



Tablas internas



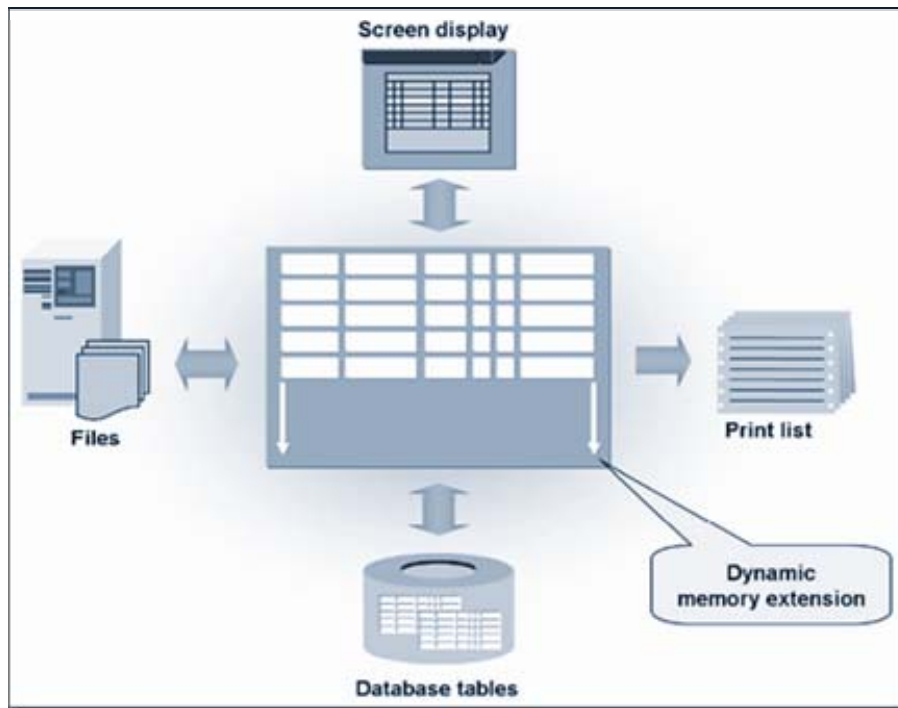
En este capítulo vamos a conocer el uso de tablas internas dentro de un programa. Los distintos tipos de tablas internas existentes, la forma de declararlas y las instrucciones y operaciones disponibles.

Objetivos

- ▶ Conocer los tipos de tablas internas existentes.
- ▶ Aprender la declaración de tablas internas.
- ▶ Aprender las distintas formas de acceso a las tablas.
- ▶ Conocer las instrucciones y operaciones permitidas en las tablas internas.

Lección 1

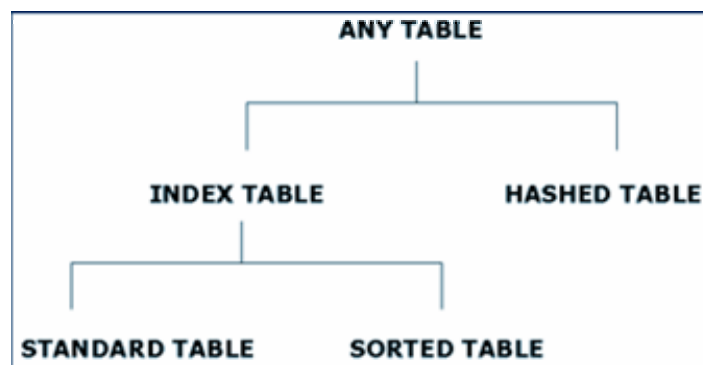
Uso de tablas internas



Las tablas internas son variables que permiten almacenar registros en memoria. Podemos considerar cada componente en una línea como una columna en una tabla interna.

Las tablas Internas son variables dinámicas compuestas de N líneas del mismo tipo. Cualquier tipo de datos ABAP puede ser usado para hacer un tipo de línea de la tabla interna.

Existen distintos tipos de tablas internas:



**Espacio reservado para los elementos de
identificación visual de la Entidad**

ANY TABLE

Es posible usar **ANY TABLE** para definir el tipo de un parámetro de tabla genérico. Las operaciones permitidas para tablas de tipo **ANY TABLE** son todas las operaciones permitidas para **STANDARD**, **SORTED** y **HASHED TABLES**.

Hay que tener en cuenta que no es posible usar índices para acceder a las tablas definidas con este tipo.

INDEX TABLE

Los tipos **STANDARD Y SORTED** pertenecen al tipo de tablas genérico **INDEX TABLE**. Una tabla indexada es aquella a la que puedes acceder mediante un índice. Es posible usar el tipo **INDEX TABLA** para especificar el tipo de un parámetro genérico en un **FORM** o **FUCTION**. El tipo **HASHED** no pertenece al tipo global **INDEX** por lo que no puede ser pasado a un parámetro definido **INDEX TABLE**

STANDARD TABLE

El acceso a una **STANDARD TABLE** es mediante una búsqueda lineal. Esto significa que el tiempo requerido para la búsqueda está relacionado linealmente al número de registros de la tabla.

SORTED TABLE

Al definir una tabla como **SORTED TABLE** esta tabla siempre se guarda con en el orden correcto. El acceso a una **SORTED TABLE** es mediante una búsqueda binaria. Si la clave no es única, el sistema obtiene la entrada con el índice más bajo. El tiempo requerido para el acceso está logarítmicamente relacionado al número de de registros de la tabla.

También es posible acceder a **SORTED TABLES** mediante operaciones de índices. Cuando insertas un registro usando un índice, el sistema chequea para asegurarse que la secuencia de orden ha sido mantenida correctamente. Por esta razón, es más lenta la inserción de entradas que en una **STANDARD TABLE**. Como regla, se debería acceder a **SORTED TABLES** siempre mediante su clave.

HASHED TABLE

Puedes entender una **HASHED TABLE** como un grupo, a cuyos elementos puedes acceder usando su clave única. A diferencia de las **STANDARD Y SORTED TABLES**, no puedes acceder a estas tablas mediante índices. Todos los registros deben tener una clave única. El tiempo de acceso es constante, sin tener en cuenta el número de registros de la tabla.

Solo es posible acceder a las tablas tipo **HASHED** usando las operaciones genéricas de clave u otras operaciones genéricas (**SORT**, **LOOP**..). No están permitidas las operaciones de índices (como **LOOP....FROM INSERT** itab dentro de un **LOOP**)


**Espacio reservado para los elementos de
identificación visual de la Entidad**

	Index tables		Hash table
Table kind	STANDARD TABLE	SORTED TABLE	HASHED TABLE
Access with index 			
Access with key 			
Unique / Non-unique key	NON- UNIQUE	UNIQUE NON-UNIQUE	UNIQUE
Access using	Mainly index	Mainly keys	Keys only

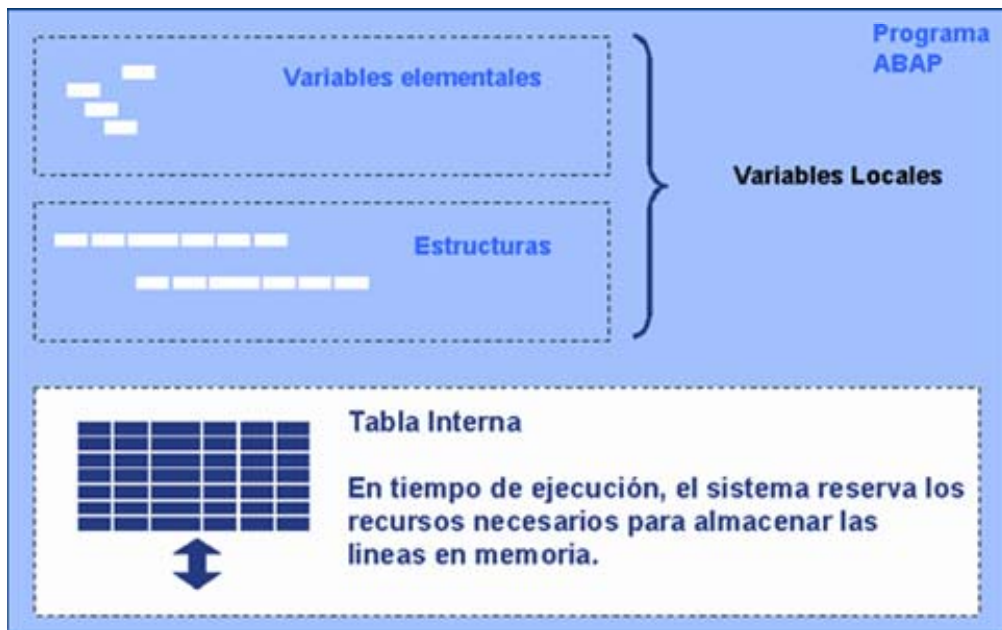
Debemos definir la siguiente información para especificar el tipo de tabla:

- **Tipo de Línea:** Columnas requeridas, sus nombres y tipos, definiendo un tipo de estructura como tipo de línea.
- **Clave:** Qué columnas deben ser columnas clave. En qué orden.
- **Tipo de tabla:** STANDARD, SORTED Y HASHED. El tipo de acceso es lo que determina el tipo de tabla que se ha de elegir.
- **El tipo de acceso** define cómo el sistema en tiempo de ejecución accede a una entrada de tabla. Hay dos tipos de acceso en ABAP: por índice o por clave.

CARRID CONNID DISTANCE				Line type	
Line index				Key fields	Key
1	AA	0017	2,572	Sequence	
2	LH	0400	6,162	Unique / Non-unique key	
3	LH	0402	5,136		
4	QF	0005	10,000		
5	SQ	0866	1,625		
6	UA	0007	2,572		

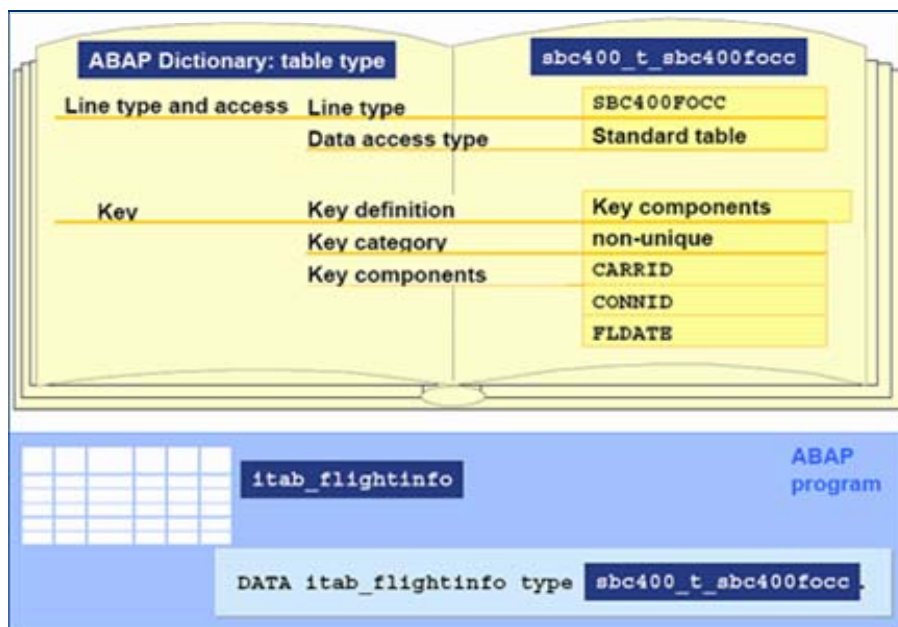


Espacio reservado para los elementos de identificación visual de la Entidad




Podemos definir tablas internas de distintas formas. A continuación vamos a ver algunos ejemplos.

- Declaración de tablas internas con tipos globales:



Espacio reservado para los elementos de identificación visual de la Entidad

- Declaración de tablas internas con tipos locales



```
itab_flightinfo
```

ABAP program

```
TYPES: flightinfo_type
       TYPE STANDARD TABLE OF sbc400focc
       WITH NON-UNIQUE KEY carrid connid fldate.

DATA itab_flightinfo type flightinfo_type.
```

Definición libre de tablas internas

```
TYPES: BEGIN OF str_type,
       carrid TYPE s_carr_id,
       connid TYPE s_conn_id,
       ... ,
       END OF str_type.

DATA itab TYPE STANDARD
          SORTED TABLE OF str_type
          HASHED
          WITH ... KEY ...
```

Local structure type

Internal table

Espacio reservado para los elementos de
identificación visual de la Entidad

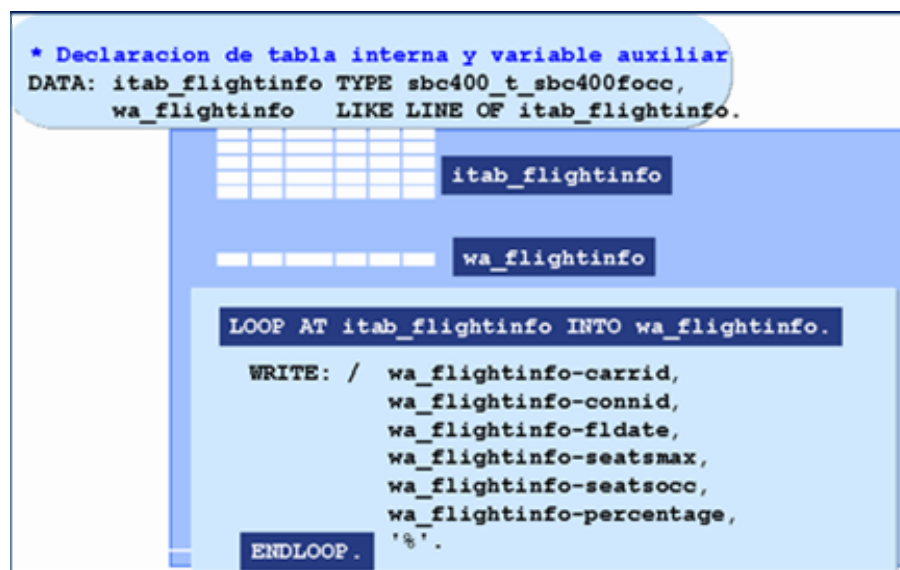
Lección 2

Instrucciones simples para el output de datos

Lectura secuencial de tabla interna

Es posible editar y leer los contenidos de una tabla interna usando un **LOOP**. En el ejemplo indicado, una línea se copia desde la tabla interna **it_flightinfo** a la estructura **wa_flightinfo**. Los campos de la estructura pueden editarse. Podemos crear una lista usando **WRITE**.

Para cambiar los contenidos de la tabla interna, primero hay que cambiar el valor del campo de la estructura dentro del loop y después sobrescribir la línea de la tabla interna con **MODIFY**.



Espacio reservado para los elementos de identificación visual de la Entidad

Lectura indexada de tabla interna

Es posible restringir el acceso a ciertas líneas usando **INDEX**. Esto se usa en tablas standard y ordenadas.

En este ejemplo se muestra la sintaxis para editar el loop solo en las cinco primeras líneas de la tabla interna.

En el segundo ejemplo se lee la tercera línea de la tabla.

```
LOOP AT itab_flightinfo INTO wa_flightinfo
  FROM 1 TO 5.
  WRITE: / wa_flightinfo-carriid,
           wa_flightinfo-connid,
           wa_flightinfo-fldate,
           wa_flightinfo-seatsmax,
           wa_flightinfo-seatsocc,
           wa_flightinfo-percentage,
           ' '.
ENDLOOP.
ENDLOOP.

READ TABLE itab_flightinfo INTO wa_flightinfo
  INDEX 3.
  WRITE: / wa_flightinfo-carriid,
           wa_flightinfo-connid,
           wa_flightinfo-fldate,
           wa_flightinfo-seatsmax,
           wa_flightinfo-seatsocc,
           wa_flightinfo-percentage,
           ' '.
```

Lectura mediante clave en tabla interna.

Podemos restringir el acceso a ciertos valores de los campos clave usando **WHERE**. Es posible hacer esto en todos los tipos de tablas. El acceso usando campos clave en tablas ordenadas y tablas hashed es más eficiente que en tablas estándar.

Todos los campos clave deben estar definidos en accesos clave con **WITH TABLE KEY**.

```
LOOP AT itab_flightinfo INTO wa_flightinfo
  WHERE carriid = 'LH'.
  WRITE: / wa_flightinfo-carriid,
           wa_flightinfo-connid,
           wa_flightinfo-fldate,
           wa_flightinfo-seatsmax,
           wa_flightinfo-seatsocc,
           wa_flightinfo-percentage,
           ' '.
ENDLOOP.

READ TABLE itab_flightinfo INTO wa_flightinfo
  WITH TABLE KEY carriid = 'LH'
                 connid = '0400'
                 fldate = sy-datum.

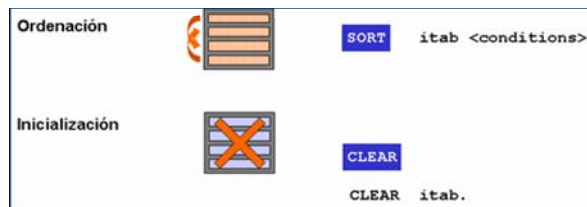
IF sy-subrc = 0.
  WRITE: / wa_flightinfo-seatsmax,
           wa_flightinfo-seatsocc,
           wa_flightinfo-percentage,
           ' '.
ENDIF.
```


**Espacio reservado para los elementos de
identificación visual de la Entidad**

Después de hacer un Read a una tabla interna podemos comprobar si la lectura ha obtenido resultado chequeando el campo SY-SUBRC que en caso positivo tiene el valor 0.

Podemos realizar una serie de operaciones en las tablas internas:

- **SORT** Podemos ordenar tablas en orden creciente o decreciente de cualquier columna.
- **CLEAR** Vuelve el contenido de la tabla a su valor inicial de acuerdo con el tipo de columna.
- **REFRESH** Hace lo mismo que CLEAR.
- **FREE** Borra la tabla interna y libera la memoria.



Lección 3

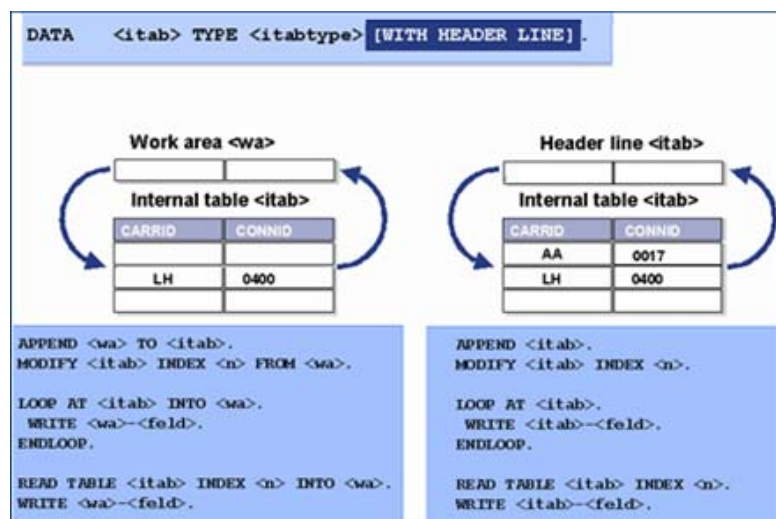
Asignación de valores

Tablas internas con cabecera:

Es posible definir tablas internas con o sin cabecera. Una tabla interna con cabecera consta de un área de trabajo (cabecera) y el contenido real de la tabla, que son diseccionados con el mismo nombre

La manera de interpretar el nombre depende del contexto en el que se usa. Declaramos una tabla interna con cabecera usando **WITH HEADER LINE**.

Para prevenir errores, se recomienda crear tablas internas sin cabeceras. Sin embargo, en las tablas internas con cabecera podemos usar sintaxis más cómoda para ciertas operaciones.



Podemos realizar las siguientes operaciones con tablas internas:

APPEND Añade el contenido de una estructura (que tiene el mismo tipo que el tipo de línea) al final de una tabla interna.

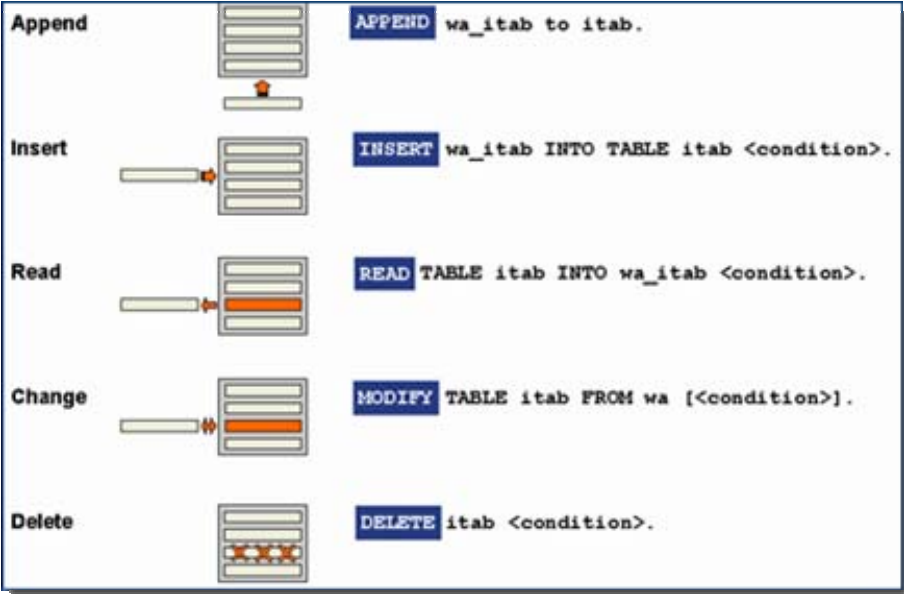
INSERT Inserta el contenido de una estructura que tiene el mismo tipo que el tipo de línea en una tabla interna.

READ Copia el contenido de una línea de la tabla interna en una estructura que tiene el mismo tipo que el tipo de línea.

MODIFY Sobreescribe una línea de la tabla interna con el contenido de una estructura que tiene el mismo tipo que el tipo de línea.

DELETE Borra una línea de la tabla interna.

Espacio reservado para los elementos de identificación visual de la Entidad



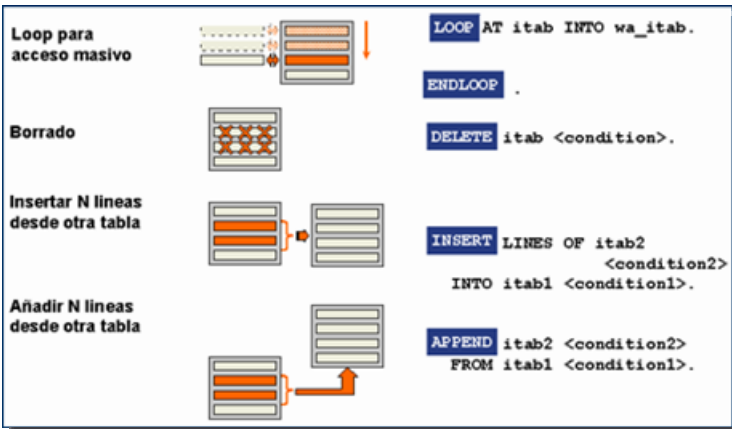
Y para varios registros:

LOOP ... ENDLOOP El **LOOP** sitúa las filas de la tabla interna una por una en la estructura especificada en el **INTO**. La estructura debe tener el mismo tipo que la fila de la tabla interna. Todas las operaciones de registros deben ser ejecutadas dentro del loop.

DELETE Borra las filas de la tabla interna que satisfacen la condición <condition>.

INSERT Copia el contenido de varias filas de una tabla interna en otra tabla interna.

APPEND Añade el contenido de varias filas de una tabla interna en otra tabla interna.



Espacio reservado para los elementos de identificación visual de la Entidad

Es posible ver el contenido de la tabla interna en tiempo de ejecución mediante el debugger. Incluso podemos añadir, modificar o borrar líneas. También podemos ver cual es el contenido de la cabecera de la tabla en cada momento.

