

HANA 1909 - ABAP



GEOSystem

We Make IT Value



Agenda do Treinamento

Tema
Hana - Arquitetura
HANA Studio/Eclipse - Configuração
HANA – Conceitos Iniciais
HANA – Novo Open SQL
HANA – SQL Nativo
HANA – ADBC
HANA – CDS View
HANA - Procedures AMDP
HANA – Visões
HANA - Novo ALV
Conclusão

Objetivo

Este curso servirá para você:

- Conhecer o ABAP HANA;
- Familiarizar-se com os principais conceitos HANA;
- Familiarizar-se com o Hana Studio ou Eclipse;
- Aproveitar toda capacidade da arquitetura HANA;



Introdução ao Hana

Introdução ao Hana

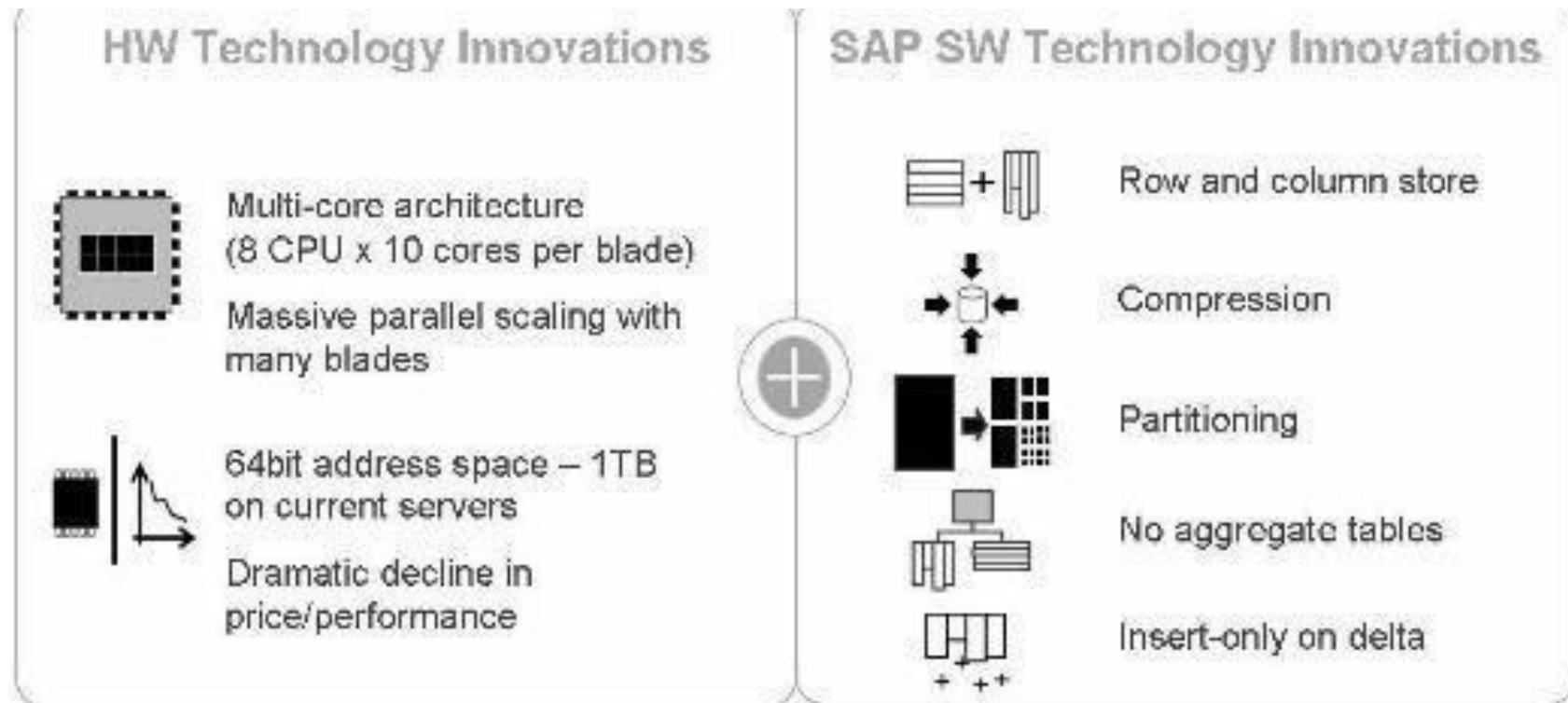
- Database é uma plataforma de gerenciamento de dados centralizada na memória principal. O banco de dados SAP HANA é executado no SUSE Linux Enterprises Server e baseia-se na linguagem C ++.

Introdução ao Hana

- HANA é a nova plataforma da SAP que combina hardware de alta performance com software que consiga utilizar tal hardware.
- ✓ A SAP desenvolveu HANA para atender os seguintes objetivos:
 - ✓ Otimizar aplicações novas e/ou já existentes;
 - ✓ Escalabilidade;
 - ✓ Combinação de paradigmas para armazenamento de dados;
 - ✓ Compatibilidade;
 - ✓ Suporte a Cloud;
 - ✓ Execução lógica dentro da camada de banco de dados.

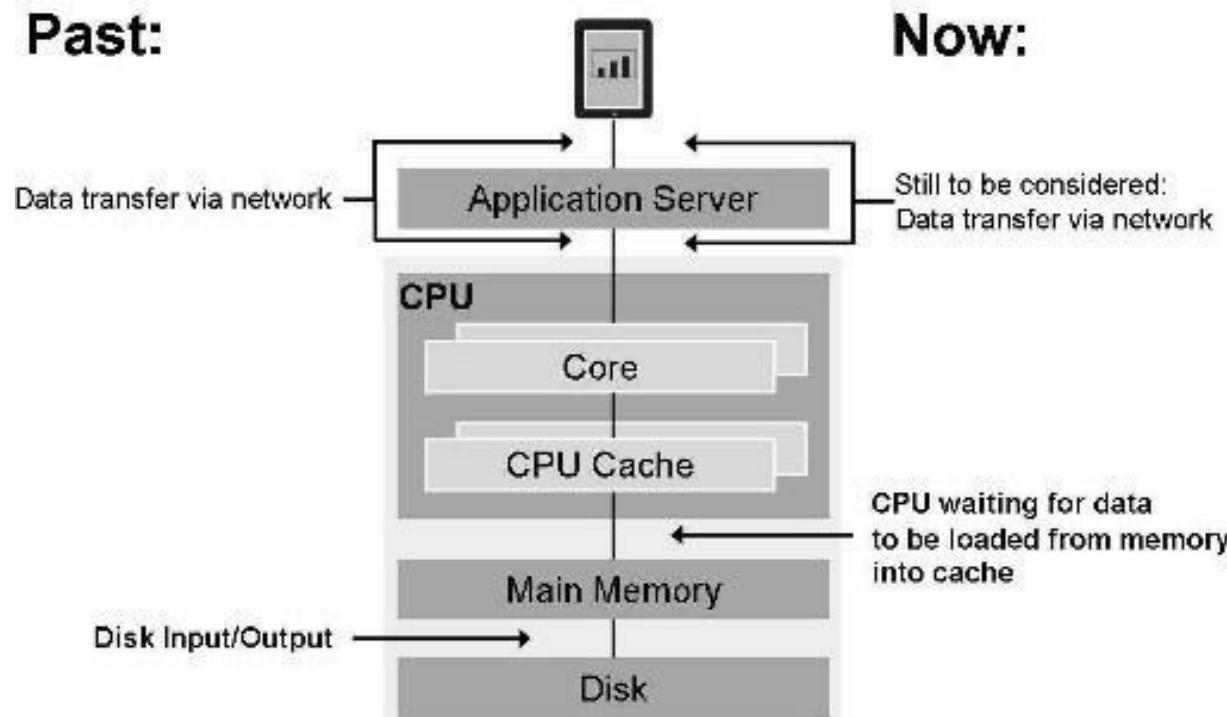
Introdução ao Hana

■ Inovações do HANA:



Introdução ao Hana

- Nível de hardware - Comparaçõao:



Introdução ao Hana

■ Nível de Software – Armazenamento de Dados

O SAP HANA pode ser usado para fins de OLAP (análise on-line) e OLTP (transação on-line) em um único banco de dados.

Com a crescente necessidade de análise de grandes volumes de dados no menor tempo possível, bancos de dados baseados em uso intensivo de memória RAM (In-Memory) estão cotados como sendo a solução mais viável.

A principal diferença entre os bancos de dados In-Memory e os “tradicionais” está na forma de armazenamento dos dados, *Column Store vs Row Store*

Introdução ao Hana

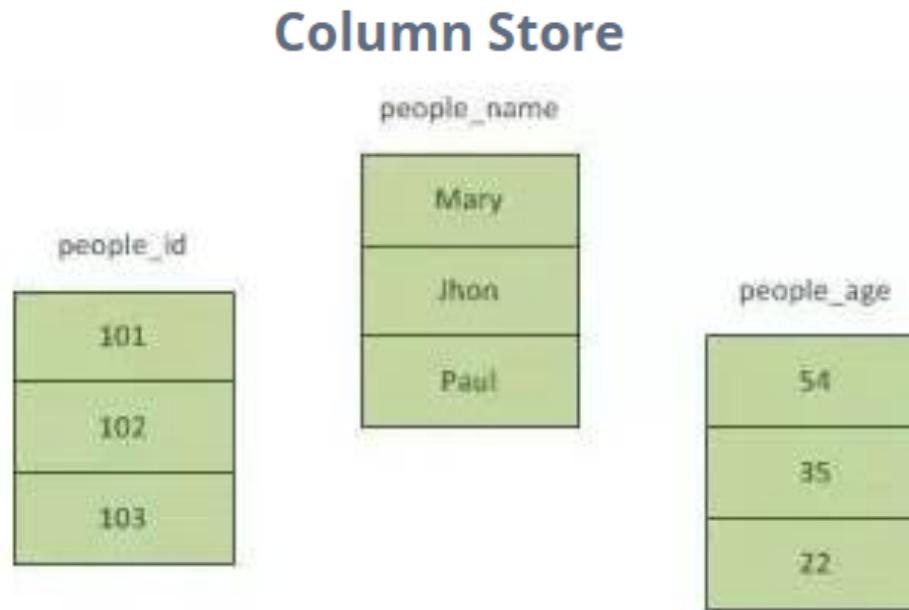
- Nível de Software – Armazenamento de Dados

Row Store

people_id	people_name	people_age
101	Mary	54
102	Jhon	35
103	Paul	22

Introdução ao Hana

- Nível de Software – Armazenamento de Dados



O armazenamento de colunas oferece bom desempenho para operações de gravação e, ao mesmo tempo, otimiza a operação de leitura.

Introdução ao Hana

Table

Country	Product	Sales
US	Alpha	3.000
US	Beta	1.250
JP	Alpha	700
UK	Alpha	450

Row Store

Row 1	US
	Alpha
	3.000
Row 2	US
	Beta
	1.250
Row 3	JP
	Alpha
	700
Row 4	UK
	Alpha
	450

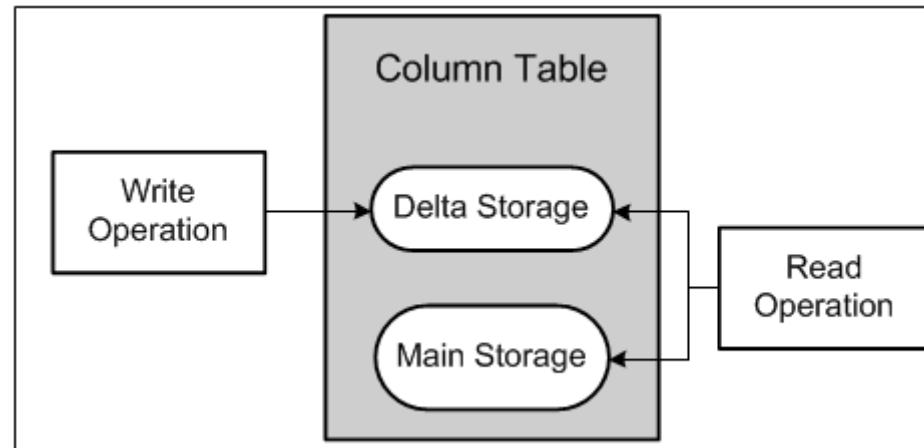
Column Store

Country	US
	US
	JP
	UK
Product	Alpha
	Beta
	Alpha
	Alpha
Sales	3.000
	1.250
	700
	450

Introdução ao Hana

■ Nível de Software – Armazenamento de Dados

Ler e escrever o desempenho da operação otimizado com menos de duas estruturas de dados.



Introdução ao Hana

■ Nível de Software – Compressão de dados

Column "fname"	
recID	fname
...	...
39	John
40	Mary
41	Jane
42	John
43	Peter
...	...



Dictionary for "fname"	
valueID	Value
...	...
23	John
24	Mary
25	Jane
26	Peter
...	...

Attribute Vector for "fname"	
position	valueID
...	...
39	23
40	24
41	25
42	23
43	26
...	...

Introdução ao Hana

■ Nível de Software – Compressão de dados

Attribute	# of Distinct Values	Size
first name	5 million	49 Byte
last name	8 million	50 Byte
gender	2	1 Byte
country	200	49 Byte
city	1 million	49 Byte
birthday	40 000	2 Byte
	Sum	200 Byte

Vamos fazer de conta que nossa tabela tenha 8 bilhões de linhas e o comprimento do registro seja de 200 bytes, então teremos:

$$8.000.000.000 \times 200 = 1.6 \text{ TB}$$

Peguemos o primeiro campo “first name”, para conseguir substituir os dados por bits tem que saber quantos bits são necessários para 49 bytes de dados e assim por diante, conforme tabela acima, para isso usamos uma função logarítmica de base binária:

$$\text{Log}_2(5.000.000) = 23$$

Fazendo este cálculo temos 23 bits que são necessários para armazenarmos os mesmos 49 bytes, agora vamos fazer uma comparação:

Antes:

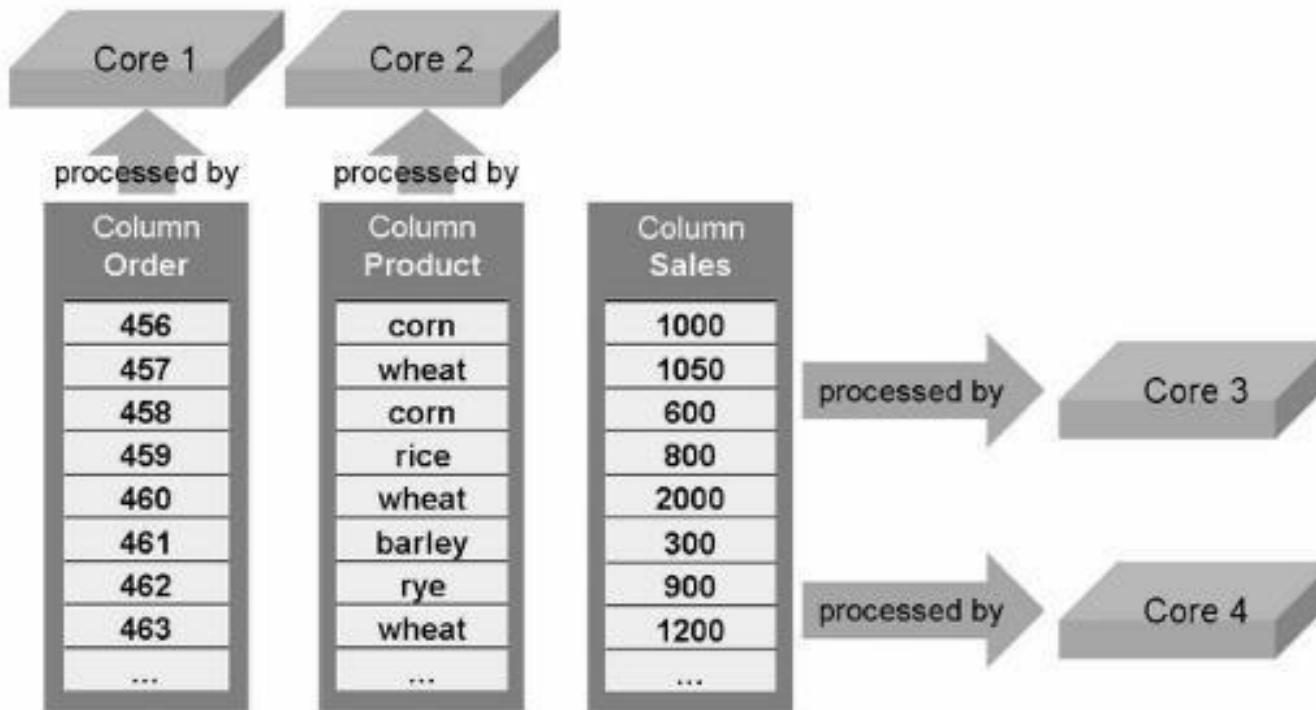
$$8.000.000.000 \times 49 \text{ Bytes} = 392.000.000.000 \text{ Bytes} / 1024 / 1024 / 1024 = 365.1 \text{ GB}$$

Depois:

$$8.000.000.000 \times 23 \text{ Bits} = 184.000.000.000 \text{ Bits} / 8 / 1024 / 1024 / 1024 = 21.4 \text{ GB}$$

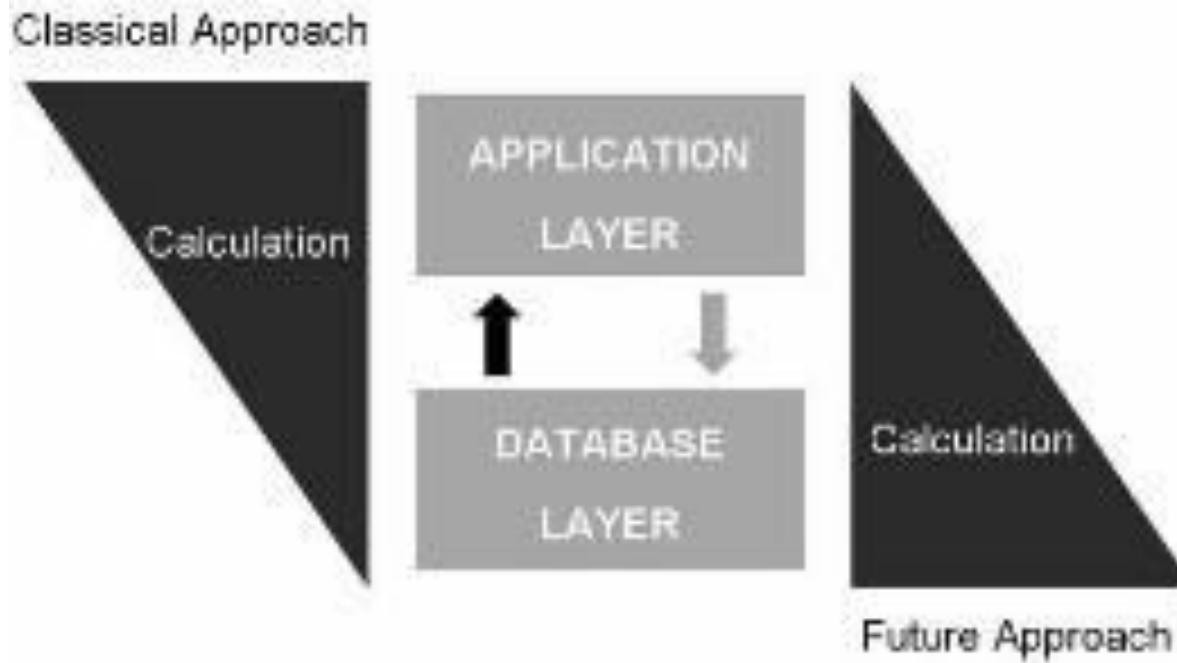
Introdução ao Hana

■ Nível de Software – Particionamento



Introdução ao Hana

- Mudanças Conceituais para desenvolvedor:



No Classical Approach, os aplicativos atuais executam muitas operações com intensidade de dados na camada de aplicativos, enquanto na abordagem SAP HANA, os aplicativos de alto desempenho delegam operações intensas de dados à plataforma in-memory.

Introdução ao Hana

- Mudanças a partir do ABAP 7.4 aprimoraram alguns pontos para adequar ao SAP Hana:
 - Foi ajustado o For All Entries para manter performático em plataforma HANA
 - Open SQL está sendo estendido constantemente para atender ao SAP HANA.
 - Select's com case, casting de dados, maior número de joins
 - Forma de integrar procedure criada no banco de dados com o código ABAP
 - Pode ser visualizada views baseada em colunas do SAPHANA pela SE11
 - Foram adicionadas novas ferramentas de análise de performance e erros

Introdução ao Hana

- Mudanças a partir do ABAP 7.4
 - Foram adicionadas novas ferramentas de análise de performance e erros
 - Alguns componentes reutilizáveis(Alv, por exemplo) foram refeitos para otimizar o SAP HANA
 - Foi adicionado uma nova aba nas propriedades da tabela pela SE11, chamada “Propriedades DB”.
 - Nela é possível, alterar a forma de armazenamento da tabela: Linha, coluna ou baseado em objetos

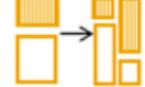
Programação em HANA – Conceitos iniciais

Programação em HANA

■ Melhores Práticas

- Antes do HANA, existem 5 regras que a SAP define como “regras de ouro”, que apoiam qualquer aplicação ABAP feita.
- São elas:

Programação em HANA

Icon	Rule	Details / Examples	Shift in priorities for SAP HANA
	Keep the result sets small	<ul style="list-style-type: none"> Do not retrieve rows from the database and discard them on the application server using CHECK or EXIT, e.g. in SELECT loops Make the WHERE clause as specific as possible 	
	Minimize amount of transferred data	<ul style="list-style-type: none"> Use SELECT with a field list instead of SELECT * in order to transfer just the columns you really need Use aggregate functions (COUNT, MIN, MAX, SUM, AVG) instead of transferring all the rows to the application server 	
	Minimize the number of data transfers	<ul style="list-style-type: none"> Use JOINs and / or sub-queries instead of nested SELECT loops Use SELECT ... FOR ALL ENTRIES instead of lots of SELECTs or SELECT SINGLES Use array variants of INSERT, UPDATE, MODIFY, and DELETE 	
	Minimize the search overhead	<ul style="list-style-type: none"> Define and use appropriate secondary indexes 	
	Keep load away from the database	<ul style="list-style-type: none"> Avoid reading data redundantly Use table buffering (if possible) and do not bypass it Sort Data in Your ABAP Programs 	

Programação em HANA

■ Melhores Práticas - Regras que se tornaram mais importante com HANA

- Keep results sets small e Minimize amount of transferred data:
 - Regra continua válida para o HANA
 - Deve-se reduzir o máximo possível o tamanho de memória usado para transferir dados do banco de dados para o servidor
 - Para isso utilizar condição Where condizente com a regra de negócio, e também NUNCA utilizar SELECT *
 - Atenção especial para o SELECT *, pois como vimos anteriormente, o HANA por tratar tabelas com base em colunas, tem-se um custo muito maior para busca todos os campos de 1 registro
 - Com o HANA, evite buscar registros desnecessários e depois descartar no programa através de CHECK ou EXIT.

Programação em HANA

■ Melhores Práticas – Regras que se tornaram mais importante com HANA

- Minimize the number of database accesses:
 - Regra continua válida para o HANA
 - O SAP tanto HANA quanto o clássico, tem custo para cada conexão SQL que envolve: parseamento, execução e etc.
 - Portanto a regra de ouro, diz para evitar chamar a mesma tabela várias vezes.
 - No HANA esse item se torna mais importante. Devido a tecnologia IN-MEMORY, podemos realizar menos chamadas ao Banco de Dados do que no ambiente Clássico. Utilize join's, queries no lugar de select's alinhados
 - Tentar sempre fazer com que as linhas e colunas necessárias sejam feitas em uma chamada SQL
 - Use For All Entries no lugar de vários select's

Programação em HANA

■ Melhores Práticas – Regras que mudaram com HANA

- Minimize search overhead
 - No SAP clássico, as buscas de dados devem ser feita preferencialmente com chave primária ou através de índice secundário
 - No HANA, full tables can em tabelas não são demorados, portanto clausulas where que não estejam dentro de índices podem ser realizadas sem preocupação
 - Obs: Cada índice secundário consome memória e causa lentidão na inserção de registros. Com servidor HANA, índices secundários podem ser removidos.

Programação em HANA

■ Melhores Práticas – Regras que mudaram com HANA

- Keep unnecessary load away from DB
 - No SAP clássico, o Banco de dados era utilizado apenas para seleção dos mesmos. A manipulação dos dados era feito dentro do programa ABAP
 - No HANA, manipulação de dados é muito mais rápido dentro do banco de dados, portanto quanto mais coisa for feita no SELECT ou em procedure maior ganho o programador terá

Programação em HANA

■ Exemplo 1

- Descrição do Exemplo: Cliente solicitou saber quanto estava faturando por avião,
- Pensando da forma HANA, um exemplo de código ficaria da seguinte forma:

```
TYPES:  
BEGIN OF ty_sflight,  
    agencynum TYPE sbook-agencynum,  
    planetype TYPE sflight-planetype,  
    forcuram    TYPE sbook-forcuram,  
    loccuram    TYPE sbook-loccuram,  
END OF ty_sflight.  
  
DATA: t_sflight TYPE TABLE OF ty_sflight.  
  
SELECT a~agencynum b~planetype sum( a~forcuram ) sum( a~loccuram )  
    FROM ( sbook as a INNER JOIN sflight as b ON a~carrid = b~carrid AND  
            a~connid = b~connid AND  
            a~fldate = b~fldate )  
    INTO TABLE t_sflight  
    GROUP BY a~agencynum b~planetype  
    ORDER BY a~agencynum.  
    IF sy-subrc EQ 0.  
  
ENDIF.
```

Programação em HANA

■ Exemplo 2:

- Com a mesma descrição do exemplo, criamos um programa com melhores práticas voltados para o SAP clássico, e analisamos nos dois ambientes.

```
SELECT carrid connid fldate agencynum forcuram loccuram
      FROM sbook INTO TABLE t_sbook.
      IF sy-subrc EQ 0.
        SORT: t_sbook BY carrid connid fldate.
        t_sbook_aux[] = t_sbook.
        DELETE ADJACENT DUPLICATES FROM t_sbook_Aux COMPARING carrid connid fldate.
        SELECT carrid connid fldate planetype FROM sflight INTO TABLE t_sflight
          FOR ALL ENTRIES IN t_sbook_aux
          WHERE
            carrid = t_sbook_aux-carrid AND
            connid = t_sbook_aux-connid AND
            fldate = t_sbook_aux-fldate.
      ENDIF.
      LOOP AT t_sbook INTO w_sbook.
        CLEAR: w_collect.
        READ TABLE t_sflight INTO w_sflight
        WITH KEY
          carrid = w_sbook-carrid
          connid = w_sbook-connid
          fldate = w_sbook-fldate BINARY SEARCH.
        IF sy-subrc EQ 0.
          w_collect-planetype = w_sflight-planetype.
        ENDIF.
        w_collect-agencynum = w_sbook-agencynum.
        w_collect-forcuram = w_sbook-forcuram.
        w_collect-loccuram = w_sbook-loccuram.
        COLLECT w_collect INTO t_collect.
      ENDLOOP.
      SORT: t_collect by agencynum.
```

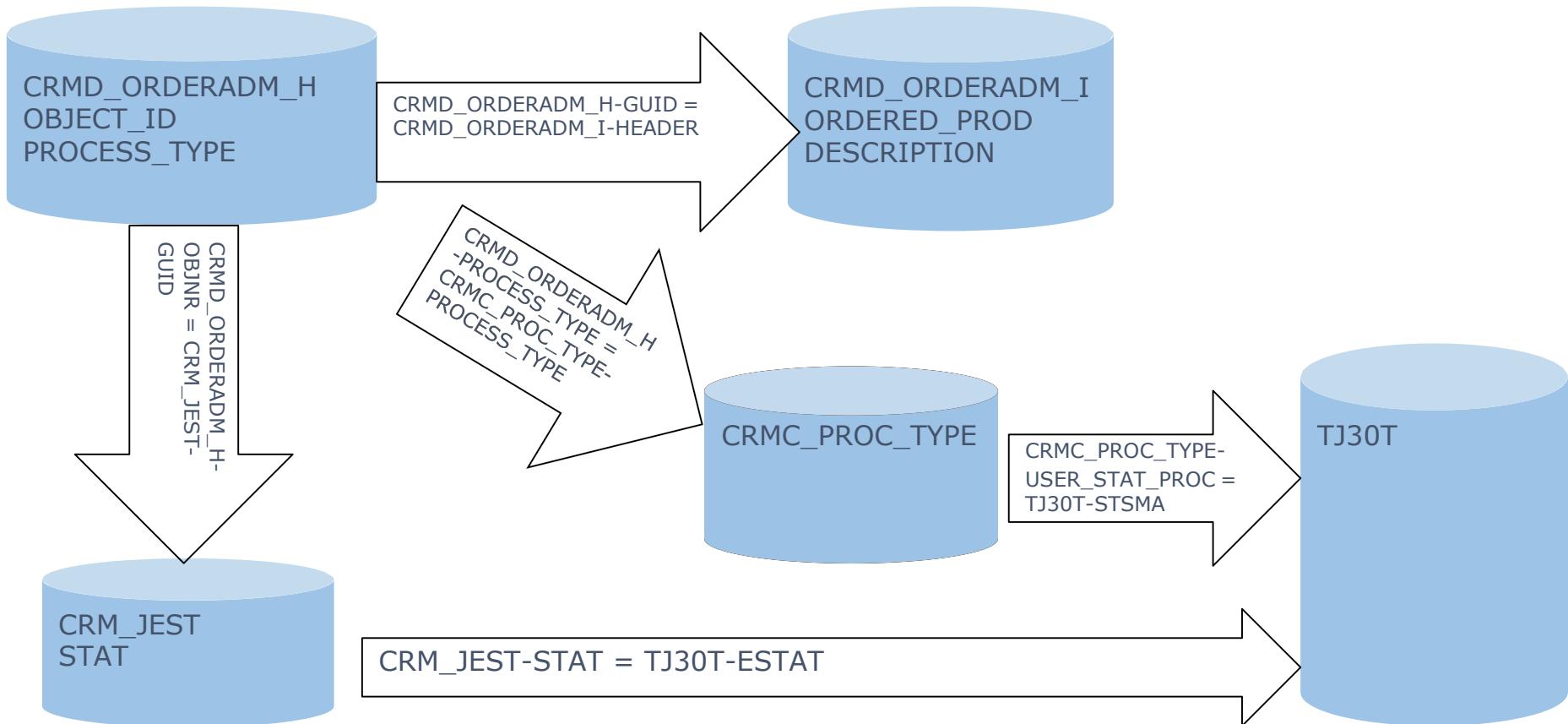
Programação em HANA

■ Exemplo 3

- Descrição: Cliente deseja um relatório com todas as reclamações abertas, informando o tipo da reclamação. Relatório conterá as seguintes informações:
 - Nº da reclamação -> crmd_orderadm_h-object_id,
 - Tipo da reclamação -> crmd_orderadm_h-process_type,
 - Código do status da reclamação -> crm_jest-stat,
 - Descrição do status da reclamação -> tj30t-txt30,
 - Produtoreclamado->crmd_orderadm_i-ordered_prod,
 - Descrição do Produto->crmd_orderadm_i-description,

Programação em HANA

■ Exemplo 3



Programação em HANA

■ Exemplo 3

- Segue os dois programas:
 - Um com as melhores práticas para o ABAP clássico e outra com as melhores práticas para o ABAP HANA



programa com melhores praticas sem hana.txt



programa com melhores praticas do hana.txt

Programação em HANA

■ Exercício 1:

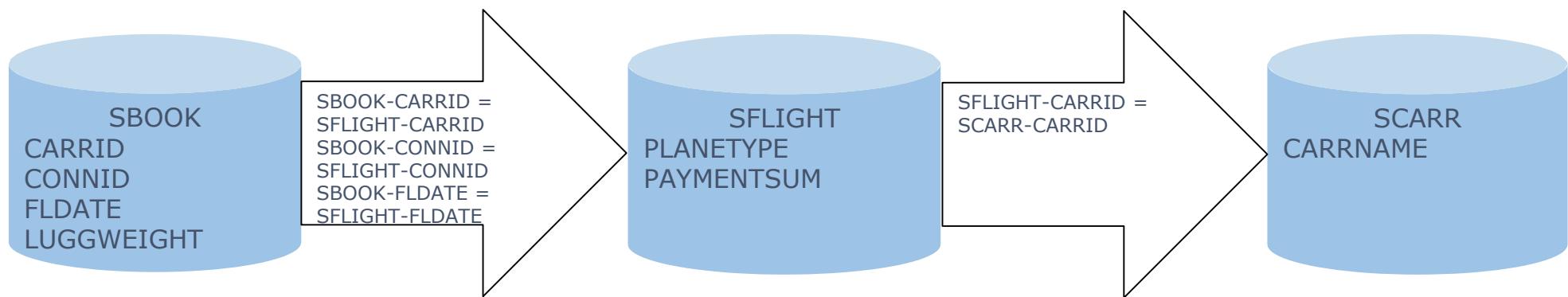
Crie 1 relatório com as definições abaixo, com as melhores práticas clássicas.

Definição do relatório:

- Todas companhias aéreas querem saber quanto em Kg de bagagem cada voo possui.
- Os dados que precisam ter no relatório, são:
 - SBOOK-CARRID -> Cód da companhia aérea
 - SCARR-CARRNAME-> Nome da companhia aérea
 - SBOOK-CONNID-> Cod do Vôo
 - SBOOK-FLDATE-> Data do Vôo
 - SBOOK-LUGGWEIGHT ->Peso da Bagagem
 - SFLIGHT-PLANETYPE-> Tipo do Avião
 - SFLIGHT-PAYMENTSUM-> Valor total em \$ por vôo
- Ordenar o resultado por CARRID, CONNID e FLDATE

Programação em HANA

■ Exercício 1:



Programação em HANA

■ Exercício 2:

Com base no relatório gerado no exercício 1, re-desenhe para o modelo HANA e compare o tempo dos dois relatórios criados.

Definição do relatório:

- Todas companhias aéreas querem saber quanto em Kg de bagagem cada voo possui.
- Os dados que precisam ter no relatório, são:
 - SBOOK-CARRID -> Cód da companhia aérea
 - SCARR-CARRNAME-> Nome da companhia aérea
 - SBOOK-CONNID-> Cod do Vôo
 - SBOOK-FLDATE-> Data do Vôo
 - SBOOK-LUGGWEIGHT ->Peso da Bagagem
 - SFLIGHT-PLANETYPE-> Tipo do Avião
- Ordenar o resultado por CARRID, CONNID e FLDATE

Programação em HANA

■ Exercício 3: Desafio

Com base no relatório gerado no exercício 2, re-desenhe para o modelo HANA e compare o tempo dos dois relatórios criados.

Definição do relatório:

- Todas companhias aéreas querem os voos que possuem o total de bagagem maior do que 2.000 kg.
- Os dados que precisam ter no relatório, são:
 - SBOOK-CARRID -> Cód da companhia aérea
 - SCARR-CARRNAME-> Nome da companhia aérea
 - SBOOK-CONNID-> Cod do Vôo
 - SBOOK-FLDATE-> Data do Vôo
 - SBOOK-LUGGWEIGHT ->Peso da Bagagem
 - SFLIGHT-PLANETYPE-> Tipo do Avião
- Ordenar o resultado por CARRID, CONNID e FLDATE

Programação em HANA – Novo Open SQL

Programação em HANA

■ Novo Open-SQL

- Tendo em vista que a forma de desenvolver sofreu mudanças bruscas e sabendo que o OpenSQL é limitado, a SAP está trabalhando a cada Support Package para adicionar funcionalidades do SQL Nativo para o Open SQL.
- A partir da versão ABAP 7.4 SP05, a SAP já adicionou itens importantes que devem ser usados para desenvolvimento HANA.
- O curso destaca as mais importantes

Programação em HANA

- Novo Open-SQL – Declaração de Dados:

```
SELECT scarr~carrname,
       spfli~connid, spfli~cityfrom, spfli~cityto
  FROM scarr INNER JOIN spfli ON scarr~carrid = spfli~carrid
 WHERE scarr~carrid = @carrier
 ORDER BY scarr~carrid
 INTO TABLE @DATA(output2).
```

The screenshot shows the SAP HANA Studio interface for defining a table variable. The table is named 'OUTPUT2'. It has four columns: 'CARRNAME [C(20)]', 'CONNID [N(4)]', 'CITYFROM [C(20)]', and 'CITYTO [C(20)]'. The data is populated with three rows for American Airlines flights from New York to San Francisco and Frankfurt.

Linha	CARRNAME [C(20)]	CONNID [N(4)]	CITYFROM [C(20)]	CITYTO [C(20)]
1	American Airlines	0017	NEW YORK	SAN FRANCISCO
2	American Airlines	0064	SAN FRANCISCO	NEW YORK
3	American Airlines	0026	FRANKFURT	NEW YORK

Programação em HANA

■ Novo Open-SQL – Expressões

```
DATA(offset) = 10000.  
SELECT id, num1, num2,  
       cast( num1 AS fltp ) / cast( num2 AS fltp ) AS ratio,  
       div( num1, num2 ) AS div,  
       mod( num1, num2 ) AS mod,  
       @offset + abs( num1 - num2 ) AS sum  
  FROM demo_expressions  
 ORDER BY SUM DESCENDING  
 INTO TABLE @DATA(results).
```

Tabela RESULTS							
Caracts.		Standard [10x7(40)]					
Inserir coluna		Colunas...					
Linha	ID [C(1)]	NUM1 [I(4)]	NUM2 [I(4)]	RATIO [F(8)]	DIV [I(4)]	MOD [I(4)]	SUM [I(4)]
1	0	94	8	1.175000000000000E+01	11	6	10086
2	2	87	10	8.699999999999993E+00	8	7	10077
3	5	4	80	5.000000000000003E-02	0	4	10076
4	4	84	16	5.250000000000000E+00	5	4	10068
5	6	68	37	1.8378378378378379E+00	1	31	10031
6	8	44	74	5.9459459459459463E-01	0	44	10030
7	1	29	56	5.1785714285714290E-01	0	29	10027
8	3	12	37	3.2432432432432434E-01	0	12	10025
9	9	17	4	4.250000000000000E+00	4	1	10013
10	7	19	30	6.3333333333333330E-01	0	19	10011

Programação em HANA

■ Novo Open-SQL – Case

```
itab = VALUE #(  
    ( id = 'x' char1 = 'aaaaaa' char2 = 'bbbbbb' )  
    ( id = 'y' char1 = 'xxxxxx' char2 = 'yyyyyy' )  
    ( id = 'z' char1 = 'mmmmmm' char2 = 'nnnnnn' ) ).  
  
DELETE FROM demo_expressions.  
INSERT demo_expressions FROM TABLE @itab.  
  
DATA else TYPE c LENGTH 10 VALUE 'ffffff'.  
SELECT id, char1, char2,  
       CASE char1  
           WHEN 'aaaaaa' THEN ( char1 && char2 )  
           WHEN 'xxxxxx' THEN ( char2 && char1 )  
           ELSE @else  
       END AS text  
     FROM demo_expressions  
     INTO TABLE @DATA(results).
```

RESULTS

ID	CHAR1	CHAR2	TEXT
x	aaaaaa	bbbbbb	aaaaaabbbbb
y	xxxxxx	yyyyyy	yyyyyyxxxxxx
z	mmmmmm	nnnnnn	ffffff

Programação em HANA

■ Novo Open-SQL – Case

```
CONSTANTS: both_l  TYPE c LENGTH 20 VALUE 'Both < 50',
            both_gt TYPE c LENGTH 20 VALUE 'Both >= 50',
            others   TYPE c LENGTH 20 VALUE 'Others'.

SELECT num1, num2,
       CASE WHEN num1 < 50 AND num2 < 50 THEN @both_l
             WHEN num1 >= 50 AND num2 >= 50 THEN @both_gt
             ELSE @others
       END AS group
  FROM demo_expressions
 ORDER BY group
 INTO TABLE @DATA(results).
```

RESULTS

NUM1	NUM2	GROUP
41	19	Both < 50
40	23	Both < 50
96	6	Others
18	90	Others
48	51	Others
60	20	Others
26	98	Others
67	21	Others
62	42	Others
56	36	Others

Programação em HANA

■ Novo Open-SQL – Agregações

CHAR1	CHAR2	NUM1	NUM2	CHAR1	CHAR2	NUM1	NUM2
AA	AA	1	4	BA	AA	8	7
AA	AA	4	2	BA	AA	8	9
AA	AA	7	1	BA	BA	1	8
AA	AA	7	3	BA	BA	6	1
AA	AB	3	2	BA	BB	4	1
AA	AB	10	4	BA	BB	8	4
AA	AB	9	6	BA	BB	5	5
AA	BA	6	7	BB	AA	2	10
AA	BB	7	4	BB	AB	3	4
AA	BB	4	5	BB	BA	10	7
AB	AA	10	10				
AB	BA	8	7				
AB	BA	5	8				
AB	BA	2	2				
AB	BB	4	8				
AB	BB	3	4				

Programação emHANA

■ Novo Open-SQL – Agregações

```
SELECT char1 && '_' && char2 AS group,
       MAX( num1 + num2 ) AS max,
       MIN( num1 + num2 ) AS min
  FROM demo_expressions
 GROUP BY char1, char2
 HAVING MIN( num1 * num2 ) > 25
 ORDER BY group
 INTO TABLE @DATA(grouped_having) .
```

GROUP	MAX	MIN
AA_BA	13	13
AB_AA	20	20
BA_AA	17	15
BB_BA	17	17

Programação em HANA

■ Exercício 1:

■ Criar um relatório, onde seja mostrado por voo(SBOOK-CARRID, SBOOK-CONNID, SBOOK-FLDATE) e categoria do cliente(SBOOK-CUSTTYPE) qual é a bagagem(SBOOK-LUGGWEIGHT) mínima e máxima. Se a categoria do cliente for igual a B, exiba em outra coluna “Bagagem por empresa”, caso a categoria do cliente seja igual P, exibe em outra coluna “Bagagem Particular”

■ Obs: Buscar dados em apenas um único select

Programação em HANA

■ Solução – Exercício1:

```
SELECT carrid,
       connid,
       fldate,
       custtype,
       MAX( LUGGWEIGHT ) as maximo,
       MIN( LUGGWEIGHT ) as miminmo,
       CASE
           WHEN custtype = 'B' then 'Bagagem por Empresa'
           WHEN custtype = 'P' then 'Bagagem Particular'
       END as texto_bagagem
  FROM sbook INTO TABLE @DATA(t_data)
 GROUP BY carrid, connid, fldate, custtype.
] IF sy-subrc EQ 0.
-
ENDIF.
```

Programação em HANA

■ Exercício 2:

- Criar um relatório, onde seja mostrado por voo(SFLIGHT-CARRID, SFLIGHT-CONNID, SFLIGHT-FLDATE), quantos lugares tem no total(SFLIGHT-SEATSMAX), qual é a % de lugares de classe executiva(SFLIGHT-SEATSMAX_B) em relação ao total e qual é a % de lugares de primeira classe(SFLIGHT-SEATSMAX_F) em relação ao total

Programação em HANA

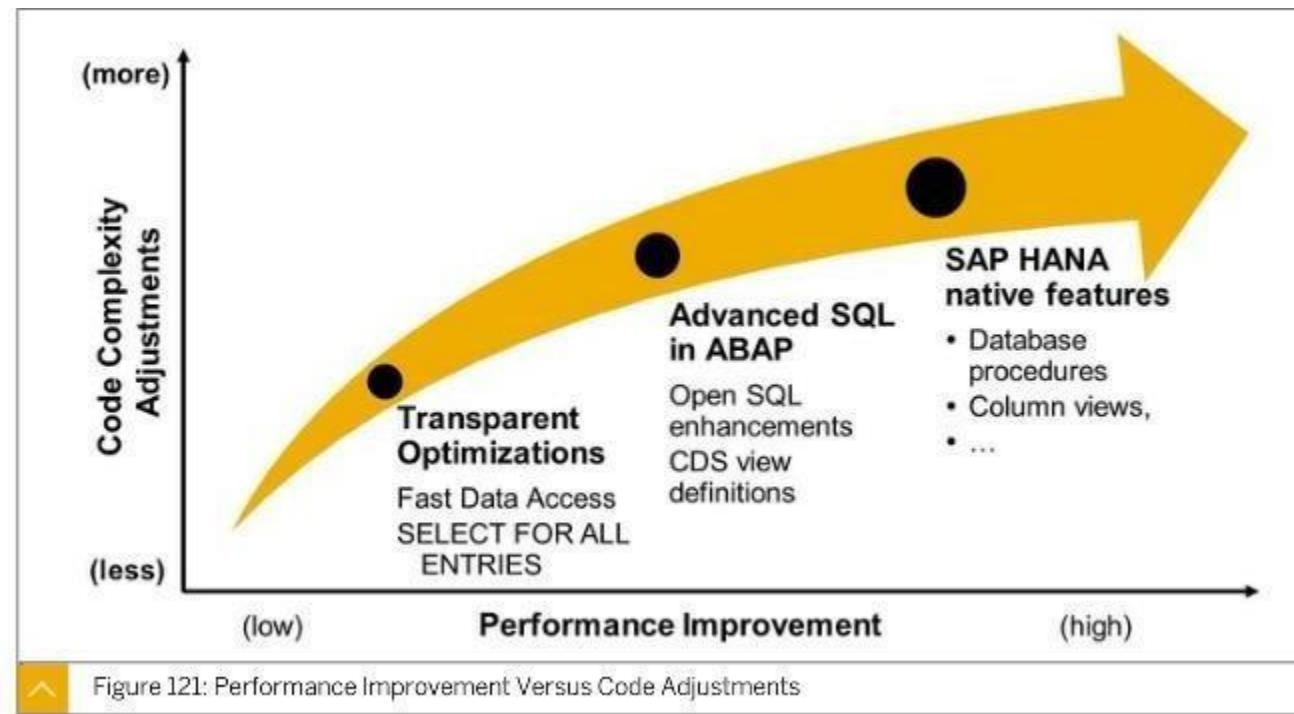
■ Solução - Exercício2:

```
SELECT carrid,
       connid,
       fldate,
       SEATSMAX,
       ( CAST( SEATSMAX_B  as fltp ) * CAST( 100 as fltp ) ) / CAST( SEATSMAX as fltp ) as num_exec,
       ( CAST( SEATSMAX_F  as fltp ) * CAST( 100 as fltp ) ) / CAST( SEATSMAX as fltp ) as num_prim
  FROM sflight INTO TABLE @DATA(t_data)
 GROUP BY carrid, connid, fldate, SEATSMAX, SEATSMAX_B, SEATSMAX_f.
] IF sy-subrc EQ 0.
.
ENDIF.
```

Programação em HANA – SQL Nativo

Programação em HANA

- Mesmo com a SAP trabalhando para obter um novo OPEN SQL, ainda assim tem situações em que será necessário utilizar SQL Nativo para obter o máximo de performance do HANA.
- Por conta disso, a SAP desenvolveu o HANA Studio. Para facilitar o desenvolvedor de criar queries com SQL Nativo





Programação em SQL Nativo

- Principais Características de Sintaxe para SQL NATIVO:
 - Obrigatoriedade de colocar MANDT na clausula where e nas ligações dos JOIN's
 - Campos que serão selecionados separados por ','
 - Em Caso de Join's é usado '.' ao invés de '~' para identificar tabela + Campo
 - Não existe comando como "For All Entries" ou "Into Corresponding fields of table"

Programação em SQL Nativo

Classical Open SQL:

```
SELECT carrid connid cityfrom cityto
      FROM spfli
      INTO ...
 WHERE carrid = 'LH' ...
 ORDER BY carrid connid.
```

Comma-separated field list

Native SQL:

```
SELECT carrid, connid, cityfrom, cityto
      FROM repzme.spfli
 WHERE carrid = 'LH' ...
       and mandt = sy-mandt
 ORDER BY carrid, connid
```

Database schema has to be specified
(if not user's default schema)

No automatic client handling
Client is 'just a key field'

Programação em SQL Nativo

Classical Open SQL:

```
SELECT b~carrid a~carrname b~connid b~cityfrom b~cityto  
      FROM scarr AS a INNER JOIN spfli AS B  
        ON a~carrid = b~carrid  
      INTO ...  
 WHERE b~carrid = 'LH' AND b~connid = '0400'  
 ORDER BY b~carrid.
```

Dot separates qualifier
from column name

Native SQL:

```
SELECT b.carrid, b.connid, a.carrname, b.cityfrom, b.cityto  
      FROM repzme.scarr [AS] a INNER JOIN repzme.spfli [AS] b  
        ON a.carrid = b.carrid AND a.mandt = b.mandt  
      INTO ...  
 WHERE b.carrid = 'LH' AND b.connid = '0400'  
       AND b.mandt = '800'  
 ORDER BY b.mandt, b.carrid
```

No automatic client handling
Client is 'just a key field'

Programação em HANA – SQL NATIVO - ADBC

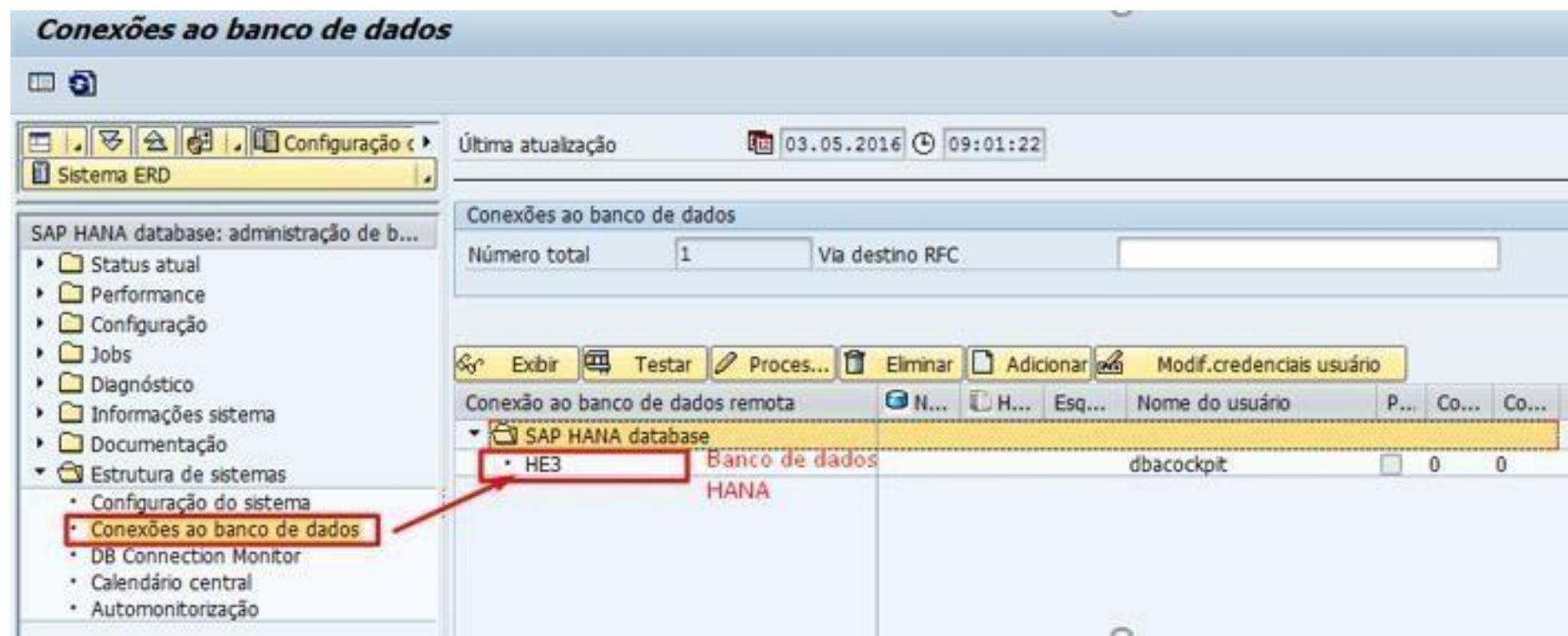
Typical call sequence

- Required data declarations
----->
- Create statement object (and DB connection if necessary)
----->
- Execute query, passing SQL query as string
----->
- Assign internal table for result
----->
- Retrieve result
----->
- Close result
----->
- Handle exceptions
----->

```
DATA:  
  lo_sql      TYPE REF TO cl_sql_statement,  
  lv_sql      TYPE string,  
  lx_sql      TYPE REF TO cx_sql_exception,  
  lo_result   TYPE REF TO cl_sql_result_set,  
  lr_flight   TYPE REF TO data,  
  lt_flight   TYPE STANDARD TABLE OF sflight.  
  
TRY.  
  CREATE OBJECT lo_sql EXPORTING con_ref =  
    cl_sql_connection->get_connection( 'HANADB' ).  
  
  CONCATENATE `SELECT * FROM SFLIGHT WHERE`  
    `mandt='`, sy-mandt,  
    ` AND carrid = 'LH'` INTO lv_sql.  
  lo_result = lo_sql->execute_query( lv_sql ).  
  
  GET REFERENCE OF lt_flight INTO lr_flight.  
  lo_result->set_param_table( lr_flight ).  
  lo_result->next_package( ).  
  
  lo_result->close( ).  
  
CATCH cx_sql_exception INTO ...  
  
ENDTRY.
```

Programação em HANA – SQL NATIVO - ADBC

- Para obter a conexão com o Banco de dados Hana, acessar a transação DBACOCKPIT e ir na pasta Conexões ao banco de dados. A Conexão hana deverá sempre ficar na pasta “SAP HANA DATABASE”



Programação em HANA – SQL NATIVO - ADBC

- Exercício 1:
 - Criar dois relatórios: um Clássico (Com novo open SQL) e outro utilizando SQL Nativo via ADBC, onde seja mostrado por voo (SBOOK-CARRID, SBOOK-CONNID, SBOOK-FLDATE) quantos assentos foram destinados para fumantes e qual é o número de assentos disponíveis no avião.
 - Relatório deve ter os seguintes campos:
 - SBOOK-CARRID, SBOOK-CONNID, SBOOK-FLDATE, NUMERO DE FUMANTES (calculado), SFLIGHT-SEATSMAX
 - Obs1: Número de assentos disponíveis no avião: SFLIGHT-SEATSMAX
 - Obs2: Para saber quais passagens foram compradas com assento de fumante (sbook-smoker= X)
 - Ao final gere a ST12 de cada relatório e compare as diferenças

Programação em HANA – SQL NATIVO - ADBC

- Exercício 2:
 - DESAFIO
 - Com os relatórios criados (modo clássico e ADBC) no exercício 1, adicionar uma coluna no relatório que seja a % dos fumantes em relação ao número total do avião
 - Obs: Para obter o número total de assentos no Voo, ir na **SFLIGHT** e obter o campo **SEATSMAX**
 - Medir novamente a ST12

Programação em HANA - CDS

Programação em HANA – CDS

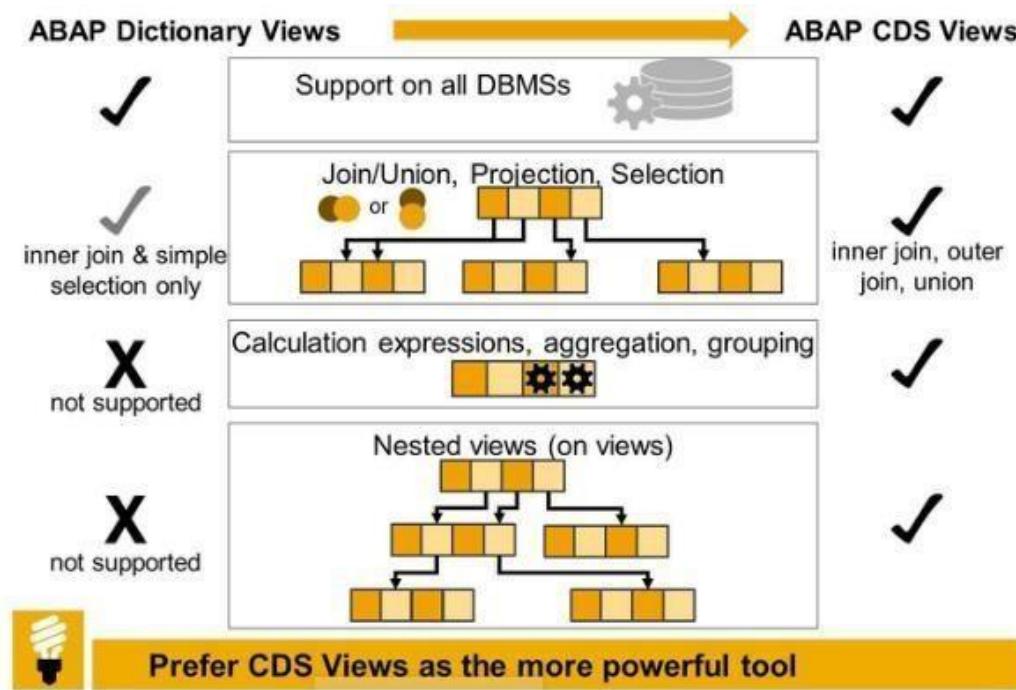
- Core Data Services (CDS) são visões nas quais permite você ler, manipular, e definir dados utilizando SQL NATIVO
- CDS consiste de 3 sub-linguagens:
 - DDL;
 - QL;
 - DCL.

The Core Data Services (CDS) are a collection of domain-specific languages and services for defining and consuming semantically rich data models.

DDL	QL	DCL
 Data Definition Language	 Query Language	 Data Control Language
Model and retrieve data on a semantic level higher than SQL Extends native SQL means for increased productivity	Consume CDS entities via Open SQL in ABAP Fully transparent SQL extensions	Define authorizations for CDS views Modelled and declarative approach Integrates with classic authorization concepts

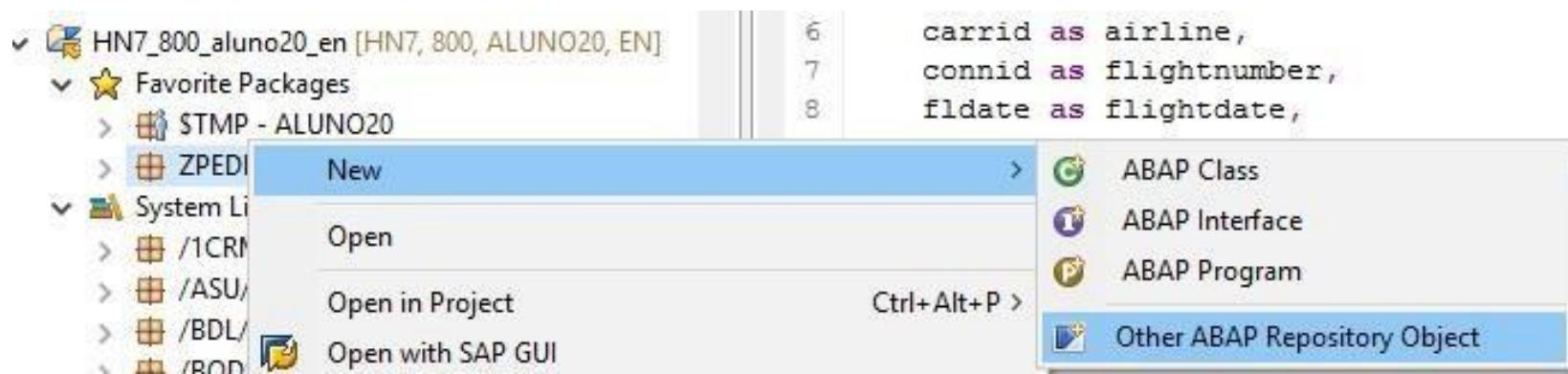
Programação em HANA – CDS

- Visões CDS deve substituir as visões da SE11, sempre que o desenvolvedor precisar de dados que envolvam cálculos e agregações complexas
- O principal objetivo do CDS é conseguir criar uma estrutura, com cálculos e agregações necessárias, de modo que a mesma se torne reutilizável para os N's programas ABAP



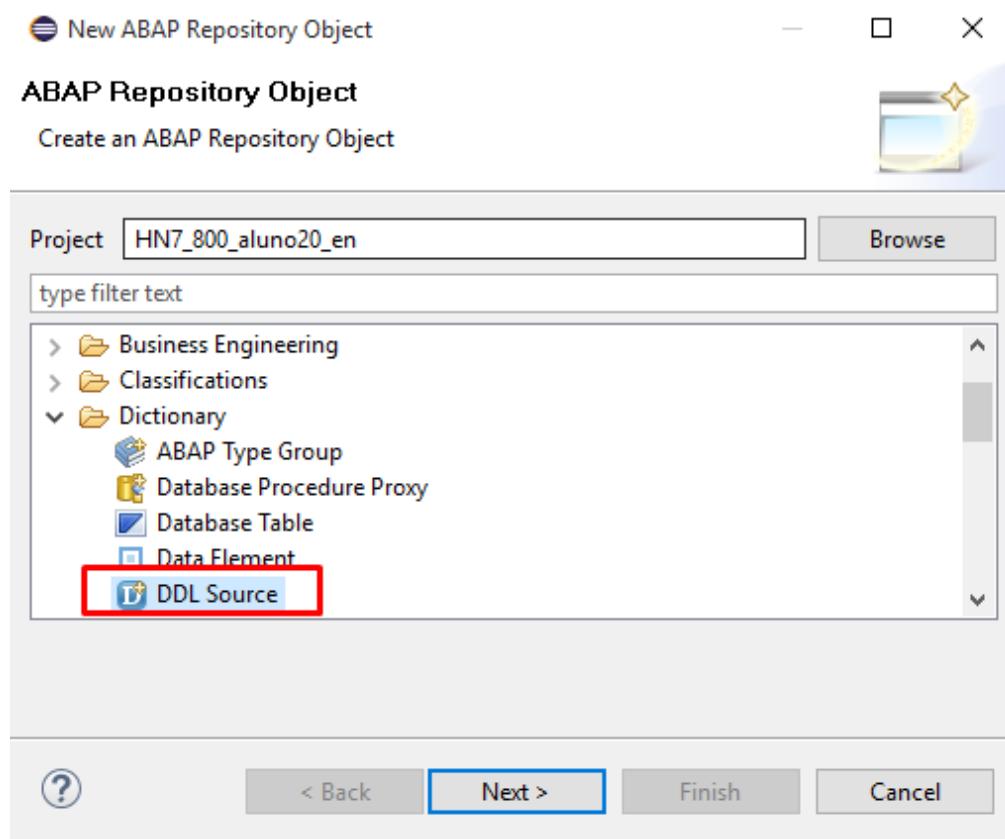
Programação em HANA – CDS

- Exemplo 1:
 - View CDS apesar de ser criado no servidor SAP, só pode ser criado utilizando o ECLIPSE, para isso, acesse o caminho abaixo:
 - No seu pacote de desenvolvimento, clique com o botão direito e acesse: NEW-> Other ABAP Repository Object



Programação em HANA – CDS

- Exemplo 1:
 - Na próxima janela procure pela pasta Dictionary e em seguida clicar em DDL SOURCE:



Programação em HANA – CDS

- Exemplo 1:
 - Na tela seguinte, colocar um nome para sua view DDL

New DDL Source

DDL Source

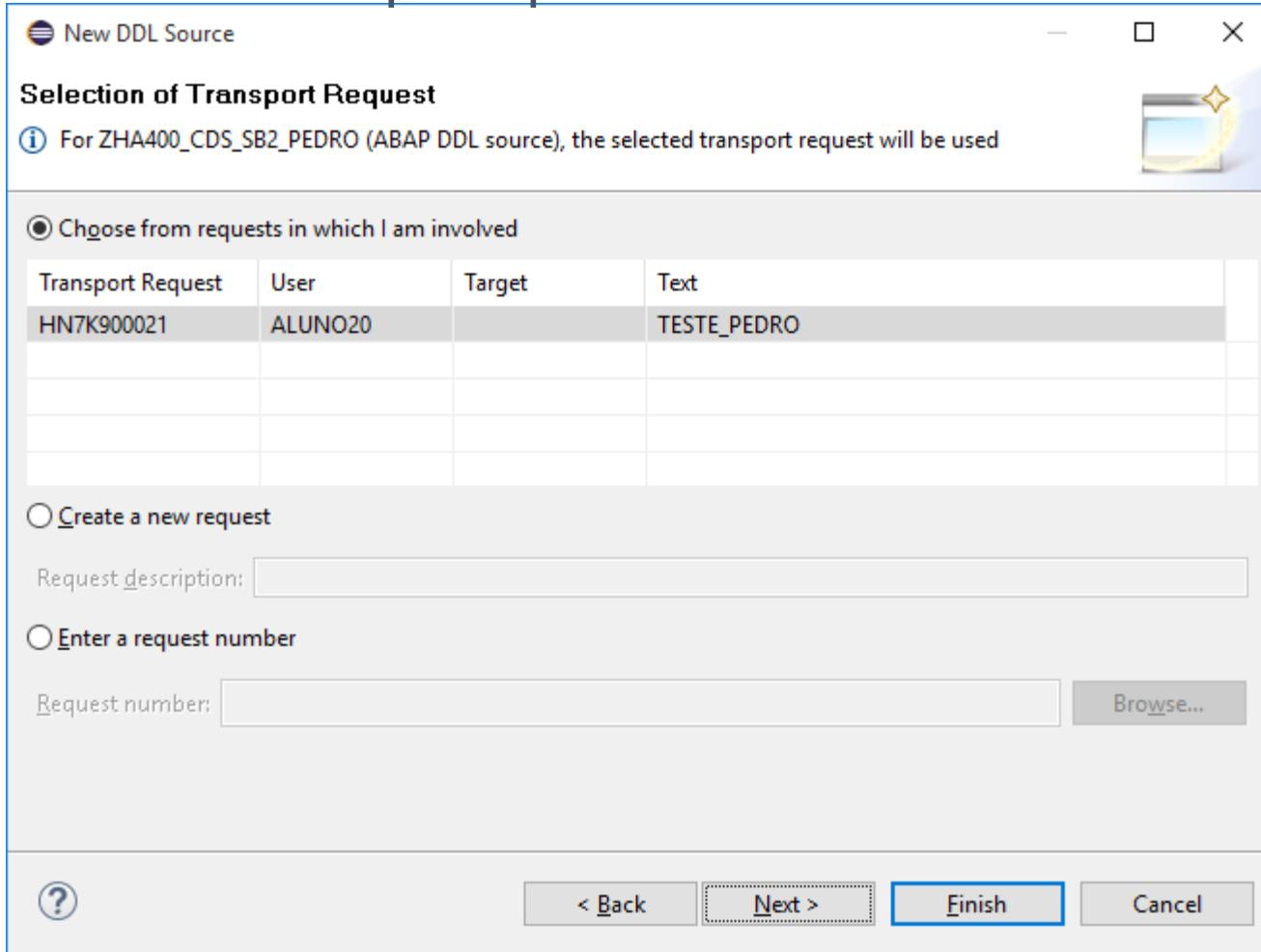
Create a DDL source



Project *	HN7_800_aluno20_en	Browse...
Package: *	ZPEDRO	Browse...
<input type="checkbox"/> Add to favorite packages		
Name: *	ZHA400_CDS_SB2_PEDRO	
Description: *	teste	
Original Language:	EN	

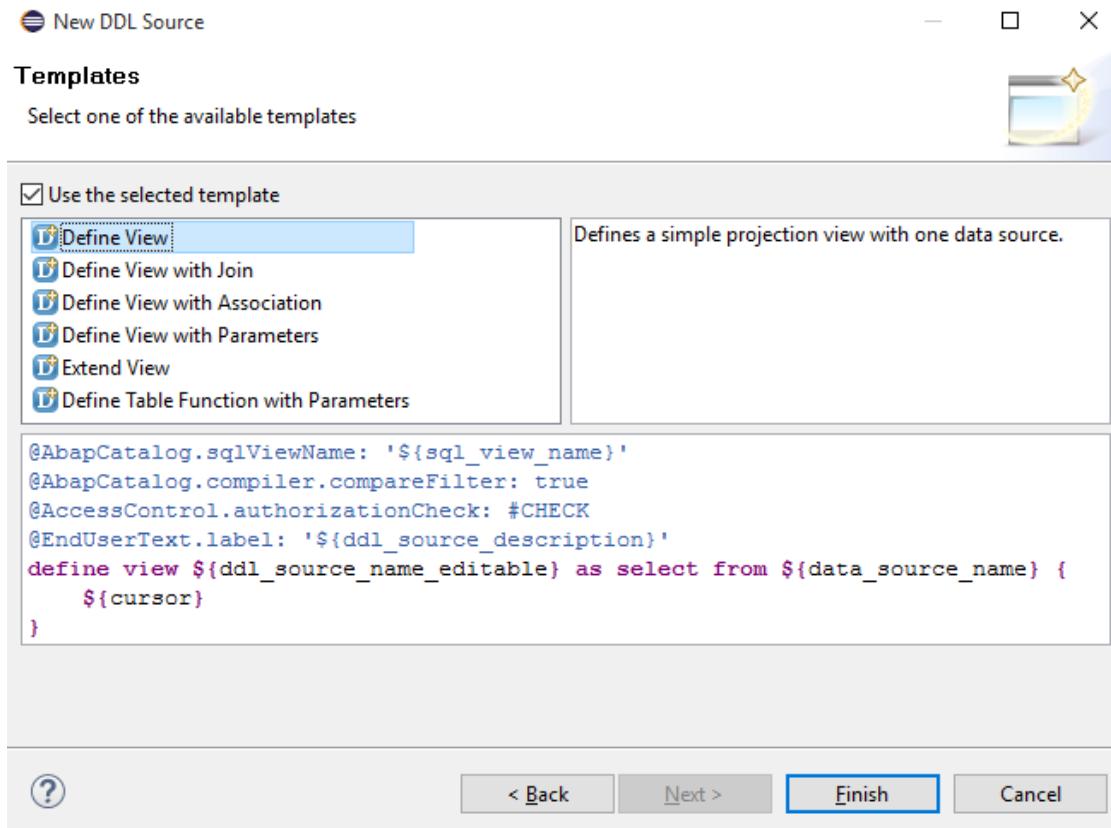
Programação em HANA – CDS

- Exemplo 1:
 - Indicar uma request para armazenar a view



Programação em HANA – CDS

- Exemplo 1:
 - Na próxima tela, o Eclipse já provê templates iniciais para começar o desenvolvimento da sua view.
 - Nesse exemplo, estamos usando a mais simples: Define View



Programação em HANA – CDS

- Exemplo 1:
 - Código CDS:
 - Forma 1:

```
1 @AbapCatalog.sqlViewName: 'Zha400_Cds_PD1'  
2 @EndUserText.label: 'Teste do livro'  
3 define view Zha400_Cds_Sb1_Pedro as select from sbook {  
4     carrid as airline,  
5     connid as flightnumber,  
6     fldate as flightdate,  
7     bookid as id,  
8     customid as customer }
```

- Forma 2:
- ```
@AbapCatalog.sqlViewName: 'Zha400_Cds_PD1'
@EndUserText.label: 'Teste do livro'
define view Zha400_Cds_Sb1_Pedro as select
 carrid as airline,
 connid as flightnumber,
 fldate as flightdate,
 bookid as id,
 customid as customer
from sbook
```

# Programação em HANA – CDS

- Exemplo 1:
  - SE11:

*Dictionary: Display View*

The screenshot shows the SAP Dictionary: Display View interface. The title bar reads "Dictionary: Display View". The toolbar contains various icons for navigation and editing. Below the toolbar, there are two tabs: "DDL SQL View" and "ZHA400\_CDS\_PD1" (which is active). A "Short Description" field contains the value "Teste do livro". Below these are several tabs: Attributes, Table/Join Conditions, View Fds (which is selected), Selection Conditions, and Maint.Status. The main area is titled "Table fields" and displays a table with the following data:

| View field   | Table | Field    | Key                                 | Data elem. | M...                     | DTyp | Length | Short description        |
|--------------|-------|----------|-------------------------------------|------------|--------------------------|------|--------|--------------------------|
| MANDT        | SBOOK | MANDT    | <input checked="" type="checkbox"/> | S_MANDT    | <input type="checkbox"/> | CLNT | 3      | Client                   |
| AIRLINE      | SBOOK | CARRID   | <input checked="" type="checkbox"/> | S_CARR_ID  | <input type="checkbox"/> | CHAR | 3      | Airline Code             |
| FLIGHTNUMBER | SBOOK | CONNID   | <input checked="" type="checkbox"/> | S_CONN_ID  | <input type="checkbox"/> | NUMC | 4      | Flight Connection Number |
| FLIGHTDATE   | SBOOK | FLDATE   | <input checked="" type="checkbox"/> | S_DATE     | <input type="checkbox"/> | DATS | 8      | Flight date              |
| ID           | SBOOK | BOOKID   | <input checked="" type="checkbox"/> | S_BOOK_ID  | <input type="checkbox"/> | NUMC | 8      | Booking number           |
| CUSTOMER     | SBOOK | CUSTOMID | <input type="checkbox"/>            | S_CUSTOMER | <input type="checkbox"/> | NUMC | 8      | Customer Number          |

# Programação em HANA – CDS

- Exemplo 2:

```
1 @AbapCatalog.sqlViewName: 'Zha400_Cds_PD1'
2 @EndUserText.label: 'Teste do livro'
3 define view Zha400_Cds_Sb1_Pedro as select
4 carrid as airline,
5 connid as flightnumber,
6 fldate as flightdate,
7 bookid as id,
8 customid as customer,
9 @Semantics.currencyCode: true
10 forcurkey as currency,
11 @Semantics.amount.currencyCode: 'currency'
12 loccuram as ammount
13 from sbook
14
```

... annotation ...  
... @<annotation> ...

**Effect**

Specifies an annotation in the definition of an element of a SELECT list in a CDS view in ABAP CDS. Annotations can be listed before and after the element.

- In front of the element, the character @ must be placed before the name annotation of the annotation.
- Following the element, the characters @< must be placed before the name annotation of the annotation.

The table below shows the possible predefined annotations, which can be specified, and their meanings. All other annotations are user-defined annotations.

| annotation                    | value                                        | Meaning                                                                    |
|-------------------------------|----------------------------------------------|----------------------------------------------------------------------------|
| EndUserText.label             | Character string with maximum 60 characters  | Short text describing the element                                          |
| EndUserText.quickInfo         | Character string                             | Tooltip of the element                                                     |
| Semantics.amount.currencyCode | Name of an element defined as a currency key | Element is defined as a currency field to which a currency key is assigned |

# Programação em HANA – CDS

- Exemplo 3:

```
@AbapCatalog.sqlViewName: 'Zha400_cds_PD3'
@EndUserText.label: 'Teste complexo'
define view Zha400_Cds_Sb3_Pedro as select
CONCAT(carrid, connid) as flightno,
fldate,
bookid,
CONCAT(
 CONCAT(SUBSTRING(order_date,5,2), '-'),
 SUBSTRING(order_date,1,4)) as ordermonth,

case smoker
when 'X' then cast(loccuram as abap.fltp)* 1.03
else cast(loccuram as abap.fltp)* 0.98
end as adjustment_amount,

currency_conversion(amount => loccuram,
 source_currency => loccurkey,
 target_currency => cast('EUR' as abap.cuky(5)),
 exchange_rate_date => order_date) as euro_amount

from sbook
```

# Programação em HANA – CDS

- Exemplo 3:

Zha400\_Cds\_Sb3\_Pedro

Raw Data

100 Rows retrieved - 220 ms

| flightno | fdate      | bookid   | ordermonth | adjustment_amount     | euro_amount |  |
|----------|------------|----------|------------|-----------------------|-------------|--|
| LH0401   | 2014-12-05 | 00038575 | 10-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038613 | 10-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038614 | 10-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038630 | 11-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038704 | 09-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038710 | 10-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038733 | 07-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038748 | 10-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038850 | 07-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-05 | 00038870 | 08-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00038967 | 08-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00038985 | 07-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039014 | 09-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039098 | 08-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039103 | 07-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039126 | 09-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039165 | 10-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039169 | 08-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039176 | 11-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039199 | 07-2014    | 6.200460000000005E+02 | 632.70      |  |
| LH0401   | 2014-12-12 | 00039229 | 09-2014    | 6.200460000000005E+02 | 632.70      |  |

# Programação em HANA – CDS

- Exemplo 3:

```
8 REPORT zteste_pedro.
9
10
11 SELECT * FROM zha400_cds_pd3 INTO TABLE @DATA(t_results)
12 UP TO 200 rows.
13 IF sy-subrc EQ 0.
14
15 ENDIF.
```

| T_RESULTS                 |              |                 |               |               |                |                  |                 |
|---------------------------|--------------|-----------------|---------------|---------------|----------------|------------------|-----------------|
| Standard [200x7(88)]      |              |                 |               |               |                |                  |                 |
| Insert Column Columns ... |              |                 |               |               |                |                  |                 |
| Row                       | MANDT [C(3)] | FLIGHTNO [C(8)] | FLDATE [D(8)] | BOOKID [N(8)] | ORDERMONTH ... | ADJUSTMENT_A...  | EURO_AMOUNT ... |
| 1                         | 800          | LH0401          | 20141205      | 00038521      | 11-2014        | 6.20046000000... | 632.70          |
| 2                         | 800          | LH0401          | 20141205      | 00038575      | 10-2014        | 6.20046000000... | 632.70          |
| 3                         | 800          | LH0401          | 20141205      | 00038613      | 10-2014        | 6.20046000000... | 632.70          |
| 4                         | 800          | LH0401          | 20141205      | 00038614      | 10-2014        | 6.20046000000... | 632.70          |
| 5                         | 800          | LH0401          | 20141205      | 00038630      | 11-2014        | 6.20046000000... | 632.70          |
| 6                         | 800          | LH0401          | 20141205      | 00038704      | 09-2014        | 6.20046000000... | 632.70          |
| 7                         | 800          | LH0401          | 20141205      | 00038710      | 10-2014        | 6.20046000000... | 632.70          |
| 8                         | 800          | LH0401          | 20141205      | 00038733      | 07-2014        | 6.20046000000... | 632.70          |
| 9                         | 800          | LH0401          | 20141205      | 00038748      | 10-2014        | 6.20046000000... | 632.70          |
| 10                        | 800          | LH0401          | 20141205      | 00038850      | 07-2014        | 6.20046000000... | 632.70          |
| 11                        | 800          | LH0401          | 20141205      | 00038870      | 08-2014        | 6.20046000000... | 632.70          |

# Programação em HANA – CDS

- O que aCDS suporta:

## String & arithmetic expressions

- Literals
- Operators: +, -, \*, /, unary -
- Built-in functions, e.g. CONCAT

## CAST expression

- Length determined at activation time
- No nesting of CAST expressions

## CASE expression

## Special built-in functions

- Currency and unit conversion
- COALESCE

## Alias required for result columns

```
define view ha400_ddl_sb5 as
select from sbook {
CONCAT(carrid, connid) as flightno,
fldate, bookid,
CONCAT(CONCAT(
SUBSTRING(order_date,5,2), '-'),
SUBSTRING(order_date,1,4)) as ordermonth,
case smoker
when 'X' then
cast(loccuram as abap.fltp)*1.03
else cast(loccuram as abap.fltp)*0.98
end as adjusted_amount,
currency_conversion(
amount => loccuram,
source_currency => loccurkey,
target_currency =>
cast('EUR' as abap.cuky(5)),
exchange_rate_date => order_date
) as euro_amount }
```

# Programação em HANA – CDS

- O que aCDS suporta:

## Like in Open SQL:

WHERE clause supported

JOINS supported

- INNER, LEFT OUTER, RIGHT OUTER join
- Before or after the field list
- No SELECT \* together with joins

## Aggregations, Grouping, Filtering supported

- All columns not aggregated must be listed in GROUP BY
- HAVING clause filters resulting groups, aggregate functions supported

```
define view ha400_ddl_sbc0 as
 select from sbook as book
 inner join scustom as customer
{
 book.carrid,
 book.loccurkey as currcode,
 sum(book.loccuram) as total_amount,
 max(book.loccuram) as max_amount,
 min(book.loccuram) as min_amount,
 customer.region,
 customer.discount
} where customer.country = 'US'
group by book.carrid, book.loccurkey,
 customer.region,
 customer.discount
having sum(book.loccuram) > 100
```

# Programação em HANA – CDS

- Exemplo 4:

```
1 @AbapCatalog.sqlViewName: 'Zha400_Cds_PD1'
2 @EndUserText.label: 'Teste do livro'
3 define view Zha400_Cds_Sb1_Pedro as select
4 carrid as airline,
5 connid as flightnumber, CDS PRINCIPAL
6 fldate as flightdate,
7 bookid as id,
8 customid as customer,
9 @Semantics.currencyCode: true
10 forcurkey as currency,
11 @Semantics.amount.currencyCode: 'currency'
12 loccuram as ammount
13 from sbook
```

---

```
@AbapCatalog.sqlViewAppendName: 'ZHA400_APPEND'
extend view ZHA400_CDS_SB1_PEDRO with ZHA400_CDS_SB1E_PEDRO
{
 case reserved
 when 'X' then 'Reserved'
 else 'Booked'
 end as reserved
}
```

# Programação em HANA – CDS

- Exemplo 4:
  - SE11:

**Dictionary: exibir visão**

Visão SQL DDL ZHA400\_APPEND ativo  
Descrição breve Erzeugt aus Ddl Source ZHA400\_CDS\_SB1E\_PEDRO

Caracts. Cpos.visão

| Tabelas | Campos de visão                                                             |
|---------|-----------------------------------------------------------------------------|
| SBOOK   | Camp.visão Nome da ta Nome campo Ch. Elem.dados M... Tpt                    |
|         | MANDT SBOOK MANDT <input checked="" type="checkbox"/> S_MANDT CLN           |
|         | AIRLINE SBOOK CARRID <input checked="" type="checkbox"/> S_CARR_ID CHA      |
|         | FLIGHTNUMBER SBOOK CONNID <input checked="" type="checkbox"/> S_CONN_ID NUM |
|         | FLIGHTDATE SBOOK FLDATE <input checked="" type="checkbox"/> S_DATE DAT      |
|         | ID SBOOK BOOKID <input checked="" type="checkbox"/> S_BOOK_ID NUM           |
|         | CUSTOMER SBOOK CUSTOMID <input type="checkbox"/> S_CUSTOMER NUM             |
|         | CURRENCY SBOOK FORCURKEY <input type="checkbox"/> S_CURR CUK                |
|         | AMMOUNT SBOOK LOCCURAM <input type="checkbox"/> S_L_CUR_PR CUR              |
|         | .APPEND ZHA400_APPEND *                                                     |
|         | RESERVED DDDDLCHARTYPES CCHAR8 CHA                                          |

**Dictionary: exibir visão**

Visão SQL DDL ZHA400\_APPEND vo  
Descrição breve Erzeugt aus Ddl Source ZHA400\_CDS\_SB1E\_PEDRO

Caracts. Cpos.visão

Última modificação ALUNO20 04.05.2016  
Pacote ZPEDRO  
Idioma original EN Inglês

Visão append ZHA400\_CDS\_PD1

## Programação em HANA – CDS

- Exercício 1:
  - Refazer o exercício 1 do ADBC, transformando o ADBC em uma CDS view.
  - Com a CDS view criada, criar um programa, selecionando dados dela.
  - Medir o tempo entre o ADBC e a CDS View.

# Programação em HANA – CDS

- Solução - Exercício 1:

```
@AbapCatalog.sqlViewName: 'ZTESTEPEDRO'
@EndUserText.label: 'Teste exercicio1'

define view ztestepedro_db as
select
sbook.carrid,
 sbook.connid,
 sbook.fldate,
 count(*) as fumanos,
 sflight.seatsmax

 from (sbook as sbook
inner join sflight as sflight on sbook.carrid = sflight.carrid and
 sbook.connid = sflight.connid and
 sbook.fldate = sflight.fldate)
where
 sbook.smoker = 'X'
group by sbook.carrid, sbook.connid, sbook.fldate, sflight.seatsmax
```

# Programação em HANA – CDS

- Solução - Exercício 1:

```
□ *->
 *& Report ZTESTE9
 *&
 &-----
 *&
 *&
 *&
 *&
 &-----
 REPORT ZTESTE9.

 SELECT * FROM ZTESTEPEDRO INTO TABLE @DATA(t_result).
```

## Programação em HANA – CDS

- Exercício 2:
  - Criar uma CDSVIEW que exiba os dados da sflight da seguinte forma
    - **SFLIGHT-CARRID + ‘-’ + SFLIGHT-CONNID**
    - DATA do Voo -> **SFLIGHT-FLDATE**
    - Tipo do avião ->**SFLIGHT-PLANETYPE**
    - Capacidade classe econômica ->**SFLIGHT-SEATSMAX**
    - Assentos ocupados da Classe econômica ->**SFLIGHT-SEATSOCC**

# Programação em HANA – CDS

- Solução - Exercício 2:

```
1@AbapCatalog.sqlViewName: 'ZTESTEPEDRO2'
2@AbapCatalog.compiler.CompareFilter: true
3@EndUserText.label: 'teste exercicio'
4define view Zpedro2 as select
5 CONCAT(CONCAT(sflight.carrid, '-'), sflight.connid) as voo_num,
6 sflight.planetype,
7 sflight.fldate,
8 sflight.seatsmax,
9 sflight.seatsocc
0 from sflight as sflight
```

# Programação em HANA – CDS

- Solução - Exercício 2:

| AB | voo_num | AB | planetype | 📅 | fldate     | 12 | seatsmax | 12 | seatsocc |
|----|---------|----|-----------|---|------------|----|----------|----|----------|
| AA | 0017    |    | 747-400   |   | 2014-12-03 |    | 385      |    | 370      |
| AA | 0017    |    | 747-400   |   | 2014-12-10 |    | 385      |    | 367      |
| AA | 0017    |    | 747-400   |   | 2014-12-17 |    | 385      |    | 372      |
| AA | 0017    |    | 747-400   |   | 2014-12-24 |    | 385      |    | 362      |
| AA | 0017    |    | 747-400   |   | 2014-12-31 |    | 385      |    | 375      |
| AA | 0017    |    | 747-400   |   | 2015-01-07 |    | 385      |    | 363      |
| AA | 0017    |    | 747-400   |   | 2015-01-14 |    | 385      |    | 372      |
| AA | 0017    |    | 747-400   |   | 2015-01-21 |    | 385      |    | 371      |
| AA | 0017    |    | 747-400   |   | 2015-01-28 |    | 385      |    | 364      |
| AA | 0017    |    | 747-400   |   | 2015-02-04 |    | 385      |    | 371      |
| AA | 0017    |    | 747-400   |   | 2015-02-11 |    | 385      |    | 370      |
| AA | 0017    |    | 747-400   |   | 2015-02-18 |    | 385      |    | 369      |
| AA | 0017    |    | 747-400   |   | 2015-02-25 |    | 385      |    | 372      |
| AA | 0017    |    | 747-400   |   | 2015-03-04 |    | 385      |    | 373      |
| AA | 0017    |    | 747-400   |   | 2015-03-11 |    | 385      |    | 371      |
| AA | 0017    |    | 747-400   |   | 2015-03-18 |    | 385      |    | 374      |

# Programação em HANA – CDS

- Exercício 3
  - Criar um append da view criada no exercício anterior, adicionando três colunas
    - Campos do relatório:
      - Número de assentos da classe econômica disponíveis (Sflight-seatsmax – sflight-seatsocc)
      - Total em valor (SFLIGHT-PAYMENTSUM)
      - Moeda (SFLIGHT-CURRENCY)
      - Adicionar uma coluna:
        - Se o SFLIGHT-PAYMENTSUM for > 190.000,00 “Voo lucrativo”, porém SFLIGHT-PAYMENTSUM seja menor que 190.000,00 exibir “Voo com pouco lucro”, caso não haja nenhuma das duas opções, exibir “Voo sem margem de lucro”
    - Obs: Vincular Moeda montante usando @(anotação)

# Programação em HANA – CDS

- Solução - Exercício 3

```
@AbapCatalog.sqlViewAppendName: 'ZTESTEPEDRO3'
extend view ZPEDRO2 with Zpedro3 {
 sflight.seatsmax - sflight.seatsocc as dif_assentos,
 @Semantics.amount.currencyCode: 'moeda'
 sflight.paymentsum,
 @Semantics.currencyCode: true
 sflight.currency as moeda,
 case
 when sflight.paymentsum > 190000 then 'VOO LUCRATIVO'
 when sflight.paymentsum < 190000 then 'VOO COM POUCO LUCRO'
 else 'VOO SEM MARGEM DE LUCRO'
 end as lucro
}
```

# Programação em HANA – CDS

- Solução - Exercício 3

| AB      | voo_num | AB | planetype | CALENDAR | fldate     | 12 | seatsmax | 12 | seatsocc | 12 | dif_assentos | AB        | paymentsum | AB  | moeda | AB                  | lucro |
|---------|---------|----|-----------|----------|------------|----|----------|----|----------|----|--------------|-----------|------------|-----|-------|---------------------|-------|
| AA-0017 | 747-400 |    |           |          | 2014-12-03 |    | 385      |    | 370      |    | 15           | 191270.64 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2014-12-10 |    | 385      |    | 367      |    | 18           | 191456.70 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2014-12-17 |    | 385      |    | 372      |    | 13           | 193144.19 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2014-12-24 |    | 385      |    | 362      |    | 23           | 186563.14 |            | USD |       | VOO COM POUCO LUCRO |       |
| AA-0017 | 747-400 |    |           |          | 2014-12-31 |    | 385      |    | 375      |    | 10           | 193013.13 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-01-07 |    | 385      |    | 363      |    | 22           | 188233.95 |            | USD |       | VOO COM POUCO LUCRO |       |
| AA-0017 | 747-400 |    |           |          | 2015-01-14 |    | 385      |    | 372      |    | 13           | 193520.62 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-01-21 |    | 385      |    | 371      |    | 14           | 193854.76 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-01-28 |    | 385      |    | 364      |    | 21           | 188132.38 |            | USD |       | VOO COM POUCO LUCRO |       |
| AA-0017 | 747-400 |    |           |          | 2015-02-04 |    | 385      |    | 371      |    | 14           | 192374.46 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-02-11 |    | 385      |    | 370      |    | 15           | 193770.14 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-02-18 |    | 385      |    | 369      |    | 16           | 191503.09 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-02-25 |    | 385      |    | 372      |    | 13           | 194332.73 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-03-04 |    | 385      |    | 373      |    | 12           | 192526.75 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-03-11 |    | 385      |    | 371      |    | 14           | 191482.16 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-03-18 |    | 385      |    | 374      |    | 11           | 192619.78 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-03-25 |    | 385      |    | 364      |    | 21           | 189824.13 |            | USD |       | VOO COM POUCO LUCRO |       |
| AA-0017 | 747-400 |    |           |          | 2015-04-01 |    | 385      |    | 374      |    | 11           | 193474.16 |            | USD |       | VOO LUCRATIVO       |       |
| AA-0017 | 747-400 |    |           |          | 2015-04-08 |    | 385      |    | 364      |    | 21           | 190365.51 |            | USD |       | VOO LUCRATIVO       |       |

# Programação em HANA – CDS

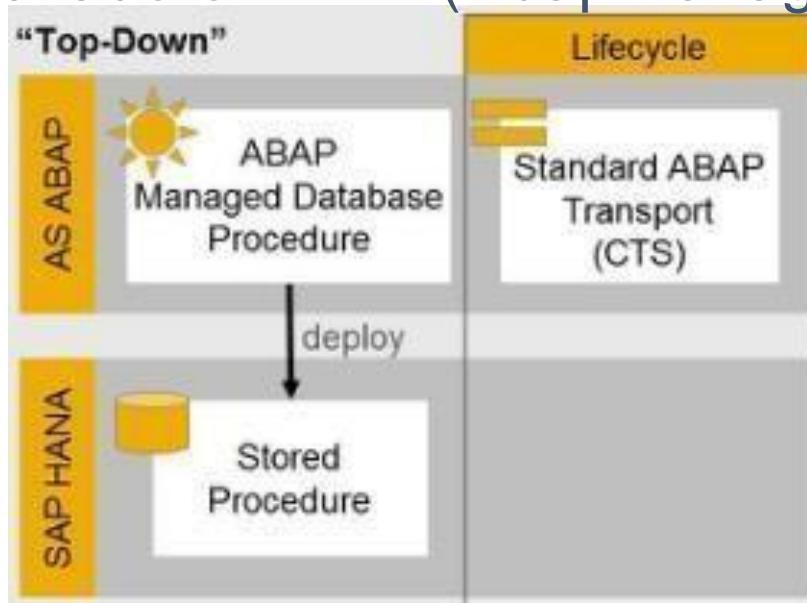
- Solução - Exercício 3

```
@AbapCatalog.sqlViewAppendName: 'ZTESTEPEDRO3'
extend view ZPEDRO2 with Zpedro3 {
 sflight.seatsmax - sflight.seatsocc as dif_assentos,
 @Semantics.amount.currencyCode: 'moeda'
 sflight.paymentsum,
 @Semantics.currencyCode: true
 sflight.currency as moeda,
 case
 when sflight.paymentsum > 190000 then 'VOO LUCRATIVO'
 when sflight.paymentsum < 190000 then 'VOO COM POUCO LUCRO'
 else 'VOO SEM MARGEM DE LUCRO'
 end as lucro
}
```

# Programação em HANA - Procedures

## Programação em HANA – Procedures

- Criação de Procedures não é novidade para o SAP, porém raramente ela era utilizada porque até então não tínhamos arquitetura que suportasse performaticamente essa solução.
- Com a plataforma HANA, a SAP viu a necessidade de criar uma metodologia de criação de procedure que fosse mais fácil do que criar via EXEC SQL.
- Com isso foi criado o AMDP (Abap Managed Database Procedure)



## Programação em HANA – Procedures

- Vantagem em desenvolver por procedures:
  - Grandes volumes de dados não são transferidos para o servidor de aplicação, sempre que possível, resultando em menor tempo de rede.
  - Menor idas e voltas ao Banco de dados
  - É possível criar variáveis locais, eliminando a necessidade de criar explicitamente tabelas temporárias para resultados intermediários
  - Transporte para ambientes via SAP Request

# Programação em HANA – Procedures

```
CLASS CL_AMDP_SAMPLE DEFINITION.
 PUBLIC SECTION.
 INTERFACES IF_AMDP_MARKER_HDB.

 * Only ABAP code possible:
 METHODS process
 IMPORTING it_param TYPE type1
 EXPORTING et_param TYPE type2.

 * ABAP code or SQLScript possible:
 METHODS execute
 IMPORTING VALUE(it_param) TYPE type1
 EXPORTING VALUE(et_param) TYPE type2.

ENDCLASS.
```

Classes with AMDPs must use interface  
IF\_AMDP\_MARKER\_HDB

Parameters must be  
Tables with elementary  
components or  
Scalar types

All AMDP method  
parameters must be passed  
by value

# Programação em HANA – Procedures

```
CLASS CL_AMDP_SAMPLE IMPLEMENTATION.
 METHOD execute
 BY DATABASE PROCEDURE
 FOR db_platform
 LANGUAGE db_language
 [OPTIONS READ-ONLY]
 USING name1 name2 etc..

 -- Write SQLScript code here.
 select * from dummy;

 ENDMETHOD.
ENDCLASS.
```

AMDP methods have to be marked with **BY DATABASE PROCEDURE**

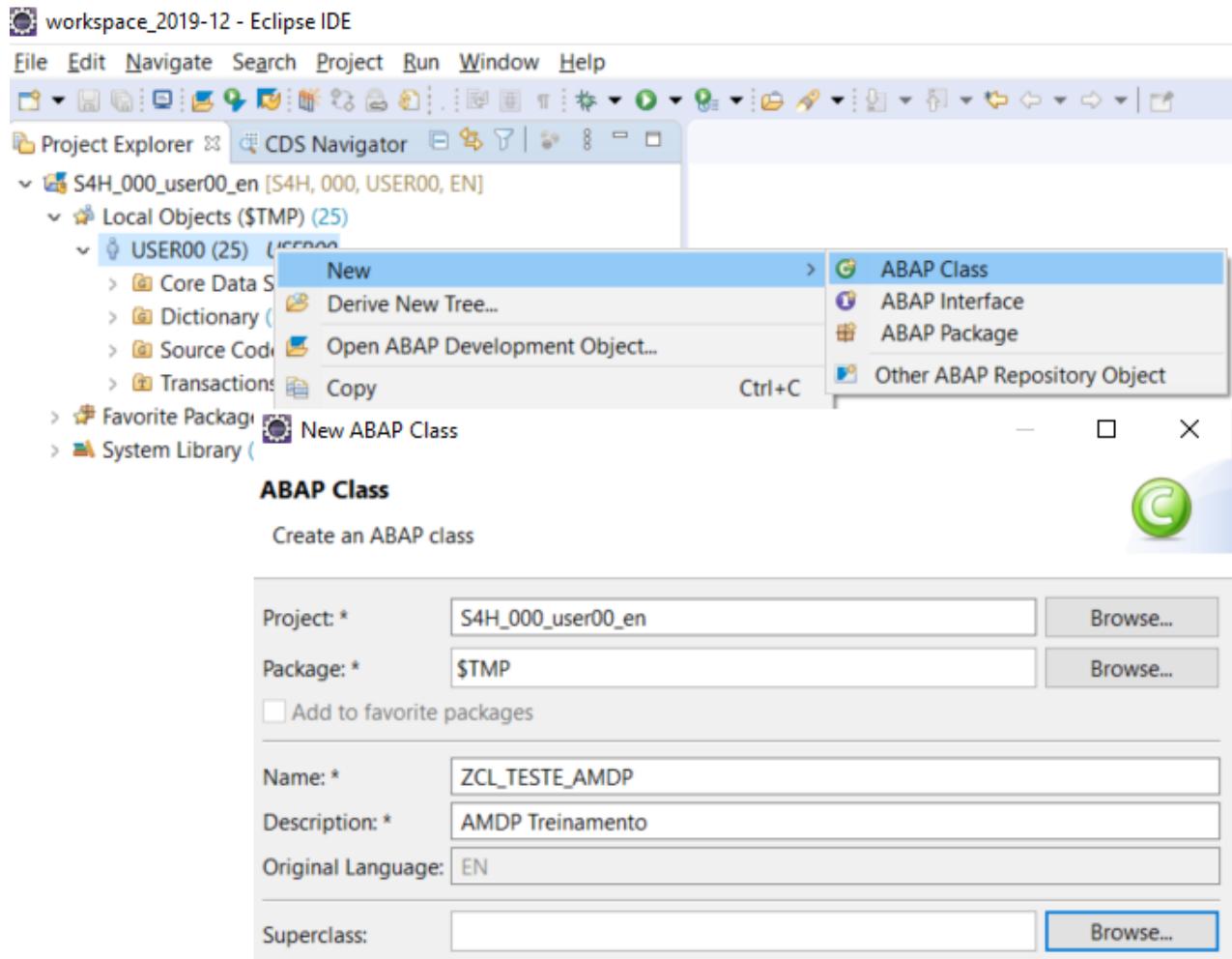
Database platform (**HDB**) and language (**SQLSCRIPT**) to be specified

ABAP Dictionary tables, views, and other AMDP methods to be declared in **USING** clause

Method body:  
contains SQLScript code  
-- marks comments

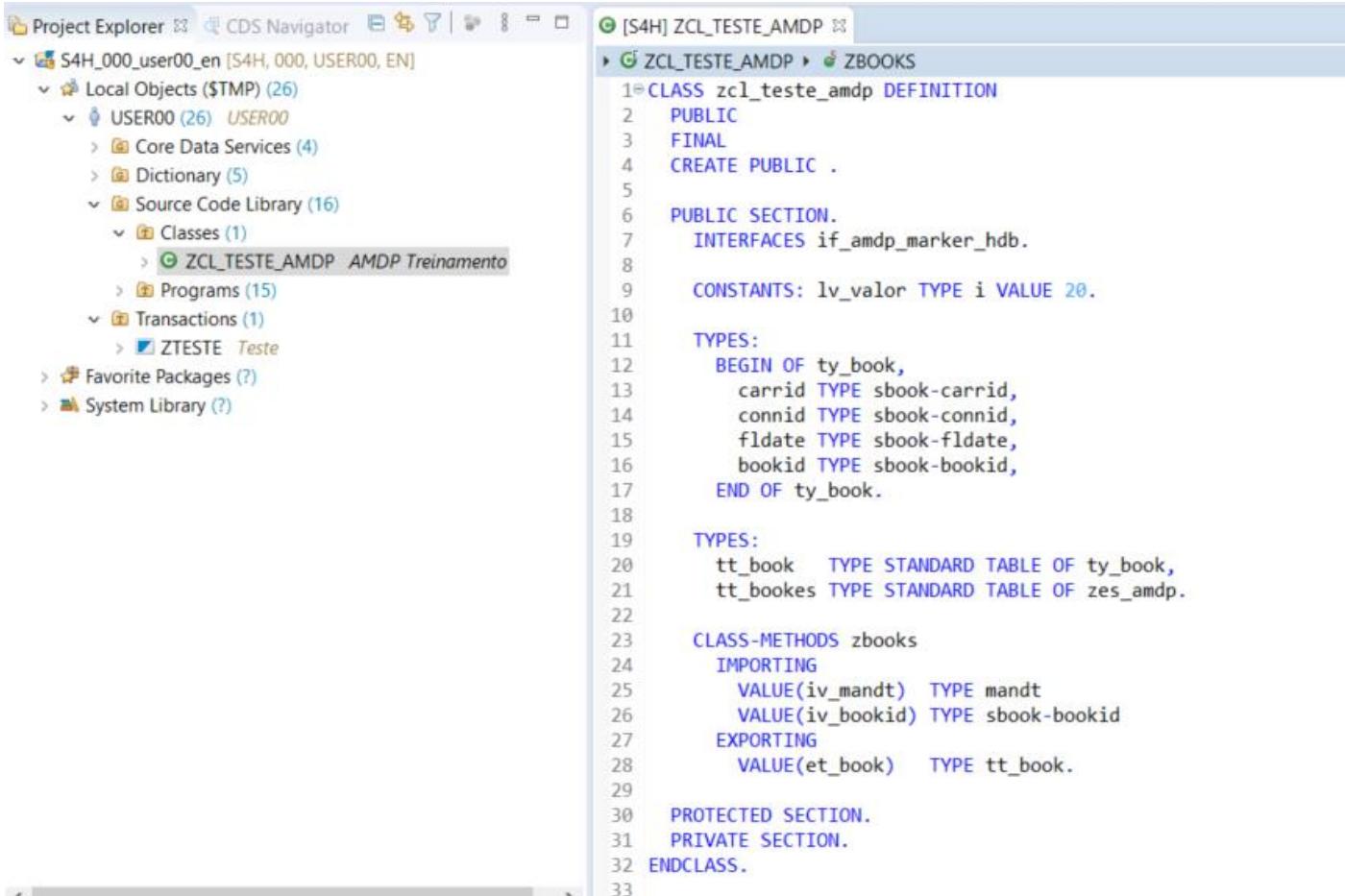
# Programação em HANA – Procedures

- Exemplo 1:



# Programação em HANA – Procedures

- Exemplo 1:

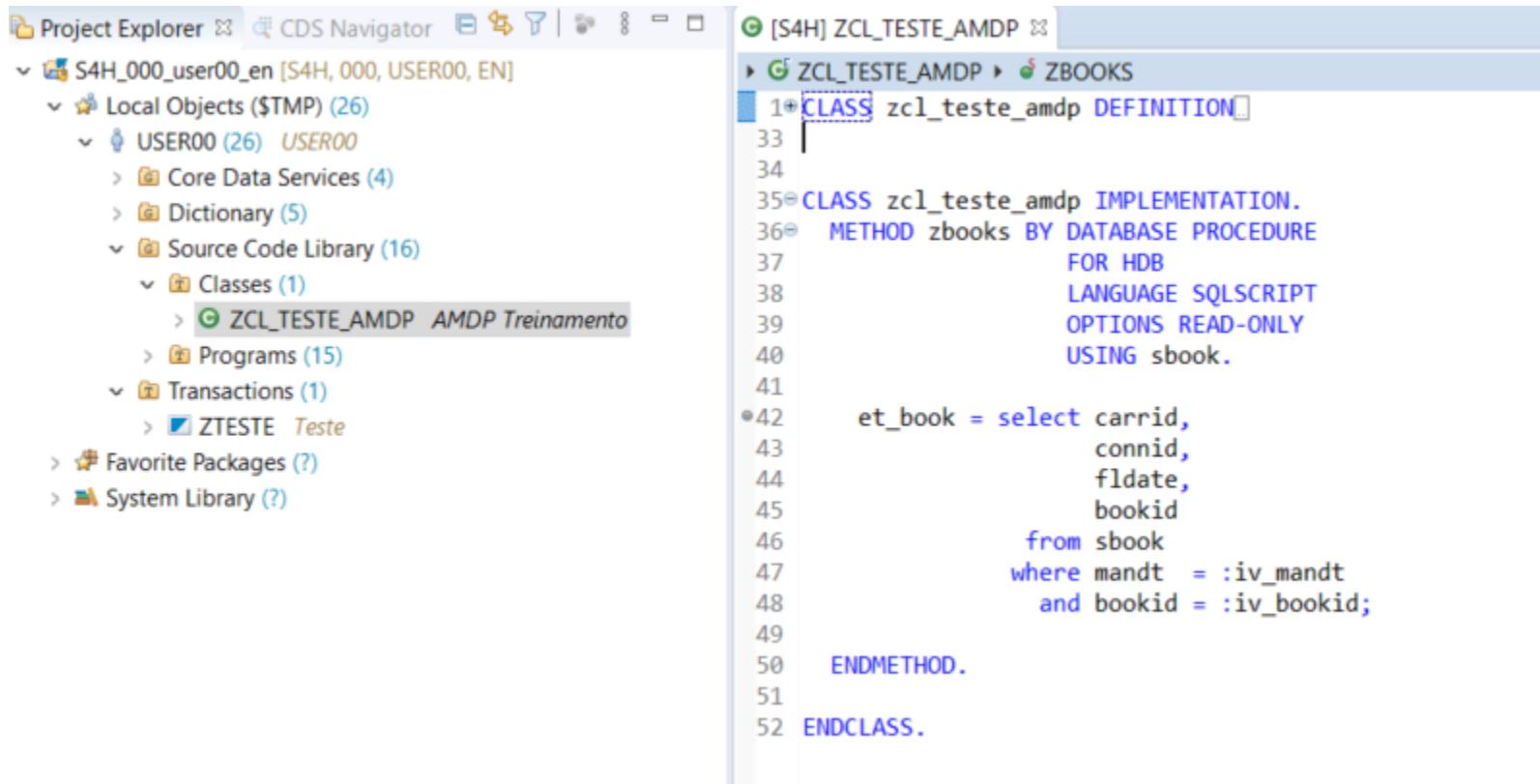


The screenshot shows the SAP Studio interface. On the left, the Project Explorer displays a hierarchy of objects under 'S4H\_000\_user00\_en'. Under 'Local Objects (\$TMP) (26)', there is a 'USER00' node which contains 'Core Data Services (4)', 'Dictionary (5)', 'Source Code Library (16)' (which includes 'Classes (1)' containing 'ZCL\_TESTE\_AMDP AMDP Treinamento', 'Programs (15)', and 'Transactions (1)' containing 'ZTESTE Teste'), and other nodes like 'Favorite Packages (?)' and 'System Library (?)'. The 'ZCL\_TESTE\_AMDP AMDP Treinamento' class is selected in the code editor. On the right, the code editor shows the ABAP code for this class:

```
1 CLASS zcl_tepte_amdp DEFINITION
2 PUBLIC
3 FINAL
4 CREATE PUBLIC .
5
6 PUBLIC SECTION.
7 INTERFACES if_amdp_marker_hdb.
8
9 CONSTANTS: lv_valor TYPE i VALUE 20.
10
11 TYPES:
12 BEGIN OF ty_book,
13 carrid TYPE sbook-carrid,
14 connid TYPE sbook-connid,
15 fldate TYPE sbook-fldate,
16 bookid TYPE sbook-bookid,
17 END OF ty_book.
18
19 TYPES:
20 tt_book TYPE STANDARD TABLE OF ty_book,
21 tt_books TYPE STANDARD TABLE OF zes_amdp.
22
23 CLASS-METHODS zbooks
24 IMPORTING
25 VALUE(iv_mandt) TYPE mandt
26 VALUE(iv_bookid) TYPE sbook-bookid
27 EXPORTING
28 VALUE(et_book) TYPE tt_book.
29
30 PROTECTED SECTION.
31 PRIVATE SECTION.
32 ENDCLASS.
```

# Programação em HANA – Procedures

- Exemplo 1:



The screenshot shows the SAP Studio interface. On the left, the Project Explorer displays a hierarchy of objects under 'S4H\_000\_user00\_en'. Under 'Local Objects (\$TMP) (26)', there are 'USER00 (26)' which includes 'Core Data Services (4)', 'Dictionary (5)', and 'Source Code Library (16)'. The 'Source Code Library' contains 'Classes (1)' with 'ZCL\_TESTE\_AMDP AMDP Treinamento', 'Programs (15)', and 'Transactions (1)' with 'ZTESTE Teste'. Under 'Favorite Packages (1)' is 'System Library (1)'. On the right, the code editor titled '[S4H] ZCL\_TESTE\_AMDP' shows the implementation of the 'zcl\_teste\_amdp' class. The code defines a method 'zbooks' that performs a database query to select book details based on mandt and bookid.

```
1+CLASS zcl_teste_amdp DEFINITION.
33
34
35+CLASS zcl_teste_amdp IMPLEMENTATION.
36+ METHOD zbooks BY DATABASE PROCEDURE
37 FOR HDB
38 LANGUAGE SQLSCRIPT
39 OPTIONS READ-ONLY
40 USING sbook.
41
42 et_book = select carrid,
43 connid,
44 fldate,
45 bookid
46 from sbook
47 where mandt = :iv_mandt
48 and bookid = :iv_bookid;
49
50 ENDMETHOD.
51
52 ENDCLASS.
```

# Programação em HANA – Procedures

- Exemplo 1:

**ABAP Editor: Change Report ZUSER00AMDP**

Report                    ZUSER00AMDP                    Active

```
1 *->-->
2 *& Report ZUSER00AMDP
3 *&-->
4 *&
5 *&-->
6 REPORT ZUSER00AMDP.
7
8 CALL METHOD zcl_teste_amdp=>zbooks(
9 EXPORTING
10 iv_mandt = sy-mandt
11 iv_bookid = 1
12 IMPORTING
13 et_book = DATA(lt_sbook)
14).
15
16 LOOP AT lt_sbook ASSIGNING FIELD-SYMBOL(<fs_book>).
17 WRITE: / <fs_book>-carrid,
18 <fs_book>-connid,
19 <fs_book>-fldate,
20 <fs_book>-fldate.
21 ENDLOOP.
```

# Programação em HANA – Procedures

- Exemplo 2

```
[S4H] ZCL_EXEMPLO02 > GET_FLIGHT_PRICE
1 CLASS zcl_exemplo02 DEFINITION
2 PUBLIC
3 FINAL
4 CREATE PUBLIC .
5
6 PUBLIC SECTION.
7 INTERFACES if_amdp_marker_hdb.
8
9 TYPES:
10 BEGIN OF ty_flight,
11 carrid TYPE sflight-carrid,
12 price TYPE sflight-price,
13 END OF ty_flight,
14
15 BEGIN OF ty_flight_name,
16 carrid TYPE sflight-carrid,
17 carname TYPE scarr-carrname,
18 price TYPE sflight-price,
19 END OF ty_flight_name.
20
21 TYPES:
22 tt_flight TYPE STANDARD TABLE OF ty_flight,
23 tt_flight_name TYPE STANDARD TABLE OF ty_flight_name.
24
25 CLASS-METHODS: get_flight_price
26 IMPORTING
27 VALUE(iv_curr) TYPE char3
28 EXPORTING
29 VALUE(et_flight) TYPE tt_flight.
30
31 CLASS-METHODS: get_flight_name
32 IMPORTING
33 VALUE(iv_curr) TYPE char3
34 EXPORTING
35 VALUE(et_flight_name) TYPE tt_flight_name.
36 PROTECTED SECTION.
37 PRIVATE SECTION.
38 ENDCLASS.
```

# Programação em HANA – Procedures

The screenshot shows the SAP Studio interface with the Project Explorer on the left and the code editor on the right.

**Project Explorer:**

- S4H\_000\_user00\_en [S4H, 000, USER00, EN]
  - Local Objects (\$TMP) (29)
    - USER00 (29) **USER00**
      - Core Data Services (4)
        - Data Definitions (4)
      - Dictionary (7)
        - Data Elements (1)
        - Structures (1)
        - Views (5)
      - Source Code Library (18)
        - Classes (3)
        - Programs (15)
        - Transactions (1)
  - Favorite Packages (?)
  - System Library (?)

**Code Editor:**

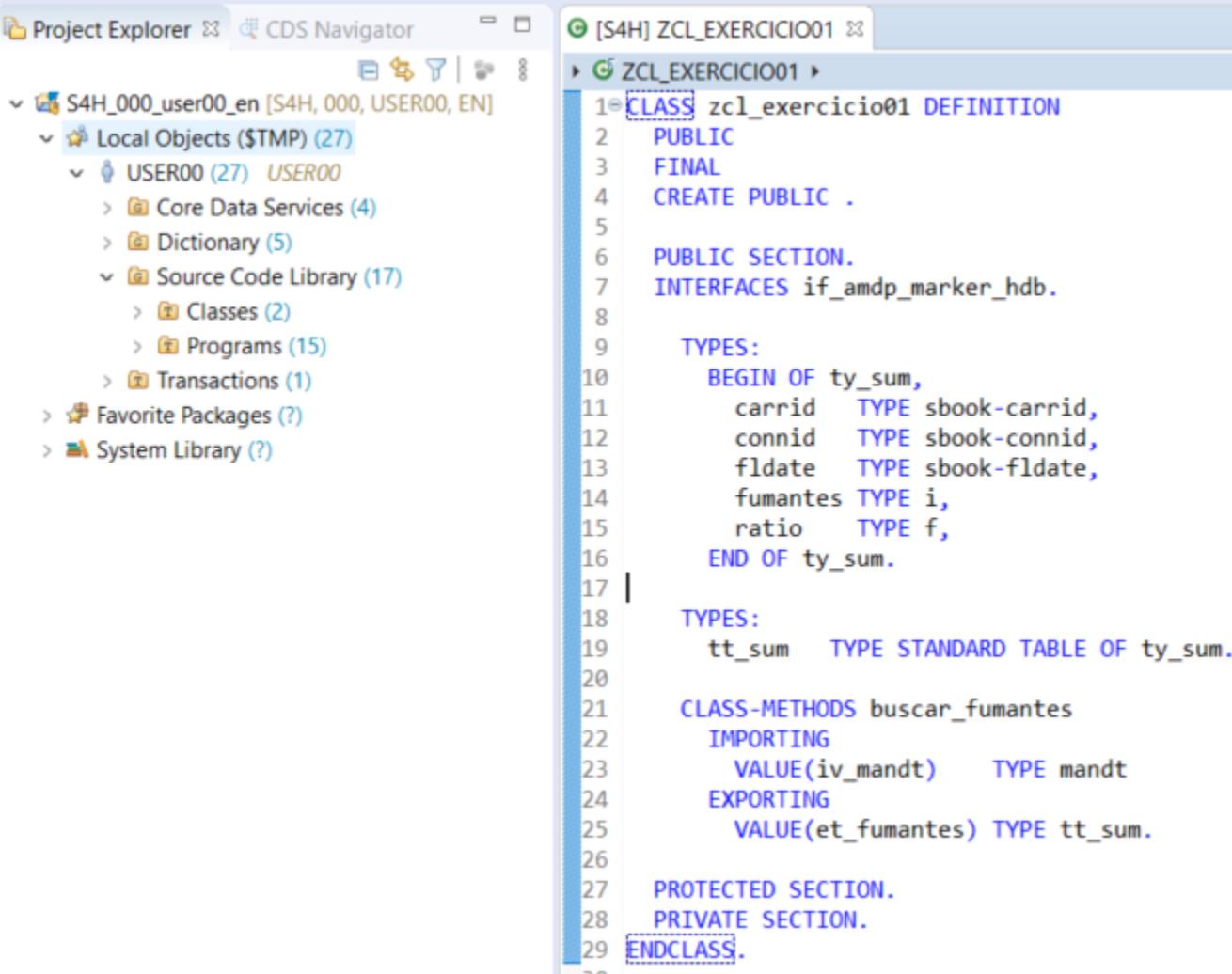
```
*1 CLASS zcl_exemplo02 DEFINITION.
39 |
40=CLASS zcl_exemplo02 IMPLEMENTATION.
41= METHOD get_flight_price BY DATABASE PROCEDURE
42 FOR HDB
43 LANGUAGE SQLSCRIPT
44 OPTIONS READ-ONLY
45 USING sflight.
46
47 IT_TEMP = SELECT * from sflight;
48
49 et_flight = SELECT CARRID, SUM(PRICE) AS PRICE
50 FROM :IT_TEMP
51 WHERE currency = :iv_curr
52 GROUP BY CARRID;
53
54 ENDMETHOD.
55
56= METHOD get_flight_name BY DATABASE PROCEDURE
57 FOR HDB
58 LANGUAGE SQLSCRIPT
59 OPTIONS READ-ONLY
60 USING scarr zcl_exemplo02=>get_flight_price.
61
62 call "ZCL_EXEMPL02=>GET_FLIGHT_PRICE"(iv_curr => :iv_curr, et_flight => :ET_FLIGHT);
63
64 et_flight_name = SELECT a.carrid, b.carrname, a.price
65 from :ET_FLIGHT as a
66 left outer join scarr as b on
67 a.carrid = b.carrid;
68
69 ENDMETHOD.
70
71 ENDCLASS.
```

## Programação em HANA – Procedures

- Exercicio1:
  - Refazer o exercício abaixo, criando uma procedure e um programa que execute essa procedure:
    - Com os relatórios criados(modo clássico e ADBC) no exercício 1, adicionar uma coluna no relatório que seja a % dos fumantes em relação ao numero total do avião
    - Obs: Para obter o numero total de assentos no Vôo, ir na **SFLIGHT** e obter o campo **SEATSMAX**

# Programação em HANA – Procedures

- Solução – Exercicio1:



The screenshot shows the SAP Studio interface with the Project Explorer on the left and the code editor on the right.

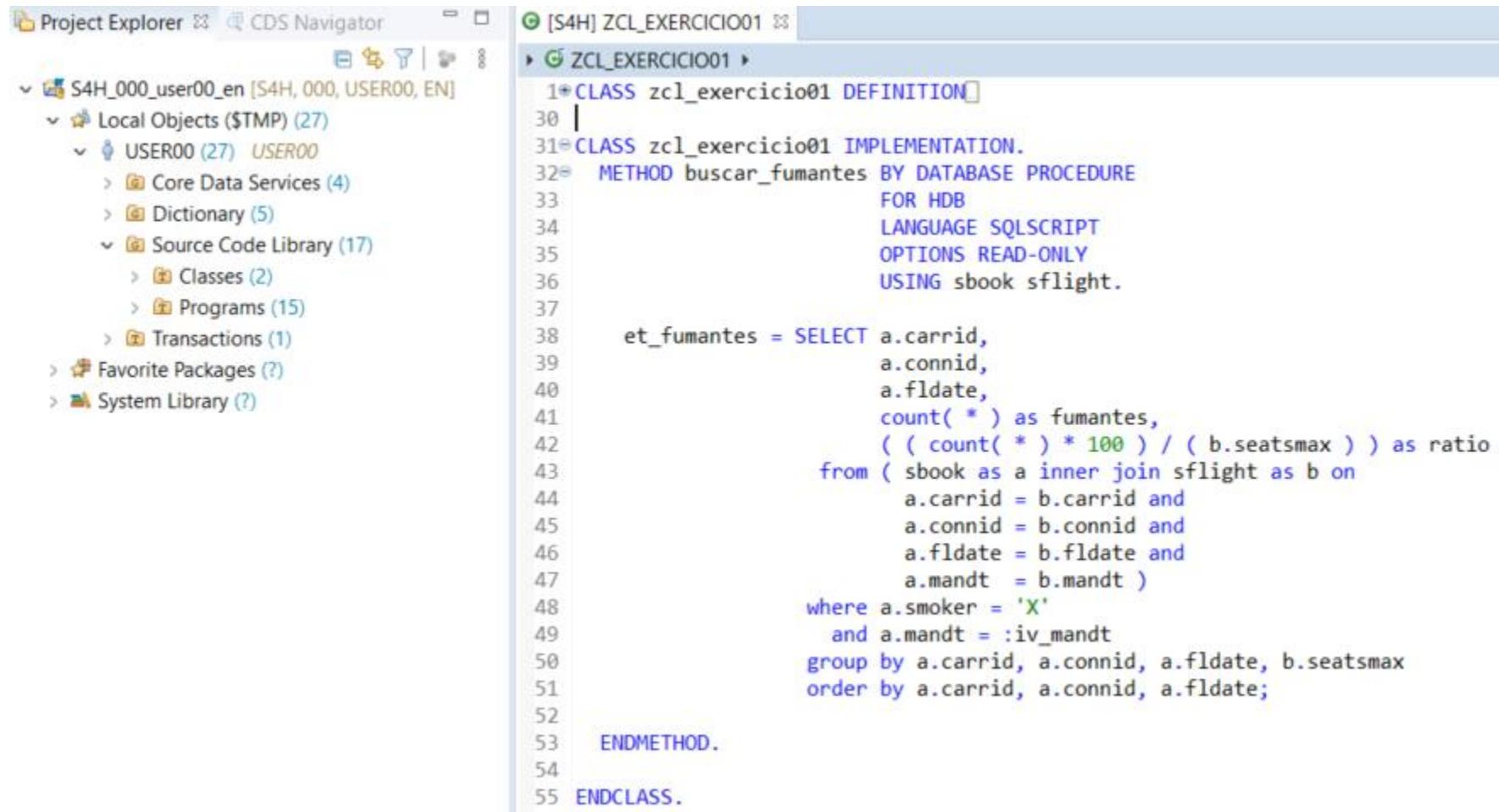
**Project Explorer:**

- S4H\_000\_user00\_en [S4H, 000, USER00, EN]
  - Local Objects (\$TMP) (27)
    - USER00 (27) *USER00*
      - Core Data Services (4)
      - Dictionary (5)
      - Source Code Library (17)
        - Classes (2)
        - Programs (15)
        - Transactions (1)

```
1 CLASS zcl_exercicio01 DEFINITION
2 PUBLIC
3 FINAL
4 CREATE PUBLIC .
5
6 PUBLIC SECTION.
7 INTERFACES if_amdp_marker_hdb.
8
9 TYPES:
10 BEGIN OF ty_sum,
11 carrid TYPE sbook-carrid,
12 connid TYPE sbook-connid,
13 fldate TYPE sbook-fldate,
14 fumantes TYPE i,
15 ratio TYPE f,
16 END OF ty_sum.
17
18 TYPES:
19 tt_sum TYPE STANDARD TABLE OF ty_sum.
20
21 CLASS-METHODS buscar_fumantes
22 IMPORTING
23 VALUE(iv_mandt) TYPE mandt
24 EXPORTING
25 VALUE(et_fumantes) TYPE tt_sum.
26
27 PROTECTED SECTION.
28 PRIVATE SECTION.
29 ENDCLASS.
```

# Programação em HANA – Procedures

- Solução – Exercicio1:



The screenshot shows the SAP Studio interface. On the left, the Project Explorer displays a tree structure for 'S4H\_000\_user00\_en [S4H, 000, USER00, EN]'. Under 'Local Objects (\$TMP) (27)', there is a node 'USER00 (27) USER00' which contains 'Core Data Services (4)', 'Dictionary (5)', 'Source Code Library (17)' (which further contains 'Classes (2)' and 'Programs (15)'), and 'Transactions (1)'. Other nodes include 'Favorite Packages (?)' and 'System Library (?)'. On the right, the code editor shows the content of the file 'ZCL\_EXERCICIO01'. The code defines a class 'zcl\_exercicio01' with an implementation method 'buscar\_fumantes'.

```
1*CLASS zcl_exercicio01 DEFINITION
30 |
31*CLASS zcl_exercicio01 IMPLEMENTATION.
32* METHOD buscar_fumantes BY DATABASE PROCEDURE
33 FOR HDB
34 LANGUAGE SQLSCRIPT
35 OPTIONS READ-ONLY
36 USING sbook sflight.
37
38 et_fumantes = SELECT a.carrid,
39 a.connid,
40 a.fldate,
41 count(*) as fumantes,
42 ((count(*) * 100) / (b.seatsmax)) as ratio
43 from (sbook as a inner join sflight as b on
44 a.carrid = b.carrid and
45 a.connid = b.connid and
46 a.fldate = b.fldate and
47 a.mandt = b.mandt)
48 where a.smoker = 'X'
49 and a.mandt = :iv_mandt
50 group by a.carrid, a.connid, a.fldate, b.seatsmax
51 order by a.carrid, a.connid, a.fldate;
52
53 ENDMETHOD.
54
55 ENDCLASS.
```

# Programação em HANA – Procedures

- Solução – Exercicio1:

```
CALL METHOD zcl_exercicio01=>buscar_fumantes
 EXPORTING
 iv_mandt = sy-mandt
 IMPORTING
 et_fumantes = DATA(lt_fumantes)
 .
```

## Programação em HANA – Procedures

- Exercicio2:
  - Criar um relatório das 4 formas possíveis (Procedure, CDS VIEW, OPEN SQL e ADBC) para mostrar a partir de um range de data de seleção (SBOOK- ORDERDATE), quantas passagens cada cliente comprou
  - O relatório deverá conter:
    - SBOOK-CUSTOMID – ID DO CLIENTE
    - SBOOK-PASSNAME – NOME DO CLIENTE
    - Total de Número de passagens compradas por período
    - SCUSTOM-CITY – Cidade do cliente
    - SCUSTOM-EMAIL – EMAIL DO CLIENTE
    - SBUSPART-CONTPHONO – Telefone do cliente
    - Caso numero de passagens for:
      - Entre 1000 e 500, exibir “Cliente VIP”
      - Entre 250 e 500, exibir “Cliente Preferencial”
      - Entre 0 e 250, exibir “Cliente Normal”
  - Obs: Não contar passagens canceladas (SBOOK-CANCELED = X)

# Programação em HANA – Procedures

- Solução – Exercicio 2:
- PROCEDURE AMDP

```
CLASS ZCL_TESTE IMPLEMENTATION.
METHOD busca_total BY DATABASE PROCEDURE FOR HDB
 LANGUAGE SQLSCRIPT
 OPTIONS READ-ONLY
 USING sbook scustom sbuspart.

et_total_pas =
 SELECT sbook.customid as id_pass,
 sbook.passname as nome_pass,
 count(*) as num_tot_pass,
 scustom.city as cid_pass,
 scustom.email as email_pass,
 sbuspart.conphono as tel_pass,
 CASE
 WHEN count(*) BETWEEN 500 AND 1000 then 'Cliente VIP'
 WHEN count(*) BETWEEN 250 AND 499 then 'Cliente Preferencial'
 ELSE 'Cliente Normal'
 END AS text_cliente
 FROM
 ((sbook as sbook INNER JOIN scustom as scustom
 ON
 sbook.customid = scustom.id AND
 sbook.mandt = scustom.mandt)
 LEFT OUTER JOIN sbuspart as sbuspart
 ON
 sbook.customid = sbuspart.buspartnum)
 GROUP BY sbook.customid, sbook.passname, scustom.city, scustom.email,
 sbuspart.conphono;
```

# Programação em HANA – Procedures

- Solução – Exercício 2:

- CDS VIEW

```
@AbapCatalog.sqlViewName: ''
@AbapCatalog.compiler.CompareFilter: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'exericicio'
define view zteste as select
 sbook.customid as id_pass,
 sbook.passname as nome_pass,
 count(*) as num_tot_pass,
 scustom.city as cid_pass,
 scustom.email as email_pass,
 sbuspart.contphono as tel_pass
 from
 ((sbook as sbook inner join scustom as scustom
 on
 sbook.customid = scustom.id and
 sbook.mandt = scustom.mandt)
 left outer join sbuspart as sbuspart
 on
 sbook.customid = sbuspart.busparthnum)
 group by sbook.customid, sbook.passname, scustom.city, scustom.email,
 sbuspart.contphono;
```

# Programação em HANA – Procedures

- Solução – Exercício 2:
  - OPEN SQL

```
SELECT
 id_pass,
 nome_pass,
 num_tot_pass,
 cid_pass,
 email_pass,
 tel_pass,
 CASE
 WHEN num_tot_pass BETWEEN 500 AND 1000 THEN 'Cliente VIP'
 WHEN num_tot_pass BETWEEN 250 AND 499 THEN 'Cliente Preferencial'
 ELSE 'Cliente Normal'
 END AS texto_cliente
 FROM ztesteex5 AS ztesteex5 INTO TABLE @DATA(t_total_cds).
IF t_total_cds IS NOT INITIAL.

ENDIF.
```

# Programação em HANA – Procedures

- Solução – Exercicio 2:
  - ADBC:

```
TRY.
 CREATE OBJECT lo_sql EXPORTING con_ref = cl_sql_connection->get_connection('BEH').
 lo_result = lo_sql->execute_query(
 | SELECT sbook.customid as id_pass, | &&
 | sbook.passname as nome_pass, | &&
 | count(*) as num_tot_pass, | &&
 | scustom.city as cid_pass, | &&
 | scustom.email as email_pass, | &&
 | sbuspart.contphono as tel_pass, | &&
 | CASE | &&
 | WHEN count(*) BETWEEN 500 AND 1000 then 'Cliente VIP' | &&
 | WHEN count(*) BETWEEN 250 AND 499 then 'Cliente Preferencial' | &&
 | ELSE 'Cliente Normal' | &&
 | END AS text_cliente | &&
 | FROM | &&
 | (SAPABAP1.sbook as sbook INNER JOIN SAPABAP1.scustom as scustom | &&
 | ON | &&
 | sbook.customid = scustom.id AND | &&
 | sbook.mandt = scustom.mandt) | &&
 | LEFT OUTER JOIN SAPABAP1.sbuspart as sbuspart | &&
 | ON | &&
 | sbook.customid = sbuspart.buspartnum) | &&
 | WHERE sbook.mandt = '{ sy-mandt }' | &&
 | GROUP BY sbook.customid, sbook.passname, scustom.city, scustom.email, | &&
 | sbuspart.contphono
).

 lo_result->set_param_table(REF #(t_total_adbc)).
 lo_result->next_package().
 lo_result->close().
```

# Programação em Hana - Visões HANA

## Programação em HANA – Visões HANA

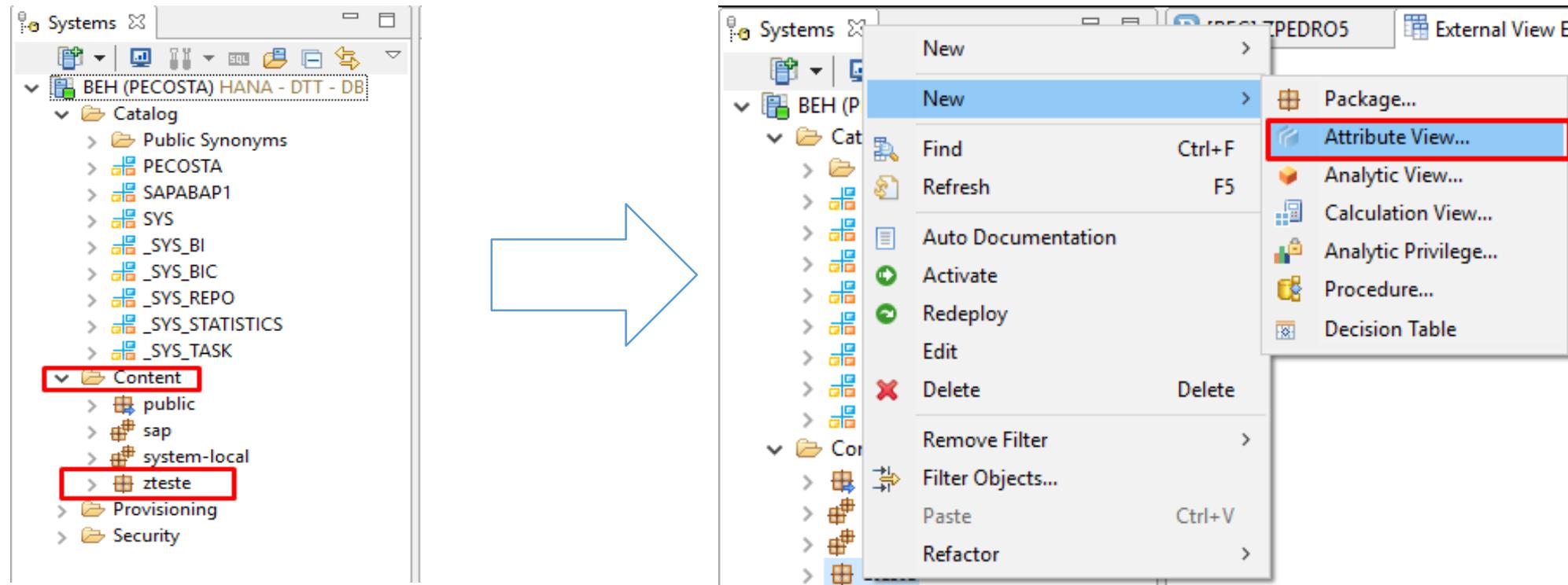
- Além de todas as soluções propostas acima para desenvolvimento com SQL Nativo e aproveitamento do HANA, a SAP provê outro conjunto de ferramentas chamadas Visões HANA
- Muitas vezes, codificar tudo em SQL torna difícil a reusabilidade do resultado da seleção de dados e cálculos. Portanto, essas visões foram criadas para reaproveitar o resultado em diferentes programas e também para diminuir a programação SQL
- A SAP provê três tipos de visões: Attribute View, Analytic View e Calculation View

## Programação em HANA – Visões HANA

- A Attribute View é similar ao CDS View:
  - View de dados mestre SAP que pode conter:
    - Filtros;
    - Colunas com cálculos gráficos;
    - Possui mais funcionalidades que uma visão na SE11;
    - Criação da View dentro do Banco de Dados Hana
    - Permite conectar a visão com o Dicionário ABAP(SE11)

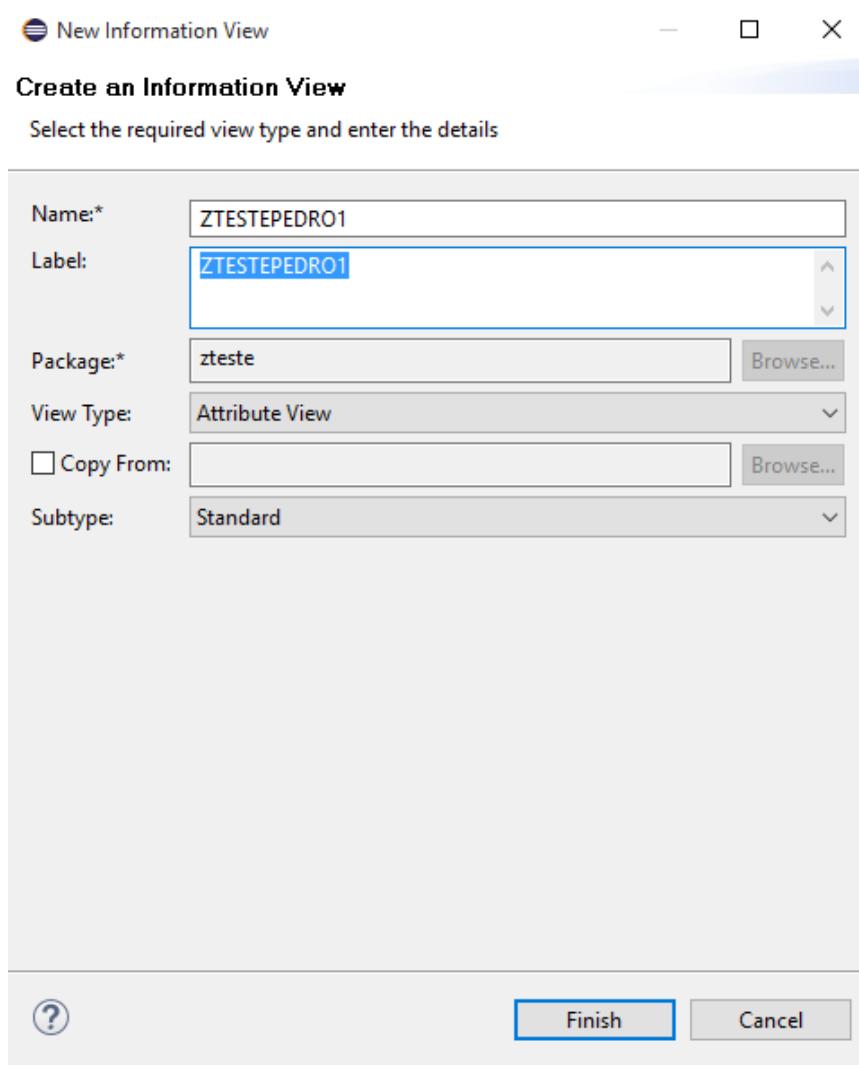
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



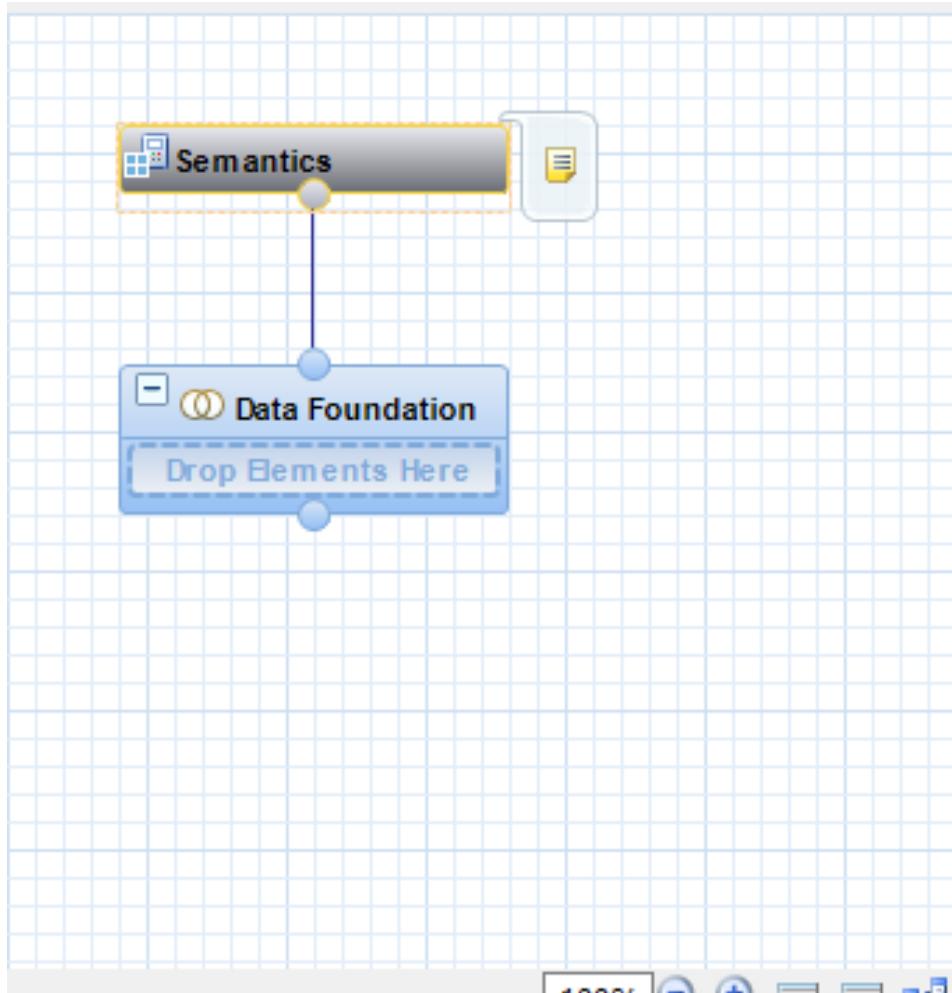
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



# Programação em HANA – Visões HANA

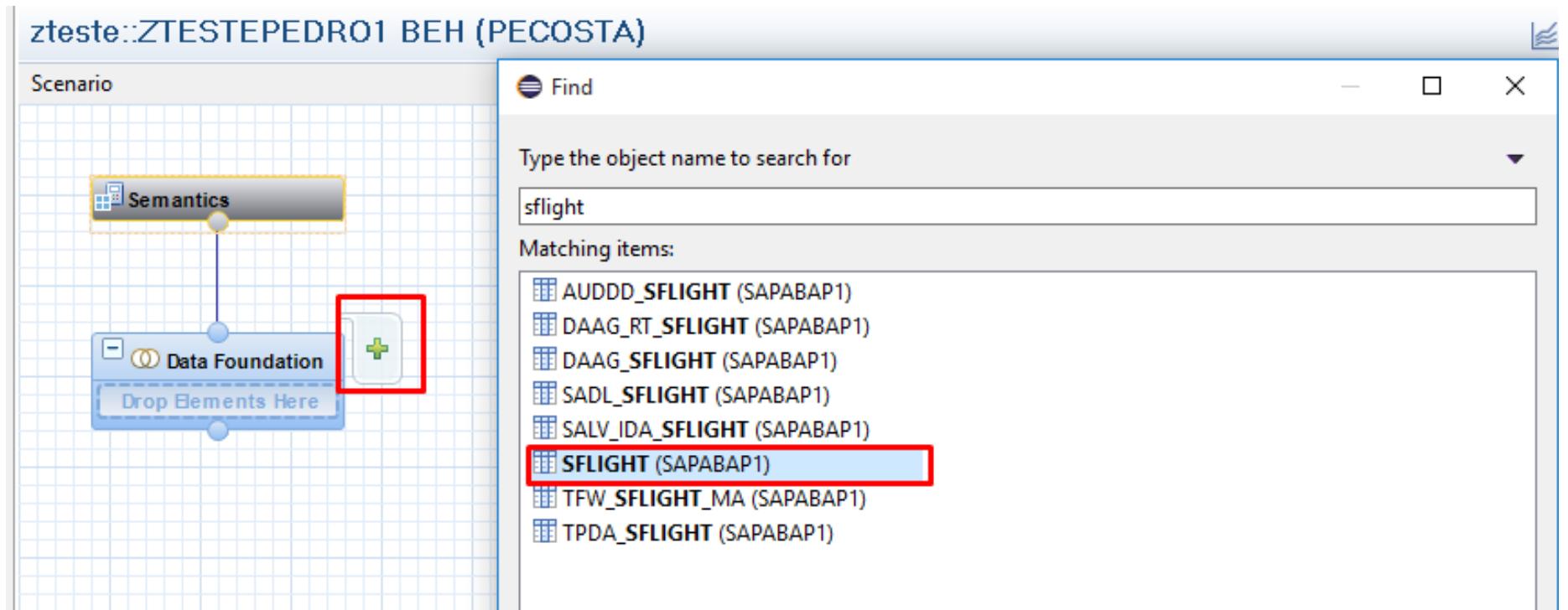
- Exemplo 1 – Attribute View:



- Data Foundation é onde terá as tabelas que farão ligação com a view bem como, regras, e colunas calculadas
- Semantics são as informações de como a visão deve ser criada: Indicar quais campos são chaves, bem como a posição e entre outros

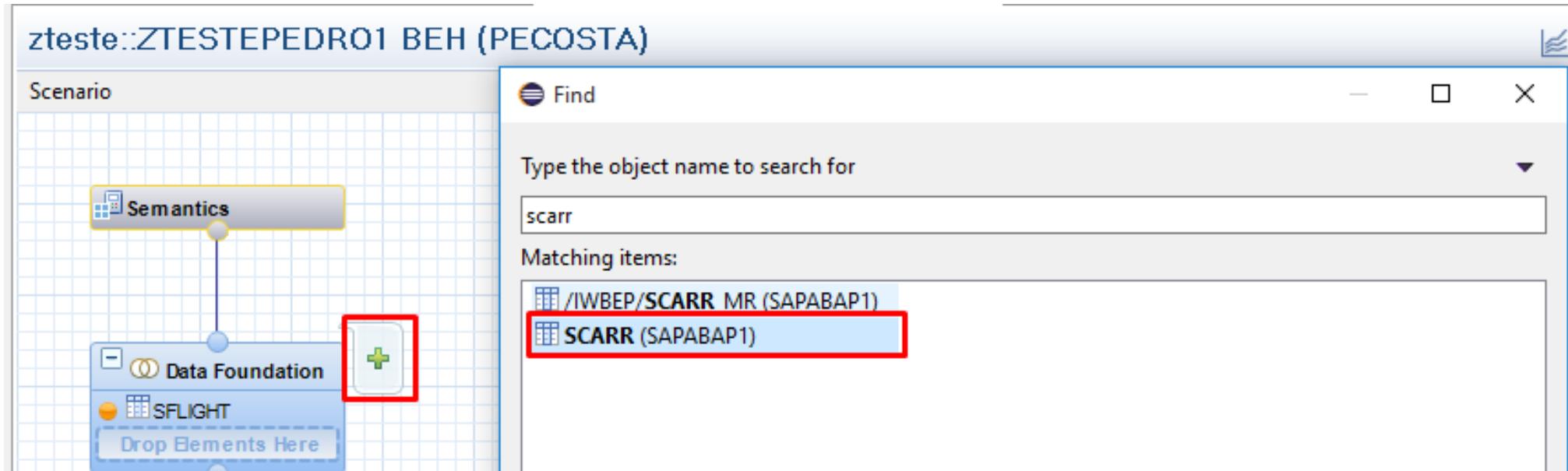
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



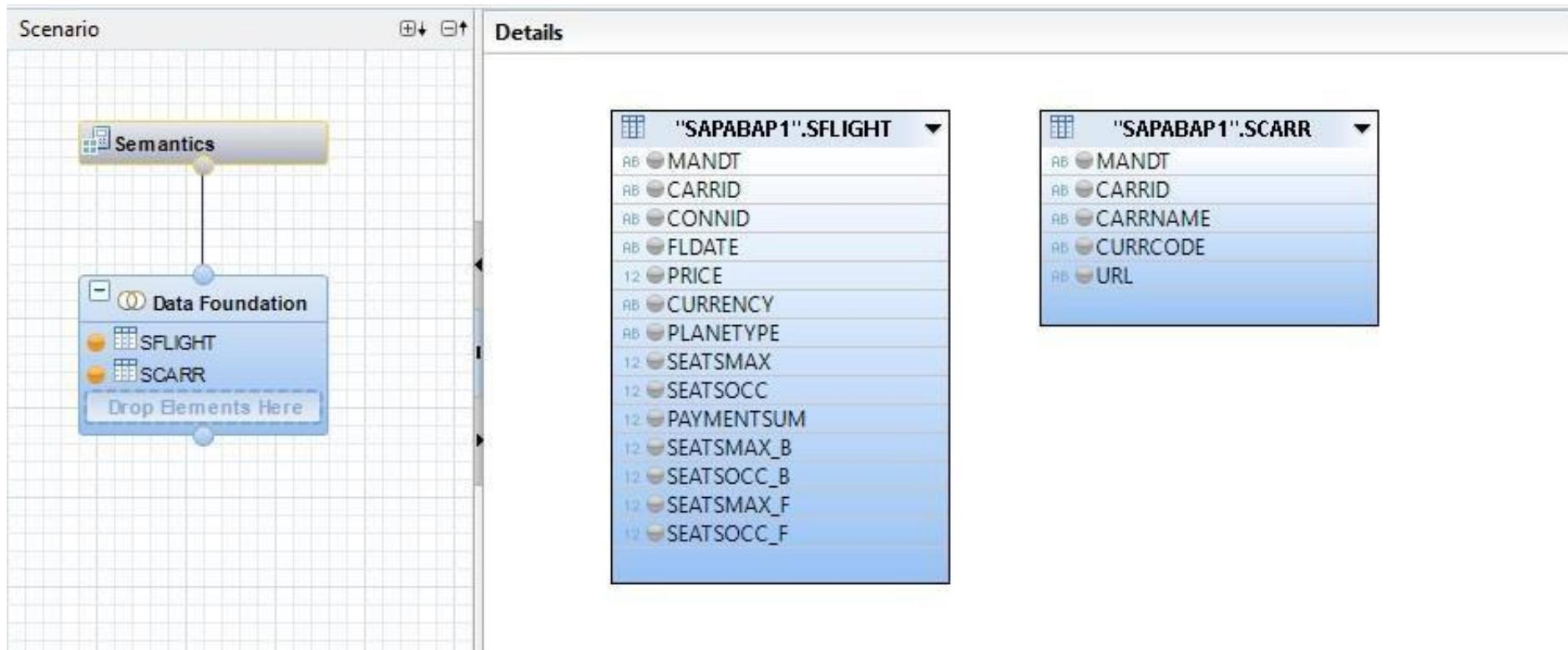
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



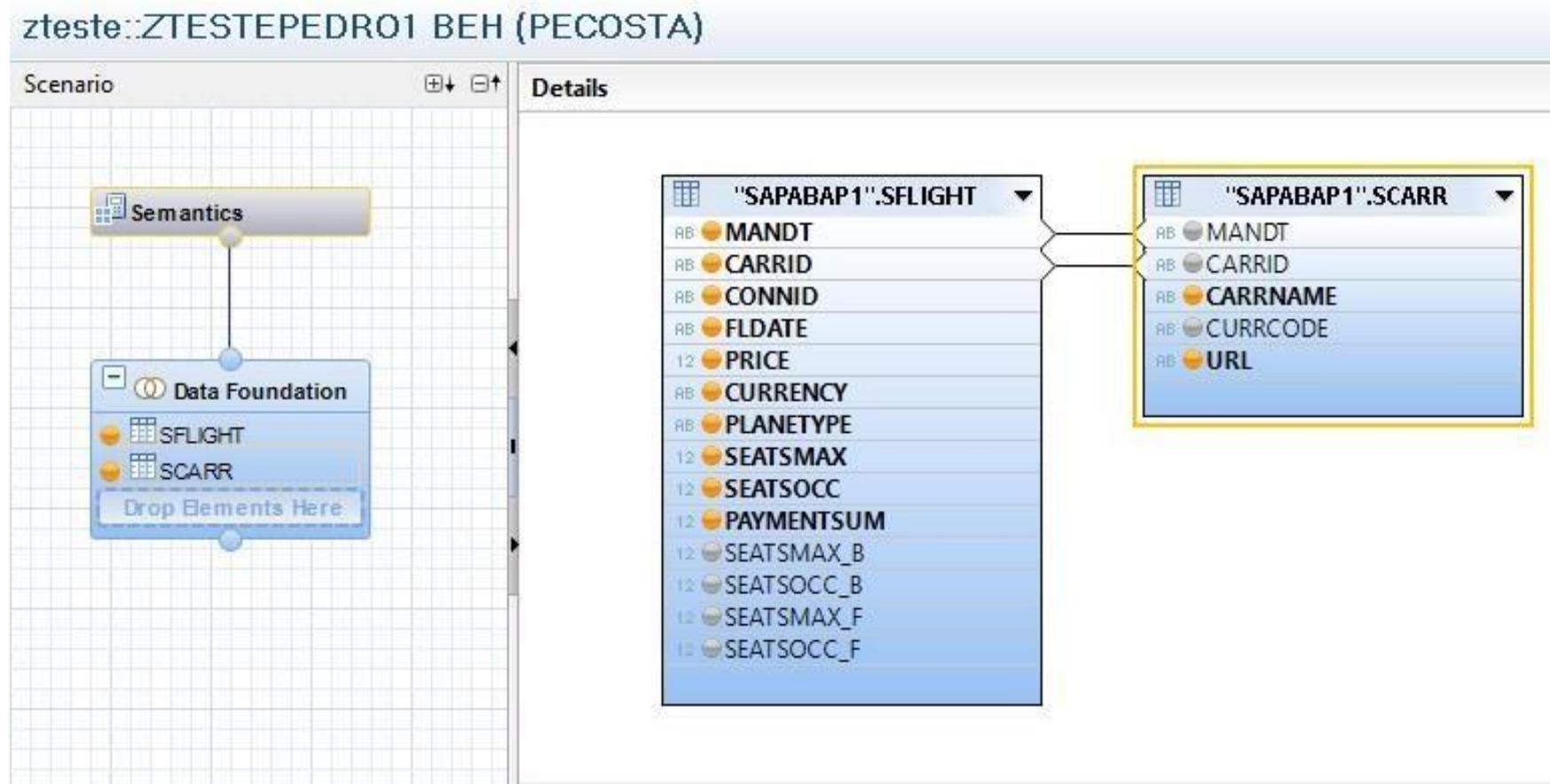
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



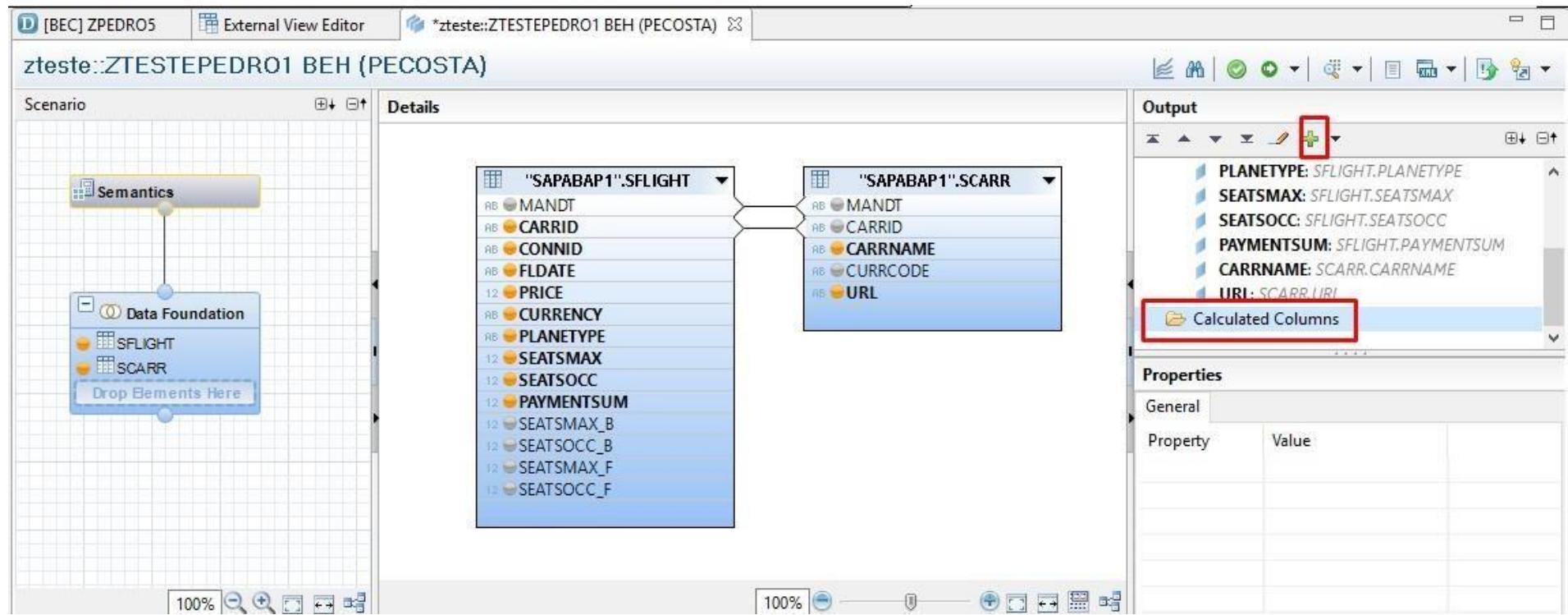
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



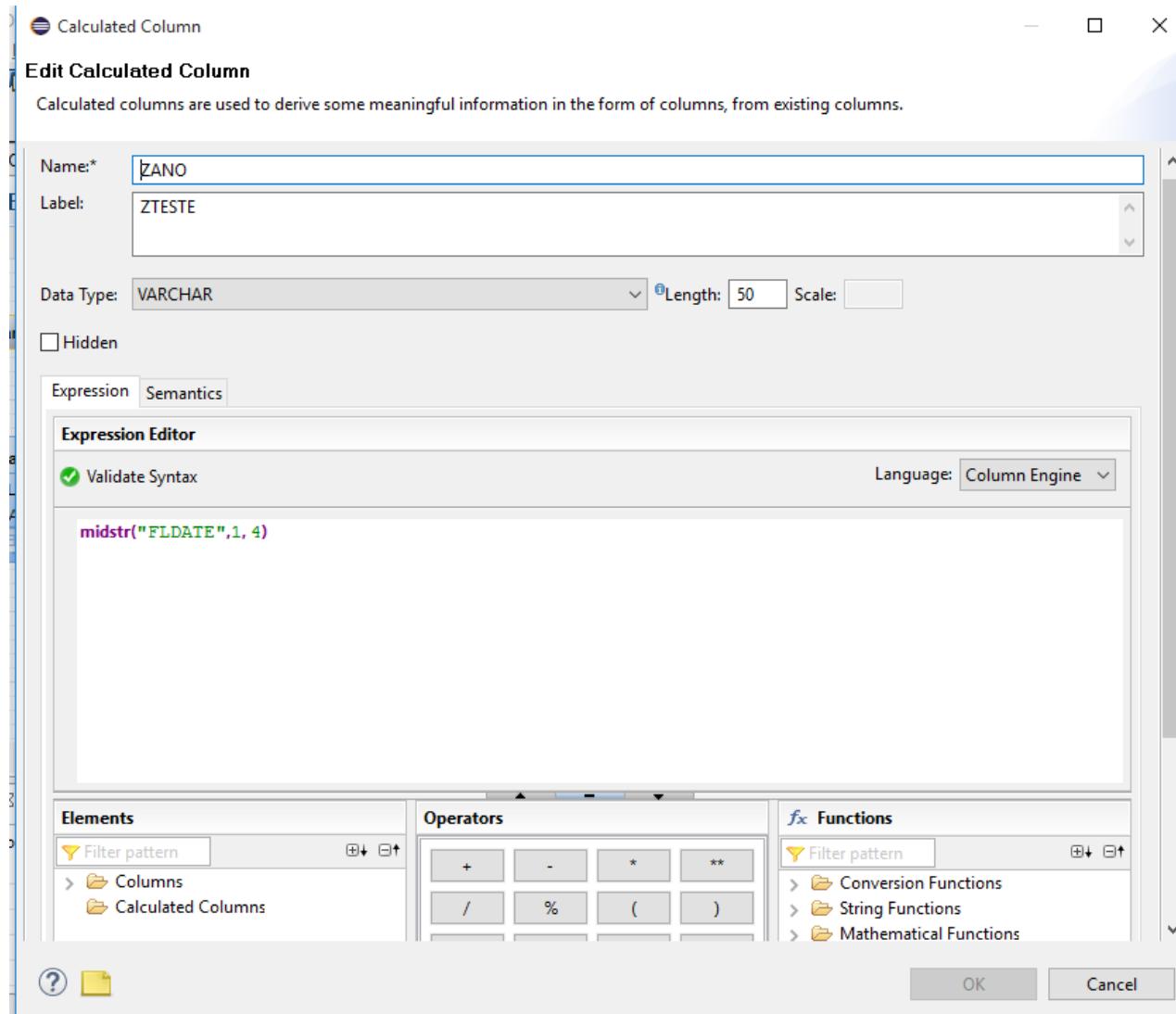
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



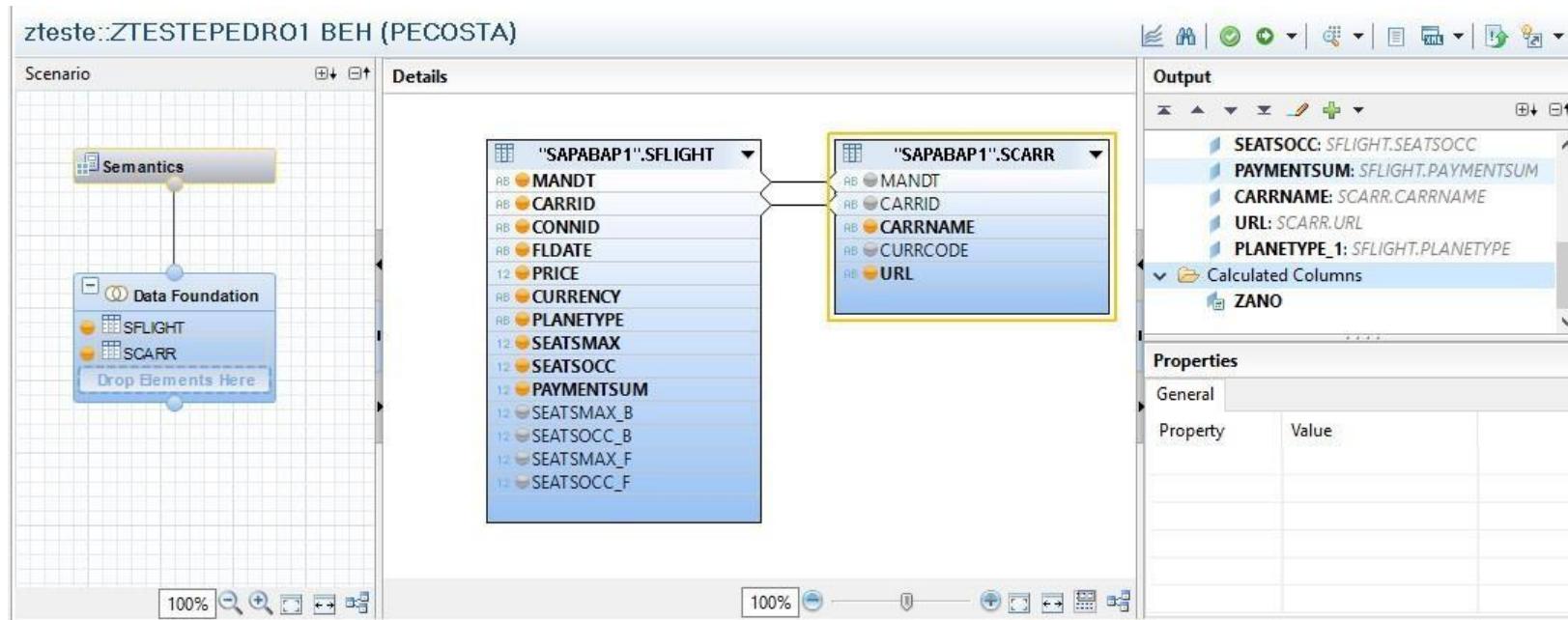
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



# Programação em HANA – Visões HANA

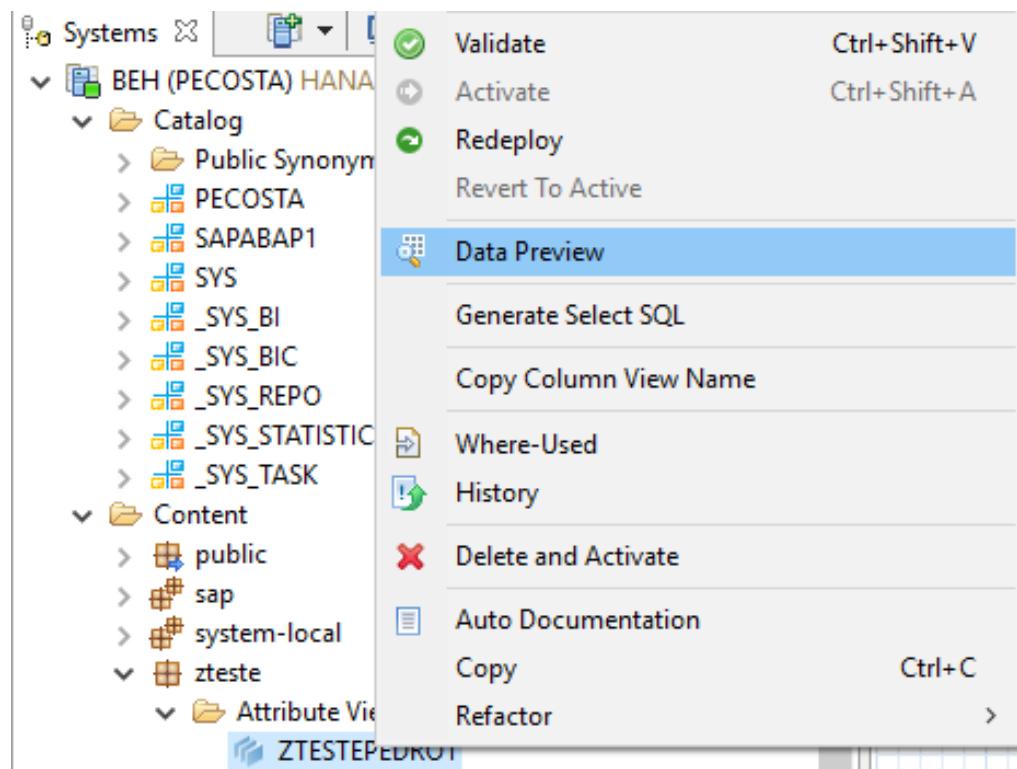
- Exemplo 1 – Attribute View:

The screenshot shows the SAP HANA Studio interface for creating an Attribute View. The left sidebar navigation tree includes 'Systems', 'Catalog' (with 'Public Synonyms', 'PECOSTA', 'SAPABAP1', 'SYS', '\_SYS\_BI', '\_SYS\_BIC', '\_SYS\_REPO', '\_SYS\_STATISTICS', '\_SYS\_TASK'), 'Content' (with 'public', 'sap', 'system-local'), and 'zteste' (with 'Attribute Views (4)' containing 'ZTESTEPEDRO1', 'ZVIEW\_TESTE', 'ZVIEW\_TESTE1', 'ZVIEW\_TESTE2') and 'Analytic Views (1)'. The main workspace shows a 'Scenario' diagram with 'Semantics' and 'Data Foundation' nodes, and a table mapping local columns to labels. The table details are as follows:

| Type | Key | Name       | Label      | Semantic Type | Label Column | Hidden                   | Conversion Functions |
|------|-----|------------|------------|---------------|--------------|--------------------------|----------------------|
| RB   |     | MANDT      | MANDT      |               |              | <input type="checkbox"/> |                      |
| RB   |     | CARRID     | CARRID     |               |              | <input type="checkbox"/> |                      |
| RB   |     | CONNID     | CONNID     |               |              | <input type="checkbox"/> |                      |
| RB   |     | FLDATE     | FLDATE     |               |              | <input type="checkbox"/> |                      |
| 12   |     | PRICE      | PRICE      |               |              | <input type="checkbox"/> |                      |
| RB   |     | CURRENCY   | CURRENCY   |               |              | <input type="checkbox"/> |                      |
| RB   |     | PLANETYPE  | PLANETYPE  |               |              | <input type="checkbox"/> |                      |
| 12   |     | SEATSMAX   | SEATSMAX   |               |              | <input type="checkbox"/> |                      |
| 12   |     | SEATSOCC   | SEATSOCC   |               |              | <input type="checkbox"/> |                      |
| 12   |     | PAYMENTSUM | PAYMENTSUM |               |              | <input type="checkbox"/> |                      |
| RB   |     | CARRNAME   | CARRNAME   |               |              | <input type="checkbox"/> |                      |
| RB   |     | URL        | URL        |               |              | <input type="checkbox"/> |                      |

# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:

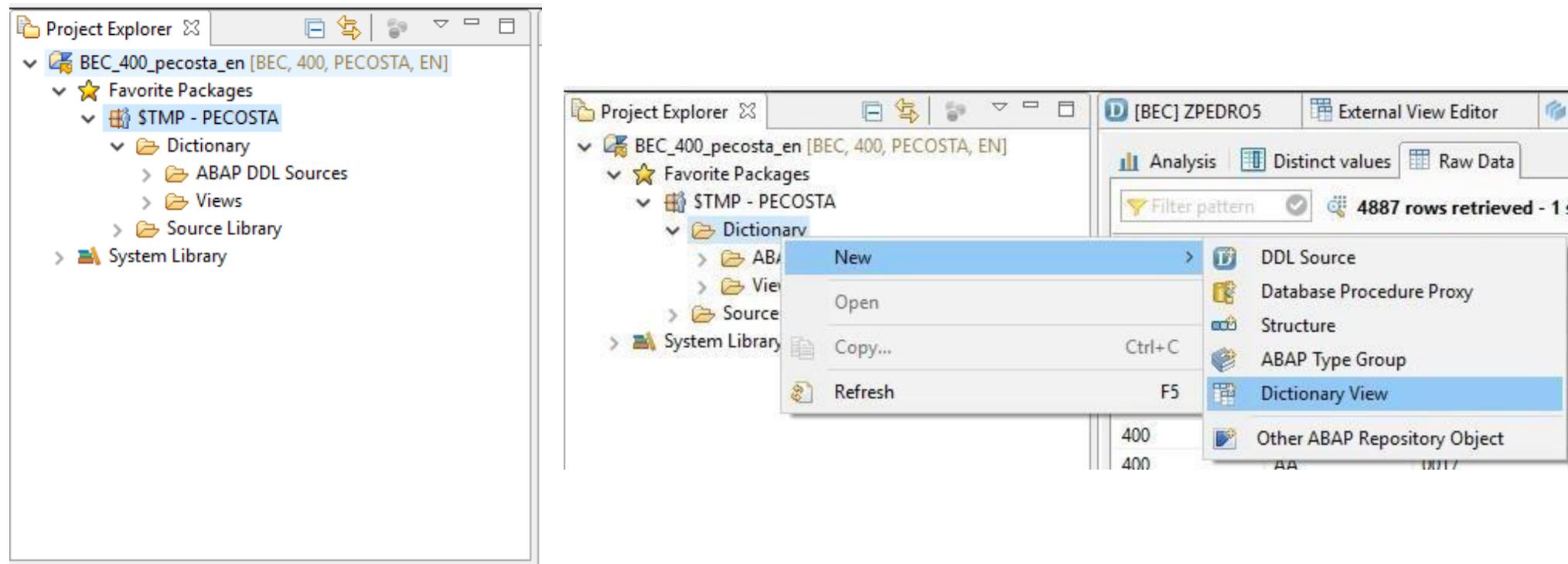
The screenshot shows a SAP HANA Studio interface displaying the results of an Attribute View query. The results are presented in a grid table with the following columns:

| AB  | MANDT | AB | CARRID | AB   | CONNID | AB       | FDATE | 12     | PRICE | AB | CURRENCY | AB | PLANETYPE | 12  | SEATSMAX |
|-----|-------|----|--------|------|--------|----------|-------|--------|-------|----|----------|----|-----------|-----|----------|
| 400 | AA    |    |        | 0017 |        | 20141203 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20141210 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20141217 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20141224 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20141231 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150107 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150114 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150121 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150128 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150204 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150211 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150218 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150225 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150304 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150311 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |
| 400 | AA    |    |        | 0017 |        | 20150318 |       | 422.94 | USD   |    | 747-400  |    |           | 385 |          |

The interface includes various navigation and filtering tools at the top and right side of the table.

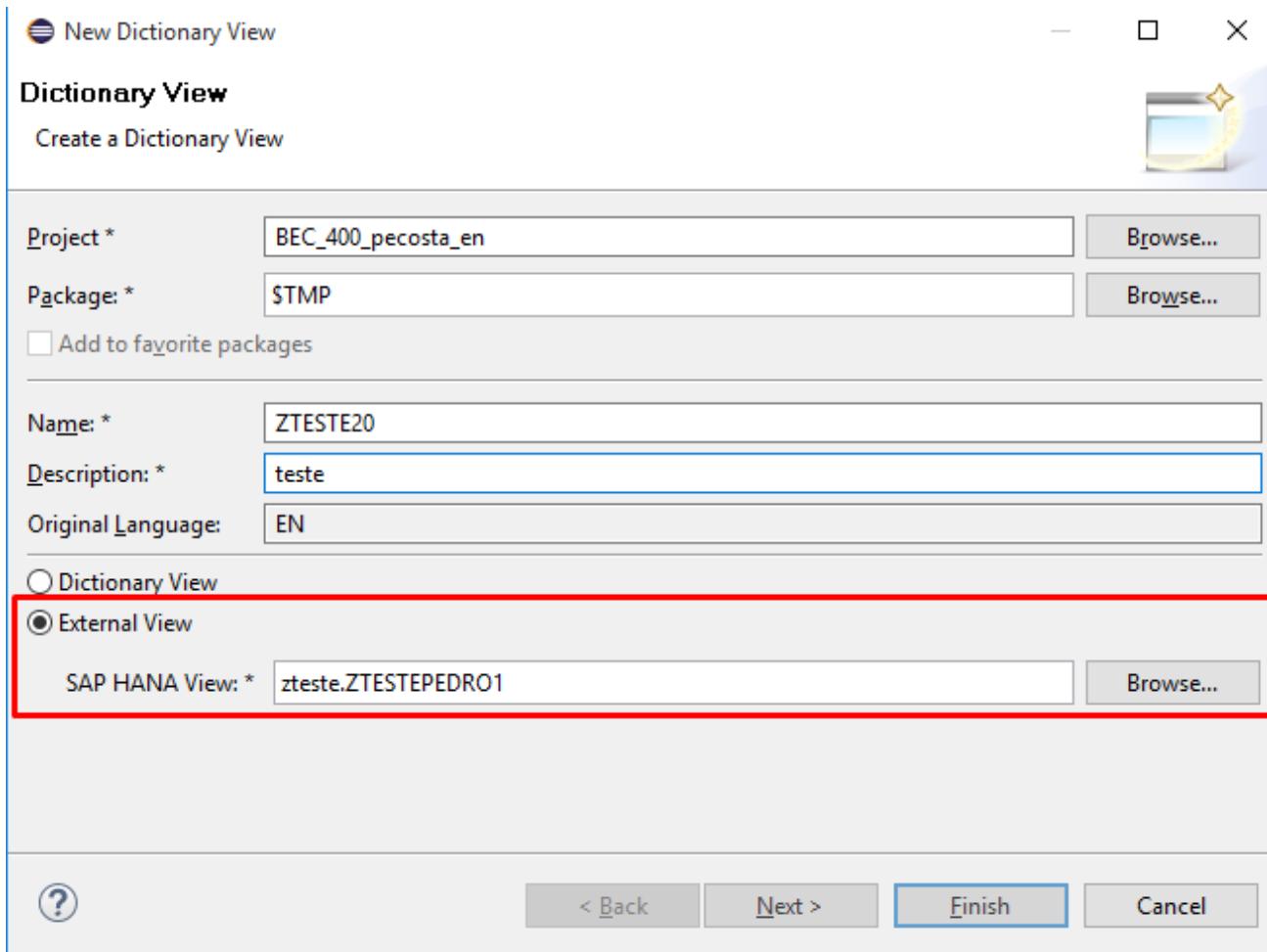
# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View – Replicando para SE11:



# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:



# Programação em HANA – Visões HANA

- Exemplo 1 – Attribute View:

The screenshot shows the SAP HANA Studio interface. On the left, a context menu is open over a dictionary view named "ZTESTE20". The menu items include "New Dictionary View", "Open", "Open in Project" (with a keyboard shortcut "Ctrl+Alt+P"), "Open With", "Copy..." (with a keyboard shortcut "Ctrl+C"), "Delete" (with a keyboard shortcut "Delete"), "Get where-used list...", "Open Data Preview", "Refresh" (with a keyboard shortcut "F5"), "Share Link...", and "Activate". The "Activate" option is highlighted with a red box. On the right, the "Dictionary: Display View" window is displayed. It shows the external view "ZTESTE20" is active, with a short description "teste". The window has tabs for "Attributes", "Table/Join Conditions", "View Fds", "Selection Conditions", and "Maint.Status". The "Attributes" tab is selected, showing a table of fields. The table has columns: View field, Table, Field, Key, Data elem., M..., DTyp, Length, and Short description. The data in the table is as follows:

| View field  | Table    | Field       | Key                                 | Data elem. | M...                     | DTyp | Length | Short description |
|-------------|----------|-------------|-------------------------------------|------------|--------------------------|------|--------|-------------------|
| MANDT       | ZTESTE20 | MANDT       | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CLNT | 3      |                   |
| CARRID      | ZTESTE20 | CARRID      | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 3      |                   |
| CONNID      | ZTESTE20 | CONNID      | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 4      |                   |
| FLDATE      | ZTESTE20 | FLDATE      | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 8      |                   |
| PRICE       | ZTESTE20 | PRICE       | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | DEC  | 15     |                   |
| CURRENCY    | ZTESTE20 | CURRENCY    | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 5      |                   |
| PLANETYPE   | ZTESTE20 | PLANETYPE   | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 10     |                   |
| SEATSMAX    | ZTESTE20 | SEATSMAX    | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | INT4 | 10     |                   |
| SEATSOCC    | ZTESTE20 | SEATSOCC    | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | INT4 | 10     |                   |
| PAYMENTSUM  | ZTESTE20 | PAYMENTSUM  | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | DEC  | 17     |                   |
| CARRNAME    | ZTESTE20 | CARRNAME    | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 20     |                   |
| URL         | ZTESTE20 | URL         | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 255    |                   |
| PLANETYPE_1 | ZTESTE20 | PLANETYPE_1 | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 10     |                   |
| ZANO        | ZTESTE20 | ZANO        | <input checked="" type="checkbox"/> |            | <input type="checkbox"/> | CHAR | 50     |                   |

## Exercicio 1 – Visões HANA

- Refaça os exercícios 2 e 3 do CDS View, para uma Attribute View. Crie conexão da Attribute View com a SE11, realize seleção de dados através de um programa e compare o tempo da CDS VIEW com a Attribute View:
  - SFLIGHT-CARRID + ‘-’ + SFLIGHT-CONNID
  - DATA do Voo -> SFLIGHT-FLDATE
  - Tipo do avião -> SFLIGHT-PLANETYPE
  - Capacidade classe econômica -> SFLIGHT-SEATSMAX
  - Assentos ocupados da Classe econômica -> SFLIGHT-SEATSOCC
  - Número de assentos da classe econômica disponíveis(Sflight-seatsmax –sflight-seatsocc)
  - Total em valor(SFLIGHT-PAYMENTSUM)
  - Moeda(SFLIGHT-CURRENCY)
  - Se o SFLIGHT-PAYMENTSUM for > 190.000,00 “Voo lucrativo”, porém SFLIGHT-PAYMENTSUM seja menor que 190.000,00 exibir “Voo com pouco lucro”, caso não haja nenhuma das duas opções, exibir “Voo sem margem de lucro definida”

# Solução Exercício 1 – Visões HANA

zteste::ZTESTEEEX30 BEH (PECOSTA)

Scenario: zteste::ZTESTEEEX30 BEH (PECOSTA)

Details:

- "SAPABAP1".SFLIGHT
- AB CARRID
- AB CONNID
- AB FLDAT
- 12 PRICE
- RB CURRENCY
- RB PLANETYPE
- 12 SEATSMAX
- 12 SEATSOCC
- 12 PAYMENTSUM
- 12 SEATSMAX\_B
- 12 SEATSOCC\_B
- 12 SEATSMAX\_F
- 12 SEATSOCC\_F

Output:

- FLDATE: SFLIGHT.FLDAT
- PLANETYPE: SFLIGHT.PLANETYPE
- SEATSMAX: SFLIGHT.SEATSMAX
- SEATSOCC: SFLIGHT.SEATSOCC
- PAYMENTSUM: SFLIGHT.PAYMENTSUM
- CURRENCY: SFLIGHT.CURRENCY

Calculated Columns:

- NUM\_VOO
- ASS\_ECON\_DISP
- TEXT\_PAYMENT

Properties:

| Property | Value  |
|----------|--------|
| Name     | CARRID |

# Solução Exercicio 1 – Visões HANA

Calculated Column

## Edit Calculated Column

Calculated columns are used to derive some meaningful information in the form of columns, from existing columns.

Name\*: **NUM\_VOO**

Label: **NUM\_VOO**

Data Type: **VARCHAR** Length: **50** Scale: **0**

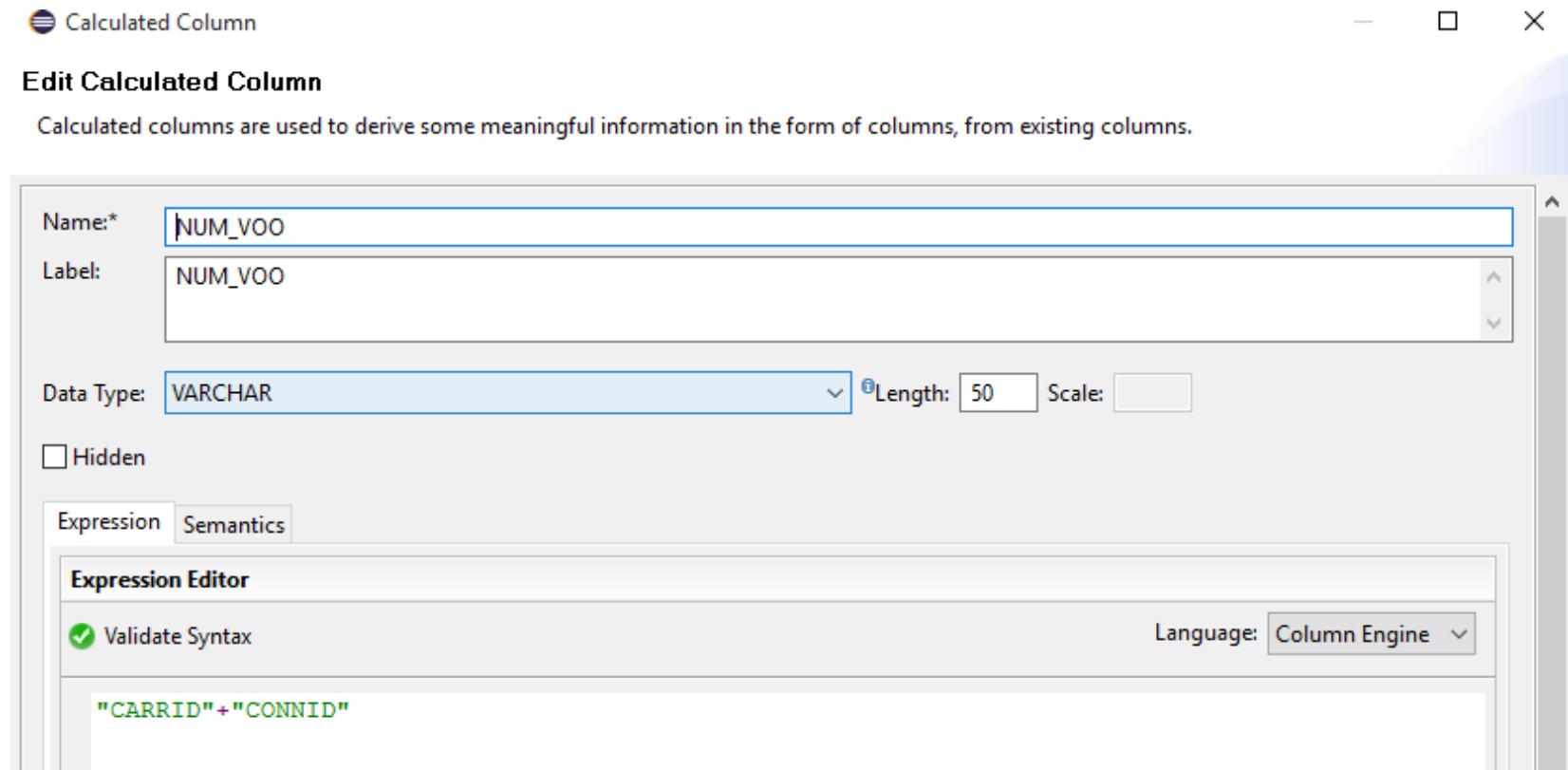
Hidden

**Expression Semantics**

**Expression Editor**

Validate Syntax Language: **Column Engine**

**"CARRID"+"CONNID"**



# Solução Exercicio 1 – Visões HANA

Calculated Column

## Edit Calculated Column

Calculated columns are used to derive some meaningful information in the form of columns, from existing columns.

Name\*: ASS\_ECON\_DISP

Label: ASS\_ECON\_DISP

Data Type: INTEGER Length: Scale:

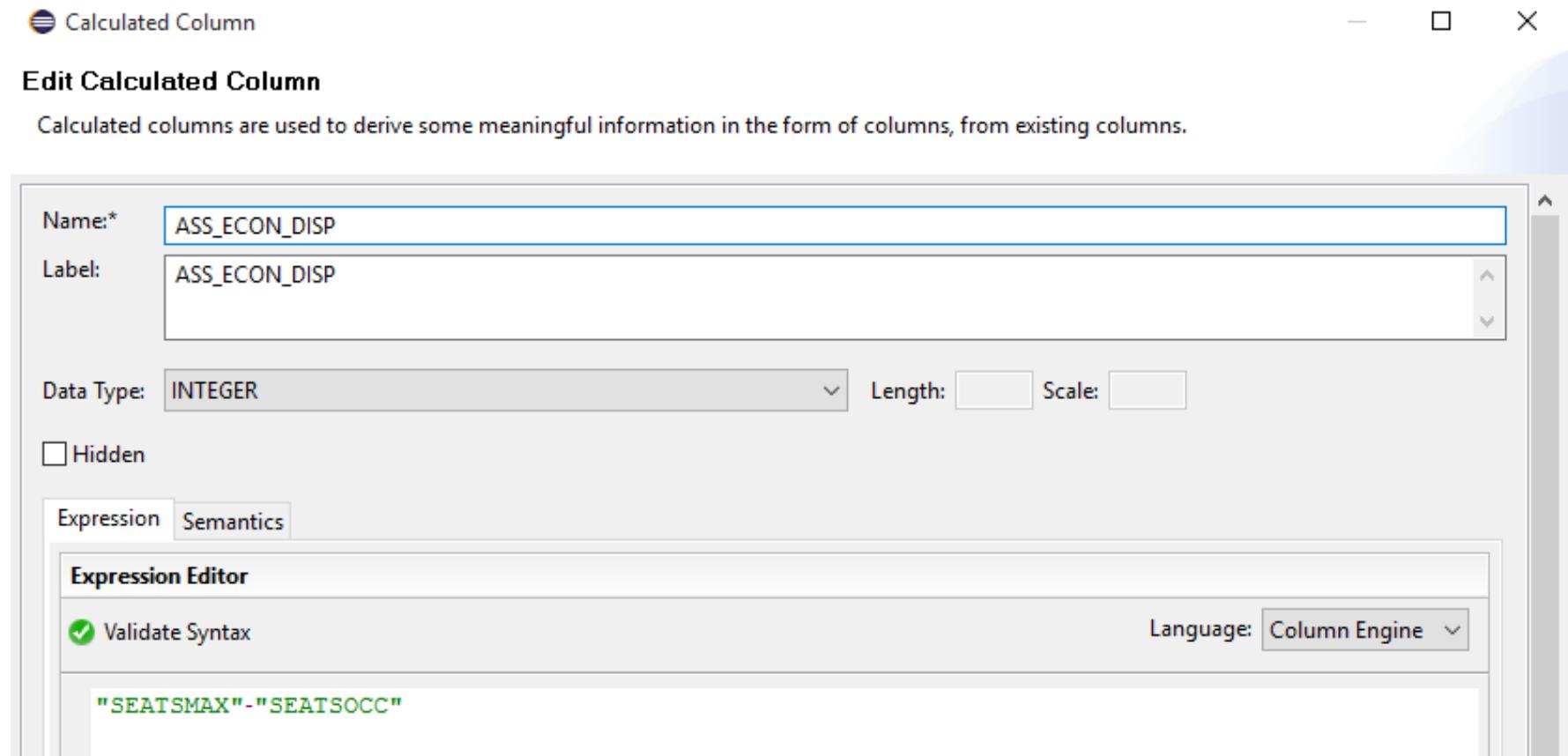
Hidden

Expression Semantics

Expression Editor

Validate Syntax Language: Column Engine

"SEATSMAX"- "SEATSOCC"



# Solução Exercicio 1 – Visões HANA

Calculated Column

## Edit Calculated Column

Calculated columns are used to derive some meaningful information in the form of columns, from existing columns.

The screenshot shows the 'Edit Calculated Column' dialog box. The 'Name\*' field is set to 'TEXT\_PAYMENT'. The 'Label' field also contains 'TEXT\_PAYMENT'. The 'Data Type' is 'VARCHAR' with a length of 50 and no scale. The 'Hidden' checkbox is unchecked. Below the main fields, there are two tabs: 'Expression' (selected) and 'Semantics'. The 'Expression Editor' section contains a checked 'Validate Syntax' button and a 'Language' dropdown set to 'Column Engine'. The expression itself is: `if("PAYMENTSUM" > float(190000), 'Voo Lucrativo', if("PAYMENTSUM" < float(190000), 'Voo com pouca margem de lucro', 'Voo com margem de lucro não definida'))`.

# Solução Exercicio 1 – Visões HANA

The screenshot shows the SAP HANA Studio interface for creating a view. The title bar indicates the view is named `*zteste::ZTESTEEEX30 BEH (PECOSTA)`. The left pane displays a semantic model diagram with nodes for `Semantics`, `Data Foundation`, and `ISFLIGHT`. The right pane shows the `Details` tab with the `Local` section selected. The table lists 12 columns:

| Type | Key                                 | Name             | Label       | Semantic Type | Label Column | Hidden                              | Conversion Functions |
|------|-------------------------------------|------------------|-------------|---------------|--------------|-------------------------------------|----------------------|
|      | <input checked="" type="checkbox"/> | AB MANDT         | MANDT       |               |              | <input type="checkbox"/>            |                      |
|      | <input checked="" type="checkbox"/> | AB NUM_VOO       | NUM_VOO     |               |              | <input type="checkbox"/>            |                      |
|      | <input checked="" type="checkbox"/> | AB FLDAT         | FLDATE      |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | AB CARRID        | CARRID      |               |              | <input checked="" type="checkbox"/> |                      |
|      | <input type="checkbox"/>            | AB CONNID        | CONNID      |               |              | <input checked="" type="checkbox"/> |                      |
|      | <input type="checkbox"/>            | AB PLANETYPE     | PLANETY...  |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | 12 SEATSMAX      | SEATSMAX    |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | 12 SEATSOCC      | SEATSOCC    |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | 12 PAYMENTSUM    | PAYMEN...   |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | AB CURRENCY      | CURRENCY    |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | 12 ASS_ECON_DISP | ASS_ECO...  |               |              | <input type="checkbox"/>            |                      |
|      | <input type="checkbox"/>            | AB TEXT_PAYMENT  | TEXT_PAY... |               |              | <input type="checkbox"/>            |                      |

# Solução Exercicio 1 – Visões HANA

## Programação em HANA – Visões HANA

- A Calculation View é a mais completa das View's que o SAP Hana Oferece:
  - Nessa View você pode criar graficamente a junção, união, agregação, projeção de qualquer tabela ou view que estiver no banco de dados
  - Permite criar colunas mais complexas do que a attribute view
  - Permite conexão com SE11
  - Além disso, é possível realizar operações de SUM, AVG, COUNT, MIN e outros em colunas de valor e inteiro

# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:

New Information View    — □ ×

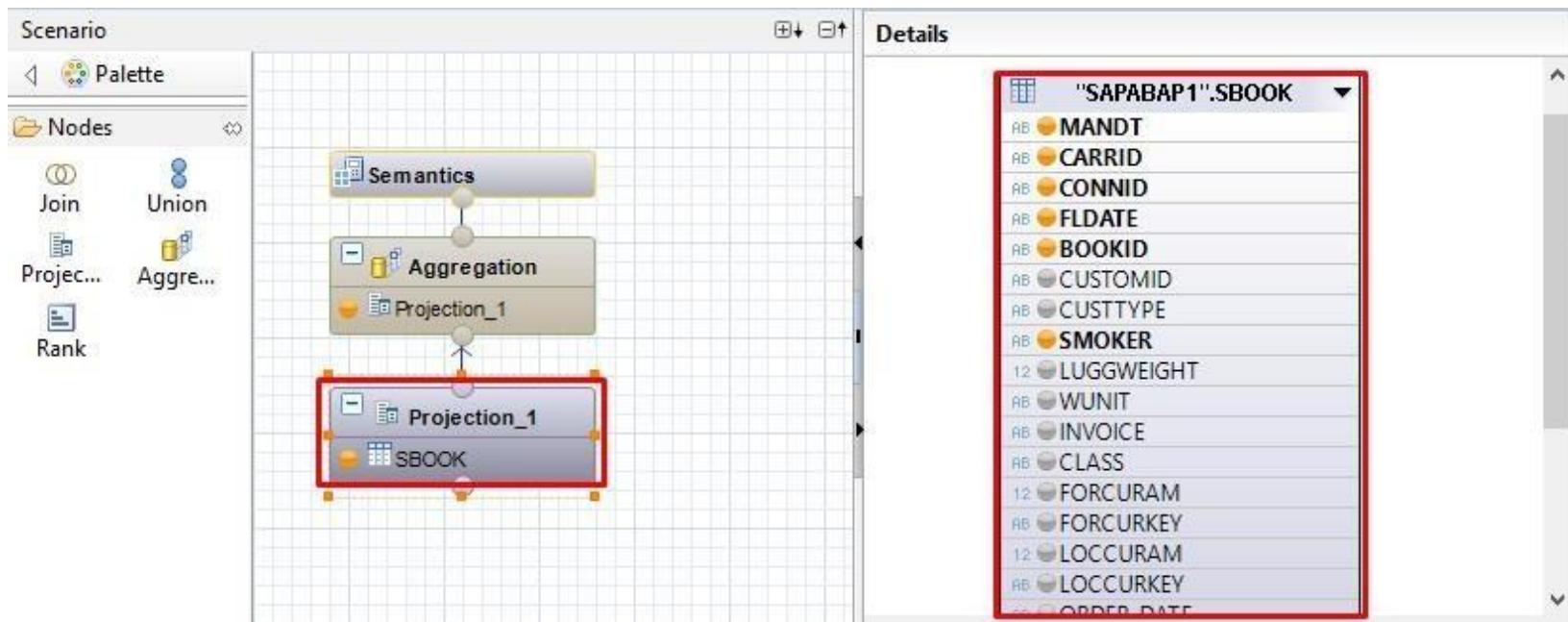
### Create an Information View

Select the required view type and enter the details

|                                                                                                                                                                |                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Name:*                                                                                                                                                         | ZVIEW_CALC2                                                   |
| Label:                                                                                                                                                         | ZVIEW_CALC2                                                   |
| Package:*                                                                                                                                                      | zteste                                                        |
| View Type:                                                                                                                                                     | Calculation View                                              |
| <input type="checkbox"/> Copy From:                                                                                                                            | <input type="text"/> <input type="button" value="Browse..."/> |
| Subtype:                                                                                                                                                       | Standard                                                      |
| Calculation View                                                                                                                                               |                                                               |
| Type:                                                                                                                                                          | Graphical                                                     |
| Data Category: <sup>①</sup>                                                                                                                                    | CUBE                                                          |
| <input type="checkbox"/> With Star Join                                                                                                                        |                                                               |
| <b>Note:</b> If you set the Data Category as Cube, the default node is Aggregation.<br>You can change the default node to Star Join by selecting the checkbox. |                                                               |

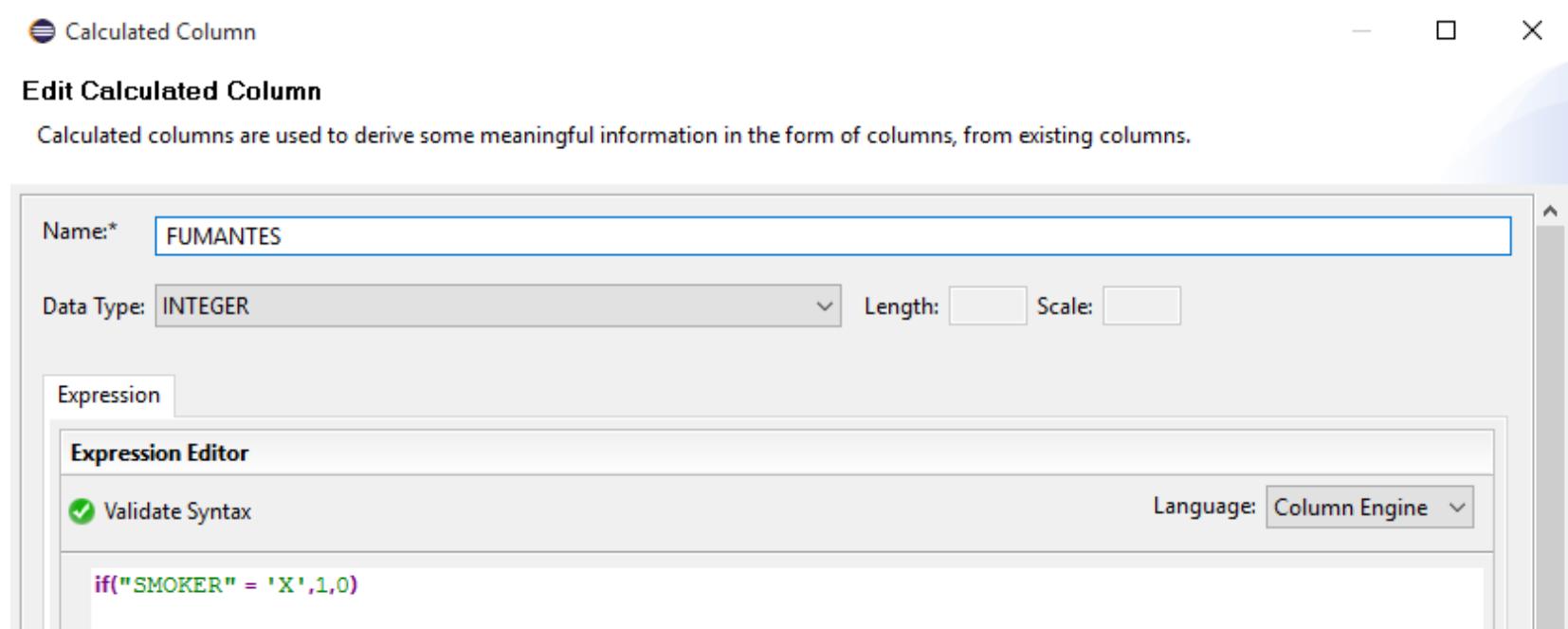
# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:



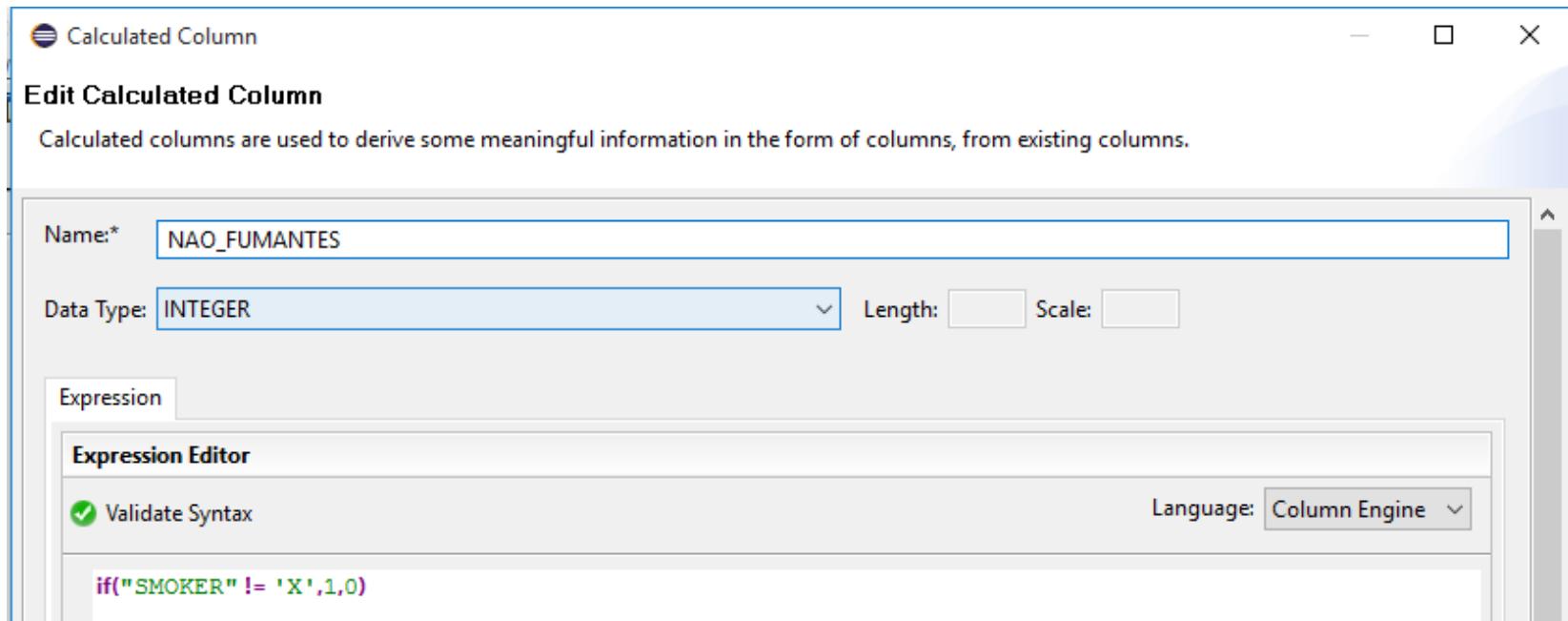
# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:



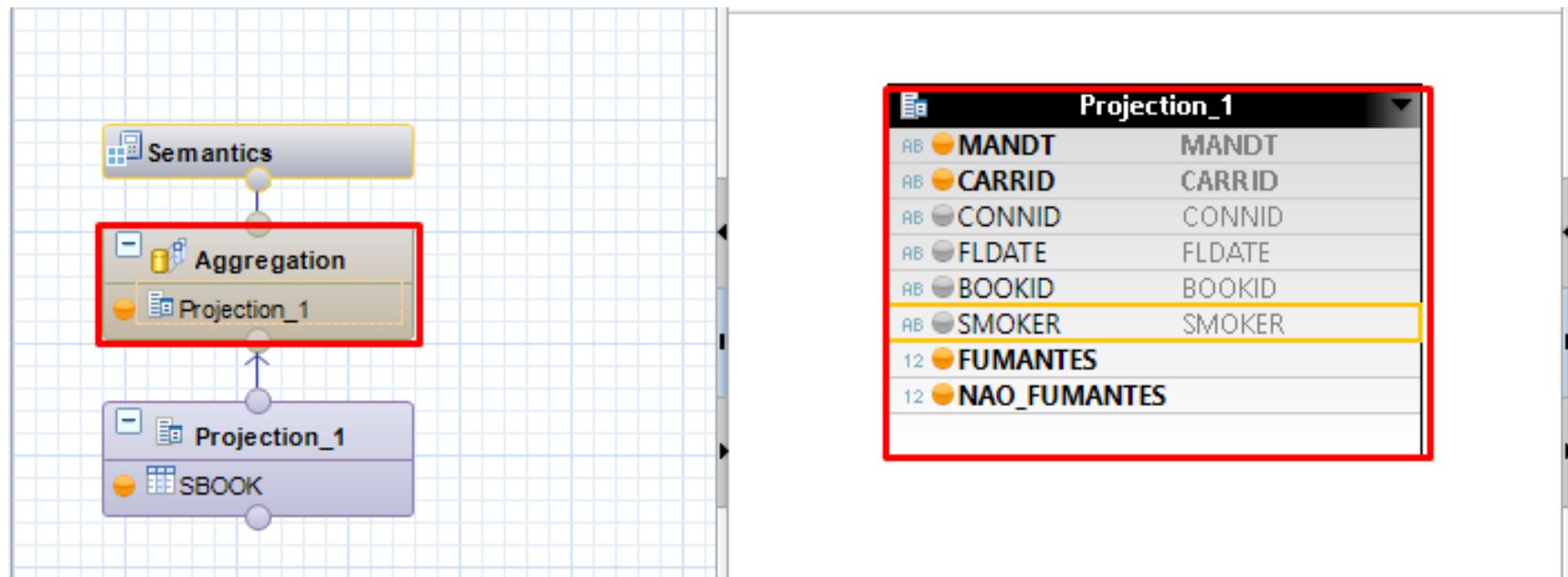
# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:



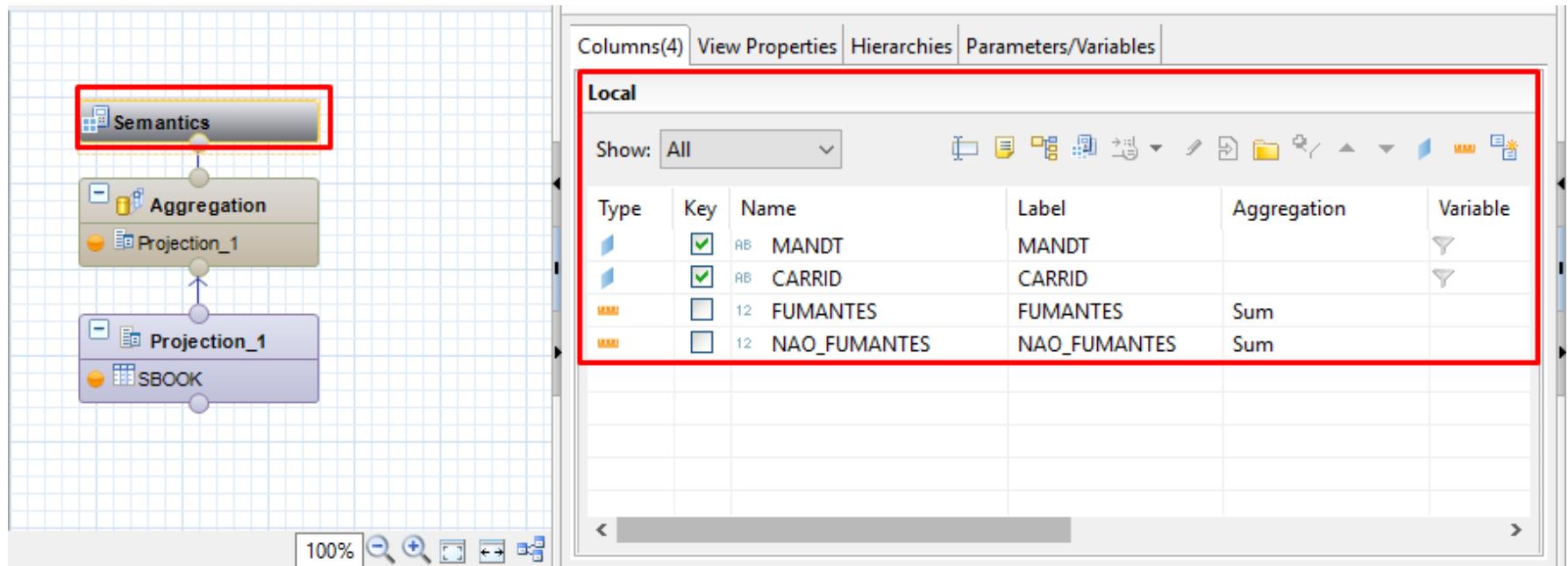
# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:



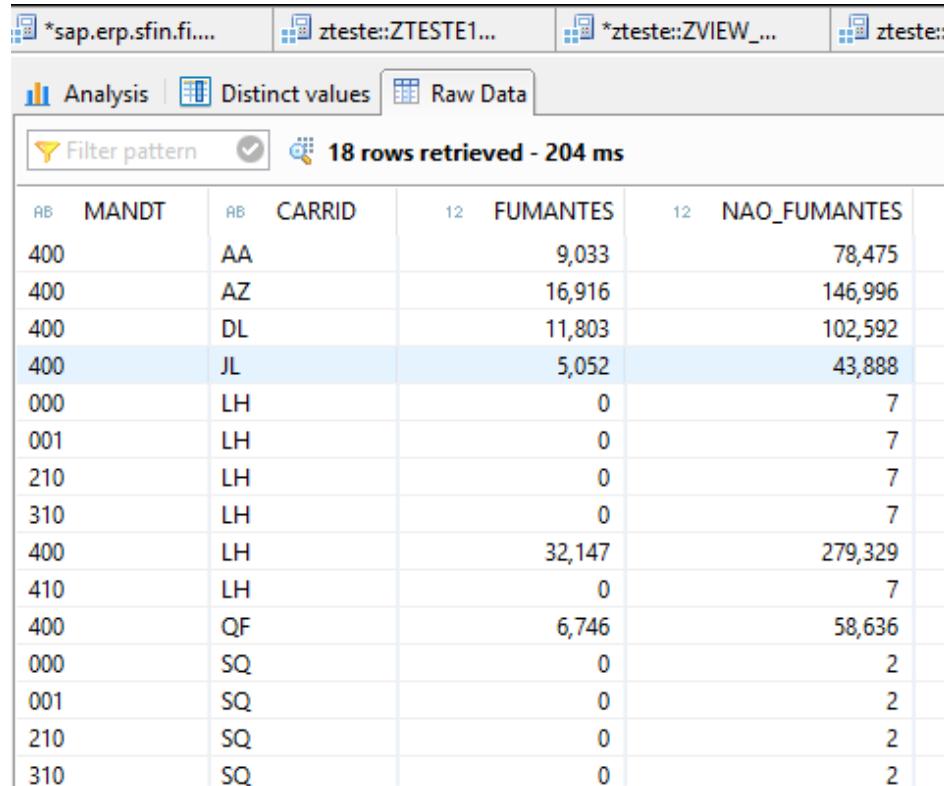
# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:



# Programação em HANA – Visões HANA

- Exemplo 1 – Calculation View:



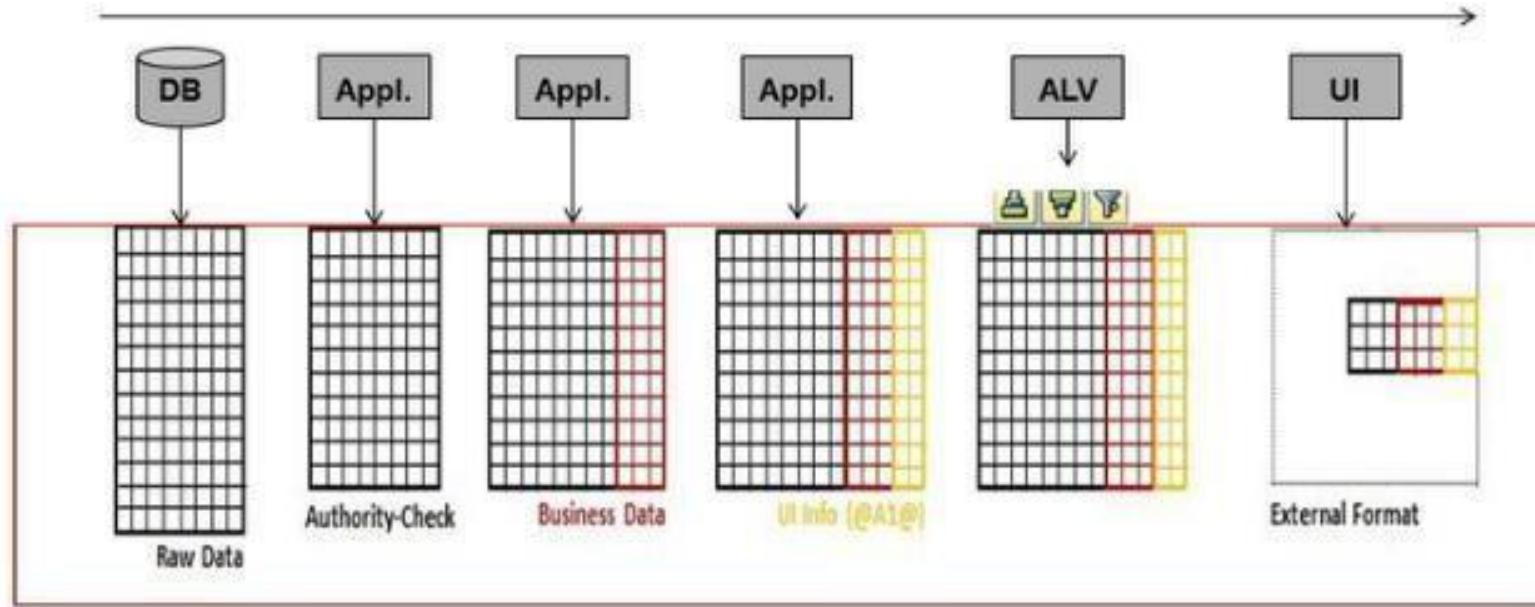
The screenshot shows a SAP HANA studio interface with a Calculation View result set. The view has four tabs at the top: Analysis (selected), Distinct values, and Raw Data. A message bar indicates "18 rows retrieved - 204 ms". The result set displays data from three tables: MANDT, CARRID, and FUMANTES. The columns are AB, MANDT, AB, CARRID, 12, FUMANTES, and 12, NAO\_FUMANTES.

| AB  | MANDT | AB | CARRID | 12 | FUMANTES | 12 | NAO_FUMANTES |
|-----|-------|----|--------|----|----------|----|--------------|
| 400 | AA    |    |        |    | 9,033    |    | 78,475       |
| 400 | AZ    |    |        |    | 16,916   |    | 146,996      |
| 400 | DL    |    |        |    | 11,803   |    | 102,592      |
| 400 | JL    |    |        |    | 5,052    |    | 43,888       |
| 000 | LH    |    |        |    | 0        |    | 7            |
| 001 | LH    |    |        |    | 0        |    | 7            |
| 210 | LH    |    |        |    | 0        |    | 7            |
| 310 | LH    |    |        |    | 0        |    | 7            |
| 400 | LH    |    |        |    | 32,147   |    | 279,329      |
| 410 | LH    |    |        |    | 0        |    | 7            |
| 400 | QF    |    |        |    | 6,746    |    | 58,636       |
| 000 | SQ    |    |        |    | 0        |    | 2            |
| 001 | SQ    |    |        |    | 0        |    | 2            |
| 210 | SQ    |    |        |    | 0        |    | 2            |
| 310 | SQ    |    |        |    | 0        |    | 2            |

ALV HANA

# Programação em HANA – ALV HANA

## ALV Clássico:



- Relevant data is extracted into an internal table after the application performs the necessary authorization checks
- Calculated Business Data columns may be added by the application
- UI information (for example icons or links) is added
- ALV services manipulate the data (sent via an internal table) based on user interaction
- Required data is displayed on the screen

# Programação em HANA – ALV HANA

## Classical ALV Problems

Working with huge amounts of data (potentially millions of records)

Initial Load – All data needs to be loaded from the database into an internal table

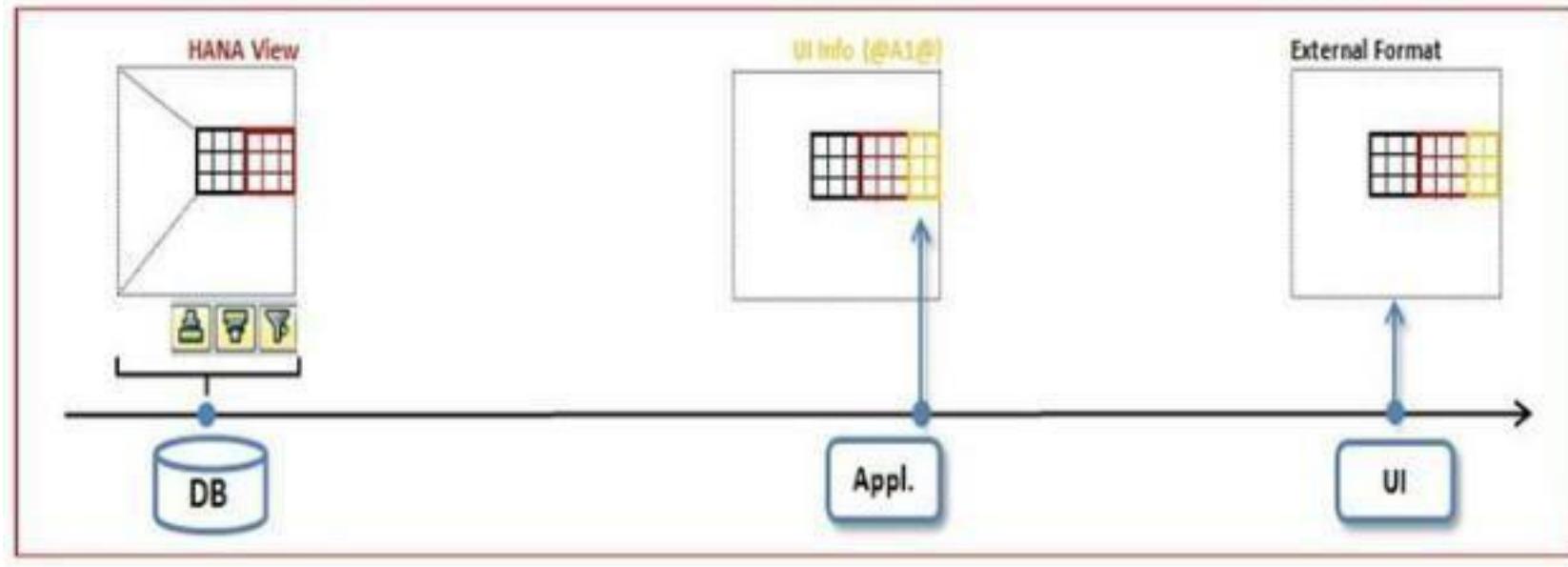
Sorting, filtering, grouping can take a substantial amount of time



- Users may restrict the amount of data to be shown to try to speed up processing
- Waiting too long before results displayed
- Runtime errors can occur due to memory consumption

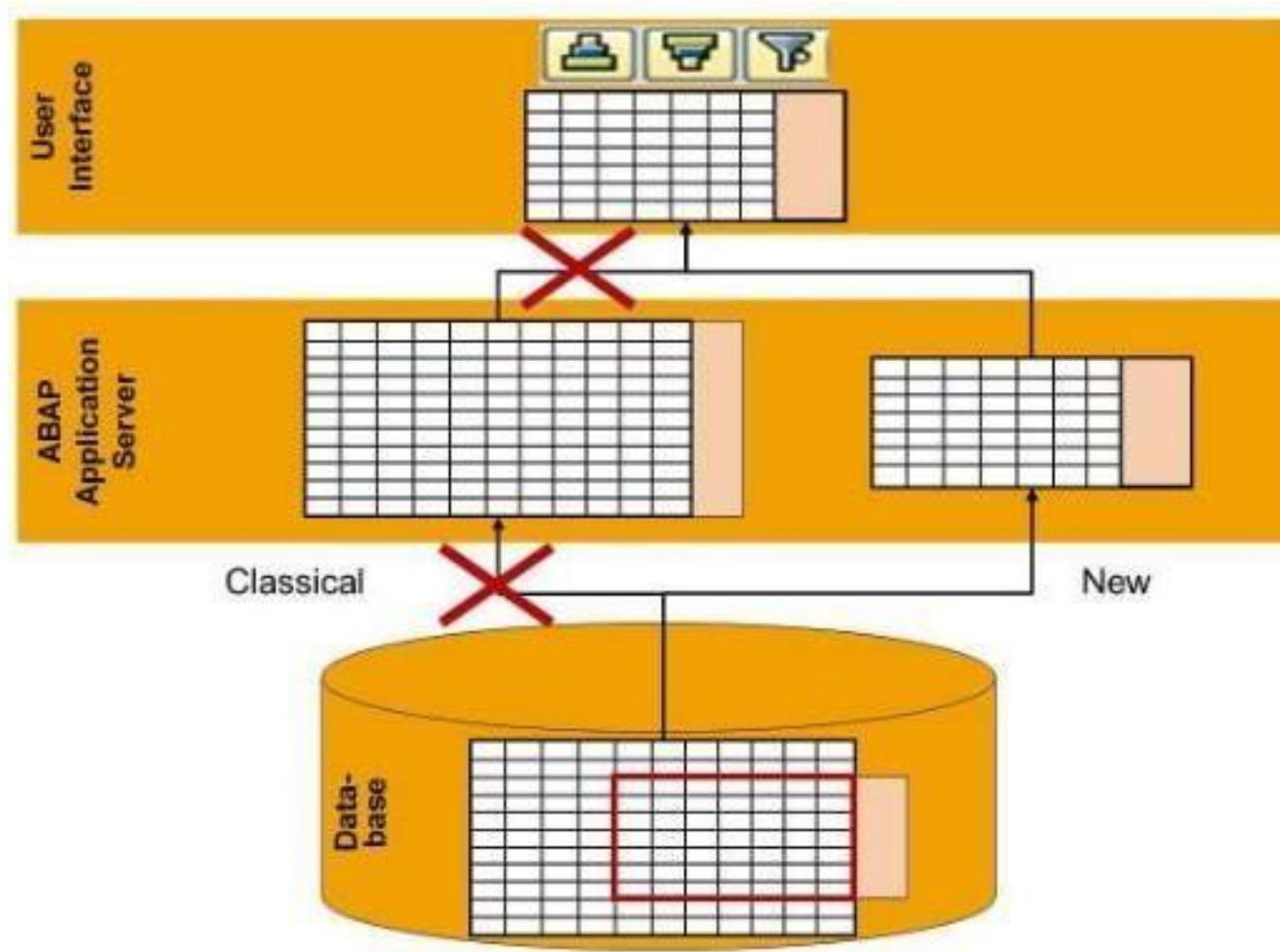
# Programação em HANA – ALV HANA

## ALV HANA:



- Authority checks, table functions, paging, sorting, filtering are now performed by the database via the relevant parameters
- UI information is still handled by the application
- Amount of data is severely restricted by the database before it is displayed on the UI and it is not required to store it in an internal table.

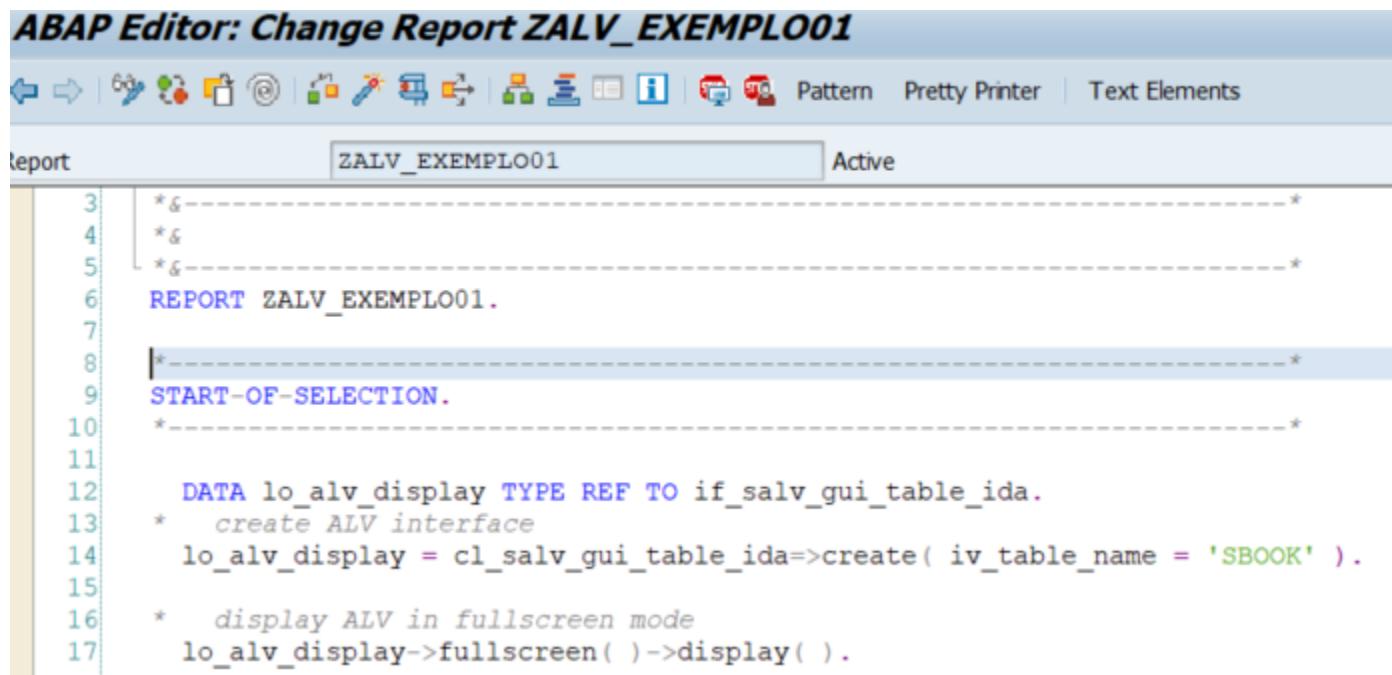
# Programação em HANA – ALV HANA



# Programação em HANA – ALV HANA

- Exemplo 1:

Exibir os dados da tabela SBOOK no ALV HANA através do método create da classe cl\_salv\_gui\_table\_ida.



The screenshot shows the ABAP Editor interface with the title "ABAP Editor: Change Report ZALV\_EXEMPLO01". The report name "ZALV\_EXEMPLO01" is selected in the dropdown. The code area contains the following ABAP code:

```
REPORT ZALV_EXEMPLO01.

START-OF-SELECTION.

DATA lo_alv_display TYPE REF TO if_salv_gui_table_ida.
* create ALV interface
lo_alv_display = cl_salv_gui_table_ida->create(iv_table_name = 'SBOOK').

* display ALV in fullscreen mode
lo_alv_display->fullscreen()->display().
```

# Programação em HANA – ALV HANA

- Exemplo 1:

SAP

The screenshot shows a SAP ALV HANA interface with a title bar 'SAP'. Below it is a toolbar with various icons. The main area is titled 'teste' and contains a table with flight booking data. The table has 18 columns: ID, No., Flight Date, Booki..., Cust. N..., B/P cu..., Smok..., Wt, Unit, Inv.fl..., Cla..., Amou..., Curr., Amou..., Curr., Booking da..., and Sales... . The data consists of 20 rows, each representing a flight booking. The first row is highlighted in yellow. The table includes scroll bars on the right and bottom.

| ID | No. | Flight Date | Booki... | Cust. N... | B/P cu... | Smok... | Wt | Unit | Inv.fl... | Cla... | Amou... | Curr. | Amou... | Curr. | Booking da... | Sales... |
|----|-----|-------------|----------|------------|-----------|---------|----|------|-----------|--------|---------|-------|---------|-------|---------------|----------|
| LH | 402 | 08.12.2014  | 78147    | 2803       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 11.08.2014    | 0        |
| LH | 402 | 08.12.2014  | 78167    | 4160       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 11.08.2014    | 0        |
| LH | 402 | 08.12.2014  | 78307    | 531        | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 24.08.2014    | 0        |
| LH | 402 | 08.12.2014  | 78312    | 3719       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 28.07.2014    | 0        |
| LH | 402 | 08.12.2014  | 78330    | 636        | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 03.09.2014    | 0        |
| LH | 402 | 08.12.2014  | 78371    | 2326       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 11.09.2014    | 0        |
| LH | 402 | 08.12.2014  | 78374    | 2414       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 11.10.2014    | 0        |
| LH | 402 | 08.12.2014  | 78429    | 4634       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 14.07.2014    | 0        |
| LH | 402 | 08.12.2014  | 78443    | 1290       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 03.10.2014    | 0        |
| LH | 402 | 08.12.2014  | 78482    | 1888       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 24.08.2014    | 0        |
| LH | 402 | 15.12.2014  | 78561    | 680        | P         |         | 0  | KG   | X         | Y      | 666,00  | EUR   | 666,00  | EUR   | 05.12.2014    | 0        |
| LH | 402 | 15.12.2014  | 78617    | 2338       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 03.09.2014    | 0        |
| LH | 402 | 15.12.2014  | 78637    | 1996       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 06.09.2014    | 0        |
| LH | 402 | 15.12.2014  | 78724    | 660        | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 10.08.2014    | 0        |
| LH | 402 | 15.12.2014  | 78752    | 3253       | P         |         | 0  | KG   | X         | Y      | 666,00  | EUR   | 666,00  | EUR   | 21.08.2014    | 0        |
| LH | 402 | 15.12.2014  | 78757    | 2651       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 05.10.2014    | 0        |
| LH | 402 | 15.12.2014  | 78796    | 3678       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 23.09.2014    | 0        |
| LH | 402 | 15.12.2014  | 78799    | 957        | P         |         | 0  | KG   | X         | Y      | 666,00  | EUR   | 666,00  | EUR   | 02.10.2014    | 0        |
| LH | 402 | 15.12.2014  | 78806    | 1214       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 24.08.2014    | 0        |
| LH | 402 | 15.12.2014  | 78892    | 4333       | P         |         | 0  | KG   |           | Y      | 666,00  | EUR   | 666,00  | EUR   | 23.10.2014    | 0        |

## Programação em HANA – ALV HANA

- Exemplo 2:

Exibir os dados da tabela SBOOK no ALV HANA através do método create da classe cl\_salv\_gui\_table\_ida passando filtro de data.

```
6 REPORT ZALV_EXEMPLO02.
7
8 TABLES: SBOOK.
9
10 DATA: lo_additional_condition TYPE REF TO if_salv_ida_condition,
11 lo_condition_factory TYPE REF TO if_salv_ida_condition_factory.
12
13 *0... Declara os Parametros
14 *PARAMETERS: TYPE .
15
16 *1... Declara SELECT-OPTIONS
17 SELECT-OPTIONS S_DATE FOR SBOOK-FLDATE.
18
19 START-OF-SELECTION.
20 *2... Cria o objeto de referência ALV with IDA
21 DATA(lo_alv_display) = cl_salv_gui_table_ida->create('SBOOK').
```

# Programação em HANA – ALV HANA

- Exemplo 2:

```
22
23 *3... Constroi o range de tabelas
24 DATA(lo_range_collector) = NEW cl_salv_range_tab_collector().
25
26 lo_range_collector->add_ranges_for_name(iv_name = 'FLDATE'
27 it_ranges = S_DATE[]) .
28
29 lo_range_collector->get_collected_ranges(
30 IMPORTING
31 et_named_ranges = DATA(lt_select_options)) .
32
33 *4... Build conditions
34 | * lo_condition_factory = lo_alv_display->condition_factory() .
35 | * lo_additional_condition = lo_condition_factory->EQUALS(name = ''
36 | value = '') .
37
38 *5... Seleciona as opções e condições
39 lo_alv_display->set_select_options(it_ranges = lt_select_options
40 io_condition = lo_additional_condition) .
41
42 *5... Exibe ALV
43 lo_alv_display->fullscreen()->display().
```

# Programação em HANA – ALV HANA

- Exemplo 2:

**ALV - Exercício 02**

Data da Passagem  to

**ALV - Exercício 02 (#2.331)**

| Carr. | No. | Flight Date | Booki | Cust. N. | B/P cu | Smok | Wt      | Unit | Invoi | Cla | Amount   | Curr. | Amount | Curr. | Booking date | Sales | Agency | Ca | Reserv | Name of the Passenger   | Tit | Pass. D. |
|-------|-----|-------------|-------|----------|--------|------|---------|------|-------|-----|----------|-------|--------|-------|--------------|-------|--------|----|--------|-------------------------|-----|----------|
| AA    | 17  | 19.12.2019  | 1     | 3915     | P      |      | 17,4000 | KG   |       | C   | 922,01   | EUR   | 845,88 | USD   | 15.09.2019   | 0     | 114    |    |        | Matthias Schneider      |     |          |
| AA    | 17  | 19.12.2019  | 2     | 3935     | P      |      | 13,4000 | KG   |       | C   | 845,88   | USD   | 845,88 | USD   | 21.08.2019   | 0     | 222    |    |        | Matthias Madeira        |     |          |
| AA    | 17  | 19.12.2019  | 3     | 552      | P      |      | 9,8000  | KG   |       | C   | 803,58   | USD   | 803,58 | USD   | 08.08.2019   | 29    | 0      |    |        | Siegfried Kirk          |     |          |
| AA    | 17  | 19.12.2019  | 4     | 4036     | P      |      | 23,9000 | KG   |       | C   | 922,01   | EUR   | 845,88 | USD   | 05.09.2019   | 0     | 104    |    |        | Sophie Montero          |     |          |
| AA    | 17  | 19.12.2019  | 5     | 3431     | P      |      | 25      | KG   | X     | C   | 875,90   | EUR   | 803,58 | USD   | 23.10.2019   | 0     | 87     |    |        | Achim Prinz             |     |          |
| AA    | 17  | 19.12.2019  | 6     | 2604     | P      |      | 19,8000 | KG   |       | C   | 543,36   | GBP   | 803,58 | USD   | 07.12.2019   | 0     | 107    |    |        | Kurt Schneider          |     |          |
| AA    | 17  | 19.12.2019  | 7     | 2157     | P      |      | 18,9000 | KG   |       | C   | 845,88   | USD   | 845,88 | USD   | 30.11.2019   | 0     | 111    |    |        | Lothar Buehler          |     |          |
| AA    | 17  | 19.12.2019  | 8     | 2283     | P      |      | 10,9000 | KG   |       | C   | 829,82   | EUR   | 761,30 | USD   | 13.11.2019   | 0     | 117    |    |        | Christoph Gueldenpennig |     |          |
| AA    | 17  | 19.12.2019  | 9     | 2042     | P      |      | 15,3000 | KG   | X     | C   | 543,36   | GBP   | 803,58 | USD   | 02.12.2019   | 0     | 109    |    |        | Guillermo Kramer        |     |          |
| AA    | 17  | 19.12.2019  | 10    | 4255     | P      |      | 0       | KG   | X     | C   | 719,00   | USD   | 719,00 | USD   | 14.09.2019   | 0     | 111    |    |        | Theresa Moyano          |     |          |
| AA    | 17  | 19.12.2019  | 11    | 4255     | P      | X    | 21,8000 | KG   |       | C   | 783,71   | EUR   | 719,00 | USD   | 08.09.2019   | 0     | 294    |    |        | Theresa Moyano          |     |          |
| AA    | 17  | 19.12.2019  | 12    | 3466     | P      |      | 27,1000 | KG   |       | C   | 692,51   | AUD   | 845,88 | USD   | 10.09.2019   | 0     | 102    |    |        | Ludwig Matthaeus        |     |          |
| AA    | 17  | 19.12.2019  | 13    | 1373     | P      |      | 27,5000 | KG   | X     | C   | 783,71   | EUR   | 719,00 | USD   | 20.11.2019   | 0     | 105    |    |        | Annemarie Gueldenpennig |     |          |
| AA    | 17  | 19.12.2019  | 14    | 2144     | P      |      | 12,2000 | KG   | X     | C   | 719,00   | USD   | 719,00 | USD   | 28.07.2019   | 0     | 125    |    |        | Ilya Moyano             |     |          |
| AA    | 17  | 19.12.2019  | 15    | 4414     | P      |      | 10,4000 | KG   | X     | C   | 875,90   | EUR   | 803,58 | USD   | 07.12.2019   | 0     | 61     |    |        | Christoph Ryan          |     |          |
| AA    | 17  | 19.12.2019  | 16    | 3355     | P      |      | 0       | KG   | X     | C   | 7.068,29 | ZAR   | 761,30 | USD   | 08.09.2019   | 0     | 254    |    |        | Ludwig Cesari           |     |          |
| AA    | 17  | 19.12.2019  | 17    | 3304     | P      |      | 14,4000 | KG   |       | C   | 783,71   | EUR   | 719,00 | USD   | 10.11.2019   | 0     | 224    |    |        | Theresa Pan             |     |          |
| AA    | 17  | 19.12.2019  | 18    | 1767     | P      |      | 13,5000 | KG   |       | C   | 783,71   | EUR   | 719,00 | USD   | 13.09.2019   | 0     | 121    |    |        | James Moyano            |     |          |
| AA    | 17  | 19.12.2019  | 19    | 1383     | P      |      | 0       | KG   | X     | C   | 803,58   | USD   | 803,58 | USD   | 08.11.2019   | 0     | 319    |    |        | Carmel Kramer           |     |          |
| AA    | 17  | 19.12.2019  | 20    | 724      | P      |      | 0       | KG   | X     | C   | 922,01   | EUR   | 845,88 | USD   | 15.10.2019   | 0     | 117    |    |        | Anneliese Simonen       |     |          |
| AA    | 17  | 19.12.2019  | 21    | 4349     | P      | X    | 0       | KG   | X     | C   | 1.525,75 | SGD   | 845,88 | USD   | 07.10.2019   | 0     | 103    |    |        | James Montero           |     |          |
| AA    | 17  | 19.12.2019  | 22    | 1027     | P      |      | 0       | KG   | X     | C   | 829,82   | EUR   | 761,30 | USD   | 29.11.2019   | 0     | 93     |    |        | Horst Leisert           |     |          |
| AA    | 17  | 19.12.2019  | 23    | 3793     | P      |      | 21,6000 | KG   |       | C   | 761,30   | USD   | 761,30 | USD   | 26.09.2019   | 0     | 301    |    |        | Andrej Pratt            |     |          |
| AA    | 17  | 19.12.2019  | 24    | 2641     | P      |      | 0       | KG   | X     | C   | 514,78   | GBP   | 761,30 | USD   | 22.10.2019   | 0     | 295    |    |        | Jean Marshall           |     |          |
| AA    | 17  | 19.12.2019  | 25    | 262      | B      | X    | 10,7000 | KG   |       | C   | 812,04   | USD   | 812,04 | USD   | 27.10.2019   | 0     | 301    |    |        |                         |     |          |
| AA    | 17  | 19.12.2019  | 26    | 2829     | P      |      | 0       | KG   |       | C   | 829,82   | EUR   | 761,30 | USD   | 08.12.2019   | 0     | 105    |    |        | Roland Pan              |     |          |
| AA    | 17  | 19.12.2019  | 27    | 937      | P      |      | 0       | KG   |       | C   | 922,01   | EUR   | 845,88 | USD   | 31.10.2019   | 0     | 61     |    |        | Christoph Sommer        |     |          |

# Programação em HANA – ALV HANA

- Diferenças entre ALV Clássico e ALV HANA:

| Feature                | Classic ALV               | ALV for HANA                                                    | Limitations for ALV on HANA                                           |
|------------------------|---------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------|
| Multi-db support       | Yes                       | Not at present                                                  |                                                                       |
| Sorting                | Yes                       | Yes                                                             | Without currency/unit consideration (must first sort on currency/UOM) |
| Selection Screen       | Dealt with by application | Yes – select-options and parameters to be passed to the ALV API |                                                                       |
| Filtering              | Yes                       | Yes                                                             |                                                                       |
| Aggregation            | Yes                       | Yes                                                             | No unit/currency split if aggregating amounts/currency values         |
| Grouping               | No                        | Yes                                                             |                                                                       |
| Find                   | Yes                       | No                                                              |                                                                       |
| Text and Fuzzy Search  | No                        | Yes                                                             |                                                                       |
| Support for HANA Views | Dealt with by application | Yes                                                             |                                                                       |
| Authorization Checks   | Dealt with by application | Yes                                                             |                                                                       |
| Export to Excel        | Yes                       | Yes                                                             | Restricted to 10,000 records                                          |
| Printing               | Yes                       | Yes                                                             | Restricted to 10,000 records                                          |

## Programação em HANA – ALV HANA

- Diferenças entre ALV Clássico e ALV HANA:

| Feature                          | Classic ALV | ALV for HANA | Limitations for ALV on HANA |
|----------------------------------|-------------|--------------|-----------------------------|
| Personalization                  | Yes         | Yes          |                             |
| Cells with links, buttons, icons | Yes         | No           |                             |
| Graphics                         | Yes         | No           |                             |
| Editable                         | Yes         | No           |                             |
| Fixed Columns                    | Yes         | No           |                             |
| Double-click                     | Yes         | No           |                             |

## Programação em HANA – ALV HANA

- Exercício 1:
  - Crie um programa com dois campos de tela de seleção:
    - ID – SCUSTOM-ID
    - NAME – SCUSTOM-NAME
  - Crie um relatório ALV usando ALVHANA e usando ALV Clássico, selecionando dados da tabela SCUSTOM

# Programação em HANA – ALV HANA

## ALV HANA

```
1 *-----*
2 *& Report ZALV_EXEC01HANA
3 *-----*
4 *&
5 *-----*
6 REPORT zalv_exec01hana.
7 |
8 TABLES: scustom, sbook.
9
10 * ALV COM SELEÇÕES
11
12 DATA: lo_additional_condition TYPE REF TO if_salv_ida_condition,
13 lo_condition_factory TYPE REF TO if_salv_ida_condition_factory.
14
15 *0... Declara os Parametros
16 *PARAMETERS: TYPE .
17
18 *1... Declara SELECT-OPTIONS
19 SELECT-OPTIONS s_is FOR scustom-id.
20 SELECT-OPTIONS s_name FOR scustom-name.
21
22 START-OF-SELECTION.
23 *2... Cria o objeto de referência ALV with IDA
24 DATA(lo_alv_display) = cl_salv_gui_table_ida->create('SCUSTOM').
25
26
27 " set available fields
28 lo_alv_display->field_catalog()->set_available_fields(its_field_names = VALUE if_salv_gui_types_ida->yts_field_name((CONV fieldname('ID'))
29 (CONV fieldname('NAME')))) .
30 " set display order of fields
31 lo_alv_display->default_layout()->set_visible_fields(it_visible_fields = VALUE if_salv_gui_types_ida->yt_field_name((CONV fieldname('ID'))
32 (CONV fieldname('NAME')))) .
```

# Programação em HANA – ALV HANA

## ALV HANA

```
33
34 ④ *
35 ④ **3... Constroi o range de tabelas
36 DATA(lo_range_collector) = NEW cl_salv_range_tab_collector().
37 *
38 lo_range_collector->add_ranges_for_name(iv_name = 'ID'
39 it_ranges = s_is[]) .
40 *
41 lo_range_collector->add_ranges_for_name(iv_name = 'NAME'
42 it_ranges = s_name[]) .
43
44 lo_range_collector->get_collected_ranges(
45 IMPORTING
46 et_named_ranges = DATA(lt_select_options)) .
47 *
48 ④ **4... Build conditions
49 lo_condition_factory = lo_alv_display->condition_factory() .
50 ④ * lo_additional_condition = lo_condition_factory->equals(name = 'ID'
51 ④ * value = s_is) .
52
53 ④ * lo_additional_condition = lo_condition_factory->equals(name = 'NAME'
54 ④ * value = s_name) .
55 *
56 ④ *5... Seleciona as opções e condições
57 lo_alv_display->set_select_options(it_ranges = lt_select_options
58 io_condition = lo_additional_condition) .
59
60 ④ *5... Exibe ALV
61 lo_alv_display->fullscreen()->display() .
```

# Programação em HANA – ALV HANA

**ALV HANA**

ID  to  

NAME  to  

**ALV HANA (#4.637)**

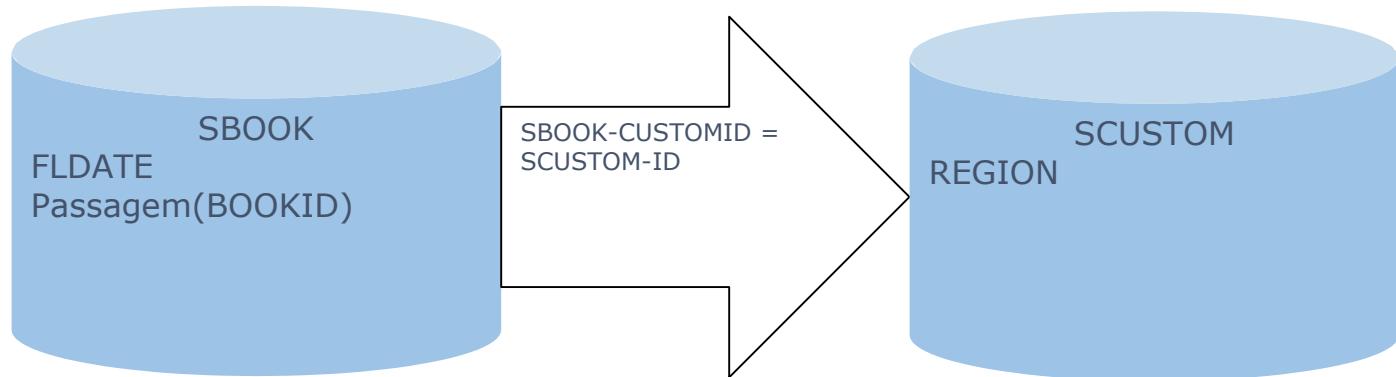
| Cust. N... | Customer name  |
|------------|----------------|
| 1          | SAP AG         |
| 2          | Andreas Klotz  |
| 3          | Hans Bullinger |
| 4          | Staerck        |
| 5          | Martin         |
| 6          | Starr          |
| 7          | King           |
| 8          | Moore          |

# Programação em HANA – ALV HANA

## ALV CLÁSSICO

```
1 *-----*
2 *& Report ZALV_EXEC01
3 *-----*
4 *&
5 *&
6 REPORT ZALV_EXEC01.
7
8 TABLES scustom.
9
10 "a. Selection Screen
11 DATA ls_scustom TYPE scustom.
12 SELECT-OPTIONS s_id FOR scustom-id.
13 SELECT-OPTIONS s_name FOR scustom-name.
14
15 START-OF-SELECTION.
16 "b. Data Selection [ALL ROWS - ALL COLUMNS] from database into [ITAB]
17 " on the ABAP server using values of Selection Screen
18 DATA lt_scustom TYPE STANDARD TABLE OF scustom.
19 SELECT * FROM scustom INTO CORRESPONDING FIELDS OF TABLE lt_scustom
20 WHERE id IN s_id
21 AND name IN s_name.
22
23 "c. Show selected data of [ITAB] in ALV: here object model in Fullscreen Mode
24 CALL METHOD cl_salv_table=>factory(IMPORTING r_salv_table = DATA(lo_alv_display)
25 CHANGING t_table = lt_scustom).
26 lo_alv_display->display().
```

- Exercicio 2 -Desafio:
  - Crie relatório ALVHANA, onde o usuário possa pesquisar por várias REGIÃO(SCUSTOM-REGION) em range de data(SBOOK-FLDATE) utilizando CDS VIEW e será informado no relatório ALV por Região selecionada o numero total de passagens compradas



# Programação em HANA – ALV HANA

```
D [S4H] ZUSER00_CDSALVEXEC2 ✘
1 @AbapCatalog.sqlViewName: 'ZALV_EXEC2'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'ALV EXEC 2'
6 define view ZUSER00_CDSALVEXEC2 as
7 select
8 --sbook.bookid,
9 scustom.id,
10 sbook.fldate,
11 scustom.country,
12 @DefaultAggregation: #COUNT
13 count(*) as total
14
15 from (sbook as sbook inner join scustom as scustom on
16 sbook.customid = scustom.id)
17 group by scustom.id,
18 sbook.fldate,
19 scustom.country
20
```

# Programação em HANA – ALV HANA

```
1 *-----*
2 |*& Report ZALV_EXERCICIO02
3 |*-----*
4 |*&
5 |*-----*
6 REPORT ZALV_EXERCICIO02.
7
8 TABLES: SCUSTOM, SBOOK.
9
10 * ALV COM SELEÇÕES
11
12 DATA: lo_additional_condition TYPE REF TO if_salv_ida_condition,
13 lo_condition_factory TYPE REF TO if_salv_ida_condition_factory.
14
15 *0... Declara os Parametros
16 L *PARAMETERS: TYPE .
17
18 *1... Declara SELECT-OPTIONS
19 SELECT-OPTIONS S_CTRY FOR SCUSTOM-id.
20 SELECT-OPTIONS S_DATE FOR SBOOK-FLDATE.
21
22 START-OF-SELECTION.
23 *2... Cria o objeto de referência ALV with IDA
24 DATA(lo_alv_display) = cl_salv_gui_table_ida->CREATE_FOR_CDS_VIEW('ZUSER00_CDSALVEXEC2').

25
26
27 * " set available fields
28 *lo_alv_display->field_catalog()->set_available_fields(its_field_names = VALUE if_salv_gui_types_ida->yts_field_name((CONV fieldname('ID'))
29 * (CONV fieldname('NAME')))) .
30 * " set display order of fields
31 *lo_alv_display->default_layout()->set_visible_fields(it_visible_fields = VALUE if_salv_gui_types_ida->yt_field_name((CONV fieldname('ID'))
32 * (CONV fieldname('NAME')))) .
33 .
```

# Programação em HANA – ALV HANA

```
34 */
35 /* **3... Constroi o range de tabelas
36 DATA(lo_range_collector) = NEW cl_salv_range_tab_collector().
37
38 lo_range_collector->add_ranges_for_name(iv_name = 'ID'
39 it_ranges = S_CTRY[]).
40
41 lo_range_collector->add_ranges_for_name(iv_name = 'FLDATE'
42 it_ranges = S_DATE[]).
43
44 lo_range_collector->get_collected_ranges(
45 IMPORTING
46 et_named_ranges = DATA(lt_select_options)).
47 */
48 /* **4... Build conditions
49 ** lo_condition_factory = lo_alv_display->condition_factory().
50 ** lo_additional_condition = lo_condition_factory->EQUALS(name = ''
51 ** value = '').
52 *
53 /* **5... Seleciona as opções e condições
54 lo_alv_display->set_select_options(it_ranges = lt_select_options
55 io_condition = lo_additional_condition).
56
57 /* **6... Exibe ALV
58 lo_alv_display->fullscreen()->display().
```

# Programação em HANA – ALV HANA

**ALV - EXercício 02**

Região

Data da Passagem   to   
to

| Cust. No. | Flight Date | Ct... |
|-----------|-------------|-------|
| 00002948  | 19.12.2019  | ES 1  |
| 00002947  | 19.12.2019  | US 1  |
| 00002942  | 19.12.2019  | DE 1  |
| 00002936  | 19.12.2019  | DE 1  |
| 00002932  | 19.12.2019  | DE 1  |
| 00002930  | 19.12.2019  | DE 1  |
| 00002927  | 19.12.2019  | CH 1  |
| 00002925  | 19.12.2019  | US 1  |
| 00002923  | 19.12.2019  | DE 1  |
| 00002917  | 19.12.2019  | DE 1  |
| 00002910  | 19.12.2019  | DE 1  |
| 00002906  | 19.12.2019  | DE 1  |
| 00002905  | 19.12.2019  | FR 1  |
| 00002903  | 19.12.2019  | FR 1  |
| 00002902  | 19.12.2019  | US 1  |
| 00002898  | 19.12.2019  | DE 1  |
| 00002896  | 19.12.2019  | US 1  |
| 00002895  | 19.12.2019  | DE 1  |

HANA XS

## SAP HANA XS - Extended Application Services

- O SAP HANA estende bastante a função tradicional de servidor de banco de dados.
- O SAP HANA funciona como uma plataforma abrangente para o desenvolvimento e a execução de aplicativos nativos com uso intensivo de dados que são executados com eficiência no SAP HANA, aproveitando a arquitetura em memória e os recursos de execução paralela.

# SAP HANA XS - Extended Application Services

## Front-end Technologies

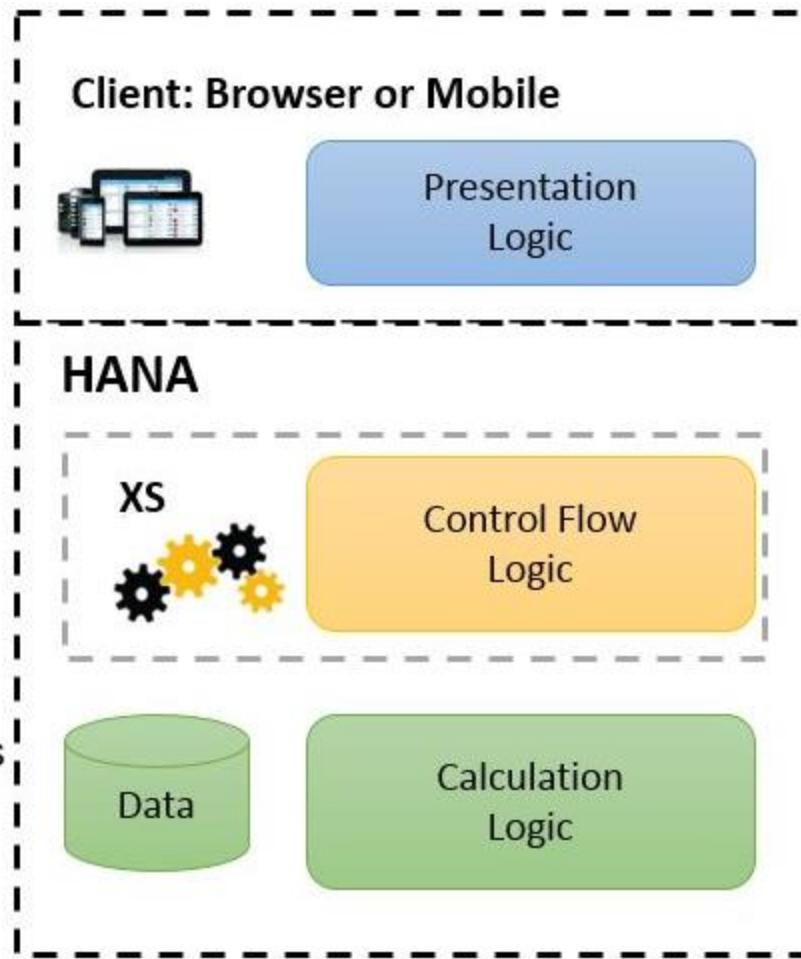
- http/https
- HTML5/SAPUI5
- Client-side JavaScript

## Control Flow Technologies

- oData
- Server-side JavaScript

## Data Processing Technologies

- SQL / SQLScript
- Data Modeling



# SAP HANA XS - Extended Application Services

- Ao reestruturar seu aplicativo dessa maneira, você não apenas obtém o aumento do desempenho devido à integração com a fonte de dados, mas também pode eliminar efetivamente a sobrecarga da camada intermediária entre a interface do usuário (a exibição) e o data- lógica de controle intensivo, como mostrado na figura acima.

# SAP HANA XS - Extended Application Services

- Tecnologias de processamento de dados:
  - Para aproveitar todo o potencial do HANA, devemos inserir a lógica dos dados no banco de dados possível. Toda a lógica intensiva de dados deve ser colocada nas visualizações SQL, SQLScript Procedures e SAP HANA.
  - Faz sentido colocar toda a sua lógica intensiva de dados em SQL, SQLScript Procedures e SAP HANA Views, pois essas técnicas alavancam as otimizações de tabela colunar em memória do SAP HANA, bem como o processamento paralelo em massa.

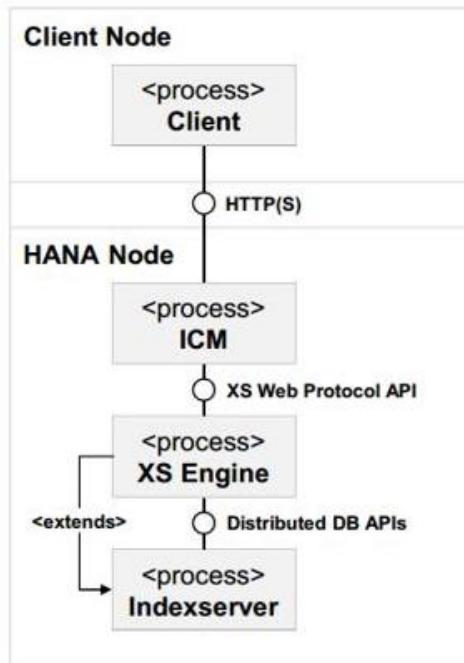
# SAP HANA XS - Extended Application Services

- Control Flow Technologies:

- Depois que as lógicas de dados são implementadas nas visualizações / procedimentos do HANA, precisamos de um servidor de aplicativos muito leve para lidar com a lógica do fluxo de controle. Esse objetivo é resolvido pelo mecanismo HANA XS.
- O HANA XS usa tecnologias como oData, XSJS etc. para expor o modelo de dados do banco de dados, com suas tabelas, visualizações e procedimentos do banco de dados, aos clientes.
- Nenhum dado é armazenado no próprio servidor SAP HANA XS. Para ler tabelas ou visualizações, modificar dados ou executar procedimentos e cálculos de banco de dados SQLScript, ele se conecta ao servidor de indexação (ou servidores, no caso de um sistema distribuído).

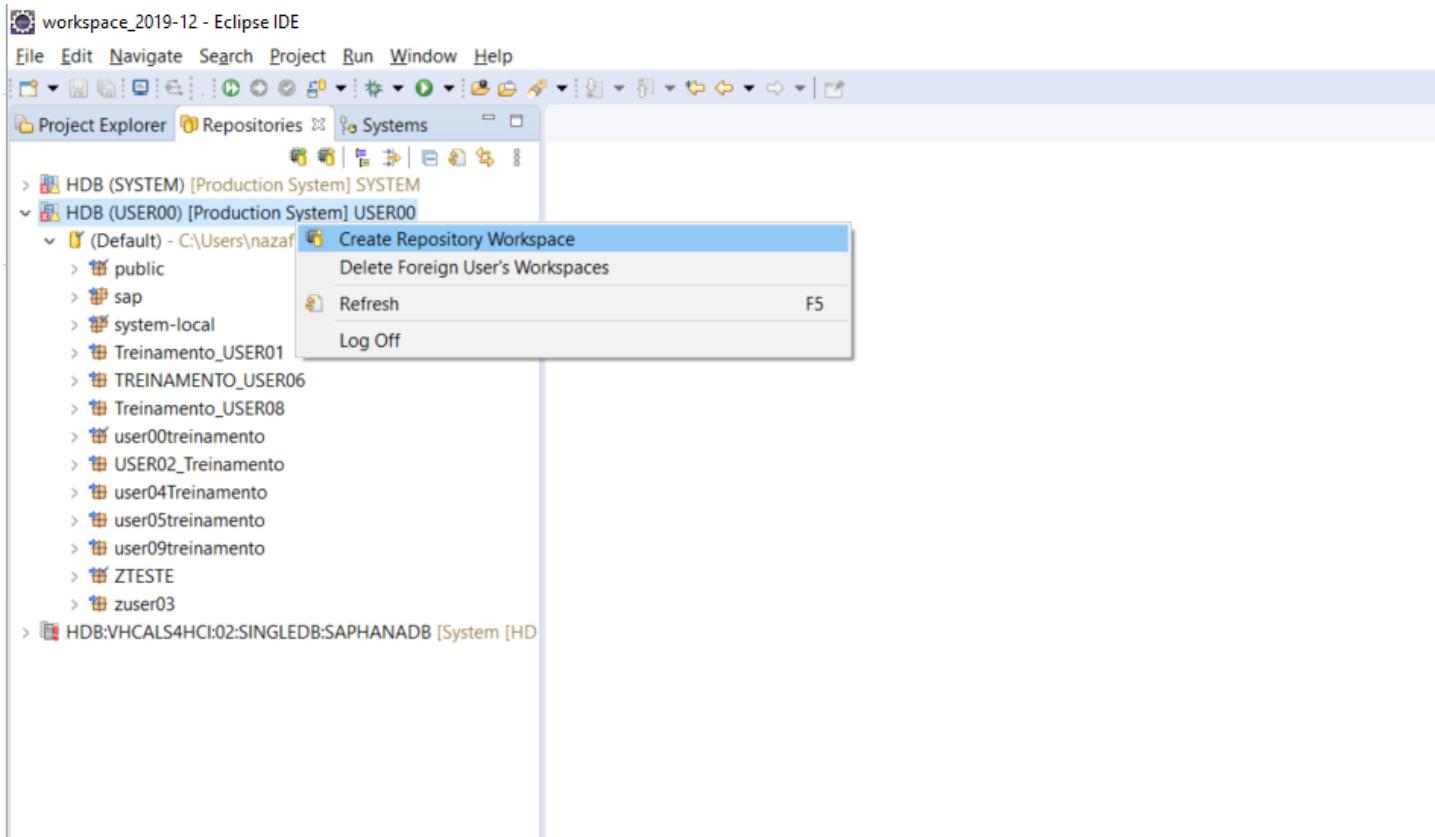
# SAP HANA XS - Extended Application Services

- Tecnologias de front-end:
  - Finalmente, HTML 5 ou SAPUI5 ou qualquer idioma da interface do usuário pode ser usado para implementar a interface do usuário. A SAP recomenda usar HTML5 ou SAPUI5 onde a lógica completa da interface do usuário é executada no lado do cliente.



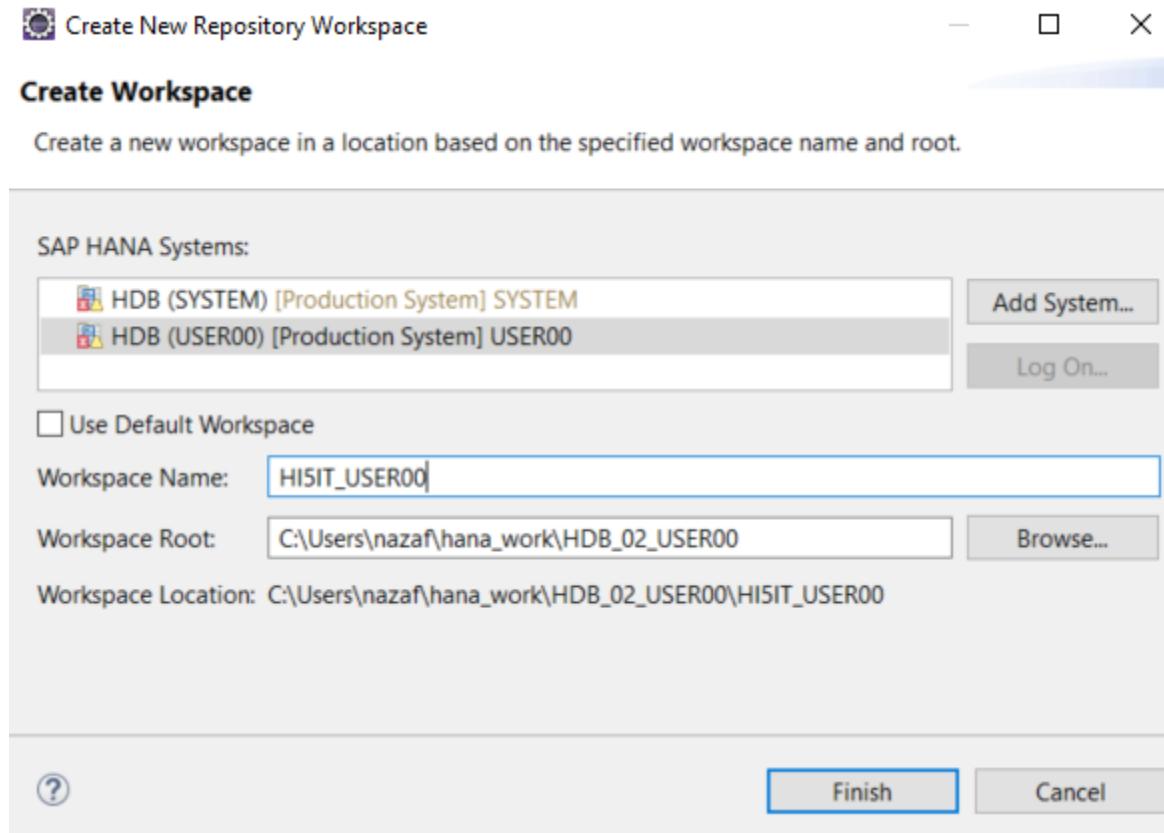
# SAP HANA XS - Extended Application Services

- Criar um repositório.



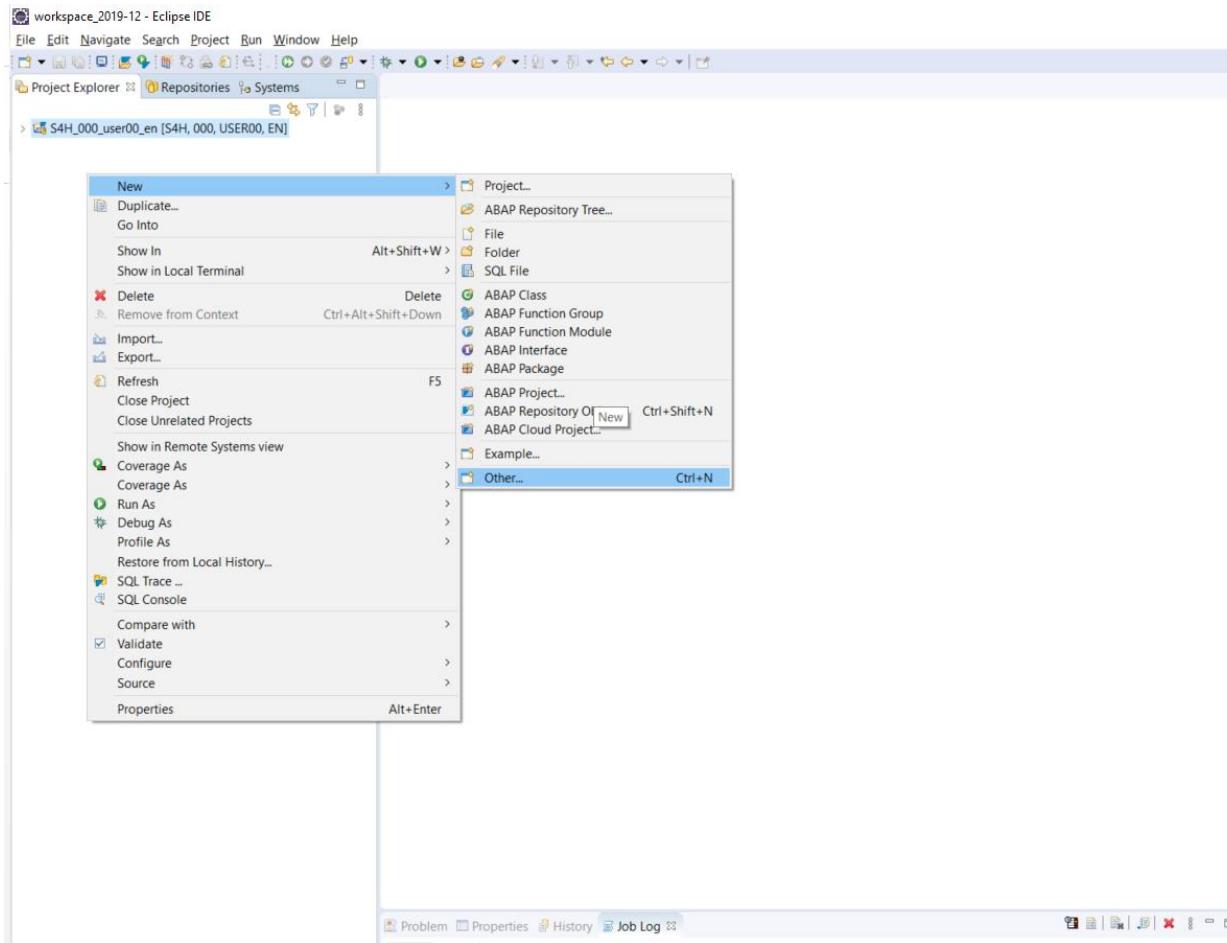
# SAP HANA XS - Extended Application Services

- Criar um repositório.



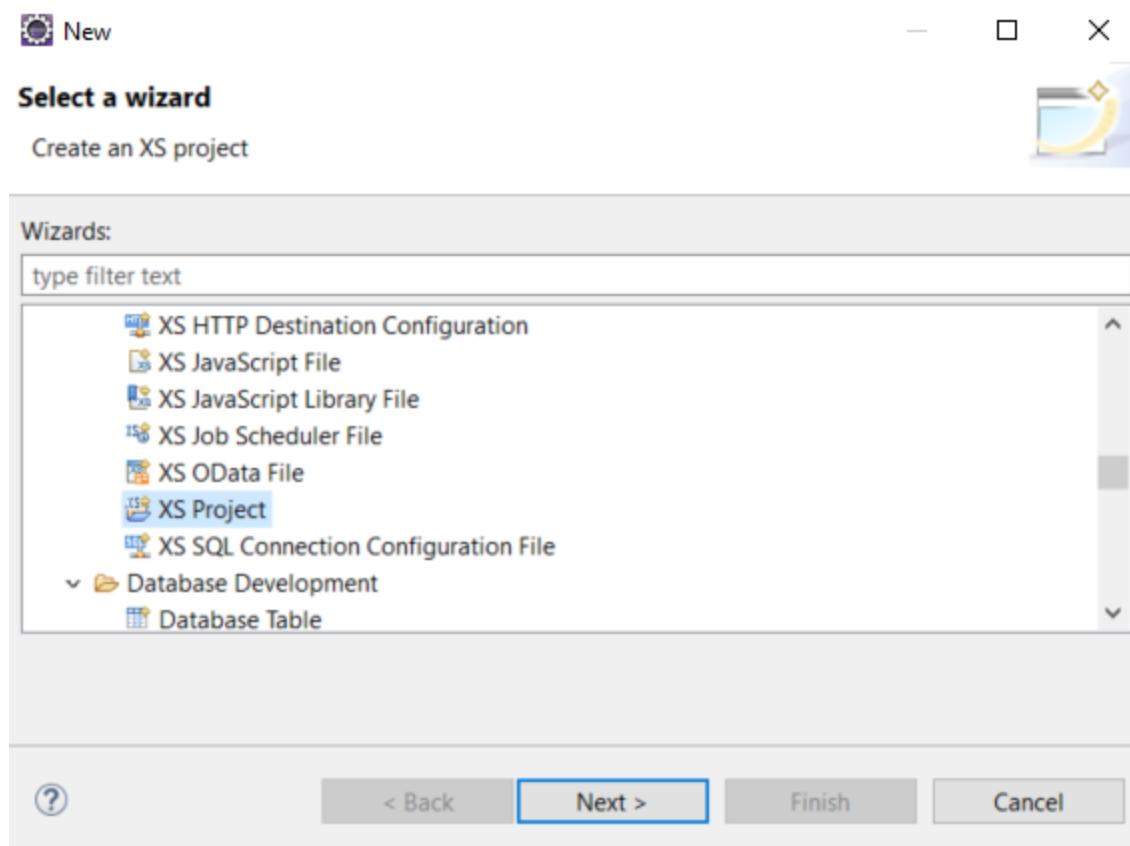
# SAP HANA XS - Extended Application Services

- CRIAR PROJETO XS



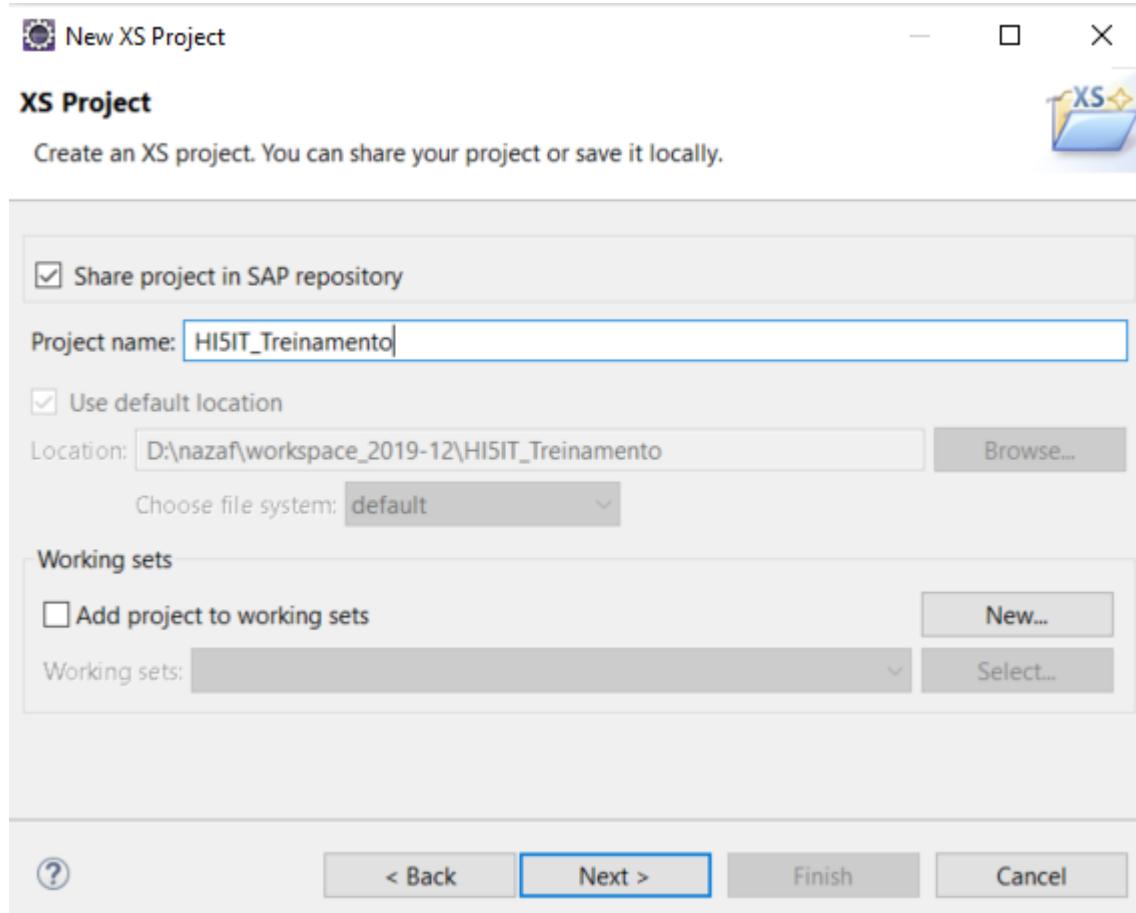
# SAP HANA XS - Extended Application Services

- CRIAR PROJETO XS



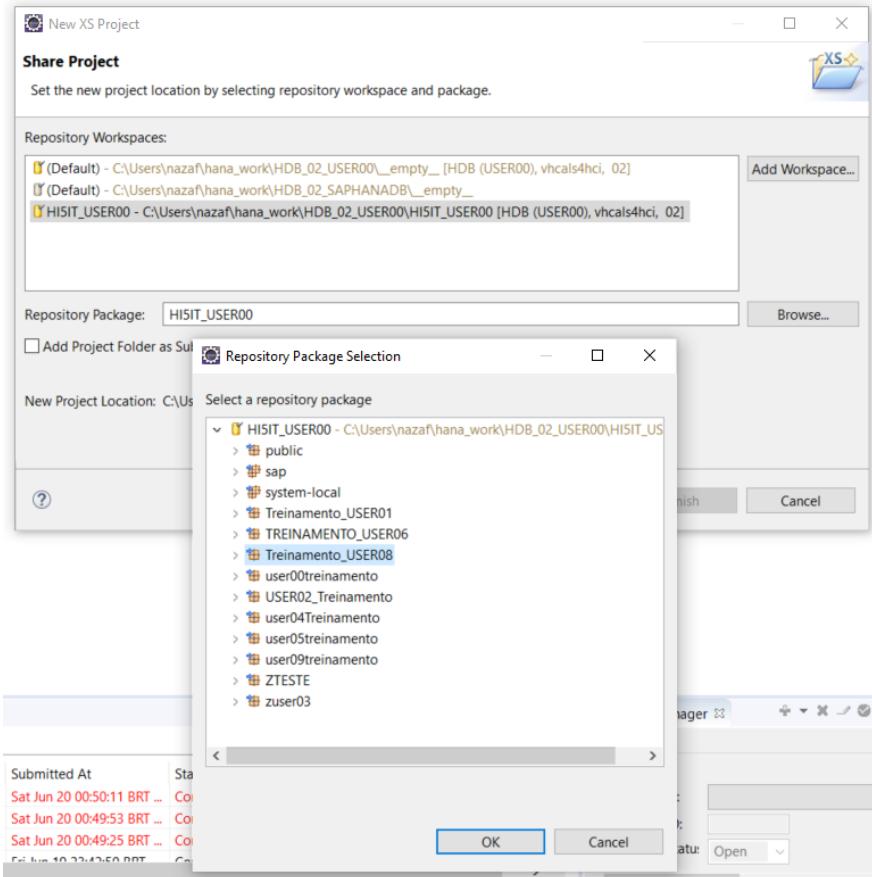
# SAP HANA XS - Extended Application Services

- CRIAR PROJETO XS



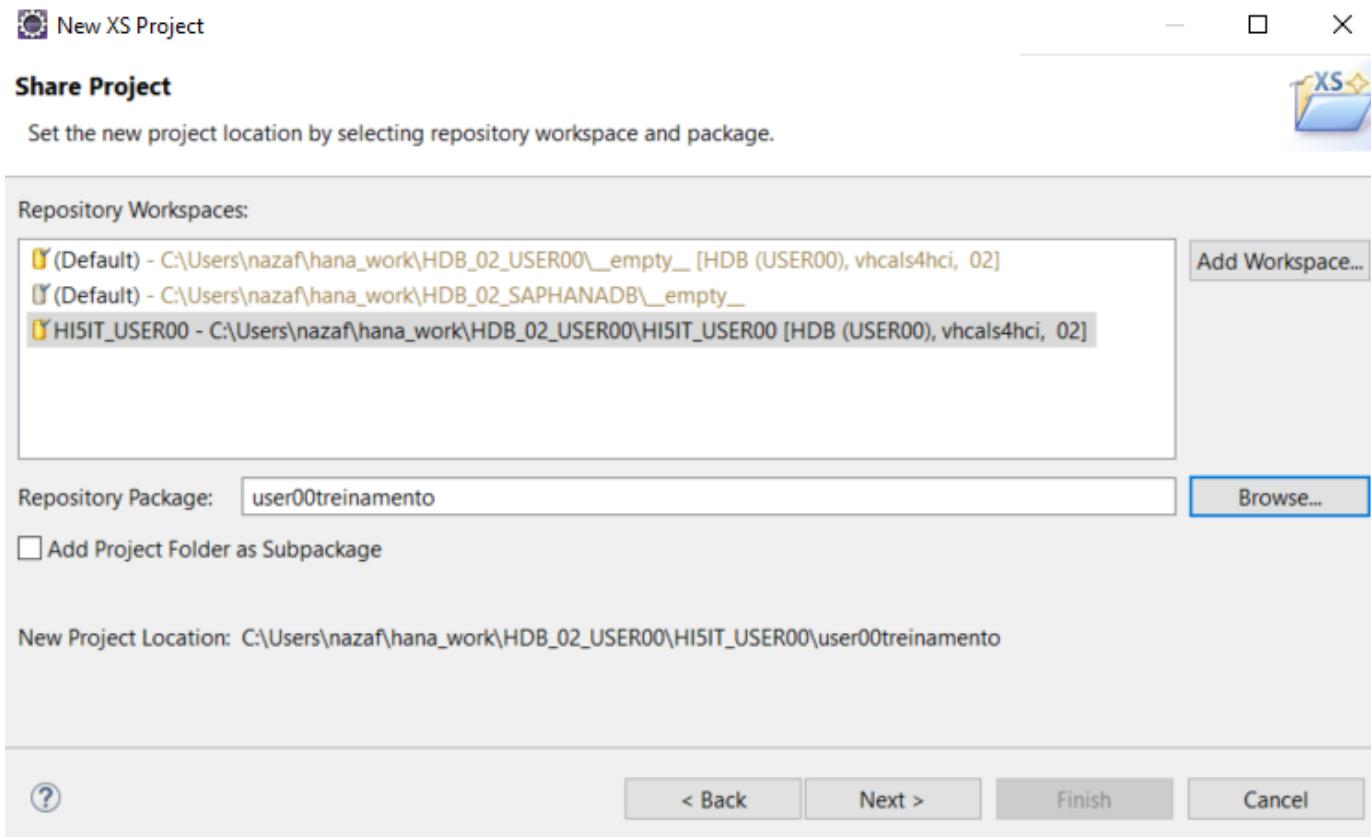
# SAP HANA XS - Extended Application Services

- CRIAR PROJETO XS – Definir pacote



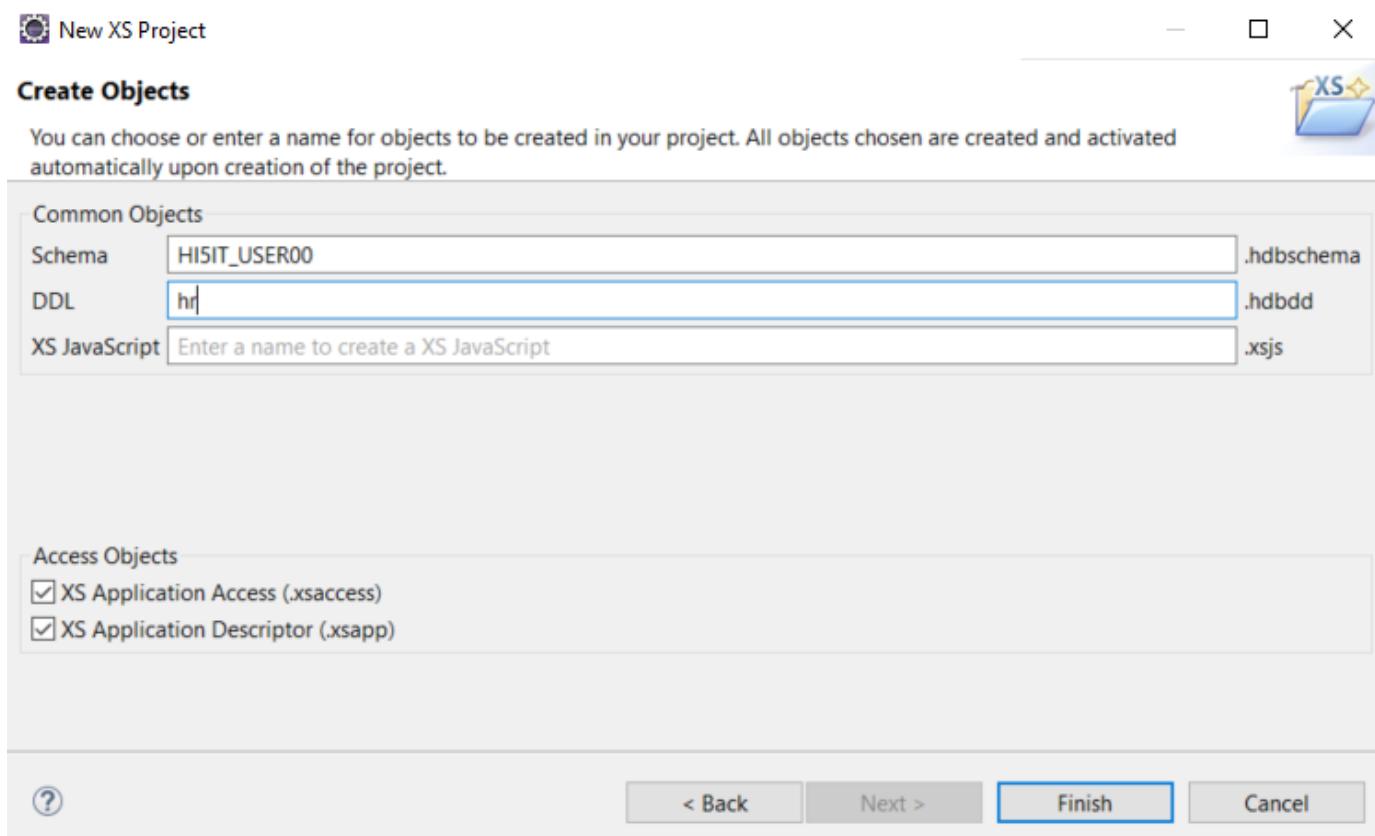
# SAP HANA XS - Extended Application Services

- CRIAR PROJETO XS



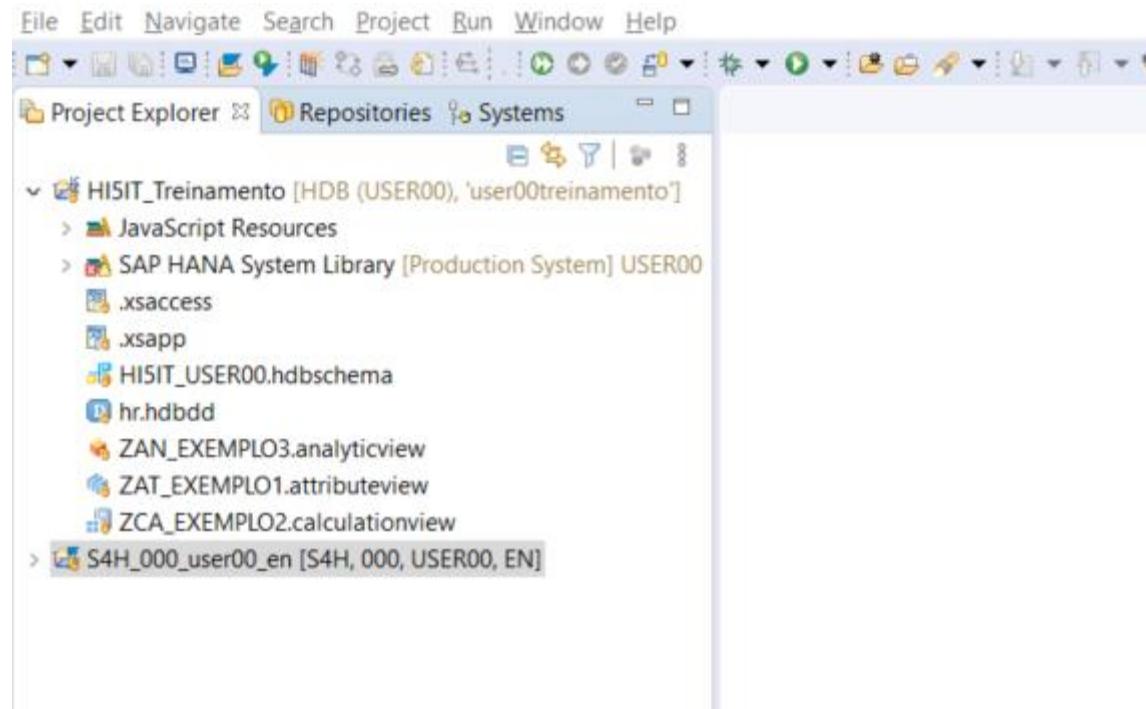
# SAP HANA XS - Extended Application Services

- CRIAR PROJETO XS – Criar objetos



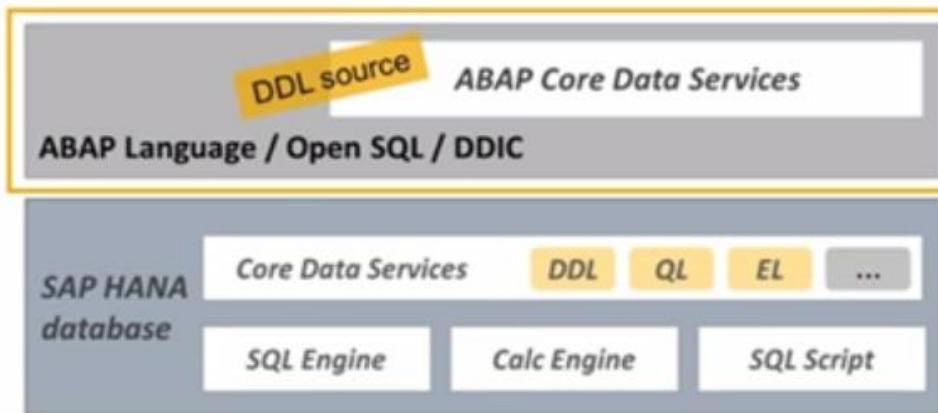
# SAP HANA XS - Extended Application Services

- ESTRUTURA PROJETO XS



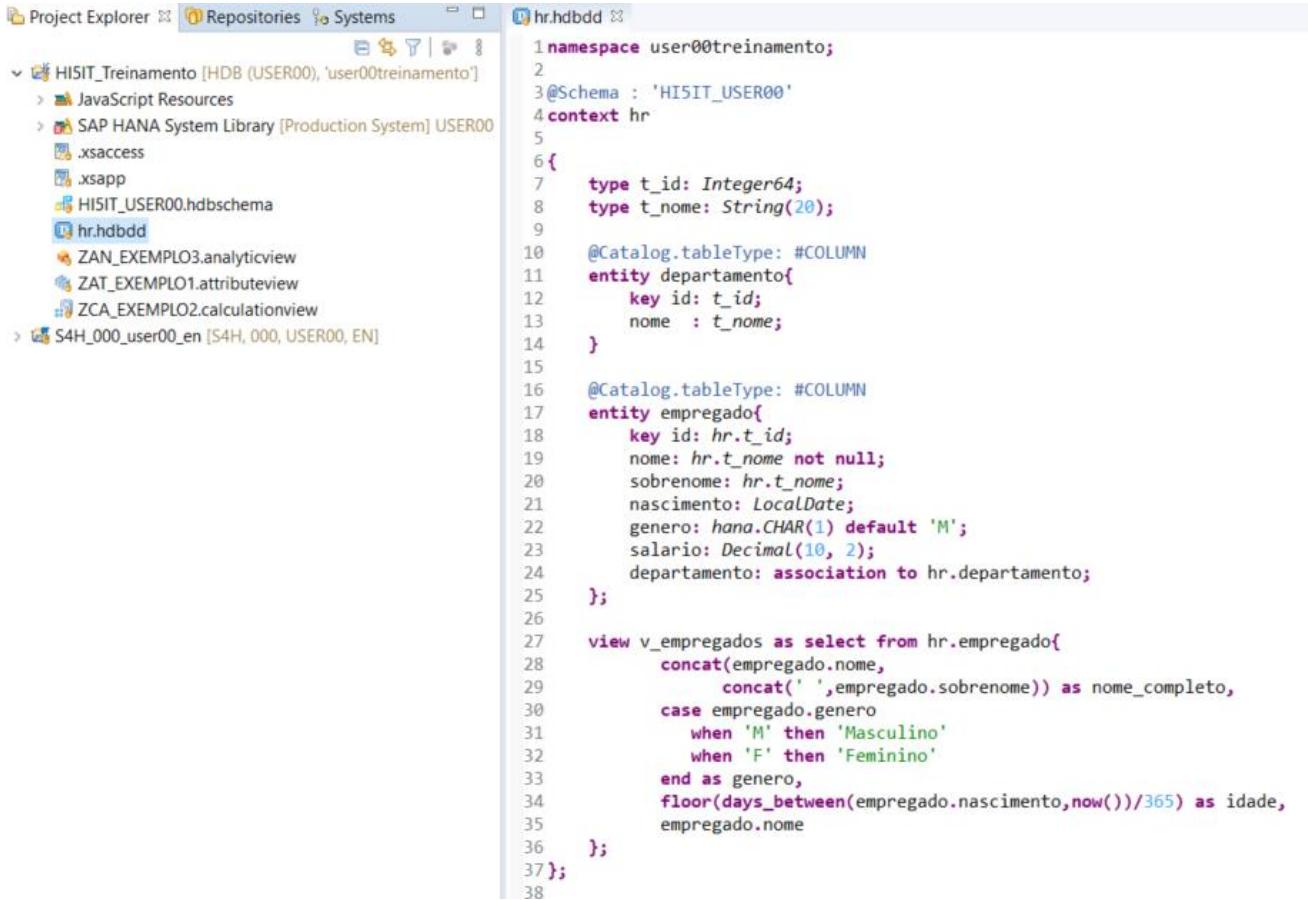
# SAP HANA XS - Extended Application Services

- SAP HANA XS permite que você crie esquema de banco de dados , tabelas , views, e sequences como arquivos de tempo de execução no repositório.
- Isto é possível com a ajuda de um artefato chamado Core data services (CDS ).



# SAP HANA XS - Extended Application Services

- Estrutura arquivo hdbdd

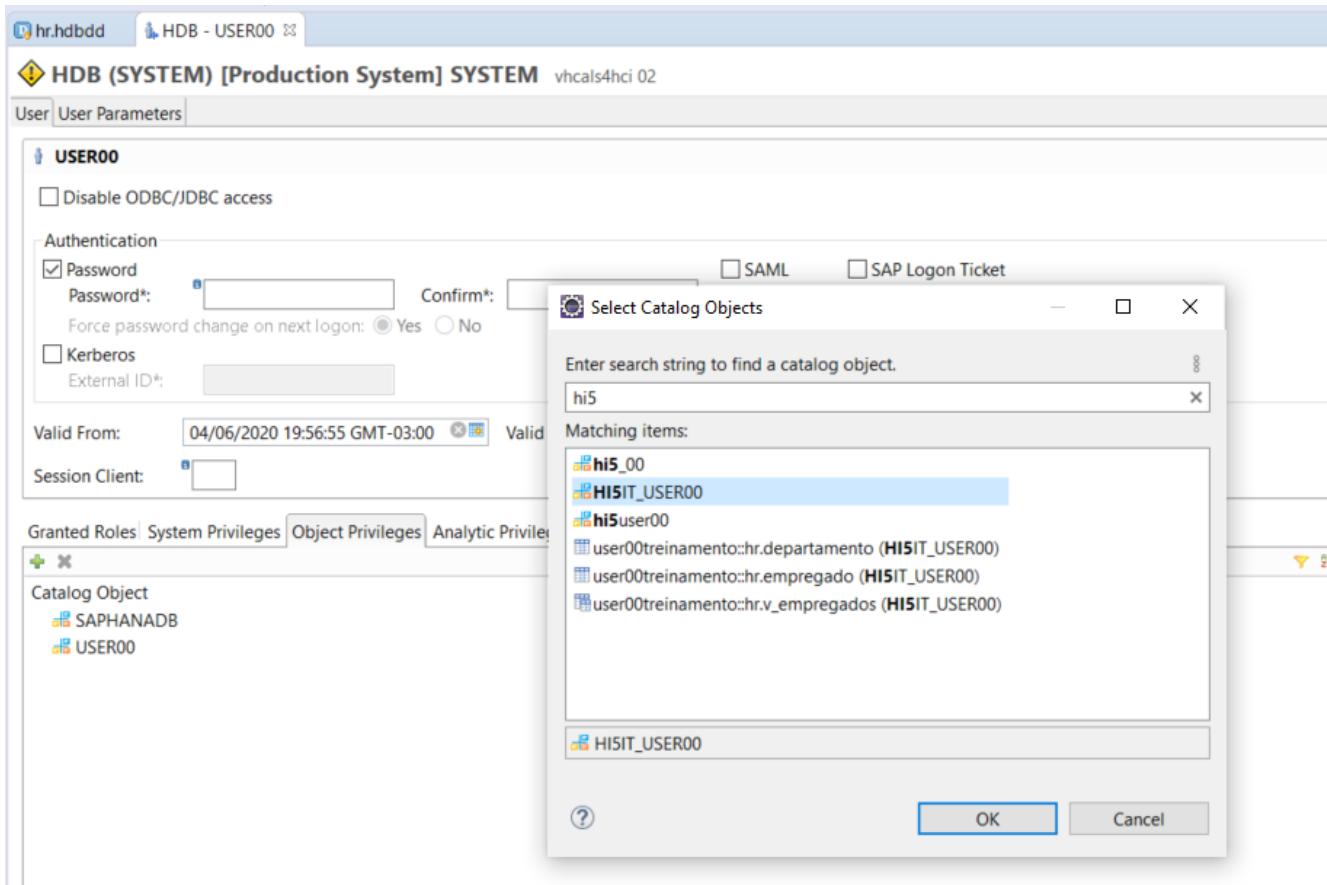


The screenshot shows the SAP Studio interface with two main panes. On the left is the Project Explorer, which lists several projects and files under the 'HI5IT\_Treinamento' project. On the right is the code editor for the file 'hr.hdbdd'. The code defines a namespace, schema, context, and various entities and views.

```
1 namespace user00treinamento;
2
3 @Schema : 'HI5IT_USER00'
4 context hr
5
6 {
7 type t_id: Integer64;
8 type t_nome: String(20);
9
10 @Catalog.tableType: #COLUMN
11 entity departamento{
12 key id: t_id;
13 nome : t_nome;
14 }
15
16 @Catalog.tableType: #COLUMN
17 entity empregado{
18 key id: hr.t_id;
19 nome: hr.t_name not null;
20 sobrenome: hr.t_name;
21 nascimento: LocalDate;
22 genero: hana.CHAR(1) default 'M';
23 salario: Decimal(10, 2);
24 departamento: association to hr.departamento;
25 };
26
27 view v_empregados as select from hr.empregado{
28 concat(empregado.nome,
29 concat(' ',empregado.sobrenome)) as nome_completo,
30 case empregado.genero
31 when 'M' then 'Masculino'
32 when 'F' then 'Feminino'
33 end as genero,
34 floor(days_between(empregado.nascimento,now())/365) as idade,
35 empregado.nome
36 };
37};
38
```

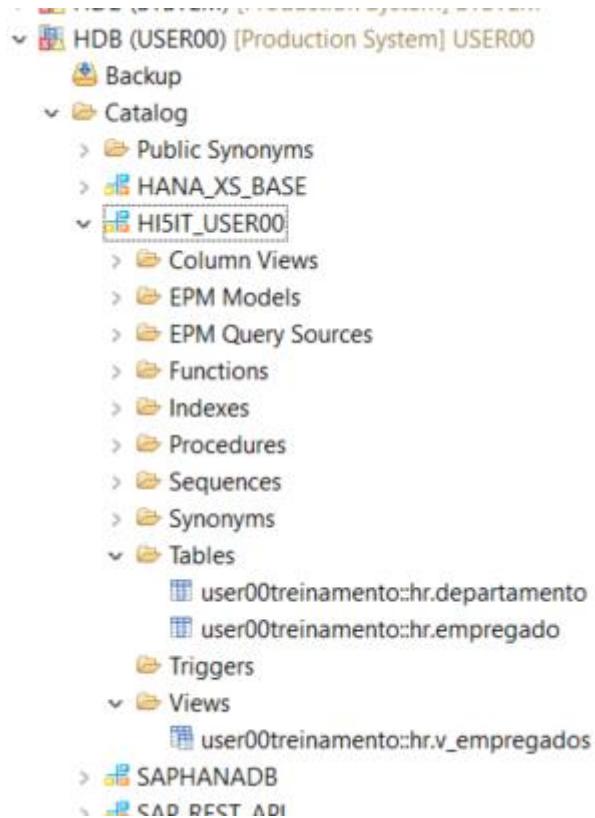
# SAP HANA XS - Extended Application Services

- Caso não tenha autorização para acessar o objeto criado, dar um GRANT para os objetos.

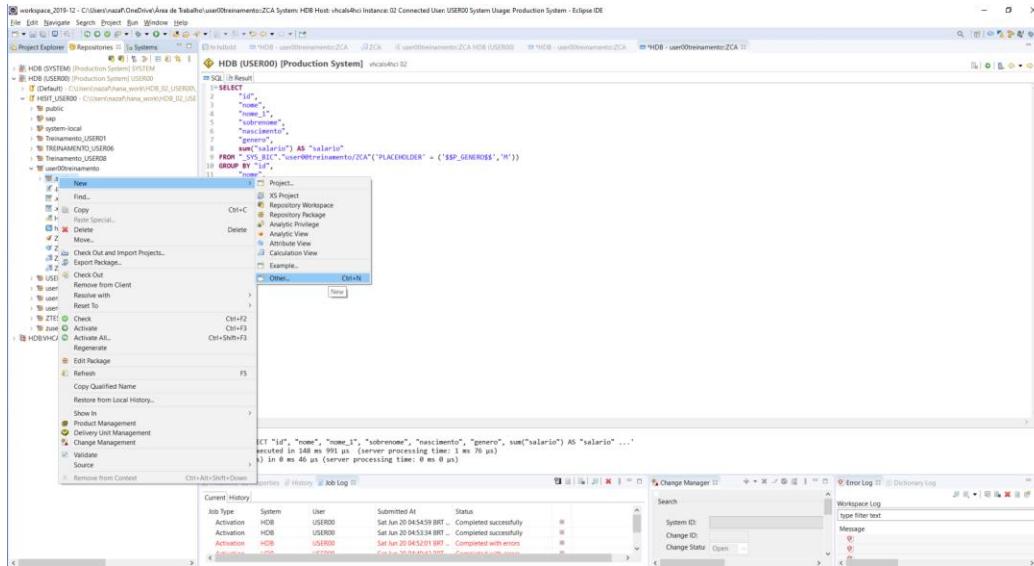


# SAP HANA XS - Extended Application Services

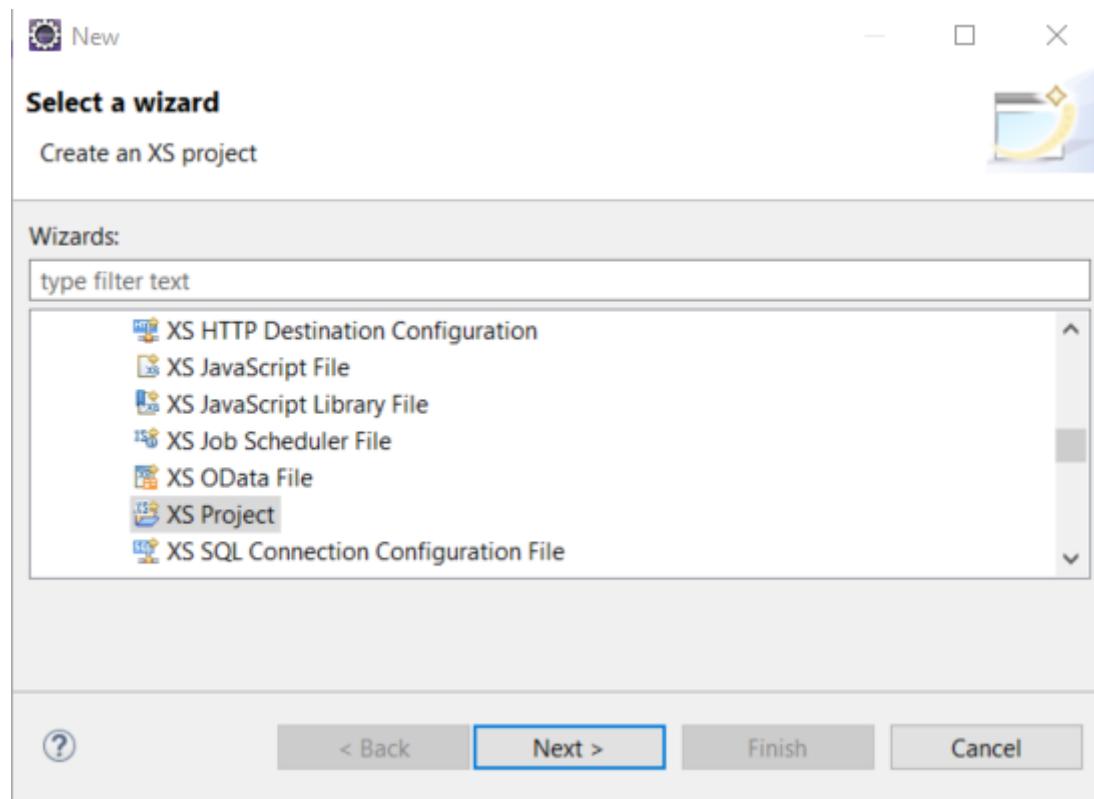
- Schema criado pelo CDS



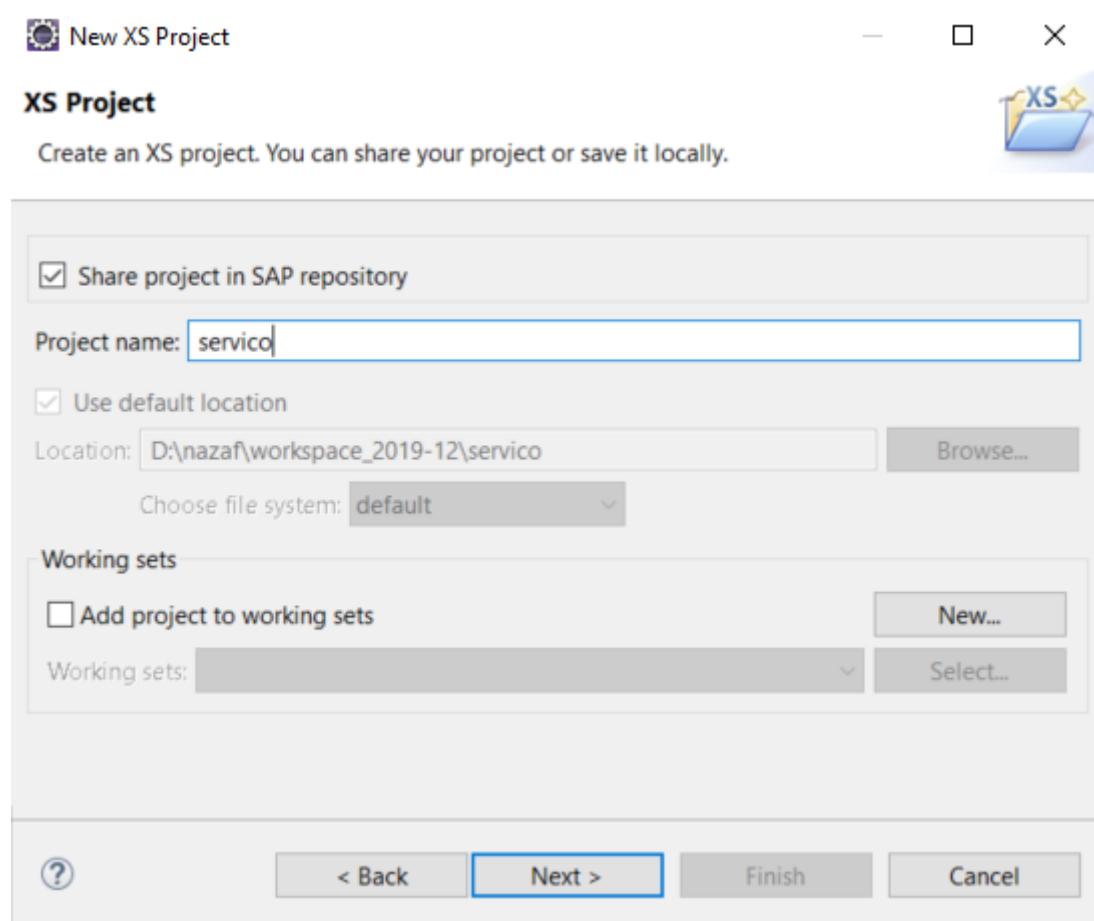
# SAP HANA XS - Extended Application Services



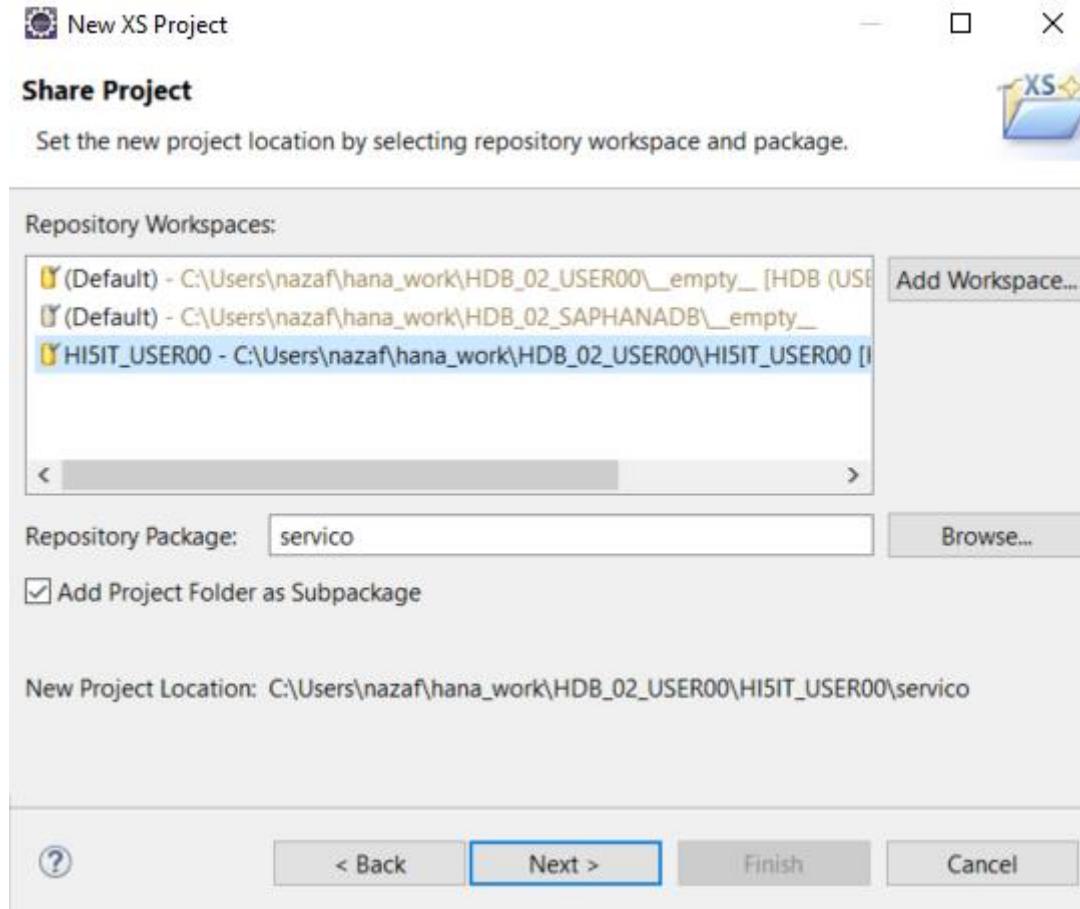
# SAP HANA XS - Extended Application Services



# SAP HANA XS - Extended Application Services



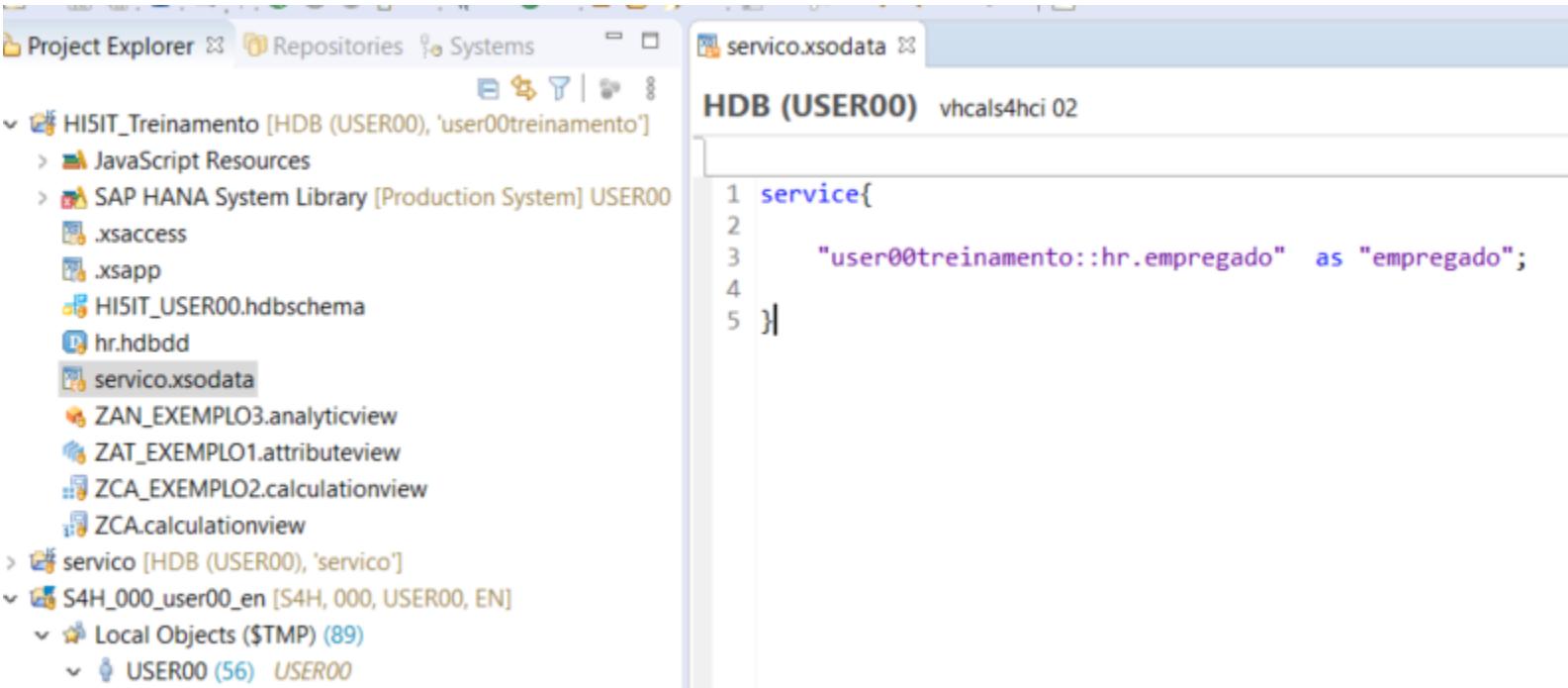
# SAP HANA XS - Extended Application Services



Escolher o diretório e criar o arquivo.

# SAP HANA XS - Extended Application Services

- XSODATA



The screenshot shows the SAP Studio interface. On the left, the Project Explorer displays a project named 'HISIT\_Treinamento' containing various SAP HANA components like 'JavaScript Resources', 'SAP HANA System Library', and several analyticviews ('ZAN\_EXEMPLO3.analyticview', 'ZAT\_EXEMPLO1.attributeview', 'ZCA\_EXEMPLO2.calculationview', 'ZCA.calculationview'). It also lists services ('servico') and local objects ('S4H\_000\_user00\_en'). On the right, the 'servico.xsodata' editor shows the XML configuration for a service named 'servico'. The code defines a service with a single endpoint mapping the 'hr.empregado' table to a local variable named 'empregado':

```
1 <service>
2
3 "user00treinamento::hr.empregado" as "empregado";
4
5 }
```

No arquivo xsodata pode configurar vários serviços.

# SAP HANA XS - Extended Application Services

# SAP HANA XS - Extended Application Services

 New Stored Procedure

**Stored Procedure**

Select the parent folder, and specify the file name and format for your procedure instance.  
Procedure template instances can only be created using the XML file format.

Enter or select the parent folder:

HISIT\_Treinamento

File System View:

- > .HDB\_VHCALS4HCI\_02\_SINGLEDB\_SAPHANADB\_(Default) [System [HDB:VHCALS4HCI:02:SING]
- > .HDB\_VHCALS4HCI\_02\_SINGLEDB\_USER00\_(Default) [System Project]
- > .HDB\_VHCALS4HCI\_02\_SINGLEDB\_USER00\_HI5IT\_USER00 [System Project]
- > HISIT\_Treinamento [HDB (USER00), 'user00treinamento'] **(Selected)**
- > S4H\_000\_user00\_en [S4H, 000, USER00, EN]
- > servico [HDB (USER00), 'servico']

File Name: ZDP\_USER00.hdbprocedure

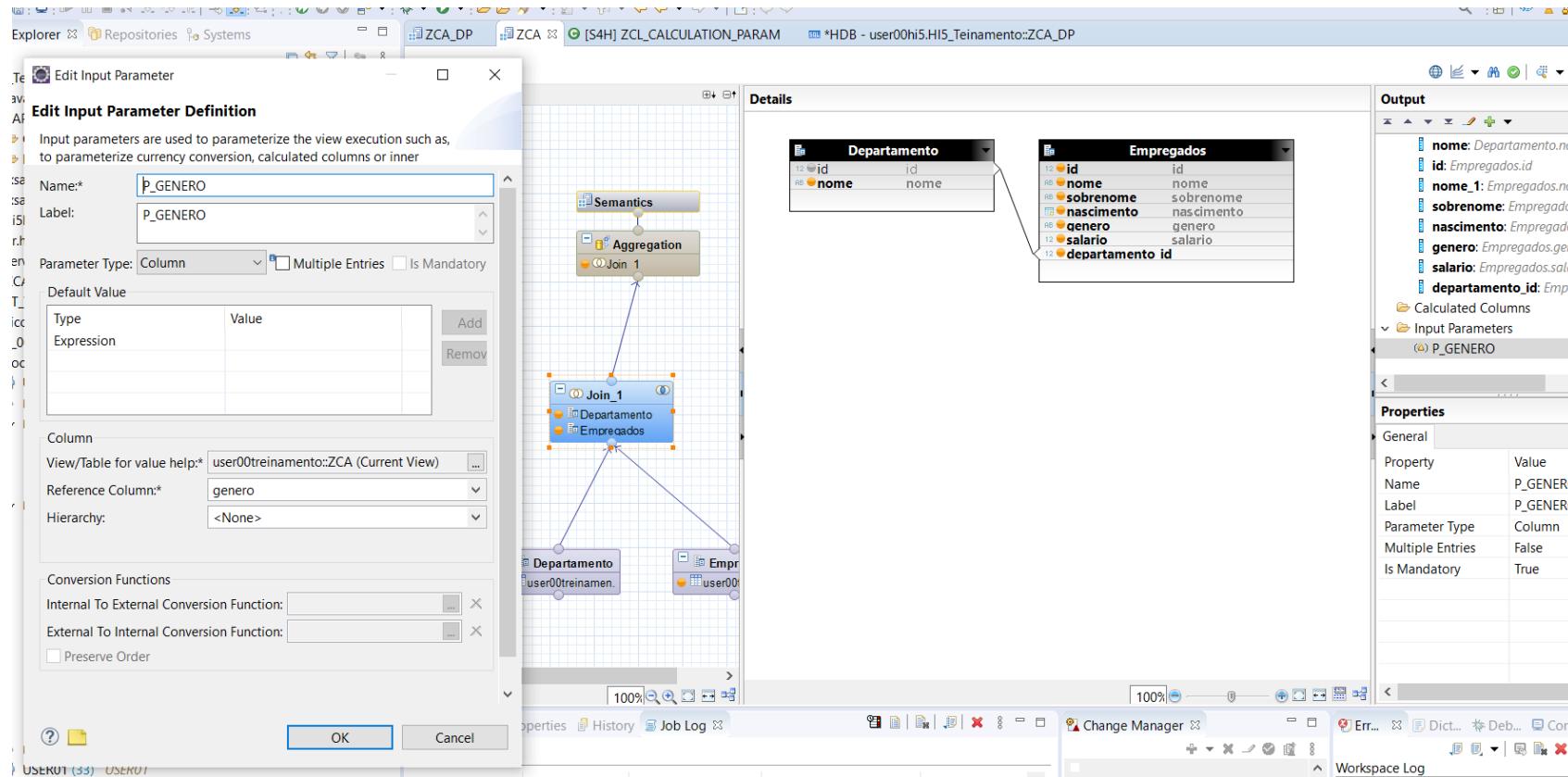
File Format: Text (.hdbprocedure)

Target Schema: HISIT\_USER00 [Browse...](#)

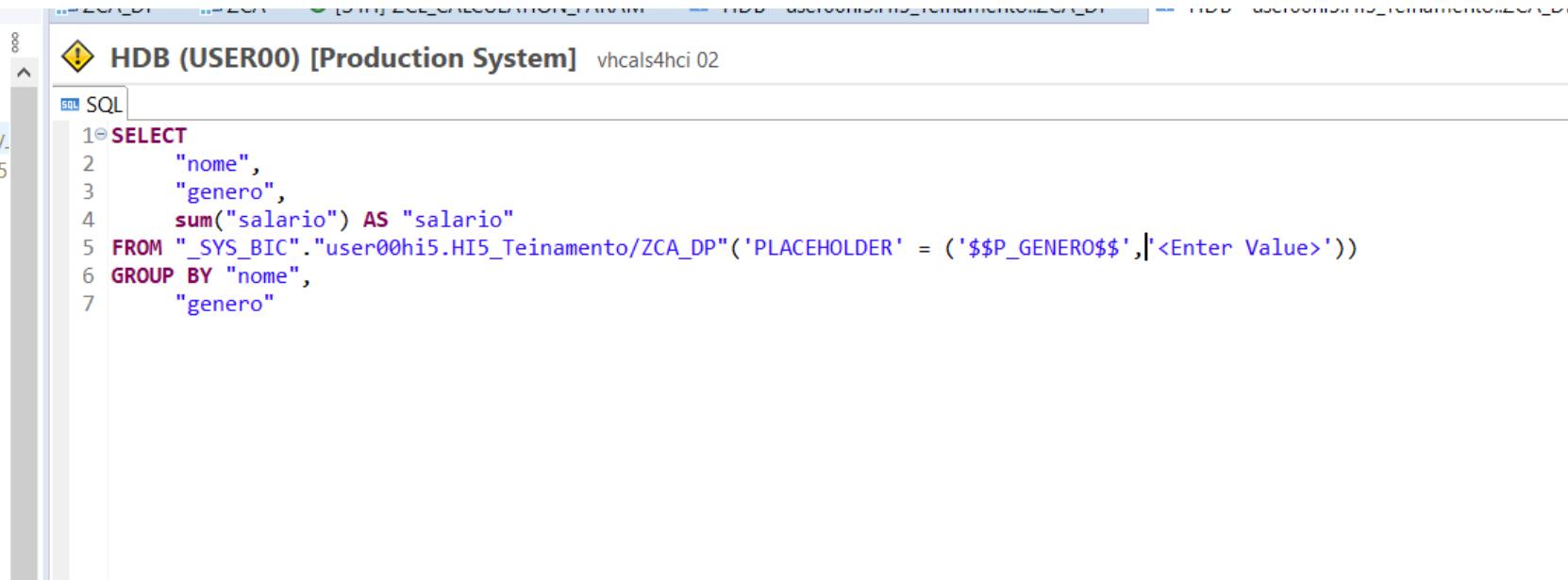
[Advanced >>](#)

Buttons at the bottom:  [Back](#) [Next >](#) [Finish](#) [Cancel](#)

# SAP HANA XS - Extended Application Services



# SAP HANA XS - Extended Application Services



The screenshot shows the SAP HANA Studio interface. The title bar indicates the connection is to 'HDB (USER00) [Production System]' on 'vhcals4hci 02'. The left sidebar has sections for 'ty' and 'II5'. The main area is a SQL editor tab, which is currently active. The SQL code is as follows:

```
1° SELECT
2 "nome",
3 "genero",
4 sum("salario") AS "salario"
5 FROM "_SYS_BIC"."user00hi5.HI5_Teinamento/ZCA_DP"('PLACEHOLDER' = ('$$P_GENERO$$','<Enter Value>'))
6 GROUP BY "nome",
7 "genero"
```

# SAP HANA XS - Extended Application Services

II5IT\_Treinamento [HDB (USER00), 'user00treinamento']  
servico [HDB (USER00), 'servico']  
4H\_000\_user00\_en [S4H, 000, USER00, EN]  
↳ Local Objects (\$TMP) (90)  
↳ USER00 (57) USER00  
↳ Core Data Services (10)  
↳ Data Definitions (10)  
↳ Dictionary (14)  
↳ Source Code Library (32)  
↳ Classes (8)  
↳ ZCL\_CALCULATION\_PARAM CALCULATION WITH PARAMETER  
↳ ZCL\_EXEMPLO02 Exemplo 2  
↳ ZCL\_EXERCICIO01 Exercício 01  
↳ ZCL\_TESTE\_AMDP AMDP Treinamento  
↳ ZCL\_USER00AMDP01 Exemplo 01  
↳ ZCL\_USER00AMDP02 Exemplo 02  
↳ ZCL\_USER00AMDPEXEC2 Exercício 02  
↳ ZCL\_ZUSER00EXEC05 Generated by SADL Generation Toolkit  
↳ Programs (24)  
↳ Transactions (1)  
↳ ZTESTE Teste  
↳ USER01 (33) USER01  
↳ Favorite Packages (0)  
↳ System Library (?)

```
► G ZCL_CALCULATION_PARAM ► S BUSCAR_EMPREGADO
3 FINAL
4 CREATE PUBLIC .
5
6 PUBLIC SECTION.
7 INTERFACES if_amdp_marker_hdb.
8
9 TYPES:
10 BEGIN of ty_empregados,
11 nome_1 type char20,
12 genero type char1,
13 salario type d,
14 END OF ty_empregados.
15
16 TYPES:
17 tt_empregados TYPE STANDARD TABLE OF ty_empregados.
18
19 CLASS-METHODS buscar_empregado
20 IMPORTING
21 VALUE(iv_genero) TYPE ty_empregados-genero
22 EXPORTING
23 VALUE(et_empregados) TYPE tt_empregados.
24 PROTECTED SECTION.
25 PRIVATE SECTION.
26 ENDCLASS.
27
28
29
30 CLASS zcl_calculation_param IMPLEMENTATION.
31 METHOD buscar_empregado BY DATABASE PROCEDURE
32 FOR HDB
33 LANGUAGE SQLSCRIPT
34 OPTIONS READ-ONLY.
35
36 et_empregados = SELECT "nome_1" as nome_1,
37 "genero" as genero,
38 sum("salario") AS salario
39 FROM "_SYS_BIC"."user00treinamento/ZCA"(PLACEHOLDER."$$P_GENERO$$"=>iv_genero)
40 GROUP BY
41 "nome_1",
42 "genero";
43
44 ENDMETHOD.
45
46 ENDCLASS.
```

# SAP HANA XS - Extended Application Services

# Bibliografia

- Livro: SAP ABAP Programming For SAP HANA
- <http://scn.sap.com/community/abap>
- <https://blogs.saphana.com/2013/08/01/sqlscript-procedure-templates-in-sap-hana/>
- [https://help.sap.com/saphelp\\_hanaplatform/helpdata/en/22/47aca08f054d30b0f749f757ec1ccd/content.htm](https://help.sap.com/saphelp_hanaplatform/helpdata/en/22/47aca08f054d30b0f749f757ec1ccd/content.htm)
- <http://scn.sap.com/community/abap/blog/2014/05/21/a-complete-guide-to-opensql-statements--step-by-step-tutorial-with-screenshots>
- <http://scn.sap.com/docs/DOC-51146>
- <http://scn.sap.com/community/abap/hana/blog/2016/02/22/core-data-services-in-abap>
- [http://www.tutorialspoint.com/sap\\_hana/sap\\_hana\\_quick\\_guide.htm](http://www.tutorialspoint.com/sap_hana/sap_hana_quick_guide.htm)
- <http://saphanatutorial.com/sap-hana-procedure-example-table-type-cursor-exception-etc/>
- <http://sapinsider.wispubs.com/Assets/Articles/2014/October/SPI-where-ABAP-meets-SAP-HANA>
- <http://www.tutorialspoint.com/sql/index.htm>
- <http://www.w3schools.com/sql/>