Programación de Base de Datos

Documentación de práctica 3

Trabajo realizado por:

Carlos Guillén Moreno

Para el profesor: Andrés Caro Lindo (correo: andresc@unex.es)

Contenido

INTRODUCCIÓN.

Para esta práctica, se usan las herramientas instaladas en la sesión anterior. Se utilizará el gestor de base de datos (Oracle 11g Express Edition) y el entorno (SQL Developer).

Ejercicios

Apartado A

Implementar una **nueva clase**, llamada `Trabajador`, que herede de la clase base `Persona`, y que incluya los atributos básicos de prácticas anteriores. Implementar también una **nueva tabla** `Empresa` que incluya instancias de la clase `Trabajador`. Debe seguirse el siguiente modelo:

```
CREATE TYPE Trabajador UNDER Persona (
  ingresos NUMBER(7,2),
  gastosFijos NUMBER(7,2),
  gastosAlim NUMBER(7,2),
  gastosRopa NUMBER(7,2),
  OVERRIDING MEMBER PROCEDURE mostrar,
  --APARTADO E:
  --en función de los años trabajando se le hará más o menos descuento
  MEMBER FUNCTION calcularDescuento (años IN NUMBER) RETURN NUMBER
);
/
```

Se crea un nuevo tipo llamado Trabajador que extiende de persona. Que extiende se refiere a que hereda del tipo Persona todos sus atributos y métodos. En este nuevo tipo añadimos los atributos:

- Ingresos: es de tipo numérico y se refiere a los ingresos que tiene cada instancia de trabajador,
- Gastos Fijos: es de tipo numérico y se trata de los gastos fijos que tiene cada instancia de trabajador.
- Gastos Alimenticios: es de tipo numérico y se trata de los gastos en alimentos que tiene cada instancia de trabajador.
- Gastos Ropa: es de tipo numérico y se trata de los gastos en ropa que tiene cada instancia de trabajador.

También se sobrescribe el método mostrar para mostrar los nuevos atributos junto con los que se hereda de Persona.

```
OVERRIDING MEMBER PROCEDURE mostrar
```

Quedaría así la nueva clase Trabajador:

```
CREATE OR REPLACE TYPE BODY Trabajador AS

  OVERRIDING MEMBER PROCEDURE mostrar IS
    vdni VARCHAR2(9);
    vpais VARCHAR2(2);
    vcontrol VARCHAR2(2);
    ventidad VARCHAR2(4);
    voficina VARCHAR2(4);
    vdc VARCHAR2(2);
    vcuenta VARCHAR2(10);

    CURSOR cursor_ver_cuentas IS
       SELECT p.DNI, c.pais, c.control, c.entidad, c.oficina, c.dc,
c.cuenta
       FROM Empresa p, table(p.cuentas) c;

  BEGIN
    dbms_output.put_line('Método mostrar() de objeto Estudiante');
    dbms_output.put_line('DNI: ' || dni);
    dbms_output.put_line('Nombre: ' || datosPersonales.nombre);
    dbms_output.put_line('Apellido 1: ' || datosPersonales.ape1);
    dbms_output.put_line('Apellido 2: ' || datosPersonales.ape2);
    dbms_output.put_line('Fecha nacimiento: ' || fechanac);
    dbms_output.put_line('Dirección: ' || direccion);
    dbms_output.put_line('CP: ' || cp);
    dbms_output.put_line('Sector: ' || sector);

    dbms_output.put_line('Existen ' || telefonos.count || '
teléfonos');
    IF (telefonos.count <> 0) THEN
      FOR i IN 1 .. telefonos.count LOOP
         dbms_output.put_line('     Teléfono ' || i || ': ' ||
telefonos(i));
      END LOOP;
    END IF;

    dbms_output.put_line('Existen ' || cuentas.count || ' cuentas
bancarias');
    OPEN cursor_ver_cuentas;
    LOOP
      FETCH cursor_ver_cuentas INTO vDNI, vpais, vcontrol, ventidad,
voficina, vdc, vcuenta;
      EXIT WHEN cursor_ver_cuentas%NOTFOUND;
      IF (vDNI = dni) THEN
        dbms_output.put_line('     IBAN: ' || vpais || vcontrol || '-'
|| ventidad || '-' || voficina || '-' || vdc || '-' || vcuenta);
      END IF;
    END LOOP;
    CLOSE cursor_ver_cuentas;

    dbms_output.put_line('Ingresos: ' || ingresos);
    dbms_output.put_line('Gastos Fijos: ' || gastosFijos);
    dbms_output.put_line('Gastos Alimentacion: ' || gastosAlim);
    dbms_output.put_line('Gastos Ropas: ' || gastosRopa);


    dbms_output.put_line('---------------------------------------');
  END; -- de Mostrar
```

```
END; -- de Trabajador
```

Como se puede observar se muestran los atributos de Persona primero y en amarillo los nuevos atributos con sus valores. De Persona se muestran los atributos dni, nombre, apellido1, apellido2, fechaNac, dirección, cp, sector, teléfonos y cuentas. De trabajador, que son los nuevos que se muestran, ingresos, gastosFijos, gastosAlim y gastosRopa.

Ahora se crea la tabla Empresa y quedaría así:

```
CREATE TABLE Empresa OF Trabajador
  (dni PRIMARY KEY)
  NESTED TABLE cuentas STORE AS cuentas_emp
  ((PRIMARY KEY(NESTED_TABLE_ID,cod)));
/
```

En esta tabla se insertarán las tuplas de tipo Trabajador. La clave primaria de la tabla será el dni de los trabajadores insertados. También se almacenarán las cuentas de los trabajadores con la declaración: `NESTED TABLE cuentas STORE AS cuentas_emp`

Realizar **inserciones** de 3 trabajadores, considerando también la inserción de referencias a los mismos en la tabla `Referencias`, tal y como se ha hecho con todas las instancias consideradas en el código fuente de ejemplo.

Inserciones de los trabajadores:

```
INSERT INTO Empresa
VALUES ('111000333', NULL,
nombreCompleto('Carlos','Guillen','Moreno'),'01/07/1989','C.
Constitución','06800','H',3,numTelfs(),cuentasTab(
  IBANTipo('047','ES','79','2038','0010','73','8365936594'), -- BBVA
  IBANTipo('053','ES','82','2048','0200','32','8465836593')  --
Liberbank
), 2000, 500, 600, 200);

INSERT INTO Empresa
VALUES ('111000888', NULL,
nombreCompleto('Lionel','Messi','Estaca'),'01/07/1989','C.
Constitución','06800','H',3,numTelfs(),cuentasTab(
  IBANTipo('047','ES','79','2038','0010','73','8365936595'), -- BBVA
  IBANTipo('053','ES','82','2048','0200','32','8465836594')  --
Liberbank
), 3000, 600, 700, 300);

INSERT INTO Empresa
VALUES ('111000999', NULL,
nombreCompleto('Cristiano','Ronaldo','Mazapan'),'01/07/1989','C.
Constitución','06800','H',3,numTelfs(),cuentasTab(
  IBANTipo('047','ES','79','2038','0010','73','8365936596'), -- BBVA
  IBANTipo('053','ES','82','2048','0200','32','8465836595')  --
Liberbank
), 4000, 700, 800, 700);
```

Inserciones de las referencias:

```
/*Referencias de las nuevas inserciones*/
INSERT INTO Referencias
  SELECT c.dni,REF(c)
  FROM Empresa c
  WHERE dni='111000333';

INSERT INTO Referencias
  SELECT c.dni,REF(c)
  FROM Empresa c
  WHERE dni='111000888';

INSERT INTO Referencias
  SELECT c.dni,REF(c)
  FROM Empresa c
  WHERE dni='111000999';
```

Se insertan las referencias usando el dni de los trabajadores de la tabla Empresa.

Probar a realizar **consultas** sobre los objetos creados e insertados, del estilo de las que se han visto en el tema de teoría. Incluir estas consultas en la documentación que se entregue de esta práctica.

```
  SELECT VALUE(c) INTO persona_c
  FROM Empresa c
  where c.DNI='111000888';
  -- Acceso a atributos
  dbms_output.put_line(persona_c.dni);
  -- Llamada a métodos
  persona_c.mostrar();
```

persona_c es un trabajador auxiliar que utilizamos para almacenar el valor de la instancia con valor en su dni de 111000888. Después mostramos la información con el método sobrescrito mostrar.

Apartado B

Establecer **relaciones** (REF) entre objetos, gestionando el atributo `relacionado` de la clase base `Persona`. Este atributo relacionará a las siguientes personas:

Se crea un atributo en la clase Persona llamado relacionado que sirve para relacionar a personas entre sí. Se adjunta subrayado:

```
CREATE TYPE Persona AS OBJECT (
  dni VARCHAR(9),
  relacionado REF Persona,
  datosPersonales nombreCompleto,
  fechanac DATE,
  direccion VARCHAR2(20),
  cp VARCHAR2(5),
  sexo VARCHAR2(1),
  sector NUMBER(2),
  telefonos numTelfs,
  cuentas cuentasTab,
  --MAP MEMBER FUNCTION ret_value RETURN NUMBER,
  --APARTADO D:
  ORDER MEMBER FUNCTION ordenar (x IN Persona) RETURN INTEGER,
  ------------------------
  MEMBER PROCEDURE mostrar,
  MEMBER FUNCTION calcularDescuento RETURN NUMBER
);
/
```

Creamos las referencias auxiliares para almacenar los punteros almacenados que son personas.

```
  r1 REF Persona;
  r2 REF Persona;
  r3 REF Persona;


  --------------------------------EJERCICIO B--------------------------
------------
--Relaciono a Carlos con Eva

SELECT r.Puntero INTO r1
FROM Referencias r
WHERE r.dni='987654321'; →dni de Eva

UPDATE PersonaTab SET relacionado=r1 WHERE dni='123456789';  →dni de
Carlos


--Relaciono a Eva con Antonio

SELECT r.Puntero INTO r2
FROM Referencias r
WHERE r.dni='111000111'; →dni de Antonio

UPDATE Universidad SET relacionado=r2 WHERE dni='987654321'; →dni de
Eva
```

```
--Relaciono a Antonio con Carlos

SELECT r.Puntero INTO r3
FROM Referencias r
WHERE r.dni='123456789'; →dni de Carlos

UPDATE Universidad SET relacionado=r3 WHERE dni='111000111'; →dni de
Antonio
  ------------------------------------------------------------------------
-------------
```

Se obtiene primero el puntero de la referencia y lo almaceno en la variable auxiliar r usando el dni de la referencia. Con ello almacenamos la persona en r y una vez conseguida, actualizamos la tupla de la tabla con la que queremos relacionarlo usando el dni y guardamos la persona almacenada en r (la referencia) en el atributo **relacionado** de la tupla.

Apartado C

Realizar **consultas** que ilustren las relaciones existentes entre las instancias almacenadas en la base de datos, haciendo uso de **REF** / **DEREF** / **VALUE**, del tipo:

Consulta sobre Eva:

```
SELECT VALUE (a) into persona_a
FROM Universidad a
WHERE a.dni= '987654321';

SELECT DEREF (a.relacionado) into persona_a
FROM Universidad a
WHERE a.dni= persona_a.dni;
dbms_output.put_line('Relacion 1: ');
persona_a.mostrar();
```

Creo una variable auxiliar llamada persona_a de tipo Persona donde almaceno la tupla de la tabla Universidad con valor en dni = 987654321 que se corresponde a Eva.

Con la función DEREF conseguimos el atributo relacionado de la tabla universidad con el dni que conseguimos con el **VALUE** y este valor lo almacenamos en la variable persona_a y, en este caso, se corresponde con la tupla Antonio. Si hubiese puesto en el atributo persona_a.dni en el SELECT DERF directamente el atributo 9876543321, hubiese obtenido el mismo resultado pero lo hago así para trabajar con el VALUE. Después llamamos al método mostrar con la variable persona_a en la que se guarda la tupla Antonio y mostramos sus datos.

Consulta sobre Carlos:

```
SELECT DEREF (x.relacionado) into persona_a
FROM PersonaTab x
WHERE x.dni= '123456789';
dbms_output.put_line('Relacion 2: ');
persona_a.mostrar();
```

Se almacena en la variable persona_a el valor del atributo relacionado de la tupla que hay insertada en la tabla PersonaTab con el atributo dni con valor 123456789 que se corresponde al dni de Carlos. Después llamamos al método mostrar con la variable persona_a en la que se guarda la tupla Eva y mostramos sus datos.

Consulta sobre Antonio:

```
SELECT DEREF (a.relacionado) into persona_a
FROM Universidad a
WHERE a.dni= '111000111';
dbms_output.put_line('Relacion 3: ');
persona_a.mostrar();
```

Se almacena en la variable persona_a el valor del atributo relacionado de la tupla que hay insertada en la tabla PersonaTab con el atributo dni con valor 111000111 que se corresponde al dni de Antonio. Después llamamos al método mostrar con la variable persona_a en la que se guarda la tupla Carlos y mostramos sus datos.

Apartado D:

Sustituir el método de comparación de objetos (**MAP**) implementándolo como de tipo **ORDER**, gestionando adecuadamente todos los cambios que ello conlleva.

En primer lugar nos dirigimos a la clase Persona y modificamos esto:

```
MAP MEMBER FUNCTION ret_value RETURN NUMBER
```

Por esto:

```
ORDER MEMBER FUNCTION ordenar (x IN Persona) RETURN INTEGER
```

Con esto podemos realizar comparaciones con otro objeto insertado por parámetros. Solo puede haber un método de comparación en cada tipo.

Llamo ordenar al método en el que comparo dos dni (el de la persona y el que introduzco) para saber cuál de ellos es mayor de los dos. Para ello también es necesario definir el método en la clase persona:

```
CREATE OR REPLACE TYPE BODY Persona AS

  --MAP MEMBER FUNCTION ret_value RETURN NUMBER IS
  --BEGIN
    --RETURN dni;
  --END;

  ORDER MEMBER FUNCTION ordenar(x in Persona) RETURN INTEGER IS
  BEGIN
    RETURN x.dni - dni;
  END ;

  MEMBER PROCEDURE mostrar IS
    --i INTEGER; -- No hace falta declarar i
    vdni VARCHAR2(9);
    vpais VARCHAR2(2);
    vcontrol VARCHAR2(2);
    ventidad VARCHAR2(4);
    voficina VARCHAR2(4);
    vdc VARCHAR2(2);
    vcuenta VARCHAR2(10);

    CURSOR cursor_ver_cuentas IS
       SELECT p.DNI, c.pais, c.control, c.entidad, c.oficina, c.dc,
c.cuenta
       FROM PersonaTab p, table(p.cuentas) c;

  BEGIN
    dbms_output.put_line('Método mostrar() de objeto Persona');
    dbms_output.put_line('DNI: ' || dni);
    dbms_output.put_line('Nombre: ' || datosPersonales.nombre);
    dbms_output.put_line('Apellido 1: ' || datosPersonales.ape1);
    dbms_output.put_line('Apellido 2: ' || datosPersonales.ape2);
    dbms_output.put_line('Fecha nacimiento: ' || fechanac);
    dbms_output.put_line('Dirección: ' || direccion);
    dbms_output.put_line('CP: ' || cp);
    dbms_output.put_line('Sector: ' || sector);

    dbms_output.put_line('Existen ' || telefonos.count || '
teléfonos');
    IF (telefonos.count <> 0) THEN
      FOR i IN 1 .. telefonos.count LOOP
         dbms_output.put_line('     Teléfono ' || i || ': ' ||
telefonos(i));
      END LOOP;
    END IF;

    dbms_output.put_line('Existen ' || cuentas.count || ' cuentas
bancarias');
    OPEN cursor_ver_cuentas;
    LOOP
      FETCH cursor_ver_cuentas INTO vDNI, vpais, vcontrol, ventidad,
voficina, vdc, vcuenta;
      EXIT WHEN cursor_ver_cuentas%NOTFOUND;
      IF (vDNI = dni) THEN
        dbms_output.put_line('     IBAN: ' || vpais || vcontrol || '-'
|| ventidad || '-' || voficina || '-' || vdc || '-' || vcuenta);
```

```
      END IF;
    END LOOP;
    CLOSE cursor_ver_cuentas;

    dbms_output.put_line('---------------------------------------');
  END; -- de Mostrar

  MEMBER FUNCTION calcularDescuento RETURN NUMBER IS
  BEGIN
    IF sector=1 THEN
      RETURN 10;
    ELSE
      RETURN 0;
    END IF;
  END; -- de calcularDescuento


END; -- de Persona
/
```

Una vez hecho esto, puedo realizar operaciones para saber qué dni es mayor de ambos (también pueden ser iguales). Para ello desde una tupla llamo al método con un parámetro de tipo persona y el resultado obtenido lo utilizamos para indicar cuál es mayor de ambos:

```
                --------APARTADO D:---------
  dbms_output.put_line('COMPARACIÓN DE OBJETOS: MÉTODOS ORDENAR');
IF (persona_a.ordenar(persona_b)<0) THEN
    dbms_output.put_line(persona_a.dni || ' es mayor que ' ||
persona_b.dni);
  ELSE
  IF (persona_a.ordenar(persona_b)=0) THEN
    dbms_output.put_line(persona_a.dni || ' es igual que ' ||
persona_b.dni);
  ELSE
    dbms_output.put_line(persona_a.dni || ' es menor que ' ||
persona_b.dni);
  END IF;
END IF;
```

En el primer IF se comprueba si el dni de la persona que llama el método es mayor que la que se introduce por parámetros. Si el valor que se genera de restar el dni la tupla insertada menos la tupla de la persona que llama al método es negativo, significa la tupla que llama al método tiene mayor valor de dni.

En el segundo IF compruebo si los valores son iguales, para ello la resta de un dni con el otro debe ser igual a 0. En ese caso, tendrían el mismo dni y se mostraría un mensaje diciendo que tienen el mismo valor de dni.

Si no es ninguno de los casos anteriores, significa que en el último ELSE se muestra un mensaje indicando que el dni de la persona que se utiliza como parámetro en el método es mayor que el dni de la persona que llama al método, por tanto, el dni de la persona que llama al método es menor y se muestra un mensaje.

Apartado E:

Añadir sobrecarga en algún método de la clase `Trabajador`, en herencia. Por ejemplo, sobrecargar el método `calcularDescuento` implementando una versión que reciba algún parámetro.

Nos dirigimos a la clase Trabajador y añadimos lo siguiente:

```
CREATE TYPE Trabajador UNDER Persona (
  ingresos NUMBER(7,2),
  gastosFijos NUMBER(7,2),
  gastosAlim NUMBER(7,2),
  gastosRopa NUMBER(7,2),
  OVERRIDING MEMBER PROCEDURE mostrar,
  --APARTADO E:
  --en función de los años trabajando se le hará más o menos descuento
  MEMBER FUNCTION calcularDescuento (años IN NUMBER) RETURN NUMBER
);
/
```

Con esto sobrecargamos el método calcularDescuento de la clase Persona y todas las tuplas de trabajadores tendrán un calcularDescuento distinto. En mi caso he optado por crear un método que en función de los años que se lleve trabajados en la empresa, se haga un tipo de descuento u otro siendo más beneficiado aquel que lleve más años. En la clase Trabajador habría que añadir lo siguiente:

```
CREATE OR REPLACE TYPE BODY Trabajador AS

  OVERRIDING MEMBER PROCEDURE mostrar IS
    vdni VARCHAR2(9);
    vpais VARCHAR2(2);
    vcontrol VARCHAR2(2);
    ventidad VARCHAR2(4);
    voficina VARCHAR2(4);
    vdc VARCHAR2(2);
    vcuenta VARCHAR2(10);

    CURSOR cursor_ver_cuentas IS
       SELECT p.DNI, c.pais, c.control, c.entidad, c.oficina, c.dc,
c.cuenta
       FROM Empresa p, table(p.cuentas) c;

  BEGIN
    dbms_output.put_line('Método mostrar() de objeto Estudiante');
    dbms_output.put_line('DNI: ' || dni);
    dbms_output.put_line('Nombre: ' || datosPersonales.nombre);
    dbms_output.put_line('Apellido 1: ' || datosPersonales.ape1);
    dbms_output.put_line('Apellido 2: ' || datosPersonales.ape2);
    dbms_output.put_line('Fecha nacimiento: ' || fechanac);
    dbms_output.put_line('Dirección: ' || direccion);
    dbms_output.put_line('CP: ' || cp);
    dbms_output.put_line('Sector: ' || sector);
```

```
    dbms_output.put_line('Existen ' || telefonos.count || '
teléfonos');
    IF (telefonos.count <> 0) THEN
      FOR i IN 1 .. telefonos.count LOOP
         dbms_output.put_line('     Teléfono ' || i || ': ' ||
telefonos(i));
      END LOOP;
    END IF;

    dbms_output.put_line('Existen ' || cuentas.count || ' cuentas
bancarias');
    OPEN cursor_ver_cuentas;
    LOOP
      FETCH cursor_ver_cuentas INTO vDNI, vpais, vcontrol, ventidad,
voficina, vdc, vcuenta;
      EXIT WHEN cursor_ver_cuentas%NOTFOUND;
      IF (vDNI = dni) THEN
        dbms_output.put_line('     IBAN: ' || vpais || vcontrol || '-'
|| ventidad || '-' || voficina || '-' || vdc || '-' || vcuenta);
      END IF;
    END LOOP;
    CLOSE cursor_ver_cuentas;

    dbms_output.put_line('Ingresos: ' || ingresos);
    dbms_output.put_line('Gastos Fijos: ' || gastosFijos);
    dbms_output.put_line('Gastos Alimentacion: ' || gastosAlim);
    dbms_output.put_line('Gastos Ropas: ' || gastosRopa);


    dbms_output.put_line('---------------------------------------');
  END; -- de Mostrar

  MEMBER FUNCTION calcularDescuento (años IN NUMBER ) RETURN NUMBER IS
    BEGIN
        IF (años>10) THEN
            RETURN 30;
        ELSE
            RETURN 15;
        END IF ;
    END ; -- de calcularDescuento


END; -- de Trabajador
/
```

Como se observa hay que añadir un número al método que se corresponde con los años trabajado en la empresa y en función de ellos se hace un descuento. Concretamente se realiza un descuento de 30% si llevas más de 10 años trabajados y, en otro caso, se aplicaría sólo un descuento de 15%.

Se adjunta una llamada al método:

```
  SELECT VALUE(c) INTO persona_c
  FROM Empresa c
  where c.DNI='111000333';
  -- Acceso a atributos
  dbms_output.put_line(persona_c.dni);
  -- Llamada a métodos
  persona_c.mostrar();
  --Apartado e:
  dbms_output.put_line('Su descuento es de: '||
persona_c.calcularDescuento(20));
```

En este caso daría 30% ya que 20>10.

Apartado F:

Mostrar consultas sobre colecciones (VARRAY / tablas anidadas) donde se ilustre la inclusión de la nueva clase `Trabajador` implementada en estos ejercicios.

Hay que conseguir de la tabla Empresa todos los dni y teléfonos de los trabajadores al igual que hemos hecho con la tabla PersonaTab. Para ello realizamos la siguiente consulta:

```
SELECT e.dni, e.telefonos -- varray
FROM Empresa e;
```

Para conseguir los dni de los trabajadores junto con sus cuentas sería así:

```
SELECT e.dni, e.cuentas -- tablas anidadas
FROM Empresa e;
```

Para finalizar se realiza una consulta para ver todos los objetos almacenados con la función DEREF:

```
-- CONSULTA DE OBJETOS ALMACENADOS
SELECT DEREF(c.Puntero)
FROM Referencias c;
```

Por último, borramos todas las tablas y secuencias:

```
/* -------------------------------- */
/* BORRAR TABLAS, OBJETOS, TIPOS ... */
/* -------------------------------- */

DROP SEQUENCE num_cuenta;

DROP TABLE PersonaTab;
DROP TABLE Universidad;
```

```
DROP TABLE Empresa;
DROP TABLE Referencias;

DROP TYPE Trabajador;
DROP TYPE Estudiante;
DROP TYPE Persona;
DROP TYPE nombreCompleto;
DROP TYPE numTelfs;
DROP TYPE cuentasTab;
DROP TYPE IBANtipo;
```

Salida por consola:

```
Type NOMBRECOMPLETO compilado


Type NUMTELFS compilado


Type IBANTIPO compilado


Type CUENTASTAB compilado


Type PERSONA compilado


Type PERSONA alterado.


Type ESTUDIANTE compilado


Type TRABAJADOR compilado


Table EMPRESA creado.


Table PERSONATAB creado.


Table UNIVERSIDAD creado.


Table REFERENCIAS creado.


Type Body PERSONA compilado


Type Body ESTUDIANTE compilado
```

```
Type Body TRABAJADOR compilado

Sequence NUM_CUENTA creado.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.
```

```
1 fila insertadas.


1 fila insertadas.


1 fila insertadas.


1 fila insertadas.


1 fila insertadas.

123456789
Método mostrar() de objeto Persona
DNI: 123456789
Nombre: Carlos
Apellido 1: Gutiérrez
Apellido 2: Pérez
Fecha nacimiento: 05/11/97
Dirección: Pz. Colón
CP: 06300
Sector: 1
Existen 2 teléfonos
     Teléfono 1: 654654654
     Teléfono 2: 678678678
Existen 3 cuentas bancarias
     IBAN: ES60-0049-1500-05-1234567892
     IBAN: ES54-2100-0900-15-5556669874
     IBAN: ES28-0182-0045-64-3571598526
----------------------------------------
Descuento: 10
666884444
Método mostrar() de objeto Persona
DNI: 666884444
Nombre: Ignacio
Apellido 1: Costa
Apellido 2: Burgos
Fecha nacimiento: 12/12/82
Dirección: C. Descubrimiento
CP: 10005
Sector: 1
Existen 4 teléfonos
     Teléfono 1: 927242526
     Teléfono 2: 927225005
     Teléfono 3: 657776398
     Teléfono 4: 602485000
Existen 2 cuentas bancarias
     IBAN: ES58-0182-0000-33-1000375786
     IBAN: ES10-2048-0110-13-1230984576
----------------------------------------
Descuento: 10
111222333
Método mostrar() de objeto Estudiante
DNI: 111222333
Nombre: Pedro
Apellido 1: Salas
Apellido 2: Gil
Fecha nacimiento: 11/05/96
```

```
Dirección: Pz. América
CP: 10005
Sector: 1
Existen 3 teléfonos
     Teléfono 1: 927229411
     Teléfono 2: 927232626
     Teléfono 3: 667555666
Existen 2 cuentas bancarias
     IBAN: ES43-2038-0111-22-0000123456
     IBAN: ES53-2048-0222-12-0022993388
Titulación: Grado Informática
Asignaturas aprobadas: 23
--------------------------------------
Descuento: 20
Se muestran las nuevas inserciones
111000333
Método mostrar() de objeto Estudiante
DNI: 111000333
Nombre: Carlos
Apellido 1: Guillen
Apellido 2: Moreno
Fecha nacimiento: 01/07/89
Dirección: C. Constitución
CP: 06800
Sector: 3
Existen 0 teléfonos
Existen 2 cuentas bancarias
     IBAN: ES79-2038-0010-73-8365936594
     IBAN: ES82-2048-0200-32-8465836593
Ingresos: 2000
Gastos Fijos: 500
Gastos Alimentacion: 600
Gastos Ropas: 200
--------------------------------------
Su descuento es de: 30
111000888
Método mostrar() de objeto Estudiante
DNI: 111000888
Nombre: Lionel
Apellido 1: Messi
Apellido 2: Estaca
Fecha nacimiento: 01/07/89
Dirección: C. Constitución
CP: 06800
Sector: 3
Existen 0 teléfonos
Existen 2 cuentas bancarias
     IBAN: ES79-2038-0010-73-8365936595
     IBAN: ES82-2048-0200-32-8465836594
Ingresos: 3000
Gastos Fijos: 600
Gastos Alimentacion: 700
Gastos Ropas: 300
--------------------------------------
111000999
Método mostrar() de objeto Estudiante
DNI: 111000999
Nombre: Cristiano
Apellido 1: Ronaldo
Apellido 2: Mazapan
Fecha nacimiento: 01/07/89
```

```
Dirección: C. Constitución
CP: 06800
Sector: 3
Existen 0 teléfonos
Existen 2 cuentas bancarias
     IBAN: ES79-2038-0010-73-8365936596
     IBAN: ES82-2048-0200-32-8465836595
Ingresos: 4000
Gastos Fijos: 700
Gastos Alimentacion: 800
Gastos Ropas: 700
---------------------------------------
Fin de las nuevas inserciones
COMPARACIÓN DE OBJETOS: MÉTODOS ORDENAR
666884444 es mayor que 111222333
OBTENCIÓN DE REFERENCIAS: CON DEREF A PERSONA
123456789
Método mostrar() de objeto Persona
DNI: 123456789
Nombre: Carlos
Apellido 1: Gutiérrez
Apellido 2: Pérez
Fecha nacimiento: 05/11/97
Dirección: Pz. Colón
CP: 06300
Sector: 1
Existen 2 teléfonos
     Teléfono 1: 654654654
     Teléfono 2: 678678678
Existen 3 cuentas bancarias
     IBAN: ES60-0049-1500-05-1234567892
     IBAN: ES54-2100-0900-15-5556669874
     IBAN: ES28-0182-0045-64-3571598526
---------------------------------------
Descuento: 10
OBJENCIÓN DE REFERENCIAS: CON DEREF A ESTUDIANTE
111222333
Método mostrar() de objeto Estudiante
DNI: 111222333
Nombre: Pedro
Apellido 1: Salas
Apellido 2: Gil
Fecha nacimiento: 11/05/96
Dirección: Pz. América
CP: 10005
Sector: 1
Existen 3 teléfonos
     Teléfono 1: 927229411
     Teléfono 2: 927232626
     Teléfono 3: 667555666
Existen 2 cuentas bancarias
     IBAN: ES43-2038-0111-22-0000123456
     IBAN: ES53-2048-0222-12-0022993388
Titulación: Grado Informática
Asignaturas aprobadas: 23
---------------------------------------
Descuento: 20
Relacion 1:
Método mostrar() de objeto Estudiante
DNI: 111000111
Nombre: Antonio
```

```
Apellido 1: Muñoz
Apellido 2: Hernández
Fecha nacimiento: 01/07/89
Dirección: C. Constitución
CP: 06800
Sector: 3
Existen 0 teléfonos
Existen 2 cuentas bancarias
     IBAN: ES79-2038-0010-73-8365936594
     IBAN: ES82-2048-0200-32-8465836593
Titulación: Grado Veterinaria
Asignaturas aprobadas: 12
--------------------------------------
Relacion 2:
Método mostrar() de objeto Estudiante
DNI: 987654321
Nombre: Eva
Apellido 1: Moreno
Apellido 2: Pozo
Fecha nacimiento: 10/05/74
Dirección: C. Justicia
CP: 10005
Sector: 3
Existen 2 teléfonos
     Teléfono 1: 927223182
     Teléfono 2: 657332211
Existen 1 cuentas bancarias
     IBAN: ES10-2100-1004-10-1004715886
Titulación: Grado Edificación
Asignaturas aprobadas: 3
--------------------------------------
Relacion 3:
Método mostrar() de objeto Persona
DNI: 123456789
Nombre: Carlos
Apellido 1: Gutiérrez
Apellido 2: Pérez
Fecha nacimiento: 05/11/97
Dirección: Pz. Colón
CP: 06300
Sector: 1
Existen 2 teléfonos
     Teléfono 1: 654654654
     Teléfono 2: 678678678
Existen 3 cuentas bancarias
     IBAN: ES60-0049-1500-05-1234567892
     IBAN: ES54-2100-0900-15-5556669874
     IBAN: ES28-0182-0045-64-3571598526
--------------------------------------

Procedimiento PL/SQL terminado correctamente.

DNI
---------
TELEFONOS
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```


123456789
NUMTELFS('654654654', '678678678')

666884444
NUMTELFS('927242526', '927225005', '657776398', '602485000')

999000111
NUMTELFS('654159753')

DNI

TELEFONOS

666999333
NUMTELFS('924333435', '600700800')

DNI

CUENTAS(COD, PAIS, CONTROL, ENTIDAD, OFICINA, DC, CUENTA)


```
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-------------------------------------------
123456789
CUENTASTAB(IBANTIPO('1', 'ES', '60', '0049', '1500', '05',
'1234567892'), IBANTIPO('2', 'ES', '54', '2100', '0900', '15',
'5556669874'), IBANTIPO('3', 'ES', '28', '0182', '0045', '64',
'3571598526'))

666884444
CUENTASTAB(IBANTIPO('058', 'ES', '10', '2048', '0110', '13',
'1230984576'), IBANTIPO('479', 'ES', '58', '0182', '0000', '33',
'1000375786'))

999000111
CUENTASTAB(IBANTIPO('123', 'ES', '16', '0049', '1551', '45',
'1555567892'))


DNI
---------
CUENTAS(COD, PAIS, CONTROL, ENTIDAD, OFICINA, DC, CUENTA)
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-------------------------------------------
666999333
CUENTASTAB(IBANTIPO('124', 'ES', '16', '0049', '1551', '45',
'1555567892'), IBANTIPO('125', 'ES', '44', '2100', '0910', '13',
'9756385634'))



DNI       Secuencia
--------- ---------
123456789 654654654
123456789 678678678
666884444 927242526
```

```
666884444 927225005
666884444 657776398
666884444 602485000
999000111 654159753
666999333 924333435
666999333 600700800

9 filas seleccionadas.


DNI       COD   PA CO ENTI OFIC DC CUENTA
--------- ----- -- -- ---- ---- -- ----------
123456789 1     ES 60 0049 1500 05 1234567892
123456789 2     ES 54 2100 0900 15 5556669874
123456789 3     ES 28 0182 0045 64 3571598526
666884444 479   ES 58 0182 0000 33 1000375786
666884444 058   ES 10 2048 0110 13 1230984576
999000111 123   ES 16 0049 1551 45 1555567892
666999333 124   ES 16 0049 1551 45 1555567892
666999333 125   ES 44 2100 0910 13 9756385634

8 filas seleccionadas.


DNI
---------
TELEFONOS
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------
111000333
NUMTELFS()

111000888
NUMTELFS()

111000999
NUMTELFS()



DNI
---------
CUENTAS(COD, PAIS, CONTROL, ENTIDAD, OFICINA, DC, CUENTA)
```

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------------------------------------
111000333
CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010', '73',
'8365936594'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836593'))

111000888
CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010', '73',
'8365936595'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836594'))

111000999
CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010', '73',
'8365936596'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836595'))



DEREF(C.PUNTERO)
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------------------------------------
PERSONA('123456789', SYSTEM.ESTUDIANTE('987654321',
'oracle.sql.REF@4c764efb', SYSTEM.NOMBRECOMPLETO('Eva', 'Moreno',
'Pozo'), '1974-05-10 00:00:00.0', 'C. Justicia', '10005', 'M', 3,
SYSTEM.NUMTELFS(927223182, 657332211),
SYSTEM.CUENTASTAB(SYSTEM.IBANTIPO('126', 'ES', '10', '2100', '1004',
```

```
'10', '1004715886')), 'Grado Edificación', 3),
NOMBRECOMPLETO('Carlos', 'Gutiérrez', 'Pérez'), '05/11/97', 'Pz.
Colón', '06300', 'H', 1, NUMTELFS('654654654', '678678678'),
CUENTASTAB(IBANTIPO('1', 'ES', '60', '0049', '1500', '05',
'1234567892'), IBANTIPO('2', 'ES', '54', '2100', '0900', '15',
'5556669874'), IBANTIPO('3', 'ES', '28', '0182', '0045', '64',
'3571598526')))
PERSONA('666884444', NULL, NOMBRECOMPLETO('Ignacio', 'Costa',
'Burgos'), '12/12/82', 'C. Descubrimiento', '10005', 'H', 1,
NUMTELFS('927242526', '927225005', '657776398', '602485000'),
CUENTASTAB(IBANTIPO('058', 'ES', '10', '2048', '0110', '13',
'1230984576'), IBANTIPO('479', 'ES', '58', '0182', '0000', '33',
'1000375786')))
PERSONA('999000111', NULL, NOMBRECOMPLETO('Francisco', 'Fernández',
'Merchán'), '12/03/79', 'C. Poniente', '10800', 'H', 2,
NUMTELFS('654159753'), CUENTASTAB(IBANTIPO('123', 'ES', '16', '0049',
'1551', '45', '1555567892')))
PERSONA('666999333', NULL, NOMBRECOMPLETO('Paula', 'Ordóñez', 'Ruiz'),
'01/02/90', 'C. Atlántico', '06800', 'M', 2, NUMTELFS('924333435',
'600700800'), CUENTASTAB(IBANTIPO('124', 'ES', '16', '0049', '1551',
'45', '1555567892'), IBANTIPO('125', 'ES', '44', '2100', '0910', '13',
'9756385634')))
ESTUDIANTE('111222333', NULL, NOMBRECOMPLETO('Pedro', 'Salas', 'Gil'),
'11/05/96', 'Pz. América', '10005', 'H', 1, NUMTELFS('927229411',
'927232626', '667555666'), CUENTASTAB(IBANTIPO('047', 'ES', '43',
'2038', '0111', '22', '0000123456'), IBANTIPO('053', 'ES', '53',
'2048', '0222', '12', '0022993388')), 'Grado Informática', 23)
ESTUDIANTE('987654321', SYSTEM.ESTUDIANTE('111000111',
'oracle.sql.REF@d7f442e0', SYSTEM.NOMBRECOMPLETO('Antonio', 'Muñoz',
'Hernández'), '1989-07-01 00:00:00.0', 'C. Constitución', '06800',
'H', 3SYSTEM.NUMTELFS(), SYSTEM.CUENTASTAB(SYSTEM.IBANTIPO('047',
'ES', '79', '2038', '0010', '73', '8365936594'),
SYSTEM.IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836593')), 'Grado Veterinaria', 12), NOMBRECOMPLETO('Eva',
'Moreno', 'Pozo'), '10/05/74', 'C. Justicia', '10005', 'M', 3,
NUMTELFS('927223182', '657332211'), CUENTASTAB(IBANTIPO('126', 'ES',
'10', '2100', '1004', '10', '1004715886')), 'Grado Edificación', 3)
ESTUDIANTE('111000111', SYSTEM.PERSONA('123456789',
'oracle.sql.REF@4ef74cb1', SYSTEM.NOMBRECOMPLETO('Carlos',
'Gutiérrez', 'Pérez'), '1997-11-05 00:00:00.0', 'Pz. Colón', '06300',
'H', 1, SYSTEM.NUMTELFS(654654654, 678678678),
SYSTEM.CUENTASTAB(SYSTEM.IBANTIPO('1', 'ES', '60', '0049', '1500',
'05', '1234567892'), SYSTEM.IBANTIPO('2', 'ES', '54', '2100', '0900',
'15', '5556669874'), SYSTEM.IBANTIPO('3', 'ES', '28', '0182', '0045',
'64', '3571598526'))), NOMBRECOMPLETO('Antonio', 'Muñoz',
'Hernández'), '01/07/89', 'C. Constitución', '06800', 'H', 3,
NUMTELFS(), CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010',
'73', '8365936594'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836593')), 'Grado Veterinaria', 12)
TRABAJADOR('111000333', NULL, NOMBRECOMPLETO('Carlos', 'Guillen',
'Moreno'), '01/07/89', 'C. Constitución', '06800', 'H', 3, NUMTELFS(),
CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010', '73',
'8365936594'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836593')), 2000, 500, 600, 200)
TRABAJADOR('111000888', NULL, NOMBRECOMPLETO('Lionel', 'Messi',
'Estaca'), '01/07/89', 'C. Constitución', '06800', 'H', 3, NUMTELFS(),
CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010', '73',
'8365936595'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836594')), 3000, 600, 700, 300)
TRABAJADOR('111000999', NULL, NOMBRECOMPLETO('Cristiano', 'Ronaldo',
'Mazapan'), '01/07/89', 'C. Constitución', '06800', 'H', 3,
```

```
NUMTELFS(), CUENTASTAB(IBANTIPO('047', 'ES', '79', '2038', '0010',
'73', '8365936596'), IBANTIPO('053', 'ES', '82', '2048', '0200', '32',
'8465836595')), 4000, 700, 800, 700)

10 filas seleccionadas.


Sequence NUM_CUENTA borrado.


Table PERSONATAB borrado.


Table UNIVERSIDAD borrado.


Table EMPRESA borrado.


Table REFERENCIAS borrado.


Type TRABAJADOR borrado.


Type ESTUDIANTE borrado.


Type PERSONA borrado.


Type NOMBRECOMPLETO borrado.


Type NUMTELFS borrado.


Type CUENTASTAB borrado.


Type IBANTIPO borrado.
```