

```

/* 1. Crea el tipo de objetos Personal con sus atributos */
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;

```

```

/* 2. Crea, como tipo heredado de Personal, el tipo de objeto Responsable con sus atributos
y que contenga:
- Método constructor 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.
- Método getNombreCompleto que permita obtener su nombre completo con el formato apellidos nombre
*/

CREATE OR REPLACE TYPE Responsable UNDER Personal (
    tipo CHAR,
    antigüedad INTEGER,

    CONSTRUCTOR FUNCTION Responsable(
        codigo INTEGER,
        nombre VARCHAR2,
        apellido1 VARCHAR2,
        apellido2 VARCHAR2,
        tipo CHAR
    ) RETURN SELF AS RESULT,

    MEMBER FUNCTION getNombreCompleto RETURN VARCHAR2
);

CREATE OR REPLACE TYPE BODY Responsable AS
    CONSTRUCTOR FUNCTION Responsable(
        codigo INTEGER,
        nombre VARCHAR2,
        apellido1 VARCHAR2,
        apellido2 VARCHAR2,
        tipo CHAR
    ) RETURN SELF AS RESULT IS
    BEGIN
        SELF.codigo := codigo;
        SELF.nombre := nombre;
        SELF.apellidos := CONCAT(apellido1, apellido2);
        SELF.tipo := tipo;
    END;

```

```

        antigüedad := 1;
        RETURN;
    END;

    MEMBER FUNCTION getNombreCompleto RETURN VARCHAR2 IS
    BEGIN
        RETURN (apellidos || ' ' || nombre);
    END getNombreCompleto;
END;

```

/* 4. Crea el tipo de objeto Zonas con sus atributos y que contenga:
 - Método MAP ordenarZonas que devuelva 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 Zonas AS OBJECT (
    codigo INTEGER,
    nombre VARCHAR2(20),
    refResponsable REF Responsable,
    codigoPostal CHAR(5),

    MAP MEMBER FUNCTION ordenarZonas RETURN VARCHAR2
);

CREATE OR REPLACE TYPE BODY Zonas AS
    MAP MEMBER FUNCTION ordenarZonas RETURN VARCHAR2 IS
        unResponsable Responsable;
    BEGIN
        SELECT Deref(refResponsable) INTO unResponsable FROM Dual;
        RETURN (unResponsable.getNombreCompleto());
    END ordenarZonas;
END;

```

/* 5. Crea una colección VARRAY llamada ListaZonas en la que se puedan almacenar hasta 10 objetos Zonas */
 CREATE OR REPLACE TYPE ListaZonas IS VARRAY(10) OF Zonas;

/* 6. Crea, como tipo heredado de Personal, el tipo de objeto Comercial con sus atributos. */
 CREATE OR REPLACE TYPE Comercial UNDER Personal (
 zonaComercial Zonas
);

/* 7. Crea una tabla TablaComerciales de objetos Comercial */
 CREATE TABLE TablaComerciales OF Comercial;

```

/* 8. Crea un bloque de código que realice las siguientes acciones:
- Guarda en una instancia listaZonas1 de dicha lista, dos Zonas ...

- Inserta en la tabla TablaComerciales las siguientes filas ...

- Obtener, de la tabla TablaComerciales, el Comercial que tiene el código 100,
asignándolo a una variable unComercial

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

DECLARE
    zona1 Zonas;
    zona2 Zonas;
    listaZonas1 ListaZonas;
    unComercial Comercial;
    refUnResponsable REF Responsable;
BEGIN
    SELECT REF(tp) INTO RefUnResponsable FROM TablaResponsables tp WHERE codigo = 5;
    zona1 := Zonas(1, 'zona 1', RefUnResponsable, '06034');

    SELECT REF(tp) INTO RefUnResponsable FROM TablaResponsables tp WHERE dni = '51083099F';
    zona2 := Zonas(2, 'zona 2', RefUnResponsable, '28003');

    listaZonas1 := ListaZonas(zona1, zona2);

    INSERT INTO TablaComerciales VALUES (100, '23401092Z', 'MARCOS', 'SUAREZ LOPEZ', 'M', '30/3/1990', zona1);
    INSERT INTO TablaComerciales VALUES (102, '6932288V', 'ANASTASIA', 'GOMES PEREZ', 'F', '28/11/1984', listaZonas1(2));

    SELECT VALUE(a) INTO unComercial FROM TablaComerciales a WHERE codigo = 100;
    unComercial.codigo := 101;
    unComercial.zonaComercial := zona2;

    INSERT INTO TablaComerciales VALUES (unComercial);
END;

/* 9. Realiza una consulta de la tabla TablaComerciales ordenada por zonaComercial para
comprobar el funcionamiento del método MAP. */
SELECT * FROM TablaComerciales ORDER BY ZonaComercial;

```