**Nested Tables**

**The functionality of nested tables is similar to that of associative arrays; however, there are differences in the nested table implementation.**

* + - **The nested table is a valid data type in a schema-level table, but an associative array is not. Therefore, unlike associative arrays, nested tables can be stored in the database.**
    - **The size of a nested table can increase dynamically, although the maximum size is 2 GB.**
    - **The “key” cannot be a negative value (unlike in the associative array). Though reference is made to the first column as key, there is no key in a nested table. There is a column with numbers.**
    - **Elements can be deleted from anywhere in a nested table, leaving a sparse table with nonsequential “keys.” The rows of a nested table are not in any particular order.**
    - **When you retrieve values from a nested table, the rows are given consecutive subscripts starting from 1.**

**Syntax**

**TYPE *type\_name* IS TABLE OF**

**{column\_type | variable%TYPE**

**| table.column%TYPE} [NOT NULL]**

**| table.%ROWTYPE**

**Nested Tables (continued)**

**Example:**

**TYPE location\_type IS TABLE OF locations.city%TYPE;**

**offices location\_type;**

**If you do not initialize a nested table, it is automatically initialized to NULL. You can initialize the offices nested table by using a constructor:**

**offices := location\_type('Bombay', 'Tokyo','Singapore', 'Oxford');**

**The complete code example and output is as follows:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**TYPE location\_type IS TABLE OF locations.city%TYPE; --Declaration**

**offices location\_type; -- Instantiation**

**table\_count NUMBER;**

**BEGIN**

**offices := location\_type('Bombay', 'Tokyo','Singapore','Oxford'); -- Initialization**

**FOR i in 1.. offices.count() LOOP**

**DBMS\_OUTPUT.PUT\_LINE(offices(i));**

**END LOOP;**

**END;**

**/**

**VARRAY**

**A variable-size array (VARRAY) is similar to an associative array, except that a VARRAY is constrained in size.**

* + - **A VARRAY is valid in a schema-level table.**
    - **Items of VARRAY type are called VARRAYs.**
    - **VARRAYs have a fixed upper bound. You have to specify the upper bound when you declare them. This is similar to arrays in C language. The maximum size of a VARRAY is 2 GB, as in nested tables.**
    - **The distinction between a nested table and a VARRAY is the physical storage mode. The elements of a VARRAY are stored inline with the table’s data unless the size of the VARRAY is greater than 4 KB. Contrast that with nested tables, which are always stored out-of-line.**
    - **You can create a VARRAY type in the database by using SQL.**

**Example:**

**TYPE location\_type IS VARRAY(3) OF locations.city%TYPE;**

**offices location\_type;**

**The size of this VARRAY is restricted to 3. You can initialize a VARRAY by using constructors. If you try to initialize the VARRAY with more than three elements, a “Subscript outside of limit” error message is displayed.**

**offices := location\_type('Bombay', 'Tokyo','Singapore','Oxford'); -- Error**

**Summary of Collection Types**

**Associative Arrays**

Associative arrays are sets of key-value pairs, where each key is unique and is used to locate a corresponding value in the array. The key can be either integer- or character-based. The array value may be of the scalar data type (single value) or the record data type (multiple values).

Because associative arrays are intended for storing temporary data, you cannot use them with SQL statements such as INSERT and SELECT INTO.

**Nested Tables**

A nested table holds a set of values. In other words, it is a table within a table. Nested tables are unbounded; that is, the size of the table can increase dynamically. Nested tables are available in both PL/SQL and the database. Within PL/SQL, nested tables are like one-dimensional arrays whose size can increase dynamically.

**Varrays**

Variable-size arrays, or varrays, are also collections of homogeneous elements that hold a fixed number of elements (although you can change the number of elements at run time). They use sequential numbers as subscripts. You can define equivalent SQL types, thereby allowing varrays to be stored in database tables.