

# *BASES DE DATOS*

## *TAREA 7*



*Diego Manuel Carrasco Castañares*

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
```

CREATE OR REPLACE TYPE Personal AS OBJECT(

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

)NOT FINAL;

/

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

```
tipo CHAR ,  
antiguedad INTEGER
```

CREATE OR REPLACE TYPE Responsable UNDER Personal (

```
tipo CHAR,  
antiguedad INTEGER,
```

CONSTRUCTOR FUNCTION Responsable (

```
codigo INTEGER,  
nombre VARCHAR2,  
primer_apellido VARCHAR2,  
segundo_apellido VARCHAR2,  
tipo CHAR)
```

RETURN SELF AS RESULT,

MEMBER FUNCTION getNombreCompleto RETURN VARCHAR2

)FINAL

INSTANTIABLE;

/

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

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

```
CREATE OR REPLACE TYPE Zonas AS OBJECT(
```

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

```
MAP MEMBER FUNCTION ordenarZonas RETURN VARCHAR2
```

```
)NOT FINAL;
/
```

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

```
zonaComercial Zonas
```

```
CREATE OR REPLACE TYPE Comercial UNDER Personal(
```

```
zonaComercial Zonas
```

```
);
/
```

**2. Crea un método constructor para el tipo de objetos "Responsable", en el que se indiquen como parámetros 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.**

```
CREATE OR REPLACE TYPE BODY Responsable AS
```

```
CONSTRUCTOR FUNCTION Responsable (
```

```
codigo INTEGER,
nombre VARCHAR2,
primer_apellido VARCHAR2,
segundo_apellido VARCHAR2,
tipo CHAR)
```

```
RETURN SELF AS RESULT IS
```

```
BEGIN
```

```
SELF.codigo := codigo;
SELF.nombre := nombre;
SELF.apellidos := primer_apellido || ' ' || segundo_apellido;
SELF.tipo := tipo;
```

```
RETURN;
```

```
END;
```

```
MEMBER FUNCTION getNombreCompleto RETURN VARCHAR2 IS

    BEGIN

        RETURN SELF.apellidos||' '||SELF.nombre;

    END getNombreCompleto;

END;
/
```

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

```
CREATE OR REPLACE TYPE BODY Responsable AS

    CONSTRUCTOR FUNCTION Responsable (

        codigo INTEGER,
        nombre VARCHAR2,
        primer_apellido VARCHAR2,
        segundo_apellido VARCHAR2,
        tipo CHAR)

    RETURN SELF AS RESULT IS

    BEGIN

        SELF.codigo := codigo;
        SELF.nombre := nombre;
        SELF.apellidos := primer_apellido||' '||segundo_apellido;
        SELF.tipo := tipo;

    RETURN;

    END;

    MEMBER FUNCTION getNombreCompleto RETURN VARCHAR2 IS

        BEGIN

            RETURN SELF.apellidos||' '||SELF.nombre;

        END getNombreCompleto;

    END;
/
```

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

codigo: 5 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

CREATE TABLE TablaResponsables OF Responsable;

INSERT INTO TablaResponsables VALUES (5, (NULL), 'ELENA', 'POSTA LLANOS', 'F', '31/03/1975', 'N', 4);

INSERT INTO TablaResponsables VALUES (6, '51083099F', 'JAVIER', 'JARAMILLO HERNANDEZ', (NULL), (NULL), 'C', (NULL));  
/

**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

DECLARE TYPE ListaZonas IS VARRAY (10) OF Zonas;

ListaZonas1 ListaZonas := ListaZonas (NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);  
refRespon REF Responsable;

BEGIN

SELECT REF (r) INTO refRespon FROM tablaResponsables r WHERE r.codigo = 5;  
ListaZonas1 (1) := zonas (1, 'zona1', refRespon, 06834);  
SELECT REF (r) INTO refRespon FROM tablaResponsables r WHERE r.dni = '51083099F';  
ListaZonas1 (2) := zonas (2, 'zona2', refRespon, 28003);

END;

/

**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)

CREATE TABLE TablaComerciales OF Comercial;

INSERT INTO TablaComerciales VALUES (comercial(100, '23401092Z', 'Marcos', 'Suarez Lopez', 'M', '20/3/1990', (NULL)));

INSERT INTO TablaComerciales VALUES (comercial(102, '6932288V', 'Anastasia', 'Gomes Perez', 'F', '28/11/1984', (NULL)));  
/

*En este apartado no he conseguido instanciar la columna zonacomercial. Por favor Chema, déjame en la retroalimentación de la tarea una posible solución a este apartado o una explicación de como hacerlo. Para poder seguir con la tarea lo he instanciado como NULL.*

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

DECLARE

unComercial Comercial;

BEGIN

SELECT VALUE (c) INTO unComercial FROM TablaComerciales c WHERE c.codigo = 100;

END;

**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**

*Este apartado tampoco consigo sacarlo. Como en los anteriores, por favor,*

**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**

```
CREATE OR REPLACE TYPE BODY zonas AS
```

```
MAP MEMBER FUNCTION ordenarZonas RETURN VARCHAR2  
IS
```

```
unResponsable Responsable;
```

```
BEGIN
```

```
SELECT Deref (refRespon) INTO unResponsable FROM Dual;
```

```
RETURN (unResponsable.getNombreCompleto());
```

```
END ordenarZonas;
```

```
END;
```

```
/
```

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

```
SELECT * FROM tablaResponsables ORDER BY ordenarZonas;
```

```
/
```