

Trabajando con collection types (VARARRAYS y Nested Tables).

Oracle soporta dos tipos de collection types : varrays y nested tables.

Varrays: listas ordenadas de objetos de longitud variables pero tamaño máximo limitado

Nested tables: Tablas que contienen tablas. A diferencia de los varrays no hay criterio de ordenación pero pueden crecer indefinidamente

Ejemplo de uso de varray:

```
CREATE TYPE phone_array AS VARRAY(10) OF VARCHAR2(12);
CREATE TABLE company1 (
    name VARCHAR2(20),
    phone phone_array
);

INSERT INTO company1 VALUES (
    'abc', phone_array('243-4758', '485-2534')
);
INSERT INTO company1 VALUES (
    'ibm', phone_array('888-478', '485-2999', '888-999', '777-478')
);
```

Ejemplo de uso de nested table:

```
CREATE TYPE phone_nested AS TABLE OF VARCHAR2(12);
CREATE TABLE company2 (
    name VARCHAR2(20),
    phone phone_nested
)
NESTED TABLE phone STORE AS phone_nr_table;
INSERT INTO company2 VALUES (
    'abc', phone_nested('243-4758', '485-2534')
);
```

Ejemplo de selects para varrays y nested tables (equivalentes para company1 y company2)

```
SELECT * FROM company1;

SELECT c.name, t.* FROM company1 c, TABLE(c.phone) t;

SELECT name, t.* FROM company1, TABLE(SELECT phone FROM company1) t;

SELECT t.COLUMN_VALUE FROM company1 c, TABLE (c.phone) t;
```

Los varray y nested tables pueden serlo a su vez de otros objetos y no solo de tipos simples como el ejemplo anterior. Fijaros en este ejemplo:

```
CREATE TYPE person_typ AS OBJECT (
    idno NUMBER,
    name VARCHAR2(30),
    phone VARCHAR2(20));
```

En este caso vamos a crear un nested table de estos objetos

```
CREATE TYPE people_typ AS TABLE OF person_typ; -- nested table type
/
```

Y ahora una tabla que contenga una columna tipo nested table *people_typ*:

```
CREATE TABLE people_tab (
    group_no NUMBER,
    people_column people_typ )
NESTED TABLE people_column STORE AS people_column_nt;
```

Insertamos el grupo 100 formado por dos personas:

```
INSERT INTO people_tab VALUES (
    100,
    people_typ( person_typ(1, 'John Smith', '1-650-555-0135'),
```

```
person_typ(2, 'Diane Smith', NULL));
```

Para visualizarlos mediante select:

```
SELECT c.group_no, t.* from people_tab c, TABLE(c.people_column) t;
```

GROUP_NO	IDNO	NAME	PHONE
100	1	John Smith	1-650-555-0135
100	2	Diane Smith	

Obtener información sobre la colección desde PL/SQL

- COUNT devuelve el número de filas.
- EXISTS devuelve TRUE si la fila existe.
- FIRST/LAST devuelve el índice de la primera y última fila.
- NEXT/PRIOR devuelve la fila anterior o posterior a la actual.
- LIMIT informa del número máximo de elementos que puede contener .

Modificar los elementos de la colección

- DELETE borra uno o más elementos usando su índice.
- EXTEND añade nuevas filas.
- TRIM elimina filas.

Ejemplo creando un bloque anónimo de PL/SQL

```
DECLARE
prueba_phone phone_array;
BEGIN
SELECT phone INTO prueba_phone FROM company1 WHERE name='abc' ;
FOR i IN prueba_phone.FIRST .. prueba_phone.LAST
LOOP
DBMS_OUTPUT.PUT_LINE('Telefonos= ' || prueba_phone(i));
END LOOP;
END;
```

Ejemplo de contar el número de elementos (solo para Nested Tables)

```
SELECT name,CARDINALITY(phone) "numero de telefonos" from company2;
```