

Zumbis no HANA – Parte 3 – Um monte de tralha nova

15/10/2014 10:57

Nesta semana o curso explorou bastante as novas features do OpenSQL (o SELECT vosso de cada dia) e uma nova funcionalidade chamada Core Data Services, onde você poderá criar novas views malucas para consumir dados do BD.

A evolução do OpenSQL

Quando você escreve um SELECT no seu ABAP, você não precisa preocupar-se com o banco de dados que está por debaixo dos panos. Isso não é magia, é o OpenSQL criando uma camada de abstração para você. Com a ascensão da versão 7.4 do ABAP, uma pancada de novas features foram implantadas no OpenSQL, tendo como destaque aquelas que permitem que você jogue mais processamento para o BD.

A partir daqui, você verá que o *SELECT virou bagunça*. Vamos às novas features:

Declarações de tabela direto no SELECT

```
SELECT so_id,
       currency_code,
       gross_amount
FROM snwd_so
INTO TABLE @DATA(lt_result).
```

Sim, a declaração do LT_RESULT está dentro do SELECT. Aliás isso é uma nova coisa da 7.4, com o DATA(algo) você pode declarar o que precisa e sair usando.

Usar variáveis como colunas no retorno do SELECT

Precisa preencher uma coluna com um Valor Fixo? Com um 'X'? Seus problemas acabaram:

```
SELECT so~so_id,
       'X' AS literal_x,
       42 AS literal_42
FROM snwd_so AS so
INTO TABLE @DATA(lt_result).
```

A tabela LT_RESULT teria um campo preenchido todo com 'X', outro todo com 42. E aquela vírgula **não** está errada.

Mas daí você fala "putz, e se eu precisasse preencher com 'X' dependendo de uma condição?"

Usar CASE dentro dos campos selecionados

```
SELECT so_id,
       CASE delivery_status
       WHEN ' ' THEN 'OPEN'
```

```

    WHEN 'D' THEN 'DELIVERED'
    ELSE delivery_status
    END AS delivery_status_long
FROM snwd_so
INTO TABLE @DATA(lt_simple_case).

```

Sabe como é, o banco de dados que se vire!

Quer fazer contas no SELECT? Também pode

```

SELECT ( 1 + 1 ) AS two,
       ( @lv_discount * gross_amount )
       AS red_gross_amount,
       CEIL( gross_amount )
       AS ceiled_gross_amount
FROM snwd_so
INTO TABLE @DATA(lt_result).

```

O banco de dados que se vire elevado ao quadrado.

Ah, já que pode tudo, concatena também nessa budegas

```

SELECT company_name && ' (' && legal_form && ' )'
FROM snwd_bpa
TABLE @DATA(lt_result).

```

'&&' é uma nova-mas-não-tão-nova forma de concatenar strings. Tente aí no seu ambiente 'texto = 'será que' && 'funfa?'; vai que cola.

Tem mais um monte de parafernália, esses exemplos são só o começo. Se você quiser ver a lista completa, acesse [esse blog](#) de um dos papas do ABAP. É tanta coisa que dava para fazer um ABAPZombie novo só para essa 7.4 .

Eu gostei, mas tenho calafrios só de pensar nas bizarrices que vão aparecer. Certamente vai aparecer algum louco para jogar a lógica inteira dentro do SELECT, o que será correto do ponto de vista do "code-pushdown". Mas eu temo pela manutenção desses códigos e pela bizarrice de comentários, como explicado pela Dai [neste post](#).

Já ouvi muitas vezes que "entender um JOIN é mto complexo, usa FOR ALL ENTRIES e pronto", portanto, imagino qual será a aderência a essas lógicas que tendem a deixar os SELECTs bem mais complicados de serem entendidos. Sinceramente, duvido que a comunidade em geral aplique esses conceitos no curto prazo... mas eu não sou a mãe diná e posso estar errado.

Não achei muito intuitiva a leitura do código com essas novas features. Os exemplos são muito simplistas e consideram pequenos cálculos localizados. Imagino como ficaria um SELECT na BSEG cheio de condições e contas retardadas, que é a realidade do dia-a-dia.

Mas não priemos cânico, tem uma coisa que pode ajudar...

ABAP CDS – Core Data Services

Esse negócio de jogar a lógica para dentro do banco é *serious business*. Tanto que a SAP criou uma nova forma de definir views, através de uma espécie de script criado de dentro do ABAP. Dentro do Eclipse, **e somente dentro do Eclipse**, você terá a opção de criar um novo tipo de objeto chamado “DDL Source”, que nada mais é do que um script que gera uma view dentro do dicionário de dados.

O lance é que essa história de code-pushdown pode gerar lógicas que podem ser reutilizadas dentro de diversas aplicações dentro do ABAP, e pelo CDS você conseguirá reutilizar o seus SELECTs e JOINs cheio de regras malucas em programas distintos.

Dê uma olhada em como é o código dentro de um DDL:

```
@AbapCatalog.sqlViewName: 'ZDDL$CDS_00'
define view zcdsv_simple as
  select from snwd_so
  {
    key so_id as order_id,
    currency_code,
    gross_amount
  }
```

Para consumir essa CDS view, você acessa como se fosse um SELECT normal:

```
DATA lt_cds TYPE STANDARD TABLE OF zcdsv_simple.
SELECT *
FROM zcdsv_simple
INTO TABLE @lt_cds.
```

Mas esse negócio de CDS view pode ficar beeeeeeeem mais complexo. Veja só um exemplo mais trabalho, com associações e regras de negócio:

```
@AbapCatalog.sqlViewName: 'ZDDL$CDS_32B'
define view zcdsv_filter_example_vov as
  select from snwd_so as so
    association [1] to snwd_bpa as business_partners
      on so.buyer_guid = business_partners.node_key
    association [0..1] to zcdsv_filter_example_base as invoice_headers
      on so.node_key = invoice_headers.order_guid
  {
    key so.so_id as order_id,
    -- value 01 means customer
    business_partners[ bp_role = '01' ].company_name as customer_name,
    -- filter 1..n association on first position
    invoice_headers.invoice_items[1: inv_item_pos = '000000010'].currency_code,
    invoice_headers.invoice_items[1: inv_item_pos = '000000010'].gross_amount
  }
  where invoice_headers.header_guid is not null
```

E no seu código, consumir a view continua sendo algo bem simples:

```
DATA lt_cds TYPE STANDARD TABLE OF zcdsv_filter_example_vov.
SELECT *
FROM zcdsv_filter_example_vov
INTO TABLE @lt_cds.
```

Pode parecer que um monte de coisa redundante, já que você consegue atingir os mesmos resultados usando as novas features do OpenSQL. Mas pense no reuso: qualquer programa que precise aplicar aquela regra de negócio,

abusando do processamento do banco de dados, pode consumir os dados da mesma CDS view.

Essa divisão de responsabilidades também é algo positivo. Como eu disse, tenho calafrios só de pensar no caos que alguns programadores poderão criar com selects monstruosos. Ter um tipo específico de objeto, com uma linguagem e regras específicas pode beneficiar muito no entendimento geral da aplicação.

Vale mencionar também que essas CDS views não são uma feature nativa do HANA. Ele existirá e funcionará independente do banco de dados conectado ao SAP, o que também acredito ser algo positivo.

Ou você acha que é só a SAP que está nessa onda do banco de dados in-memory?

Há muito para falar sobre o CDS, mas acredito que sai do escopo deste post. Ficou interessando em entender um pouco mais sobre esse novo tipo de view? [Este link](#) é um bom começo.

Pensamentos Rândomicos

Estamos quase no fim, semana que vem é a última semana do curso. Acredito que iremos explorar um pouco mais a forma de criar procedures do HANA direto do ABAP. Fiquem ligados para a última parte deste guia!

Fica aqui minha dúvida filosfal da semana:

- Ok galera, esse monte de coisas bacanas é **realmente** bacana. Mas se com o HANA o SAP passa a ficar mais rápido sem muitas alterações (desde que o programa já siga as guidelines clássicas de performance), será mesmo que o povo irá ligar para implantar esse monte de coisas nos códigos Z? Entendo que em aplicações mais robustas esse approach do code-pushdown faça todo o sentido, mas o ABAP do dia-a-dia, que precisa fazer um relatório genérico e idiota não vai aplicar absolutamente nada dos novos conceitos. Só o tempo dirá quais das novas ferramentas realmente serão aderidas pela comunidade. ABAP Unit por exemplo está aí há anos, e ninguém dá a mínima... isso me faz imaginar qual será o novo "ABAP Unit" dessa leva de features do code-pushdown.

Até a próxima semana!

Abraços a todos que vão tacar toda a lógica no SELECT memo e que se dane!a

Comentários

Breno — 07/04/2017 17:53

Olá Maurício, parabéns pelos excelentes posts sobre o assunto e suas dúvidas filosófais, acredito que são compartilhadas por todos! rs.

Quanto mais leio sobre o assunto SAP HANA, mais vejo que a SAP está seguindo o "caminho normal". Explicarei.

Se analisarmos a forma como nasceu a linguagem ABAP, uma linguagem de extensão do sistema ERP para geração de relatórios, vemos que cada vez mais, a ideia da SAP hoje é a de criar um ambiente de desenvolvimento avançado e competitivo ao redor de sua plataforma. Quem à 45 anos atrás imaginaria um cenário de big data, data scientist, IoT, Industrie 4.0, etc...?

Quase tudo que vemos à respeito do HANA nos remete ao que já existe no mundo exterior (mundo fora do SAP). Deixar o banco se virar, criar views, stored procedures, selects cabulosos entre outros, são cenários comuns para

praticamente todos os desenvolvedores que trabalharam algum dia com banco de dados. (Não entrarei no mérito do DB colunar). Talvez, o fato disso não ter acontecido antes se deu por estratégia ou para manter a abstração com muitos bancos de dados que a ferramenta poderia trabalhar aliada às formalidades alemãs, rígidas.

Acredito, que a SAP esteja “correndo atrás do prejuízo” (calma ABAPs afoitos, WAIT UP TO 10 SECONDS.) e temos que acompanhar! É possível enxergar as oportunidades das novas tecnologias em um mundo onde a informação tem que ser minerada de dados que somam montantes extraordinários e para isso, precisam de pessoas especializadas (ABAPs com a cabeça aberta) que munidas das ferramentas mais atuais conseguirão manter a empresa no topo por mais 45 anos pelo menos! Muitas oportunidades estão nascendo agora, mas os profissionais terão que abrir a mente sem preconceito, e se livrar de antigos cacoetes para descobrir esse novo mundo que ela está criando.

Um forte abraço! Vamos estudar!

Custodio — 16/10/2014 09:01

Fala Maurício,

Mais um grande post, estou curtindo o “resumo da semana” que você está fazendo!

Só um pequeno adendo, o link que você colocou pras novidades do ABAP 740 já não é tão novo (740 SP5). O novíssimo é esse aqui (740 SP8): <http://scn.sap.com/community/abap/blog/2014/10/13/abap-language-news-for-release-740-sp08>

Abraço!

Daniel Jesus — 15/10/2014 14:53

Essa do CDS é legal pacas heim, e de quebra obriga os caboclos a usar o Eclipse. O tradicional report podia virar um tipo de objeto ALV+DDL já pensou que irado?

Diego Henrique Gomes — 15/10/2014 11:40

Sempre agregando!!! valeu pelo post!!! Tenho medo se vão utilizar essas armas para o bem ou para o mal, vamos aguardar...