

IES GONZALO NAZARENO

PROFESOR: CONCEPCION GUIADO JURADO

ASIGNATURA: GESTIÓN DE BASES DE DATOS

CURSO: 1º ASIR

PRACTICA: PROYECTO PYTHON-BBDD ORACLE

INDICE

- 1- Muestra el nombre de cada empresa junto con su respectiva localidad.**
- 2- Muestra cargos que empiecen por una subcadena especificada.**
- 3- Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado**
- 4- Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.**
- 5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.**
- 6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.**

Ejercicio1: Muestra el nombre de cada empresa junto con su respectiva localidad.

```
def mostrar_empresas_por_localidad(db):
    cursor = db.cursor()
    try:
        cursor.execute("SELECT Localidad, COUNT(*) FROM EMPRESA GROUP BY Localidad")
        resultados = cursor.fetchall()
        localidad_actual = None
        total_empresas = 0
        for localidad, nombre_empresa in resultados:
            if localidad != localidad_actual:
                print(f"Localidad: {localidad}")
                localidad_actual = localidad
                print(f"Nº de empresa: {nombre_empresa}")
                total_empresas += 1
        cursor.execute("SELECT COUNT(*) FROM EMPRESA")
        total_empresas_en_db = cursor.fetchone()[0]
        print(f"Total de empresas: {total_empresas_en_db}")
    except Exception as e:
        print("Se ha producido un error al ejecutar la consulta:", e)
    finally:
        cursor.close()
```

```
C:\Users\aletr\OneDrive\Escritorio\BBDD\ProyectoBBDD - Oracle>python3 programa.py

1- Muestra el nombre de cada empresa junto con su respectiva localidad.
2- Muestra cargos que empiecen por una subcadena especificada.
3- Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado
4- Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.
5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.
6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.
7- Salir

Selecciona una opción: 1

Localidad: Utrera
Nº de empresa: 1
Localidad: Sevilla
Nº de empresa: 2
Localidad: Cadiz
Nº de empresa: 1
Localidad: Cordoba
Nº de empresa: 1
Localidad: Almeria
Nº de empresa: 1
Localidad: Jaen
Nº de empresa: 1
Localidad: Huelva
Nº de empresa: 1
Localidad: Malaga
Nº de empresa: 1
Localidad: Granada
Nº de empresa: 1
Total de empresas: 10
```

Ejercicio2: Muestra cargos que empiecen por una subcadena especificada.

```
def buscar_cargos(conexion, subcadena):
    cursor = conexion.cursor()
    consulta = ("SELECT Cargo FROM PERSONA_DE_CONTACTO WHERE Cargo LIKE :subcadena")
    cursor.execute(consulta, {'subcadena': subcadena + '%'})

    cargos_encontrados = cursor.fetchall()

    if cargos_encontrados:
        print("Cargos encontrados que comienzan con '{}':".format(subcadena))
        for cargo in cargos_encontrados:
            print(cargo[0])
```

else:

```
print("No se encontraron cargos que comiencen con '{}".format(subcadena))
```

```
1- Muestra el nombre de cada empresa junto con su respectiva localidad.
2- Muestra cargos que empiecen por una subcadena especificada.
3- Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado
4- Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.
5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.
6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.
7- Salir

Selecciona una opción: 2

Introduce la subcadena para buscar los cargos: In
Cargos encontrados que comienzan con 'In':
Informatico
Ingeniero
```

Ejercicio3: Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado

```
def mostrar_area_trabajo_empleado(db, nombre_empleado):
```

```
    try:
```

```
        cursor = db.cursor()
```

```
        sql = """SELECT Nombre, Descripcion
```

```
            FROM AreaTrabajo
```

```
            WHERE ID_AreaTrabajo = (
```

```
                SELECT ID_AreaTrabajo
```

```
                FROM AreaTrabajo
```

```
                WHERE Nombre = :nombre_empleado
```

```
            )"""
```

```
        cursor.execute(sql, {'nombre_empleado': nombre_empleado})
```

```
        resultado = cursor.fetchone()
```

```
        if resultado:
```

```
            print(f"Área de trabajo de {nombre_empleado}:")
```

```
            print(f"Nombre: {resultado[0]}\nDescripción: {resultado[1]}")
```

```
        else:
```

```
            print(f"No se encontró información del área de trabajo para {nombre_empleado}")
```

```
    except Exception as e:
```

```
        print(f"Ocurrió un error: {e}")
```

```
1- Muestra el nombre de cada empresa junto con su respectiva localidad.
2- Muestra cargos que empiecen por una subcadena especificada.
3- Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado
4- Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.
5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.
6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.
7- Salir

Selecciona una opción: 3

Introduce el nombre del empleado: Alejandro
Área de trabajo de Alejandro:
Nombre: Alejandro
Descripción: Área de desarrollo de software
```

Ejercicio4: Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.

```
def agregar_empresa(conexion, cif, nombre, direccion, localidad):
```

```
    try:
```

```
        cursor = conexion.cursor()
```

```
        sql = "INSERT INTO Empresa (CIF, Nombre, Direccion, Localidad) VALUES
```

```
(:cif, :nombre, :direccion, :localidad)"
```

```
        val = {'cif': cif, 'nombre': nombre, 'direccion': direccion, 'localidad': localidad}
```

```

        cursor.execute(sql, val)
        conexion.commit()
        print("Empresa agregada correctamente.")
    except Exception as e:
        print(f"Error al agregar la empresa: {e}")

```

```

def actualizar_tabla_empresa(db):
    cursor = db.cursor()
    sql = "SELECT * FROM EMPRESA"
    try:
        cursor.execute(sql)
        registros = cursor.fetchall()
        print("TABLA ACTUALIZADA EMPRESA:")
        for result in registros:
            print("CIF:", result[0])
            print("Nombre:", result[1])
            print("Direccion:", result[2])
            print("Localidad:", result[3])
            print("-----")
    except:
        print("Se ha producido un error al mostrar la tabla.")

```

```

1- Muestra el nombre de cada empresa junto con su respectiva localidad.
2- Muestra cargos que empujen por una subcadena especificada.
3- Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado
4- Pide por teclado los datos de una nueva empresa, luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.
5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.
6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.
7- Salir

Selecciona una opción: 4
Introduce el cif de la empresa: K55443322
Introduce el nombre de la empresa: ALDI
Introduce la dirección de la empresa: Calle Real
Introduce la localidad de la empresa: Huelva
Empresa agregada correctamente.
TABLA ACTUALIZADA EMPRESA:
CIF: K55443322
Nombre: ALDI
Direccion: Calle Real
Localidad: Huelva
-----
CIF: A12345678
Nombre: Mercadona
Direccion: Calle Sevilla
Localidad: Utrera
-----
CIF: B46066361
Nombre: Ford
Direccion: Calle Arenal
Localidad: Sevilla
-----
CIF: C28328508
Nombre: UPS
Direccion: Calle Pureza
Localidad: Cadiz

```

Ejercicio5: Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.

```

def eliminar_personas_por_prefijo(db, prefijo):
    try:
        cursor = db.cursor()
        sql = "DELETE FROM PERSONA_DE_CONTACTO WHERE Cargo LIKE :prefijo"
        cursor.execute(sql, {'prefijo': prefijo + '%'})
        filas_afectadas = cursor.rowcount
        print(f"Se han eliminado {filas_afectadas} registros donde el nombre comienza con '{prefijo}'.")
        db.commit()
    except Exception as e:

```

```
print(f"Se ha producido un error al eliminar los clientes: {e}")
db.rollback()
```

```
def actualizar_tabla_persona_contacto(db):
    cursor = db.cursor()
    sql = "SELECT * FROM PERSONA_DE_CONTACTO"
    try:
        cursor.execute(sql)
        registros = cursor.fetchall()
        print("TABLA ACTUALIZADA PERSONA DE CONTACTO:")
        for result in registros:
            print("Codigo_ContactoEmpresa:", result[0])
            print("Cargo:", result[1])
            print("Nombre:", result[2])
            print("Correo_electronico:", result[3])
            print("CIF:", result[4])
            print("-----")
    except:
        print("Se ha producido un error al mostrar la tabla.")
```

```
1- Muestra el nombre de cada empresa junto con su respectiva localidad.
2- Muestra cargos que empiecen por una subcadena especificada.
3- Pide por teclado un nombre de empleado y muestra la descripción del área de trabajo asociada a ese empleado
4- Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.
5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.
6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.
7- Salir

Selecciona una opción: 5

Introduce el prefijo que quieras eliminar: Pro
Se han eliminado 2 registros donde el nombre comienza con 'Pro'.
TABLA ACTUALIZADA PERSONA DE CONTACTO:
Codigo_ContactoEmpresa: 111
Cargo: Gerente
Nombre: Juan
Correo_electronico: juan.ruiz@gmail.com
CIF: A12345678
-----
Codigo_ContactoEmpresa: 222
Cargo: Desarrollador
Nombre: Daniel
Correo_electronico: daniel.diane@gmail.com
CIF: B46066361
-----
Codigo_ContactoEmpresa: 333
Cargo: Coordinador
Nombre: Mario
Correo_electronico: mario.ruiz@gmail.com
CIF: C28328508
-----
Codigo_ContactoEmpresa: 444
Cargo: Informático
Nombre: Cristian
Correo_electronico: cristian.pinillos@gmail.com
CIF: D65005357
-----
Codigo_ContactoEmpresa: 666
Cargo: Ingeniero
Nombre: Pablo
Correo_electronico: pablo.rodri@gmail.com
CIF: G09266298
-----
Codigo_ContactoEmpresa: 777
Cargo: Analista
Nombre: Angela
Correo_electronico: angela.gonz@gmail.com
CIF: W0049001A
-----
Codigo_ContactoEmpresa: 999
Cargo: Arquitecta
Nombre: Paula
```

Ejercicio6: Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.

```
def actualizar_area_trabajo(db, ID_AreaTrabajo, campo, nuevo_valor):
    sql = f"UPDATE AreaTrabajo SET {campo} = :nuevo_valor WHERE ID_AreaTrabajo"
    = :ID_AreaTrabajo"
    try:
        cursor = db.cursor()
        cursor.execute(sql, {'nuevo_valor': nuevo_valor, 'ID_AreaTrabajo': ID_AreaTrabajo})
        db.commit()
```

```
print("La información ha sido actualizada correctamente.")
except Exception as e:
    db.rollback()
    print("No se ha podido actualizar la información:", e)
```

```
def actualizar_tabla_area_trabajo(db):
    cursor = db.cursor()
    sql = "SELECT * FROM AreaTrabajo"
    try:
        cursor.execute(sql)
        registros = cursor.fetchall()
        print("TABLA ACTUALIZADA PERSONA DE CONTACTO:")
        for result in registros:
            print("ID_AreaTrabajo:", result[0])
            print("Nombre:", result[1])
            print("Descripcion:", result[2])
            print("-----")
    except:
        print("Se ha producido un error al mostrar la tabla.")
```

```
1- Muestra el nombre de cada empresa junto con su respectiva localidad.
2- Muestra cargos que empiecen por una subcadena especificada.
3- Pide por teclado un nombre de empleado y muestre la descripción del área de trabajo asociada a ese empleado
4- Pide por teclado los datos de una nueva empresa. Luego, inserta los datos en la tabla empresa y muestra una tabla actualizada con toda la información de la tabla empresa.
5- Elimina todos los cargos cuyos nombres comienzan con un prefijo específico.
6- Actualiza la información de la descripción de la tabla área de trabajo solicitando al usuario el código de área de trabajo y el nombre que desea actualizar.
7- Salir

Selecciona una opción: 6
Introduce la ID del Área de Trabajo que desea actualizar: 111
Introduce el campo que desea actualizar (Nombre, Descripción): Nombre
Introduce el nuevo valor para el campo seleccionado: Conchí
La información ha sido actualizada correctamente.
TABLA ACTUALIZADA PERSONA DE CONTACTO:
ID_AreaTrabajo: 111
Nombre: Conchí
Descripcion: Area de desarrollo de software
-----
ID_AreaTrabajo: 222
Nombre: Raul
Descripcion: Recursos Humanos
-----
ID_AreaTrabajo: 333
Nombre: Daniel
Descripcion: Ventas
-----
ID_AreaTrabajo: 444
Nombre: Pablo
Descripcion: Soporte Tecnico
-----
ID_AreaTrabajo: 555
Nombre: Fran
Descripcion: Marketing
-----
ID_AreaTrabajo: 666
Nombre: Manuel
Descripcion: Investigacion y Desarrollo
-----
```