

MAP MEMBER FUNCTION

A MAP member function : Usa un valor escalar como elemento de comparación.

Ejemplo

```
CREATE OR REPLACE TYPE rectangle_typ AS OBJECT (  
  len NUMBER,  
  wid NUMBER,  
  MAP MEMBER FUNCTION area RETURN NUMBER);  
/  
  
CREATE OR REPLACE TYPE BODY rectangle_typ AS  
MAP MEMBER FUNCTION area RETURN NUMBER IS  
BEGIN  
  RETURN len * wid;  
END area;  
END;  
/  
  
create table parcelas(  
  nombre varchar2(100),  
  extension      rectangle_typ);  
  
insert into parcelas values('finca1',rectangle_typ(10,1));  
insert into parcelas values('finca2',rectangle_typ(1,10));  
insert into parcelas values('finca3',rectangle_typ(4,4));  
insert into parcelas values('finca4',rectangle_typ(7,2));  
  
select nombre,b.rectangle_typ.area from parcelas b order by extension desc;  
  
NOMBRE  
-----  
  
finca3  
finca4  
finca1  
finca2
```

Tambien sirve para hacer comparaciones de objetos en un bloque PL/SQL

```
DECLARE  
  r1 rectangle_typ;  
  r2 rectangle_typ;  
BEGIN  
  -- comparamos con valores iguales de superficie  
  r1 := rectangle_typ(10,1);  
  r2 := rectangle_typ(5,2);  
  
  IF r1 = r2 THEN  
    DBMS_OUTPUT.put_line( 'Son iguales.' );  
  ELSE  
    DBMS_OUTPUT.put_line( 'Son distintos.' );  
  END IF;  
  
  -- Comparamos con valores distintos de superficie  
  r1 := rectangle_typ(10,1);  
  r2 := rectangle_typ(2,9);  
  
  IF r1 = r2 THEN  
    DBMS_OUTPUT.put_line( 'Son iguales.' );  
  ELSE  
    DBMS_OUTPUT.put_line( 'Son distintos.' );  
  END IF;  
  
END;
```

ORDER MEMBER FUNCTION

Son para hacer comparaciones de un objeto frente a otro.

Un order method siempre va a recibir como parámetro un objeto del mismo tipo que será el que se compare con el propio objeto. (SELF)

Los order method son útiles para comparaciones complejas entre objetos. Pero no sirven para ordenaciones como los MAP

Ejemplo:

```
CREATE OR REPLACE TYPE location_typ AS OBJECT (  
    building_no NUMBER,  
    city VARCHAR2(40),  
    ORDER MEMBER FUNCTION match (l location_typ) RETURN INTEGER );  
/  
  
CREATE OR REPLACE TYPE BODY location_typ AS  
    ORDER MEMBER FUNCTION match (l location_typ) RETURN INTEGER IS  
    BEGIN  
        IF building_no < l.building_no THEN  
            RETURN -1;  
        ELSIF building_no > l.building_no THEN  
            RETURN 1;  
        ELSE  
            RETURN 0;  
        END IF;  
    END;  
END;  
  
DECLARE  
    loc location_typ;  
    secloc location_typ;  
    a number;  
BEGIN  
    loc :=NEW location_typ(300, 'San Francisco');  
    secloc :=NEW location_typ(200, 'Redwood Shores');  
    a := loc.match(secloc);  
    DBMS_OUTPUT.PUT_LINE('order (1 es mayor, -1 es menor):' ||a); -- prints order:1  
END;
```