

## Colecciones

Las bases de datos relacionales y orientadas a objetos permiten el almacenamiento de colecciones de elementos en una columna. En oracle, esta operación se lleva a cabo mediante una estructura de datos denominada *VARRAY*, similar a los arrays de otros lenguajes de programación; elementos del mismo tipo, identificados por su posición u otro índice. Anexo a este concepto se introduce el de las *tablas anidadas*, que facilitan almacenar una tabla en una columna de otra tabla.

### Tipo VARRAY

Mediante la sentencia del ejemplo se crea un varray de 3 elementos de nombre telefono compuesto de varchar2 de 9 posiciones.

```
CREATE TYPE TELEFONO AS VARRAY(3) OF VARCHAR2(9)
```

Para acceder a la información del VARRAY se utiliza la orden *DESC* (*DESC TELEFONO*). Por supuesto es posible crear tablas con columnas de tipo VARRAY:

```
CREATE TABLE AGENDA (  
    NOMBRE VARCHAR2(25),  
    TLF TELEFONO ## Columna de tipo VARRAY  
);  
  
INSERT INTO AGENDA VALUES (  
    'JUAN', TELEFONO('600123456', '987654321', '912345678')  
);  
  
INSERT INTO AGENDA (NOMBRE, TLF) VALUES ('FRANCISCO', ('678543210'));
```

No es posible establecer condiciones sobre los elementos almacenados en el VARRAY. Además, los valores que contiene, solo pueden ser accedidos o recuperados como bloque, no de forma individualizada (en PL/SQL sí se puede).

```
SELECT TLF FROM AGENDA;  
  
## Es conveniente usar alias  
SELECT A.TLF FROM AGENDA A;  
  
## Actualización de los datos del VARRAY  
UPDATE AGENDA SET TLF = TELEFONO('689123456', '675123489')  
WHERE NOMBRE = 'FRANCISCO';
```

Ejemplo de recorrido de VARRAY mediante PL/SQL;

```
DECLARE  
    CURSOR C1 IS SELECT * FROM AGENDA;
```

```

        CAD VARCHAR2(50);
BEGIN
    FOR I IN C1 LOOP
        DBMS_OUTPUT.PUT_LINE(I.NOMBRE || ', Número de teléfonos del contacto: ' ||
I.TLF.COUNT);
        CAD := '*';

        ##Recorrido del VARRAY
        FOR J IN 1 .. I.TLF.COUNT LOOP
            CAD := CAD || I.TLF(J) || '*';
        END LOOP;
        DBMS_OUTPUT.PUT_LINE(CAD);
    END LOOP;
END;

## SALIDA
JUAN, Número de teléfonos del contacto: 3
*600123456*987654321*912345678*
FRANCISCO, , Número de teléfonos del contacto: 2
*689123456*675123489*

```

Ejemplo de procedimiento para insertar datos en la AGENDA:

```

CREATE OR REPLACE PROCEDURE INSERTAR_AGENDA (N VARCHAR2, T TELEFONO) AS
BEGIN
    INSERT INTO AGENDA VALUES(N, T);
END;

## LLAMADA AL PROCEDIMIENTO
BEGIN
    INSERTAR_AGENDA('LUIS', TELEFONO('967654321'));
    INSERTAR_AGENDA('MARIANO', TELEFONO('984654321', '987600321'));
    COMMIT;
END;

```

Funciones sobre métodos de colecciones:

Argumento	Resultado
COUNT	Devuelve el número de elementos de la colección
EXISTS	TRUE si existe la fila, false en caso contrario
FIRST/LAST	Índice del primer y último elementos, respectivamente
NEXT/PRIOR	Elemento siguiente/posterior
LIMIT	Número máximo de elementos que puede contener la colección
DELETE	Elimina TODOS los elementos de la colección
EXTEND	Añade un elemento (Nulo) a la colección, si se le pasa un número, añade tantos nulos como indique el número
TRIM	Elimina el último elemento de la colección, o si se le pasa un número, tantos elementos como indique el número

Crea una función almacenada que reciba un nombre de la agenda y devuelva el primer

teléfono que tenga. Mediante un bloque PL/SQL, realiza llamadas a la función.

La función deberá controlar si la persona tiene o no teléfonos, y en caso de no tenerlos que muestre un mensaje indicándolo.

```
DECLARE
    TLF TELEFONO := TELEFONO(NULL, NULL, NULL);
BEGIN
    SELECT TELEF INTO TLF FROM AGENDA WHERE NOMBRE = 'MARIANO';

    DBMS_OUTPUT.PUT_LINE('Número de teléfonos actuales: ' || TLF.COUNT);
    DBMS_OUTPUT.PUT_LINE('Índice del primer teléfono: ' || TLF.FIRST);
    DBMS_OUTPUT.PUT_LINE('Índice del último teléfono: ' || TLF.LAST);
    DBMS_OUTPUT.PUT_LINE('Máximo número de teléfonos admitidos: ' || TLF.LIMIT);

    ## AÑADIR UN TELÉFONO AL VARRAY
    TLF.EXTEND;
    TLF(TLF.COUNT) := '123456789';
    UPDATE AGENDA A SET A.TLF = TLF WHERE NOMBRE = 'MARIANO';

    ## ELIMINAR
    SELECT TLF INTO TLF FROM AGENDA WHERE NOMBRE = 'LUIS';
    TLF.TRIM; ## Borra el último
    TLF.DELETE; ## Borra todos

    UPDATE AGENDA A SET A.TLF = TLF WHERE NOMBRE = 'LUIS';
END;
```

Crea un VARRAY de 3 elementos de tipo Persona.

Crea una tabla denominada GRUPOS, con dos columnas:

- \* NOMBRE\_GRUPO VARCHAR2(25)
- \* varray definido previamente

Inserta datos en la tabla de grupos (3 grupos) mediante un procedimiento.  
Recorre la tabla grupos mostrando la información de cada grupo (nombre del grupo y datos de sus miembros).