TABLE

- El tipo table , es una colección de datos de tipo pila con todos los datos del mismo tipo.
- La sintaxis sería:
  type nombre_tipo_tabla is table of tipo_elemento índex by {BYNARY_NTEGER|PLS_INTEGER}

- En este tipo de colecciones podemos utilizar registros como elementos.
- De este modo, creamos una matriz , cuyas columnas son cada uno de los campos del registro.
- Se accede a cada fila por un indice , que es la posición del array.
- La notación para acceder a un elemento sería variable(posición)
- Para acceder al campo de un registro si fuese este el tipo de elementos del array, usaremos la siguiente notación:

  variable(posicion).campo

- Es muy habitual utilizar este tipo de estructuras para almacenar datos con la salida de una consulta.

Ejemplo:

```
type tablaNumeros is table of number index by Binary_integer;
datos tablaNumeros;
begin
datos(1) := 1;
datos(2):=2;
for i in 1.. datos.count loop
dbms_output.put_line ('Employee Number: '||datos(i));
end loop;
end;
```

- Como vemos no tiene tamaño predefinido , por tanto irá creciendo según necesitemos.
- Como vemos las tablas son muy útiles para mover un conjunto de datos con un único
- Entre paréntesis utilizamos el índice, que no tiene por qué ser correlativo, es decir, después de datos(2) podremos guardar datos (200):=1000; datos(50):=9; Oracle reorganiza la tabla, guardando antes el índice 50 que el 200
- Los indices pueden ser normalmente BINARY_INTEGER aunque podría ser Varchar2 o un tipo de un campo de alguna tabla. %type.

Ejemplos:

type tablaalumnos is table of ralumnos index by Binary_Integer

type tablaalumnos is table of alumnnos%rowtype index by Binary_Integer

- Para movernos con estos índices que no tienen que se correlativos necesitamos las siguientes funciones
- Las colecciones tienen asociados unos atributos que permiten realizar ciertas operaciones o conocer ciertos valores

datos.count: Devuelve el número de elementos de la tabla
datos.delete(posición): Borra el elemento con índice indicado en posición.
datos.delete: Borra la tabla completa
datos.delete(posini,posfin): Elimina los elementos que se encuentran entre la posición posini y posfin.
datos.exists(posición): Devuelve true o false dependiendo de si existe o no el índice posición
datos.first: Devuelve el menor índice. El indice donde se encuentra el primer elemento de la colección
datos.last: Devuelve el mayor índice
datos.next(posición): Devuelve el siguiente índice a posición. Si hago datos.next(datos.last) levanta una excepción.
datos.prior(posición): Devuelve el índice anterior a posición.

- Para trabajar con las colecciones es muy habitual usar la salida de una consulta select que devuelva más de una fila y se almacena en una colección
- Para ello se utiliza Bulk Collect

Sintaxis

Select columnas Bulk Collect into colección from restodelaconsulta;

- Las variables con las que trabaja Bulk siempre debe ser de tipo colección y se usa tanto en consultas select como en fetch.