

Las Colecciones y sus Métodos

Estos métodos son Funciones o Procedimientos incorporados en las **Colecciones**, los mismos facilitan su manejo y mantenimiento. Para hacer uso de ellos usamos la notación de punto.

Sintaxis:

collection_name.

```
{COUNT  
| DELETE [ (index [, index]) ]  
| EXISTS (index)  
| EXTEND [ (number [, index]) ]  
| FIRST  
| LAST  
| LIMIT  
| NEXT (index)  
| PRIOR (index)  
| TRIM [ (number) ]  
}
```

En la Sintaxis:

collection_name: Es el nombre de la variable tipo **Colección**.

Método	Descripción.
COUNT	Devuelve el número de elementos que contiene la Colección , muy útil porque no siempre se conoce el tamaño actual de una Colección . Puede utilizar COUNT dondequiera que se permita una expresión numérica. En el caso de los VARRAYs , el valor de COUNT siempre es igual a LAST . Para las Tablas Anidadas y las Matrices Asociativas , COUNT es normalmente igual a LAST . A menos que se eliminen elementos del centro, si esto ocurre, COUNT sería menor que LAST .
DELETE	Solo puede ser usado con Tablas Anidadas y/o Matrices Asociativas . Tiene Tres variantes: DELETE elimina todos los elementos de una Colección . DELETE(n) elimina el elemento n . Si n es nulo, DELETE(n) no hace nada. DELETE(m, n) elimina todos los elementos del rango m..n . Si m es mayor que n o si m o n es nulo, DELETE(m, n) no hace nada.
EXISTS	EXISTS(n) devuelve TRUE si existe el elemento n de una Colección , de lo contrario, devuelve FALSE . Puede utilizar EXISTS para evitar una excepción al hacer referencia a un elemento inexistente. Cuando se le pasa un subíndice fuera de rango, EXISTS devuelve FALSE en lugar de lanzar la excepción SUBSCRIPT_OUTSIDE_LIMIT .
EXTEND	Es un método exclusivo de las Tablas Anidadas y Los VARRAYs . Este procedimiento tiene tres formas: EXTEND agrega un elemento nulo. EXTEND(n) agrega n elementos nulos. EXTEND(n, i) agrega n copias del elemento i . EXTEND opera sobre el tamaño interno de una Colección . Si EXTEND encuentra elementos eliminados, los incluye en su recuento.

FIRST	FIRST devuelve el primer (más pequeño) valor de subíndice en una colección. Los valores de subíndice suelen ser enteros, pero en el caso de las Matrices Asociativas también pueden ser cadenas. Si la Colección está vacía, FIRST devuelve NULL . Para Los VARRAYs , FIRST siempre devuelve 1.
LAST	LAST devuelve el último (más grande) valor de subíndice en una Colección . Al igual que FIRST , si la Colección está vacía, LAST devuelve NULL . Para Los VARRAYs , LAST siempre es igual a COUNT . Para las Tablas Anidadas y las Matrices Asociativas , LAST es normalmente igual a COUNT . A menos que se eliminen elementos del centro, si esto ocurre, LAST sería mayor que COUNT .
PRIOR(n)	Retorna el número de índice que antecede al índice n .
NEXT(n)	Retorna el número de índice que sigue(sucede) al índice n .
LIMIT	Para las Tablas Anidadas y las Matrices Asociativas (No tienen un tamaño máximo), LIMIT devuelve NULL . Para Los VARRAYs , LIMIT devuelve el número máximo de elementos que un VARRAY puede contener (Especificado en la definición de tipo).
TRIM	Es un método exclusivo de las Tablas Anidadas y Los VARRAYs . Este procedimiento tiene dos formas: TRIM elimina un elemento del final de una Colección . TRIM(n) elimina n elementos del final de una Colección . Si n es mayor que COUNT , TRIM(n) eleva la excepción SUBSCRIPT_BEYOND_COUNT . TRIM opera en el tamaño interno de una Colección . Si TRIM encuentra elementos eliminados, los incluye en su recuento.

Algunas Directrices:

- Los métodos de las **Colecciones** no se pueden ser invocados desde Sentencias **SQL**.
- Sólo es posible usar el método **EXISTS** en **Colecciones** nulas. Si aplica otro método a dichas **Colecciones**, se eleva la excepción: **COLLECTION_IS_NULL**.
- Los métodos **EXISTS**, **COUNT**, **LIMIT**, **FIRST**, **LAST**, **PRIOR** y **NEXT** son funciones; **EXTEND**, **TRIM** y **DELETE** son procedimientos.