**Colecciones**

Estas estructuras nos permiten manejar conjuntos de datos de manera eficiente, evitando consultas todo el rato a la base de datos.

**Tipos de tablas**

1. **Tablas indexadas:**Sirven para guardar colecciones de elementos usando un indice para acceder a estas.

* **Declaración:**
  + t\_datos\_students define una estructura que almacena filas completas de la tabla students.
  + INDEX BY BINARY\_INTEGER permite usar cualquier número como índice (incluso negativos o no consecutivos).
* **Uso:**
  + Podemos asignar valores directamente usando una clave (v\_students(10000)).
  + No requiere inicialización previa: la asignación crea automáticamente el espacio necesario.
* **Caso de Uso:**
  + En nuestro sistema, esto es ideal para buscar estudiantes por ID rápidamente, como un diccionario o mapa en memoria.

## Tablas anidadas

Son como mini tablas dentro de otra tabla

1. **Declaración:**
   * Parecido a las tablas indexadas, pero sin INDEX BY.
   * Requiere inicialización explícita (t\_course\_list()).
2. **Métodos Clave:**
   * EXTEND: Añade espacio para nuevos elementos.
   * TRIM: Elimina elementos del final.
   * DELETE(n): Elimina un elemento específico (no es consecutivo).
3. **Diferencia Principal:**
   * Lo que le diferencia de las tablas indexadas es que las tablas anidadas pueden almacenarse directamente en columnas de la base de datos. Por ejemplo, podríamos guardar la lista de cursos de un estudiante en una sola columna.

## 3. VARRAYS

1. **Declaración:**
   * VARRAY(5) define un array de un tamaño fijo (máximo 5 elementos).
   * Los índices siempre comienzan en 1 y deben ser consecutivos.
2. **Restricciones:**
   * Si intentamos añadir un sexto elemento, generará un error.
   * Esto lo hace ideal para estructuras con límites predefinidos, como los horarios fijos de un curso.
3. **Persistencia en BD:**
   * Al igual que las tablas anidadas, los VARRAY pueden almacenarse en columnas de tablas.

## 4. Métodos comunes

* **COUNT**: Nos dice cuántos elementos hay actualmente en la colección.
* **EXISTS(n)**: Verifica si un índice existe antes de acceder a él, evitando errores.
* **FIRST/LAST**: Útiles para recorrer todos los elementos de manera segura.
* **LIMIT** (solo VARRAYs): Indica el tamaño máximo permitido.

## Conclusión

| Tipo | Flexibilidad | Almacenable e | Uso Recomendado |
| --- | --- | --- | --- |
| Indexadas | Máxima | No | Búsquedas rápidas por clave |
| Indexadas | Alta | Sí | Listas dinámicas |
| Varrays | Limitada | Sí | Listas de tamaño fijo  (ej: horarios) |

### 