```
-- Los ejercicios se basan en un Sistema de Información
de un Instituto

DROP TABLE Alumnado;
DROP TABLE Profesorado;
DROP TYPE Alumno;
DROP TYPE ListaCursos;
DROP TYPE Cursos;
DROP TYPE Profesor;
-------------------------------------------------------
-------------------------------------
--  1. Creamos los siguientes elementos:
-------------------------------------------------------
-------------------------------------
--  Tipo de objetos "MiembroEscolar" con los atributos:
--          codigo          INTEGER
--          dni             VARCHAR2( 10 )
--          nombre          VARCHAR2( 30 )
--          apellidos       VARCHAR2( 30 )
--          sexo            VARCHAR2(  1 )
--          fecha_nac       DATE
-------------------------------------------------------
-------------------------------------
CREATE TABLE MiembroEscolar (
    codigo INT,
    dni VARCHAR(10),
    nombre VARCHAR(30),
    apellidos VARCHAR(30),
    sexo VARCHAR(1),
    fecha_nac DATE,
    PRIMARY KEY (codigo)
);


-------------------------------------------------------
-------------------------------------
--  Tipo heredado de "MiembroEscolar", el tipo de
```

```
objeto "Profesor" con los atributos:
--          especialidad    VARCHAR2( 20 )
--          antiguedad      INTEGER
-------------------------------------------------------
-------------------------------------
CREATE TABLE Profesor (
    codigo INT,
    especialidad VARCHAR(20),
    antiguedad INT,
    PRIMARY KEY (codigo),
    FOREIGN KEY (codigo) REFERENCES
MiembroEscolar(codigo)
);

-------------------------------------------------------
-------------------------------------
--  Tipo de objeto "Cursos" con los atributos:
--          codigo          INTEGER
--          nombre          VARCHAR2( 20 )
--          refProfe        REF Profesor
--          max_Alum        INTEGER
--          fecha_Inic      DATE
--          fecha_Fin       DATE
--          num_Horas       INTEGER
-------------------------------------------------------
-------------------------------------
CREATE TABLE Cursos (
    codigo INT,
    nombre VARCHAR(20),
    refProfe INT,
    max_Alum INT,
    fecha_Inic DATE,
    fecha_Fin DATE,
    num_Horas INT,
    PRIMARY KEY (codigo),
    FOREIGN KEY (refProfe) REFERENCES Profesor(codigo)
);
```

```
-------------------------------------------------------
-------------------------------------
--  Tipo heredado de "MiembroEscolar", el tipo de
objeto "Alumno" con los atributos:
--          zonaAlumno   Cursos
-------------------------------------------------------
-------------------------------------
CREATE TABLE Alumno (
    codigo INT,
    zonaAlumno INT,
    PRIMARY KEY (codigo),
    FOREIGN KEY (codigo) REFERENCES
MiembroEscolar(codigo),
    FOREIGN KEY (zonaAlumno) REFERENCES Cursos(codigo)
);

-------------------------------------------------------
-------------------------------------
--  2. Creamos un método constructor para el tipo de
objetos "Profesor" con los parámetros código, nombre,
primer apellido, segundo apellido y especialidad.
--     Este método debe asignar al atributo apellidos
los datos del primer apellido y segundo apellido que se
han pasado como parámetros, uniéndolos con un espacio
entre ellos.
-------------------------------------------------------
-------------------------------------
-- Un comportamiento similar requiere la creación de un
procedimiento almacenado para insertar datos en la
tabla "Profesor"

DELIMITER //
CREATE PROCEDURE insertar_profesor(
    IN p_codigo INT,
    IN p_nombre VARCHAR(30),
```

```
    IN p_apellido1 VARCHAR(30),
    IN p_apellido2 VARCHAR(30),
    IN p_especialidad VARCHAR(20)
)
BEGIN
    DECLARE p_apellidos VARCHAR(60);
    SET p_apellidos = CONCAT(p_apellido1, ' ',
p_apellido2);

    INSERT INTO Profesor (codigo, nombre, apellidos,
especialidad)
    VALUES (p_codigo, p_nombre, p_apellidos,
p_especialidad);
END //
DELIMITER ;



------------------------------------------------------
-------------------------------------
--  3. Creamos un método getNombreCompleto para el tipo
de objetos "Profesor", que permita obtener el nombre
completo con el formato "apellidos nombre".
------------------------------------------------------
-------------------------------------
-- Un comportamiento similar requiere la utilización de
funciones definidas por el usuario

DELIMITER //

CREATE FUNCTION getNombreCompleto(
    p_apellidos VARCHAR(60),
    p_nombre VARCHAR(30)
)
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
```

```
    DECLARE nombre_completo VARCHAR(100);
    SET nombre_completo = CONCAT(p_apellidos, ' ',
p_nombre);
    RETURN nombre_completo;
END //

DELIMITER ;


-----------------------------------------------------
-------------------------------------
--  4. Creamos una tabla "Profesorado" de objetos
"Profesor".
-----------------------------------------------------
-------------------------------------
-- Un comportamiento similar se realiza con el CREATE
TABLE ... LIKE
CREATE TABLE Profesorado LIKE Profesor;

-----------------------------------------------------
-------------------------------------
--     Insertamos estos dos objetos "Profesor":
-----------------------------------------------------
-------------------------------------
--          codigo:             2
--          dni:                51083099F
--          nombre:             MARÍA LUISA
--          apellidos:          FABRE BERDÚN
--          sexo:               F
--          fecha_nac:          31/03/1975
--          especialidad:       TECNOLOGÍA
--          antiguedad:         4
-----------------------------------------------------
-------------------------------------
-- Insertar el primer profesor
INSERT INTO Profesorado (codigo, dni, nombre,
apellidos, sexo, fecha_nac, especialidad, antiguedad)
```

```
VALUES (2, '51083099F', 'MARÍA LUISA', 'FABRE BERDÚN',
'F', '1975-03-31', 'TECNOLOGÍA', 4);

-------------------------------------------------------
-------------------------------------
-- El segundo objeto "Profesor" se debe crear usando el
método constructor creado previamente y con estos
datos:
-------------------------------------------------------
-------------------------------------
--          codigo:             3
--          nombre:             JAVIER
--          apellidos:          JARAMILLO HERNÁNDEZ
--          especialidad:       LENGUA
-------------------------------------------------------
-------------------------------------
-- Insertar el segundo profesor utilizando los datos
proporcionados
INSERT INTO Profesorado (codigo, nombre, apellidos,
especialidad)
VALUES (3, 'JAVIER', 'JARAMILLO HERNÁNDEZ', 'LENGUA');


-------------------------------------------------------
-------------------------------------
--  5. Creamos una colección VARRAY llamada
"ListaCursos" en la que se puedan almacenar hasta 10
objetos "Cursos".
-------------------------------------------------------
-------------------------------------
-- En Mysql no hay el tipo de dato VARRAY como en
Oracle; sin embargo, se puede lograr un comportamiento
similar
-- utilizando tablas normales y estableciendo
relaciones entre ellas.
CREATE TABLE ListaCursos (
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    profesor_id INT,
    codigo_curso INT,
    FOREIGN KEY (profesor_id) REFERENCES
Profesorado(codigo),
    FOREIGN KEY (codigo_curso) REFERENCES
Cursos(codigo)
);


-------------------------------------------------------
-------------------------------------
--     Guardamos en una instancia "listaCursos1" de
dicha lista, dos cursos:
-------------------------------------------------------
-------------------------------------
--          codigo:             1
--          nombre:             Curso 1
--          refResponsable:     Referencia al
responsable cuyo código es 3
--          max_Alumn:          20
--          fecha_Inic:         1/6/2011
--          fecha_Fin:          30/6/2011
--          num_Horas:          30
-------------------------------------------------------
-------------------------------------
--          codigo:             2
--          nombre:             Curso 2
--          refResponsable:     Referencia al
responsable cuyo DNI sea 51083099F
--          max_Alumn:          20
--          fecha_Inic:         1/6/2011
--          fecha_Fin:          30/6/2011
--          num_Horas:          30
-------------------------------------------------------
-------------------------------------
CREATE TABLE Cursos (
    codigo INT PRIMARY KEY,
```

```
    nombre VARCHAR(20),
    max_Alum INT,
    fecha_Inic DATE,
    fecha_Fin DATE,
    num_Horas INT
);

CREATE TABLE Profesor_Cursos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    profesor_id INT,
    curso_id INT,
    FOREIGN KEY (profesor_id) REFERENCES
Profesorado(codigo),
    FOREIGN KEY (curso_id) REFERENCES Cursos(codigo)
);

-- Insertar el primer curso y relacionarlo con el
profesor cuyo código es 3
INSERT INTO Cursos (codigo, nombre, max_Alum,
fecha_Inic, fecha_Fin, num_Horas)
VALUES (1, 'Curso 1', 20, '2011-06-01', '2011-06-30',
30);

INSERT INTO Profesor_Cursos (profesor_id, curso_id)
VALUES (3, 1);

-- Insertar el segundo curso y relacionarlo con el
profesor cuyo DNI es 51083099F
INSERT INTO Cursos (codigo, nombre, max_Alum,
fecha_Inic, fecha_Fin, num_Horas)
VALUES (2, 'Curso 2', 20, '2011-06-01', '2011-06-30',
30);

INSERT INTO Profesor_Cursos (profesor_id, curso_id)
SELECT codigo, 2 FROM Profesorado WHERE dni =
'51083099F';
```

```
-- Insertar los dos cursos en la tabla ListaCursos y
asociarlos con los respectivos profesores
INSERT INTO ListaCursos (profesor_id, codigo_curso)
VALUES
(3, 1), -- Asociar el curso 1 al profesor con código 3
(
    SELECT p.codigo
    FROM Profesorado p
    WHERE p.dni = '51083099F'
), -- Asociar el curso 2 al profesor con DNI 51083099F
2; -- Código del curso 2

-----------------------------------------------------
-------------------------------------
--  6. Creamos una tabla "Alumnado" de objetos
"Alumno".
-----------------------------------------------------
-------------------------------------
-- Un comportamiento similar se realiza con el CREATE
TABLE ... LIKE
CREATE TABLE Alumnado LIKE Alumno;


-----------------------------------------------------
-------------------------------------
--     Insertamos las siguientes filas:
-----------------------------------------------------
-------------------------------------
--          codigo:             100
--          dni:                76401092Z
--          nombre:             MANUEL
--          apellidos:          SUÁREZ IBÁÑEZ
--          sexo:               M
--          fecha_nac:          30/3/1990
--          cursoAlumno:        objeto creado
anteriormente para el primer curso
-----------------------------------------------------
```

```
-------------------------------------
-- Insertar el primer alumno
INSERT INTO Alumnado (codigo, dni, nombre, apellidos,
sexo, fecha_nac, cursoAlumno)
VALUES (100, '76401092Z', 'MANUEL', 'SUÁREZ IBÁÑEZ',
'M', '1990-03-30', 1); -- Asociado al primer curso

-------------------------------------------------------
-------------------------------------
--          codigo:             102
--          dni:                6915588V
--          nombre:             MILAGROSA
--          apellidos:          DÍAZ PÉREZ
--          sexo:               F
--          fecha_nac:          28/10/1984
--          cursoAlumno:        objeto que se encuentra
en la segunda posición de "listaCursos1 (debe tomarse
de la lista)
-------------------------------------------------------
-------------------------------------
-- Insertar el segundo alumno
INSERT INTO Alumnado (codigo, dni, nombre, apellidos,
sexo, fecha_nac, cursoAlumno)
VALUES (102, '6915588V', 'MILAGROSA', 'DÍAZ PÉREZ',
'F', '1984-10-28', 2); -- Asociado al segundo curso


-------------------------------------------------------
-------------------------------------
--  7. Obtenemos de la tabla "Alumnado", el alumno que
tiene el codigo 100,
--     asignándoselo a una variable "unAlumnado".
-------------------------------------------------------
-------------------------------------
-- Declarar variables para almacenar los datos del
alumno
DECLARE unAlumnado_nombre VARCHAR(30);
```

```
DECLARE unAlumnado_apellidos VARCHAR(60);
DECLARE unAlumnado_sexo VARCHAR(1);
DECLARE unAlumnado_fecha_nac DATE;
DECLARE unAlumnado_curso INT; -- Aquí puedes ajustar el
tipo de datos según la estructura de tu tabla Alumnado

-- Realizar la consulta para obtener el alumno con
código 100 y asignar los resultados a las variables
SELECT nombre, apellidos, sexo, fecha_nac, cursoAlumno
INTO unAlumnado_nombre, unAlumnado_apellidos,
unAlumnado_sexo, unAlumnado_fecha_nac, unAlumnado_curso
FROM Alumnado
WHERE codigo = 100;


-----------------------------------------------------
-------------------------------------
--  8. Modificamos el código del Alumno guardado en la
variable "unAlumno", asignando
--     un valor 101, y su curso debe ser el segundo que
se creó anteriormente.
-----------------------------------------------------
-------------------------------------


-----------------------------------------------------
-------------------------------------
--     Insertamos ese alumno en la tabla "Alumnado".
-----------------------------------------------------
-------------------------------------
-- Insertar el alumno modificado en la tabla Alumnado
INSERT INTO Alumnado VALUES (unAlumno);


-----------------------------------------------------
-------------------------------------
--  9. Creamos un método MAP "ordenarCursos" para el
```

```
tipo "Cursos". Este método debe
--     retornar el nombre completo del profesor al que
hace referencia cada curso.
-----------------------------------------------------
-------------------------------------
CREATE FUNCTION ordenarCursos() RETURNS VARCHAR(255)
BEGIN
    DECLARE resultado VARCHAR(255);
    SELECT GROUP_CONCAT(
        CONCAT(
            'Curso ', codigo, ': Profesor ',
obtener_nombre_profesor(refProfe)
        )
        ORDER BY codigo
        SEPARATOR '; '
    ) INTO resultado
    FROM Cursos;
    RETURN resultado;
END;


-----------------------------------------------------
-------------------------------------
--     Para obtener el nombre se debe utilizar el
método getNombreCompleto creado previamente.
-----------------------------------------------------
-------------------------------------
CREATE FUNCTION obtener_nombre_profesor(profesor_id
INT) RETURNS VARCHAR(100)
BEGIN
    DECLARE nombre_completo VARCHAR(100);
    SELECT CONCAT(apellidos, ', ', nombre) INTO
nombre_completo FROM Profesorado WHERE codigo =
profesor_id;
    RETURN nombre_completo;
END;
```

```
-------------------------------------------------------
-------------------------------------
-- 10. Realizamos una consulta en la tabla "Alumnado"
ordenada por "cursoAlumno" para
--     comprobar el funcionamiento del método MAP.
-------------------------------------------------------
-------------------------------------
#SELECT * FROM Alumnado ORDER BY cursoAlumno;
SELECT *,
       ordenarCursos() AS cursos_profesores
FROM Alumnado
ORDER BY cursoAlumno;
```