

PROGRAMACION

# EJERCICIOS

ORACLE®  
PL/SQL



ABRIL 2019

# BASE DE DATOS

## Indice

EJERCICIO DE PL/SQL 2 Oracle.	3
EJERCICIO 1	4
EJERCICIO 2	5
EJERCICIO 2	6
BIOGRAFÍA	6

# BASE DE DATOS

## EJERCICIO DE PL/SQL Oracle.

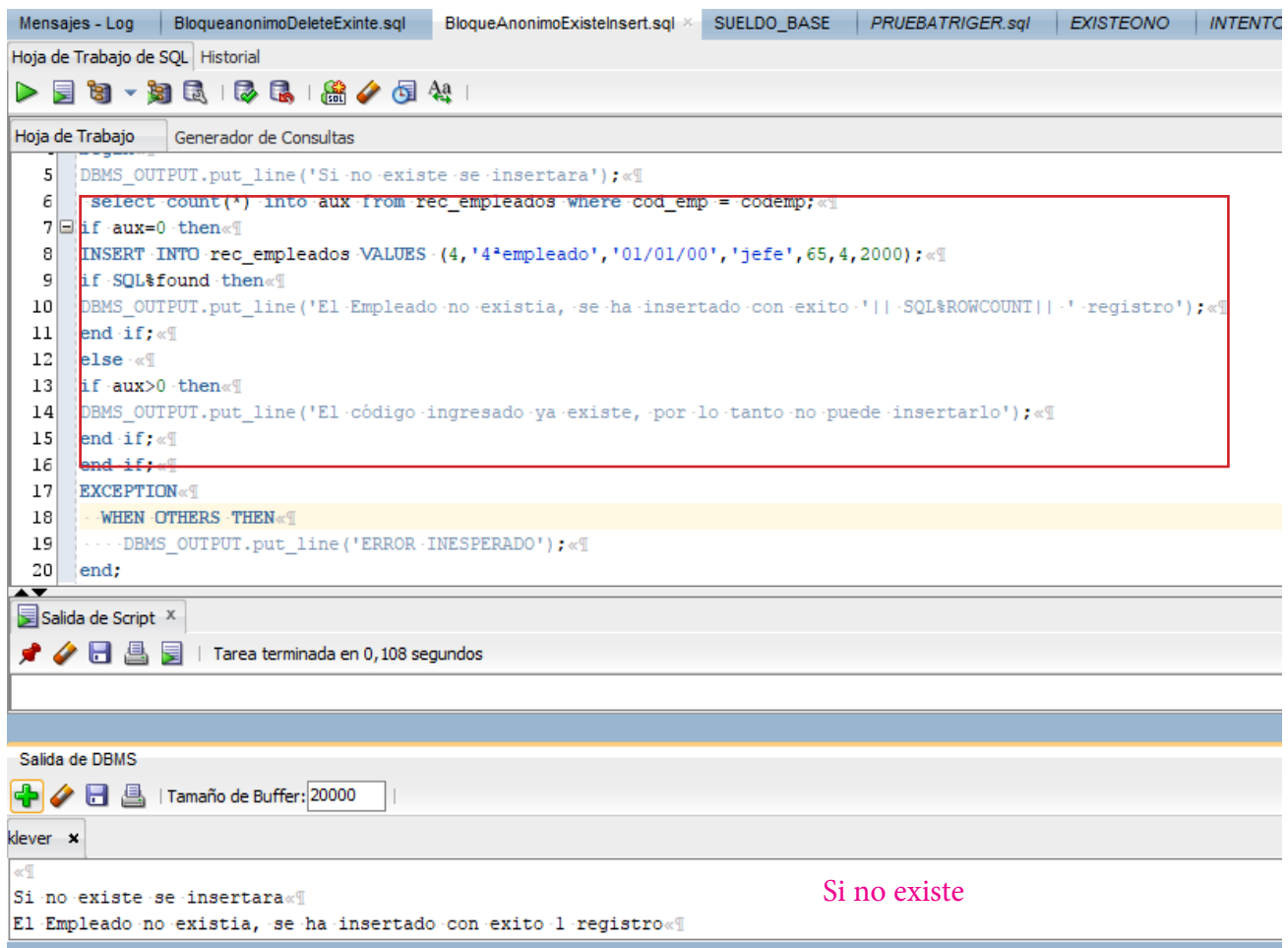
- 1) Implementar un bloque PL/SQL anónimo que
  - Verifique que una fila no existe.
  - Si no existe que la inserte.
- 2) Implementar bloque PL/SQL anónimo que dé de baja una fila comprobando previamente que existe.
- 3) Explicar como funciona las colecciones en PL/SQL con un ejemplo.
  - a) Leer todos los datos de los empleados y almacenarlos en memoria.
- b) Una vez leído todos los empleados recorrer las variables en memoria y mostrar por pantalla la información de cada empleado

# BASE DE DATOS

## Ejercicio 1

- 1) Implementar un bloque PL/SQL anónimo que
  - Verifique que una fila no existe.
  - Si no existe que la inserte.

Se a utilizado un select que cuenta las fila que esta condicionada con la palabra clave, ademas de SQL%FOUND para comprobar si el insert se completo.



```
5 DBMS_OUTPUT.put_line('Si no existe se insertara');«  
6 select count(*) into aux from rec_empleados where cod_emp = codemp;«  
7 if aux=0 then«  
8 INSERT INTO rec_empleados VALUES (4,'4*empleado','01/01/00','jefe',65,4,2000);«  
9 if SQL%found then«  
10 DBMS_OUTPUT.put_line('El Empleado no existia, se ha insertado con exito'||SQL%ROWCOUNT||' registro');«  
11 end if;«  
12 else«  
13 if aux>0 then«  
14 DBMS_OUTPUT.put_line('El código ingresado ya existe, por lo tanto no puede insertarlo');«  
15 end if;«  
16 end if;«  
17 EXCEPTION«  
18 WHEN OTHERS THEN«  
19 DBMS_OUTPUT.put_line('ERROR INESPERADO');«  
20 end;
```

Salida de Script x

Tarea terminada en 0,108 segundos

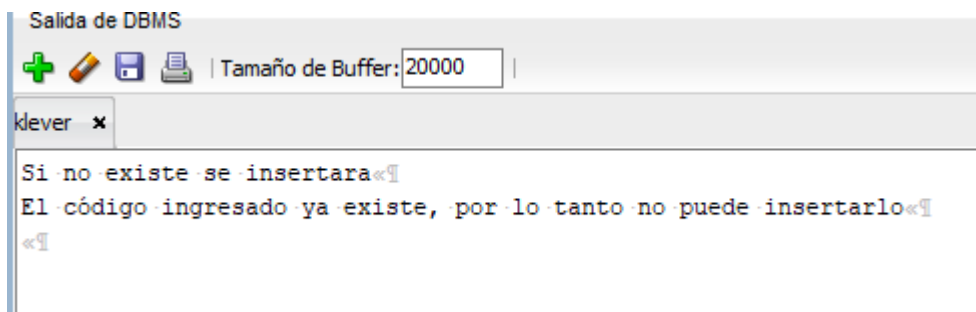
Salida de DBMS

Tamaño de Buffer: 20000

klever x

«  
Si no existe se insertara«  
El Empleado no existia, se ha insertado con exito 1 registro«

Si no existe



```
Salida de DBMS
```

Tamaño de Buffer: 20000

klever x

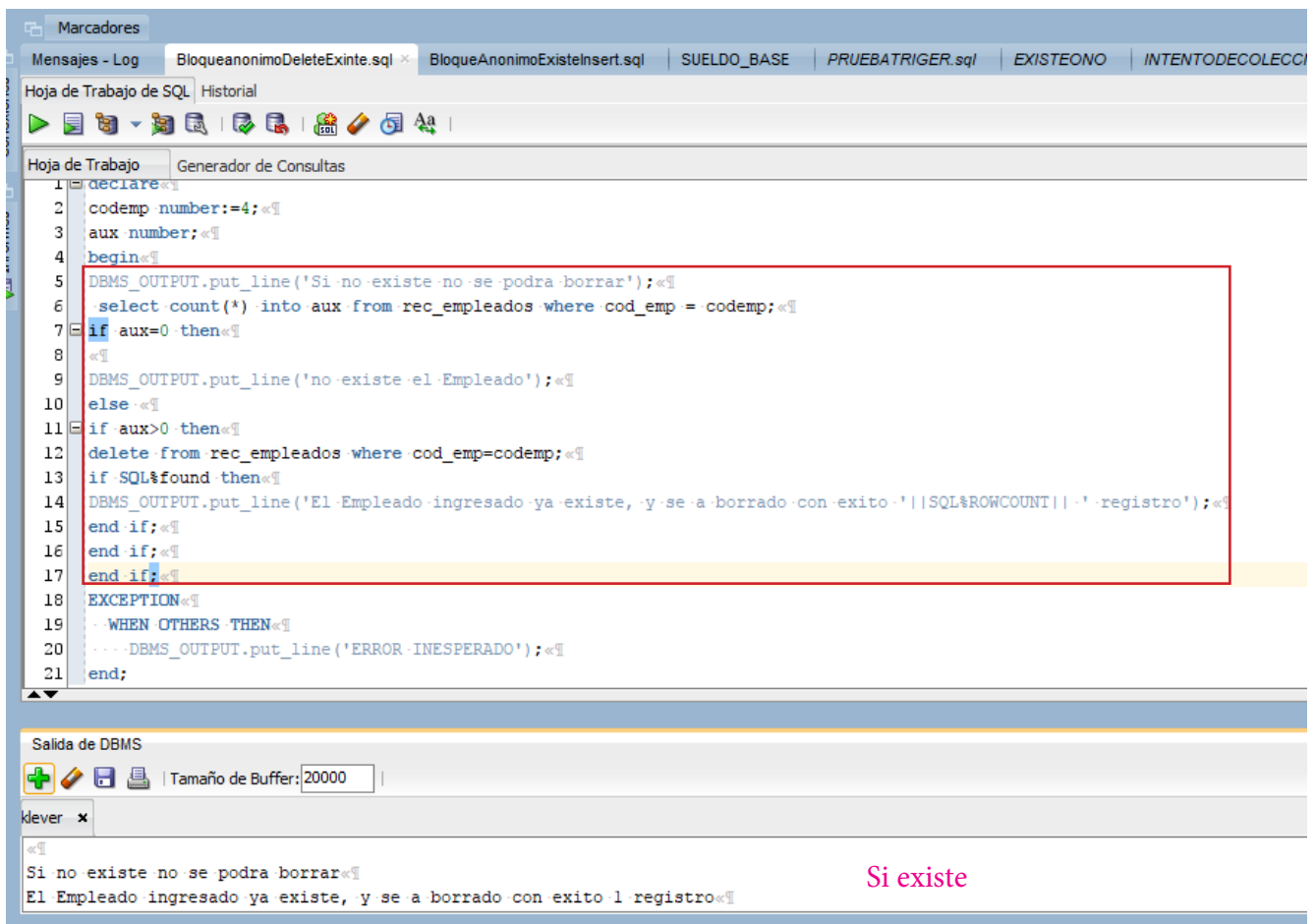
«  
Si no existe se insertara«  
El código ingresado ya existe, por lo tanto no puede insertarlo«  
«

Si existe

# BASE DE DATOS

## Ejercicio 2

- 2) Implementar bloque PL/SQL anónimo que dé de baja una fila comprobando previamente que existe. Se a utilizado un select que cuenta las fila que esta condicionada con la palabra clave, ademas de SQL%FOUND para comprobar si el delete se completo. Es parecido al anterior.



```
1 declare
2   codemp number:=4;
3   aux number;
4   begin
5     DBMS_OUTPUT.put_line('Si no existe no se podra borrar');
6     select count(*) into aux from rec_empleados where cod_emp = codemp;
7     if aux=0 then
8       DBMS_OUTPUT.put_line('no existe el Empleado');
9     else
10      if aux>0 then
11        delete from rec_empleados where cod_emp=codemp;
12        if SQL%found then
13          DBMS_OUTPUT.put_line('El Empleado ingresado ya existe, y se a borrado con exito'||SQL%ROWCOUNT||' registro');
14        end if;
15      end if;
16    end if;
17  end if;
18  EXCEPTION
19    WHEN OTHERS THEN
20      DBMS_OUTPUT.put_line('ERROR INESPERADO');
21  end;
```

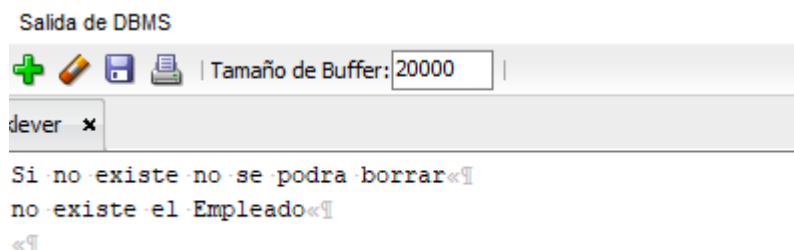
Salida de DBMS

Tamaño de Buffer: 20000

dever x

«  
Si no existe no se podra borrar«  
El Empleado ingresado ya existe, y se a borrado con exito 1 registro«

Si existe



```
Salida de DBMS

Tamaño de Buffer: 20000

dever x

«  
Si no existe no se podra borrar«  
no existe el Empleado«  
«
```

Si no existe

# BASE DE DATOS

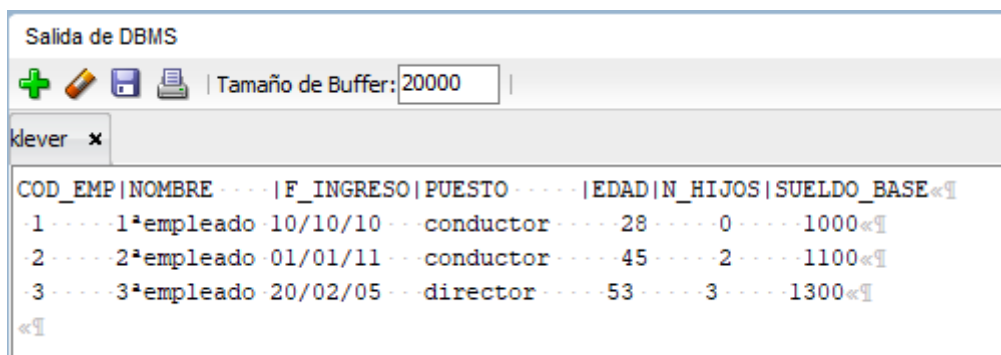
## Ejercicio3

- Leer todos los datos de los empleados y almacenarlos en memoria.
- Una vez leído todos los empleados recorrer las variables en memoria y mostrar por pantalla la información de cada empleado

Para poder trabajar con datos de gran variedad se a utilizado un cursor que es un tipo de matris para registros y que lo almacena en memoria para poder trabajar con los datos de la tabla o tablas seleccionadas.



```
1 DECLARE«¶
2 cursor c_codemp is«¶
3 select* from rec_empleados ;«¶
4 emp c_codemp%rowtype;«¶
5 e emp.cod_emp%type;«¶
6 tn rec_empleados.sueldo_base%type;«¶
7 cont rec_empleados.sueldo_base%type;«¶
8 error_mes exception;«¶
9 BEGIN«¶
10 DBMS_OUTPUT.put_line('COD_EMP|NOMBRE ----|F_INGRESO|PUESTO ----|EDAD|N_HIJOS|SUELDO_BASE');
11 open c_codemp;«¶
12 fetch c_codemp into emp;«¶
13 while c_codemp%FOUND loop«¶
14 e:=emp.cod_emp;«¶
15 DBMS_OUTPUT.put_line(' '||EMP.COD_EMP||' - - - - -'|EMP.NOMBRE||' - '|EMP.F_INGRESO||' - - -'|EMP
16 fetch c_codemp into emp;«¶
17 end loop;«¶
18 EXCEPTION«¶
19 WHEN OTHERS THEN«¶
20 DBMS_OUTPUT.put_line('ERROR INESPERADO');«¶
21 END;
```



```
Salida de DBMS
+ + + + + | Tamaño de Buffer: 20000 |
klever x
COD_EMP|NOMBRE ----|F_INGRESO|PUESTO ----|EDAD|N_HIJOS|SUELDO_BASE«¶
-1 - - - -1ªempleado-10/10/10--conductor--28--0--1000«¶
-2 - - - -2ªempleado-01/01/11--conductor--45--2--1100«¶
-3 - - - -3ªempleado-20/02/05--director--53--3--1300«¶
«¶
```

## Biografía

Los archivos de este trabajo se encuentran en este Repositorio.  
<https://github.com/klipsmf4/Base-de-datos>

Las Colecciones de PL/SQL  
<https://magicplsql.blogspot.com/2017/01/las-colecciones-de-plsql.html>