

## Object Types: Varray and Table Collections

---

- Object Types
  - Varray Collections:
    - Single-dimensional
    - Has size limit
    - By default allow NULL
    - Elements are all the same data type
    - Both varray and Table data types have three possible states:
      - NULL
      - Empty
      - Populated
    - Use this when the number of elements is known, the elements need to be accessed in order or the order needs to be guaranteed
    - Example:

```
SQL> DECLARE
2  /* Declare a collection variable with a constructor call. */
3  lv_stooges SQL_VARRAY := sql_varray('Moe','Larry');
4  BEGIN
5  /* Print the number and limit of elements. */
6  dbms_output.put_line(
7    'Count ['||lv_stooges.COUNT||'] '||
8    'Limit ['||lv_stooges.LIMIT||']');
9
10 /* Extend space and assign to the new index. */
11 lv_stooges.EXTEND;
12
13 /* Print the number and limit of elements. */
14 dbms_output.put_line(
15   'Count ['||lv_stooges.COUNT||'] '||
16   'Limit ['||lv_stooges.LIMIT||']');
17
18 /* Assign a new value. */
19 lv_collection(lv_stooges.COUNT) := 'Curly';
20
21 /* Iterate across the collection to the total number of elements. */
22 FOR i IN 1..lv_stooges.COUNT LOOP
23   dbms_output.put_line(lv_stooges(i));
24 END LOOP;
25 END;
26 /
```

## ○ Table Collections

- Single-dimensional structure
- No limits (except database resources)
- Must have same data type
- Default allows NULL
- When using Attribute Data Type (ADT) collection must use PL/SQL to update or add an element
- You want to use a table collection when a number of elements is unknown, when the number of order is unknown
- Example:

```
SQL> CREATE OR REPLACE FUNCTION add_element
2  ( pv_table   SQL_TABLE
3  , pv_element VARCHAR2 ) RETURN SQL_TABLE IS
4
5  /* Declare a local table collection. */
6  lv_table   SQL_TABLE := sql_table();
7  BEGIN
8
9  /* Check for an initialized collection parameter. */
10 IF pv_table.EXISTS(1) THEN -- A suboptimal comparison.
11   lv_table := pv_table;
12 END IF;
13
14 /* Check for a not null element before adding it. */
15 IF pv_element IS NOT NULL THEN
16   /* Extend space and add an element. */
17   lv_table.EXTEND;
18   lv_table(lv_table.COUNT) := pv_element;
19 END IF;
20
21 /* Return the table collection with its new member. */
22 RETURN lv_table;
23 END;
24 /
```