse professor")

#============= opcao 3====================

def cursodisc():

    try:

        c = int(input("Informe o número de um curso: "))

        sql = conn.cursor()

        sql.execute(f"select coddisciplina, nomedisc from disciplinas, disciplinasxprofessores where codigodisc=coddisciplina and curso={c}")

        pdf.setFillColor('black')

        pdf.setTitle('Disciplinas de curso')

        pdf.setFont("Helvetica-Oblique", 20)

        pdf.drawString(210, 800, 'Disciplinas de curso')

        pdf.setFont("Helvetica-Oblique", 15)

        dis = sql.fetchall()

        x = 700

        if sql.rowcount > 0:

            for i in dis:

                pdf.drawString(20, x, f"Disciplina: {i[1]}")

                x -= 30

        else:

           pdf.drawString(20, 700, f"Disciplina inexistente")

    except:

        print("Ocorreu um erro ao procurar por disciplinas")

#========opcao 4===============

def cursoprofs():

    try:

        c = int(input("Informe o numero de um curso: "))

        sql = conn.cursor()

        sql.execute(f"select nomeprof, registro, codprofessor from professores, disciplinasxprofessores where curso={c} and codprofessor=registro")

        nomes = sql.fetchall()

        pdf.setFillColor('black')

        pdf.setTitle('professores de curso')

        pdf.setFont("Helvetica-Oblique", 20)

        pdf.drawString(210, 800, 'professores de curso')

        pdf.setFont("Helvetica-Oblique", 15)

        x = 700

        if sql.rowcount > 0:

            for i in nomes:

                pdf.drawString(20, x, f"Nome: {i[0]} ")

                x -= 30

    except:

        print("Ocorreu um erro ao procurar por professor")

#=====opcao 5==========

def cargaletiva():

    try:

        c = int(input("Informe o numero de um curso: "))

        sql = conn.cursor()

        sql.execute(f"select sum(cargahoraria),anoletivo from disciplinasxprofessores where curso={c} group by anoletivo")

        x = 700

        dados = sql.fetchall()

        pdf.setFillColor('black')

        pdf.setTitle('Carga total por ano letivo')

        pdf.setFont("Helvetica-Oblique", 20)

        pdf.drawString(150, 800, 'Carga total por ano letivo')

        pdf.setFont("Helvetica-Oblique", 15)

        if sql.rowcount > 0:

            for i in dados:

                pdf.drawString(20, x, f"Ano letivo: {i[1]}-- Carga horária total: {i[0]} ")

                x -= 30

    except:

        print("Ocorreu um erro ao procurar por carga horária")

#-====main=====-

if banco() == 1:

    while True:

        try:

            print("="\*30)

            print("[1] - Dados de professor por meio de seu registro\n")

            print("[2] - Dados de professor por meio de uma inicial de nome(somente uma letra)\n")

            print("[3] - Nomes de disciplinas por meio de um curso\n")

            print("[4] - Nomes de professores por meio de um curso\n")

            print("[5] - Carga horária total de um curso meio de um ano letivo\n")

            escolha = int(input('Escolha uma opção:  '))

            while escolha < 1 or escolha > 5:

                escolha = int(input('Escolha uma opção corretamente:  '))

            nomearquivo = input("Digite um nome para o pdf: ")

            pdf = canvas.Canvas(f'{nomearquivo}.pdf')

            if escolha == 1:

                registroprof()

            elif escolha == 2:

                caractereprof()

            elif escolha == 3:

                cursodisc()

            elif escolha == 4:

                cursoprofs()

            else:

                cargaletiva()

            pdf.save()

            print(f'{nomearquivo} foi criado ')

        except:

            print("Digite um número!!!\n")

            continue