Entendendo o FOR ALL ENTRIES

BY [FLÁVIO FURLAN](http://abap101.com/author/furlan/" \o "Posts by Flávio Furlan) · FEBRUARY 25, 2011

[Compartilhe!](http://www.addtoany.com/share_save#url=http%3A%2F%2Fabap101.com%2F2011%2F02%2F25%2Fentendendo-o-for-all-entries%2F&title=Entendendo%20o%20FOR%20ALL%20ENTRIES&description=)

Atendendo a pedidos, vamos dar uma olhada no uso da cláusula FOR ALL ENTRIES no SELECT.

Isso é uma particularidade do Open SQL. Se não estou enganado, não há nada parecido em entras linguagens e é uma mão na roda quando queremos fazer um “JOIN” entre uma tabela transparente e uma tabela interna.

Usando

Apenas para entender, vamos fazer um SELECT com JOIN:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728 | REPORT zp\_for\_all\_entries\_join.    TYPES: BEGIN OF ty\_sbook,  carrid TYPE sbook-carrid,  connid TYPE sbook-connid,  fldate TYPE sbook-fldate,  name TYPE scustom-name,  END OF ty\_sbook.    DATA: wa\_sbook TYPE ty\_sbook,  it\_sbook TYPE TABLE OF ty\_sbook.    START-OF-SELECTION.    SELECT sbook~carrid sbook~connid sbook~fldate scustom~name  FROM sbook  JOIN scustom  ON sbook~mandt = scustom~mandt AND  sbook~customid = scustom~id  INTO TABLE it\_sbook  UP TO 200 ROWS.    LOOP AT it\_sbook INTO wa\_sbook.  WRITE: / wa\_sbook-carrid,  wa\_sbook-connid,  wa\_sbook-fldate,  wa\_sbook-name.  ENDLOOP. |

[**view raw**](https://gist.github.com/furlan/839773/raw/ecd3769b2786b75bef0cf194d22726af6d7639fa/zp_for_all_entries_join)[**zp\_for\_all\_entries\_join**](https://gist.github.com/furlan/839773#file-zp_for_all_entries_join) hosted with ❤ by **[GitHub](https://github.com/)**

Nesse caso, temos um JOIN para buscar as informações do nome do cliente que está na tabela SCUSTOM, usando a chave estrangeira CUSTOMID (SCUSTOM-ID). Poderíamos fazer também de maneira diferente, usando a cláusula FOR ALL ENTRIES:

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253 | REPORT zp\_for\_all\_entries.    TYPES: BEGIN OF ty\_sbook,  carrid TYPE sbook-carrid,  connid TYPE sbook-connid,  fldate TYPE sbook-fldate,  customid TYPE sbook-customid,  END OF ty\_sbook.    TYPES: BEGIN OF ty\_scustom,  id TYPE scustom-id,  name TYPE scustom-name,  END OF ty\_scustom.    DATA: wa\_sbook TYPE ty\_sbook,  it\_sbook TYPE TABLE OF ty\_sbook,  it\_sbook\_sel LIKE it\_sbook,  wa\_scustom TYPE ty\_scustom,  it\_scustom TYPE TABLE OF ty\_scustom.    START-OF-SELECTION.    SELECT carrid connid fldate customid  FROM sbook  INTO TABLE it\_sbook  UP TO 200 ROWS.    CHECK it\_sbook[] IS NOT INITIAL.  INSERT LINES OF it\_sbook INTO TABLE it\_sbook\_sel.    DELETE ADJACENT DUPLICATES FROM it\_sbook\_sel COMPARING customid.    SELECT id name FROM scustom  INTO TABLE it\_scustom  FOR ALL ENTRIES IN it\_sbook\_sel  WHERE id = it\_sbook\_sel-customid.    REFRESH it\_sbook\_sel[].    SORT it\_scustom BY id.    LOOP AT it\_sbook INTO wa\_sbook.  READ TABLE it\_scustom INTO wa\_scustom  WITH KEY id = wa\_sbook-customid  BINARY SEARCH.    WRITE: / wa\_sbook-carrid,  wa\_sbook-connid,  wa\_sbook-fldate,  wa\_sbook-customid,  wa\_scustom-name.    ENDLOOP. |

**[view raw](https://gist.github.com/furlan/839854/raw/e84e32711326a6e589725eb38dc1750a99cbe26a/zp_for_all_entries)[zp\_for\_all\_entries](https://gist.github.com/furlan/839854" \l "file-zp_for_all_entries)** hosted with ❤ by **[GitHub](https://github.com/)**

Nesse caso, usamos os dados que já tínhamos na tabela interna it\_sbook para selecionar somente os clientes que estão nela, no campo CUSTOMID.

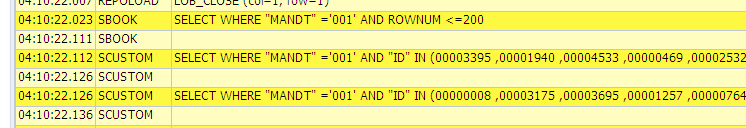
Não estou dizendo que o programa abaixo substitui o JOIN do primeiro exemplo, mas simplesmente para mostrar a utilização do FOR ALL ENTRIES. Em termos de performance, devemos tomar alguns cuidados:

Verificar se a tabela do FOR ALL ENTRIES está preenchida, caso contrário todos os dados da SCUSTOM será retornado

Não usar quando a tabela do FOR ALL ENTRIES for muito grande. Veja a explicação abaixo.

Como Funciona

Como eu disse acima, não temos nada parecido nas diversas variações de SQL, então vamos ver como o Data Base Interface executa o FOR ALL ENTRIES.

[](http://abap101.com/wp-content/uploads/2011/02/FORALLENTRIES_SQL.png)

*FOR ALL ENTRIES SQL Statement*

O Data Base Interface converte para um SELECT com um WHERE usando operador IN listando todas as ocorrências da tabelas do FOR ALL ENTRIES.

Por isso que se se a tabela do FOR ALL ENTRIES estiver em vazia, o Data Base Interface não monta o WHERE, executando um SELECT sem ela, podendo retornar toda a tabela SCUSTOM.

Outra medida que deve ser tomada é a eliminação das entradas repetidas, usando o comando DELETE ADJACENT DUPLICATES. Mas cuidado, pois ao eliminar as entradas repetidas, você irá perder dados da tabela it\_sbook, pois eu criei uma cópia dela somente para executar o SELECT e depois ela é eliminada, usando o comando REFRESH.