

# BBDD OBJETO-RELACIONALES HERENCIA

Javier García-Retamero Redondo

# HERENCIA



# HERENCIA

## ¿QUÉ PODEMOS HACER EN LOS SUBTIPOS?

- Definir atributos propios
- Definir su propios métodos
- Redefinir los métodos que heredan

# SUPERTIPO



# DEFINIMOS EL SUPERTIPO

## DEFINIMOS EL TIPO

```
CREATE OR REPLACE TYPE TIPO_PERSONA AS OBJECT(  
    DNI VARCHAR2(10),  
    NOMBRE VARCHAR2(25),  
    FEC_NAC DATE,  
    MEMBER FUNCTION EDAD RETURN NUMBER,  
    FINAL MEMBER FUNCTION GET_DNI RETURN VARCHAR2,  
    MEMBER FUNCTION GET_NOMBRE RETURN VARCHAR2,  
    MEMBER PROCEDURE VER_DATOS  
) NOT FINAL;
```

No se puede redefinir este método

Se puede heredar de este objeto

# DEFINIMOS EL SUPERTIPO

## DEFINIMOS EL CUERPO DEL TIPO

```
CREATE OR REPLACE TYPE BODY TIPO_PERSONA AS
  MEMBER FUNCTION EDAD RETURN NUMBER IS
    ED NUMBER;
  BEGIN
    ED := TO_CHAR(SYSDATE, 'YYYY') - TO_CHAR(FEC_NAC, 'YYYY');
  RETURN ED;
  END;
  --
  FINAL MEMBER FUNCTION GET_DNI RETURN VARCHAR2 IS
  BEGIN
    RETURN DNI;
  END;
  --
```

# DEFINIMOS EL SUPERTIPO

```
MEMBER FUNCTION GET_NOMBRE RETURN VARCHAR2 IS
BEGIN
    RETURN NOMBRE;
END;
--
MEMBER PROCEDURE VER_DATOS IS
BEGIN
    DBMS_OUTPUT.PUT_LINE(DNI || '*' || NOMBRE || '*' || EDAD());
END;
--
END;
/
```

# SUB-TIPO





# DEFINIMOS EL SUB-TIPO

## DEFINIMOS EL TIPO

```
CREATE OR REPLACE TYPE TIPO_ALUMNO UNDER TIPO_PERSONA(  
    CURSO VARCHAR2(10),  
    NOTA_FINAL NUMBER,  
    MEMBER FUNCTION NOTA RETURN NUMBER,  
    OVERRIDING MEMBER PROCEDURE VER_DATOS  
);
```

Indicamos que es un subtipo  
de persona

Vamos a sobrecribir el  
método del padre

# DEFINIMOS EL SUB-TIPO

## DEFINIMOS EL CUERPO DEL TIPO

```
CREATE OR REPLACE TYPE BODY TIPO_ALUMNO AS
  MEMBER FUNCTION NOTA RETURN NUMBER IS
  BEGIN
    RETURN NOTA_FINAL;
  END;
  --
  OVERRIDING MEMBER PROCEDURE VER_DATOS IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE(CURSO || '*' || NOTA_FINAL);
  END;

END;
```

# EJEMPLOS DE USO



# EJEMPLO DE USO: PL/SQL

DECLARE

A1 TIPO\_ALUMNO := TIPO\_ALUMNO(NULL, NULL, NULL, NULL, NULL);

A2 TIPO\_ALUMNO := TIPO\_ALUMNO('871234533A', 'PEDRO', '12/12/1996', 'SEGUNDO', 7);

NOM A1.NOMBRE%TYPE;

DNI A1.DNI%TYPE;

NOTAF A1.NOTA\_FINAL%TYPE;

BEGIN

A1.NOTA\_FINAL := 8;

A1.CURSO := 'PRIMERO';

A1.NOMBRE := 'JUAN';

A1.FEC\_NAC := '20/10/1997';

A1.VER\_DATOS;

| TIPO_ALUMNO |              |
|-------------|--------------|
| DNI         | TIPO_PERSONA |
| NOMBRE      |              |
| FEC_NAC     |              |
| CURSO       | AÑADIDOS     |
| NOTA_FINAL  |              |

# EJEMPLO DE USO: PL/SQL

```
NOM := A2.GET_NOMBRE();  
DNI := A2.GET_DNI();  
NOTAF := A2.NOTA();
```

```
A2.VER_DATOS;  
DBMS_OUTPUT.PUT_LINE(A1.EDAD());  
DBMS_OUTPUT.PUT_LINE(A2.EDAD());  
END;
```

# EJEMPLO DE USO: SQL

```
CREATE TABLE TALUMNOS OF TIPO_ALUMNO (  
    DNI PRIMARY KEY  
);
```

```
INSERT INTO TALUMNOS VALUES ('871234533A', 'PEDRO', '12/12/1996', 'SEGUNDO', 7);  
INSERT INTO TALUMNOS VALUES ('809004534B', 'MANUEL', '12/12/1997', 'TERCERO', 8);
```

```
SELECT * FROM TALUMNOS;
```

```
SELECT DNI, NOMBRE, CURSO, NOTA_FINAL FROM TALUMNOS;
```

```
SELECT P.GET_DNI(), P.GET_NOMBRE(), P.EDAD(), P.NOTA()  
    FROM TALUMNOS P;
```