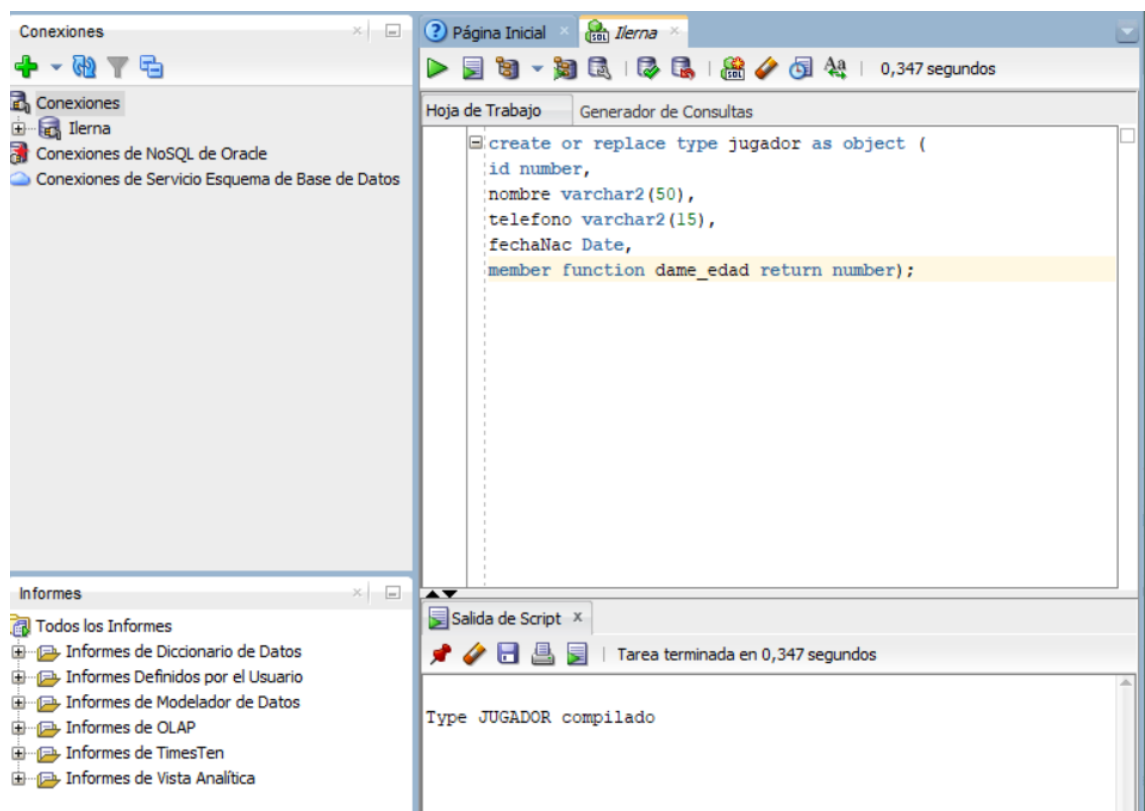


UF4 PAC02 – BASES DE DATOS OBJETO-RELACIONAL.

Ejercicios propuestos:

1. Cree un tipo de objeto jugador con los atributos id, nombre, teléfono y fecha de nacimiento y el método edad, que devuelve la edad y una referencia.

```
create or replace type jugador as object (  
  id number,  
  nombre varchar2 ( 80 ),  
  telefono varchar2 ( 10 ),  
  fnacimiento date,  
  member function get_edad return number );
```



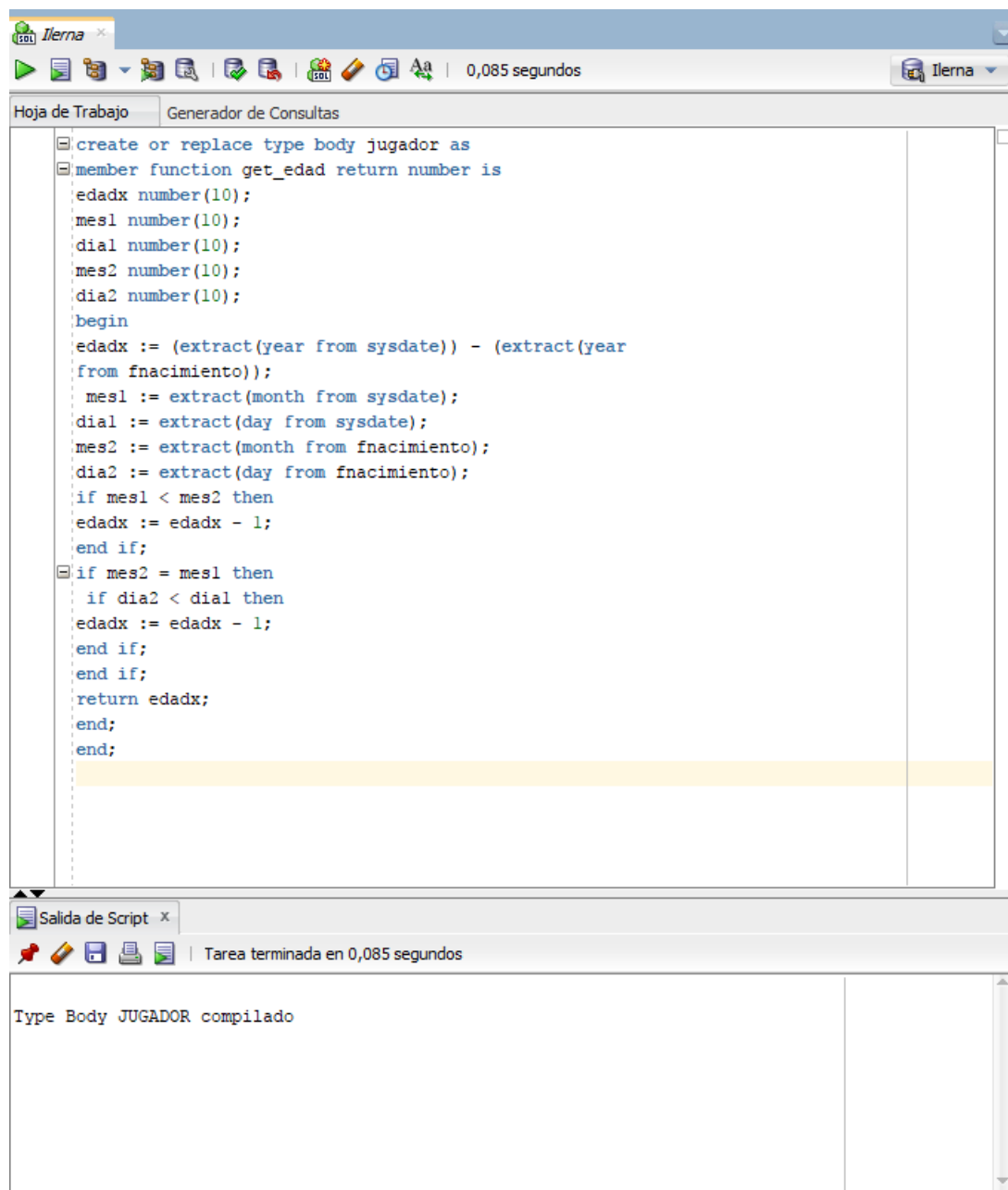
2. Defina el cuerpo de un método PL/SQL que calcule la edad para el objeto anterior.

```
create or replace type body jugador as member function get_edad return  
number is  
  edadx number ( 10 );  
  mes1 number ( 10 ); dia1 number ( 10 );  
  mes2 number ( 10 );  
  dia2 number ( 10 );  
begin  
  edadx := ( extract ( year from sysdate )) - ( extract ( year  
from fnacimiento ));
```

```

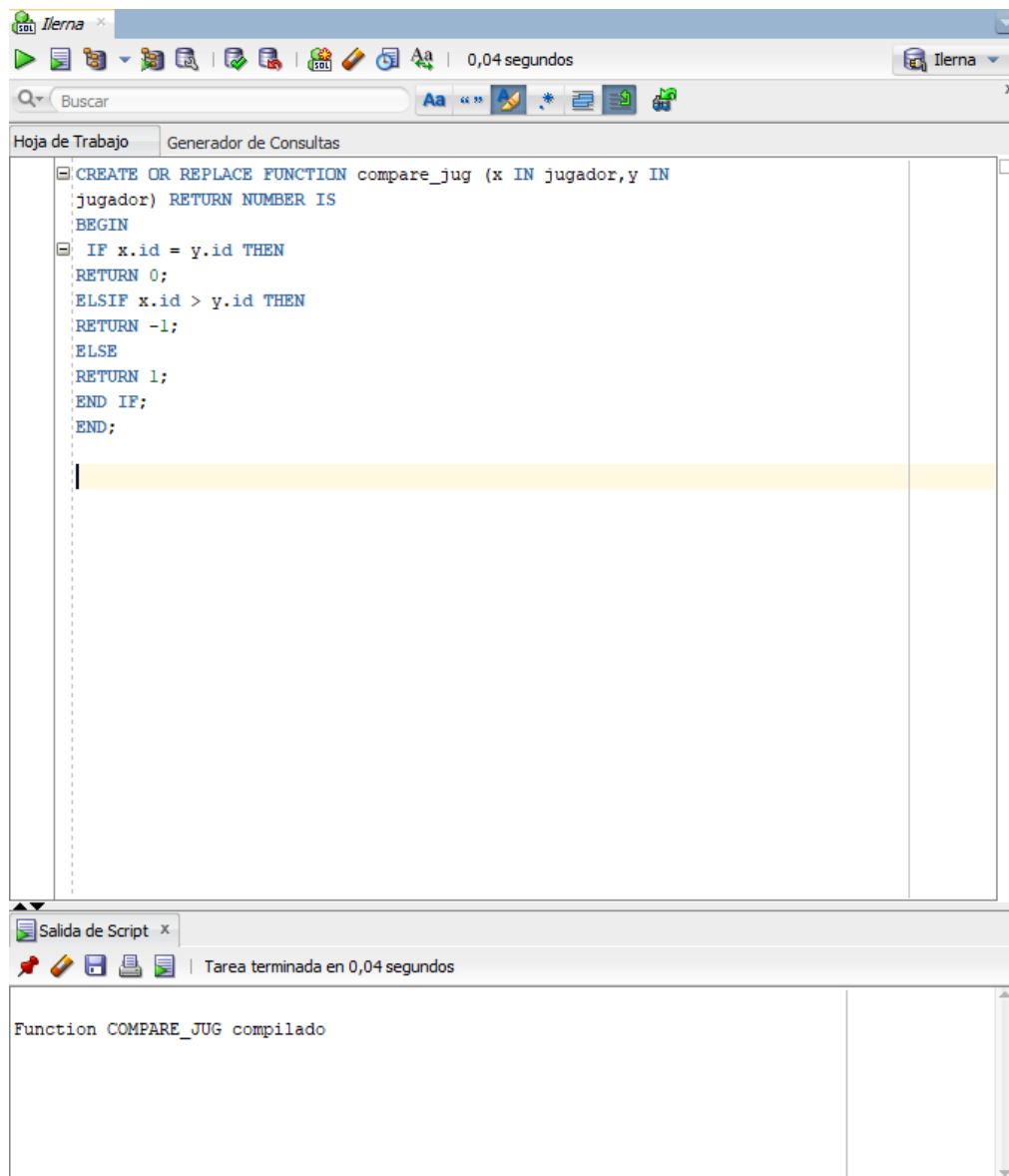
mes1 := extract ( month from sysdate );
dia1 := extract ( day from sysdate );
mes2 := extract ( month from fnacimiento ); dia2 := extract ( day from
fnacimiento );
if mes1 < mes2 then edadx := edadx - 1;
end if;
if mes2 = mes1 then
if dia2 < dia1 then edadx := edadx - 1;
end if;
end if;
return edadx ; end;
end;

```



3. Defina el cuerpo de un método PL/SQL que compare dos jugadores j1 y j2, devolviendo negativo si el id del jugador que se pasa por parámetro es mayor, 1 si es menor y 0 si son iguales.

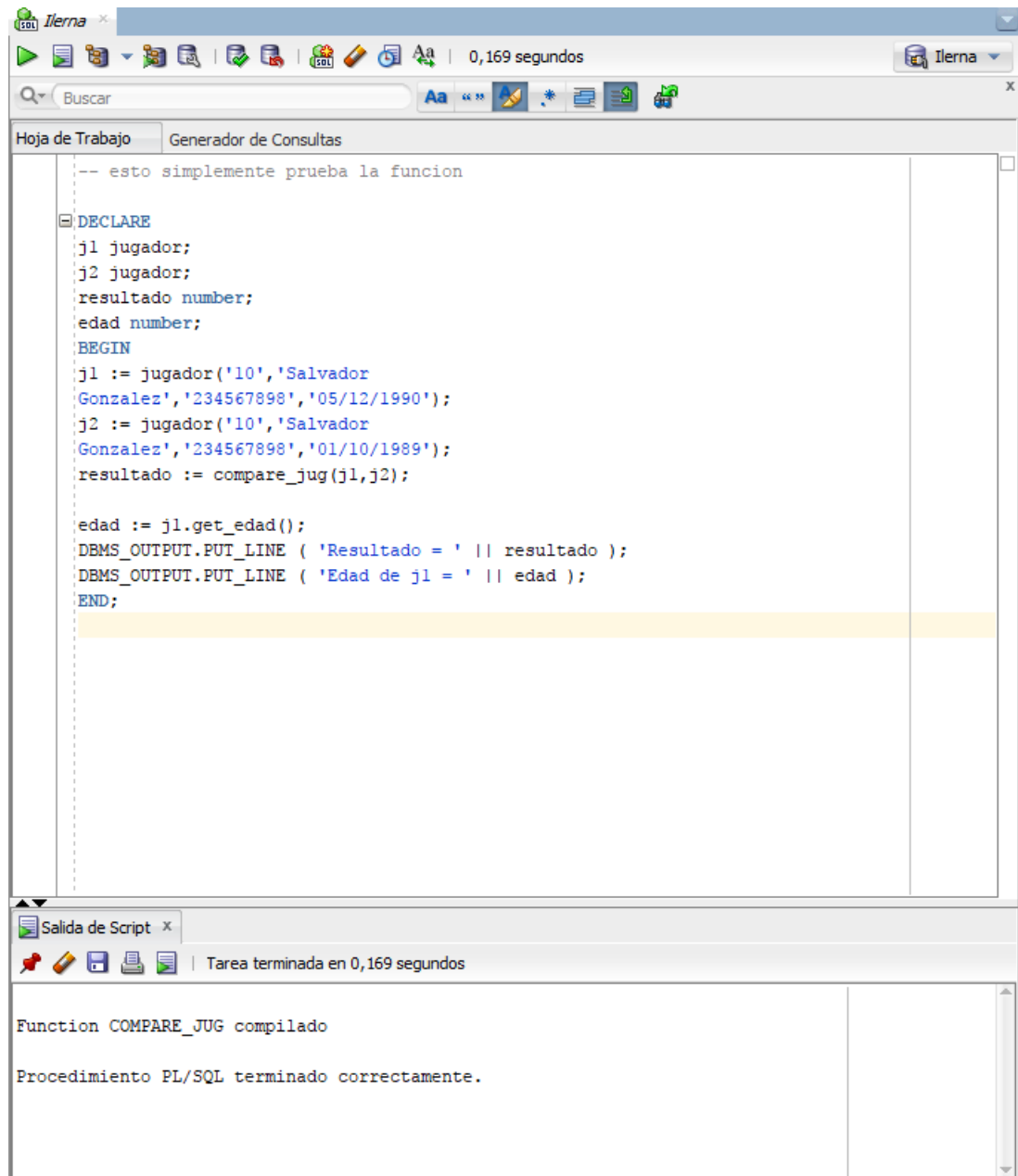
```
create or replace function compare_jug ( x in jugador , y in jugador ) return
number is
begin
if x.id = y.id then
return 0 ;
elsif x.id > y.id then
return -1; else
return 1; end if;
end;
```



```

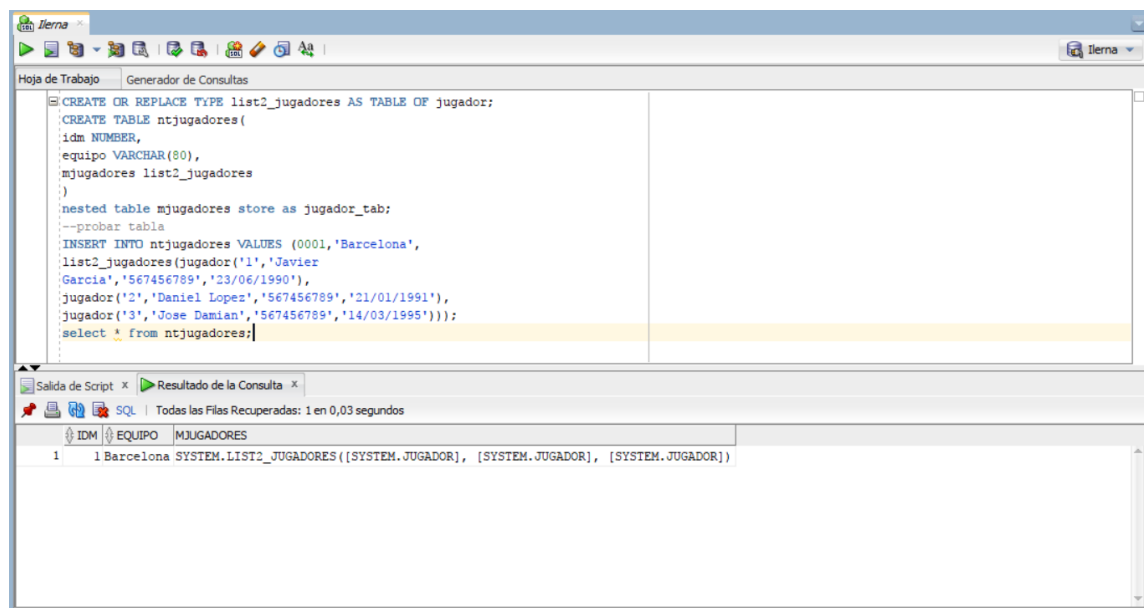
-- probamos la función
declare
j1 jugador;
j2 jugador; resultado number; edad number;
begin
j1 := jugador ( '10' , 'salvador gonzalez' , '234567898' , '05/12/1990' ); j2 :=
jugador ( '10' , 'salvador gonzalez' , '234567898' , '01/10/1989' );
resultado := compare_jug ( j1 , j2 );
edad := j1.get_edad ();
dbms_output.put_line ( 'resultado = ' || resultado );
dbms_output.put_line ( 'edad de j1 = ' || edad );
end;

```



4. Crear una tabla ntjugadores y almacenar en el campo jugadores los jugadores de cada uno. Además, almacenará un identificador y el nombre del equipo.

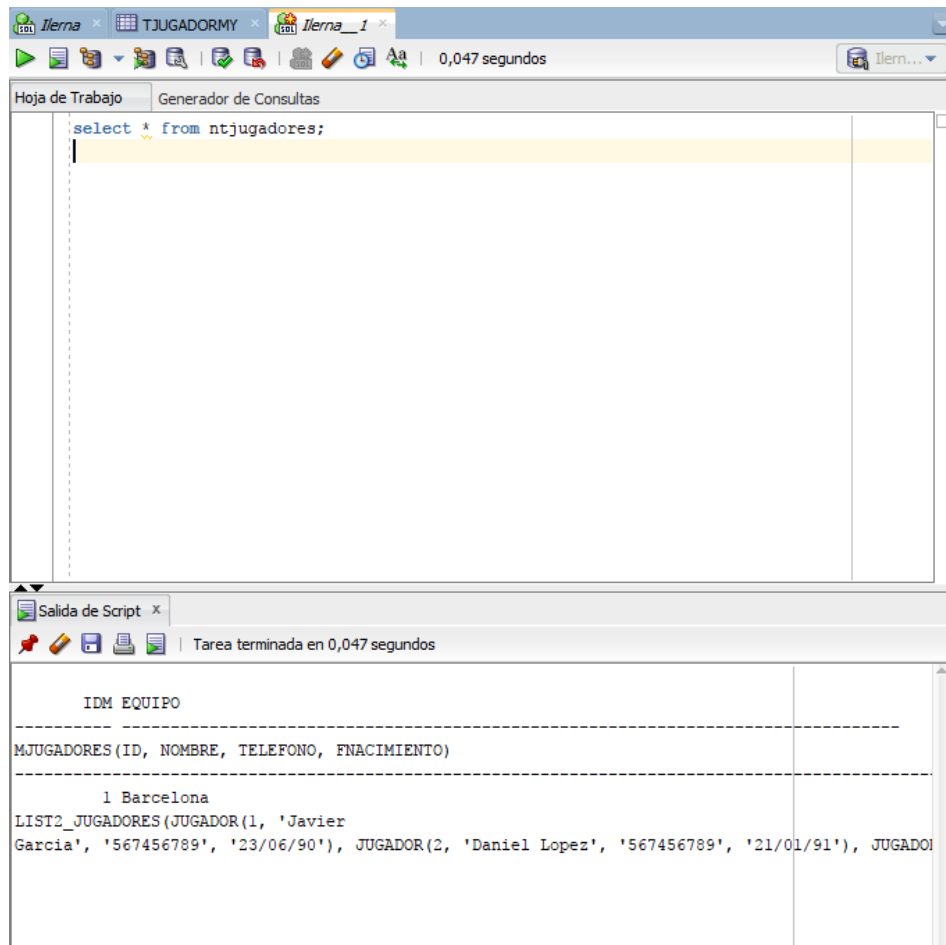
```
create or replace type list2_jugadores as table of jugador;  
create table ntjugadores( idm number,  
equipo varchar ( 80 ), mjugadores list2_jugadores )  
nested table mjugadores store as jugador_tab;
```



-- probamos la tabla insertando datos en ella

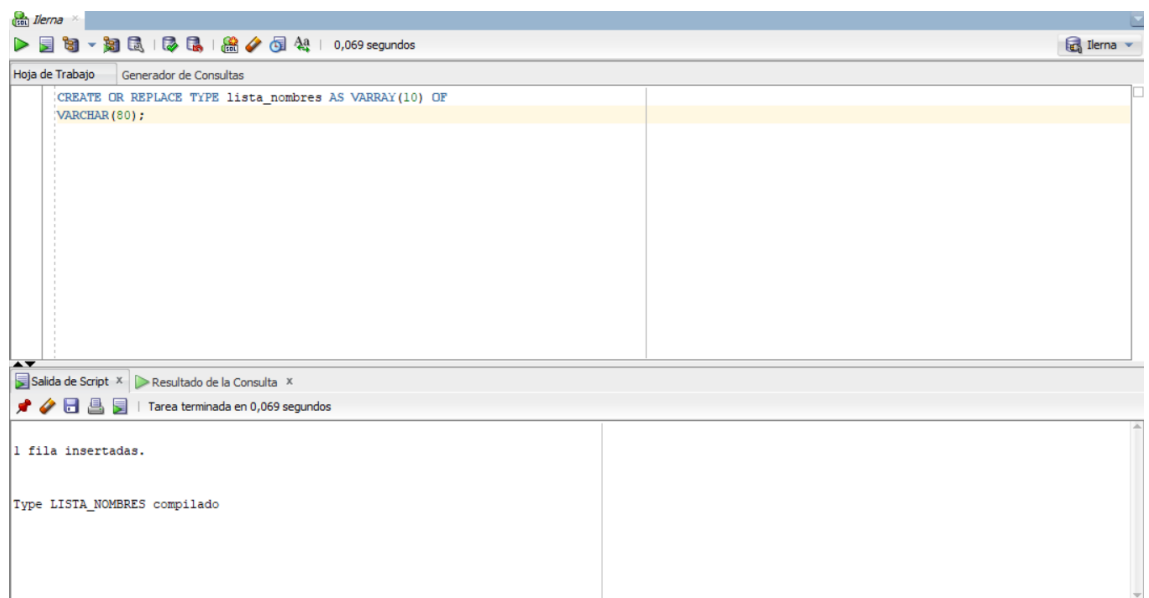
```
INSERT INTO ntjugadores VALUES ( 0001 , 'Barcelona', list2_jugadores ( jugador  
( '1', 'Javier Garcia' , '567456789' , '23/06/1990' ) , jugador ( '2', 'Daniel Lopez' ,  
'567456789' , '21/01/1991' ) , jugador ( '3' , 'Jose Damian' , '567456789' ,  
'14/03/1995' )))
```

```
select * from ntjugadores;
```



- Definir un tipo de datos para almacenar una lista ordenada de nombres de jugadores de equipo.

`create or replace type lista_nombres as varray (10) of varchar (80);`



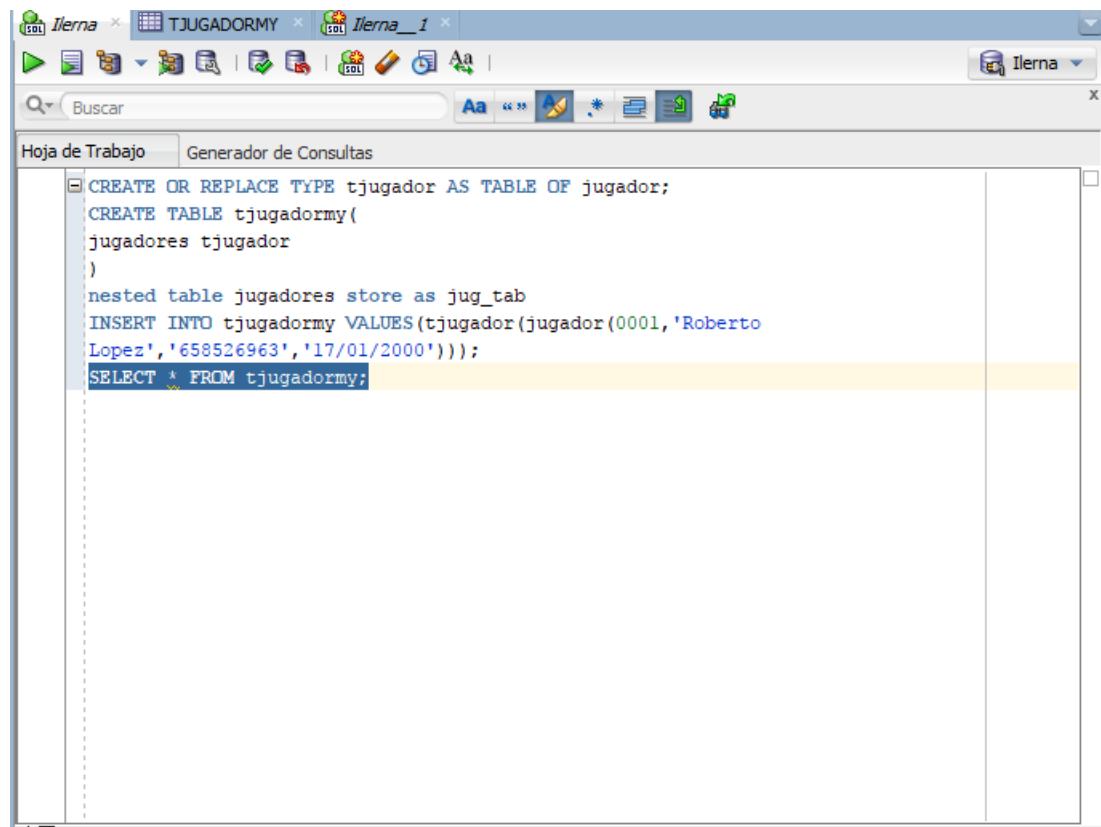
6. Cree una tabla de tipo tjugador para después insertar una fila mediante su constructor.

```
create or replace type tjugador as table of jugador;  
create table tjugadormy (jugadores tjugador)
```

```
nested table jugadores store as jug_tab
```

```
insert into tjugadormy values ( tjugador ( jugador ( '0001' , 'roberto lopez' ,  
'658526963' , '01/17/2000' )));
```

```
select * from tjugadormy;
```



Columnas	Datos	Model	Restricciones	Permisos	Estadísticas	Disparadores	Flashback	Dependencias	Detalles	Particiones	Índice
Acciones...											
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS						
1 JUGADORES	TJUGADOR	Yes	(null)	1	(null)						

0,26899999 segundos

Hoja de Trabajo Generador de Consultas

```
SELECT * FROM tjugadormy;
```

Salida de Script x

Tarea terminada en 0,269 segundos

```
JUGADORES (ID, NOMBRE, TELEFONO, FNACIMIENTO)

TJUGADOR (JUGADOR (1, 'Roberto
Lopez', '658526963', '17/01/00'))
```