6

Tipos compuestos ORACLE Colecciones

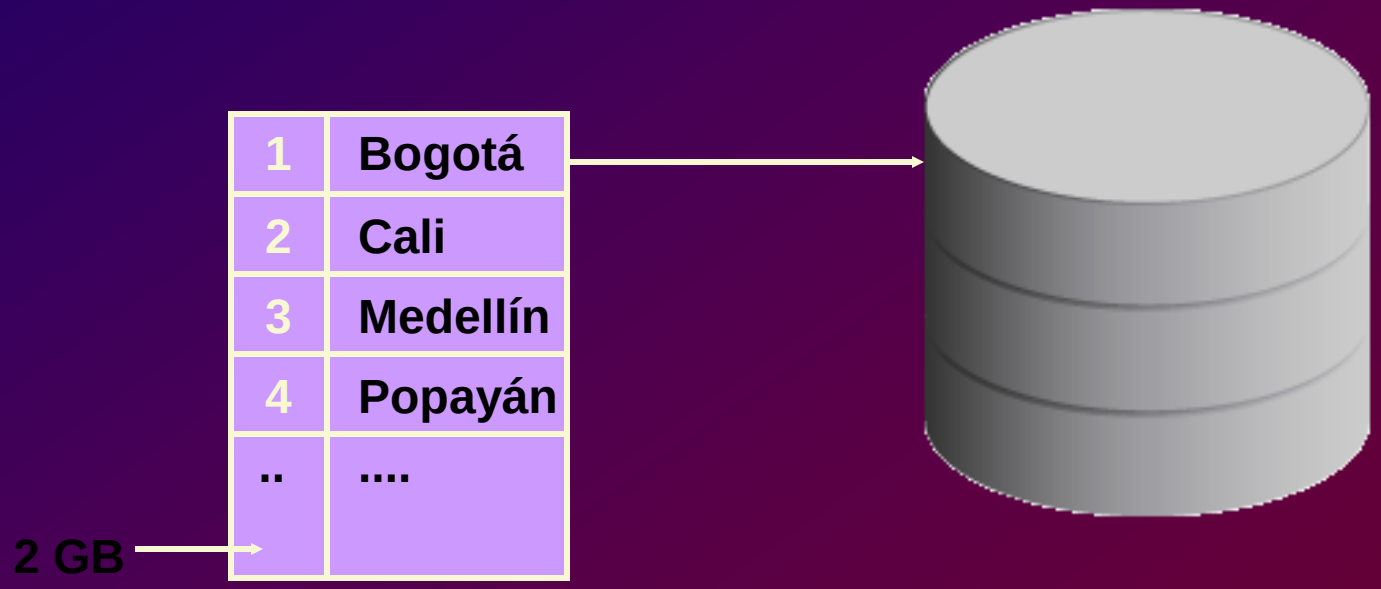
Bases de datos II

Universidad del Cauca

Ing. Wilson Ortega

ORACLE®

Tablas anidadas



ORACLE®

Tablas anidadas - definición

- Tiene una columna con números en secuencia que se considera como columna clave (posición de cada elemento iniciando en “1”).
- La clave no puede ser un valor negativo, a diferencia que con la tabla `INDEX BY`.
- Los elementos se pueden suprimir desde cualquier lugar de una tabla anidada dejando una tabla dispersa con claves no secuenciales.

ORACLE®

Métodos de tablas

Método	Descripción
EXISTS(*n*)	Devuelve TRUE si existe el elemento *n* en una tabla PL/SQL.
COUNT	Devuelve el número de elementos que en ese momento contenga una tabla PL/SQL.
FIRST LAST	Devuelve el primer y último número de índice de una tabla PL/SQL. Devuelve NULL si la tabla PL/SQL está vacía.
PRIOR(*n*)	Devuelve el número de índice que precede al índice *n* en una tabla PL/SQL.
NEXT(*n*)	Devuelve el número de índice que sucede al índice *n* en una tabla PL/SQL.
DELETE	DELETE elimina todos los elementos de una tabla PL/SQL. DELETE(n) elimina el elemento *n* de una tabla PL/SQL. DELETE(m, n) elimina todos los elementos del rango *m* ... *n* de una tabla PL/SQL.

ORACLE®

Creación de Tablas anidadas

Sintaxis

```
TYPE nombre_tipo_tabla IS TABLE OF
 {column_type | variable%TYPE
 | table.column%TYPE} [NOT NULL]
 | table.%ROWTYPE
```

A partir de la definición de la tabla se pueden crear variables de ese tipo:

```
Identificador nombre_tipo_tabla; -- Crea una tabla NULL. Se debe
inicializar para poder agregar valores.
```


ORACLE®

Tablas anidadas – Ejemplos (I)

```
SET SERVEROUTPUT ON;
DECLARE
 TYPE tabla_nombres IS TABLE OF VARCHAR(50);
 v_nombres tabla_nombres;

BEGIN
  v_nombres := tabla_nombres('Andres','Camila','Gloria','Javier');
  dbms_output.put_line('Nombres en lista: ' || v_nombres.COUNT);
  dbms_output.put_line('Nombres en (1):' || v_nombres(1));
END ;
```

```
dbms_output.put_line('Nombres en (0):' || v_nombres(0)); -- EXCEPCIÓN
```

```
v_nombres(1) := 'Pedro';  -- OK
v_nombres(5) := 'Pedro'; -- EXCEPCIÓN
```

```
v_nombres.EXTEND(1);
v_nombres(5) := 'Pedro'; -- OK
```


ORACLE®

Tablas anidadas – Ejemplos (II)

```
DECLARE
 TYPE tabla_anidada_estudiantes IS TABLE OF ESTUDIANTE%ROWTYPE;
 v_tblestudiantes tabla_anidada_estudiantes;
BEGIN
 SELECT * INTO v_tblestudiantes(1) FROM ESTUDIANTE WHERE CODIGO = 15;
 dbms_output.put_line('Estudiante (1): ' || v_tblestudiantes(1).NOMBRE);
END ;
-------
ORA-06531: Reference to uninitialized collection
```

```
DECLARE
 TYPE tabla_anidada_estudiantes IS TABLE OF ESTUDIANTE%ROWTYPE;
 v_tblestudiantes tabla_anidada_estudiantes;
BEGIN
 v_tblestudiantes := tabla_anidada_estudiantes();
 v_tblestudiantes.EXTEND(1);
 SELECT * INTO v_tblestudiantes(1) FROM ESTUDIANTE WHERE CODIGO = 15;
 dbms_output.put_line('Estudiante (1): ' || v_tblestudiantes(1).NOMBRE);
END ;
```


ORACLE®

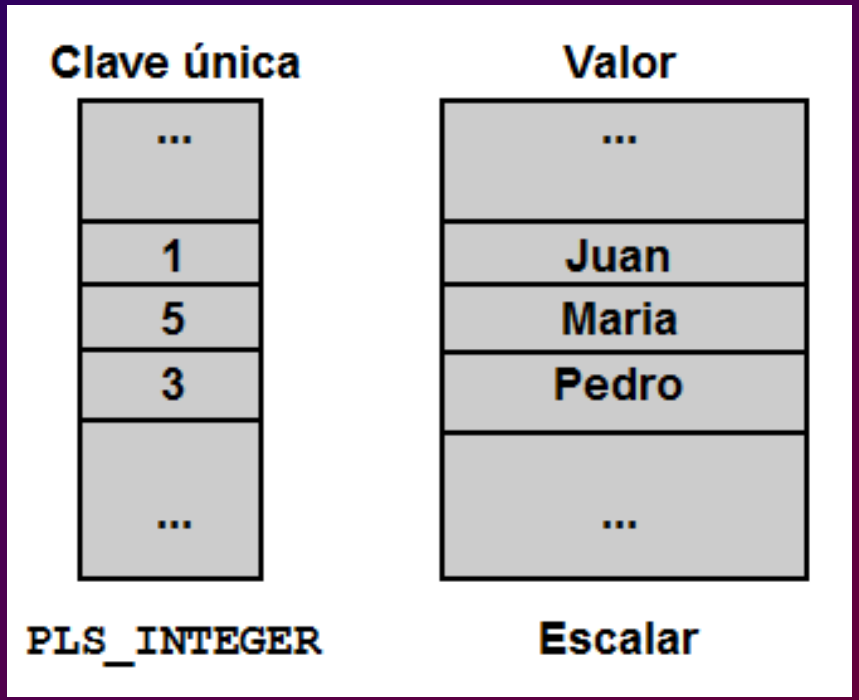
Tablas anidadas – Ejemplos (III)

Usando BULK COLLECT no es necesario inicializar la tabla.

```
SET SERVEROUTPUT ON;
DECLARE
 TYPE tabla_anidada_estudiantes IS TABLE OF ESTUDIANTE%ROWTYPE;
 v_tblestudiantes tabla_anidada_estudiantes;
BEGIN

SELECT * BULK COLLECT INTO v_tblestudiantes FROM ESTUDIANTE;

FOR i IN v_tblestudiantes.FIRST..v_tblestudiantes.LAST LOOP
  dbms_output.put_line('Estudiante (' || i || '):' ||
v_tblestudiantes(i).NOMBRE);
END LOOP ;

END ;
```


ORACLE®

Estructura de la tabla INDEX BY



ORACLE®

Arrays asociativos o Tablas INDEX BY

- Son estructuras PL/SQL con dos columnas:
  - Una columna de tipo cadena o entero que actúa como clave primaria. La clave puede ser numérica (Normalmente se usa el tipo PLS_INTEGER debido a que necesitan menos almacenamiento que NUMBER. La clave puede ser también del tipo VARCHAR2.
  - Una columna de tipo de dato escalar o de registro para contener valores. Si la columna es de tipo escalar, sólo puede contener un valor. Si la columna es de tipo de registro, podrá contener varios valores.
- No tienen límite en cuanto a tamaño. Las claves pueden ser positivas y negativas. Las claves de las tablas INDEX BY no están en secuencia.

ORACLE®

Creación de Tablas INDEX BY

Sintaxis

```
TYPE nombre_tipo_tabla IS TABLE OF
      {column_type | variable%TYPE
      | table.column%TYPE} [NOT NULL]
      | table%ROWTYPE
      [INDEX BY PLS_INTEGER | BINARY_INTEGER
       | VARCHAR2(<size>)];
```

A partir de la definición de la tabla se pueden crear variables de ese tipo:

```
Identificador nombre_tipo_tabla; -- Crea una tabla vacía y es posible
agregar elementos usando cualquier índice
```


ORACLE®

Tablas INDEX BY Ejemplos (I)

```
DECLARE
TYPE tabla_nombres_estudiantes is TABLE OF ESTUDIANTE.NOMBRE%TYPE  INDEX BY PLS_INTEGER;
tbl_nom_est tabla_nombres_estudiantes;
BEGIN
  tbl_nom_est(1) := 'Juan';
  tbl_nom_est(5) := 'Maria';
  tbl_nom_est(3) := 'Carlos';
  dbms_output.put_line('Estudiante con codigo 1 ' || tbl_nom_est(1) );

  IF tbl_nom_est.EXISTS(2) THEN
    dbms_output.put_line('Estudiante con codigo 2 ' || tbl_nom_est(2) );
  ELSE    dbms_output.put_line('No existe un estudiante con codigo 2');
  END IF; -- excepción si tbl_nom_est(2) no existe
  dbms_output.put_line('Estudiantes en lista: ' || tbl_nom_est.COUNT); --3
  dbms_output.put_line('Menor código en lista: ' || tbl_nom_est.FIRST); -- 1
  dbms_output.put_line('Mayor código en lista: ' || tbl_nom_est.LAST); -- 5
  dbms_output.put_line('Código anterior al de Carlos: '||tbl_nom_est.PRIOR(3)); -- 1
  dbms_output.put_line('Código anterior al de Juan: '||tbl_nom_est.PRIOR(1)); -- NUll
  dbms_output.put_line('Codigo posterior al de Juan: ' || tbl_nom_est.NEXT(1)); -- 3
  tbl_nom_est.DELETE(3);
  dbms_output.put_line('Estudiantes en lista: ' || tbl_nom_est.COUNT);
  tbl_nom_est.DELETE();
  dbms_output.put_line('Estudiantes en lista: ' || tbl_nom_est.COUNT);
END;
```


ORACLE®

Tablas INDEX BY Ejemplos (II)

```
SET SERVEROUTPUT ON
DECLARE
 TYPE tabla_ciudades IS TABLE OF NUMBER INDEX BY VARCHAR2(50);
 tbl_ciudades tabla_ciudades;

BEGIN

    tbl_ciudades('Popayán') := 300000;
    tbl_ciudades('Bogotá') := 2000000;
    tbl_ciudades('Cali') := 500000;

    dbms_output.put_line( tbl_ciudades('Bogotá') );
    dbms_output.put_line( tbl_ciudades.PRIOR('Popayán') ); -- Cali
    dbms_output.put_line( tbl_ciudades.NEXT('Cali') ); -- Popayán
    dbms_output.put_line( tbl_ciudades.COUNT );


END ;
```

Ejercicio: Imprima en pantalla el contenido de la tabla usando una estructura repetitiva.

ORACLE®

Tablas INDEX BY Ejemplos (III)

```
SET SERVEROUTPUT ON
DECLARE
 TYPE tabla_estudiantes IS TABLE OF ESTUDIANTE%ROWTYPE INDEX BY PLS_INTEGER;
 tbl_estudiantes tabla_estudiantes;
 v_codigo ESTUDIANTE.CODIGO%TYPE := 15;
BEGIN
    SELECT * INTO tbl_estudiantes(v_codigo) FROM ESTUDIANTE WHERE CODIGO = v_codigo;
    dbms_output.put_line('Estudiante con codigo ' || v_codigo || ' : ' ||
    tbl_estudiantes(v_codigo).NOMBRE );
END ;
```

```
SET SERVEROUTPUT ON
DECLARE
 TYPE tabla_estudiantes IS TABLE OF ESTUDIANTE%ROWTYPE INDEX BY ESTUDIANTE.NOMBRE%TYPE;
 tbl_estudiantes tabla_estudiantes;
 v_nombre ESTUDIANTE.NOMBRE%TYPE := 'Sara';
BEGIN
    SELECT * INTO tbl_estudiantes(v_nombre) FROM ESTUDIANTE WHERE CODIGO = 15;
    dbms_output.put_line('Estudiante con nombre ' || v_nombre || ' : Código: ' ||
    tbl_estudiantes(v_nombre).CODIGO );
END ;
```

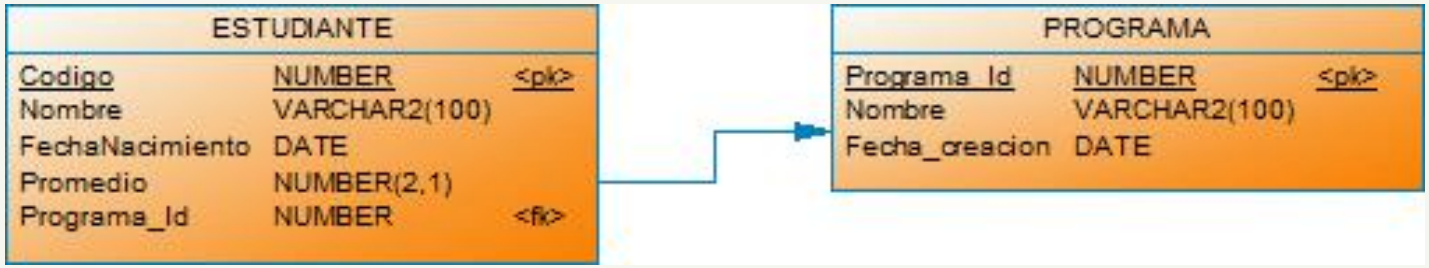

ORACLE®

Tablas INDEX BY Ejemplos (III)

```
DECLARE
  TYPE tabla_estudiantes IS TABLE OF ESTUDIANTE%ROWTYPE INDEX BY PLS_INTEGER;
  tbl_estudiantes tabla_estudiantes;
  num_estudiantes NUMBER;
BEGIN

 SELECT * BULK COLLECT INTO tbl_estudiantes FROM ESTUDIANTE WHERE EST_PROGRAMA IS NOT
NULL;
 FOR i IN tbl_estudiantes.FIRST..tbl_estudiantes.LAST LOOP
        dbms_output.put_line('Estudiante: ' || tbl_estudiantes(i).EST_NOMBRE1 );
 END LOOP;
END ;
```


ORACLE®

Ejercicio



- Cree un bloque anónimo que imprima en pantalla el código y el nombre de todos los estudiantes que pertenecen al programa cuyo id digita el usuario.
- En un bloque anónimo, inserte los siguientes datos de programas en una tabla anidada:
  - Id: 25, Nombre: Medicina
  - Id: 56, Nombre: Contaduría
  - Id: 67, Nombre: Sistemas

Luego, imprima los nombre de los programas almacenados.

ORACLE

VARRAY

- Las matrices de tamaño variable (VARRAY) son similares a las tablas PL/SQL, a excepción de que una VARRAY tiene límite en cuanto a tamaño.
- Es válido en una tabla a nivel de esquema.
- Tienen un límite superior fijo. Se debe especificar el límite superior cuando se declaran. (Arrays lenguaje C).
- El tamaño máximo de una VARRAY es de 2 GB, como en las tablas anidadas.
- Los elementos de VARRAY se almacenan de forma contigua en memoria y no en la base de datos.

ORACLE®

VARRAY – Ejemplos (I)

```
DECLARE
 TYPE array_nombres IS VARRAY(4) OF VARCHAR(50);
 v_nombres array_nombres;
BEGIN
  v_nombres := array_nombres('Andres','Camila','Gloria','Javier');
  dbms_output.put_line('Nombres en lista: ' || v_nombres.COUNT);
  dbms_output.put_line('Nombres en (1):' || v_nombres(1));
END ;
```

```
DECLARE
 TYPE array_nombres IS VARRAY(4) OF VARCHAR(50);
 v_nombres array_nombres;
BEGIN
  v_nombres := array_nombres('Andres','Camila','Gloria','Javier');
  dbms_output.put_line('Nombres en lista: ' || v_nombres.COUNT);
  dbms_output.put_line('Nombres en (1):' || v_nombres(1));
  v_nombres.EXTEND(1);
  v_nombres(5):= 'Pedro'; -- EXCEPTION!!
END ;
```


ORACLE®

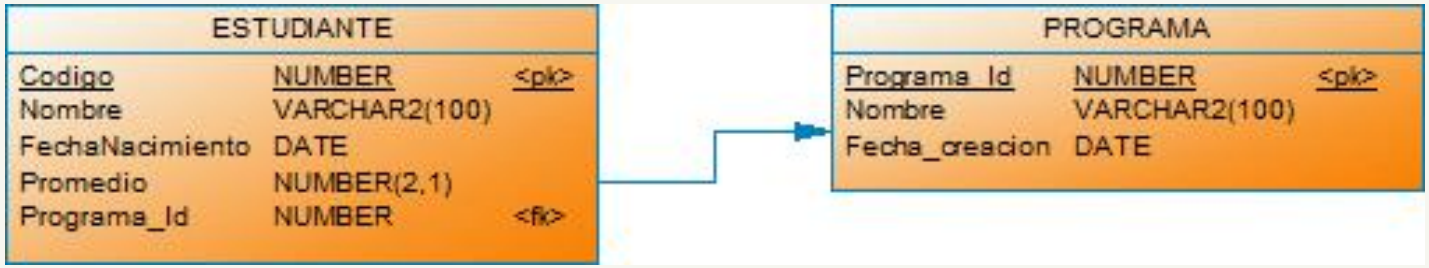
VARRAY – Ejemplos (II)

```
DECLARE
 TYPE tipo_array_estudiantes IS VARRAY(2) OF ESTUDIANTE%ROWTYPE;
 v_estudiantes tipo_array_estudiantes;
BEGIN
 SELECT * BULK COLLECT INTO v_estudiantes FROM ESTUDIANTE;
 FOR i IN v_estudiantes.FIRST..v_estudiantes.LAST LOOP
  dbms_output.put_line('Estudiante (' || i || '):' || v_estudiantes(i).NOMBRE);
 END LOOP ;
END ;
-----
Si hay más de dos estudiantes ???
 ORA-22165: given index [3] must be in the range of [1] to [2]
```

```
SELECT * BULK COLLECT INTO v_estudiantes FROM ESTUDIANTE WHERE ROWNUM < 3;
```


ORACLE®

Ejercicio



- Cree un procedimiento que reciba como parámetro el id de un programa. Luego se muestra el nombre del programa y código y nombre de cada estudiante. Al final se muestra el número de estudiantes matriculados.
  - Use una tabla index by
  - Use una tabla anidada

- Cree un bloque anónimo que almacene en un VARRAY los tres estudiantes con mayor promedio y luego muestre su nombre y promedio en pantalla.

ORACLE

Bibliografía

- **Oracle® Database** PL/SQL Language Reference -11*g* Release 1 (11.1) - 2009

ORACLE®