

Asignatura	Bases de datos (DAW_BD)	Creado por	Ines Caro
Tarea	BD07. - Uso de bases de datos objeto-relacionales.	Fecha	28/03/2021
		Fecha de entrega	04/04/2021

Enunciado

Realiza los siguientes ejercicios. Antes de comenzar lee con detalle el enunciado de la tarea íntegramente y consulta con tu tutor o tutora las dudas que te puedan surgir.

Los ejercicios se basan en un Sistema de Información de una empresa de ventas que gestiona los comerciales y las zonas que tienen asignadas. Una zona puede estar asignada a varios comerciales. Cada zona tiene un único jefe que es el responsable de zona, pudiendo encargarse de varias zonas.

1. Crea el tipo de objetos "**Personal**" con los siguientes atributos:

```
codigo INTEGER,
dni VARCHAR2(10),
nombre VARCHAR2(30),
apellidos VARCHAR2(30),
sexo VARCHAR2(1),
fecha_nac DATE

SQL> CREATE OR REPLACE TYPE Personal AS OBJECT(
2  codigo INTEGER,
3  dni VARCHAR2(10),
4  nombre VARCHAR2(30),
5  apellidos VARCHAR2(30),
6  sexo VARCHAR2(1),
7  fecha_nac DATE
8 );
9 /
```

- Crea, como tipo heredado de "**Personal**", el tipo de objeto "**Responsable**" con los siguientes atributos:

```
tipo CHAR ,
antiguedad INTEGER

SQL> CREATE OR REPLACE TYPE Responsable UNDER Personal(
2  tipo CHAR,
3  antiguedad INTEGER
4 );
5 /
```

- Crea el tipo de objeto "**Zonas**" con los siguientes atributos:

```
codigo INTEGER,
nombre VARCHAR2(20),
refRespon REF Responsable,
codigoPostal CHAR(5),

SQL> CREATE OR REPLACE TYPE Zonas AS OBJECT(
2  codigo INTEGER,
3  nombre VARCHAR2(20),
4  refRespon REF Responsable,
5  codigoPostal CHAR(5)
6 );
7 /
```

- Crea, como tipo heredado de "**Personal**", el tipo de objeto "**Comercial**" con los siguientes atributos:

```
zonaComercial Zonas

SQL> CREATE OR REPLACE TYPE Comercial UNDER Personal(
2  zonaComercial Zonas
3 );
4 /
```

2. Crea un método constructor para el tipo de objetos "Responsable", en el que se indiquen como parámetros el código, nombre, primer apellido, segundo apellido y tipo. Este método debe asignar al atributo apellidos los datos de primer apellido y segundo apellido que se han pasado como parámetros, uniéndolos con un espacio entre ellos.

```
SQL> CREATE OR REPLACE TYPE Responsable UNDER Personal(  
2  CONSTRUCTOR FUNCTION Responsable(codigo INTEGER, nombre VARCHAR2, apellido1 VARCHAR2, apellido2 VARCHAR2, tipo CHAR)  
   RETURN SELF AS RESULT  
3 );  
4 /
```

```
SQL> CREATE OR REPLACE TYPE BODY Responsable AS  
2  CONSTRUCTOR FUNCTION Responsable (codigo INTEGER, nombre VARCHAR2, apellido1 VARCHAR2, apellido2 VARCHAR2, tipo CHAR)  
   RETURN SELF AS RESULT IS  
3  BEGIN  
4  SELF.codigo := codigo;  
5  SELF.nombre := nombre;  
6  SELF.apellidos := apellido1 || ' ' || apellido2;  
7  SELF.tipo := tipo;  
8  RETURN;  
9  END;  
10 END;  
11 /
```

3. Crea un método getNombreCompleto para el tipo de objetos Responsable que permita obtener su nombre completo con el formato apellidos nombre

```
SQL> CREATE OR REPLACE TYPE BODY Responsable AS  
2  MEMBER FUNCTION getNombreCompleto RETURN VARCHAR2 IS  
3  nombreCompleto VARCHAR2(100);  
4  BEGIN  
5  nombreCompleto := SELF.nombre || ' ' || SELF.apellidos;  
6  RETURN nombreCompleto;  
7  END;  
8  /
```

4. Crea una tabla TablaResponsables de objetos Responsable. Inserta en dicha tabla dos objetos Responsable.

```
codigo: 5  
dni: 51083099F  
nombre: ELENA  
apellidos: POSTA LLANOS  
sexo: F  
fecha_nac: 31/03/1975  
tipo: N  
antiguedad: 4
```

El segundo objeto "Responsable" debes crearlo usando el método constructor que has realizado anteriormente. Debes usar los siguientes datos:

```
codigo: 6  
nombre: JAVIER  
apellidos: JARAMILLO HERNANDEZ  
tipo: C
```

```
SQL> CREATE TABLE TablaResponsables OF Responsable  
2  INSERT INTO TablaResponsables VALUES(5,'51083099F','ELENA','POSTA LLANOS','F','31-03-1975','N',4)  
3  INSERT INTO TablaResponsables VALUES(6,'JAVIER','JARAMILLO HERNANDEZ','C')  
4  END;
```

5. Crea una colección **VARRAY** llamada **ListaZonas** en la que se puedan almacenar hasta 10 objetos **Zonas**.
Guarda en una instancia **listaZonas1** de dicha lista, dos **Zonas**.

```
codigo: 1
nombre: zona 1
refResponsable: Referencia al responsable cuyo codigo es 5
codigo postal: 06834
codigo: 2
nombre: zona 2
refResponsable: Referencia al responsable cuyo DNI es 51083099F.
codigo postal: 28003
```

```
SQL> CREATE OR REPLACE TYPE ListaZonas as VARRAY(10) OF Zonas
2  DECLARE
3  listaZonas1 ListaZonas;
4  Zona1 Zonas;
5  Zona2 Zonas;
6  RefResp REF Responsable;
7  SELECT REF(r) INTO RefResp FROM ListaResponsables r WHERE r.codigo=5;
8  Zona1 := new Zonas(1,'ZONA 1',RefResp,'06834');
9  SELECT REF(r) INTO RefResp FROM ListaResponsables r WHERE r.dni='51083099F';
10 Zona2 := new Zonas(2,'ZONA 2',RefResp,'28003');
11 listaZonas1 := ListaZonas(Zona1,Zona2);
12 END;
13 /
```

6. Crea una tabla **TablaComerciales** de objetos **Comercial**. Inserta en dicha tabla las siguientes filas:

```
codigo: 100
dni: 23401092Z
nombre: MARCOS
apellidos: SUAREZ LOPEZ
sexo: M
fecha_nac: 30/3/1990
zonacomercial: objeto creado anteriormente para la zona 1

codigo: 102
dni: 6932288V
nombre: ANASTASIA
apellidos: GOMES PEREZ
sexo: F
fecha_nac: 28/11/1984
zonacomercial: objeto que se encuentre en la segunda posición de "listaZonas1"
                (debe tomarse de la lista)
```

```
SQL> CREATE TABLE TablaComerciales OF Comercial
2  INSERT INTO TablaComerciales VALUES(100,'23401092Z','MARCOS','SUAREZ LOPEZ','M','30-03-1990',listaZonas1(1))
3  INSERT INTO TablaComerciales VALUES(102,'6932288V','ANASTASIA','GOMES PEREZ','F','28-11-1984',listaZonas1(2))
4  END;
```

7. Obtener, de la tabla **TablaComerciales**, el **Comercial** que tiene el código 100, asignándoselo a una variable **unComercial**.

```
SQL> DECLARE
2  unComercial Comercial;
3  BEGIN
4  SELECT VALUE(c) INTO unComercial FROM TablaComerciales c WHERE c.codigo=100;
5  END;
6  /
```

8. Modifica el código del Comercial guardado en esa variable `unComercial` asignando el valor 101, y su `zona` debe ser la segunda que se había creado anteriormente. Inserta ese Comercial en la tabla `TablaComerciales`

```
SQL> CREATE OR REPLACE TYPE BODY ListaZonas AS
2  DECLARE
3  unComercial Comercial;
4  BEGIN
5  unComercial.codigo=101;
6  unComercial.zonaComercial := ListaZonas1(2);
7  INSERT INTO TablaResponsables VALUES(unComercial);
8  END;
9  /
```

9. Crea un método `MAP ordenarZonas` para el tipo `Zonas`. Este método debe retornar el nombre completo del `Responsable` al que hace referencia cada `zona`. Para obtener el nombre debes utilizar el método `getNombreCompleto` que se ha creado anteriormente

```
SQL> CREATE OR REPLACE TYPE BODY Zonas AS
2  MAP MEMBER FUNCTION ordenarZonas RETURN VARCHAR2 IS
3  respo Responsable;
4  BEGIN
5  SELECT Deref(RefResp) INTO respo FROM DUAL;
6  RETURN respo.getNombreCompleto();
7  END;
8  END;
9  /
```

10. Realiza una consulta de la tabla `TablaComerciales` ordenada por `zonaComercial` para comprobar el funcionamiento del método `MAP`.

```
SELECT * FROM TablaResponsables ORDER BY zonaComercial;
```