

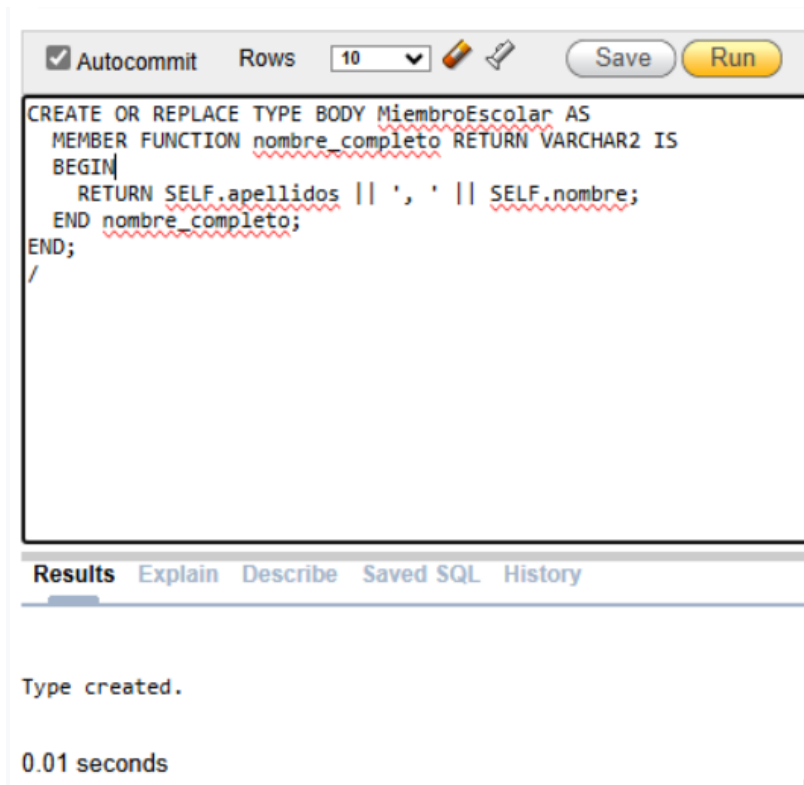
Tarea para BD07.

Detalles de la tarea de esta unidad.

Enunciado.

1. Crea el tipo de objetos MiembroEscolar con los siguientes **atributos**: codigo (nº entero), dni, nombre, apellidos y fecha de nacimiento y un **método** que permita obtener el nombre completo con el formato apellidos, nombre (ejemplo: García López, José Juan)

```
CREATE OR REPLACE TYPE MiembroEscolar AS OBJECT (  
    codigo NUMBER,  
    dni VARCHAR2(20),  
    nombre VARCHAR2(50),  
    apellidos VARCHAR2(100),  
    fecha_nacimiento DATE,  
    MEMBER FUNCTION nombre_completo RETURN VARCHAR2  
) NOT FINAL;  
/  
  
CREATE OR REPLACE TYPE BODY MiembroEscolar AS  
    MEMBER FUNCTION nombre_completo RETURN VARCHAR2 IS  
    BEGIN  
        RETURN SELF.apellidos || ', ' || SELF.nombre;  
    END nombre_completo;  
END;  
/
```



The screenshot shows the Oracle SQL Developer interface. At the top, there is a toolbar with 'Autocommit' checked, 'Rows' set to 10, and 'Save' and 'Run' buttons. The main text area contains the following SQL code:

```
CREATE OR REPLACE TYPE BODY MiembroEscolar AS
  MEMBER FUNCTION nombre_completo RETURN VARCHAR2 IS
  BEGIN
    RETURN SELF.apellidos || ', ' || SELF.nombre;
  END nombre_completo;
END;
/
```

Below the text area, there is a tabbed interface with 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying the message 'Type created.' and the execution time '0.01 seconds'.

2. Crea, como tipo heredado de `MiembroEscolar`, el tipo de objeto `Profesor` con los siguientes **atributos**: especialidad y un **método constructor** con cinco parámetros (código, nombre, primer apellido, segundo apellido y especialidad).

```
CREATE OR REPLACE TYPE Profesor UNDER MiembroEscolar (
  especialidad VARCHAR2(50),
  CONSTRUCTOR FUNCTION Profesor(
    p_codigo NUMBER,
    p_nombre VARCHAR2,
    p_apellido1 VARCHAR2,
    p_apellido2 VARCHAR2,
    p_especialidad VARCHAR2
  ) RETURN SELF AS RESULT
);
/
```

```
CREATE OR REPLACE TYPE BODY Profesor AS
  CONSTRUCTOR FUNCTION Profesor(
    p_codigo NUMBER,
    p_nombre VARCHAR2,
    p_apellido1 VARCHAR2,
    p_apellido2 VARCHAR2,
    p_especialidad VARCHAR2
  ) RETURN SELF AS RESULT IS
  BEGIN
```

```
    SELF.codigo := p_codigo;
    SELF.nombre := p_nombre;
    SELF.apellidos := p_apellido1 || ' ' || p_apellido2;
    SELF.especialidad := p_especialidad;
    SELF.dni := NULL;
    SELF.fecha_nacimiento := NULL;
    RETURN;
END Profesor;
END;
/
```

3. Crea la tabla Tprofesores de objetos Profesor

```
CREATE TABLE Tprofesores OF Profesor;
```

inserta en dicha tabla varios objetos profesor usando tanto el constructor por defecto

Constructor por defecto

```
INSERT INTO Tprofesores VALUES (
    101,
    '11111111A',
    'Juan José',
    'García López',
    TO_DATE('1980-05-15', 'YYYY-MM-DD'),
    'Matemáticas'
);
```

```
INSERT INTO Tprofesores VALUES (
    102,
    '22222222B',
    'Ana',
    'Martín Sanz',
    TO_DATE('1975-11-20', 'YYYY-MM-DD'),
    'Lengua'
);
```

como el método constructor realizado anteriormente.

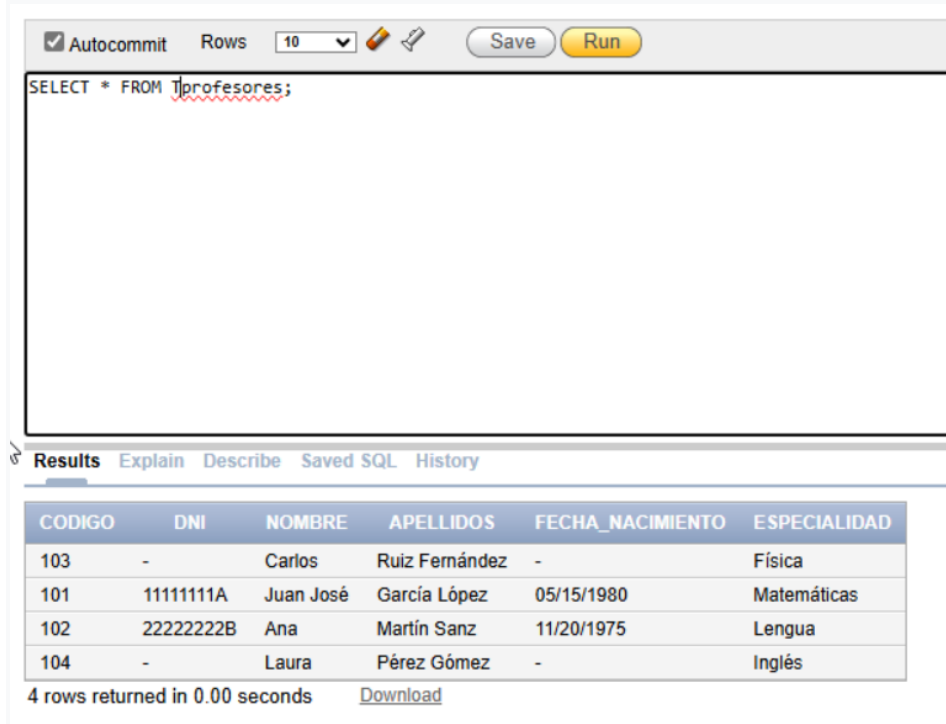
Constructor explícito

```
INSERT INTO Tprofesores VALUES (  
  Profesor(103, 'Carlos', 'Ruiz', 'Fernández', 'Física')  
);
```

```
INSERT INTO Tprofesores VALUES (  
  Profesor(104, 'Laura', 'Pérez', 'Gómez', 'Inglés')  
);*
```

Después haz un select de toda la tabla y copia el resultado.

```
SELECT * FROM Tprofesores;
```



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with 'Autocommit' checked, 'Rows' set to 10, and 'Save' and 'Run' buttons. Below the toolbar, the SQL query 'SELECT * FROM Tprofesores;' is entered in the editor. The 'Run' button is highlighted. Below the editor, the 'Results' tab is selected, showing a table with 4 rows and 6 columns: CODIGO, DNI, NOMBRE, APELLIDOS, FECHA_NACIMIENTO, and ESPECIALIDAD. The table contains the following data:

CODIGO	DNI	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	ESPECIALIDAD
103	-	Carlos	Ruiz Fernández	-	Física
101	11111111A	Juan José	García López	05/15/1980	Matemáticas
102	22222222B	Ana	Martín Sanz	11/20/1975	Lengua
104	-	Laura	Pérez Gómez	-	Inglés

Below the table, it says '4 rows returned in 0.00 seconds' and there is a 'Download' link.

4. Crea el tipo de objeto Grupo con los **atributos**: codigo (nº entero), nombre, refTutor (referencia un objeto profesor) y nº máximo de alumnos, y con un **método MAP** que debe retornar el nombre completo del profesor al que hace referencia cada grupo (utilizar el método de MiembroEscolar que obtiene el nombre completo).

```
CREATE OR REPLACE TYPE Grupo AS OBJECT (  
    codigo NUMBER,  
    nombre VARCHAR2(50),  
    refTutor REF Profesor,  
    max_alumnos NUMBER,  
    MAP MEMBER FUNCTION get_nombre_profesor RETURN VARCHAR2  
);  
/
```

```
CREATE OR REPLACE TYPE BODY Grupo AS  
  
    MAP MEMBER FUNCTION get_nombre_profesor RETURN VARCHAR2 IS  
        v_tutor Profesor;  
BEGIN  
    SELECT Deref(self.refTutor) INTO v_tutor FROM dual;  
  
    IF v_tutor IS NULL THEN  
        RETURN NULL;  
    ELSE  
        RETURN v_tutor.nombre_completo();  
    END IF;  
EXCEPTION  
    WHEN OTHERS THEN  
  
        RETURN NULL;  
END get_nombre_profesor;  
  
END;
```

5. Crea las tablas Tgrupos de objetos GRUPO

```
CREATE TABLE Tgrupos OF Grupo;
```

inserta varios grupos con la referencia al tutor que consideres para ellos (obtener cada uno usando un select de los almacenados en la tabla Tprofesores).

```
INSERT INTO Tgrupos (codigo, nombre, refTutor, max_alumnos) VALUES (  
    1,  
    '1º ESO A',  
    (SELECT REF(p) FROM TProfesores p WHERE p.codigo = 101), -- REF al  
profesor 101  
    25  
);
```

```
INSERT INTO Tgrupos (codigo, nombre, refTutor, max_alumnos) VALUES (  
    2,  
    '2º ESO B',  
    (SELECT REF(p) FROM TProfesores p WHERE p.codigo = 102), -- REF al  
profesor 102  
    30  
);
```

```
INSERT INTO Tgrupos (codigo, nombre, refTutor, max_alumnos) VALUES (  
    3,  
    '1º BACH C',  
    (SELECT REF(p) FROM TProfesores p WHERE p.codigo = 103), -- REF al  
profesor 103  
    20  
);
```

```
INSERT INTO Tgrupos (codigo, nombre, refTutor, max_alumnos) VALUES (  
    4,  
    '2º BACH A',  
    NULL, -- Sin tutor asignado  
    22  
);
```

Después haz un select de la tabla con todos los atributos (incluidos los atributos del profesor referenciado) y copia aquí el resultado de la consulta.

```
SELECT g.codigo AS grupo_codigo,  
       g.nombre AS grupo_nombre,  
       g.max_alumnos,  
       Deref(g.refTutor).codigo AS tutor_codigo,
```

```

    Deref(g.refTutor).nombre AS tutor_nombre,

    Deref(g.refTutor).apellidos AS tutor_apellidos,

    Deref(g.refTutor).especialidad AS tutor_especialidad

FROM TGrupos g;

```

<input checked="" type="checkbox"/> Autocommit	Rows	10			Save	Run
--	------	----	--	--	------	-----

```

SELECT g.codigo AS grupo_codigo,
       g.nombre AS grupo_nombre,
       g.max_alumnos,
       Deref(g.refTutor).codigo AS tutor_codigo,
       Deref(g.refTutor).nombre AS tutor_nombre,
       Deref(g.refTutor).apellidos AS tutor_apellidos,
       Deref(g.refTutor).especialidad AS tutor_especialidad
FROM TGrupos g;

```

Results Explain Describe Saved SQL History

GRUPO_CODIGO	GRUPO_NOMBRE	MAX_ALUMNOS	TUTOR_CODIGO	TUTOR_NOMBRE	TUTOR_APELLIDOS	TUTOR_ESPECIALIDAD
4	2º BACH A	22	-	-	-	-
1	1º ESO A	25	101	Juan José	García López	Matemáticas
2	2º ESO B	30	102	Ana	Martín Sanz	Lengua
3	1º BACH C	20	103	Carlos	Ruiz Fernández	Física
5	1º ESO B	26	101	Juan José	García López	Matemáticas

5 rows returned in 0.00 seconds Download

Copié mal la

6. Crea, como tipo heredado de MiembroEscolar, el tipo de objeto Alumno con el atributo grupoAlumno (que será un objeto de Grupo).

```
CREATE OR REPLACE TYPE Alumno UNDER MiembroEscolar (
```

```
    grupoAlumno Grupo
```

```
);
```

```
/
```

Crea una tabla TAlumnos de objetos Alumno.

```
CREATE TABLE TAlumnos OF Alumno (
```

```
    dni PRIMARY KEY
```

```
);
```

```
/
```

Inserta varios alumnos con alguno de los grupos almacenados en la tabla Tgrupos.

```
INSERT INTO TAlumnos (dni, nombre, apellidos, fecha_nacimiento,
grupoAlumno) VALUES (
    '44444444D', 'Pedro', 'Gómez Sánchez', TO_DATE('2010-03-12', 'YYYY-MM-DD'),
    (SELECT VALUE(g) FROM Tgrupos g WHERE g.codigo = 1)
);
```

```
INSERT INTO TAlumnos (dni, nombre, apellidos, fecha_nacimiento,
grupoAlumno) VALUES (
    '55555555E', 'Lucía', 'Fernández Vidal', TO_DATE('2010-07-25', 'YYYY-MM-DD'),
    (SELECT VALUE(g) FROM Tgrupos g WHERE g.codigo = 1)
);
```

```
INSERT INTO TAlumnos (dni, nombre, apellidos, fecha_nacimiento,
grupoAlumno) VALUES (
    '66666666F', 'Marcos', 'Alonso Pérez', TO_DATE('2009-01-18', 'YYYY-MM-DD'),
    (SELECT VALUE(g) FROM Tgrupos g WHERE g.codigo = 2)
);
```

```
INSERT INTO TAlumnos (dni, nombre, apellidos, fecha_nacimiento,
grupoAlumno) VALUES (
    '77777777G', 'Sofía', 'Jiménez Ruiz', TO_DATE('2008-09-05', 'YYYY-MM-DD'),
    (SELECT VALUE(g) FROM Tgrupos g WHERE g.codigo = 3)
);
```

```
INSERT INTO TAlumnos (dni, nombre, apellidos, fecha_nacimiento,
grupoAlumno) VALUES (
    '88888888H', 'David', 'Moreno López', TO_DATE('2007-12-30', 'YYYY-MM-DD'),
    (SELECT VALUE(g) FROM Tgrupos g WHERE g.codigo = 4)
);
```



```
SELECT
  a.dni,
  a.nombre_completo() AS nombre_alumno,
  a.fecha_nacimiento,
  a.grupoAlumno.codigo AS codigo_grupo,
  a.grupoAlumno.nombre AS nombre_grupo
FROM TAlumnos a;
```

Results Explain Describe Saved SQL History

DNI	NOMBRE_ALUMNO	FECHA_NACIMIENTO	CODIGO_GRUPO	NOMBRE_GRUPO
44444444D	Gómez Sánchez, Pedro	03/12/2010	1	1º ESO A
55555555E	Fernández Vidal, Lucía	07/25/2010	1	1º ESO A
77777777G	Jiménez Ruiz, Sofía	09/05/2008	3	1º BACH C
88888888H	Moreno López, David	12/30/2007	4	2º BACH A
66666666F	Alonso Pérez, Marcos	01/18/2009	2	2º ESO B

5 rows returned in 0.01 seconds

[Download](#)

7. Select que muestre todos atributos (incluidos los del objeto grupoalumno) de la tabla TAlumnos ordenados por el atributo grupoAlumno y copia aquí el resultado.

```
SELECT
  a.dni,
  a.nombre,
  a.apellidos,
  a.fecha_nacimiento,
  a.grupoAlumno.codigo AS codigo_grupo,
  a.grupoAlumno.nombre AS nombre_grupo,
  a.grupoAlumno.max_alumnos AS max_alumnos_grupo,
  a.grupoAlumno.get_nombre_profesor() AS nombre_tutor
FROM TAlumnos a
ORDER BY a.grupoAlumno;
```

<input checked="" type="checkbox"/> Autocommit Rows: 10 Save Run							
<pre> SELECT a.dni, a.nombre, a.apellidos, a.fecha_nacimiento, a.grupoAlumno.codigo AS codigo_grupo, a.grupoAlumno.nombre AS nombre_grupo, a.grupoAlumno.max_alumnos AS max_alumnos_grupo, a.grupoAlumno.get_nombre_profesor() AS nombre_tutor FROM TAlumnos a ORDER BY a.grupoAlumno; </pre>							
Results Explain Describe Saved SQL History							
DNI	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	CODIGO_GRUPO	NOMBRE_GRUPO	MAX_ALUMNOS_GRUPO	NOMBRE_TUTOR
44444444D	Pedro	Gómez Sánchez	03/12/2010	1	1º ESO A	25	García López, Juan José
55555555E	Lucía	Fernández Vidal	07/25/2010	1	1º ESO A	25	García López, Juan José
66666666F	Marcos	Alonso Pérez	01/18/2009	2	2º ESO B	30	Martín Sanz, Ana
77777777G	Sofía	Jiménez Ruiz	09/05/2008	3	1º BACH C	20	Ruiz Fernández, Carlos
88888888H	David	Moreno López	12/30/2007	4	2º BACH A	22	-

Comprueba que se ordena correctamente según el método MAP del objeto grupo.

Los resultados están ordenados por el nombre del tutor, lo cual confirma que la Cláusula order by está usando la función MAP

Después modifica el nombre y/o apellidos a uno de los profesores y

UPDATE TProfesores p

SET p.apellidos = 'García Actualizado'

WHERE p.nombre = 'Juan José' AND p.apellidos = 'García López';

Ejecuta de nuevo el SELECT para comprobar que funciona el método MAP.

```

SELECT
  a.dni,
  a.nombre,
  a.apellidos,
  a.fecha_nacimiento,
  a.grupoAlumno.codigo AS codigo_grupo,
  a.grupoAlumno.nombre AS nombre_grupo,
  a.grupoAlumno.max_alumnos AS max_alumnos_grupo,
  a.grupoAlumno.get_nombre_profesor() AS nombre_tutor
FROM TAlumnos a
ORDER BY a.grupoAlumno;

```

Autocommit Rows 10 Save Run

```
SELECT
a.dni,
a.nombre,
a.apellidos,
a.fecha_nacimiento,
a.grupoAlumno.codigo AS codigo_grupo,
a.grupoAlumno.nombre AS nombre_grupo,
a.grupoAlumno.max_alumnos AS max_alumnos_grupo,
a.grupoAlumno.get_nombre_profesor() AS nombre_tutor
FROM TAlumnos a
ORDER BY a.grupoAlumno;
```

Results Explain Describe Saved SQL History

DNI	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	CODIGO_GRUPO	NOMBRE_GRUPO	MAX_ALUMNOS_GRUPO	NOMBRE_TUTOR
44444444D	Pedro	Gómez Sánchez	03/12/2010	1	1º ESO A	25	García Actualizado, Juan José
55555555E	Lucía	Fernández Vidal	07/25/2010	1	1º ESO A	25	García Actualizado, Juan José
66666666F	Marcos	Alonso Pérez	01/18/2009	2	2º ESO B	30	Martín Sanz, Ana
77777777G	Sofía	Jiménez Ruiz	09/05/2008	3	1º BACH C	20	Ruiz Fernández, Carlos
88888888H	David	Moreno López	12/30/2007	4	2º BACH A	22	-

5 rows returned in 0.01 seconds Download

No ha cambiado mucho, pero porque he hecho el test mal, voy a probar con otro UPDATE diferente

```
UPDATE TProfesores p
SET p.apellidos = 'Zamorano Sanz'
WHERE p.nombre = 'Ana' AND p.apellidos = 'Martín Sanz';
```

Ahora si que cambia el orden de Martín a Zamorano

Autocommit Rows 10 Save Run

```
SELECT
a.dni,
a.nombre,
a.apellidos,
a.fecha_nacimiento,
a.grupoAlumno.codigo AS codigo_grupo,
a.grupoAlumno.nombre AS nombre_grupo,
a.grupoAlumno.max_alumnos AS max_alumnos_grupo,
a.grupoAlumno.get_nombre_profesor() AS nombre_tutor
FROM TAlumnos a
ORDER BY a.grupoAlumno;
```

Results Explain Describe Saved SQL History

DNI	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	CODIGO_GRUPO	NOMBRE_GRUPO	MAX_ALUMNOS_GRUPO	NOMBRE_TUTOR
44444444D	Pedro	Gómez Sánchez	03/12/2010	1	1º ESO A	25	García Actualizado, Juan José
55555555E	Lucía	Fernández Vidal	07/25/2010	1	1º ESO A	25	García Actualizado, Juan José
77777777G	Sofía	Jiménez Ruiz	09/05/2008	3	1º BACH C	20	Ruiz Fernández, Carlos
66666666F	Marcos	Alonso Pérez	01/18/2009	2	2º ESO B	30	Zamorano Sanz, Ana
88888888H	David	Moreno López	12/30/2007	4	2º BACH A	22	-

5 rows returned in 0.00 seconds Download