

UF4_PAC02

Para la realización de estos ejercicios use la base de datos liga que está colgada en la plataforma.

Ejercicios

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
        --DBMS_OUTPUT.put_line('hola');
    end if;
end;
```

```

        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;
```

```

--probar funcion
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;

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

select * from ntjugadores;
```

5. 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;
```