**Tabela Interna Standard**

Criar tabela;

Criar linha na tabela;

Inserir dados:

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s. (aqui já criamos a tabela com a estrutura pré-definida da estrutura, com a coluna de ID e de nome)

DATA ls\_cliente TYPE zcursto\_cliente\_s. (aqui criamos uma linha para a tabela que criamos (l de linha, s de estrutura)).

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’. (aqui inserimos os dados que queremos na linha mas ainda não os pusemos na tabela)

APPEND ls\_cliente TO lt\_cliente. (aqui inserimos a linha com os dados na tabela.)

* Serve para armazenar informação mas não de forma permanente (para armazenarmos de forma permanente devemos usar tabelas de bancos de dados);
* Declara-se da mesma forma que se declara uma variável: lt (l de variável local, t de tabela);
* Zcurso\_cliente\_s é uma estrutura pública que já existe no SAP e que pode ser usada por todos. Se carregarmos 2 vezes no nome reencaminha-nos para a estrutura que nestecaso é composto por ID (int4) e por nome (char100).
* zcursto\_cliente\_s porque estamos a criar uma linha para a tabela que tem como base/tipo esta estrutura.

Limpar dados da tabela:

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

CLEAR lt\_cliente. (limpa todos os dados da tabela. A tabela continua a existir, a estrutura também, só limpa os dados e a linha que inserimos.)

Modificar dados pelo índice da tabela:

(exemplo: em vez de João no cliente 2, quero colocar João Silva)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João Silva’.

MODIFY lt\_cliente FROM ls cliente INDEX 2. (o índice é o número da linha da tabela. NÃO CONFUNDIR COM O ID. No abap o índice começa no 1, não no 0 como no javascript.)

Remover dados pelo índice da tabela:

(exemplo: remover a linha do ID 2)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

DELETE lt\_cliente INDEX 2.

Remover dados por filtro:

(exemplo: remover a linha do ID 1)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

DELETE lt\_cliente WHERE id = 1.

Ler uma linha por índice:

(exemplo: ler a linha do ID 2)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

CLEAR ls\_cliente.

READ TABLE lt\_cliente INTO ls\_cliente INDEX 2.

(primeiro limpamos a estrutura para termos a certeza que os dados que estão a ser lidos são os dados da nossa tabela e não dados que podem estar pré-escritos/definidos na estrutura. Este comando só limpa os dados que possam estar na estrutura, não apaga nada do que inserimos na tabela.

Depois da leitura feita, os dados que foram lidos podem ser vistos na estrutura, é para lá que vão é o que “INTO ls\_cliente” significa. Também podemos escrevê-los na tela, basta escrever o comando WRITE depois da leitura)

Ler uma linha por filtro:

(exemplo: ler a linha do ID 2)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

CLEAR ls\_cliente.

READ TABLE lt\_cliente INTO ls\_cliente WITH KEY id = 2.

Ler várias linhas de uma tabela:

(exemplo: Exibir na tela os dados da tabela)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

LOOP AT lt\_cliente INTO ls\_cliente.

WRITE ls\_cliente-id.

WRITE ls\_cliente-nome.

NEW-LINE.

ENDLOOP.

(este loop vai passar por todas as linhas da tabela, uma a uma, independentemente do tamanho da tabela. Os dados são armazenados na mesma na estrutura - é o que “INTO ls\_cliente” significa, mas neste caso estamos a dizer ao loop para escrever (WRITE) na tela o que ele está a ler sempre que passa por uma linha nova e para dar um espaço entre cada informação que escreve (NEW-LINE)).

Modificar uma linha em loop:

(exemplo: quero colocar um 2 a seguir ao nome de todos os clientes)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 3.

ls\_cliente-nome = ‘Silva’.

APPEND ls\_cliente TO lt\_cliente.

LOOP AT lt\_cliente INTO ls\_cliente.

Ls\_cliente-nome = | { ls\_cliente - nome }2 |.

MODIFY lt\_cliente FROM ls\_cliente.

ENDLOOP.

(como queremos pôr um 2 no fim do nome de cada cliente, concatenamos o 2 e o campo do nome. Depois usamos o comando MODIFY para introduzir as alterações que fizemos na tabela, porque na concatenação alteramos os dados através da estrutura, depois ainda temos que os colocar na tabela - como quando criamos uma nova linha que primeiro inserimos os dados na linha da estrutura e depois é que os passamos para a tabela.)

exemplo: agora quero modificar apenas o noem dos clientes que têm ID 1.

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 3.

ls\_cliente-nome = ‘Silva’.

APPEND ls\_cliente TO lt\_cliente.

LOOP AT lt\_cliente INTO ls\_cliente.

IF ls\_cliente-ID = 1.

Ls\_cliente-nome = | { ls\_cliente - nome }2 |.

MODIFY lt\_cliente FROM ls\_cliente.

ENDIF.

ENDLOOP.

Ordenar dados:

(exemplo: ordenar por ordem decrescente de ID)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 3.

ls\_cliente-nome = ‘Silva’.

APPEND ls\_cliente TO lt\_cliente.

SORT lt\_cliente BY id DESCENDING.

(podemos usar mais do que um critério, por exemplo nome por ordem alfabética descendente e id ascendente: SORT lt\_cliente BY nome DESCENDING id ASCENDING.)

Apagar linhas duplicadas:

(exemplo: remover o cliente de ID 3 que está duplicado)

REPORT ZZADRI\_TABELA\_INTERNA

START-OF-SELECTION

DATA lt\_cliente TYPE STANDARD TABLE OF zcurso\_cliente\_s.

DATA ls\_cliente TYPE zcursto\_cliente\_s.

ls\_cliente-id = 1.

ls\_cliente-nome = ‘José’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 2.

ls\_cliente-nome = ‘João’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 3.

ls\_cliente-nome = ‘Silva’.

APPEND ls\_cliente TO lt\_cliente.

ls\_cliente-id = 3.

ls\_cliente-nome = ‘Silva’.

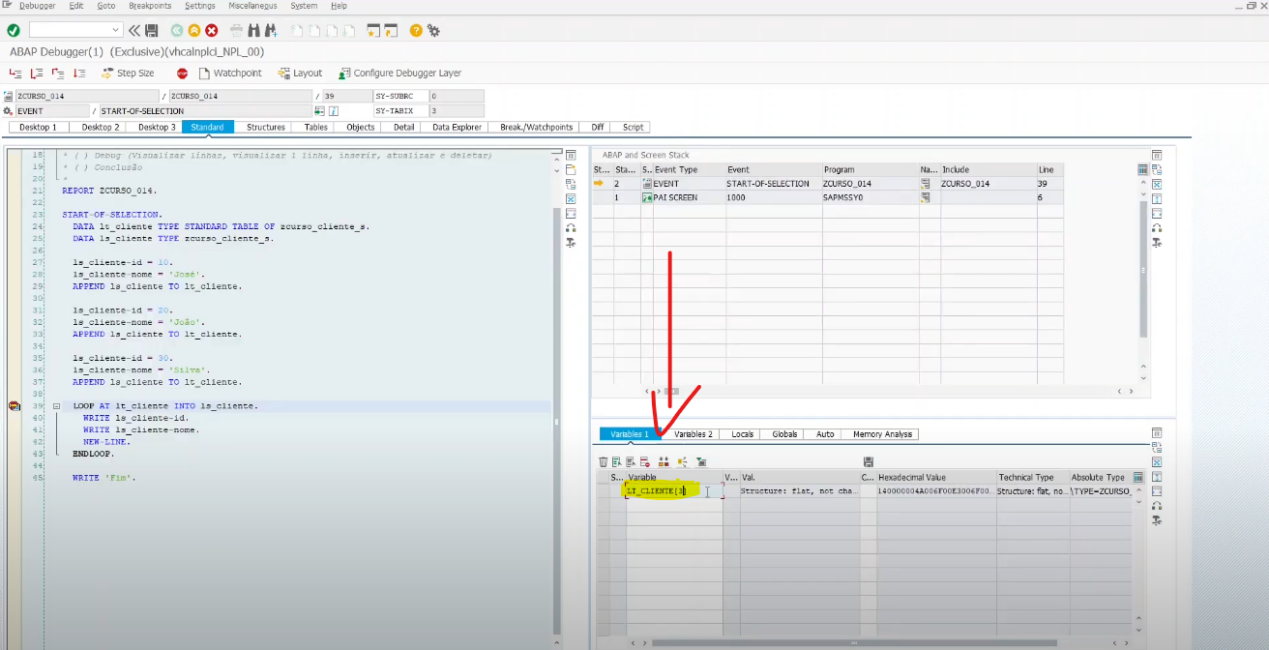
APPEND ls\_cliente TO lt\_cliente.

SORT lt\_cliente BY id ASCENDING. (temos sempre que PRIMEIRO ORDENAR a tabela.)

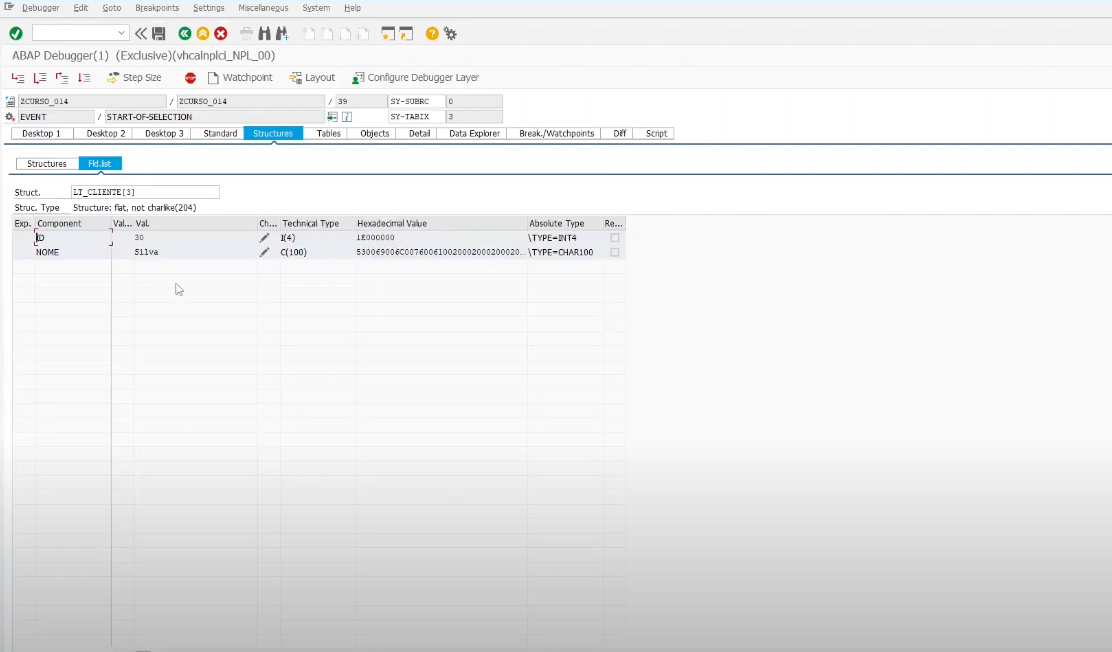
DELETE ADJACENT DUPLICATES FROM lt\_cliente COMPARING id. (neste caso escolhemos apagar através do ID, mas podemos escolher outros critérios.)

Tela de Debug:

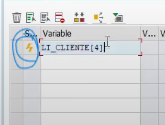
(visualizar linhas, inserir, atualizar e apagar)



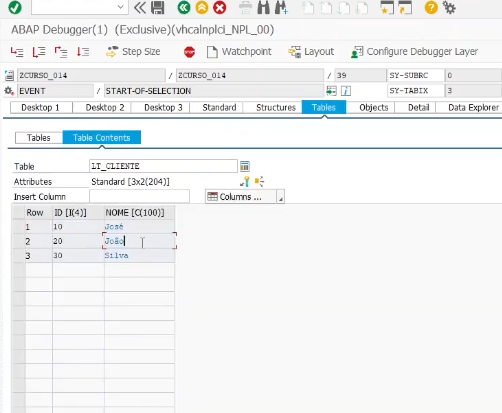
* Inserindo o índice na variável (ex: LT\_CLIENTE[3]), e clicando 2 vezes na variável, vai-me mostrar só a informação dessa linha (se não inserir nenhum valor nos parênteses [ ] significa que queremos ver todo o conteúdo da tabela):



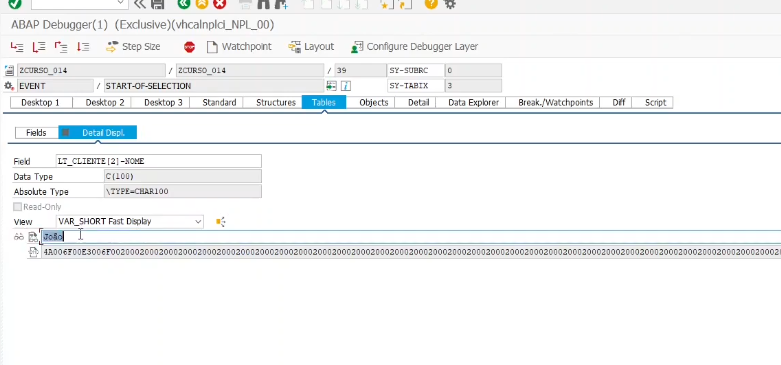
Se introduzirmos um índice que não existe, aparece um sinal a informar que a variável é inválida:



* Quando estamos a ver o conteúdo da tabela podemos modificar os campos lá diretamente:



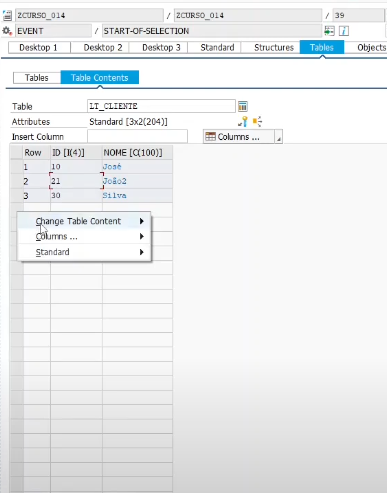
Damos dois cliques em cima do campo que queremos mudar

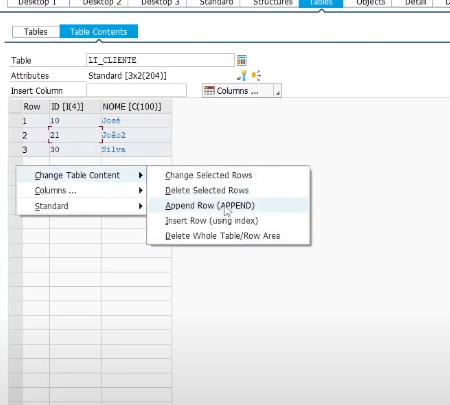


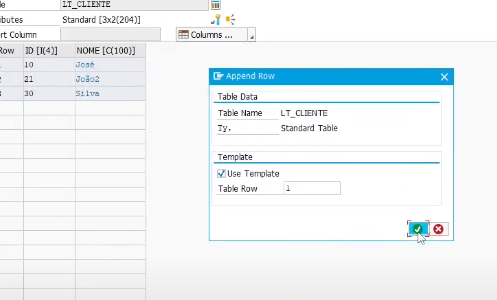
Carregamos no lápis ao lado do campo e escrevemos o que queremos.

Enter e já fica guardado.

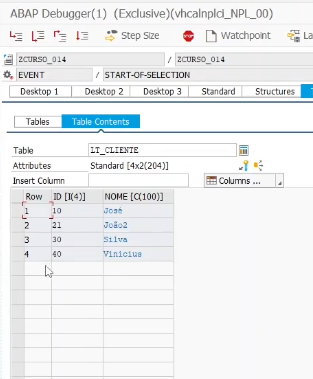
* Podemos inserir linhas:







Aqui selecionamos que estrutura queremos copiar, neste caso vamos copiar a da linha 1



Inserimos os dados que queremos pôr na nova linha e já está.

* Podemos eliminar linhas:

