**Definition of collection**

* A collection is an array like structure which can store multiple values (with multiple rows with one or more columns)
* It is an user defined data type
* A collection helps to process or handle multiple rows within PLSQL program

**Advantages**

1. Helps to handle multiple-rows within PLSQL program (similar to cursor)
2. Helps to return multiple values from a procedure or a function (multiple rows) (similar to ref cursors)
3. Helps to improve the performance of PLSQL program using a concept called **BULK-BINDING.**
4. A collection is re-usable (as it can be stored in the database)
5. In collection, random access of the elements is possible in addition to sequential access.
6. In collection, reverse access of the elements is possible in addition to sequential access.

**Types**

1. Associative arrays (or) index-by tables (or) PL/SQL tables
2. Nested tables
3. VARRAYS (variable arrays)

**Associative arrays**

It is an un-ordered set of elements with all the indexes are pre-defined.

**Properties of Associative Arrays**

* Here the indexes are in-built (INDEX BY PLS-INTEGER)
* They are unbounded (no size should be given during the creation)
* Max size is between -232 to 231
* They can take negative subscripts
* They can take character subscripts
* They are not stored in DB
* They are not re-usable
* They can be used in PLSQL level
* Initialization is not required
* EXTEND method is not required to create a new slot (because all the slots are predefined)
* Here we can delete element/slot individually
* They are always “SPARSE “(non-continuous subscripts) in nature
* Accessing a slot which is not existing which will throw no\_data\_found error

**Nested table**

It is an un-ordered set of elements with all the indexes are not pre-defined.

**Properties of Nested Tables**

* Here the indexes are not pre-defined
* They are unbounded (no size should be given during the creation)
* Max size is between 1 to 231
* They cannot take negative subscripts
* They cannot take character subscripts
* They can be stored in DB
* They are re-usable
* They are used in both SQL and PLSQL level
* Initialization is required
* EXTEND method is required to create a new slot (because all the slots are not pre-defined)
* Here we can delete element/slot individually
* They are initially “DENSE” in nature but after deletion it can become “SPARSE(non-continuous) “in nature
* Accessing a slot which is not existing will not throw no\_data\_found error

**VARRAYS**

It is an ordered set of elements with all the indexes are not pre-defined.

* Here the indexes are not pre-defined
* They are bounded (should give size)
* Max size is between 1 to 231
* They cannot take negative subscripts
* They cannot take character subscripts
* They are stored in DB, they are re-usable
* They can be used in both SQL and PLSQL level
* Initialization is required
* EXTEND method is required to create a new slot
* Here we cannot delete element/slot individually but we can delete from last
* They are always “DENSE “in nature
* Accessing a slot which is not existing which will not throw no\_data\_found error

**Collection methods /attributes**

* **First** – returns the first index number in a collection which has a first value
* **Last -** returns the index number in a collection which has a last value
* **Count** – returns the number of elements that a collection currently contains
* **Prior (n)** – returns the index number of the previous value with respect to “n” in a collection
* **next (n)** – returns the index number of the next value with respect to “n” in a collection
* **exists (n)** – returns true if the nth element in a collection exists otherwise it returns false
* **extend (n)** – appends n elements/slots to a collection
* **trim** – removes the one element from the end of the collection
  + **trim (n)** – removes the n elements from the end of the collection
* **delete** – removes all elements from a collection
* **Delete (m,n)** – removes all elements in the range m..n from an associative array or nested table
  + If m is larger than n or if m or n is null, delete (m,n) does nothing
* **Limit** – applicable only for varrays because it has boundary (returns the size of the varray)

**vtemp**

|  |  |
| --- | --- |
| 1 |  |
| 2 | 600 |
| 3 | 800 |
| 4 |  |
| 5 | 900 |
| 6 | 1000 |
| 7 |  |
| 8 |  |

**Vtemp.first = 2**

**Vtemp.last = 6**

**Vtemp.count = 4**

**Vtemp.prior(5) = 3**

**Vtemp.next (3) = 5**

**Vtemp.exists (2) = T**

**Vtemp.exists (7) = T**

Vtemp.exists (9) = F

Vtemp.extend(5);

Vtemp.extend (5,2); from 5th slot value 2 is added