

Documentación

Encabezado

```
import sys
import cx_Oracle
```

Se hace uso de las librerías de sys y cx_Oracle, sys principalmente en los manejos de excepciones y cx_Oracle para llamar a los procedimientos, funciones, establecer un cursor y establecer una conexión con la base de datos.

```
username = 'SYSTEM'
password = 'admin'
host = 'localhost'
port = '1521'
service_name = 'XE'

dsn = cx_Oracle.makedsn(host, port, service_name)

connection = cx_Oracle.connect(username, password, dsn)

cursor = connection.cursor()
```

Funciones

Insertar empleado

Mediante la siguiente función se piden los parámetros para insertar un empleado, tanto la llave primaria que es el empno, el nombre del empleado, trabajo, empno del manager, el día, mes y año en que se contrató al empleado, salario del empleado, comisión y el departamento al que pertenece. Mediante el curso llama al procedimiento **Add_emp**. De detectarse una excepción lanza un error.

```
def insert_emp():
    print("Escribe el id del empleado:")
    empno=int(input())
    print("Escribe el nombre del empleado:")
    e_name=input()
    print("Escribe el trabajo del empleado:")
    e_job=input()
    print("Escribe el id del manager del empleado:")
```

```

e_mgr=int(input())

print("Escribe el día de contartación:")
day=int(input())
print("Escribe el mes de contartación:")
month=int(input())
print("Escribe el año de contartación:")
year=int(input())
print("Escribe el salario del empleado:")
e_sal=int(input())
print("Escribe la comisión del empleado:")
e_comm=int(input())
print("Escribe el departamento del empleado:")
e_dept=int(input())

e_date = (f"{day}-{month}-{year}")
try:
    cursor.callproc('Add_emp', (empno,e_name,e_job,
e_mgr,e_date,e_sal, e_comm,e_dept))
    print("Completado")
except cx_Oracle.DatabaseError as e:
    error, = e.args
    if error.code == 20001:
        print("Error: El ID especificado ya existe en la base
de datos.")
    else:
        print("Error inesperado:", error.message)

```

Procedimiento almacenado

El siguiente procedimiento almacenado toma los parámetros que se piden en la función anterior, verifica si existe el empno en la base de datos, en cuyo caso lanza una excepción, de lo contrario convierte el string de fecha y mediante un sql dinamico inserta al empleado en la base de datos y confirma el cambio.

```

CREATE OR REPLACE PROCEDURE Add_emp(i_empno NUMBER, ename
VARCHAR2,ejob VARCHAR2,emgr VARCHAR2,hire_date VARCHAR2,sal
NUMBER,comm NUMBER,dept NUMBER)
IS
    id_count NUMBER;
    OpSql VARCHAR2(500);
    t_date DATE;

```

```

BEGIN
    SELECT COUNT(*)
    INTO id_count
    FROM EMP
    WHERE EMPNO = i_empno;
    IF id_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El ID ya existe');
    END IF;
    t_date := TO_DATE(hire_date, 'DD-MM-YYYY');
    OpSql := 'INSERT INTO EMP VALUES
(:dnum,:enam,:ej,:emg,:dat,:s,:c,:dep)';
    EXECUTE IMMEDIATE OpSql USING
i_empno,ename,ejob,emgr,t_date,sal,comm,dept;
    COMMIT;
END;

```

Actualizar empleado

La funcion `update_emp` pide el empno que se va a modificar y de un menú de posibles campos a modificar pide que se inserte alguno, pues este parámetro se manda a la llamada del procedimiento, también se pide el nuevo valor por el que se va a cambiar. Una vez que se tienen todos los datos se llama al procedimiento `'Update_emp'` para actualizar al empleado. De detectarse un error lanza una excepción.

```

def update_emp():
    print("ID del empleado:")
    try:
        empno = int(input())
        print("Campo a cambiar:")
        print("1 Nombre del empleado")
        print("2 Trabajo del empleado")
        print("3 Manager del empleado")
        print("4 Fecha de contratacion")
        print("5 Salario")
        print("6 Comisión")
        print("7 Departamento")
        opt = int(input())
        camp = ""
        if(opt==1):
            camp="ENAME"
        elif(opt==2):
            camp="JOB"

```

```
elif(opt==3):
    camp="MGR"
elif(opt==4):
    camp="HIREDATE"
elif(opt==5):
    camp="SAL"
elif(opt==6):
    camp="COMM"
elif(opt==7):
    camp="DEPTNO"
else:
    print("NO se conoce la opción")
    exit()
if(opt!=1 and opt!=2 and opt!=4):
    try:
        print("Nuevo valor:")
        new = int(input())
    except:
        print("No es un numero")
        exit()
elif(opt==4):
    print("NUEVA FECHA:")
    print("Ingresa día:")
    try:
        day = int(input())
    except:
        print("No es un numero")
        exit()
    print("Ingresa mes:")
    try:
        month = int(input())
    except:
        print("No es un numero")
        exit()
    print("Ingresa año:")
    try:
        year = int(input())
    except:
        print("No es un numero")
        exit()
    new = (f"{day}-{month}-{year}")
else:
```

```

        print("Nuevo valor:")
        new = input()
    try:
        cursor.callproc('Update_emp', (empno, camp, new))
        print("Completado")
    except cx_Oracle.DatabaseError as e:
        error, = e.args
        if error.code == 20001:
            print("Error: El ID especificado no existe en la base
de datos.")
        else:
            print("Error inesperado:", error.message)
    except:
        print("No es un numero")

```

Procedimiento almacenado

El procedimiento mediante el empno y un campo inserta el nuevo valor en el campo mediante un sql dinámico, en caso de que empno no exista en la tabla lanza una excepción, caso contrario actualiza el campo solicitado con el nuevo valor.

```

CREATE OR REPLACE PROCEDURE Update_emp(i_empno NUMBER, camp
VARCHAR2, new_val VARCHAR2)
IS
    id_count NUMBER;
    OpSql VARCHAR2(500);
BEGIN
    SELECT COUNT(*)
    INTO id_count
    FROM EMP
    WHERE empno = i_empno;
    IF id_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El ID especificado no
existe');
    END IF;
    OpSql := 'UPDATE EMP SET ' || camp || ' = :enew WHERE empno = :eid' ;
    EXECUTE IMMEDIATE OpSql USING new_val, i_empno;
    COMMIT;
END;

```

Eliminar empleado

La función siguiente elimina a un empleado sólo mediante su empno, llama a un procedimiento llamado 'Delete_emp'. En caso de detectarse un error lanza una excepción.

```
def delete_emp():
    print("ID del empleado:")
    try:
        empno = int(input())
        try:
            cursor.callproc('Delete_emp', [empno])
            print("Completado")
        except cx_Oracle.DatabaseError as e:
            error, = e.args
            if error.code == 20001:
                print("Error: El ID especificado no existe en la base
de datos.")
            else:
                print("Error inesperado:", error.message)
    except:
        print("No es un numero")
```

Procedimiento almacenado

El siguiente procedimiento almacenado mediante el i_empno o empno que se le pase comprueba si existe en la base de datos de lo contrario lanza una excepción, en caso de que exista el empleado que se va a eliminar mediante un sql dinámico se llama a un delete que elimina al empleado según su empno.

```
CREATE OR REPLACE PROCEDURE Delete_emp(i_empno NUMBER)
IS
    id_count NUMBER;
    OpSql VARCHAR2(500);
BEGIN
    SELECT COUNT(*)
    INTO id_count
    FROM EMP
    WHERE empno = i_empno;
    IF id_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El ID especificado no
existe');
    END IF;
    OpSql := 'DELETE FROM EMP WHERE empno = :eid' ;
    EXECUTE IMMEDIATE OpSql USING i_empno;
    COMMIT;
```

```
END;
```

Insertar departamento

Mediante la siguiente función se piden los parámetros para insertar un departamento, tanto la llave primaria que es el deptno, el nombre del departamento y la localización del departamento. Mediante el curso llama al procedimiento `Add_depto`. De detectarse una excepción lanza un error ya sea si el deptno existe o si este no cumple con el estándar.

```
def insert_dept():
    print("Incerte:")
    try:
        print("ID del departamento:")
        try:
            deptno = int(input())
        except:
            print("No es un numero")
        print("Nombre del departamento:")
        dname = input()
        print("Localización del departamento:")
        loc = input()

        cursor.callproc('Add_depto', (deptno, dname, loc))

        print("Completado")
    except cx_Oracle.DatabaseError as e:
        error, = e.args
        if error.code == 20001:
            print("Error: El ID especificado ya existe en la base
de datos.")
        elif error.code == 20002:
            print("Error: El ID especificado tiene que ser
divisible entre 10.")
        else:
            print("Error inesperado:", error.message)
```

Procedimiento almacenado

El siguiente procedimiento almacenado toma los parámetros que se piden en la función anterior, verifica si existe el deptno en la base de datos, en cuyo caso lanza una excepción, también lanza una excepción si no se cumple con un estándar de que todos los departamentos deben de tener un deptno que sea divisible entre 10, si se cumplen las condiciones mediante un sql dinamico inserta al empleado en la base de datos y confirma el cambio.

```

CREATE OR REPLACE PROCEDURE Add_depto(i_deptno NUMBER, dname
VARCHAR2, loc VARCHAR2)
IS
    id_count NUMBER;
    OpSql VARCHAR2(500);
BEGIN
    SELECT COUNT(*)
    INTO id_count
    FROM DEPT
    WHERE deptno = i_deptno;
    IF id_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El ID ya existe');
    END IF;
    IF (i_deptno/10) = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'El ID debe ser divisible
entre 10');
    END IF;
    OpSql := 'INSERT INTO DEPT VALUES (:dnum,:dnam,:dloc)';
    EXECUTE IMMEDIATE OpSql USING i_deptno,dname,loc;
    COMMIT;
END;

```

Actualizar departamento

La funcion update_dept pide el deptno que se va a modificar y de un menú de posibles campos a modificar pide que se inserte alguno, pues este parámetro se manda a la llamada del procedimiento, también se pide el nuevo valor por el que se va a cambiar. Una vez que se tienen todos los datos se llama al procedimiento 'Update_depto' para actualizar al departamento. De detectarse un error lanza una excepción.

```

def update_dept():
    print("ID del departamento:")
    try:
        deptno = int(input())
        print("Campo a cambiar:")
        print("1 Nombre del departamento")
        print("2 Localización del departamento")
        opt = int(input())
        camp = ""
        if(opt==1):
            camp="DNAME"
        elif(opt==2):

```



```

        camp="LOC"
    else:
        print("NO se conoce la opción")
        exit()

    print("Nuevo valor:")
    new = input()
    try:
        cursor.callproc('Update_depto', (deptno, camp, new))
        print("Completado")
    except cx_Oracle.DatabaseError as e:
        error, = e.args
        if error.code == 20001:
            print("Error: El ID especificado no existe en la base
de datos.")
        else:
            print("Error inesperado:", error.message)
    except:
        print("No es un numero")

```

Procedimiento almacenado

El procedimiento mediante el deptno y un campo inserta el nuevo valor en el campo mediante un sql dinámico, en caso de que deptno no exista en la tabla lanza una excepción, caso contrario actualiza el campo solicitado con el nuevo valor.

```

CREATE OR REPLACE PROCEDURE Update_depto(ideptno NUMBER, camp
VARCHAR2, new_val VARCHAR2)
IS
    id_count NUMBER;
    OpSql VARCHAR2(500);
BEGIN
    SELECT COUNT(*)
    INTO id_count
    FROM DEPT
    WHERE deptno = ideptno;
    IF id_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'El ID especificado no
existe');
    END IF;
    OpSql := 'UPDATE DEPT SET ' || camp || ' = :dnew WHERE deptno = :did'
;
    EXECUTE IMMEDIATE OpSql USING new_val, ideptno;

```

```
COMMIT;  
END;
```

Eliminar empleado

La función siguiente elimina a un departamento mediante su deptno, llama a un procedimiento llamado 'Delete_depto'. En caso de detectarse un error lanza una excepción.

```
def delete_dept():  
    print("ID del departamento:")  
    try:  
        deptno = int(input())  
        try:  
            cursor.callproc('Delete_depto', [deptno])  
            print("Completado")  
        except cx_Oracle.DatabaseError as e:  
            error, = e.args  
            if error.code == 20001:  
                print("Error: El ID especificado no existe en la base  
de datos.")  
            else:  
                print("Error inesperado:", error.message)  
    except:  
        print("No es un numero")
```

Procedimiento almacenado

El siguiente procedimiento almacenado mediante el deptno que se le pasa, comprueba si existe en la base de datos de lo contrario lanza una excepción, en caso de que exista el departamento que se va a eliminar mediante un sql dinámico se llama a un delete que elimina el departamento según su deptno.

```
CREATE OR REPLACE PROCEDURE Delete_depto(ideptno NUMBER)  
IS  
    id_count NUMBER;  
    OpSql VARCHAR2(500);  
BEGIN  
    SELECT COUNT(*)  
    INTO id_count  
    FROM DEPT  
    WHERE deptno = ideptno;  
    IF id_count = 0 THEN  
        RAISE_APPLICATION_ERROR(-20001, 'El ID especificado no  
existe');
```

```

END IF;

OpSql := 'DELETE FROM DEPT WHERE deptno = :did' ;

EXECUTE IMMEDIATE OpSql USING ideptno;

COMMIT;

END;

```

Número de empleado en un departamento

Mediante la siguiente función, tomamos el deptno y ejecutamos un bloque anónimo mediante el cursor usando la función execute donde le pasamos una variable result que será almacenada en numer_empleados, usando la función noEmp_depto al cual le pasamos el deptno que se insertó, nos muestra el resultado si no se detectaron excepciones.

```

def select_num_emp_by_id():
    print("ID del departamento:")
    try:
        deptno = int(input())
        try:
            numero_empleados = cursor.var(cx_Oracle.NUMBER)
            cursor.execute("BEGIN :result := noEmp_depto(:id_dept);
END;", result=numero_empleados, id_dept=deptno)
            print("Total de empleados en ",deptno," es
",numero_empleados.getvalue())
        except cx_Oracle.DatabaseError as e:
            error, = e.args
            if error.code == 20001:
                print("Error: El ID especificado no existe en la base
de datos.")
            else:
                print("Error inesperado:", error.message)
        except:
            print("No es un numero")

```

Función almacenada

Esta función almacenada retorna el numero de empleados que se encuentran en un departamento.

```

CREATE OR REPLACE FUNCTION noEmp_depto(
    dept_id NUMBER
)
RETURN NUMBER

```

```

IS
    num_emp NUMBER;
    OpSql VARCHAR2(1000);
BEGIN

    OpSql := 'SELECT COUNT(*) FROM EMP WHERE deptno = :dept_id';

    EXECUTE IMMEDIATE OpSql INTO num_emp USING dept_id;
    RETURN num_emp;
END num_emp_by_id_dept;

```

Menú

El menú consiste en el siguiente bloque de código, el cual muestra las opciones disponibles, y mediante una serie de ifs si la entrada es igual a la condición ejecuta la función correspondiente.

```

opt=0
while (opt!=9):
    print("Menú")
    print("1.-Incertar empleado")
    print("2.-Incertar departamento")
    print("3.-Actualizar empleado")
    print("4.-Actualizar departamento")
    print("5.-Eliminar empleado")
    print("6.-Eliminar departamento")
    print("7.-Ver número de empleados asignados a departamento")
    print("Incerte una opción:")
    opt=int(input())
    if (opt==1):
        insert_emp()
    elif (opt==2):
        insert_dept()
    elif (opt==3):
        update_emp()
    elif (opt==4):
        update_dept()
    elif (opt==5):
        delete_emp()
    elif (opt==6):
        delete_dept()
    elif (opt==7):

```

```
select_num_emp_by_id()
```

Finalmente cierra el cursor y la conexión con la base de datos.

```
cursor.close()  
connection.close()
```

Pruebas

Insertar empleado

Aplicación

```
Menú
1.-Insertar empleado
2.-Insertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
1
Escribe el id del empleado:
3129
Escribe el nombre del empleado:
JHON
Escribe el trabajo del empleado:
PROG
Escribe el id del manager del empleado:
7369
Escribe el día de contartación:
12
Escribe el mes de contartación:
12
Escribe el año de contartación:
2000
Escribe el salario del empleado:
13000
Escribe la comisión del empleado:
0
Escribe el departamento del empleado:
10
Completado
```

Base de datos

■	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
■		7369 SMITH	CLERK		7902 17-DEC-80		800	<NULL> 20
■		7499 ALLEN	SALESMAN		7698 20-FEB-81	1600	300	30
■		7521 WARD	SALESMAN		7698 22-FEB-81	1250	500	30
■		7566 JONES	MANAGER		7839 02-APR-81	2975	<NULL>	20
■		7654 MARTIN	SALESMAN		7698 28-SEP-81	1250	1400	30
■		7698 BLAKE	MANAGER		7839 01-MAY-81	2850	<NULL>	30
■		7782 CLARK	MANAGER		7839 09-JUN-81	2450	<NULL>	10
■		7788 SCOTT	ANALYST		7566 19-APR-87	3000	<NULL>	20
■		7839 KING	PRESIDENT	<NULL>	17-NOV-81	5000	<NULL>	10
■		7844 TURNER	SALESMAN		7698 08-SEP-81	1500	0	30
■		7876 ADAMS	CLERK		7788 23-MAY-87	1100	<NULL>	20
■		7900 JAMES	CLERK		7698 03-DEC-81	950	<NULL>	30
■		7902 FORD	ANALYST		7566 03-DEC-81	3000	<NULL>	20
■		7934 MILLER	CLERK		7782 23-JAN-82	1300	<NULL>	10
■		3129 JHON	PROG		7369 12-DEC-00	13000	0	10

Insertar departamento

Aplicación

```
Menú
1.-Insertar empleado
2.-Insertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
2
Incerte:
ID del departamento:
50
Nombre del departamento:
DEV
Localización del departamento:
ANGELES
Completado
```

Base de datos

	DEPTNO	DNAME	LOC
	10	ACCOUNTING	NEW YORK
	20	RESEARCH	DALLAS
	30	SALES	CHICAGO
	40	OPERATIONS	BOSTON
	50	DEV	ANGELES

Actualizar empleado

Aplicación

```
Menú
1.-Incertar empleado
2.-Incertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
3
ID del empleado:
3129
Campo a cambiar:
1 Nombre del empleado
2 Trabajo del empleado
3 Manager del empleado
4 Fecha de contratacion
5 Salario
6 Comisión
7 Departamento
3
Nuevo valor:
7788
Completado
```

Base de datos

■	3129 JHON	PROG	7788 12-DEC-00	13000	0	10
---	-----------	------	----------------	-------	---	----

Actualizar departamento

Aplicación

```
Menú
1.-Incertar empleado
2.-Incertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
4
ID del departamento:
50
Campo a cambiar:
1 Nombre del departamento
2 Localización del departamento
2
Nuevo valor:
MISSISIPI
Completado
```

Base de datos

	50 DEV	MISSISIPI
---	--------	-----------

Eliminar empleado

Aplicación

```
Menú
1.-Incertar empleado
2.-Incertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
5
ID del empleado:
3129
Completado
```

Base de datos

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
	7369	SMITH	CLERK	7902	17-DEC-80	800	<NULL>	20
	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
	7566	JONES	MANAGER	7839	02-APR-81	2975	<NULL>	20
	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	<NULL>	30
	7782	CLARK	MANAGER	7839	09-JUN-81	2450	<NULL>	10
	7788	SCOTT	ANALYST	7566	19-APR-87	3000	<NULL>	20
	7839	KING	PRESIDENT	<NULL>	17-NOV-81	5000	<NULL>	10
	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
	7876	ADAMS	CLERK	7788	23-MAY-87	1100	<NULL>	20
	7900	JAMES	CLERK	7698	03-DEC-81	950	<NULL>	30
	7902	FORD	ANALYST	7566	03-DEC-81	3000	<NULL>	20
	7934	MILLER	CLERK	7782	23-JAN-82	1300	<NULL>	10

Eliminar departamento

Aplicación

```
Menú
1.-Incertar empleado
2.-Incertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
6
ID del departamento:
50
Completado
```

Base de datos

<input type="checkbox"/>	DEPTNO	DNAME	LOC
<input type="checkbox"/>	10	ACCOUNTING	NEW YORK
<input type="checkbox"/>	20	RESEARCH	DALLAS
<input type="checkbox"/>	30	SALES	CHICAGO
<input type="checkbox"/>	40	OPERATIONS	BOSTON

Ver número de empleados en un departamento

Aplicación

```
Menú
1.-Incertar empleado
2.-Incertar departamento
3.-Actualizar empleado
4.-Actualizar departamento
5.-Eliminar empleado
6.-Eliminar departamento
7.-Ver número de empleados asignados a departamento
Incerte una opción:
7
ID del departamento:
10
Total de empleados en 10 es 3.0
```