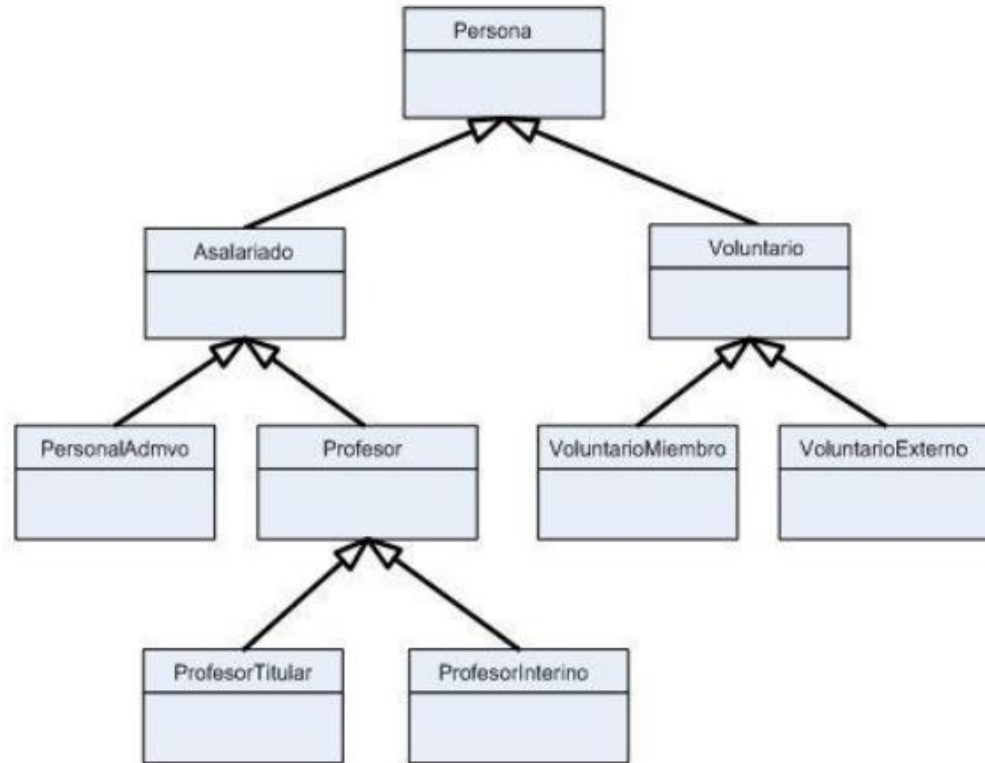




1. Herencia
2. Identificadores
3. Referencias

Herencia, identificadores y referencias

Herencia



Acciones que podemos realizar en un subtipo

- Añadir nuevos atributos que su supertipo no tenga.
- Añadir nuevos métodos que su supertipo no tenga.
- Añadir nuevos métodos con el mismo nombre que los métodos heredados del supertipo. Esto se denomina OVERLOADING.
- Modificar la implementación de algún método heredado de su supertipo. Esto se denomina OVERRIDING.

```
create or replace type person_t as object(  
  idno NUMBER,  
  name VARCHAR2(30),  
  surname1 VARCHAR2(50),  
  phone NUMBER,
```

```
  FINAL MAP MEMBER FUNCTION get_idno RETURN NUMBER,  
  MEMBER PROCEDURE show_complete_name ) NOT FINAL;
```

```
CREATE OR REPLACE TYPE BODY person_t AS  
  FINAL MAP MEMBER FUNCTION get_idno RETURN NUMBER IS  
    BEGIN  
      RETURN idno;  
    END;  
  MEMBER PROCEDURE show_complete_name IS  
    BEGIN  
      DBMS_OUTPUT.PUT_LINE('COMPLETE NAME: ' || name || ' ' || surname1);  
    END;  
END;
```

```
CREATE OR REPLACE TYPE empleado_t UNDER person_t(  
    surname2 VARCHAR2(50),  
    sueldo NUMBER,  
    OVERRIDING MEMBER PROCEDURE show_complete_name,  
    MEMBER PROCEDURE actualizar_sueldo (s NUMBER)  
);
```

```
CREATE OR REPLACE TYPE BODY empleado_t AS  
    OVERRIDING MEMBER PROCEDURE show_complete_name IS  
        BEGIN  
            DBMS_OUTPUT.PUT_LINE('COMPLETE NAME: ' || name || ' ' || surname1 || ' ' || surname2);  
        END;  
    MEMBER PROCEDURE actualizar_sueldo (s NUMBER) IS  
        BEGIN  
            sueldo := s;  
        END;  
    END;
```

```
SET serveroutput ON
DECLARE
persona person_t;
empleado empleado_t;
BEGIN
    persona:=person_t(1,'Esteban','Perez',668596213);
    empleado:= empleado_t(1, 'Pedro', 'Gonzalez', 670564123,'Perez',1500);
    persona.show_complete_name(); -- prints COMPLETE NAME: Esteban Perez
    empleado.show_complete_name(); -- prints COMPLETE NAME: Pedro Gonzalez Perez
END;
```

Tablas de objetos con herencia

```
CREATE TABLE person_object_table OF person_t (idno primary key );

INSERT INTO person_object_table VALUES(
person_t(18440523,'Esteban','Perez',685569213));
INSERT INTO person_object_table VALUES(
empleado_t(19854524,'Pedro', 'Gonzalez', 645569879,'Saez',1000));
SELECT p.get_idno() FROM person_object_table p;
```


Tablas de objetos con herencia

```
select * from person_object_table p;
```

	IDNO	NAME	SURNAME1	PHONE
1	18440523	Esteban	Perez	685569213
2	19854524	Pedro	Gonzalez	645569879

```
select * from person_object_table p where value(p) is of (empleado_t);
```

	IDNO	NAME	SURNAME1	PHONE
1	19854524	Pedro	Gonzalez	645569879

Identificadores

```
create type coche as object(marca varchar2(20), modelo varchar(20));  
create table vehiculos of coche;  
insert into vehiculos values (coche('SEAT','IBIZA'));  
select object_id, object_value from vehiculos;
```

OBJECT_ID	OBJECT_VALUE
1	821F56E3C56341D588FEE42BA6D29535 [SYS.COCHES]

Referencias

```
CREATE OR REPLACE TYPE EMPLOYEE_T AS OBJECT(  
  NAME CHAR(50),  
  MANAGER REF EMPLOYEE_T  
);
```

```
CREATE TABLE EMPLOYEES OF EMPLOYEE_T;  
INSERT INTO EMPLOYEES VALUES (EMPLOYEE_T('Pepe Moya',null));  
INSERT INTO EMPLOYEES VALUES (EMPLOYEE_T('Jaime Perez',  
(select REF(e) from employees e where e.name='Pepe Moya')));
```

SCOPED REFERENCES

- Se puede restringir que el contenido de una REF provenga únicamente de una tabla específica.

```
CREATE TABLE contacts_ref_table(  
  contact_ref REF person_t SCOPE IS person_object_table,  
  contact_date DATE);  
  
INSERT INTO contacts_ref_table VALUES ((SELECT REF(e) FROM person_object_table e WHERE e.idno=19854524), SYSDATE);
```

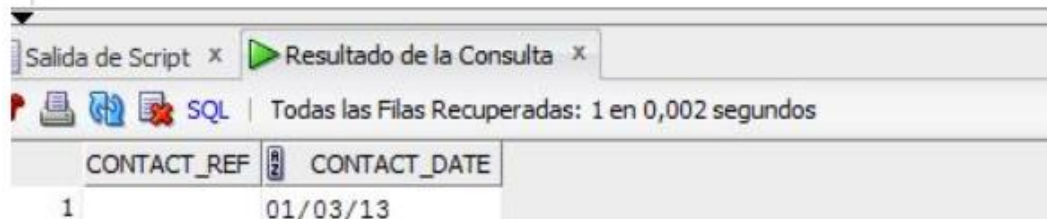
DANGLING REFERENCES

- El borrado de un objeto identificado como REF, hace que este objeto no este disponible.

```
CREATE TABLE contacts_ref(  
  contact_ref REF person_t,  
  contact_date DATE);
```

```
INSERT INTO contacts_ref VALUES ((SELECT REF(e) FROM person_object_table e WHERE e.idno=18440523),SYSDATE);
```

```
DELETE FROM person_object_table WHERE idno=19854524;  
SELECT * FROM contacts_ref_table C WHERE contact_ref IS DANGLING;
```



Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,002 segundos

CONTACT_REF	CONTACT_DATE
1	01/03/13


DEFERENCING REFERENCES

- Para acceder a un objeto identificado como REF, se denomina *dereferencing REF*.

```
select * from employees e;
```

Salida de Script x


Resultado de la Consulta x




 | Todas las Filas Recuperadas: 2 en 0,002 segundos

NAME	MANAGER
1 Pepe Moya	(null)
2 Jaime Perez	[SYS.EMPLOYEE_T]

DEFERENCING REFERENCES

```
select deref(e.manager).name from employees e where e.name='Jaime Perez';
```

Salida de Script x  Resultado de la Consulta x

   SQL | Todas las Filas Recuperadas: 1 en 0,002 segundos

	DEREF(E.MANAGER).NAME
--	-----------------------

1	Pepe Moya
---	-----------

“La belleza y la fealdad son un espejismo porque los demás terminan viendo nuestro interior”

FRIDA KAHLO

