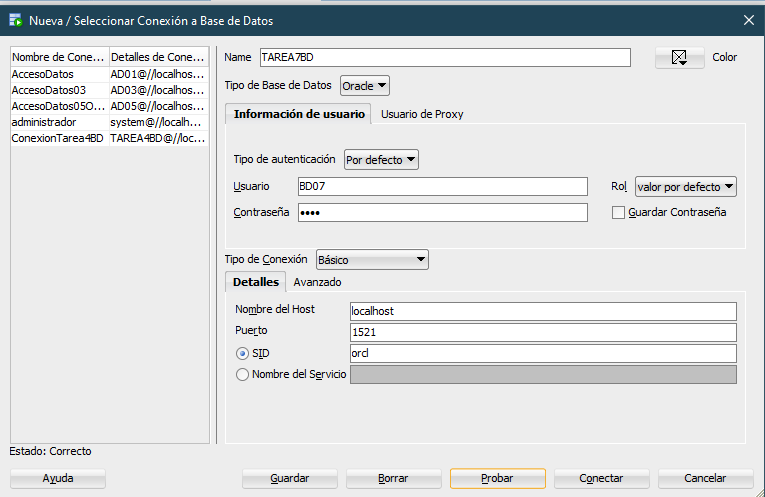
|  |  |
| --- | --- |
| Tarea 7 BD Curso 20/21 | Fecha ordinaria de entrega : 28/04/2021 |
| Nombre y apellidos:  Jose Antonio Acevedo Paredes. | |

Enunciado.

Creado usuario BD07 creamos una conexión nueva para realizar la tarea llamada TAREA7BD.



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

codigo INTEGER,

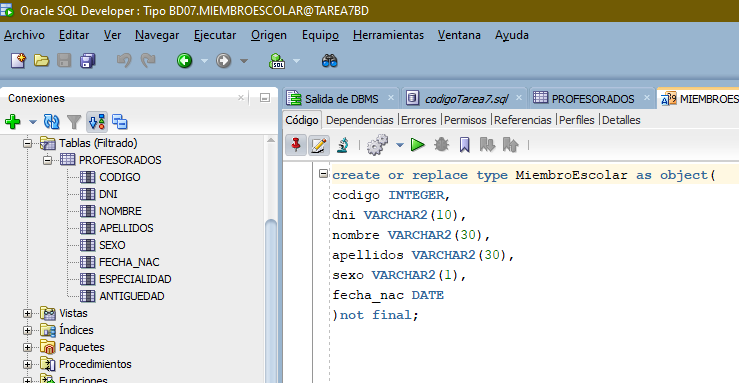
dni VARCHAR2(10),

nombre VARCHAR2(30),

apellidos VARCHAR2(30),

sexo VARCHAR2(1),

fecha\_nac DATE



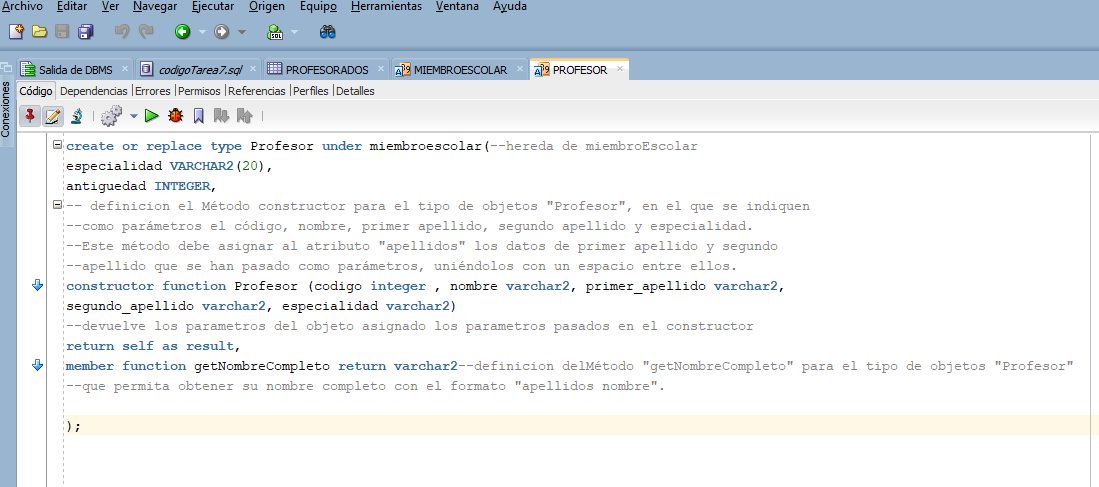
2. Crea, como tipo heredado de "MiembroEscolar", el tipo de objeto "Profesor" con los siguientes atributos y métodos:

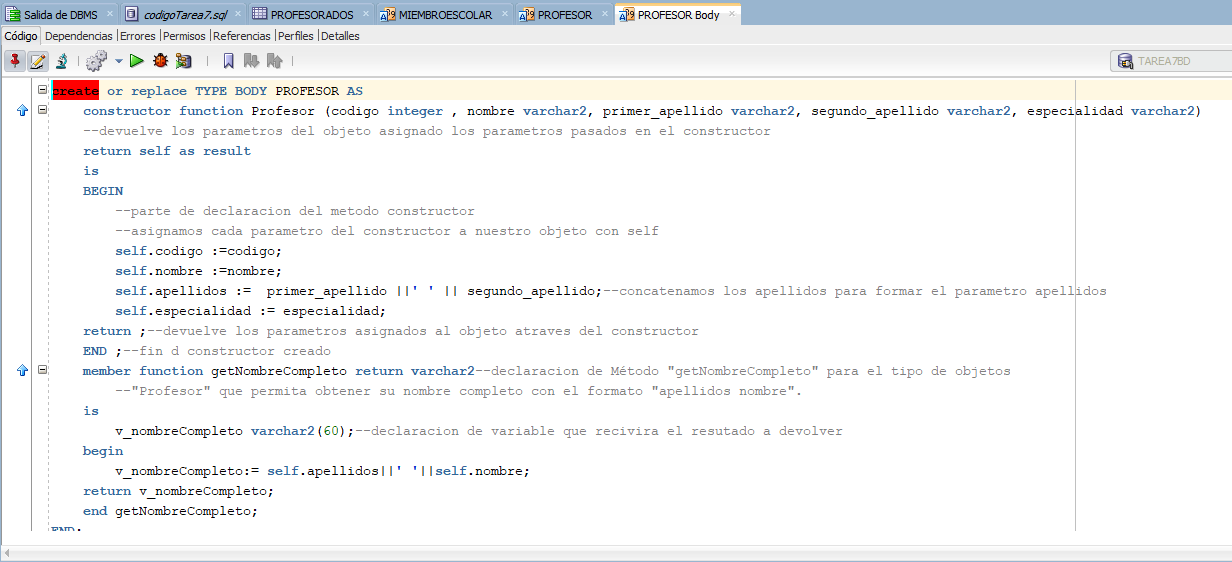
especialidad VARCHAR2(20),

antiguedad INTEGER

Método constructor para el tipo de objetos "Profesor", en el que se indiquen como parámetros el código, nombre, primer apellido, segundo apellido y especialidad. 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" para el tipo de objetos "Profesor" que permita obtener su nombre completo con el formato "apellidos nombre".





3. Crea el tipo de objeto "Cursos" con los siguientes atributos y métodos:

codigo INTEGER,

nombre VARCHAR2(20),

refProfe REF Profesor,

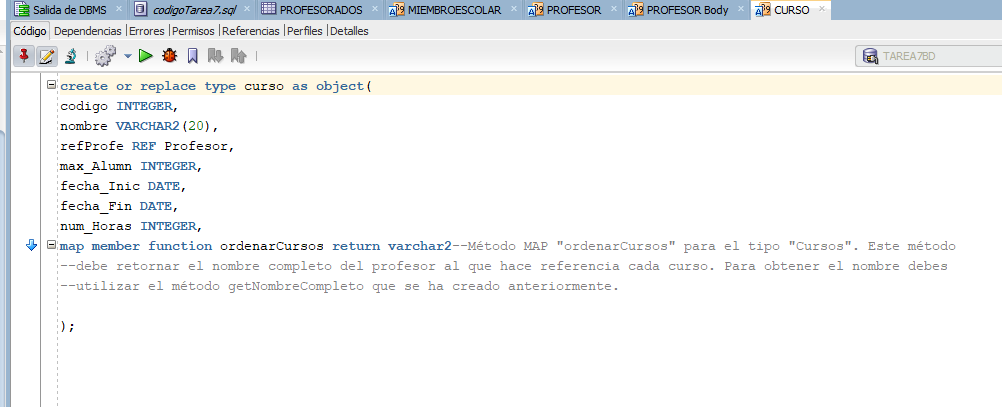
max\_Alumn INTEGER,

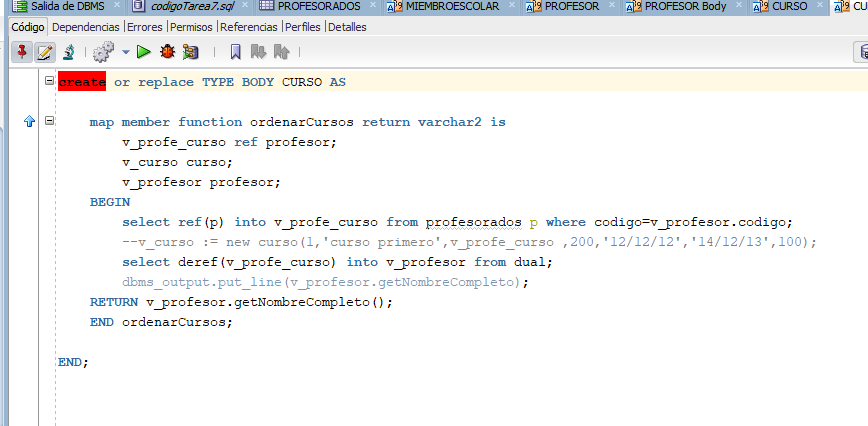
fecha\_Inic DATE,

fecha\_Fin DATE,

num\_Horas INTEGER

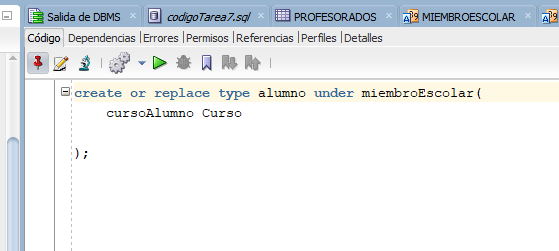
Método MAP "ordenarCursos" para el tipo "Cursos". Este método debe retornar el nombre completo del profesor al que hace referencia cada curso. Para obtener el nombre debes utilizar el método getNombreCompleto que se ha creado anteriormente.



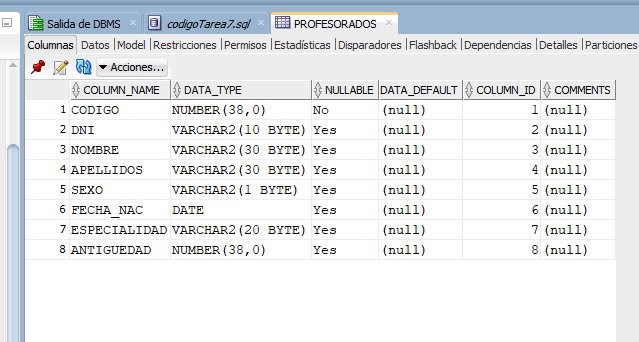


4. Crea, como tipo heredado de "MiembroEscolar", el tipo de objeto "Alumno" con los siguientes atributos:

cursoAlumno Cursos



5. Crea un tipo de tabla "Profesorado" de objetos "Profesor".



6. Inserta en dicha tabla dos objetos "Profesor". El primero de ellos con los datos:

codigo: 2

dni: 51083099F

nombre: MARIA LUISA

apellidos: FABRE BERDUN

sexo: F

fecha\_nac: 31/03/1975

especialidad: TECNOLOGIA

antiguedad: 4

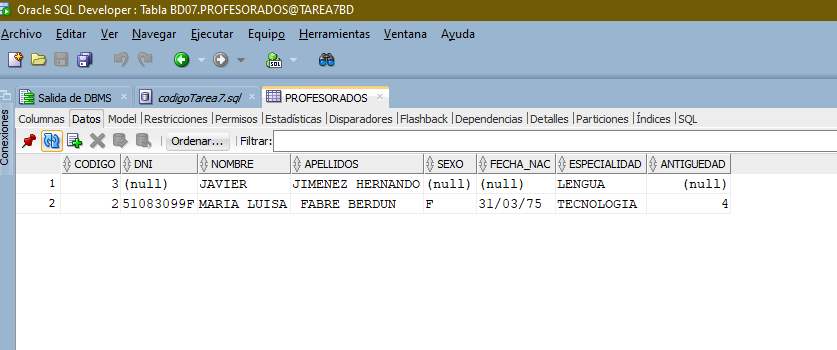
El segundo objeto "Profesor" debes crearlo usando el método constructor que has realizado anteriormente. Debes usar los siguientes datos:

codigo: 3

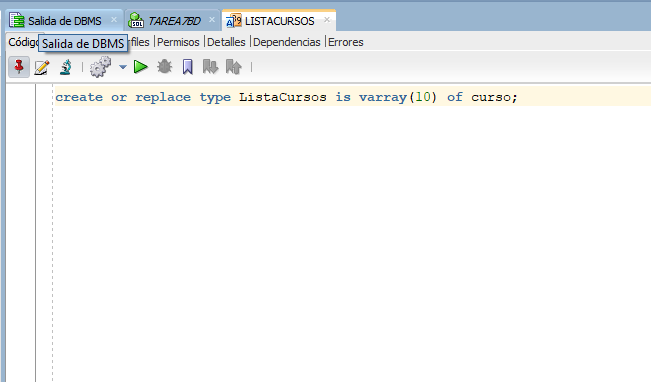
nombre: JAVIER

apellidos: JIMENEZ HERNANDO

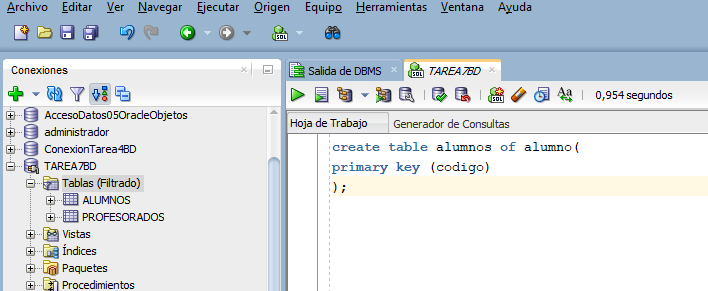
especialidad: LENGUA



7. Crea un tipo colección VARRAY llamada "ListaCursos" en la que se puedan almacenar hasta 10 objetos "Cursos".



8. Crea una tabla "Alumnado" de objetos "Alumno".



9. Crea un bloque de código que haga todo lo siguiente:

a) Guarda en una instancia "listaCursos1" de dicha lista, los dos cursos siguientes:

El primer curso que debes almacenar en dicha lista debe tener los siguientes datos:

codigo: 1

nombre: Curso 1

refProfe: Referencia al profesor cuyo codigo es 3.

max\_Alumn: 20

fecha\_Inic: 1/6/2011

fecha\_Fin: 30/6/2011

num\_Horas: 30

El segundo curso que debes almacenar en dicha lista debe tener los siguientes datos:

codigo: 2

nombre: Curso 2

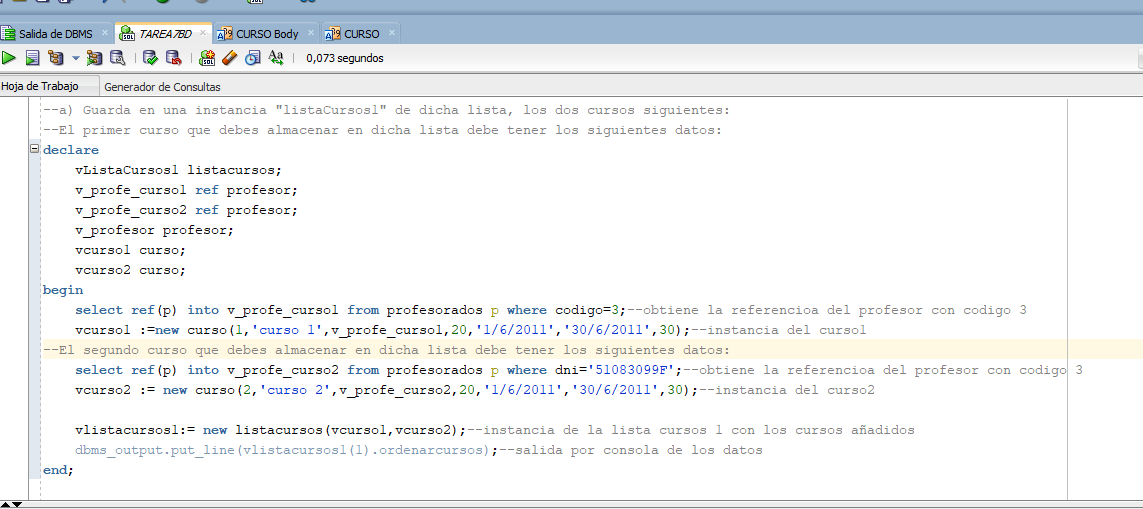
refProfe: Referencia al profesor cuyo DNI es 51083099F.

max\_Alumn: 20

fecha\_Inic: 1/6/2011

fecha\_Fin: 30/6/2011

num\_Horas: 30



b) Inserta en la tabla "Alumnado" las siguientes filas:

codigo: 100

dni: 76401092Z

nombre: MANUEL

apellidos: SUAREZ IBAÑEZ

sexo: M

fecha\_nac: 30/6/1990

cursoAlumno: objeto creado anteriormente para el primer curso

codigo: 102

dni: 6915588V

nombre: MILAGROSA

apellidos: DIAZ PEREZ

sexo: F

fecha\_nac: 28/10/1984

cursoAlumno: objeto que se encuentre en la segunda posición de "listaCursos1"

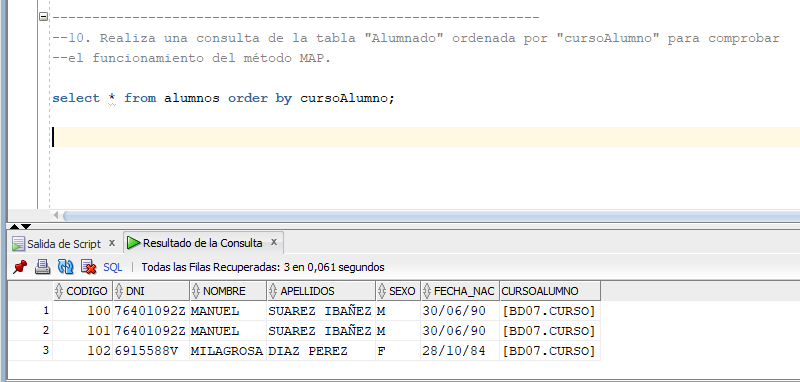
 (debe tomarse de la lista)

Se añade las líneas al final del bloque anónimo antes de end;

c) Obtener, de la tabla "Alumnado", el alumno que tiene el código 100, asignándoselo a una variable "unAlumno".

d) Modifica el código del alumno guardado en esa variable "unAlumno" asignando el valor 101, y su curso debe ser el segundo que se había creado anteriormente. Inserta ese alumno en la tabla "Alumnado".

Para realizar estos cambios hay que pasar el codigo al bloque anónimo creado para instanciar la lista de cursos, ya que no tenemos acceso a la instancia de la lista de cursos.

10. Realiza una consulta de la tabla "Alumnado" ordenada por "cursoAlumno" para comprobar el funcionamiento del método MAP.  


**CODIGO REALIZADO PARA LA TAREA:**

Tarea 7 BD Curso 20/21 Fecha ordinaria de entrega : 28/04/2021

Nombre y apellidos:

Jose Antonio Acevedo Paredes.

codigo de la tarea:

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

respuesta:

create or replace type MiembroEscolar 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 "MiembroEscolar", el tipo de objeto "Profesor" con los siguientes atributos y métodos:

Método constructor para el tipo de objetos "Profesor", en el que se indiquen como parámetros el código, nombre, primer apellido, segundo apellido y especialidad. 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" para el tipo de objetos "Profesor" que permita obtener su nombre completo con el formato "apellidos nombre".

respuesta:

create or replace type Profesor under miembroescolar(--hereda de miembroEscolar

especialidad VARCHAR2(20),

antiguedad INTEGER,

-- definicion el Método constructor para el tipo de objetos "Profesor", en el que se indiquen

--como parámetros el código, nombre, primer apellido, segundo apellido y especialidad.

--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.

constructor function Profesor (codigo integer , nombre varchar2, primer\_apellido varchar2,

segundo\_apellido varchar2, especialidad varchar2)

--devuelve los parametros del objeto asignado los parametros pasados en el constructor

return self as result,

member function getNombreCompleto return varchar2--definicion delMétodo "getNombreCompleto" para el tipo de objetos

--"Profesor" que permita obtener su nombre completo con el formato "apellidos nombre".

);

/

---------------------------------------------------------------------------------------------------------------------------------------------------------------

create or replace TYPE BODY PROFESOR AS

constructor function Profesor (codigo integer , nombre varchar2, primer\_apellido varchar2,

segundo\_apellido varchar2, especialidad varchar2)

--devuelve los parametros del objeto asignado los parametros pasados en el constructor

return self as result

is

BEGIN

--parte de declaracion del metodo constructor

--asignamos cada parametro del constructor a nuestro objeto con self

self.codigo :=codigo;

self.nombre :=nombre;

self.apellidos := primer\_apellido ||' ' || segundo\_apellido;--concatenamos los apellidos para formar

--el parametro apellidos

self.especialidad := especialidad;

return ;--devuelve los parametros asignados al objeto atraves del constructor

END ;--fin d constructor creado

member function getNombreCompleto return varchar2--declaracion de Método "getNombreCompleto" para el tipo de objetos

--"Profesor" que permita obtener su nombre completo con el formato "apellidos nombre".

is

v\_nombreCompleto varchar2(60);--declaracion de variable que recivira el resutado a devolver

begin

v\_nombreCompleto:= self.apellidos||' '||self.nombre;

return v\_nombreCompleto;

end getNombreCompleto;

END;

/

3. Crea el tipo de objeto "Cursos" con los siguientes atributos y métodos:

codigo INTEGER,

nombre VARCHAR2(20),

refProfe REF Profesor,

max\_Alumn INTEGER,

fecha\_Inic DATE,

fecha\_Fin DATE,

num\_Horas INTEGER

--Método MAP "ordenarCursos" para el tipo "Cursos". Este método debe retornar el nombre completo del profesor al que hace

--referencia cada curso. Para obtener el nombre debes utilizar el método getNombreCompleto que se ha creado anteriormente.

--repuesta:

create or replace type curso as object(

codigo INTEGER,

nombre VARCHAR2(20),

refProfe REF Profesor,

max\_Alumn INTEGER,

fecha\_Inic DATE,

fecha\_Fin DATE,

num\_Horas INTEGER,

map member function ordenarCursos return varchar2--Método MAP "ordenarCursos" para el tipo "Cursos". Este método

--debe retornar el nombre completo del profesor al que hace referencia cada curso. Para obtener el nombre debes

--utilizar el método getNombreCompleto que se ha creado anteriormente.

);

/

---------------------------------------------------------------------------------------------------------------------------------------------------------

create or replace TYPE BODY CURSO AS

map member function ordenarCursos return varchar2 is

v\_profe\_curso ref profesor;

v\_curso curso;

v\_profesor profesor;

BEGIN

select ref(p) into v\_profe\_curso from profesorados p where codigo=3;--falta criterio comparacion

--v\_curso := new curso(1,'curso primero',v\_profe\_curso ,200,'12/12/12','14/12/13',100);

select deref(v\_profe\_curso) into v\_profesor from dual;

dbms\_output.put\_line(v\_profesor.getNombreCompleto);

RETURN v\_profesor.getNombreCompleto();

END ordenarCursos;

END;

/

--4. Crea, como tipo heredado de "MiembroEscolar", el tipo de objeto "Alumno" con los siguientes atributos:

--cursoAlumno Cursos

--respuesta:

create or replace type alumno under miembroEscolar(

cursoAlumno Curso

);

/

--5. Crea un tipo de tabla "Profesorado" de objetos "Profesor".

--respuesta:

create table of profesorados(

primary key(codigo)

);

--6. Inserta en dicha tabla dos objetos "Profesor". El primero de ellos con los datos:

--El segundo objeto "Profesor" debes crearlo usando el método constructor que has realizado anteriormente. Debes usar los

--siguientes datos:

--respuesta:

insert into profesorados values('2','51083099F','MARIA LUISA','FABRE BERDUN','F','31/03/1975','TECNOLOGIA',4);

insert into profesorados values(profesor('12','JAVIER','JIMOMNEZ','HENANDO','LENGUA'));

--7. Crea un tipo colección VARRAY llamada "ListaCursos" en la que se puedan almacenar hasta 10 objetos "Cursos".

--respuesta:

create or replace type ListaCursos is varray(10) of curso;

--8. Crea una tabla "Alumnado" de objetos "Alumno".

--respuesta:

create or replace type alumno under miembroEscolar(

cursoAlumno Curso

);

--9. Crea un bloque de código que haga todo lo siguiente:

--a) Guarda en una instancia "listaCursos1" de dicha lista, los dos cursos siguientes:

--b) Inserta en la tabla "Alumnado" las siguientes filas:

--c) Obtener, de la tabla "Alumnado", el alumno que tiene el código 100, asignándoselo a una variable "unAlumno".`

--d) Modifica el código del alumno guardado en esa variable "unAlumno" asignando el valor 101, y su curso debe ser el segundo

--que se había creado anteriormente. Inserta ese alumno en la tabla "Alumnado".

--reapuesta:

--a) Guarda en una instancia "listaCursos1" de dicha lista, los dos cursos siguientes:

declare

vListaCursos1 listacursos;

v\_profe\_curso1 ref profesor;

v\_profe\_curso2 ref profesor;

v\_profesor profesor;

vcurso1 curso;

vcurso2 curso;

--variables para manipular los datos del alumno apartado 9 b

refAlumno ref alumno;

unAlumno alumno;

begin

select ref(p) into v\_profe\_curso1 from profesorados p where codigo=3;--obtiene la referencioa del profesor con codigo 3

vcurso1 :=new curso(1,'curso 1',v\_profe\_curso1,20,'1/6/2011','30/6/2011',30);--instancia del curso1

--El segundo curso que debes almacenar en dicha lista debe tener los siguientes datos:

select ref(p) into v\_profe\_curso2 from profesorados p where dni='51083099F';--obtiene la referencioa del profesor con codigo 3

vcurso2 := new curso(2,'curso 2',v\_profe\_curso2,20,'1/6/2011','30/6/2011',30);--instancia del curso2

vlistacursos1:= new listacursos(vcurso1,vcurso2);--instancia de la lista cursos 1 con los cursos añadidos

dbms\_output.put\_line(vlistacursos1(1).codigo);--salida por consola de los datos

--b) Inserta en la tabla "Alumnado" las siguientes filas:

--las filas deben añadirse de una en una para lo cual se comenta la primera tras añadirla.

--insert into alumnos values('100','76401092Z','MANUEL','SUAREZ IBAÑEZ','M','30/6/1990',vlistacursos1(1));

--se comenta la linia para no provocar fallo de duplicidad al ejecutarla dos veces

--insert into alumnos values('102','6915588V','MILAGROSA','DIAZ PEREZ','F','28/10/1984',vlistacursos1(2));

select ref(a) into refAlumno from alumnos a where a.codigo='100';--obtenemos la referencia del alumno

select deref(refAlumno) into unAlumno from dual;--con la referencia del alumno obtenemos el alumno

DBMS\_OUTPUT.PUT\_LINE(unalumno.nombre);--nombre el alumno obtenido

--d) Modifica el código del alumno guardado en esa variable "unAlumno" asignando el valor 101, y su curso debe

--ser el segundo que se había creado anteriormente. Inserta ese alumno en la tabla "Alumnado".

unAlumno.codigo := '101';

DBMS\_OUTPUT.PUT\_LINE(unalumno.codigo);

--codigo para realizar el apartado c

--c) Obtener, de la tabla "Alumnado", el alumno que tiene el código 100, asignándoselo a una variable "unAlumno

--cambiamos el curso del primero de la lista cursos por el segundo

unalumno.cursoalumno := vlistacursos1(2);

DBMS\_OUTPUT.PUT\_LINE(unalumno.cursoalumno.codigo);

--Inserta ese alumno en la tabla "Alumnado".

insert into alumnos values(unalumno);

end;

/

--10. Realiza una consulta de la tabla "Alumnado" ordenada por "cursoAlumno" para comprobar el funcionamiento del método

--MAP.

--respuesta:

select \* from alumnos order by cursoAlumno;