Acceso a datos desde Python

Índice

1. Introducción

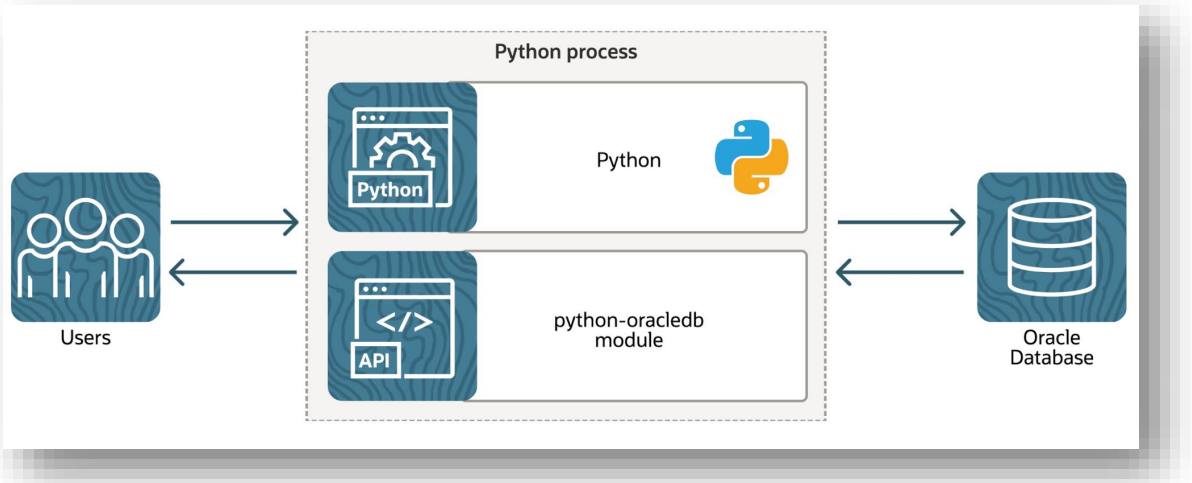
En esta sesión práctica se ilustrará el procedimiento seguido para realizar el acceso a datos desde Python, en concreto sobre **Oracle Database Express Edition**, ya instalado en sesiones anteriores. Además, se entiende que se encuentra también instalado **Oracle SQL Developer**, también en sesiones prácticas anteriores.

Sobre el sistema creado en las sesiones de prácticas anteriores, es preciso disponer del siguiente software:

- **Oracledb – *(Instalado en esta sesión práctica)***
- **Psycopg2 – *(Instalado en esta sesión práctica)***
- Anaconda, que integra Jupyter Notebook – *(Ya instalado en prácticas anteriores)*
- Python – *(Ya instalado en prácticas anteriores)*
- Oracle Database Express Edition – *(Ya instalado en prácticas anteriores)*
- Oracle SQL Developer – *(Ya instalado prácticas anteriores)*
- PostgreSQL – *(Ya instalado prácticas anteriores)*

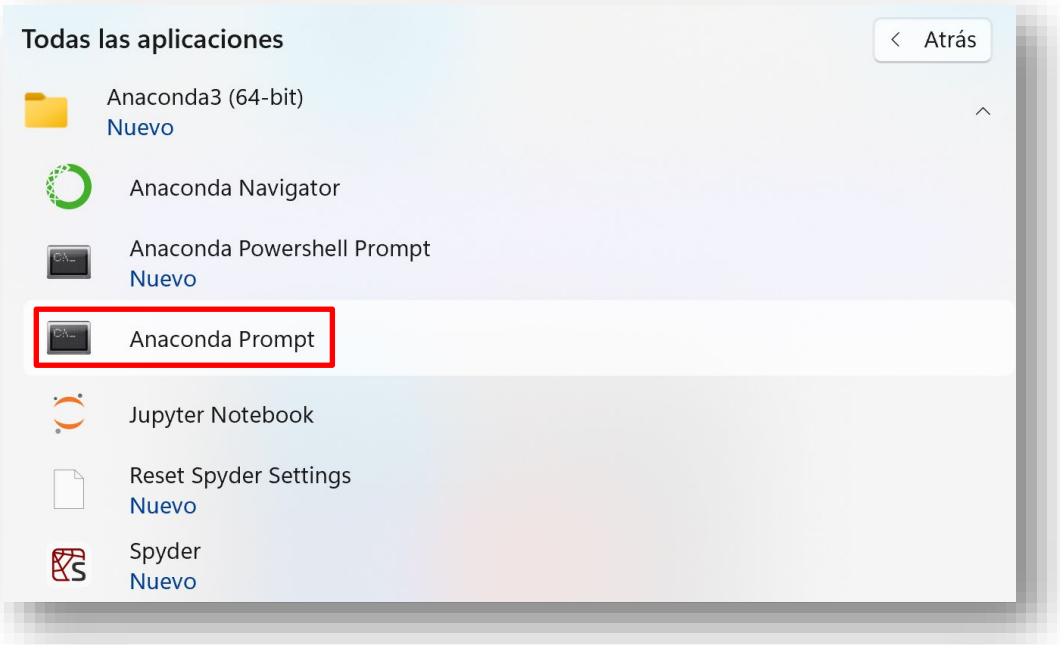
2. Preparación del entorno para Oracle

En esta ocasión, será preciso instalar también el módulo `oracledb` de Python, para poder gestionar adecuadamente el acceso a datos sobre Oracle desde Python. Tal y como se describe en la web de Oracle el acceso se resume en la siguiente figura:



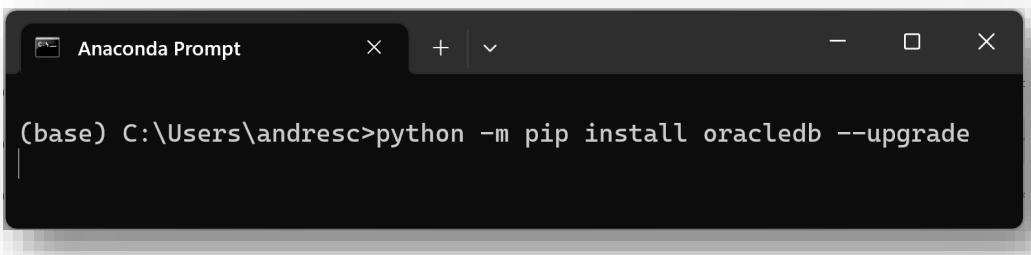
(Fuente: https://python-oracledb.readthedocs.io/en/latest/user_guide/installation.html)

Para proceder a la instalación del módulo Oracledb, es preciso abrir la consola de Anaconda:



Y teclear el siguiente comando, según se explica en:
https://python-oracledb.readthedocs.io/en/latest/user_guide/installation.html

```
python -m pip install oracledb --upgrade
```



Desde este momento, ya será posible importar la librería de acceso a Oracle desde Python en el código fuente que se desarrolle. Se usará el alias PBD para hacer compatible el código entre Oracle y PostgreSQL:

```
import oracledb as PBD
```

3. Preparación del entorno para Postgresql

Para realizar el acceso a datos a Postgresql desde Python es necesario instalar el módulo psycopg2 de Python. Para ello, es preciso abrir la consola de Anaconda y teclear el siguiente comando, según se explica en:

https://pypi.org/project/psycopg2/

```
pip install psycopg2
```

En este caso, también será posible importar la librería de acceso a Postgresql desde Python en el código fuente que se desarrolle. Se usará el alias PBD para hacer compatible el código entre Oracle y PostgreSQL:

```
import psycopg2 as PBD
```

4. Conexión a datos en Oracle

Es preciso gestionar adecuadamente la conexión con la base de datos Oracle, de modo que se almacena la conexión local con la base de datos en una variable. Siguiendo con el mismo guion que en JAVA con JDBC, se llamará conexion. Esta variable se inicializa en la función dbConectar, y será utilizada en el resto de funciones implementadas en esta sesión práctica.

En la función de conexión, tras inicializar convenientemente las variables locales encargadas de gestionar la IP, el puerto, el identificador de sesión, y las credenciales de acceso, se invoca posteriormente a la función connect. Como se ha comentado anteriormente, el resultado de esta conexión se asigna a la variable conexion, que es la que se retorna en caso de éxito. Obsérvese que, si no se ha podido establecer la conexión con la base de datos, se retornará un valor nulo (None en Python).

```
def dbConectar():
    ip = "localhost"
    puerto = 1521
    s_id = "xe"

    usuario = "system"
    contrasena = "12345"

    print("---dbConectar---")
    print("---Conectando a Oracle---")

    try:
        conexion = oracledb.connect(user=usuario, password=contrasena,
host=ip, port=puerto, sid=s_id)
```

```
        print("Conexión realizada a la base de datos",conexion)
        print(conexion.version)
        return conexion
    except PBD.DatabaseError as error:
        print("Error en la conexión")
        print(error)
        return None
```

5. Conexión a datos en Postgresql

La versión para gestionar la conexión en Postgresql es muy similar a la de Oracle. De nuevo, la conexión local con la base de datos se gestiona en una variable llamada conexion. Esta variable se inicializa en la función dbConectar, y será utilizada en el resto de funciones implementadas en esta sesión práctica.

En el caso de Posgresql, es preciso indicar la IP, el puerto, y el nombre de la base de datos, además de las credenciales de acceso. Después se invoca a la función connect. El resultado de esta conexión se asigna a la variable conexion, que es la que se retorna en caso de éxito. Si no se ha podido establecer la conexión con la base de datos, se retornará un valor nulo (None en Python).

```
def dbConectar():
    ip = "localhost"
    puerto = 5432
    basedatos = "Empresa"

    usuario = "postgres"
    contrasena = "12345"

    print("---dbConectar---")
    print("---Conectando a Postgresql---")

    try:
        conexion = psycopg2.connect(user=usuario, password=contrasena,
host=ip, port=puerto, database=basedatos)
        print("Conexión realizada a la base de datos",conexion)
        #print(conexion.version)
        return conexion
    except PBD.DatabaseError as error:
        print("Error en la conexión")
        print(error)
        return None
```

6. Desconexión

La función dbDesconectar es idéntica para ambos SGBD. A modo de ejemplo, para Oracle quedaría con la siguiente implementación, devolviendo un valor booleano indicando si ha tenido éxito la desconexión:

```
def dbDesconectar():
    print("---dbDesconectar---")
    try:
        conexion.commit()
        conexion.close()
        print("Desconexión realizada correctamente")
        return True
    except PBD.DatabaseError as error:
        print("Error en la desconexión")
        print(error)
        return False
```

7. Sentencias y ejecución de consultas

En el programa principal se llamará a la función de conexión, en primer lugar. Después se incluirán todas las funciones de trabajo y, finalmente, la llamada a la función de desconexión de la base de datos:

```
print("---Programa principal---")

conexion=dbConectar()

if (conexion is None):
    print("HA HABIDO ERROR")
else:
    print("NO HA HABIDO ERROR")

    # ...

    dbDesconectar()
print("---Fin de programa---")
```

7.1. Ejecución de sentencias SQL

El acceso a datos desde Python se realiza principalmente mediante la ejecución de sentencias SQL. Las sentencias se ejecutan utilizando principalmente los métodos `execute()` o `executemany()`. Las sentencias pueden ser consultas, sentencias de lenguaje de manipulación de datos (DML) y también de lenguaje de definición de datos (DDL). También se pueden ejecutar algunas otras declaraciones especiales.

Las sentencias SQL **no deben contener un punto y coma al final** (";") o una barra inclinada ("/") del estilo:

```
    cursor.execute("select * from Empleados;")
```

En este caso, debería especificarse:

```
    cursor.execute("select * from Empleados")
```

Como puede apreciarse, es preciso usar iteradores (en este caso, denominado `cursor`, para poder recorrer el resultado de la consulta). En este sentido, indicar que el procedimiento para realizar consultas en Python sigue el siguiente modelo:

1. Abrir (o crear) la conexión con la base de datos (`dbConectar`)
2. Crear un cursor
3. Ejecutar la consulta SQL
4. Manejar los resultados de la consulta
5. Cerrar cursor
6. Cerrar la conexión con la base de datos (`dbDesconectar`)

En los siguientes ejemplos, se entiende que la conexión y desconexión a la base de datos se realiza mediante las funciones `dbConectar`/`dbDesconectar`, implementadas anteriormente.

7.2. Consultas SQL

Las consultas solo se pueden ejecutar utilizando el método `execute()`. Tras la solicitud de ejecución, las tuplas se pueden recuperar usando alguno de estos métodos: `fetchone()`, `fetchmany()` o `fetchall()`.

- `fetchone()` permite recuperar una tupla de las solicitadas a la base de datos
- `fetchmany(N)` permite recuperar N tuplas de la base de datos
- `fetchall()` permite recuperar todas las tuplas solicitadas a la base de datos

A continuación, se presentan ejemplos de los diferentes modos de recuperación de tuplas (para la tabla `Empleados`) de Oracle desde Python.

El modo más simple consiste en ejecutar la consulta (con el método `execute`) y manejar los resultados mostrándolos en orden, mediante un bucle `for`. Como puede apreciarse, tanto la consulta como el manejo de resultados se encuentran "encerrados" entre la creación y el cierre de un iterador (denominado `cursor` en este caso).

```
def dbMostrarEmpleados1():
    print("---dbMostrarEmpleados1---")

    try:
        cursor = conexion.cursor()
        consulta = "SELECT * FROM Empleados"
        cursor.execute(consulta)
        for tupla in cursor:
            print(tupla)
        print("Número de registros recuperados:",cursor.rowcount)
        cursor.close()
    except PBD.DatabaseError as error:
        print("Error en dbMostrarEmpleados1")
        print(error)
```

La salida por consola será la siguiente:

ORACLE

```
---dbMostrarEmpleados1---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
('666666666', 'Lagos', '07-07-1991', '06800', 'M', 27000.0, 2)
('777777777', 'Salazar', '22-07-1993', '06300', 'H', 32000.0, 2)
('888888888', 'López', '10-11-1994', '10300', 'H', 32000.0, 2)
('123456789', 'Pérez', '15-11-1967', '06400', 'H', 36000.0, 3)
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', 37000.0, 3)
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', 25000.0, 3)
('999999999', 'Simón', '31-08-1988', '10600', 'M', 33000.0, 3)
('333445555', 'Campos', '12-04-1974', '06002', 'M', 50000.0, 4)
('222447777', 'Torres', '30-05-1988', '10600', 'H', 25000.0, 4)
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', 40000.0, 4)
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', 45000.0, 4)
Número de registros recuperados: 16
------------------------------
```

POSTGRESQL

```
---dbMostrarEmpleados1---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', Decimal('35000.00'), Decimal('1'))
('222222222', 'Martínez', '12-12-1991', '06800', 'M', Decimal('40000.00'), Decimal('1'))
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', Decimal('30000.00'), Decimal('1'))
('444444444', 'González', '12-09-1994', '06002', 'H', Decimal('28000.00'), Decimal('1'))
('555555555', 'Martín', '11-03-1989', '10005', 'M', Decimal('29000.00'), Decimal('2'))
('666666666', 'Lagos', '07-07-1991', '06800', 'M', Decimal('27000.00'), Decimal('2'))
('777777777', 'Salazar', '22-07-1993', '06300', 'H', Decimal('32000.00'), Decimal('2'))
('888888888', 'López', '10-11-1994', '10300', 'H', Decimal('32000.00'), Decimal('2'))
('123456789', 'Pérez', '15-11-1967', '06400', 'H', Decimal('36000.00'), Decimal('3'))
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', Decimal('37000.00'), Decimal('3'))
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', Decimal('25000.00'), Decimal('3'))
('999999999', 'Simón', '31-08-1988', '10600', 'M', Decimal('33000.00'), Decimal('3'))
('333445555', 'Campos', '12-04-1974', '06002', 'M', Decimal('50000.00'), Decimal('4'))
('222447777', 'Torres', '30-05-1988', '10600', 'H', Decimal('25000.00'), Decimal('4'))
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', Decimal('40000.00'), Decimal('4'))
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', Decimal('45000.00'), Decimal('4'))
Número de registros recuperados: 16
------------------------------
```

Como en los ejemplos se dispone de un número muy reducido de tuplas, no habrá problemas de tamaño ni de tiempo en las recuperaciones. Pero el ejemplo anterior no sería el más indicado si la tabla `Empleados` tuviese, por ejemplo, 40.000.000 de tuplas.

En muchas ocasiones resulta más conveniente recuperar las tuplas de la base de datos de una en una. Aquí es donde entra en juego el uso de los cursores para el manejo adecuado de los resultados obtenidos tras consultar una base de datos. En el segundo ejemplo, entre la declaración y el cierre del iterador se incluye la ejecución de la consulta y su manejo, en este caso recuperando una única tupla (`fetchone`) de las solicitadas a la base de datos.

```
def dbMostrarEmpleados2():
    print("---dbMostrarEmpleados2---")

    try:
        cursor = conexion.cursor()
        consulta = "SELECT * FROM Empleados"
        cursor.execute(consulta)
        tupla = cursor.fetchone()
        while tupla:
            print(tupla)
            tupla = cursor.fetchone()
        print("Número de registros recuperados:",cursor.rowcount)
```

```
        print('------------------------------')
        cursor.close()
    except PBD.DatabaseError as error:
        print("Error en dbMostrarEmpleados2")
        print(error)
```

La salida por consola para Oracle será la siguiente:

```
---dbMostrarEmpleados2---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
('666666666', 'Lagos', '07-07-1991', '06800', 'M', 27000.0, 2)
('777777777', 'Salazar', '22-07-1993', '06300', 'H', 32000.0, 2)
('888888888', 'López', '10-11-1994', '10300', 'H', 32000.0, 2)
('123456789', 'Pérez', '15-11-1967', '06400', 'H', 36000.0, 3)
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', 37000.0, 3)
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', 25000.0, 3)
('999999999', 'Simón', '31-08-1988', '10600', 'M', 33000.0, 3)
('333445555', 'Campos', '12-04-1974', '06002', 'M', 50000.0, 4)
('222447777', 'Torres', '30-05-1988', '10600', 'H', 25000.0, 4)
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', 40000.0, 4)
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', 45000.0, 4)
Número de registros recuperados: 16
------------------------------
```

Otra posibilidad sería recuperar un número determinado de tuplas de la base de datos. En el tercer ejemplo, como siempre entre la declaración y el cierre del iterador se incluye la ejecución de la consulta y su manejo, en este caso recuperando N tuplas (5 en este caso, mediante el método `fetchmany`).

```
def dbMostrarEmpleados3():
    print("---dbMostrarEmpleados3---")

    try:
        cursor = conexion.cursor()
        consulta = "SELECT * FROM Empleados"
        cursor.execute(consulta)
        numTuplas = 5
        resul = cursor.fetchmany(numTuplas)
        for tupla in resul:
            print(tupla)
        print("Número de registros seleccionados:",len(resul))
        print("Número de registros recuperados:",cursor.rowcount)
        print('------------------------------')
        cursor.close()
    except PBD.DatabaseError as error:
        print("Error en dbMostrarEmpleados3")
        print(error)
```

La salida por consola para Oracle será la siguiente:

```
---dbMostrarEmpleados3---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
Número de registros seleccionados: 5
Número de registros recuperados: 5
------------------------------
```

Si se desea recuperar todas las tuplas de la base de datos y procesarlas mediante cursores, se hará uso del método `fetchall`. Es importante destacar que, en este caso, se debe asegurar que **todas las tuplas puedan alojarse en memoria**. Como se comentó anteriormente, en los ejemplos de estas sesiones prácticas, la tabla `Empleados` contiene muy pocas tuplas, cuando es posible que en entornos profesionales haya tablas que puedan contener millones de tuplas.

```
def dbMostrarEmpleados4():
    print("---dbMostrarEmpleados4---")

    try:
        cursor = conexion.cursor()
        consulta = "SELECT * FROM Empleados"
        cursor.execute(consulta)
        resul = cursor.fetchall()
        for tupla in resul:
            print(tupla)
        print("Número de registros recuperados:",len(resul))
        print("Número de registros recuperados:",cursor.rowcount)
        print('------------------------------')
        cursor.close()
    except PBD.DatabaseError as error:
        print("Error en dbMostrarEmpleados4")
        print(error)
```

En este caso, la salida por consola para Oracle será la siguiente:

```
---dbMostrarEmpleados4---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
('666666666', 'Lagos', '07-07-1991', '06800', 'M', 27000.0, 2)
('777777777', 'Salazar', '22-07-1993', '06300', 'H', 32000.0, 2)
('888888888', 'López', '10-11-1994', '10300', 'H', 32000.0, 2)
('123456789', 'Pérez', '15-11-1967', '06400', 'H', 36000.0, 3)
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', 37000.0, 3)
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', 25000.0, 3)
('999999999', 'Simón', '31-08-1988', '10600', 'M', 33000.0, 3)
('333445555', 'Campos', '12-04-1974', '06002', 'M', 50000.0, 4)
('222447777', 'Torres', '30-05-1988', '10600', 'H', 25000.0, 4)
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', 40000.0, 4)
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', 45000.0, 4)
Número de registros recuperados: 16
Número de registros recuperados: 16
------------------------------
```

7.3. Sentencias de inserción, modificación y borrado

Las sentencias `INSERT` / `UPDATE` / `DELETE` se pueden ejecutar fácilmente siguiendo el mismo modelo visto para las consultas. Por ejemplo, para insertar un departamento en la tabla `Departamentos`, bastaría con indicar:

ORACLE

```
cursor = conexion.cursor()
numeroD = input("Código del departamento: ") # por ejemplo: 0
nombreD = input("Nombre del departamento: ") # por ejemplo: CONTABILIDAD
costeD = 0.0
sueldoD = 0.0
```

```
sentencia = "INSERT INTO Departamentos VALUES(:numeroD, :nombreD, :costeD, :sueldoD)"
cursor.execute(sentencia, [numeroD, nombreD, costeD, sueldoD])
```

Como puede apreciarse, es preciso realizar una **vinculación de variables** de la sentencia SQL con variables de Python. En **ORACLE**, en la cadena `sentencia` se especifican las variables anteponiéndoles dos puntos "**:**". La vinculación de estos valores (`:numeroD`, `:nombreD`, `:costeD`, `:sueldoD`) con las variables de Python (`numeroD`, `nombreD`, `costeD`, `sueldoD`) se realiza en la llamada `execute`. Para ello, es preciso indicar entre **[]** los nombres de las variables de Python (`numeroD`, `nombreD`, `costeD`, `sueldoD`) que se vincularán con las variables de la sentencia SQL (`:numeroD`, `:nombreD`, `:costeD`, `:sueldoD`).

La vinculación de parámetros en **POSTGRESQL** difiere ligeramente. En este caso, la vinculación se realiza especificando los parámetros con `%s`, tantas veces como tantos parámetros haya que vincular. Entre **[]** se especifican los nombres de las variables de Python (`numeroD`, `nombreD`, `costeD`, `sueldoD`) que se vincularán con las especificaciones de `%s`.

POSTGRESQL

```
cursor =  conexion.cursor()
numeroD = input("Código del departamento: ") # por ejemplo: 0
nombreD = input("Nombre del departamento: ") # por ejemplo: CONTABILIDAD
costeD = 0.0
sueldoD = 0.0

sentencia = "INSERT INTO Departamentos VALUES(%s, %s, %s, %s)"
cursor.execute(sentencia, [numeroD, nombreD, costeD, sueldoD])
```

Con independencia del SGBD, es importante realizar la **vinculación de variables** de este modo, en lugar de concatenar cadenas del estilo:

NO RECOMENDADO:

```
nombreD = "FINANZAS" # Cambiar el nombre 'CONTABILIDAD' por 'FINANZAS'
consulta = "UPDATE Departamentos SET nombredpto = '"+nombreD+"' WHERE numerodpto = 0"
cursor.execute(consulta)
```

La concatenación de cadenas a la hora de construir sentencias supone un **riesgo para la seguridad**, así como un impacto en el rendimiento. Se podrían producir inyecciones de código no deseado a la hora de construir finalmente la consulta, por lo que se recomienda que la sentencia se construya vinculando variables, para reducir riesgos de seguridad.

RECOMENDADO:

ORACLE

```
nombreD = "FINANZAS" # Cambiar el nombre 'CONTABILIDAD' por 'FINANZAS'
consulta = "UPDATE Departamentos SET nombredpto = :nombreD WHERE numerodpto = 0"
cursor.execute(consulta,[nombreD])
```

POSTGRESQL

```
nombreD = "FINANZAS" # Cambiar el nombre 'CONTABILIDAD' por 'FINANZAS'
consulta = "UPDATE Departamentos SET nombredpto = %s WHERE numerodpto = 0"
cursor.execute(consulta,[nombreD])
```

La implementación de las funciones correspondientes a la inserción, modificación y borrado se dejan propuestas como ejercicios de esta sesión práctica.

8. Ejercicios de Python con acceso a datos

Se considera el siguiente código para la función principal, donde se irán poniendo entre comentarios las llamadas a las funciones a medida que se vayan implementando, según los ejercicios propuestos.

```
print("---Programa principal---")

conexion=dbConectar()

if (conexion is None):
    print("ERROR DE CONEXIÓN")
else:
    print("CONEXIÓN REALIZADA")

    dbMostrarEmpleados1()
    dbMostrarEmpleados2()
    dbMostrarEmpleados3()
    dbMostrarEmpleados4()
    dbMostrarEmpleados5()

    dbObtenerEmpleados()
    dbConsultarEmpleados()
    dbConsultarDepartamentos()

    dbInsertarDepartamentos()
    dbConsultarDepartamentos()

    dbModificarDepartamentos()
    dbConsultarDepartamentos()

    dbBorrarDepartamentos()
    dbConsultarDepartamentos()

    dbInsertarMultiplesDepartamentos()
    dbConsultarDepartamentos()

    dbBorrarMultiplesDepartamentos()
    dbConsultarDepartamentos()

    dbDesconectar()

print("---Fin de programa---")
```

Se proponen los siguientes ejercicios, que **deben ser resueltos tanto para acceder a datos en Oracle como para Postgresql**.

A) Implementar una función nueva (`dbObtenerEmpleados`) para consultar los datos de una tupla de la tabla `Empleados`, siguiendo el siguiente guion:

```
def dbObtenerEmpleados():
    print("---dbObtenerEmpleados---")

    # Por ejemplo, buscar Empleados con código 123456789
    dniObjetivo = input("Introduce código de Empleados: ")

    try:
        ...
    except ... as error:
        print("Error. No se ha podido consultar la tupla", dniObjetivo)
```

```
        print(error)
```

De modo que produzca una salida similar a la siguiente (suponiendo que se introduzca como `dniObjetivo` 123456789:

```
---dbObtenerEmpleados---
Introduce dni de Empleados: 123456789
DNI:  123456789
Nombre:  Pérez
Fecha Nacimiento:  15-11-1967
CP:  06400
Sexo:  H
Sueldo:  36000.0
Numdept:  3
------------------------------
Número de registros recuperados: 1
------------------------------
```

B) Implementar una función nueva (`dbConsultarEmpleados`) para consultar todas las tuplas de la tabla `Empleados` y otra (`dbConsultarDepartamentos`) para consultar todas las tuplas de la tabla `Departamentos`:

```
def dbConsultarEmpleados():
    print("---dbConsultarEmpleados---")

    try:
        ...
    except ... as error:
        print("Error. No se han podido consultar los Empleados")
        print(error)

# ----------------------------------------------------------------------

def dbConsultarDepartamentos():
    print("---dbConsultarDepartamentos---")

    try:
        ...
    except ... as error:
        print("Error. No se han podido consultar las tuplas de Departamentos")
        print(error)
```

La salida de estas dos funciones sería similar a la mostrada a continuación:

```
---dbConsultarEmpleados---
DNI:  111111111
Nombre:  Sánchez
Fecha Nacimiento:  15-11-1997
CP:  10005
Sexo:  M
Sueldo:  35000.0
Numdept:  1
------------------------------
DNI:  222222222
Nombre:  Martínez
Fecha Nacimiento:  12-12-1991
CP:  06800
Sexo:  M
Sueldo:  40000.0
Numdept:  1
------------------------------
DNI:  333333333
Nombre:  Álvarez
Fecha Nacimiento:  21-08-1990
```

```
CP:  10800
Sexo:  H
Sueldo:  30000.0
Numdept:  1
------------------------------
DNI:  444444444
Nombre:  González
Fecha Nacimiento:  12-09-1994
CP:  06002
Sexo:  H
Sueldo:  28000.0
Numdept:  1
------------------------------
DNI:  555555555
Nombre:  Martín
Fecha Nacimiento:  11-03-1989
CP:  10005
Sexo:  M
Sueldo:  29000.0
Numdept:  2
------------------------------
DNI:  666666666
Nombre:  Lagos
Fecha Nacimiento:  07-07-1991
CP:  06800
Sexo:  M
Sueldo:  27000.0
Numdept:  2
------------------------------
DNI:  777777777
Nombre:  Salazar
Fecha Nacimiento:  22-07-1993
CP:  06300
Sexo:  H
Sueldo:  32000.0
Numdept:  2
------------------------------
DNI:  888888888
Nombre:  López
Fecha Nacimiento:  10-11-1994
CP:  10300
Sexo:  H
Sueldo:  32000.0
Numdept:  2
------------------------------
DNI:  123456789
Nombre:  Pérez
Fecha Nacimiento:  15-11-1967
CP:  06400
Sexo:  H
Sueldo:  36000.0
Numdept:  3
------------------------------
DNI:  666884444
Nombre:  Ojeda
Fecha Nacimiento:  12-12-1991
CP:  06300
Sexo:  H
Sueldo:  37000.0
Numdept:  3
------------------------------
DNI:  666999333
Nombre:  Ruiz
Fecha Nacimiento:  01-02-1990
CP:  10300
Sexo:  H
Sueldo:  25000.0
Numdept:  3
------------------------------
DNI:  999999999
Nombre:  Simón
Fecha Nacimiento:  31-08-1988
CP:  10600
```

```
Sexo:  M
Sueldo:  33000.0
Numdept:  3
------------------------------
DNI:  333445555
Nombre:  Campos
Fecha Nacimiento:  12-04-1974
CP:  06002
Sexo:  M
Sueldo:  50000.0
Numdept:  4
------------------------------
DNI:  222447777
Nombre:  Torres
Fecha Nacimiento:  30-05-1988
CP:  10600
Sexo:  H
Sueldo:  25000.0
Numdept:  4
------------------------------
DNI:  987654321
Nombre:  Jiménez
Fecha Nacimiento:  10-04-1971
CP:  06400
Sexo:  M
Sueldo:  40000.0
Numdept:  4
------------------------------
DNI:  000000000
Nombre:  Sevilla
Fecha Nacimiento:  17-04-1980
CP:  10800
Sexo:  M
Sueldo:  45000.0
Numdept:  4
------------------------------
Número de registros recuperados: 16
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 4
------------------------------
```

C) Implementar una función nueva (`dbInsertarDepartamentos`) para insertar una nueva tupla en la tabla `Departamentos`.

```
def dbInsertarDepartamentos():
    print("---dbInsertarDepartamentos---")

    try:
        ...
```

```
        print("Tupla insertada correctamente")
    except ... as error:
        print("Error. No se ha podido insertar el Departamento")
        print(error)
```

De modo que produzca una salida similar a la siguiente (suponiendo que se introduzca como código de departamento 0 y como nombre CONTABILIDAD):

```
---dbInsertarDepartamentos---
Código del departamento: 0
Nombre del departamento: CONTABILIDAD
Tupla insertada correctamente
------------------------------
```

D) Implementar una función nueva (dbModificarDepartamentos) para modificar la tupla que se acaba de insertar en la tabla Departamentos. Por ejemplo, cambiar el nombre del departamento 0 a FINANZAS.

```
def dbModificarDepartamentos():
    print("---dbModificarDepartamentos---")

    try:
        ...
        print("Tupla modificada correctamente")
    except ... as error:
        print("Error. No se ha podido modificar el Departamento")
        print(error)
```

De modo que produzca una salida similar a la siguiente:

```
---dbModificarDepartamentos---
Tupla modificada correctamente
```

E) Implementar una función nueva (dbBorrarDepartamentos) para borrar la tupla que se ha insertado y modificado en la tabla Departamentos (la tupla con número de departamento 0).

```
def dbBorrarDepartamentos():
    print("---dbBorrarDepartamentos---")

    try:
        ...
        print("Tupla borrada")
    except ... as error:
        print("Error. No se ha podido borrar el Departamento")
        print(error)
```

De modo que produzca una salida similar a la siguiente, tras borrar esta función al departamento 0:

```
---dbBorrarDepartamentos---
Tupla borrada
```

F) Implementar una función nueva (dbInsertarMultiplesDepartamentos) para insertar múltiples tuplas:

```
def dbInsertarMultiplesDepartamentos():
    print("---dbInsertarMultiplesDepartamentos---")
```

```
    datos = [
        ('5', 'INVESTIGACIÓN',0.0,0.0),
        ('6', 'MARKETING',0.0,0.0),
        ('7', 'VENTAS',0.0,0.0)
    ]

    try:
        ...
    except ... as error:
        print("Error. No se han podido insertar múltiples Departamentos")
        print(error)
```

De modo que produzca una salida similar a la siguiente:

```
---dbInsertarMultiplesDepartamentos---
Número de registros insertados: 3
```

G) Implementar una función nueva (`dbBorrarMultiplesDepartamentos`) para borrar múltiples tuplas, aquellas que se insertaron de forma múltiple mediante la función `dbInsertaMultiplesDepartamentos`, implementada anteriormente.

```
def dbBorrarMultiplesDepartamentos():
    print('---dbBorrarMultiplesDepartamentos---')

    datos = [['5'], ['6'], ['7']]

    try:
        ...
    except ... as error:
        print("Error. No se han podido borrar múltiples Departamentos")
        print(error)
```

De modo que produzca una salida similar a la siguiente:

```
---dbBorrarMultiplesDepartamentos---
Número de registros borrados: 3
```

9. Salida en la consola

De acuerdo con el programa principal propuesto y las diferentes funciones implementadas, a continuación, se muestra la salida final (en fondo amarillo se destacan las salidas por función; en fondo verde se destacan los datos introducidos por teclado; en fondo azul se destacan las comprobaciones sobre las inserciones y borrados realizados en las tablas):

ORACLE

```
---Programa principal---
---dbConectar---
---Conectando a Oracle---
Conexión realizada a la base de datos <oracledb.Connection to
system@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SID=xe)))>
CONEXIÓN REALIZADA
---dbMostrarEmpleados1---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
('666666666', 'Lagos', '07-07-1991', '06800', 'M', 27000.0, 2)
('777777777', 'Salazar', '22-07-1993', '06300', 'H', 32000.0, 2)
```

```
('888888888', 'López', '10-11-1994', '10300', 'H', 32000.0, 2)
('123456789', 'Pérez', '15-11-1967', '06400', 'H', 36000.0, 3)
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', 37000.0, 3)
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', 25000.0, 3)
('999999999', 'Simón', '31-08-1988', '10600', 'M', 33000.0, 3)
('333445555', 'Campos', '12-04-1974', '06002', 'M', 50000.0, 4)
('222447777', 'Torres', '30-05-1988', '10600', 'H', 25000.0, 4)
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', 40000.0, 4)
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', 45000.0, 4)
Número de registros recuperados: 16
------------------------------
---dbMostrarEmpleados2---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
('666666666', 'Lagos', '07-07-1991', '06800', 'M', 27000.0, 2)
('777777777', 'Salazar', '22-07-1993', '06300', 'H', 32000.0, 2)
('888888888', 'López', '10-11-1994', '10300', 'H', 32000.0, 2)
('123456789', 'Pérez', '15-11-1967', '06400', 'H', 36000.0, 3)
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', 37000.0, 3)
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', 25000.0, 3)
('999999999', 'Simón', '31-08-1988', '10600', 'M', 33000.0, 3)
('333445555', 'Campos', '12-04-1974', '06002', 'M', 50000.0, 4)
('222447777', 'Torres', '30-05-1988', '10600', 'H', 25000.0, 4)
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', 40000.0, 4)
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', 45000.0, 4)
Número de registros recuperados: 16
------------------------------
---dbMostrarEmpleados3---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
Número de registros seleccionados: 5
Número de registros recuperados: 5
------------------------------
---dbMostrarEmpleados4---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', 35000.0, 1)
('222222222', 'Martínez', '12-12-1991', '06800', 'M', 40000.0, 1)
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', 30000.0, 1)
('444444444', 'González', '12-09-1994', '06002', 'H', 28000.0, 1)
('555555555', 'Martín', '11-03-1989', '10005', 'M', 29000.0, 2)
('666666666', 'Lagos', '07-07-1991', '06800', 'M', 27000.0, 2)
('777777777', 'Salazar', '22-07-1993', '06300', 'H', 32000.0, 2)
('888888888', 'López', '10-11-1994', '10300', 'H', 32000.0, 2)
('123456789', 'Pérez', '15-11-1967', '06400', 'H', 36000.0, 3)
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', 37000.0, 3)
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', 25000.0, 3)
('999999999', 'Simón', '31-08-1988', '10600', 'M', 33000.0, 3)
('333445555', 'Campos', '12-04-1974', '06002', 'M', 50000.0, 4)
('222447777', 'Torres', '30-05-1988', '10600', 'H', 25000.0, 4)
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', 40000.0, 4)
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', 45000.0, 4)
Número de registros recuperados: 16
Número de registros recuperados: 16
------------------------------
---dbObtenerEmpleados---
Introduce dni de Empleados: 123456789
DNI:  123456789
Nombre:  Pérez
Fecha Nacimiento:  15-11-1967
CP:  06400
Sexo:  H
Sueldo:  36000.0
Numdept:  3
------------------------------
Número de registros recuperados: 1
------------------------------
---dbConsultarEmpleados---
DNI:  111111111
```

```
Nombre:  Sánchez
Fecha Nacimiento:  15-11-1997
CP:  10005
Sexo:  M
Sueldo:  35000.0
Numdept:  1
------------------------------
DNI:  222222222
Nombre:  Martínez
Fecha Nacimiento:  12-12-1991
CP:  06800
Sexo:  M
Sueldo:  40000.0
Numdept:  1
------------------------------
DNI:  333333333
Nombre:  Álvarez
Fecha Nacimiento:  21-08-1990
CP:  10800
Sexo:  H
Sueldo:  30000.0
Numdept:  1
------------------------------
DNI:  444444444
Nombre:  González
Fecha Nacimiento:  12-09-1994
CP:  06002
Sexo:  H
Sueldo:  28000.0
Numdept:  1
------------------------------
DNI:  555555555
Nombre:  Martín
Fecha Nacimiento:  11-03-1989
CP:  10005
Sexo:  M
Sueldo:  29000.0
Numdept:  2
------------------------------
DNI:  666666666
Nombre:  Lagos
Fecha Nacimiento:  07-07-1991
CP:  06800
Sexo:  M
Sueldo:  27000.0
Numdept:  2
------------------------------
DNI:  777777777
Nombre:  Salazar
Fecha Nacimiento:  22-07-1993
CP:  06300
Sexo:  H
Sueldo:  32000.0
Numdept:  2
------------------------------
DNI:  888888888
Nombre:  López
Fecha Nacimiento:  10-11-1994
CP:  10300
Sexo:  H
Sueldo:  32000.0
Numdept:  2
------------------------------
DNI:  123456789
Nombre:  Pérez
Fecha Nacimiento:  15-11-1967
CP:  06400
Sexo:  H
Sueldo:  36000.0
Numdept:  3
------------------------------
DNI:  666884444
Nombre:  Ojeda
```

```
Fecha Nacimiento:  12-12-1991
CP:  06300
Sexo:  H
Sueldo:  37000.0
Numdept:  3
------------------------------
DNI:  666999333
Nombre:  Ruiz
Fecha Nacimiento:  01-02-1990
CP:  10300
Sexo:  H
Sueldo:  25000.0
Numdept:  3
------------------------------
DNI:  999999999
Nombre:  Simón
Fecha Nacimiento:  31-08-1988
CP:  10600
Sexo:  M
Sueldo:  33000.0
Numdept:  3
------------------------------
DNI:  333445555
Nombre:  Campos
Fecha Nacimiento:  12-04-1974
CP:  06002
Sexo:  M
Sueldo:  50000.0
Numdept:  4
------------------------------
DNI:  222447777
Nombre:  Torres
Fecha Nacimiento:  30-05-1988
CP:  10600
Sexo:  H
Sueldo:  25000.0
Numdept:  4
------------------------------
DNI:  987654321
Nombre:  Jiménez
Fecha Nacimiento:  10-04-1971
CP:  06400
Sexo:  M
Sueldo:  40000.0
Numdept:  4
------------------------------
DNI:  000000000
Nombre:  Sevilla
Fecha Nacimiento:  17-04-1980
CP:  10800
Sexo:  M
Sueldo:  45000.0
Numdept:  4
------------------------------
Número de registros recuperados: 16
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
```

```
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 4
------------------------------
---dbInsertarDepartamentos---
Código del departamento: 0
Nombre del departamento: CONTABILIDAD
Tupla insertada correctamente
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  0
Nombredpto:  CONTABILIDAD
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 5
------------------------------
---dbModificarDepartamentos---
Tupla modificada correctamente
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  0
Nombredpto:  FINANZAS
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 5
------------------------------
---dbBorrarDepartamentos---
Tupla borrada
------------------------------
```

```
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 4
------------------------------
---dbInsertarMultiplesDepartamentos---
Número de registros insertados: 3
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  5
Nombredpto:  INVESTIGACIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  6
Nombredpto:  MARKETING
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  7
Nombredpto:  VENTAS
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 7
------------------------------
---dbBorrarMultiplesDepartamentos---
Número de registros borrados: 3
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.0
%: 0.0
------------------------------
```

```
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.0
%: 0.0
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.0
%: 0.0
------------------------------
Número de registros recuperados: 4
------------------------------
---dbDesconectar---
Desconexión realizada correctamente
---Fin de programa---
```

POSTGRESQL

```
---Programa principal---
---dbConectar---
---Conectando a Postgresql---
Conexión realizada a la base de datos <connection object at 0x000002ABC1482240; dsn:
'user=postgres password=xxx dbname=Empresa host=localhost port=5432', closed: 0>
CONEXIÓN REALIZADA
---dbMostrarEmpleados1---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', Decimal('35000.00'), Decimal('1'))
('222222222', 'Martínez', '12-12-1991', '06800', 'M', Decimal('40000.00'), Decimal('1'))
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', Decimal('30000.00'), Decimal('1'))
('444444444', 'González', '12-09-1994', '06002', 'H', Decimal('28000.00'), Decimal('1'))
('555555555', 'Martín', '11-03-1989', '10005', 'M', Decimal('29000.00'), Decimal('2'))
('666666666', 'Lagos', '07-07-1991', '06800', 'M', Decimal('27000.00'), Decimal('2'))
('777777777', 'Salazar', '22-07-1993', '06300', 'H', Decimal('32000.00'), Decimal('2'))
('888888888', 'López', '10-11-1994', '10300', 'H', Decimal('32000.00'), Decimal('2'))
('123456789', 'Pérez', '15-11-1967', '06400', 'H', Decimal('36000.00'), Decimal('3'))
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', Decimal('37000.00'), Decimal('3'))
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', Decimal('25000.00'), Decimal('3'))
('999999999', 'Simón', '31-08-1988', '10600', 'M', Decimal('33000.00'), Decimal('3'))
('333445555', 'Campos', '12-04-1974', '06002', 'M', Decimal('50000.00'), Decimal('4'))
('222447777', 'Torres', '30-05-1988', '10600', 'H', Decimal('25000.00'), Decimal('4'))
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', Decimal('40000.00'), Decimal('4'))
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', Decimal('45000.00'), Decimal('4'))
Número de registros recuperados: 16
------------------------------
---dbMostrarEmpleados2---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', Decimal('35000.00'), Decimal('1'))
('222222222', 'Martínez', '12-12-1991', '06800', 'M', Decimal('40000.00'), Decimal('1'))
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', Decimal('30000.00'), Decimal('1'))
('444444444', 'González', '12-09-1994', '06002', 'H', Decimal('28000.00'), Decimal('1'))
('555555555', 'Martín', '11-03-1989', '10005', 'M', Decimal('29000.00'), Decimal('2'))
('666666666', 'Lagos', '07-07-1991', '06800', 'M', Decimal('27000.00'), Decimal('2'))
('777777777', 'Salazar', '22-07-1993', '06300', 'H', Decimal('32000.00'), Decimal('2'))
('888888888', 'López', '10-11-1994', '10300', 'H', Decimal('32000.00'), Decimal('2'))
('123456789', 'Pérez', '15-11-1967', '06400', 'H', Decimal('36000.00'), Decimal('3'))
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', Decimal('37000.00'), Decimal('3'))
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', Decimal('25000.00'), Decimal('3'))
('999999999', 'Simón', '31-08-1988', '10600', 'M', Decimal('33000.00'), Decimal('3'))
('333445555', 'Campos', '12-04-1974', '06002', 'M', Decimal('50000.00'), Decimal('4'))
('222447777', 'Torres', '30-05-1988', '10600', 'H', Decimal('25000.00'), Decimal('4'))
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', Decimal('40000.00'), Decimal('4'))
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', Decimal('45000.00'), Decimal('4'))
Número de registros recuperados: 16
------------------------------
---dbMostrarEmpleados3---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', Decimal('35000.00'), Decimal('1'))
('222222222', 'Martínez', '12-12-1991', '06800', 'M', Decimal('40000.00'), Decimal('1'))
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', Decimal('30000.00'), Decimal('1'))
('444444444', 'González', '12-09-1994', '06002', 'H', Decimal('28000.00'), Decimal('1'))
```

```
('555555555', 'Martín', '11-03-1989', '10005', 'M', Decimal('29000.00'), Decimal('2'))
Número de registros seleccionados: 5
Número de registros recuperados: 16
------------------------------
---dbMostrarEmpleados4---
('111111111', 'Sánchez', '15-11-1997', '10005', 'M', Decimal('35000.00'), Decimal('1'))
('222222222', 'Martínez', '12-12-1991', '06800', 'M', Decimal('40000.00'), Decimal('1'))
('333333333', 'Álvarez', '21-08-1990', '10800', 'H', Decimal('30000.00'), Decimal('1'))
('444444444', 'González', '12-09-1994', '06002', 'H', Decimal('28000.00'), Decimal('1'))
('555555555', 'Martín', '11-03-1989', '10005', 'M', Decimal('29000.00'), Decimal('2'))
('666666666', 'Lagos', '07-07-1991', '06800', 'M', Decimal('27000.00'), Decimal('2'))
('777777777', 'Salazar', '22-07-1993', '06300', 'H', Decimal('32000.00'), Decimal('2'))
('888888888', 'López', '10-11-1994', '10300', 'H', Decimal('32000.00'), Decimal('2'))
('123456789', 'Pérez', '15-11-1967', '06400', 'H', Decimal('36000.00'), Decimal('3'))
('666884444', 'Ojeda', '12-12-1991', '06300', 'H', Decimal('37000.00'), Decimal('3'))
('666999333', 'Ruiz', '01-02-1990', '10300', 'H', Decimal('25000.00'), Decimal('3'))
('999999999', 'Simón', '31-08-1988', '10600', 'M', Decimal('33000.00'), Decimal('3'))
('333445555', 'Campos', '12-04-1974', '06002', 'M', Decimal('50000.00'), Decimal('4'))
('222447777', 'Torres', '30-05-1988', '10600', 'H', Decimal('25000.00'), Decimal('4'))
('987654321', 'Jiménez', '10-04-1971', '06400', 'M', Decimal('40000.00'), Decimal('4'))
('000000000', 'Sevilla', '17-04-1980', '10800', 'M', Decimal('45000.00'), Decimal('4'))
Número de registros recuperados: 16
Número de registros recuperados: 16
------------------------------
---dbObtenerEmpleados---
Introduce dni de Empleados: 123456789
DNI:  123456789
Nombre:  Pérez
Fecha Nacimiento:  15-11-1967
CP:  06400
Sexo:  H
Sueldo:  36000.00
Numdept:  3
------------------------------
Número de registros recuperados: 1
------------------------------
---dbConsultarEmpleados---
DNI:  111111111
Nombre:  Sánchez
Fecha Nacimiento:  15-11-1997
CP:  10005
Sexo:  M
Sueldo:  35000.00
Numdept:  1
------------------------------
DNI:  222222222
Nombre:  Martínez
Fecha Nacimiento:  12-12-1991
CP:  06800
Sexo:  M
Sueldo:  40000.00
Numdept:  1
------------------------------
DNI:  333333333
Nombre:  Álvarez
Fecha Nacimiento:  21-08-1990
CP:  10800
Sexo:  H
Sueldo:  30000.00
Numdept:  1
------------------------------
DNI:  444444444
Nombre:  González
Fecha Nacimiento:  12-09-1994
CP:  06002
Sexo:  H
Sueldo:  28000.00
Numdept:  1
------------------------------
DNI:  555555555
Nombre:  Martín
Fecha Nacimiento:  11-03-1989
CP:  10005
```

```
Sexo:  M
Sueldo:  29000.00
Numdept:  2
------------------------------
DNI:  666666666
Nombre:  Lagos
Fecha Nacimiento:  07-07-1991
CP:  06800
Sexo:  M
Sueldo:  27000.00
Numdept:  2
------------------------------
DNI:  777777777
Nombre:  Salazar
Fecha Nacimiento:  22-07-1993
CP:  06300
Sexo:  H
Sueldo:  32000.00
Numdept:  2
------------------------------
DNI:  888888888
Nombre:  López
Fecha Nacimiento:  10-11-1994
CP:  10300
Sexo:  H
Sueldo:  32000.00
Numdept:  2
------------------------------
DNI:  123456789
Nombre:  Pérez
Fecha Nacimiento:  15-11-1967
CP:  06400
Sexo:  H
Sueldo:  36000.00
Numdept:  3
------------------------------
DNI:  666884444
Nombre:  Ojeda
Fecha Nacimiento:  12-12-1991
CP:  06300
Sexo:  H
Sueldo:  37000.00
Numdept:  3
------------------------------
DNI:  666999333
Nombre:  Ruiz
Fecha Nacimiento:  01-02-1990
CP:  10300
Sexo:  H
Sueldo:  25000.00
Numdept:  3
------------------------------
DNI:  999999999
Nombre:  Simón
Fecha Nacimiento:  31-08-1988
CP:  10600
Sexo:  M
Sueldo:  33000.00
Numdept:  3
------------------------------
DNI:  333445555
Nombre:  Campos
Fecha Nacimiento:  12-04-1974
CP:  06002
Sexo:  M
Sueldo:  50000.00
Numdept:  4
------------------------------
DNI:  222447777
Nombre:  Torres
Fecha Nacimiento:  30-05-1988
CP:  10600
Sexo:  H
```

```
Sueldo:  25000.00
Numdept:  4
------------------------------
DNI:  987654321
Nombre:  Jiménez
Fecha Nacimiento:  10-04-1971
CP:  06400
Sexo:  M
Sueldo:  40000.00
Numdept:  4
------------------------------
DNI:  000000000
Nombre:  Sevilla
Fecha Nacimiento:  17-04-1980
CP:  10800
Sexo:  M
Sueldo:  45000.00
Numdept:  4
------------------------------
Número de registros recuperados: 16
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.00
%: 0.00
------------------------------
Número de registros recuperados: 4
------------------------------
---dbInsertarDepartamentos---
Código del departamento: 0
Nombre del departamento: CONTABILIDAD
Tupla insertada correctamente
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  0
Nombredpto:  CONTABILIDAD
Coste: 0.00
```

```
%: 0.00
------------------------------
Número de registros recuperados: 5
------------------------------
---dbModificarDepartamentos---
Tupla modificada correctamente
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  0
Nombredpto:  FINANZAS
Coste: 0.00
%: 0.00
------------------------------
Número de registros recuperados: 5
------------------------------
---dbBorrarDepartamentos---
Tupla borrada
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.00
%: 0.00
------------------------------
Número de registros recuperados: 4
------------------------------
---dbInsertarMultiplesDepartamentos---
Número de registros insertados: 3
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.00
```

```
%: 0.00
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  5
Nombredpto:  INVESTIGACIÓN
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  6
Nombredpto:  MARKETING
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  7
Nombredpto:  VENTAS
Coste: 0.00
%: 0.00
------------------------------
Número de registros recuperados: 7
------------------------------
---dbBorrarMultiplesDepartamentos---
Número de registros borrados: 3
------------------------------
---dbConsultarDepartamentos---
Numdepto:  1
Nombredpto:  PERSONAL
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  2
Nombredpto:  PRODUCCIÓN
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  3
Nombredpto:  DISEÑO
Coste: 0.00
%: 0.00
------------------------------
Numdepto:  4
Nombredpto:  DESARROLLO
Coste: 0.00
%: 0.00
------------------------------
Número de registros recuperados: 4
------------------------------
---dbDesconectar---
Desconexión realizada correctamente
---Fin de programa---
```

10. Estudio a realizar

- Probar el código facilitado y desarrollar los ejercicios propuestos en esta sesión.

EXTRA POINT: Con carácter voluntario, se presenta la posibilidad de obtener un *EXTRA POINT* a quienes deseen profundizar más en los temas presentados en este guion práctico. Para ello, en la memoria a entregar, se deberá incluir (además de los apartados obligatorios del guion) un nuevo apartado denominado EXTRA POINT e identificado con el logo BONUS, donde se presente algún aspecto novedoso o diferenciador con respecto a la propuesta práctica base.

- **Entregables**: debe entregarse

- **OPCIÓN (1)** – OBLIGATORIO

  (1) El código fuente correspondiente a los ejercicios debidamente resueltos.
  Los ejercicios deben implementarse y documentarse **para todos los SGBD**

  (2) Un PDF en el que se documente:
  a) Un anexo completo con **las salidas obtenidas tras la ejecución del script completo del código desarrollado**. **Las salidas obtenidas deben incluirse en la documentación en modo texto y no como una captura de imágenes.**

- **OPCIÓN (2)** – OPTATIVO, además del punto (1) obligatorio:

  a) Todo el **proceso de instalación** según el guion de la sesión anterior (introducción a Python).
  b) Todo el **código fuente** desarrollado y entregado en esta sesión práctica, ampliamente **comentado y documentado**, con las explicaciones necesarias y las **salidas parciales** obtenidas por cada fragmento de código. Si alguna función/procedimiento del código fuente es idéntica en todos los SGBD, es suficiente con explicar esa función/procedimiento una única vez. No es preciso documentar los ejercicios resueltos en este guion y que no formen parte del código del estudiante, de modo que solo es preciso comentar las funciones usadas en los ejercicios propuestos. **El código fuente debe incluirse en la documentación en modo texto y no como una captura de imágenes, tal y como se presenta en los guiones prácticos de la asignatura**.
  c) Particularmente, debe explicarse detalladamente todos los **aspectos novedosos** que incluyen el tema que se presenta.
  d) Todas las capturas, salidas, código… que se incluyan en la documentación deben ser **perfectamente legibles**, presentándose con la máxima claridad y corrección.