

### Exercise #1: VARRAY collections

1. In an anonymous PL/SQL block, define a type VARRAY of 10 reals that shall store exam marks.
2. In the DECLARE section, define a variable of this type and initialize it with three marks.
3. In the code section (BEGIN), complete the initialization with 7 more marks. Test!
4. Compute and display mark average.
5. Add another mark into the collection. What happens?
6. Display the maximum and real size of the collection.
7. With the help of the `trim` method, delete one mark from the collection. Then, display again the maximum and real size of the collection.
8. With the help of the `delete` method, empty the collection. Then, display again the maximum and real size of the collection.

### Exercise #2: TABLE collections

1. Copy table NOTATION of user DARMONT onto your account.
2. In an anonymous PL/SQL block define a type TABLE of reals that shall contains marks stored in table NOTATION.
3. Define a variable of this type and initialize it as empty.
4. In the code section, load into the collection all the marks (attribute NOTECC) from table NOTATION with the help of a cursor. Also display each mark on the fly. Test! Would that kind of processing have been possible with a VARRAY collection?
5. Compute and display mark average.
6. Still in your PL/SQL block, check whether your result is right by displaying the result of an SQL query using aggregate function `AVG()`.
7. Display on screen the index of the first element, the index of the last element and the number of elements in the collection.
7. Delete element of indices 1, 10 and 16, then repeat question #7 (copy/paste).
8. Display all elements in the collection.

### Exercise #3: Records

1. In an anonymous PL/SQL block, define a RECORD type composed of the following fields:
  - name, of same type than that of attribute ENAME of table EMP;
  - job, of same type than that of attribute JOB of table EMP;
  - location, of same type than that of attribute LOC of table DEPT;
  - salary, of same type than that of attribute SAL of table EMP.
2. Define a variable of this type.
3. In the code section, define a parameterized query (using character `&`) that helps load the name, job, location and total salary (i.e., raw salary + commission) of an employee whose number is entered on keyboard. Store it in the previously defined record variable. Display the result as follows: `NAME job (LOCATION): Salary.`

Hint: Use the `NVL`<sup>1</sup> function to avoid summing problems with NULL commissions.

<sup>1</sup> <http://docs.oracle.com/database/121/SQLRF/functions130.htm#SQLRF00684>