



TRAINNING

Education Services

SAP ABAP
Advanced

TRAINNING EDUCATION SERVICES

ABAP ACADEMY

Módulo ABAP Advanced – 40 horas

www.trainning.com.br

1

2

Curso Avançado de ABAP

Linguagem ABAP & Ferramentas

Tópicos

• Classe de Desenvolvimento	- 4
• Requests	- 5
• Dicionário de Dados	- 111
• Criação de Tabela	- 152
• ABAP Editor	- 193
• ABAP Report	- 198
• Construção de Programa	- 221
• Tela de Seleção	- 227
• Módulo Pool	- 277
• Bacth Input	- 316
• Call Transactions	- 347

3

Classe de Desenvolvimento

- As classes de desenvolvimento são importantes para o transporte entre sistemas de produção e desenvolvimento.
- Todos os objetos pertencentes ao desenvolvimento são combinados em uma classe de desenvolvimento, o qual é atribuído em uma request.
- Todos os objetos associados à classe de desenvolvimento \$TMP são locais e não são transportados

4

Change Request - Tipos

- Geralmente o sistema SAP é definido em três ambientes: Desenvolvimento, Qualidade e Produção.
- Para transportar os objetos de um repositório de um ambiente para outro, é utilizado as Change Requests
- Há seis tipos de Change Requests:
 - Customizing
 - Workbench
 - Transporte de cópias
 - Relocação objetos s/mudança de pacote
 - Relocação objetos c/mudança de pacote
 - Relocação de um pacote completo

5

Change Requests - Tipos

- Customizing
 - São reservadas para transportar conteúdo de tabelas do tipo customizing. Essas tabelas contém informações de configuração dos módulos funcionais (FI, MM, SD etc.) ou configurações do sistema.
- Workbench
 - São utilizadas para transportar objetos do repositório tais como programas, objetos do dicionário de dados tais como tabelas e visões

6

Change Requests - Tipos

- **Transporte de cópias**

- Esse tipo de request pode ser utilizada para transportar objetos a um sistema SAP específico.
- Os objetos são transportados com as versões que possuem no sistema atual.
- Os objetos do sistema de origem não são alterados.

7

Change Requests - Tipos

- **Relocação objetos sem mudança de pacotes**

- Esse tipo de request pode ser utilizada para o desenvolvimento de objetos em outro sistema SAP em uma base temporária.
- Esse tipo de request oferece as mesmas funcionalidades do tipo de request de trasnporte de cópias, ou seja, é possível mover os objetos de um sistema para o outro.
- Ex.: O desenvolvimento pode ser realizado em um ambiente separado, o qual não irá interferir no processo de desenvolvimento.

8

Change Requests - Tipos

- Relocação objetos com mudança de pacotes
 - Esse tipo de request pode ser utilizada para o desenvolvimento de objetos em outro sistema SAP em uma base permanente.
 - Permite a alteração a localidade original dos objetos para o sistema de destino e alterar os pacotes dos objetos ao mesmo tempo.
 - O pacote é alterado automaticamente. Quando um pacote é selecionado, os objetos adquirem o direito de transporte imediatamente após a importação no sistema de destino da request.

9

Change Requests - Tipos

- Relocação de um pacote completo
 - Esse tipo de request é utilizada quando o pacote completo de um ambiente de desenvolvimento é para ser modificado permanentemente.
 - Permite alterar a camada de transporte do pacote automaticamente.
 - Para isso é só especificar o pacote e a camada de transporte onde o pacote deverá ser atribuído. A lista de objetos da request é configurada automaticamente e contém todos os objetos do pacote.

10

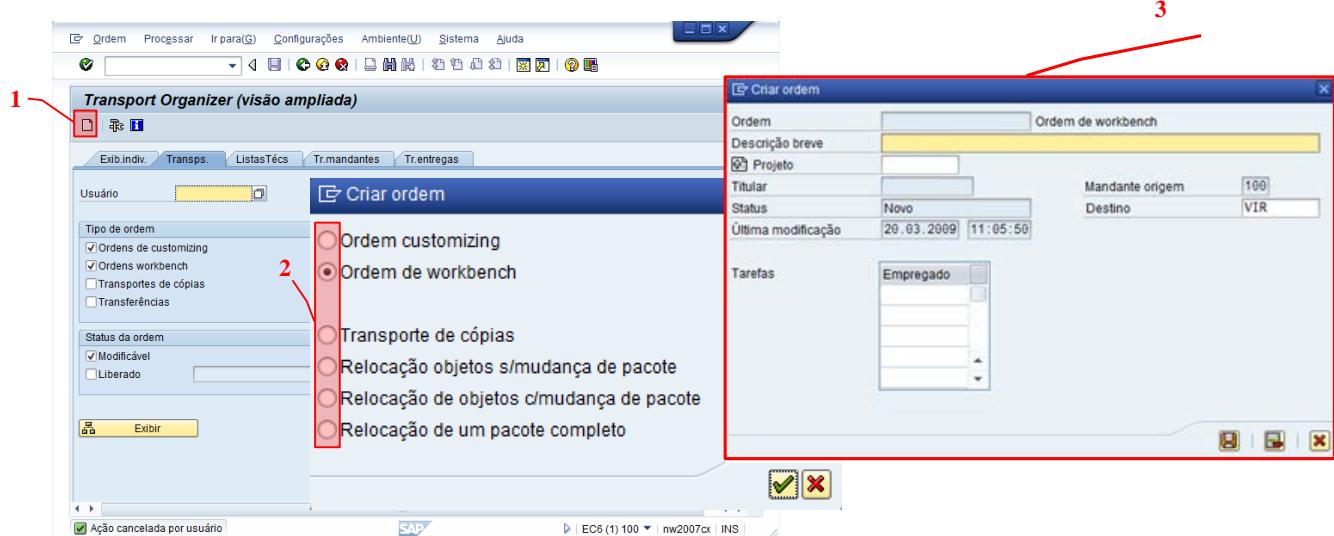
Change Requests

- As change requests possuem uma ou mais tarefas
- As tarefas podem ser atribuídas a diferentes usuários
- As tarefas contém os objetos que serão transportados entre os sistemas
- As transações de change requests são: SE01, SE09 e SE10

11

Change Requests - Criação

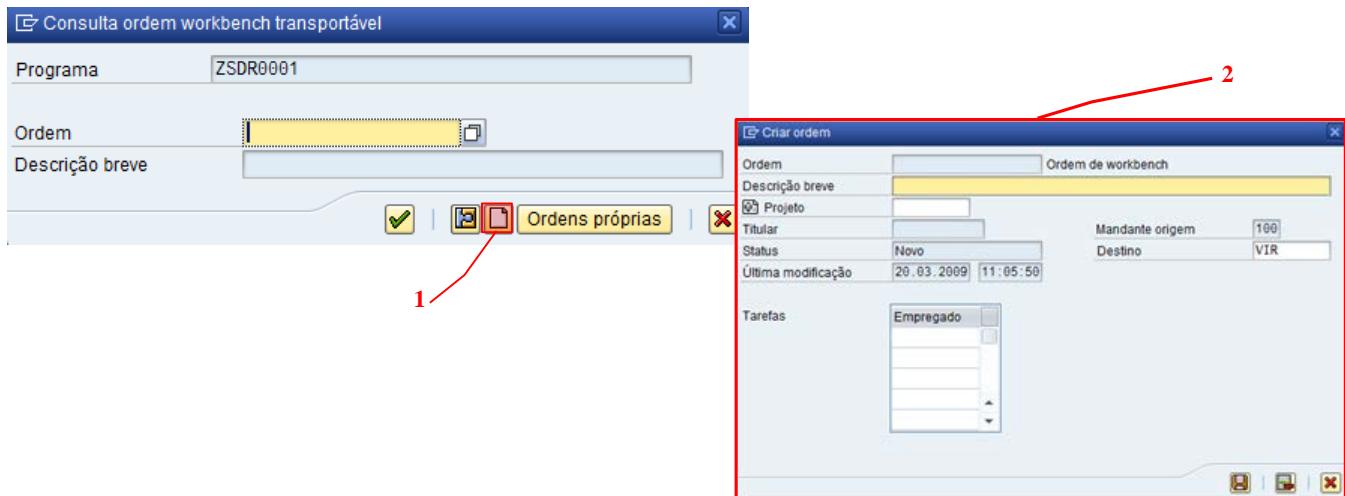
- As requests podem ser criadas através das transações SE01, SE09 ou SE10.



12

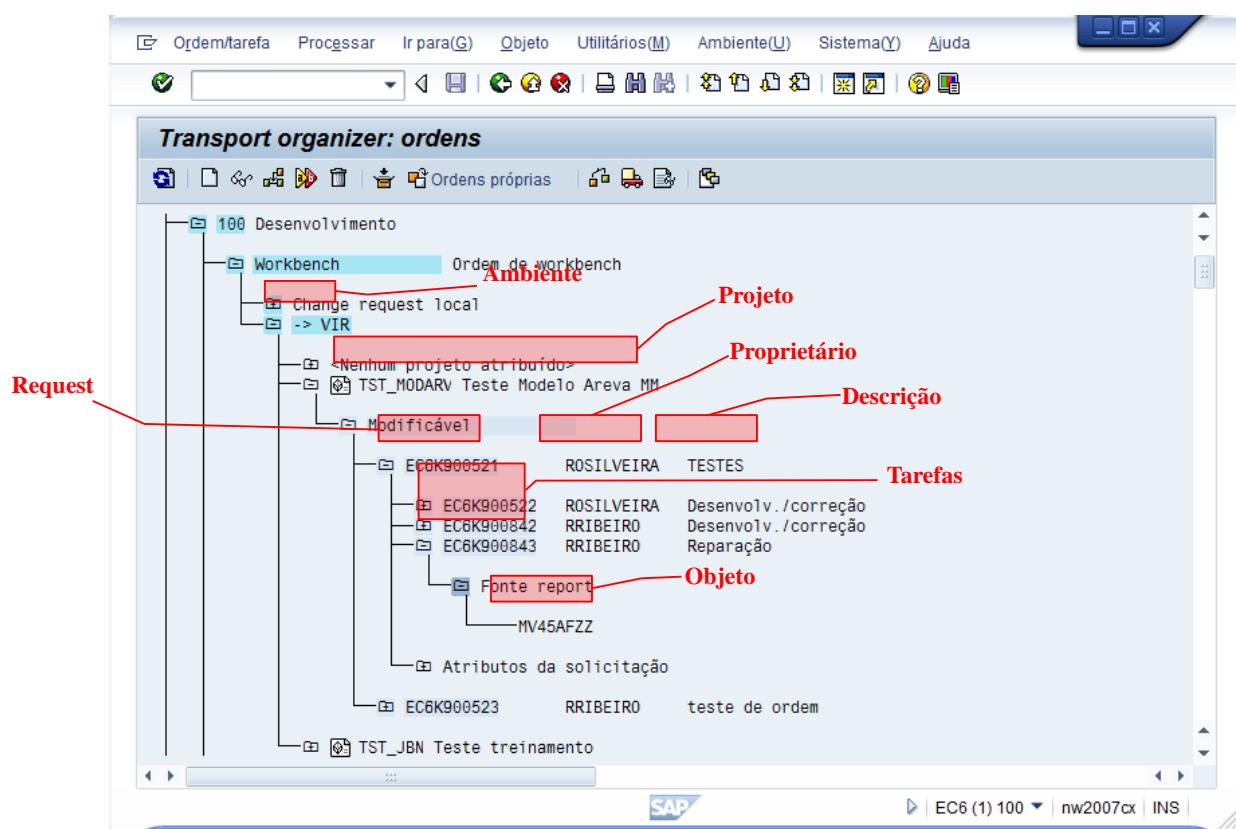
Change Requests - Criação

- As requests podem ser criadas no momento da criação de um objeto



13

Change Requests

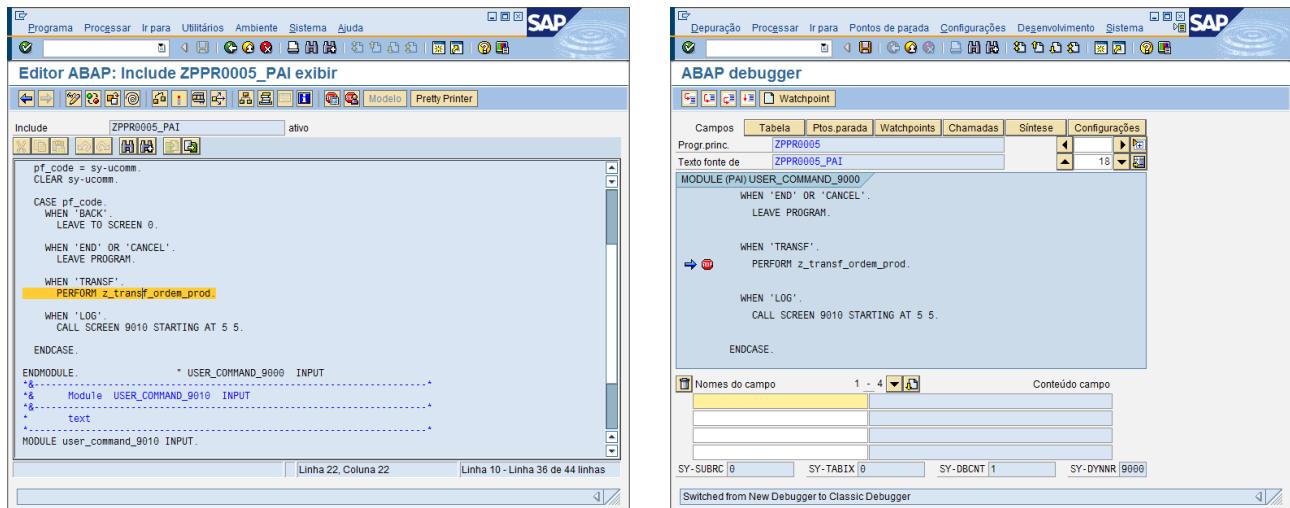


14

Debug

Debug

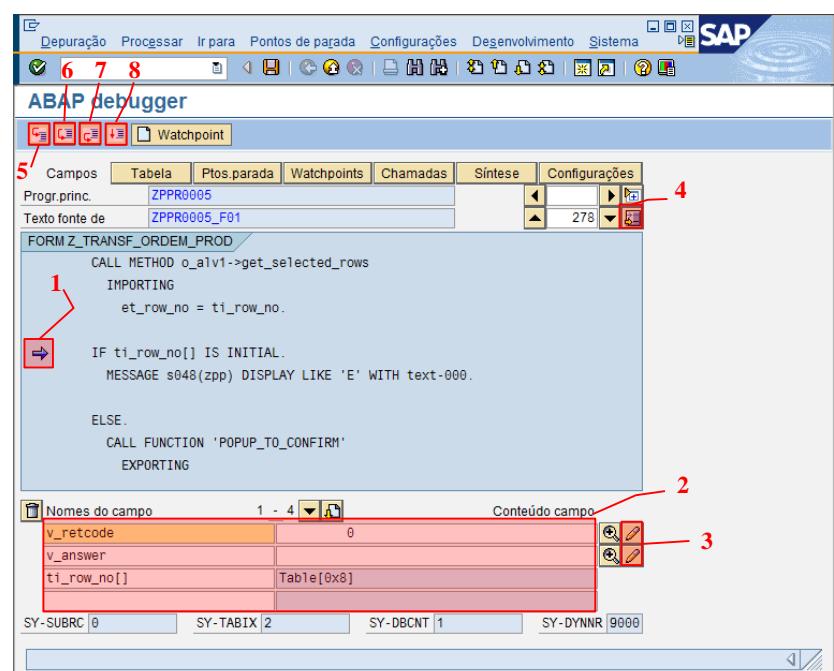
- Através dos botões “Definir/Eliminar ponto de parada externo” e “Definir/Eliminar ponto de parada de sessão”.
- É necessário entrar no código do programa, inserir o ponto de parada na linha desejada e executar o programa.



17

Debug - Funcionalidades

1. Ponteiro
2. Caixa de Variáveis
3. Botão para alterar o valor da variável
4. Exibir instrução atual
5. Etapa Individual (F5)
6. Executar rotina *(F6)
7. Retornar da Rotina (F7)
8. Avançar até o cursor (F8)



18

ABAP - Tipos de Dados & Declarações

Tópicos

- **Tipos de Dados**
- **Declaração dos tipos de dados**
 - **Variáveis**
 - **Constantes**
 - **Tipos de dados elementares**
 - **Estruturas**
 - **Tabelas internas**
 - **Campos de sistema**

Tipos de Dados

- A tabela abaixo mostra os tipos ABAP pré-definidos:

Tipo	Tamanho	Tamanho padrão	Descrição
b	1 Byte		Inteiro de 1 byte (interno)
c	1 a 65,535 caracteres	1 caracteres	Campo texto
cursor	Igual ao i	Igual ao i	Cursor de banco de dados
d	8 caracteres		Campo data
f	8 bytes		Número de ponto flutuante
i	4 bytes		Inteiro de 4 bytes
n	1 a 65,535 characters	1 caracteres	Texto numérico
p	1 a 16 bytes	8 bytes	Número "packed"
string	Variável		String de texto
s	2 bytes		Inteiro de 2 bytes (internal)
t	6 caracteres		Campo de tempo
x	1 a 65,535 bytes	1 byte	Campo de byte
xstring	Variável		String de byte

21

Tipos de Dados

- A tabela abaixo apresenta os valores permitidos e os valores iniciais

Tipo	Valores	Valor Inicial
b	De 0 a 255	0
c	Qualquer caractere alfanumérico	" " para cada posição
d	8 caracteres alfanuméricos definidos pelo usuário; Somente valores no formato AAAAMMDD, os quais são permitidos como entrada de data de acordo com as regras de calendário: AAAA(ano): 0001 a 9999, MM(mês): 01 a 12, DD(dia): 01 a 31	"00000000"
f	Os pontos de número flutuante são apresentados internamente com 16 casas decimais de acordo com o padrão IEEE-754 (dupla precisão). Os valores permitidos são -1,7976931348623157EE+308 a -2,2250738585072014EE-308 para a área negativa, o valor zero, e +2,2250738585072014EE-308 a +1,7976931348623157EE+308 para a área positiva. A validade de ambas as áreas são estendidas na direção do zero através de números "denormalized" depois do IEEE-754.	0
i	-2.147.483.648 to +2.147.483.647	0
n	Qualquer caractere alfanumérico, entretanto, os valores válidos são somente os dígitos de 0 a 9.	"0" para cada posição

22

Tipos de Dados - Continuação

Tipo	Valores	Valor Inicial
p	O tamanho válido para os números "packed" está entre 1 e 16 bytes; duas casas decimais são embaladas em um byte, onde o último byte contém somente uma posição e o sinal de positivo e negativo. Depois do separador de decimal, até 14 casas decimais são permitidas. Dependendo do tamanho do campo len e do número de casas decimais dec, aplica-se as seguintes áreas de valores: $(-10^{(2len -1)} +1) / (10^{(+dec)})$ a $(+10^{(2len -1)} -1) / (10^{(+dec)})$ em passos de $10^{(-dec)}$. Valores dentro desse intervalo são arredondados.	0
string	Igual ao tipo c	string vazia de tamanho 0
s	-32.768 to +32.767	0
t	6 caracteres alfanuméricos definidos pelo usuário; o único valor válido são os números que são interpretados como especificação de tempo nas 24 horas do formato do relógio HHMMSS. A SAP recomenda a utilizar exclusivamente 00 a 23 para HH(horas) e 00 a 59 para MM (minutos) e 00 a 59 para SS (segundos).	"000000"
x	Caracteres hexadecimais 0-9, A-F	hexadecimal 0
xstring	Igual ao tipo x	string vazia de tamanho 0

23

Declaração de tipos de dados

- O comando DATA é utilizado para declarar as variáveis, estruturas e tabelas internas.
- O comando CONSTANTS declara as constantes do programa
- O comando TYPES permite criar tipos de dados elementares, tipos de referência e tipos de tabela, o qual será utilizado como referência para variáveis, estruturas e tabelas
- O comando TABLE cria uma work-area da tabela definida

24

Variáveis

- A declaração das variáveis se dá através da sintaxe:

```
DATA { {var[(len)] TYPE abap_type [DECIMALS dec]}  
      | {var TYPE abap_type [LENGTH len] [DECIMALS dec]} }  
          [VALUE val|{IS INITIAL}]  
          [READ-ONLY].
```

Exemplo:

```
DATA v_char1(10) TYPE c. "Tipo char com 10 posições  
DATA v_int1      TYPE i. "Tipo inteiro  
DATA v_dec1      TYPE p DECIMALS 2. "Tipo compactado com 2 casas decimais  
DATA v_char2(2)   TYPE c VALUE 'ST'. "Tipo char com 2 posições e valor  
                                         "padrão ST
```

25

Tipos de dados elementares

- O comando TYPES define um tipo de dado independente ou um tipo de dados estruturado.
- Qualquer tipo de dados independente é definido com a adição TYPE e um tipo de dados ou com a adição LIKE e o objeto de dados.
- É possível criar tipo de dados elementares, tipos de referência, tipos estruturados e tipos de tabelas.
- Para o tipo de dados estruturado, utiliza-se o bloco de comandos BEGIN OF <tipo> e END OF <tipo> junto com o comando TYPES.

26

Construção de tipos de dados

- **Tipo de dados elementares**

```
TYPES: text1      TYPE c LENGTH 10,  
       text2      TYPE c LENGTH 15.
```

```
DATA: v_msg      TYPE text1, "Variável  
      v_coment   TYPE text2. "Variável
```

- **Tipos estruturados**

```
TYPES: BEGIN OF ty_mara,  
       matnr    TYPE mara-matnr,  
       mtart    TYPE mara-mtart,  
       zeinr    TYPE mara-zeinr,  
     END OF ty_mara.
```

```
DATA: wa_mara    TYPE ty_mara. "Work-area
```

27

Tipos de dados elementares

- **Tipo de tabelas**

```
TYPES spfli_sort TYPE SORTED TABLE OF spfli  
      WITH UNIQUE KEY carrid connid.
```

```
DATA: ti_spfli TYPE spfli_sort. "Criação de tabela  
      "interna
```

28

Constantes

- O comando CONSTANTS cria um objeto do tipo constante, ou seja, um objeto que não pode ter o valor alterado.
- Exemplo:

```
CONSTANTS: c_mtart      TYPE mara-mtart VALUE 'FERT'.
```

```
CONSTANTS: BEGIN OF sap_ag,
            zip_code  TYPE n LENGTH 5 VALUE '69189',
            city      TYPE string VALUE `Walldorf`,
            country   TYPE string VALUE `Germany`,
            END OF sap_ag.
```

29

Estrutura

- A estrutura pode ser criada de três formas:
 - Referência a uma tabela transparente
 - Referência a uma estrutura do dicionário de dados
 - Referenciado a um tipo de dados local
- A estrutura é criada através do comando DATA

30

Estrutura - Criação de estrutura

- Declarado com o tipo de uma tabela transparente ou estrutura do dicionário de dados

Exemplo1: Criação da estrutura (work-area) wa_mara com os componentes da tabela transparente MARA.

```
DATA wa_mara    LIKE mara. Ou  
DATA wa_mara    TYPE mara.
```

Exemplo2: Criação da estrutura (work-area) wa_return com os componentes da estrutura do dicionários de dados BAPIRET2.

```
DATA wa_return   LIKE BAPIRET2. Ou  
DATA wa_return   TYPE BAPIRET2.
```

31

Estrutura - Criação de estrutura

- Declarado com o tipo de dados local

```
TYPES: BEGIN OF ty_mara,  
        matnr  TYPE mara-matnr,  
        mtart  TYPE mara-mtart,  
        zeinr  TYPE mara-zeinr,  
      END OF ty_mara
```

Exemplo: Criação da estrutura WA_MARA com referência ao tipo de dados TY_MARA.

```
DATA: wa_mara  TYPE ty_mara.
```

32

Estrutura - Criação de estrutura

- **Declaração com o comando DATA**

Exemplo: Criação da estrutura WA_MARA com os componentes MATNR, MTART e ZEINR da tabela transparente MARA.

```
DATA: BEGIN OF wa_mara,  
      matnr  TYPE mara-matnr,  
      mtart  TYPE mara-mtart,  
      zeinr  LIKE mara-zeinr,  
    END OF wa_mara
```

33

Tabela Interna

- A tabela interna fornece um meio de armazenar dados de uma estrutura fixa e armazená-la em memória.
- Os dados são armazenados linha por linha na memória e cada linha tem a mesma estrutura.
- No ABAP, as tabelas internas substituem as funcionalidades de arrays.
- O tipo de dados de uma tabela interna é definido pelo tipo de linha, chave e tipo de tabela interna.

34

Tabela Interna - Tipo de Linha

- O tipo de linha de uma tabela interna pode ser de qualquer tipo, sendo que normalmente é uma estrutura.
- Cada componente da estrutura é uma coluna na tabela interna.
- O tipo de linha pode ser elementar ou outra tabela interna.

35

Tabela Interna - Chave

- A chave identifica a coluna da tabela
- Há dois tipos de chaves:
 - Chave standard
 - Chave definido pelo usuário
- A chave a ser especifica pode ser UNIQUE ou NON-UNIQUE
- Tabelas internas com chaves únicas, não podem ter registros duplicados com a mesma chave. A univocidade irá depender do método de acesso à tabela.

36

- O tipo de tabela define como o ABAP irá acessar os registros individuais da tabela.
- Existem três tipos de tabelas internas:
 - Standard table
 - Sorted table
 - Hashed table

37

Tabela Interna - Standard table

- A standard table possui um índice linear interno.
- O sistema pode acessar os registros pelo índice da tabela ou pela chave.
- O tempo de resposta de acesso através da chave é proporcional ao número de entradas na tabela
- A chave da tabela standard não é única e não pode ser especificada uma chave única
- Os registros são inseridos rapidamente pois não é necessário verificar a existência do registro

38

Tabela Interna - Sorted table

- Os registros da sorted table são sempre inseridos de forma ordenada.
- Possuem índice interno.
- O acesso aos registros podem ser efetuados pelo índice ou pela chave.
- O tempo de resposta pela chave é logaritmicamente proporcional ao número de entradas na tabela, devido à procura binária.
- As chaves a serem utilizadas podem ser únicas ou não únicas e devem ser especificadas na declaração da tabela interna.

39

Tabela Interna - Hashed table

- As hashed tables não possuem índice linear.
- Os registros da tabela só podem ser acessados através de chaves.
- O tempo de resposta de acesso à tabela independe da quantidade de registros e é constante, pois o sistema utiliza um algoritmo hash.
- A chave de acesso deve ser único.
- Ao definir a tabela deve ser especificado o parâmetro UNIQUE.

40

Tabela interna - Header Line

- Header line é uma work-area utilizada quando a tabela interna é processada.
- A work-area está vinculada à tabela interna e é declarada com o parâmetro WITH HEADER LINE.
- Exemplo:

```
DATA: ti_mara      TYPE mara OCCURS 0 WITH HEADER LINE.
      READ TABLE ti_mara WITH KEY matnr = 'SCE002'.
```

MATNR	MTART	ZEINR
SCE002	VERP	234545
8D08	FERT	122212
SCE001	FERT	534213
SCE002	VERP	234545
7D5F	HALB	112221

41

Criação de tabelas internas

- Standard table

```
DATA: BEGIN OF ti_mara OCCURS 0,
      matnr      TYPE mara-matnr,
      mtart      TYPE mara-mtart,
      zeinr      TYPE mara-zeinr,
    END OF ti_mara.
```

```
DATA: ti_mara      TYPE ty_mara OCCURS 0 WITH HEADER LINE.
```

```
DATA: ti_mara      TYPE TABLE OF ty_mara.
```

```
DATA: ti_mara      TYPE mara OCCURS 0 WITH HEADER LINE.
```

42

Criação de tabelas internas

- **Sorted table**

```
TYPES: BEGIN OF ty_mara OCCURS 0,  
        matnr      TYPE mara-matnr,  
        mtart      TYPE mara-mtart,  
        zeinr      TYPE mara-zeinr,  
    END OF ty_mara.
```

```
DATA: ti_mara      TYPE SORTED TABLE OF ty_mara WITH NON-UNIQUE KEY  
      matnr .
```

43

Criação de tabelas internas

- **Hashed table**

```
TYPES: BEGIN OF ty_mara OCCURS 0,  
        matnr      TYPE mara-matnr,  
        mtart      TYPE mara-mtart,  
        zeinr      TYPE mara-zeinr,  
    END OF ty_mara.
```

```
DATA: ti_mara      TYPE HASHED TABLE OF ty_mara WITH UNIQUE KEY  
      matnr .
```

44

Campos de sistema -SYST

- Os campos de sistema ABAP estão sempre disponíveis nos programas ABAP.
- O sistema em tempo de execução preenche os campos de acordo com o contexto.
- Podem ser utilizados nos programas para obter o status do sistema.
- Os campos de sistema são variáveis, mas devem ser tratados como constantes, pois ao alterar algum valor, o fluxo do sistema pode ser alterado

45

Campos de sistema - SYST

- Os campos de sistema estão definidos na estrutura do dicionário de dados SYST e nos programas estão disponíveis na estrutura SY.
- Exemplos de campos de sistema:
 - SY-MANDT – Retorna o mandante do logon
 - SY-SUBRC – Código de retorno dos comandos ABAP
 - SY-TABIX – Índice da linha (tabela interna)
 - SY-BATCH – Executa o programa em background

46

Campos de sistema - SYST

- Os campos de sistema estão definidos na estrutura do dicionário de dados SYST e nos programas estão disponíveis na estrutura SY.
- Exemplos:
 - SY-MANDT – Retorna o mandante do logon
 - SY-SUBRC – Código de retorno dos comandos ABAP
 - SY-TABIX – Índice da linha (tabela interna)
 - SY-BATCH – Executa o programa em background
 - SY-TCODE – Código da transação atual
 - SY-UNAME – Nome do usuário
 - SY-DATUM – Data atual do sistema
 - SY-UZEIT – Hora atual do sistema
 - SY-UCOMM – Código de função que acionou o PAI

47

Campos de sistema - SYST

Campo	Descrição
INDEX	Contador de loops
PAGNO	Página da lista atual
TABIX	Índice de tabelas internas
TFILL	Número de linhas de tabelas internas
DBCNT	Entradas de tabela de banco de dados processadas
FDPOS	Ocorrência em cadeia de bytes ou caracteres
COLNO	Coluna atual na lista
LINCT	Comprimento das páginas da lista
LINNO	Linha atual na lista
LINSZ	Largura da linha da lista
PAGCT	Campo do sistema obsoleto
MACOL	Nº de colunas na margem esquerda de uma lista de impressão
MAROW	Nº de colunas na margem superior de uma lista de impressão
TLENG	Comprimento das linhas de uma tabela interna
LILLI	Linha da lista selecionada
SUBRC	Código de retorno de instruções ABAP
CUCOL	Posição do cursor horizontal em PAI
CUROW	Posição do cursor vertical no PAI
LSIND	Índice da lista de ramificação
LISTI	Índice da lista exibida
STEPL	Índice da linha step loop atual
SROWS	Telas, número de linhas
SCOLS	Telas, número de colunas
LOOPC	Linhas visíveis de um step loop
TZONE	Data e hora, diferença de horas para hora de referência UTC

Campo	Descrição
DAYST	Flag de horário de verão
FDAYW	Calendário de fábrica-dia da semana
LANGU	Código de idioma do ambiente de texto atual
MODNO	Índice do modo externo
BATCH	Programa corre em background
BINPT	O programa corre em batch input
CALLD	Programa ABAP, modo de chamada do programa ABAP
DYNNR	Nº da tela atual
DYNGR	Grupo de telas da tela atual
WTITL	Marcação para cabeçalho da página standard
CPAGE	Nº da página atual de uma lista
DBNAM	Banco de dados lógico de um programa executável
MANDT	Identificação do mandante do usuário atual
PEXPI	Período de retenção spool (parâmetro de impressão)
PRINI	Campo do sistema interno
PRIMM	Saída imediata (parâmetro de impressão)
PRREL	Eliminar após saída (parâmetro de impressão)
PRBIG	Falso-rosto da seleção (parâmetro de impressão)
PRNEW	Nova ordem spool (parâmetro de impressão)
PDEST	Dispositivo de saída (parâmetro de impressão)
PLIST	Nome da ordem spool (parâmetro de impressão)
PRDSN	Nome do spool file
CALLR	Local de início de impressão da lista
RTITL	Título do progra.a imprimir (parâmetro de impressão)
PRREC	Destinatário (parâmetro de impressão)
PRTXT	Texto para falso-rosto (parâmetro de impressão)
PRABT	Departamento no falso-rosto (parâmetro de impressão)

48

Campos de sistema - SYST

Campo	Descrição
PAART	Preparação para impressão (parâmetro de impressão)
PRCOP	Número de expressões (parâmetro de impressão)
DBSYS	Sistema de banco de dados central
SYSID	Nome do sistema SAP
OPSY	Sistema operacional do servidor de aplicação
PFKEY	Status GUI atual
SAPRL	Release do sistema da SAP
TCODE	Código de transação atual
UCOMM	Código de função que acionou o PAI
SPONO	Nº spool na saída de listagem
DATUM	Data atual do servidor de aplicação
SLSET	Variante utilizada
UZEIT	Hora atual do servidor de aplicação
UNAME	Nome do usuário
LSTAT	Campo do sistema obsoleto
ABCDE	Alfabeto latino
TITLE	Conteúdo da linha de título
ENTRY	Campo do sistema interno
LISEL	Conteúdo de uma linha da lista selecionada
ULINE	Traço horizontal
CROG	Programa de chamada
LDBPG	Programa de banco de dados de um banco de dados lógico
TVAR0	Variável txt.p/caractere preenchimento em títulos da lista
TVAR1	Variável txt.p/caractere preenchimento em títulos da lista
TVAR2	Variável txt.p/caractere preenchimento em títulos da lista
TVAR3	Variável txt.p/caractere preenchimento em títulos da lista

Campo	Descrição
TVAR4	Variável txt.p/caractere preenchimento em títulos da lista
TVAR5	Variável txt.p/caractere preenchimento em títulos da lista
TVAR6	Variável txt.p/caractere preenchimento em títulos da lista
TVAR7	Variável txt.p/caractere preenchimento em títulos da lista
TVAR8	Variável txt.p/caractere preenchimento em títulos da lista
TVAR9	Variável txt.p/caractere preenchimento em títulos da lista
MSGID	Classe de mensagem
MSGTY	Tipo de mensagem
MSGNO	Nº mensagem
MSGV1	Variável mensagens
MSGV2	Variável mensagens
MSGV3	Variável mensagens
MSGV4	Variável mensagens
VLINE	Traço vertical
STACO	Primeira coluna de listas exibida
STARO	Linha superior exibida
DATAR	Marcação para entrada em um campo de tela
HOST	Nome do servidor de aplicação atual
DATLO	Data local do usuário atual
TIMLO	Hora local do usuário atual
ZONLO	Fuso horário do usuário atual

49

Linguagem ABAP

Operadores Lógicos

• Operadores simples

Igual	EQ =
Diferente	NE <> ><
Maior que	GT >
Maior ou igual a	GE >=
Menor que	LT
Menor ou igual a	LE
Intervalo de valores	BETWEEN val1 AND val2
É inicial	IS INICIAL
	NOT val1 {operador}

51

Operadores Lógicos

• Operadores para string

CONTAINS ONLY	str1 CO str2
CONTAINS NOT ONLY	str1 CN str2
CONTAINS ANY	str1 CA str2
CONTAINS NOT ANY	str1 NA str2
CONTAINS STRING	str1 CS str2
CONTAINS NO STRING	str1 NS str2
COVERS PATTERN	str1 CP str2
NOT PATTERN	str1 NP str2

52

Instruções de Controle

- Condição IF-ELSE-ENDIF

```
IF condição.
    BLOCO 1
ELSE.           ELSEIF condição.
    BLOCO 2
ENDIF.
```

- Condição CASE

```
CASE variável.
    WHEN valor1.
        BLOCO 1
    WHEN valor2.
        BLOCO 2
    WHEN OTHERS.
        BLOCO n
ENDCASE.
```

53

Instruções de Controle

- Loops

```
DO n TIMES.
    BLOCO
ENDDO.
```

CHECK condição.

CONTINUE.

```
DO.
    IF condição.
        EXIT.
    ENDIF.
    BLOCO.
ENDDO.
```

```
WHILE condição.
    BLOCO.
ENDWHILE.
```

54

Linguagem ABAP

Open SQL

Acesso ao Banco de Dados

- O R/3 é independente do sistema de banco de dados apesar das diferenças de sintaxes dos vários banco de dados.
- Cada work-process no servidor de aplicação tem uma interface de banco de dados.
- A interface do banco de dados converte todas as requisições do banco de dados do R/3 para os comandos SQL do banco de dados.

Acesso ao Banco de Dados

- Há dois métodos de acesso ao banco de dados através dos programas ABAP:
 - Open SQL
 - As declarações Open SQL são subconjuntos do padrão SQL que é totalmente integrado ao ABAP. Eles permitem acessar os dados independentemente do sistema de banco de dados que o R/3 esteja utilizando. Eles permitem ler os dados (Select) e alterar os dados (INSERT, UPDATE, DELETE)
 - SQL Nativo

57

Acesso ao Banco de Dados

- SQL Nativo
 - O SQL Nativo é vagamente integrado ao ABAP e permite o acesso a todas as funções contidos na interface de programação do respectivo sistema de base de dados.
 - Ao contrário do Open SQL, no SQL Nativo, as declarações não são verificadas e convertidas. Elas são enviadas diretamente para o sistema de banco de dados.

58

Open SQL

- O Open SQL tem os seguintes comandos:

Comando	Função
SELECT	Seleciona os dados das tabelas do banco de dados
INSERT	Adiciona registros em uma tabela do banco de dados
UPDATE	Altera o conteúdo de um registro da tabela do banco de dados
MODIFY	Insere um novo registro ou altera o conteúdo de um registro existente
DELETE	Exclui registros da tabela do banco de dados
OPEN CURSOR, FETCH, CLOSE CURSOR	Lê os registros de uma tabela do banco de dados utilizando o cursor

59

Open SQL

- Código de Retorno

- Todos os comandos do Open SQL retornam os valores para os seguintes campos de sistema:

- SY-SUBRC

- Após cada comando SQL, o campo de sistema SY-SUBRC recebe o valor 0 se a operação foi realizada com sucesso, caso contrário, o valor retornado será diferente de 0.

- SY-DBCNT

- Após a execução do comando SQL, o campo do sistema SY-DBCNT contém o número dos registros processados.

60

Open SQL - SELECT

- O SELECT é o comando utilizado para ler os registros das tabelas do banco de dados.
- A sintaxe do comando SELECT é:

```
SELECT <resultado>
INTO <destino>
FROM <origem>
[ WHERE <condição>]
[GROUP BY <campos>]
[HAVING <condição>]
[ ORDER BY <campos>]
```

61

Open SQL - SELECT

- **SELECT <resultado>**
 - A cláusula select define a estrutura dos dados a serem lidos, ou seja, um ou vários registros, os campos e se registros idênticos serão aceitos ou não.

Exemplo:

```
SELECT SINGLE matnr mtart aurt FROM ...
SELECT DISTINCT matnr mtart FROM ...
```

62

Open SQL - SELECT

- Os dados também podem ser agregados por uma coluna da tabela, através das seguintes expressões:
 - MAX: retorna o valor máximo da coluna
 - MIN: retorna o valor mínimo da coluna
 - AVG: retorna o valor médio da coluna
 - COUNT: conta os valores ou linhas:
 - COUNT (DISTINCT <campo>): retorna o número de diferentes valores das colunas
 - COUNT(*) : retorna o número total de registros da seleção

Exemplo:

```
SELECT MAX( AUART ) FROM...
```

63

Open SQL - SELECT

- INTO <destino>
 - A cláusula INTO define a área de destino nos quais o resultado do select é transferido.
 - Os valores podem ser transferidos para variáveis, work-areas e tabelas internas, desde que as variáveis de destino sejam compatíveis ou convertidos com os dados selecionados.

64

Open SQL - SELECT

Inserir em variáveis

```
DATA: v_matnr      TYPE mara-matnr,
      v_auart      TYPE mara-auart,
      v_menge      TYPE mara-menge.

SELECT matnr auart menge FROM mara
  INTO ( v_mara, v_auart, v_menge)...
```

Inserir em work-area

```
DATA: BEGIN OF wa_mara,
      v_matnr      TYPE mara-matnr,
      v_auart      TYPE mara-auart,
      v_menge      TYPE mara-menge.
END OF wa_mara.
```

```
SELECT matnr auart menge FROM mara
  INTO wa_mara
```

Inserir em work-area baseado em tabela

```
DATA: wa_mara TYPE mara.

SELECT matnr auart menge FROM mara
  INTO CORRESPONDING FIELDS OF wa_mara
```

Inserir em tabela interna

```
DATA: BEGIN OF wa_mara OCCURS 0,
      v_matnr      TYPE mara-matnr,
      v_auart      TYPE mara-auart,
      v_menge      TYPE mara-menge.
END OF wa_mara.
```

```
SELECT matnr auart menge FROM mara
  INTO TABLE ti_mara
```

Inserir em tabela interna baseado em tabela

```
DATA: wa_mara TYPE mara OCCURS 0.
```

```
SELECT matnr auart menge FROM mara
  INTO CORRESPONDING FIELDS OF TABLE ti_mara
```

65

OPEN SQL - SELECT

- **FROM <origem>**
 - A cláusula FROM especifica a tabela em que os dados especificados na cláusula select serão lidos.
 - As visões de banco de dados também pode ser utilizadas na cláusula FROM
 - Podem ser especificados uma tabelas ou mais de uma tabela, com o uso das cláusulas INNER JOIN ou OUTER JOIN.
 - Select de uma tabela

SELECT matnr auart menge FROM mara ...

66

Open SQL - SELECT

- Select em mais de uma tabela

```
SELECT ...  
...  
FROM <tab> [ INNER ] JOIN <dbtab> [ AS <alias> ] ON <cond>  
<options>
```

Exemplo:

```
SELECT m~matnr t~maktx  
  FROM mara as m INNER JOIN makt as t  
    ON m~matnr = t~matnr  
  INTO wa_mat_desc  
 WHERE m~matnr = '08D8'  
   AND t~spras = SY-LANGU.
```

67

Open SQL - SELECT

- Restringir o número de registros
 - Para restringir o número absoluto de registros incluídos na seleção, use a cláusula abaixo:

```
SELECT ... FROM <tabela> UP TO <n> ROWS
```

Onde <n> é um valor inteiro positivo e o sistema lê o máximo de <n> registros.

68

Open SQL - SELECT

- WHERE <condição>

- A cláusula WHERE restringe o número de registros selecionados especificando as condições que devem ser atingidas.

```
SELECT ...
FROM ...
INTO ...
WHERE campo1 = '8076' AND
      campo2 IN ( 'RTY', 'ERP', 'MMP' ) OR "Lista de valores"
      campo3 BETWEEN '001' AND '100' AND "Intervalos de valores"
      campo 4 LIKE (ABC%) AND "String de comparação"
      campo 5 IS NULL
```

69

Open SQL - Select

- WHERE - Condição tabular

- A cláusula where tem uma variante especial que permite derivar as condições das linhas e colunas de uma tabela interna:

```
SELECT ... FOR ALL ENTRIES IN <itab> WHERE <condição>
```

- Ao especificar um campo da tabela interna como um operando em uma condição, todas as linhas da tabela interna são lidas. A comparação então é executada para cada linha da tabela interna. Para cada registro, o sistema seleciona os registros do banco de dados que satisfaçam a seleção.
 - As linhas duplicadas são automaticamente eliminadas do resultado.
 - Se a tabela interna estiver vazia, todas as entradas da tabela do banco de dados são lida.

70

Open SQL - Insert

- O comando INSERT adiciona registro(s) na tabela interna
- Os registros podem ser adicionados individualmente ou coletivamente
- Inserindo um único registro:

```
INSERT INTO <DBTAB> VALUES <work-area> OU  
INSERT <DBTAB> FROM <work-area>.
```

- O conteúdo da work-area são adicionados à tabela do banco de dados <ITAB>. A work-area deve ter a mesma estrutura da tabela do banco de dados.

71

Open SQL - Insert

- Inserindo vários registros:

```
INSERT <DBTAB> FROM TABLE <ITAB>[ACCEPTING DUPLICATE KEYS]
```

- O conteúdo da tabela interna <ITAB> é adicionado na tabela do banco de dados <DBTAB> em uma única operação. A tabela interna deve ser do mesmo tipo da tabela do banco de dados.
- Se a tabela possuir campos chaves duplicados, um erro de execução é apresentado. Se o parâmetro ACCEPTING DUPLICATE KEYS for utilizado, não é apresentado o erro de execução, mas o campo de sistema SY-SUBRC receberá o valor 4.

72

Open SQL - Insert

Exemplo

Insere um registro individual na tabela SPFLI:

```
TABLES SPFLI.  
DATA WA TYPE SPFLI.  
WA-CARRID = 'LH'.  
WA-CITYFROM = 'WASHINGTON'.  
...  
INSERT INTO SPFLI VALUES WA.  
  
WA-CARRID = 'UA'.  
WA-CITYFROM = 'LONDON'.  
...  
INSERT SPFLI FROM WA.
```

73

Open SQL - Insert

Exemplo

Insere vários registros na tabela SPFLI:

```
DATA: ITAB TYPE HASHED TABLE OF SPFLI  
WITH UNIQUE KEY CARRID CONNID,  
WA LIKE LINE OF ITAB.  
WA-CARRID = 'UA'. WA-CONNID = '0011'. WA-CITYFROM = ...  
INSERT WA INTO TABLE ITAB.  
WA-CARRID = 'LH'. WA-CONNID = '1245'. WA-CITYFROM = ...  
INSERT WA INTO TABLE ITAB.  
WA-CARRID = 'AA'. WA-CONNID = '4574'. WA-CITYFROM = ...  
INSERT WA INTO TABLE ITAB.  
...  
INSERT SPFLI FROM TABLE ITAB ACCEPTING DUPLICATE KEYS.
```

74

Open SQL - Update

- O comando UPDATE é utilizado para alterar o conteúdo da tabela do banco de dados
- Modificando registros individuais
 - Para modificar os registros individualmente de uma tabela do banco de dados através de uma work-area, utilize o seguinte comando:

```
UPDATE <dbtab> FROM <work-area>
```

- Se a tabela do banco de dados conter um registro com a mesma chave, o registro é atualizado. Caso contrário, o registro não é alterado e o SY-SUBRC recebe o valor 4.

75

Open SQL - Update

- Modificando vários registros
 - Para modificar vários registros de uma tabela do banco de dados, o comando é:

```
UPDATE <DBTAB> FROM TABLE <ITAB>
```

- Se o sistema não puder modificar um registro devido a não existência de um registro com chave especificada, o processo não é finalizado, mas o processamento continua no próximo registro.

76

Open Sql - Update

- Modificando os registros coluna por coluna
 - Para modificar determinada colunas na tabela do banco de dados, utilize o comando:
- UPDATE <DBTAB> SET <set1> <set2> ... WHERE <COND>.
- A condição WHERE determina as linhas que serão alteradas. Se a cláusula WHERE não for especificada, todos os registros são atualizados. A expressão <set1> é o parâmetro que determina as colunas que serão atualizadas:
 - <s1> = <f>, sendo que <f> pode ser um objeto de dados ou uma coluna da tabela do banco de dados.

77

Open SQL - Update

- Exemplo de Update

Atualiza o conteúdo da coluna PLANETYPE com A310 e diminui o valor da coluna PRICE por 100 para cada entrada na tabela SFLIGHT onde CARRID seja LH e CONNID 0402.

```
UPDATE SFLIGHT SET PLANETYPE = 'A310'
                  PRICE = PRICE - '100.00'
                  WHERE CARRID = 'LH' AND CONNID = '0402'.
```

Os campos CARRID e CONNID são os campos chaves da tabela SPLFI. Todos os registros que tiverem como chave os valores AA e 0064, serão atualizados.

```
DATA WA TYPE SPFLI.
MOVE 'AA' TO WA-CARRID.
MOVE '0064' TO WA-CONNID.
MOVE 'WASHINGTON' TO WA-CITYFROM.
...
UPDATE SPFLI FROM WA.
```

78

Open SQL - Delete

- O comando DELETE é utilizado para excluir os registros da tabela do banco de dados.
- Para excluir as linhas utilizando condições utilize a seguinte sintaxe:

```
DELETE FROM <dbtab> WHERE <cond>
```

- Todas as linhas que atenderem a condição serão removidas.
- Se uma tabela interna vazia for especificada na cláusula where, todos os registros da tabela do banco de dados serão removidos.

79

Open SQL - Delete

- Ao invés de utilizar a cláusula WHERE, é possível selecionar os registros para exclusão utilizando o conteúdo de uma work-area.

```
DELETE <dbtab> FROM <work-area>.
```

- Isto exclui o registro com a mesma chave primária da work-area.

80

Open SQL - Delete

- Também é possível excluir várias linhas coletivamente utilizando uma tabela interna.

```
DELETE <dbtab> FROM TABLE <itab>
```

- Esta sintaxe exclui todos os registros da tabela do banco de dados que possuem as mesmas chaves primárias da tabela interna.
- Se o sistema não puder excluir um registro, pois a chave especificada não existe, a operação não é cancelada, mas o processo continua a partir do próximo registro.

81

Open SQL - Delete

- Exemplo de Delete

Todas os registros da tabela SFLIGHT são excluídos onde PLANETYPE tenha o valor A310 e CARRID tenha o valor LH.

```
DELETE FROM SFLIGHT WHERE PLANETYPE = 'A310' AND CARRID = 'LH'.
```

Os campos CARRID e CONNID são campos chave da tabela SPFLI. Os registros com as chaves primárias AA e 0064 são excluídas.

```
TABLES SPFLI.
```

```
DATA: BEGIN OF WA,  
      CARRID TYPE SPFLI-CARRID,  
      CONNID TYPE SPFLI-CONNID,  
      END OF WA.
```

```
MOVE 'AA' TO WA-CARRID.  
MOVE '0064' TO WA-CONNID.  
DELETE SPFLI FROM WA.
```

82

Open SQL - Delete

Uma tabela interna hashed é definida com a estrutura das chaves primárias da tabela SPFLI. Depois do preenchimento da tabela interna, os registros da tabela do banco de dados serão excluídos de acordo com os valores das chaves internas da tabela interna.

```
DATA: BEGIN OF WA,  
      CARRID TYPE SPFLI-CARRID,  
      CONNID TYPE SPFLI-CONNID,  
      END OF WA,  
      ITAB LIKE HASHED TABLE OF WA  
      WITH UNIQUE KEY CARRID CONNID.  
  
WA-CARRID = 'UA'. WA-CONNID = '0011'.  
INSERT WA INTO TABLE ITAB.  
WA-CARRID = 'LH'. WA-CONNID = '1245'.  
INSERT WA INTO TABLE ITAB.  
WA-CARRID = 'AA'. WA-CONNID = '4574'.  
INSERT WA INTO TABLE ITAB.  
...  
DELETE SPFLI FROM TABLE ITAB.
```

83

Open SQL - Considerações

- Os comandos INSERT, UPDATE e DELETE trabalham com work-area e tabela interna.
- As work-areas e tabelas internas devem ter a mesma estrutura da tabela do banco de dados que será manipulada.
- Se a operação foi realizada com sucesso, o valor do campo de sistema SY-SUBRC será 0, caso contrário será 4.
- O campo de sistema SY-DBCNT irá conter o número de registros afetados durante a operação.

84

Linguagem ABAP

Operações em Tabela Interna

Tabela Interna

- A tabela interna é utilizada para armazenar um ou mais registros em memória e pode ser acessado e manipulado de forma similar a uma tabela de banco de dados.
- Os dados da tabela interna podem ser inseridos, removidos, modificados, ordenados, comparados e lidos.
- O comando SELECT não pode ser utilizado para obter os registros de tabela interna.

Tabela Interna - Inicializar

- Como qualquer objeto de dados, a tabela interna pode ser inicializada através dos comandos:
 - **CLEAR <ITAB>**
 - Este comando restaura a tabela interna ao estado ao qual estava imediatamente após a declaração. Isto significa que a tabela não contém linhas. Entretanto a memória utilizada pela tabela interna continua alocada.
 - Se a tabela interna possuir cabeçalho, a tabela e cabeçalho possuem o mesmo nome e para apagar o conteúdo da tabela, é necessário utilizar as chaves ([]) após o nome da tabela interna, caso contrário, somente o conteúdo cabeçalho será apagado.

87

Tabela Interna - Inicializar

- O comando REFRESH realiza o mesmo procedimento do comando CLEAR[], ou seja, reinicializa a tabela interna e mantém a memória utilizada pela tabela interna alocada.
- O comando FREE reinicializa a tabela interna e libera a memória alocada.

88

Tabela Interna - Inicializar

▫ Exemplo:

```
DATA: BEGIN OF LINE,  
      COL1,  
      COL2,  
    END OF LINE.
```

```
DATA ITAB LIKE LINE OCCURS 0 WITH HEADER LINE.
```

```
CLEAR ITAB. "Limpa o cabeçalho da tabela interna  
CLEAR ITAB[ ]. "Limpa o conteúdo da tabela interna  
REFRESH ITAB. "Limpa o conteúdo da tabela interna  
FREE ITAB. "Limpa o conteúdo da tabela interna
```

89

Tabela Interna - Comparação

- As tabelas internas podem ser comparadas.
... <itab> <operador> <itab2> ...
- O operador a ser utilizado pode ser os operadores ABAP (EQ, =, NE, <>, ><, GE, >=, LE, <=, GT, >, LT, <).
- O primeiro critério de comparação entre tabelas internas é o número de linhas que eles contém.
- Se as tabelas internas comparadas contém o mesmo número de linhas, eles são comparados linha por linha e componente por componente.

90

Tabela Interna - Comparação

- Se o componente for outra tabela interna, eles são comparados recursivamente.
- Se a tabela interna estiver sendo comparada diferente da igualdade, a comparação é finalizada quando o primeiro par de componentes forem diferentes e retorna o resultado correspondente.

91

Tabela Interna - Comparação

- Exemplo

```

DATA: BEGIN OF LINE,
      COL1 TYPE I,
      COL2 TYPE I,
    END OF LINE.

DATA: ITAB LIKE TABLE OF LINE,
      JTAB LIKE TABLE OF LINE.

DO 3 TIMES.
  LINE-COL1 = SY-INDEX.
  LINE-COL2 = SY-INDEX ** 2.
  APPEND LINE TO ITAB.
ENDDO.

MOVE ITAB TO JTAB.

LINE-COL1 = 10. LINE-COL2 = 20.
APPEND LINE TO ITAB.

IF ITAB GT JTAB.
  WRITE / 'ITAB GT JTAB'.
ENDIF.

APPEND LINE TO JTAB.

IF ITAB EQ JTAB.
  WRITE / 'ITAB EQ JTAB'.
ENDIF.

LINE-COL1 = 30. LINE-COL2 = 80.
APPEND LINE TO ITAB.

IF JTAB LE ITAB.
  WRITE / 'JTAB LE ITAB'.
ENDIF.

LINE-COL1 = 50. LINE-COL2 = 60.
APPEND LINE TO JTAB.

IF ITAB NE JTAB.
  WRITE / 'ITAB NE JTAB'.
ENDIF.

IF ITAB LT JTAB.
  WRITE / 'ITAB LT JTAB'.
ENDIF.

```

92

Tabela Interna - Ordenação

- As tabelas standard ou hashed podem ser ordenadas.
- O comando para ordenação é:

```
SORT <itab> [ASCENDING|DESCENDING] [AS TEXT].
```

- O comando ordena a tabela interna <itab> em ordem ascendente pela sua chave.
- A direção da ordem pode ser especificada através das opções ASCENDING e DESCENDING, sendo que a primeira opção é a padrão.
- Quanto maior a tabela, maior será o tempo para ordená-la.

93

Tabela Interna - Ordenação

- A tabela interna pode ser ordenada pelo componente desejado através do comando abaixo:

```
SORT <itab> [ASCENDING|DESCENDING] [AS TEXT]
      BY <f1> [ASCENDING|DESCENDING] [AS TEXT]
      ...
      <fn> [ASCENDING|DESCENDING] [AS TEXT].
```

- A tabela interna é ordenada pelos componentes <f1> ... <fn>, ao invés da chave da tabela. O número de campos a serem ordenados está limitado a 250.

94

Tabela Interna - Ordenação

- **Ordenação Alfabética**

- Bem como as opções ASCENDING e DESCENDING, também pode ser especificado que os campos a serem ordenados devem ser ordenados alfabeticamente, o qual afeta o método de ordenação em strings.

95

Tabela Interna - Determinar Atributos

- Para determinar os atributos de uma tabela interna em tempo de execução que não estava disponível estaticamente, utilize o comando:

```
DESCRIBE TABLE <itab> [LINES <l>] [OCCURS <n>] [KIND <k>].
```

- O uso do parâmetro LINES retorna o número de linhas preenchidas na variável <l>.
- O uso do parâmetro OCCURS, retorna o tamanho inicial da tabela interna para a variável <n>.
- O uso do parâmetro KIND, retorna o tipo de tabela interna para a variável <k>, onde

96

Tabela Interna - Determinar Atributos

- O uso do parâmetro KIND, retorna o tipo de tabela interna para a variável <k>, onde:
 - T = Tabela standard
 - S = Tabela sorted
 - H = Tabela hashed

97

Tabela Interna - Determinar Atributos

Exemplo:

```
DATA: BEGIN OF LINE,  
      COL1 TYPE I,  
      COL2 TYPE I,  
    END OF LINE.
```

```
DATA ITAB LIKE HASHED TABLE OF LINE  
  WITH UNIQUE KEY COL1 INITIAL SIZE 10.
```

```
DATA: LIN TYPE I,  
     INI TYPE I,  
      KND TYPE C.
```

```
DESCRIBE TABLE ITAB LINES LIN OCCURS INI KIND KND.  
WRITE: / LIN, INI, KND.
```

Saída	0	10	H
-------	---	----	---

98

Tabela Interna - Determinar Atributos

Exemplo:

Saída			
1000	10		H

```

DO 1000 TIMES.
  LINE-COL1 = SY-INDEX.
  LINE-COL2 = SY-INDEX ** 2.
  INSERT LINE INTO TABLE ITAB.
ENDDO.

DESCRIBE TABLE ITAB LINES LIN OCCURSINI KIND
KND.
WRITE: / LIN,INI,KND.
```

99

Tabela Interna - Inserir linha individual

- Para inserir uma linha na tabela interna, a sintaxe é:

```
INSERT <line> TO <itab>.
```

- <line> pode ser uma work-area ou a expressão INITIAL LINE.
- A work area deve ser compatível com a tabela interna e a expressão INITIAL LINE insere um registro em branco na tabela interna.

100

Tabela Interna - Inserir linha individual

- **Regras**

- Em uma tabela interna standard, a linha é inserida no final da tabela interna
- Em uma tabela sorted, a linha é inserida de acordo com a chave da tabela. Se a chave não é única, o registro duplicado é inserido acima da entrada existente com a mesma chave.
- Em uma tabela hashed, a linha é inserida de acordo com a chave da tabela.

101

Tabela Interna - Inserir várias linhas

- **Para inserir várias linhas na tabela interna, a sintaxe é:**

```
INSERT LINES OF <itab1> [FROM <n1>] [TO <n2>] INTO TABLE <itab2>.
```

- **<itab1> e <itab2> são tabelas com tipos de linhas compatíveis. O sistema insere as linhas da tabela <itab1> um por um na tabela <itab2> usando as mesmas regras da inserção individual.**
- **Se <itab1> é uma tabela com índice (standard e sorted), pode ser especificado a primeira e a última linha da tabela que deverá ser inserida em <n1> e <n2>.**

102

Tabela Interna - Inserir Linhas

Exemplo: Preencher a tabela interna com seis entradas

```

DATA: BEGIN OF LINE,
      LAND(3) TYPE C,
      NAME(10) TYPE C,
      AGE TYPE I,
      WEIGHT TYPE P DECIMALS 2,
   END OF LINE.

DATA ITAB LIKE SORTED TABLE OF LINE
      WITH NON-UNIQUE KEY LAND NAME
      AGE WEIGHT.

LINE-LAND = 'G'. LINE-NAME = 'Hans'.
LINE-AGE = 20. LINE-WEIGHT = '80.00'.
INSERT LINE INTO TABLE ITAB.

LINE-LAND = 'USA'. LINE-NAME = 'Nancy'.
LINE-AGE = 35. LINE-WEIGHT = '45.00'.
INSERT LINE INTO TABLE ITAB.

LINE-LAND = 'USA'. LINE-NAME = 'Howard'.
LINE-AGE = 40. LINE-WEIGHT = '95.00'.
INSERT LINE INTO TABLE ITAB.

```

```

LINE-LAND = 'GB'. LINE-NAME = 'Jenny'.
LINE-AGE = 18. LINE-WEIGHT = '50.00'.
INSERT LINE INTO TABLE ITAB.

```

```

LINE-LAND = 'F'. LINE-NAME = 'Michele'.
LINE-AGE = 30. LINE-WEIGHT = '60.00'.
INSERT LINE INTO TABLE ITAB.

```

```

LINE-LAND = 'G'. LINE-NAME = 'Karl'.
LINE-AGE = 60. LINE-WEIGHT = '75.00'.
INSERT LINE INTO TABLE ITAB.

```

```

LOOP AT ITAB INTO LINE.
  WRITE: / LINE-LAND, LINE-NAME, LINE-AGE, LINE-
        WEIGHT.
ENDLOOP.

```

103

Tabela Interna - Inserir Linhas

Resultado

The screenshot shows a SAP application window with a toolbar at the top and a table in the main area. The table has columns labeled 'Programa', 'Nome', 'Idade', and 'Peso'. The data is as follows:

Programa	Nome	Idade	Peso
F	Michele	30	60,00
G	Hans	20	80,00
G	Karl	60	75,00
GB	Jenny	18	50,00
USA	Howard	40	95,00
USA	Nancy	35	45,00

104

Tabela Interna - Inserir Linhas

Exemplo: Duas tabelas internas são criadas com os mesmos componentes, mas com tipo de tabelas diferente. Cada um é preenchido com três linhas.

```

DATA: BEGIN OF LINE,
      COL1 TYPE I,
      COL2 TYPE I,
    END OF LINE.

DATA: ITAB LIKE STANDARD TABLE OF LINE,
      JTAB LIKE SORTED TABLE OF LINE
      WITH NON-UNIQUE KEY COL1 COL2.

DO 3 TIMES.
  LINE-COL1 = SY-INDEX. LINE-COL2 = SY-INDEX ** 2.
  APPEND LINE TO ITAB.
  LINE-COL1 = SY-INDEX. LINE-COL2 = SY-INDEX ** 3.
  APPEND LINE TO JTAB.
ENDDO.

INSERT LINES OF ITAB INTO TABLE JTAB.

LOOP AT JTAB INTO LINE.
  WRITE: / SY-TABIX, LINE-COL1, LINE-COL2.
ENDLOOP.

```

105

Tabela Interna - Inserir Linhas

Resultado

COL1	COL2	COL3
1	1	1
2	1	1
3	2	4
4	2	8
5	3	9
6	3	27

106

Tabela Interna - Inserir linhas summarizadas

- O comando abaixo permite inserir linhas somadas na tabela interna:

```
COLLECT <wa> INTO <itab>.
```

- Todos os campos que não fazem parte da chave da tabela deve ser do tipo numérico (F, I ou P).
- O comando COLLECT deve ser utilizado para criar tabelas summarizadas.

107

Tabela Interna - Inserir linhas summarizadas

- Quando uma linha é inserida, o sistema verifica se há uma entrada na tabela que combine com a chave. Se não houver uma entrada correspondente na tabela, o comando COLLECT tem o mesmo efeito do comando INSERT. Se uma entrada existir na tabela, o comando COLLECT não cria uma nova linha, mas adiciona o conteúdo dos campos numéricos da work-area no conteúdo dos campos numéricos da linha existente.

108

Tabela Interna - Inserir linhas summarizadas

Exemplo: Os dois primeiros comandos COLLECT trabalham como o comando INSERT. No terceiro comando COLLECT, a primeira linha da ITAB é modificada.

```
DATA: BEGIN OF LINE,
      COL1(3) TYPE C,
      COL2(2) TYPE N,
      COL3 TYPE I,
    END OF LINE.
```

```
DATA ITAB LIKE SORTED TABLE OF LINE
  WITH NON-UNIQUE KEY COL1 COL2.
```

```
LINE-COL1 = 'abc'. LINE-COL2 = '12'.
LINE-COL3 = 3.
COLLECT LINE INTO ITAB.
WRITE / SY-TABIX.
```

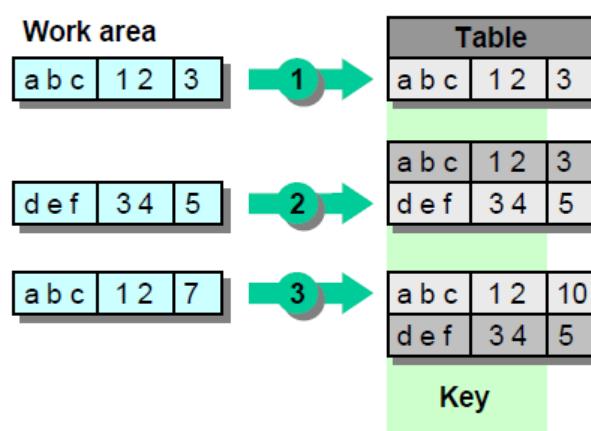
```
LINE-COL1 = 'def'. LINE-COL2 = '34'.
LINE-COL3 = 5.
COLLECT LINE INTO ITAB.
WRITE / SY-TABIX.
```

```
LINE-COL1 = 'abc'. LINE-COL2 = '12'.
LINE-COL3 = 7.
COLLECT LINE INTO ITAB.
WRITE / SY-TABIX.
```

```
LOOP AT ITAB INTO LINE.
  WRITE: / LINE-COL1, LINE-COL2, LINE-COL3.
ENDLOOP.
```

109

Tabela Interna - Inserir linhas summarizadas



110

Curso Avançado de ABAP

Dicionário de Dados

Objetivo



- O dicionário de dados permite a descrição centralizada de todos os dados utilizados no sistema sem redundância.
- Informações novas ou modificadas são fornecidas automaticamente para todos os componentes do sistema, garantindo assim, a integridade, consistência e segurança dos dados.

Características do Dicionário de Dados

- Gerência centralizada das definições de dados utilizados no sistema
- Suporte as definições dos tipo de dados de usuários (elemento de dados, estruturas e tabelas)
- Definição de estrutura de objetos de banco de dados (tabelas, índices e visão), os quais são criados automaticamente no banco de dados

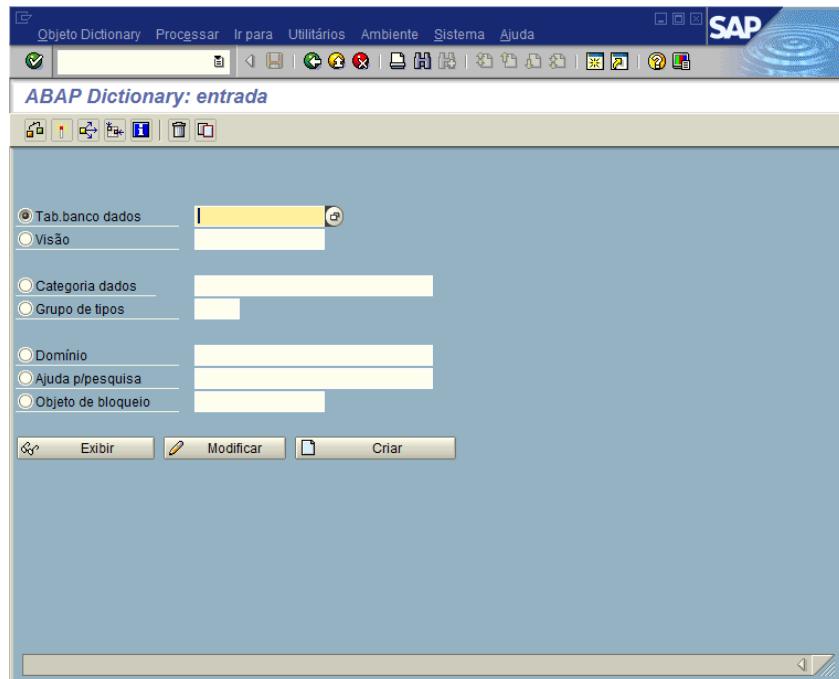
113

Características do Dicionário de Dados

- Os principais tipos de objetos do dicionário de dados são:
 - Tabelas
 - Visões
 - Tipos (elemento de dados, estruturas e categoria de tabelas)
 - Domínios
 - Ajuda de Pesquisa
 - Objetos de Bloqueio

114

Transação SE11



115

TABELAS

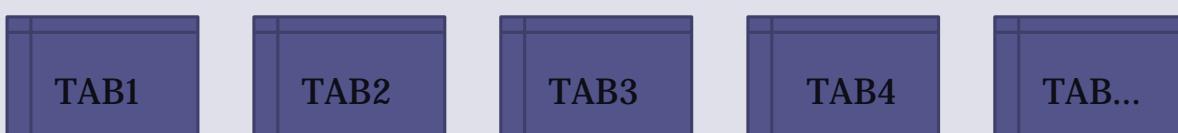
Tabelas

- As tabelas podem ser definidas independentemente do banco de dados no dicionário de dados.
- Quando a tabela é ativada, uma definição física da tabela é criada no banco de dados de acordo com a tabela criada no dicionário de dados.
- As definições das tabelas são convertidas do dicionário de dados para a definição de um determinado banco de dados.

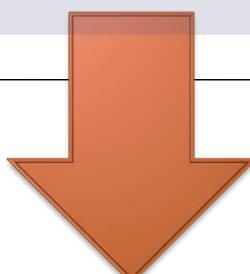
117

Ativação de tabelas

Dicionário de Dados



Banco de Dados



118

Tipos de tabelas

- Existem três tipos de tabelas no dicionário de dados. São elas:
 - Tabelas Transparentes
 - Para cada tabela transparente no dicionário de dados há uma tabela correspondente no banco de dados com o mesmo nome, campos e definições.
 - Tabelas Pool
 - Para cada tabela no banco de dados, há várias tabelas no dicionário de dados. O R/3 utiliza o pool de tabelas uma grande quantidade (dezenas a milhares) de pequenas tabelas (de 10 a 100 registros cada). As tabelas pool reduzem a quantidade de recursos necessários do banco de dados quando várias tabelas pequenas necessitam ser acessadas ao mesmo tempo.
 - Tabelas Cluster
 - As tabelas cluster são similares as tabelas pool. Eles são utilizados para armazenar poucas tabelas (de 2 a 10) mas com grande quantidade de dados. A vantagem de utilizar tabelas cluster é no caso de acessar dados de múltiplas tabelas simultaneamente, reduzindo o número de acesso ao banco de dados aumentando a performance.

119

Classe de Entrega

- A classe de entrega controla o transporte de dados da tabela, no caso de instalação, mudança de release, cópia de mandante, e no caso de transporte entre sistemas de cliente. A classe de entrega também é considerada na Atualização ampliada de tabelas.

120

Classe de Entrega

- Existem as seguintes classes de entrega:
 - **A**
 - Tabela de aplicação (dados mestre e de movimento)
 - **C**
 - Tabela de cliente, os dados são atualizados exclusivamente pelo cliente.
 - **L**
 - Tabela para arquivar dados temporários.
 - **G**
 - Tabela de cliente, a SAP pode inserir registros novos, mas não pode sobregravar ou eliminar aqueles que já existem. É necessário que o conjunto de nomes de cliente seja definido na tabela TRESC (utilizar o report RDKOR54).
 - **E**
 - Tabela de sistema com conjuntos de nomes próprios para entradas de cliente. É necessário que o conjunto de nomes de cliente seja definido na tabela TRESC (utilizar o report RDKOR54).
 - **S**
 - Tabela de sistema, as modificações de dados têm o status de modificações de programa.
 - **W**
 - Tabela de sistema (por exemplo, tabela do ambiente de desenvolvimento), cujos dados são transportados por objetos de transporte próprios (por exemplo, R3TR PROG, R3TR TABL, etc).

121

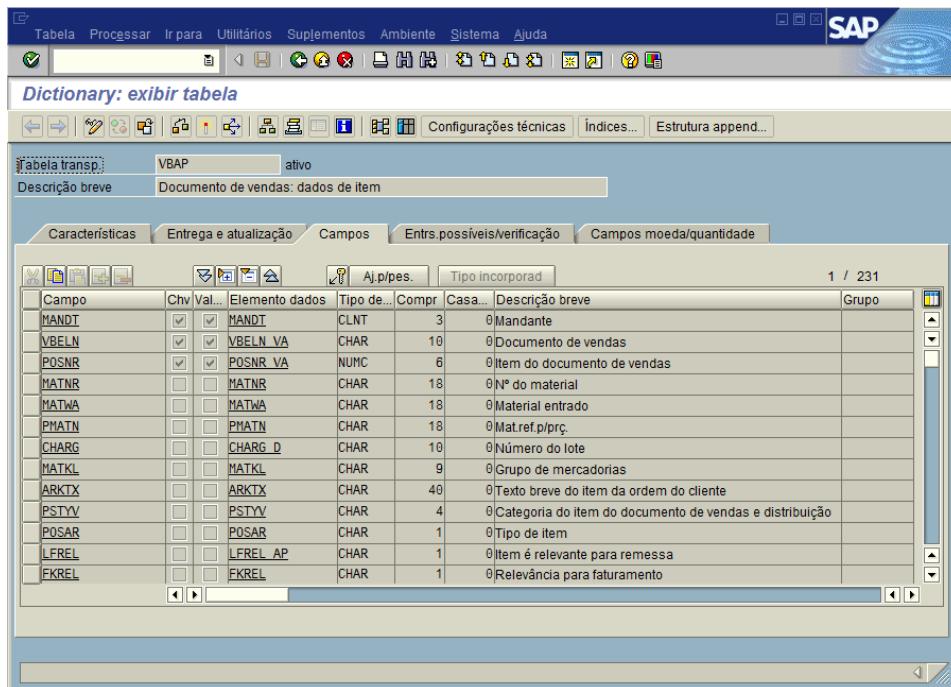
Definição de tabelas

- A definição de tabelas possuem os seguintes componentes:
 - Campos da tabela
 - Chave Externa
 - Configurações Técnicas
 - Índice

122

Campos da Tabela

- Área de definição dos campos da tabela



123

Campos da Tabela - Referência

- Se houver algum campo do tipo moeda ou quantidade, é necessário informar a tabela e o campo de referência.
- A tabela de referência deve conter um campo no formato de chave de moeda (tipo de dado CUKY) ou unidade de medida (UNIT)
- Um campo só é atribuído ao campo de referência em tempo de execução. Ex. Se um campo contém um valor monetário, a moeda é determinada de acordo com o campo de referência.

124

Campo da tabela - Referência

The screenshot shows the SAP Dictionary interface for the table VBAP. The title bar says "Dictionary: exibir tabela". The table name is "VBAP" and its status is "ativo" (active). The description is "Documento de vendas: dados de item" (Sales document: item data). The "Campos" tab is selected. A table lists fields with their data types and descriptions:

Campo	Elem.dados	Catego...	Tabela referê...	Campo ref.	Descrição breve
UEPOS	UEPOS	NUMC			Item superior em estruturas de listas técnicas
GRPOS	GRPOS	NUMC			Item alternativo a este item
ABGRU	ABGRU_VA	CHAR			Motivo de recusa p/cotações e ordens
PRODH	PRODH_D	CHAR			Hierarquia de produtos
ZWERT	DZWERT	CURR	VBAP	WAERK	Valor fixado de contrato básico na moeda do documento
ZMENG	DZMENG	QUAN	VBAP	ZIEME	Qtd.prevista em UMV
ZIEME	DZIEME	UNIT			UM qtd.prevista
UMZIZ	UMZIZ	DEC			Numerador do fator de conversão qtd.fixada -> qtd.em de
UMZIN	UMZIN	DEC			Denominador do fator conversão qtd.fixada -> qtd.em de
MEINS	MEINS	UNIT			Unidade de medida básica
SMENG	SMENG	QUAN	VBAP	MEINS	Quantidade de escala em UM base
ABLFZ	ABLFZ	QUAN	VBAP	ZIEME	Qtd.arredondada para remessa
ABDAT	ABDAT	DATS			Data accordada para a quantidade acumulada ajustada

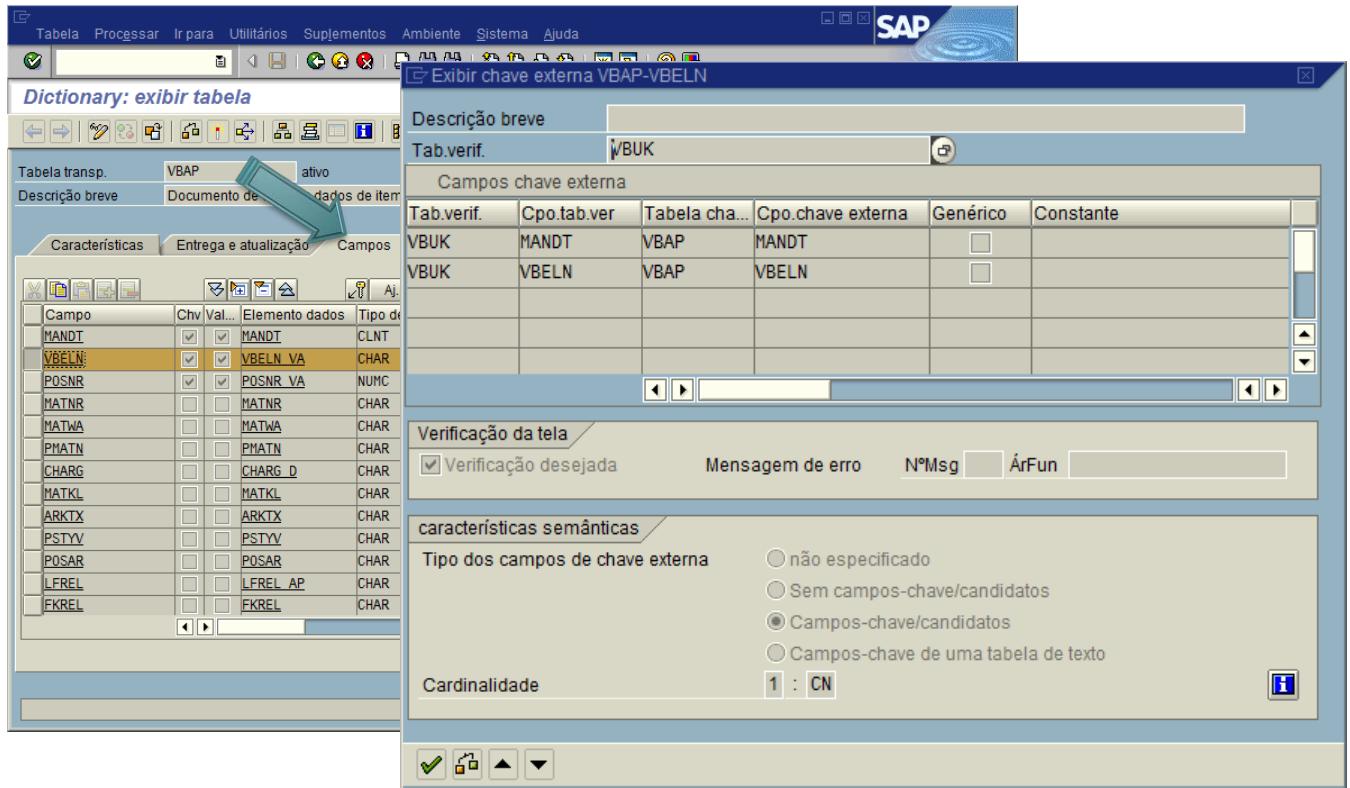
125

Chave Externa

- O relacionamento de tabelas no dicionário de dados é realizado através de chave externa
- Verificação de valores para campos de entrada
- Associar várias tabelas numa visão ou objetos de bloqueio

126

Chave Externa



127

Configurações Técnicas

- Define como a tabela será manuseada quando for criada no banco de dados, tais como o tipo de buffer da tabela e criar log de alteração de registros da tabela
- Os parâmetros da configuração técnica são:
 - Categoría de dados
 - Categoría de tamanho
 - Buffer
 - Log

128

Categoria de Dados

- A categoria de dados define o tablespace ou Dbspace do banco de dados quando ele é criado.
- As categorias de dados são:
 - APPLO (dados mestre)
 - São dados raramente atualizados como dados de um cliente (nome, telefone, endereço)
 - APPL1 (dados transacionais)
 - São dados freqüentemente atualizados. Um exemplo são os materiais de um depósito, os quais são alterados a cada nova ordem de compra
 - APPL2 (dados organizacionais)
 - São dados que são definidos quando o sistema é instalado e raramente modificado. Um exemplo é a tabela com o código dos países.
 - USR1 e USR2
 - São categorias para uso do cliente para testes de desenvolvimento.

129

Categoria de tamanho

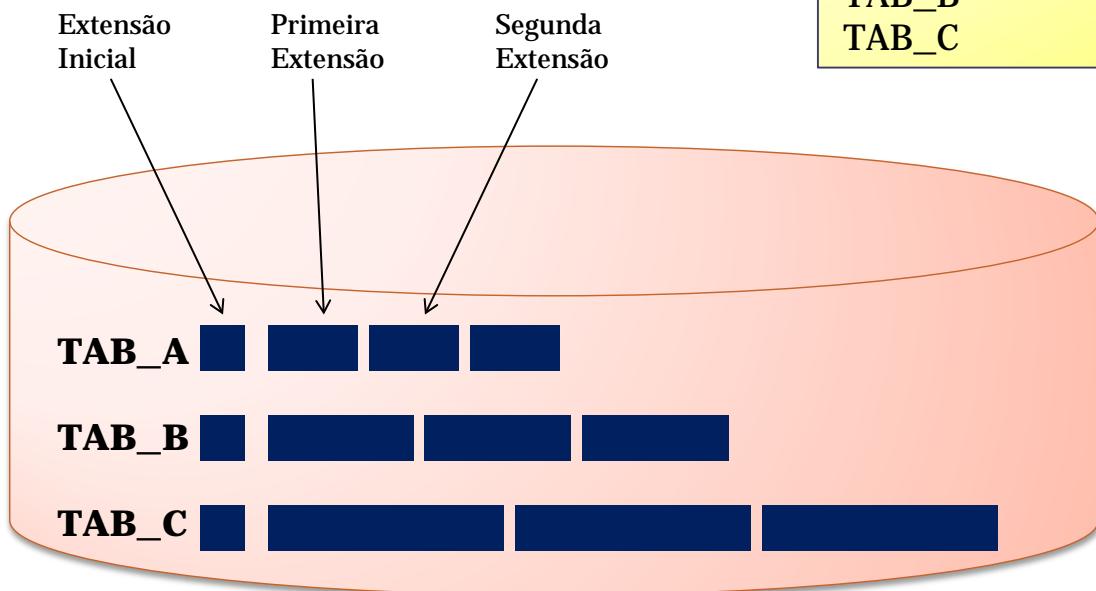
- O parâmetro categoria de tamanho define o espaço necessário esperado para a tabela no banco de dados.
- As categorias de tamanho variam de 0 a 4.
- Para cada categoria, é atribuído um tamanho fixo de memória no banco de dados, o qual depende do tipo do banco de dados utilizado

130

Categoria de tamanho

Configurações Técnicas

Categoria de Tamanho	
TAB_A	1
TAB_B	2
TAB_C	3



131

Buffer

- O tipo de buferização define quais os registros serão armazenados no buffer do servidor de aplicação quando o registro é acessado.
- Somente os campos chaves podem ser buferizados.
- Os tipos de buferização são:
 - Armazenar em buffer registros individuais
 - Armazenar em buffer áreas genéricas
 - Armazenamento em buffer completo

132

Buffer - Registros Individuais

- Com o armazenamento em buffer de registros individuais, somente os registros atualmente lidos são carregados no buffer.
- Esse tipo de buffer necessita de menos espaço de armazenamento
- Os custos administrativos são maiores, pois mais acesso ao banco de dados são necessários

133

Buffer - Registros Individuais

- Exemplo:
 - Se o tipo de armazenamento em buffer for individual, somente o registro carregado no buffer

```
SELECT SINGLE
  FROM MARD
 WHERE MATNR = '208014400'
   AND WERKS = 'M400'
   AND LGORT = 'MD40'.
```

MATNR	WERKS	LGORT
208014180	M400	MD40
208014180	M400	MD42
208014400	M400	MD40
208014400	M400	MD42
280806160	M400	MD40
280806160	M400	MD42
280808120	M400	MD40

Conteúdo do Buffer		
208014400	M400	MD40

134

Buffer - Áreas Genéricas

- Com a bufferização de áreas genéricas, todos os registros onde o critério de seleção seja genérico, são carregados no buffer.
- É necessário informar o número de campos chave para a área genérica.
- É necessário que o número entrado esteja entre 1 e o número de campos chave -1. Por exemplo, para uma tabela com 6 campos chave, só são admitidos valores entre 1 e 5.

135

Buffer - Áreas Genéricas

- No exemplo abaixo, o registro marcado é lido pelo programa. Se a tabela possuir buffer genérico, todos os registros lidos com a chave genérica (MATNR e WERKS) são gravados no buffer.

```
SELECT *
  FROM MARD
 WHERE MATNR = '280806150'
   AND WERKS = 'M400'.
```

Tabela transparente MARD

MATNR	WERKS	LGORT
208014180	M400	MD40
208014180	M400	MD42
208014400	M400	MD40
208014400	M400	MD42
280806160	M400	MD40
280806160	M400	MD42
280806160	M400	MD43

Conteúdo do Buffer

280806160	M400	MD40
280806160	M400	MD42
280806160	M400	MD43

136

Buffer - Completo

- Com a buferização completa, todos os registros da tabela são carregados no buffer a partir do primeiro acesso à tabela.
- As tabelas que devem ser buferizadas são as tabelas com poucos dados, lidos frequentemente e raramente atualizados.

137

Buffer - Completo

- No exemplo abaixo, o registro marcado é lido pelo programa. Se a tabela possuir buffer completo, todos os registros são gravados no buffer.

```
SELECT *
  FROM MARD
 WHERE MATNR = '208014400'
   AND WERKS = 'M400'
   AND LGORT = 'MD40'.
```

Tabela transparente MARD

MATNR	WERKS	LGORT
208014180	M400	MD40
208014180	M400	MD42
208014400	M400	MD40
208014400	M400	MD42
280806160	M400	MD40
280806160	M400	MD42
280806160	M400	MD43

Conteúdo do Buffer

208014180	M400	MD40
208014180	M400	MD42
208014400	M400	MD40
208014400	M400	MD42
280806160	M400	MD40
280806160	M400	MD42
280806160	M400	MD43

Log

- O uso do flag de Log é utilizado quando há a necessidade de saber quais registros foram alterados.
- Se o log estiver ativo, cada alteração nos dados da tabela (com os comandos UPDATE e DELETE) pelo usuário ou pelo programa é registrado na tabela de log DBTABPRT.
- O log deve ser usado com cuidado pois pode impactar a performance do sistema e se houver acesso de vários usuários simultaneamente pode causar a situação de Lock.

139

Índice

- O índice é utilizado para aumentar a eficiência na busca do registro em determinada tabela.
- O índice pode ser considerado um cópia reduzida de uma tabela contendo determinados campos. Essa cópia é sempre ordenada.
- O índice contém um ponteiro ao registro correspondente da tabela selecionada, permitindo que os campos não contidos no índice sejam lidos.

140

Índice

- Existem dois tipos de índice:
 - Índice Primário
 - O índice primário possui os campos chaves da tabela e o ponteiro para os campos não-chave da tabela. É criado automaticamente.
 - Índice Secundário
 - O índice secundário é necessário se a tabela é acessada freqüentemente de um modo que não aproveita os campos chaves da tabela
- Podem ser criados mais de um índice secundário, mas deve-se levar em consideração a performance, pois a cada registro inserido na tabela, o sistema tem que armazenar os dados do índice.
- O sistema identifica automaticamente qual índice será utilizado se houver mais de um índice.

141

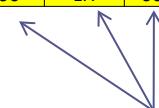
Índice

- **Índice Primário**

MANDT	CARRID	COUNTNUM	P
100	AA	00000001	7
100	BA	00000001	9
100	BA	00000002	12
100	BA	00000003	5
100	BA	00000004	2
100	LH	00000001	10
100	LH	00000002	4
100	LH	00000003	8
100	LH	00000004	11
100	LH	00000005	1
100	LH	00000006	13
100	LH	00000007	6
100	LH	00000008	14
100	UA	00000001	3

Tabela SCOUNTER

MANDT	CARRID	COUNTNUM	AIRPORT
100	LH	00000005	ACA
100	BA	00000004	ACE
100	UA	00000001	BER
100	LH	00000002	LCY
100	BA	00000003	LHR
100	LH	00000007	BER
100	AA	00000001	DEN
100	LH	00000003	FRA
100	BA	00000001	LCY
100	LH	00000001	LGW
100	LH	00000004	LHR
100	BA	00000002	MUC
100	LH	00000006	RTM
100	LH	00000008	HAM



Campos chaves

142

Índice

- Índice Secundário

```
SELECT *
FROM SCOUNTER
WHERE AIRPORT = 'LCY'.
```

Binary Search

AIRPORT	P
ACA	1
ACE	2
BER	3
BER	6
DEN	7
FRA	8
HAM	14
LCY	4
LCY	9
LGW	10
LHR	5
LHR	11
MUC	12
RTM	13

Tabela SCOUNTER

MANDT	CARRID	COUNTNUM	AIRPORT
100	LH	00000005	ACA
100	BA	00000004	ACE
100	UA	00000001	BER
100	LH	00000002	LCY
100	BA	00000003	LHR
100	LH	00000007	BER
100	AA	00000001	DEN
100	LH	00000003	FRA
100	BA	00000001	LCY
100	LH	00000001	LGW
100	LH	00000004	LHR
100	BA	00000002	MUC
100	LH	00000006	RTM
100	LH	00000008	HAM

143

Categoria de Ampliação

- A seleção de uma categoria de ampliação é necessária, uma vez que nas ampliações de tabelas e de estruturas em programas sem verificação unicode ativa, principalmente nas verificações de tipo e em conjunto com estruturas profundas, podem ocorrer erros de sintaxe e em tempo de execução.
- Em função da definição de estrutura, os botões de rádio permitidos da caixa de diálogo estão prontos para entrada

144

Categoria de Ampliação (ECC 6.0)

- As opções de categoria de ampliação são:
 - não ampliável
 - A estrutura não pode ser ampliada.
 - ampliável e em forma de caractere
 - Todas os componentes de estrutura e as respectivas ampliações têm de ser em forma de caractere (C, N, D ou T).
 - ampliável e em forma de caractere ou numérica
 - A estrutura e a respectiva ampliação não pode conter categorias de dados profundas (tabelas, referências, cadeias).
 - ampliável conforme pretendido
 - A estrutura e a respectiva ampliação pode conter componentes, cuja categoria de dados é facultativa.
 - não classificada
 - Esta categoria pode ser selecionada, por exemplo, para um status de transição, mas não ao criar estruturas.

145

146

DOMÍNIOS E ELEMENTO DE DADOS

Domínio

- O domínio é um objeto que contém as informações técnicas de um campo, como o tipo de dados e o tamanho do campo.
- No domínio também é definido o intervalo de valores, os quais podem ser valores individuais ou um conjunto de valores ou uma tabela de verificação.
- No domínio é definido as rotinas de conversão.

147

Domínio - Tipos de dados

- Tipos de Dados do Dicionário

TIPO	DESCRIÇÃO
ACCP	Período contábil AAAAMM
CHAR	Cadeias de caracteres com no máximo 255 caracteres
CLNT	Mandante
CUKY	Código da moeda, é referido por campos CURR
CURR	Campo de moeda (divisa), arquivado como DEC
DATS	Campo de data (AAAMMDD), arquivado como carac(8)
DEC	Campo de cálculo ou montante com vírgula e sinal pos./neg.
FLTP	Número em ponto flutuante com precisão de 8 bytes
INT1	1 byte nº inteiro 0 a 255
INT2	2 byte nº inteiro -32767 a 32767
INT4	4-byte nº inteiro -2177483647 a 2177483647
LANG	Código de idioma sempre 1 caracter
LCHR	Seq. longa de caracteres maior que 255 e necessita campo INT2
LRAW	Seq. longa de bytes, necessita campo INT2
NUMC	Campo de caracteres só com dígitos máximos de 255 dígitos
PREC	Precisão de um campo QUAN
QUAN	Campo quantidade, aponta p/campo de unidades de formato UNIT
RAW	Seqüência de bytes não interpretada
TIMS	Campo de hora (hhmmss), arquivado como caract.(6)
VARC	Seq. longa caracteres, não suportada a partir da versão 3.0
UNIT	Código de unidades para campos QUAN

148

Elemento de Dados

- O elemento de dados é um objeto que contém as possíveis descrições utilizadas na telas dos programas e ajudas de pesquisa.
- No elemento de dados, é especificado o domínio a ser utilizado.
- O elemento de dados é atribuído aos campos das tabelas, estruturas e programas ABAP.

149

Domínio & Elemento de Dados

- O domínio só pode ser utilizado pelo elemento de dados e ambos se completam.
- Um domínio pode ser utilizado por vários elemento de dados, mas o elemento de dados só pode ter um domínio.
- O campo de uma tabela herda as características do elemento de dados, tais como a descrição e ajuda de pesquisa e o tipo de dado e tamanho do domínio atribuído ao elemento de dados.

150

Campo & Elemento de Dados & Domínio

O tipo de dados e o tamanho são sempre consistente

TABELA 1



TABELA 2

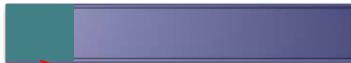


TABELA 3



Elemento de Dados 1

Elemento de Dados 2

Domínio

151

152

Criação de Tabela

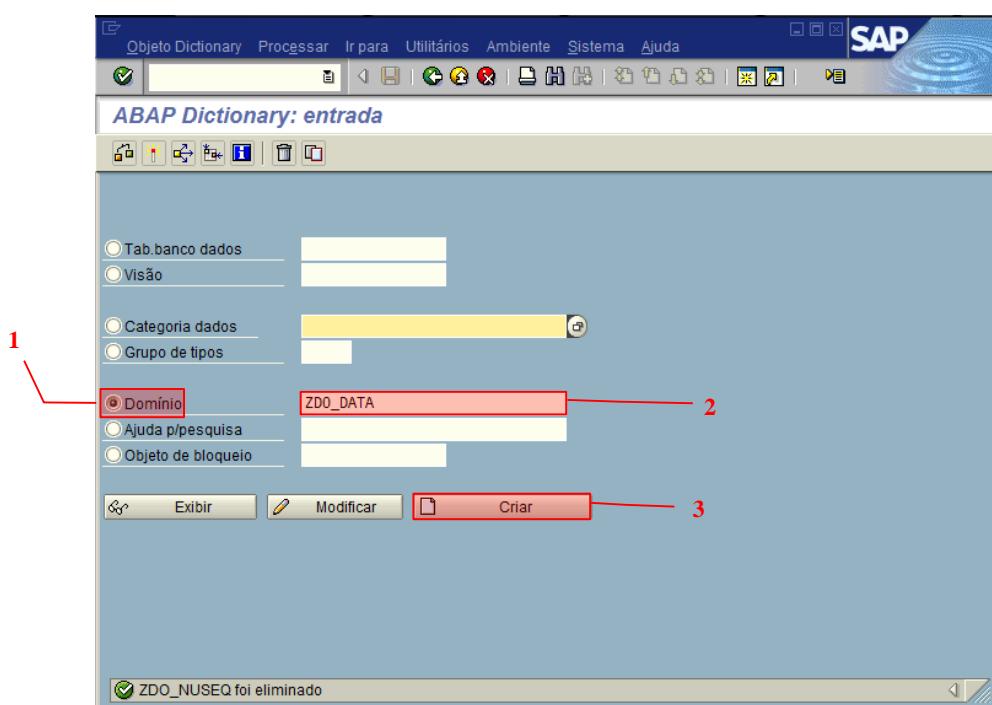
- Antes de criar a tabela, é necessário criar os domínios que serão atribuídos aos elementos de dados os quais serão atribuídos aos campos da tabela.

153

154

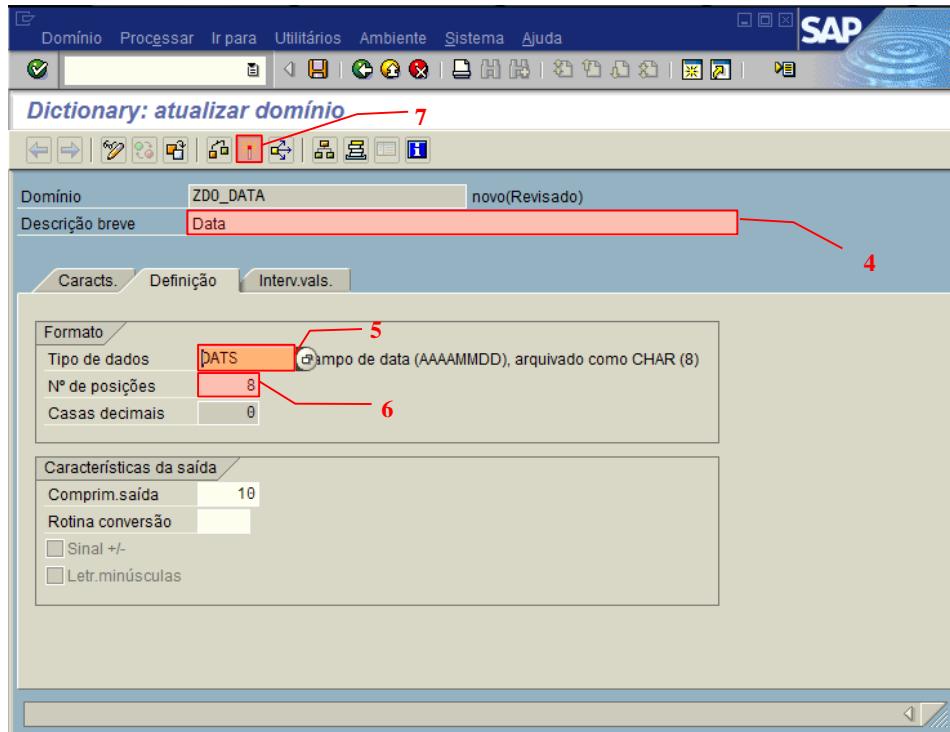
Criação do Domínio - SE11

1. Marcar o botão de rádio “Domínio”
2. Inserir o nome do domínio (sempre começa com a letra Z)
3. Clicar no botão “Criar”



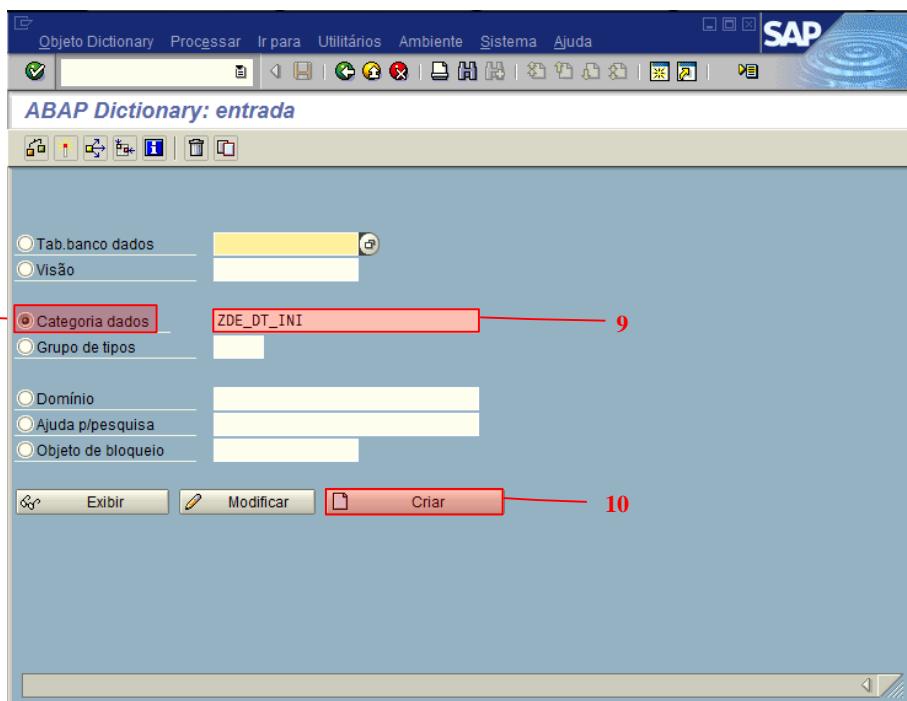
Criação do Domínio - Definição

4. Inserir a descrição do domínio
5. Informar o tipo de dados
6. Informar a quantidade de posições
7. Ativar o domínio



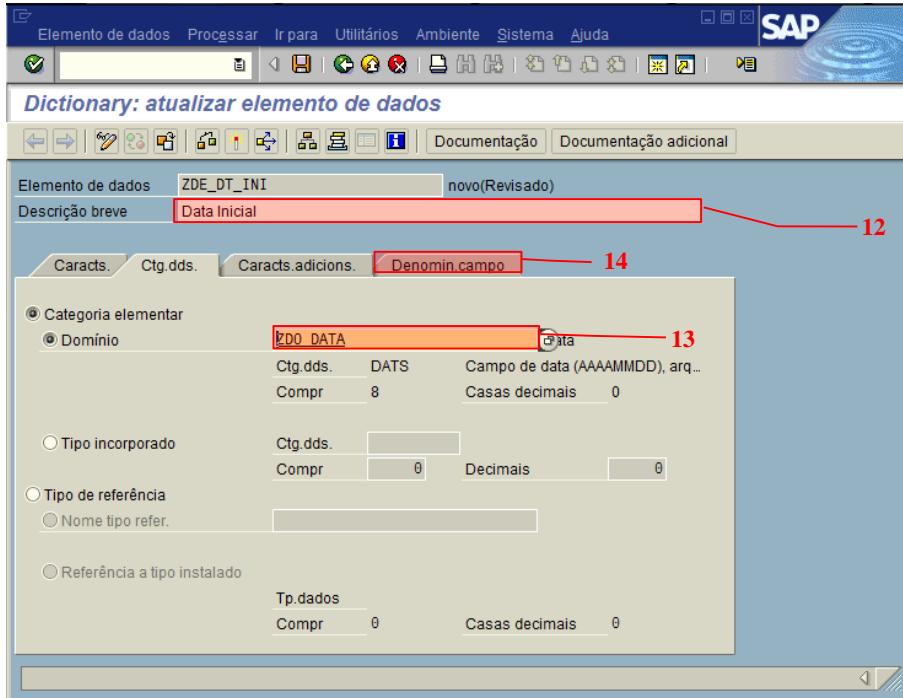
Criação do Elemento de Dados - SE11

8. Marcar o botão de rádio “Categoria dados”
9. Inserir o nome do elemento de dados (sempre começa com a letra Z)
10. Clicar no botão “Criar”
11. Selecionar a opção “Elemento de Dados”



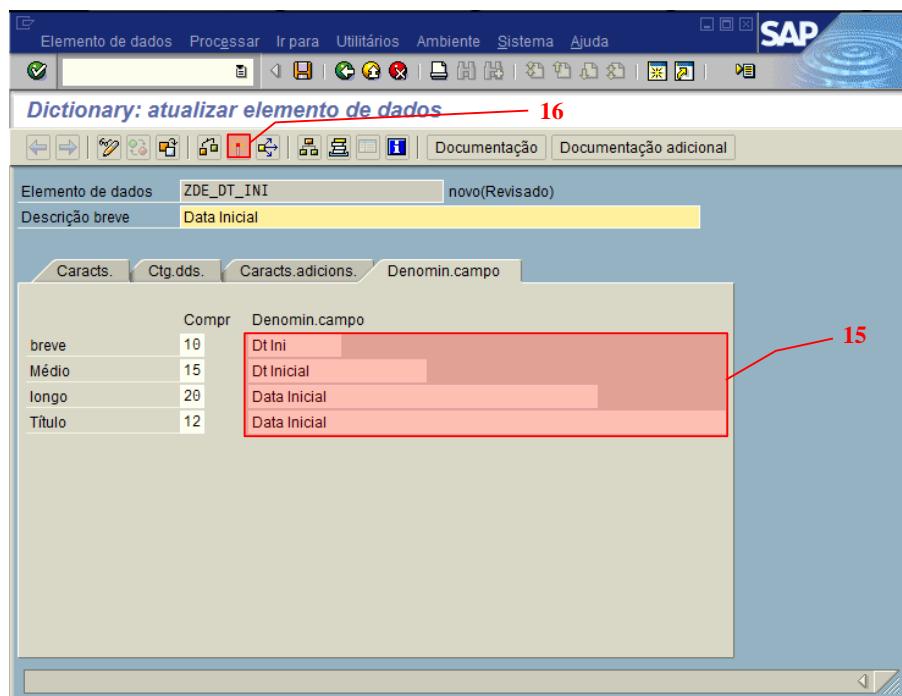
Criação do Elemento de Dados - Ctg.dds

12. Inserir a descrição
13. Inserir o nome do domínio criado
14. Clicar na aba “Denomin.campo” para inserir os textos do campo



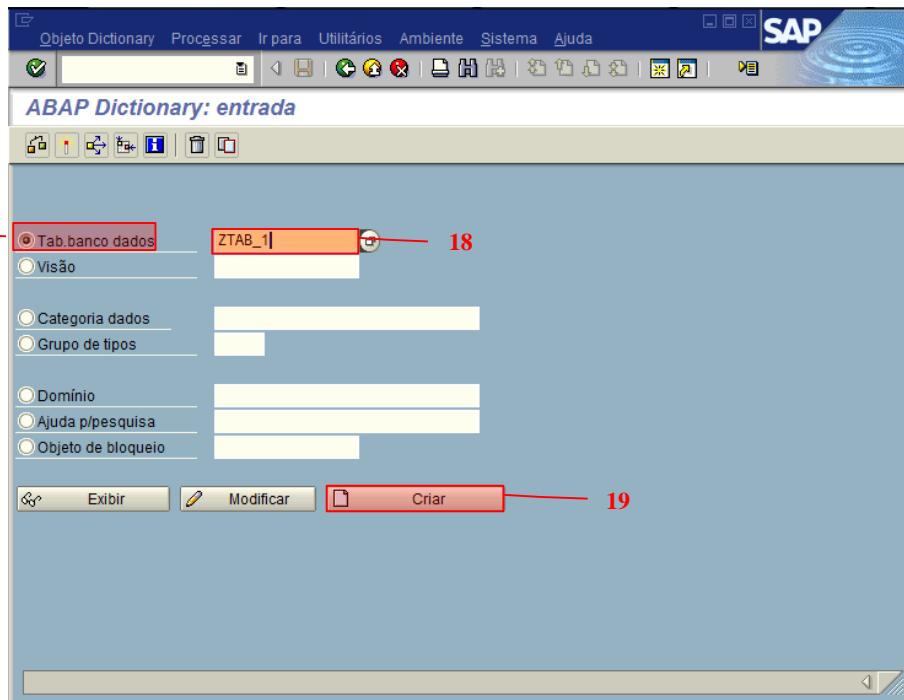
Criação do Elemento de Dados - Denomim.campo

15. Inserir a descrição
16. Ativar o elemento de dados



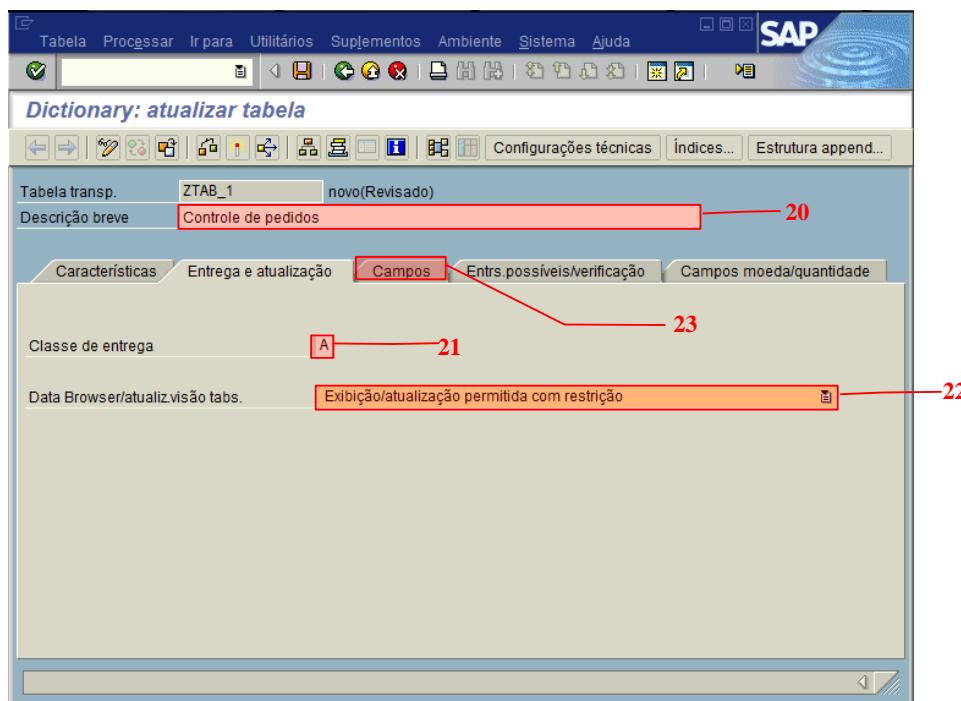
Criação da Tabela - SE11

17. Marcar o botão de rádio “Tab. Banco dados”
18. Inserir o nome da tabela (sempre começa com a letra Z)
19. Clicar no botão “Criar”



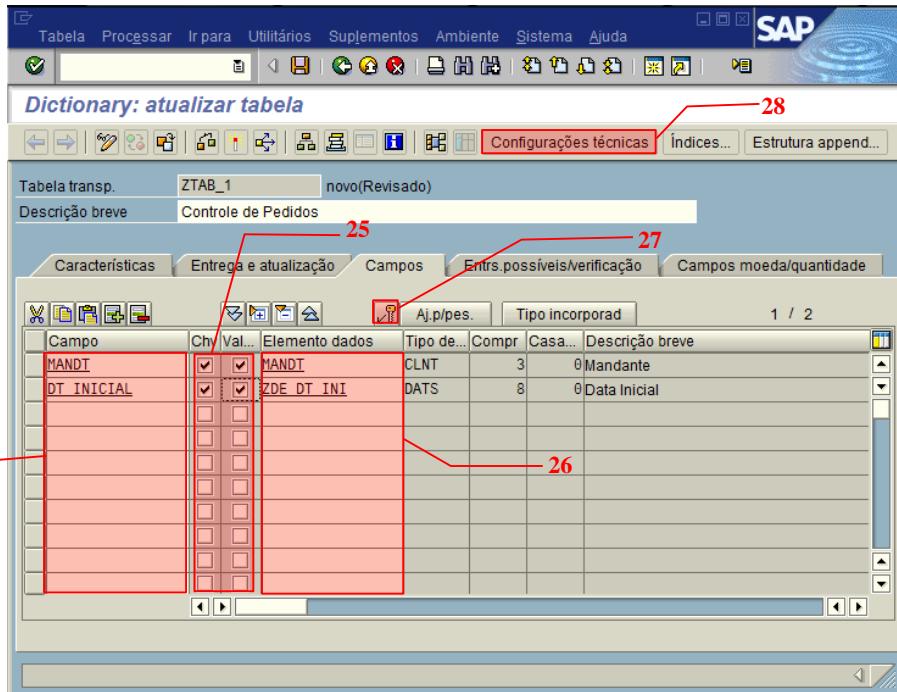
Criação da Tabela - Entrega e atualização

20. Inserir a descrição da tabela
21. Definir a classe de entrega
22. Definir a regra de atualização da tabela
23. Clicar na aba “Campos”



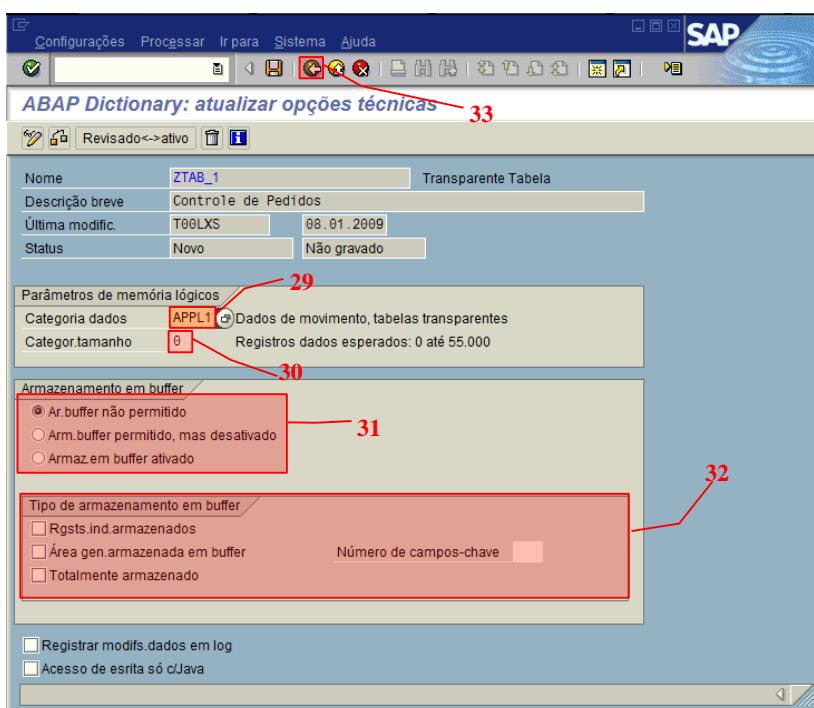
Criação da Tabela - Campos

24. Definir o nome do campo (nomenclatura livre)
25. Definir se o campo é chave e se o campo é NOT NULL
26. Inserir o nome do elemento de dados
27. Definir a chave externa (opcional)
28. Clicar no botão “Configurações técnicas” (obrigatório)



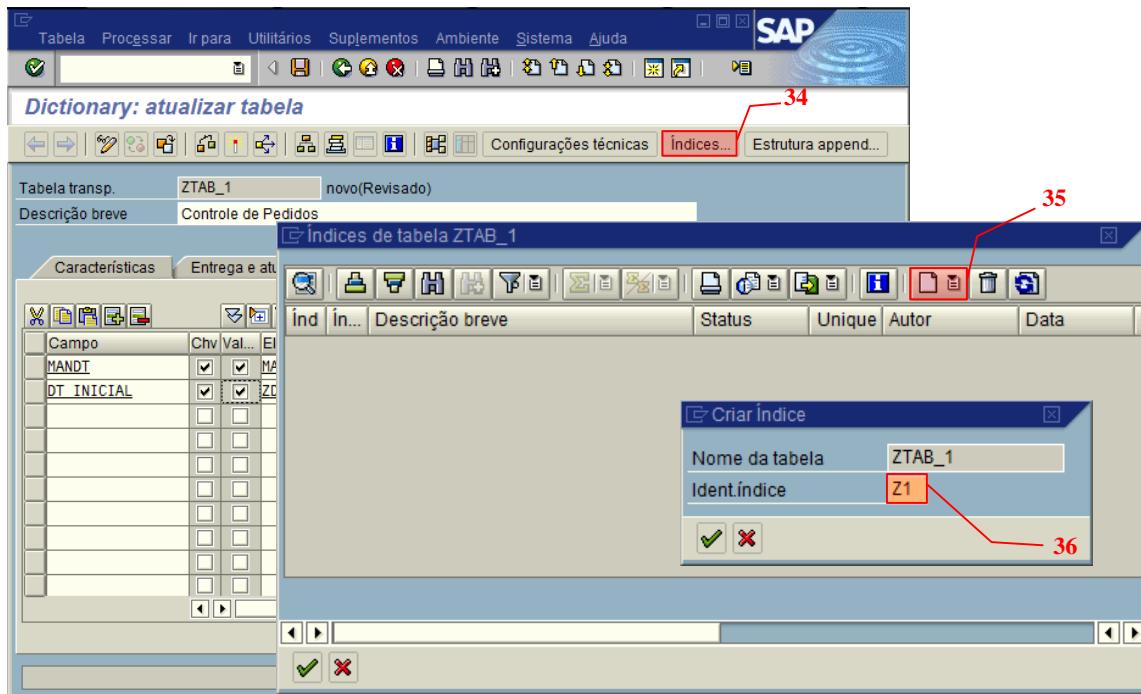
Criação da Tabela - Opções técnicas

29. Definir a categoria de dados
30. Definir a categoria de tamanho
31. Ativar o buffer se necessário
32. Definir o tipo de armazenamento em buffer caso o buffer tenha sido ativado
33. Voltar para a tela anterior



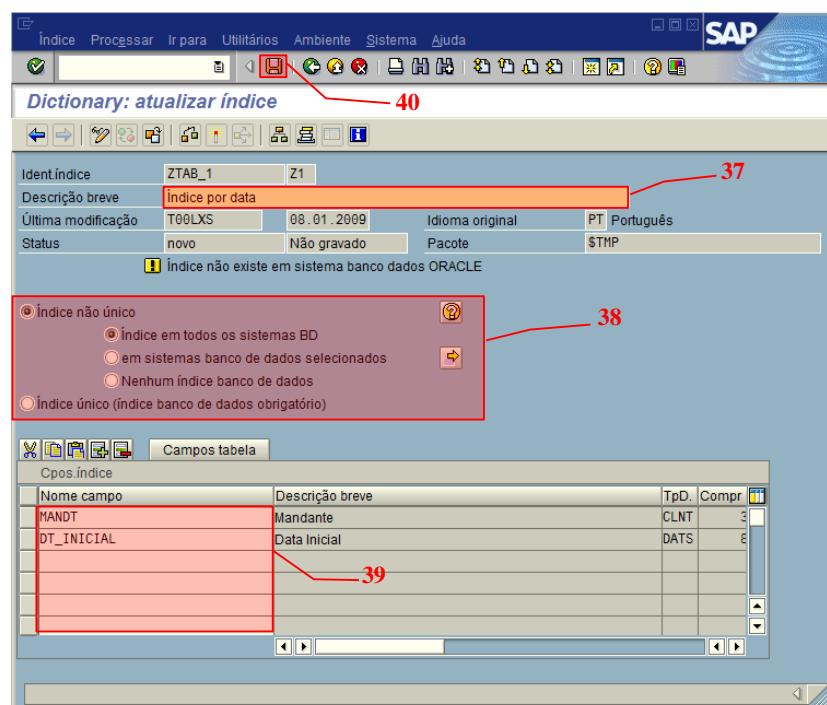
Criação da Tabela - Índice

34. Clicar no botão “Índices...” para abrir a janela de índice
35. Clicar no botão “Criar” e selecionar a opção desejada para abrir a janela de identificação do índice
36. Inserir o código de identificação do índice



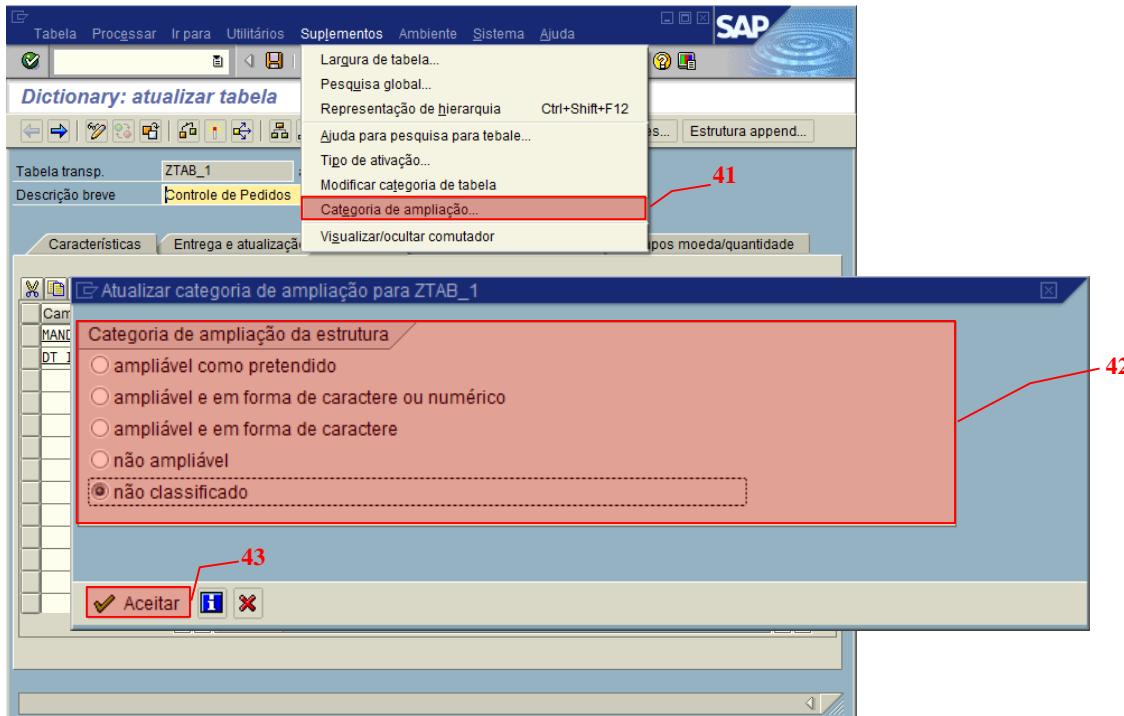
Criação da Tabela - Índice

- 37) Inserir a descrição do índice
- 38) Indicar os parâmetros do índice
- 39) Indicar os campos da tabela que farão parte do índice
- 40) Salvar o índice



Criação da Tabela - Categoria de Ampliação

- 41) No menu “Suplementos”, selecionar a opção “Categoria de ampliação...”
- 42) Selecionar a categoria de ampliação pretendida
- 43) Clicar em “Aceitar”



Visualizar dados da tabela

- Os dados incluídos em uma tabela podem ser visualizados através das transações SE16 ou SE16N, sendo esta última a versão **mais nova** nas

Data Browser: Tabela MARA 200 acertos						
Tabela:	MARA	Campos exibidos:	16 de 210 Colunas chave fixas:	2 Larg. lista 0250		
210	519405690936	10.10.2007	LELISS	04.06.2008	B00PMJ	KDEGB
210	9101650037	10.10.2007	LELISS	16.09.2008	T00FIV	KCDQZXB
210	901537000747	26.08.2007	JEFFERSON	09.09.2008		KCD
210	901641110907	14.12.2007	MARCELO	16.09.2008	T00FIV	KCDQZPEGB
210	905200749058	02.01.2008	LELISS	00.00.0000		K
210	905292000001	25.08.2007	LELISS	16.09.2008	T00FIV	KCDQZPEGBV
210	905351150008	24.10.2007	LELISS	01.02.2008		KCDQ
210	905411600001	24.10.2007	LELISS	01.02.2008	RENATO	KCDQ
210	922901TA0001	09.10.2007	LELISS	00.00.0000		K
210	922901TA0002	09.10.2007	LELISS	00.00.0000		K
210	922901TA0003	09.10.2007	LELISS	00.00.0000		K
210	922901TA0004	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0005	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0006	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0007	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0008	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0009	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0010	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0011	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0012	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0013	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0014	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0015	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0016	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0017	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0018	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0019	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0020	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0021	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0022	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0023	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0024	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0025	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0026	21.12.2007	LELISS	01.02.2008	RENATO	KCD
210	922901TA0027	09.01.2008	LELISS	00.00.0000		K
210	922901TA0028	09.01.2008	LELISS	00.00.0000		K
210	922901TA0029	09.01.2008	LELISS	00.00.0000		K
210	922901TA0030	09.01.2008	LELISS	00.00.0000		K
210	922901TA0031	09.01.2008	LELISS	00.00.0000		K
210	922901TA0032	09.01.2008	LELISS	00.00.0000		K
210	922901TA0033	09.01.2008	LELISS	00.00.0000		K
210	940703300197	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300198	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300199	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300200	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300201	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300202	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300203	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300204	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300205	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300206	17.12.2007	MARCELO	19.09.2008	T00FIV	KCDQZDX
210	940703300207	09.01.2008	LELISS	00.00.0000		K
210	940703300208	09.01.2008	LELISS	00.00.0000		K
210	940703300209	09.01.2008	LELISS	00.00.0000		K
210	940703300210	09.01.2008	LELISS	00.00.0000		K
210	940703300211	09.01.2008	LELISS	00.00.0000		K
210	940703300212	09.01.2008	LELISS	00.00.0000		K
210	940703300213	09.01.2008	LELISS	00.00.0000		K
210	940703300214	09.01.2008	LELISS	00.00.0000		K
210	940703300215	09.01.2008	LELISS	00.00.0000		K
210	940703300216	09.01.2008	LELISS	00.00.0000		K
210	940703300217	09.01.2008	LELISS	00.00.0000		K
210	940703300218	09.01.2008	LELISS	00.00.0000		K
210	940703300219	09.01.2008	LELISS	00.00.0000		K
210	940703300220	09.01.2008	LELISS	00.00.0000		K
210	940703300221	09.01.2008	LELISS	00.00.0000		K
210	940703300222	09.01.2008	LELISS	00.00.0000		K
210	940703300223	09.01.2008	LELISS	00.00.0000		K
210	940703300224	09.01.2008	LELISS	00.00.0000		K
210	940703300225	09.01.2008	LELISS	00.00.0000		K
210	940703300226	09.01.2008	LELISS	00.00.0000		K
210	940703300227	09.01.2008	LELISS	00.00.0000		K
210	940703300228	09.01.2008	LELISS	00.00.0000		K
210	940703300229	09.01.2008	LELISS	00.00.0000		K
210	940703300230	09.01.2008	LELISS	00.00.0000		K
210	940703300231	09.01.2008	LELISS	00.00.0000		K
210	940703300232	09.01.2008	LELISS	00.00.0000		K
210	940703300233	09.01.2008	LELISS	00.00.0000		K
210	940703300234	09.01.2008	LELISS	00.00.0000		K
210	940703300235	09.01.2008	LELISS	00.00.0000		K
210	940703300236	09.01.2008	LELISS	00.00.0000		K
210	940703300237	09.01.2008	LELISS	00.00.0000		K
210	940703300238	09.01.2008	LELISS	00.00.0000		K
210	940703300239	09.01.2008	LELISS	00.00.0000		K
210	940703300240	09.01.2008	LELISS	00.00.0000		K
210	940703300241	09.01.2008	LELISS	00.00.0000		K
210	940703300242	09.01.2008	LELISS	00.00.0000		K
210	940703300243	09.01.2008	LELISS	00.00.0000		K
210	940703300244	09.01.2008	LELISS	00.00.0000		K
210	940703300245	09.01.2008	LELISS	00.00.0000		K
210	940703300246	09.01.2008	LELISS	00.00.0000		K
210	940703300247	09.01.2008	LELISS	00.00.0000		K
210	940703300248	09.01.2008	LELISS	00.00.0000		K
210	940703300249	09.01.2008	LELISS	00.00.0000		K
210	940703300250	09.01.2008	LELISS	00.00.0000		K
210	940703300251	09.01.2008	LELISS	00.00.0000		K
210	940703300252	09.01.2008	LELISS	00.00.0000		K
210	940703300253	09.01.2008	LELISS	00.00.0000		K
210	940703300254	09.01.2008	LELISS	00.00.0000		K
210	940703300255	09.01.2008	LELISS	00.00.0000		K
210	940703300256	09.01.2008	LELISS	00.00.0000		K
210	940703300257	09.01.2008	LELISS	00.00.0000		K
210	940703300258	09.01.2008	LELISS	00.00.0000		K
210	940703300259	09.01.2008	LELISS	00.00.0000		K
210	940703300260	09.01.2008	LELISS	00.00.0000		K
210	940703300261	09.01.2008	LELISS	00.00.0000		K
210	940703300262	09.01.2008	LELISS	00.00.0000		K
210	940703300263	09.01.2008	LELISS	00.00.0000		K
210	940703300264	09.01.2008	LELISS	00.00.0000		K
210	940703300265	09.01.2008	LELISS	00.00.0000		K
210	940703300266	09.01.2008	LELISS	00.00.0000		K
210	940703300267	09.01.2008	LELISS	00.00.0000		K
210	940703300268	09.01.2008	LELISS	00.00.0000		K
210	940703300269	09.01.2008	LELISS	00.00.0000		K
210	940703300270	09.01.2008	LELISS	00.00.0000		K
210	940703300271	09.01.2008	LELISS	00.00.0000		K
210	940703300272	09.01.2008	LELISS	00.00.0000		K
210	940703300273	09.01.2008	LELISS	00.00.0000		K
210	940703300274	09.01.2008	LELISS	00.00.0000		K
210	940703300275	09.01.2008	LELISS	00.00.0000		K
210	940703300276	09.01.2008	LELISS	00.00.0000		K
210	940703300277	09.01.2008	LELISS	00.00.0000		K
210	940703300278	09.01.2008	LELISS	00.00.0000		K
210	940703300279	09.01.2008	LELISS	00.00.0000		K
210	940703300280	09.01.2008	LELISS	00.00.0000		K
210	940703300281	09.01.2008	LELISS	00.00.0000		K
210	940703300282	09.01.2008	LELISS	00.00.0000		K
210	940703300283	09.01.2008	LELISS	00.00.0000		K
210	940703300284	09.01.2008	LELISS	00.00.0000		K
210	940703300285	09.01.2008	LELISS	00.00.0000		K
210	940703300286	09.01.2008	LELISS	00.00.0000		K
210	940703300287	0				

Visão

- Os dados de uma visão são derivadas de uma ou mais tabelas, mas não são armazenadas fisicamente.
- Uma visão pode ser utilizado por um programa ABAP para a seleção de dados.
- O primeiro passo é definir as tabelas bases
- No segundo passo, essas tabelas devem ser associadas através de condições Join
- O terceiro passo é definir os campos das tabelas base que deverão estar presentes na visão
- O quarto passo é definir as condições de seleção que restringem os registros a serem apresentados na visão

167

Visão

- Existem quatro tipos de visão:
 - Visão de banco de dados
 - Visão de projeção
 - Visão de atualização
 - Visão de ajuda

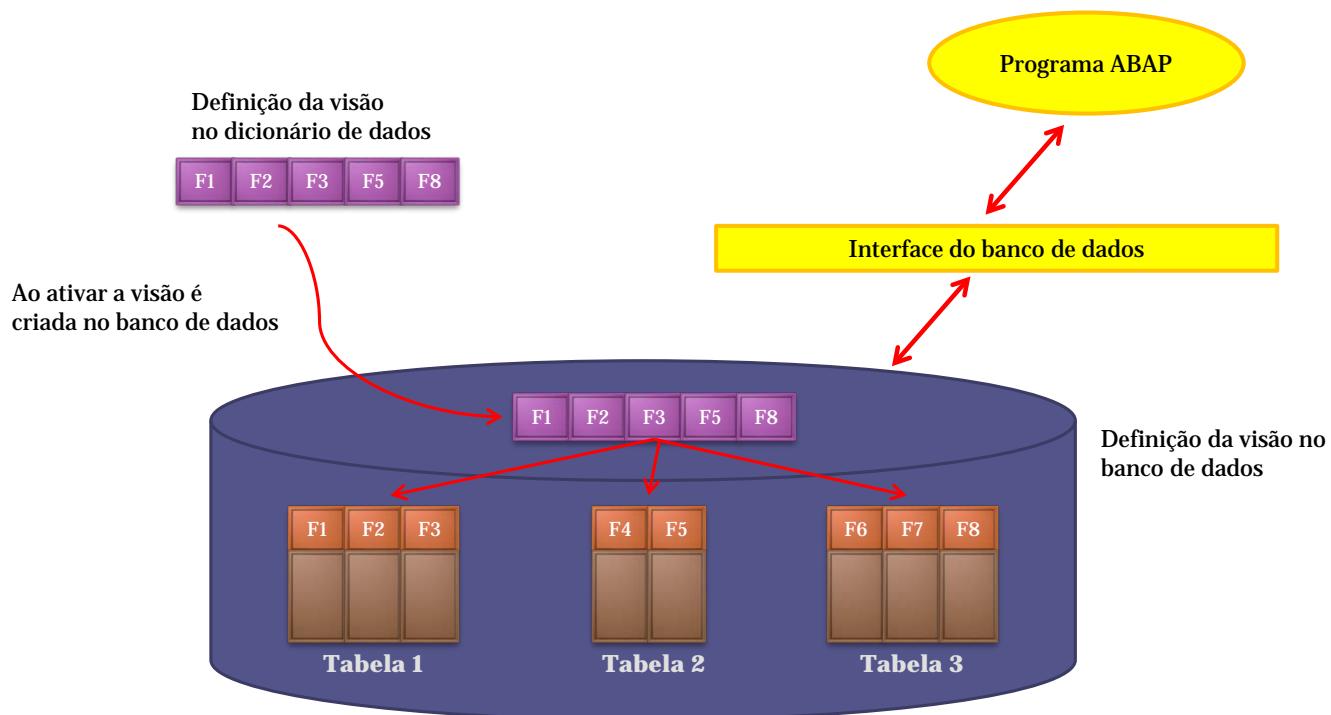
168

Visão de banco de dados

- Geralmente, os dados estão distribuídos entre várias tabelas. Uma visão fornece um meio de obter estes dados distribuídos.
- A operação de join é realizada no banco de dados, minimizando o número de acesso ao banco de dados.
- Se a visão possuir somente uma tabela, o sistema permite a inserção de registro, caso contrário somente a leitura das tabelas
- O acesso aos dados de uma visão é geralmente mais rápido do que acesso individual as tabelas

169

Visão de banco de dados



170

Visão de projeção

- Visão de projeção é utilizado para ocultar campos de uma tabela.
- A visão de projeção contém apenas uma tabela base. Não é possível definir condições de seleção.
- Não há um objeto correspondente no banco de dados para a visão de projeção.
- É possível acessar tabelas clusters e pool com a visão de projeção

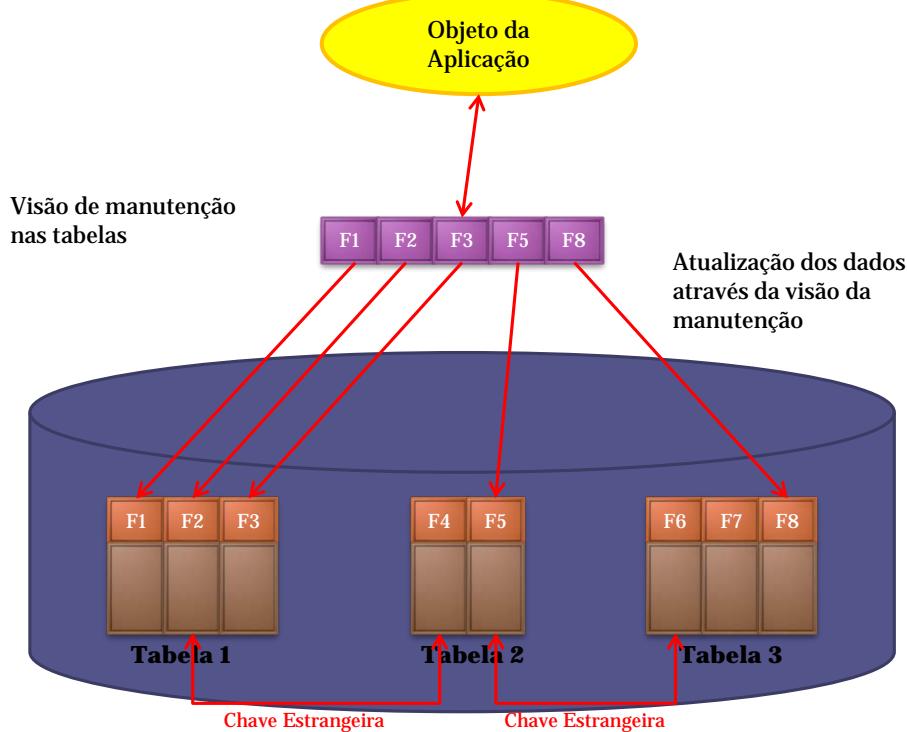
171

Visão de atualização

- A visão de atualização fornece um meio fácil de atualizar objetos de aplicações complexas.
- Todas as tabelas de uma visão de atualização devem estar associadas com chaves estrangeiras, pois as condições de join na visão de manutenção são derivadas das chaves estrangeiras.
- Não é possível entrar as condições de join como a visão de banco de dados.
- O mecanismos de manutenção é fornecido através da transação SE54, para criar o programa de manutenção e a transação SM30 para executar o programa de atualização.

172

Visão de atualização



173

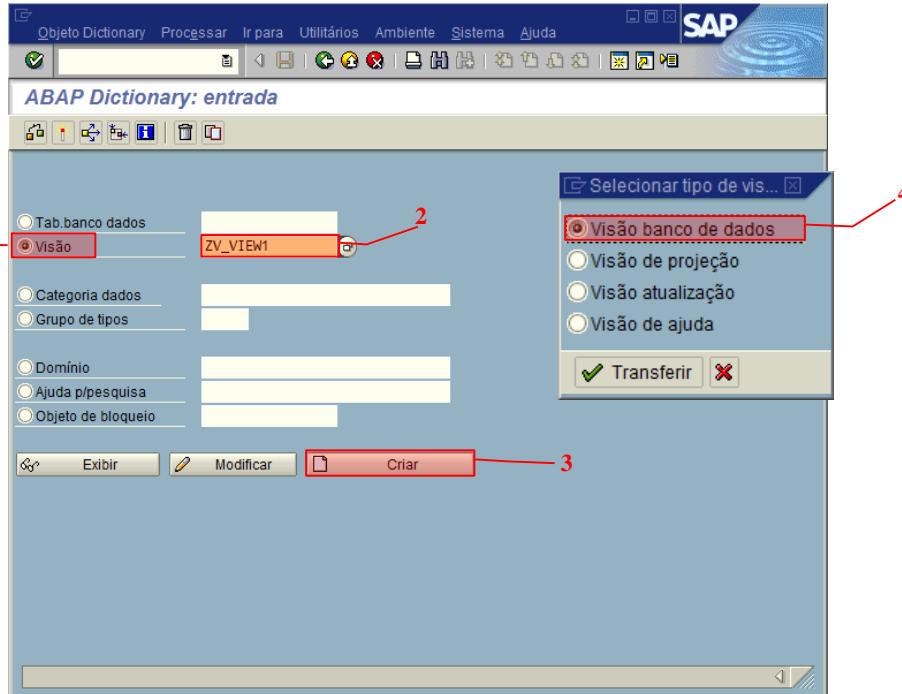
Visão de ajuda

- A visão de ajuda é utilizada na ajuda de pesquisa.
- O método de seleção de uma ajuda de pesquisa é tabela ou visão.
- Se for necessário utilizar uma visão de banco de dados, será implementado a condição inner join. Se for necessário a condição outer join, a solução é a visão de ajuda.
- Todas as tabelas definidas na visão de ajuda deve estar associadas com chave estrangeira.

174

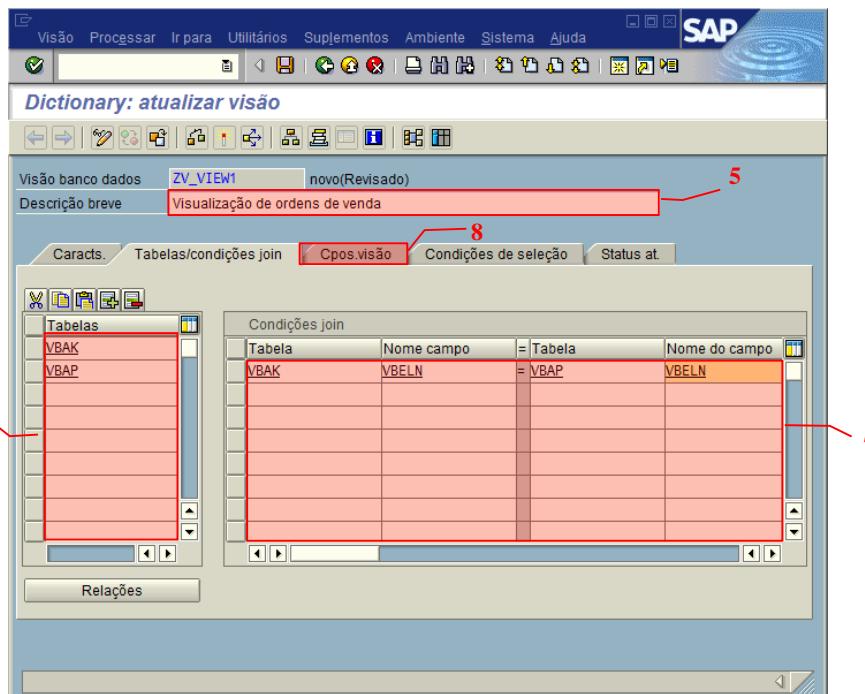
Criação - Visão de banco de dados

1. Marcar o botão de rádio “Visão”
2. Inserir o nome da visão (sempre começa com a letra Z)
3. Clicar no botão “Criar”
4. Selecionar a opção “Visão de banco de dados”



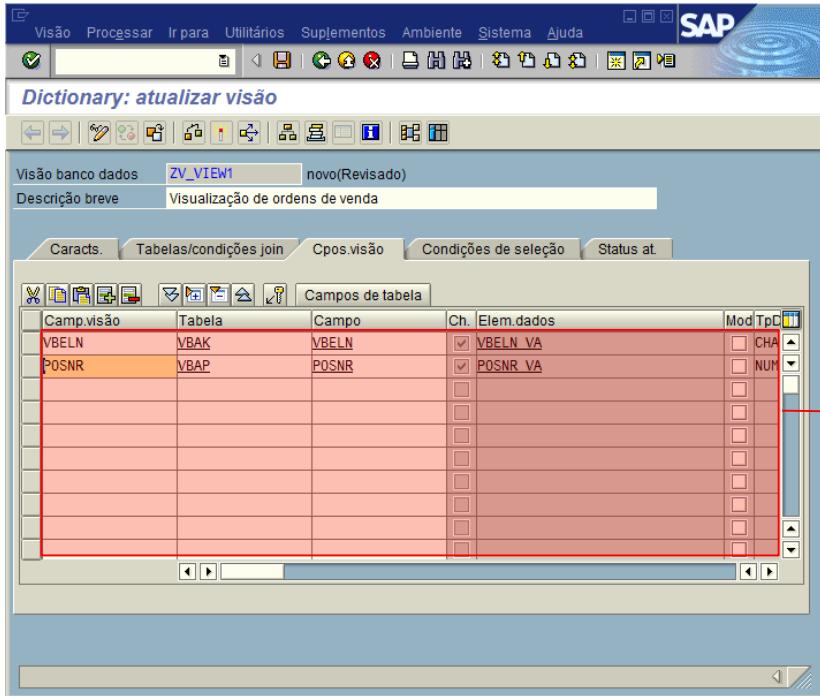
Criação - Visão de DB - tabelas/condições join

5. Inserir a descrição da visão
6. Definir as tabelas bases
7. Definir as condições de Join
8. Clicar na aba Cpos.visão



Criação - Visão de DB - Campos Visão

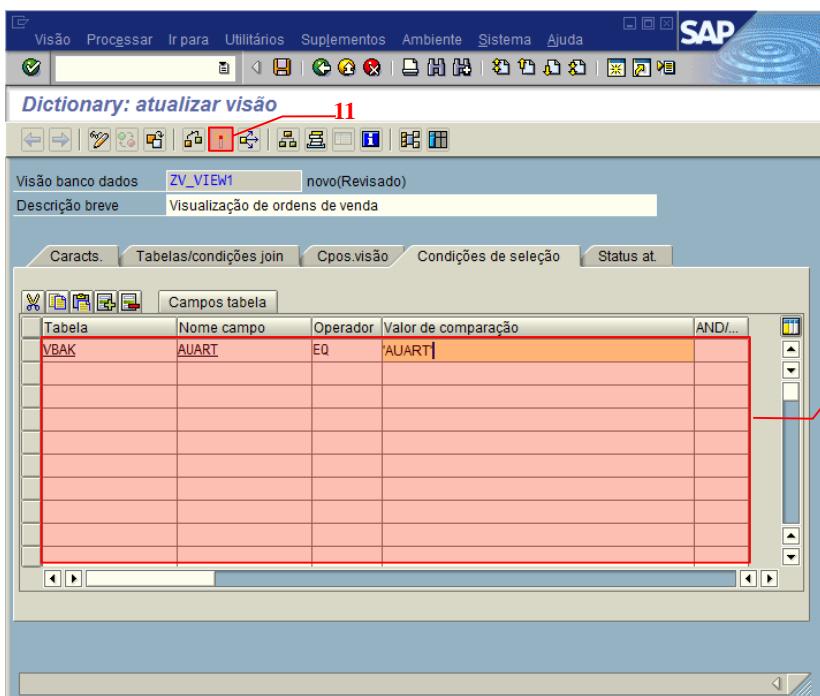
- ### 9. Especificar os campos da visão



9

Criação - Visão de DB - condição de seleção

10. Definir a condição de seleção
 11. Ativar a visão



10

Estruturas

- As estruturas são utilizadas para definir os dados da interface dos module pools e telas e para definir os tipo de parametros dos módulos de função.
- As estruturas possuem componentes que podem referenciar a um tipo elementar (via elemento de dados ou diretamente especificando o tipo de dados e tamanho), outra estrutura ou uma categoria de tabela.
- As estruras podem possuir vários níveis.

179

Ajudas de Pesquisa

- A ajuda de pesquisa (F4) é uma função standard do SAP.
- Ele permite apresentar ao usuário uma lista de possíveis entradas para um determinado campo.
- É útil quando um campo requer a entrada de um valor formal.

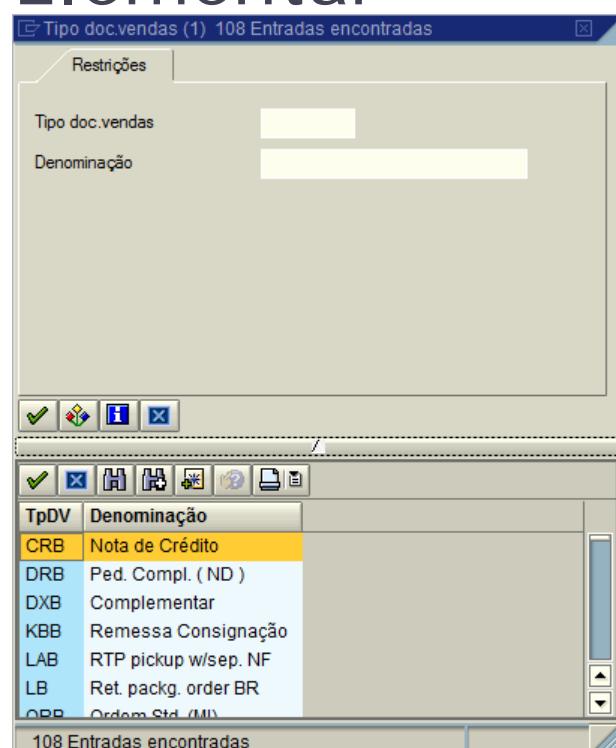
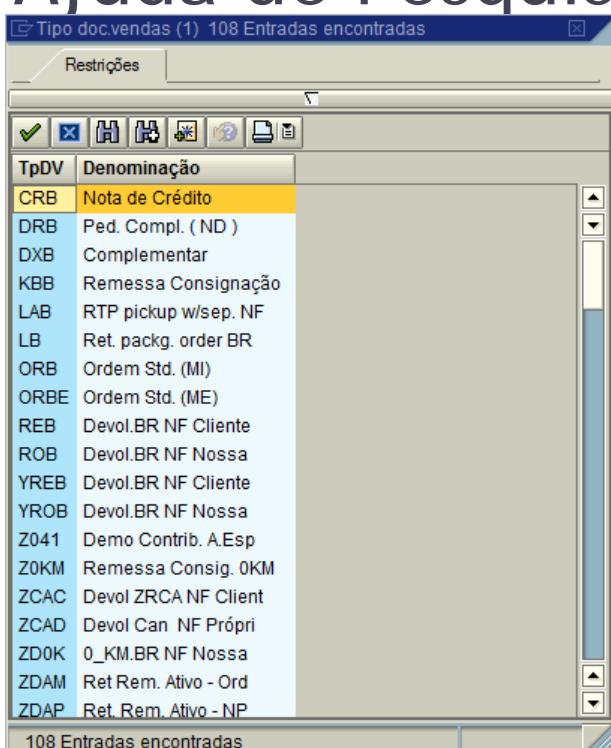
180

Ajuda de Pesquisa - Processo

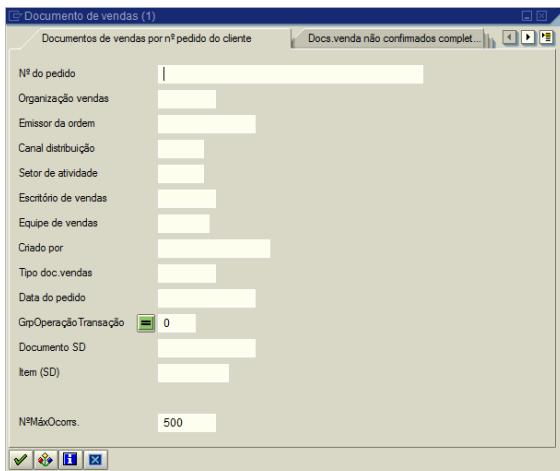
1. O usuário clica no botão de match-code relacionado ao campo para obter uma lista de possíveis valores para o campo
 2. O sistema apresenta opções de pesquisa para os valores
 3. O usuário especifica o critério de seleção e inicia a pesquisa
 4. O sistema determina os valores que satisfazam as restrições e apresenta o resultado em uma lista
 5. O usuário seleciona o valor através de duplo clique.
 6. O valor selecionado é transferido para o campo

181

Ajuda de Pesquisa Elementar



Ajuda de Pesquisa Coletiva



Nº do pedido	OrgV	EmissorOrd	CDst	SA	EscrV	EqVs	Criado por	TpDV	DataPedido	GOT	Documento	Itm
0000000000010	MV30	487	37	10	MR30	R04	DUTRA	ORB	21.12.1998	0	301	000000
000000023	MV30	487	37	10	MR30	R04	DUTRA	ORB	13.01.1999	0	209	000000
00199	MV30	479	33	10	MR30	R04	VENDA/002	ORB	19.01.1999	0	361	000000
00299	MV20	23	23	10			DUTRA	ORB	12.01.1999	0	203	000000
00299	MV30	456	33	10	MR30	R04	DUTRA	ORB	04.11.1998	0	71	000000
0086/98	MV30	464	33	10	MA14		DUTRA	ORB	12.01.1999	0	66	000000
01/99	MV30	437	37	10	MR30	R04	DUTRA	ORB	12.01.1999	0	76	000000
01/99	MV30	456	33	10	MR30	R04	DUTRA	ORB	12.01.1999	0	208	000000
0381	MV20	44	13	10			VENDA/002	ORB	12.01.1999	0	3	000000
062	MV30	365	38	10	MA10		PROJECT	ORB	00.00.0000	0	327	000000
10098	MV30	430	33	10	MR30	R04	DUTRA	ORB	16.01.1999	0	73	000000
101298	MV20	127	27	10			VENDA/002	ORB	11.01.1999	0	9900000001	000000
117.280	MV10	520	10	10			EXPED/001	Z010	11.01.1999	0	9900000003	000000
117.281	MV10	520	10	10			PROJECT	Z010	11.01.1999	0	159	000000
1444292	MV30	21	38	10	MA10		PROJECT	ORB	00.00.0000	0	19	000000
1444453	MV30	270	38	10	MA10		PROJECT	ORB	00.00.0000	0	16	000000
1445725	MV30	295	38	10	MA10		PROJECT	ORB	00.00.0000	0	200	000000
1445739	MV30	339	38	10	MA10		PROJECT	ORB	00.00.0000	0	17	000000
1446219	MV30	94	38	10	MA10		PROJECT	ORB	00.00.0000	0	157	000000
1446220	MV30	283	38	10	MA10		PROJECT	ORB	00.00.0000	0	18	000000
1446223	MV30	98	38	10	MA10		PROJECT	ORB	00.00.0000	0	159	000000
1446226	MV30	308	38	10	MA10		PROJECT	ORB	00.00.0000	0	79	000000
1447378	MV30	270	38	10	MA10		PROJECT	ORB	00.00.0000	0	159	000000

Existem mais de 500 entradas possíveis

183

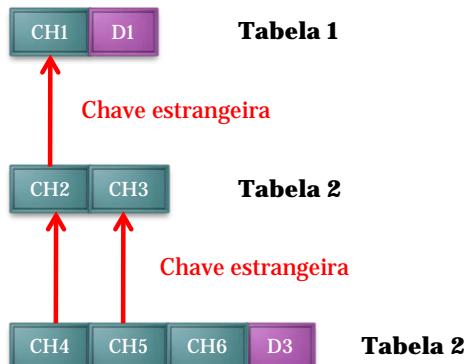
Objeto de bloqueio

- O R/3 sincroniza simultaneamente o acesso de vários usuários aos mesmos registros através de mecanismos de bloqueio.
- Os bloqueios são ativados e desativados através de módulos de função.
- Os módulos de função de bloqueio são criados automaticamente pelo dicionário de dados.
- O objeto de bloqueio pode ter uma ou mais tabelas.
- O nome do objeto de bloqueio deve começar com a letra E.

184

Objeto de bloqueio - Estrutura

- As tabelas nas quais os registros devem ser bloqueados através de objetos de bloqueio são definidos com as respectivas chaves.



185

Objeto de bloqueio - Parâmetros

- Os parâmetros de bloqueio de uma tabela consiste nas chaves da mesma.
- Os parâmetros de bloqueio são utilizados como parâmetros de entrada nos módulos de função de bloqueio para ativar ou desativar o bloqueio.

Chamada do módulo de função de bloqueio com as chaves
CH1 = 2 e CH3 = 1

Tabela 1	
CH1	D1
1	...
2	...
3	...

Tabela 2		
CH2	CH3	D2
1	1	...
2	1	...
2	2	...
3	1	...
3	3	...

Tabela 3				
CH4	CH5	CH6	D3	D4
1	1	A
2	1	A
2	1	B
2	1	C
2	3	A
2	3	B
3	1	A

Os registros em vermelho estão bloqueados

186

Objeto de bloqueio - Modo

- O modo de bloqueio controla como os usuários irão acessar os dados simultaneamente.
- Pode ser atribuído separadamente para cada tabela do objeto de bloqueio
- O acesso por mais de um usuário pode ser efetuado pelos seguintes modos:
 - Bloqueio de escrita (Exclusive Lock)
 - Os dados bloqueados só podem ser lidos ou processados por um único usuário. Tanto a solicitação de outra proteção contra gravação como a solicitação de um bloqueio de leitura são rejeitadas.
 - Bloqueio de leitura (Shared Lock)
 - Vários usuários podem acessar simultaneamente os mesmos dados para leitura. Contudo, assim que um usuário processar os dados, um segundo usuário já não pode acessar estes dados. As solicitações de novos bloqueios de leitura são aceitas, mesmo se forem provenientes de outros usuários. Uma proteção contra gravação é rejeitada.
 - Bloqueio de escrita ampliado (Exclusive but not cumulative)
 - Enquanto as proteções contra gravação podem ser solicitadas várias vezes e canceladas sucessivamente pela mesma transação, uma proteção contra gravação ampliada só pode ser solicitada uma vez pela mesma transação. Qualquer outra solicitação de um bloqueio é rejeitada.

187

Objeto de bloqueio - Módulo de Função

- Quando o objeto de bloqueio é ativado, dois módulos de função são criados.
 - ENQUEUE_<nome do objeto de bloqueio>
 - DEQUEUE_<nome do objeto de bloqueio>

188

Objeto de bloqueio - Módulo de Função - Parâmetros

- Campos do objetos de bloqueio
 - São os campos chaves definidos no dicionário de dados
 - Passar o valor 'X' para o campo que deseja bloquear
 - Se não for passado o valor 'X', será feito um bloqueio genérico

189

Transações

- SE11 – Dicionário de Dados
- SE16/SE16N – Exibição dos dados de tabelas
- SE14 – Utilitário de banco de dados
- SM30 – Atualização de visão de tabelas
- SE93 – Atualização de transação

190

Exercícios

a) Criar as seguintes tabelas no SAP

Notas Fiscais de Entrada de Material para Agrupamento com o nome: (Z##NFKGR)

Campo	Elemento de Dados	Domínio
NUMDOC	J_1BDOCNUM	
NFNUM	PRÓPRIO	NUM 6
LIFNR	LIFNR	
CGC_CPF	PRÓPRIO	PRÓPRIO

Itens das Notas Fiscais com o nome:
(Z##NFPGR):

Campo	Elemento de Dados	Domínio
NUMDOC	J_1BDOCNUM	
ITNUM	J_1BITMNUM	
MATNR	MATNR	
MENGE	J_1BNETQTY	
MEINS	PRÓPRIO	CUKY 3
VALOR	J_1BNETPRI	
MOEDA	PRÓPRIO	PRÓPRIO

Exercícios

- b) Criar os relacionamentos entre as duas tabelas do exercício anterior e as tabelas SAP
- c) Criar uma visão para listar os fornecedores e os respectivos materiais comercializados pelos mesmos.

- A visão deve conter os seguintes campos:
 - LIFNR (Número do fornecedor)
 - NAME1 (Nome do fornecedor)
 - MATNR (Código do material)
 - MAKTX (Descrição do material pelo idioma de logon)

Curso Avançado de ABAP

Abap Report

Tópicos

- ABAP Editor
- ABAP Report

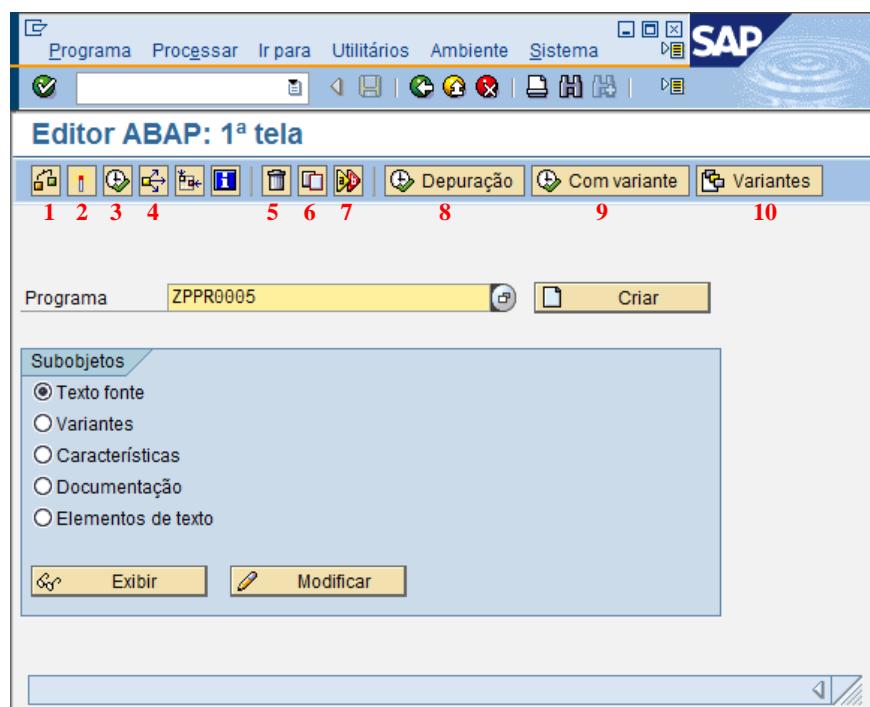


ABAP Editor

- O ABAP Editor é a ferramenta para a criação de programas ABAP
- A transação do ABAP Editor é SE38
- No ABAP Editor é possível:
 - Criar/Modifica/Visualizar o programa
 - Copiar e renomear o programa
 - Excluir o programa do sistema
 - Ativar o programa
 - Alterar as características do programa
 - Documentar o programa
 - Criar/Modificar/Visualizar os elementos de texto do programa

195

ABAP Editor - Tela inicial



1. Verificar a sintaxe do código
2. Ativar o programa
3. Executar o programa
4. Pesquisar onde o programa é utilizado
5. Excluir o programa do sistema
6. Copiar o programa
7. Renomear o programa
8. Depurar o programa
9. Depurar o programa com variante
10. Gerenciar variante do programa

196

ABAP Editor - Editor

The screenshot shows the SAP ABAP Editor interface. The title bar reads "Editor ABAP: Report ZPPR0005 modificar". The menu bar includes "Programa", "Processar", "Ir para", "Utilitários", "Ambiente", "Sistema", and "Ajuda". The toolbar has various icons for file operations. The main area displays the ABAP code for program ZPPR0005:

```

14: REPORT zppr0005 MESSAGE-ID zpp.
15:
16: INCLUDE zppr0005_top.
17: INCLUDE zppr0005_sel.
18: INCLUDE zppr0005_pbo.
19: INCLUDE zppr0005_pai.
20: INCLUDE zppr0005_f01.
21:
22: /**
23:  *** EVENTO: INITIALIZATION
24:  ****
25: INITIALIZATION.
26:
27:   v_dynnrr = '9000'.
28:   v_repid = sy-repid.
29:
30: /**
31:  *** EVENTO: START-OF-SELECTION
32:  ****
33: START-OF-SELECTION.
34:
35:   PERFORM z_seleciona_dados CHANGING v_retcode.
36:   CHECK v_retcode IS INITIAL.
37:
38:   CATT SCREEN !0000!

```

The status bar at the bottom indicates "ABAP Li 1 Co 1".

1. Exibir/Modificar código
2. Ativar/Desativar programa
3. Carregar outro programa
4. Pesquisar onde o programa é utilizado
5. Excluir o programa do sistema
6. Copiar o programa
7. Renomear o programa
8. Depurar o programa
9. Depurar o programa com variante
10. Gerenciar variante do programa

197

198

ABAP Report

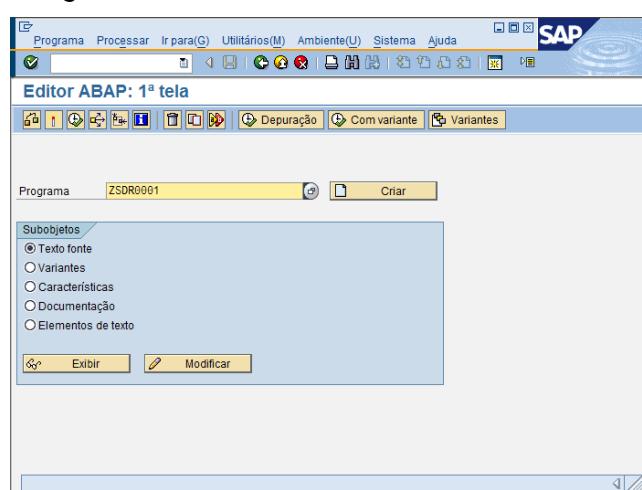
Características

- As principais características de um ABAP Report são:
 - É um programa executável
 - Possui tela de seleção, a qual é definida através de comandos ABAP

199

ABAP Report - Manutenção

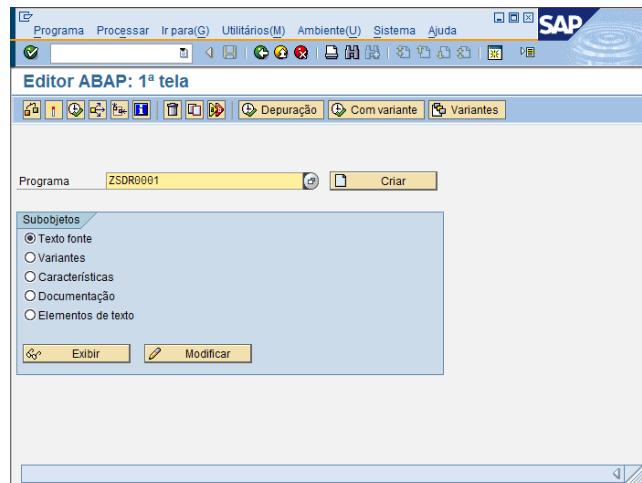
- A manutenção de programas ABAP Report é efetuado através da transação SE38
- As opções de manutenção são:
 - Texto fonte
 - Variantes
 - Características
 - Documentação
 - Elemento de Texto



200

ABAP Report - Texto Fonte

- Como o próprio nome já diz, texto fonte é onde é feito a criação/manutenção dos programas ABAP



201

ABAP Report - Variantes

- Variante é um grupo de valores de critério de seleção.
- Em vez de entrar os mesmos critérios de seleção cada vez que iniciar o programa de obtenção de dados, o usuário pode entrar e gravar em uma variante os valores que sempre utiliza como critérios de seleção
- É possível criar e especificar uma variante para o programa de obtenção de dados em cada visão de análise.

202

ABAP Report - Características

- Nas características do programa são definidos:
 - Título
 - Tipo de programa
 - Status
 - Aplicação
 - Grupo de autorizações
 - Pacote
 - Banco de dados lógico
 - Versão tela seleção
 - Bloqueio do editor
 - Verificação unicode ativas
 - Aritmética em ponto fixo
 - Início com variante

203

ABAP Report - Características

ABAP: características do programa ZSDR0001 modificar

Título	
Idioma original	PT Português
Criado	19.01.2009
Última modif.	19.01.2009
Status	inativo
Atributos	
Tipo	1 Programa executável
Status	
Aplicação	
Grupo autorizações	
Pacote	\$TMP Temporary Objects (never transported!)
Banco de dados lógico	
Versão tela seleção	
<input type="checkbox"/> Bloqueio do editor	<input checked="" type="checkbox"/> Aritmética em ponto fixo
<input type="checkbox"/> Verifs.unicode ativas	<input checked="" type="checkbox"/> Início via variante

Gravar

204

ABAP Report - Características

- Título
 - Descrição breve da funcionalidade do programa
- Tipo de programa
 - Neste campo é necessário informar o modo de execução do programa. As opções são:
 - 1. Programa executável: Declarar o programa como Executável, ou seja, que podem ser executados sem a necessidade de se criar transações para ele.
 - F. Grupo de funções: Declarar o programa como módulo de função. Utilizado pelo SAP na criação de módulo de função
 - I. Include: Declarar o programa como Include. Um include é um programa independente que possui duas funcionalidades:
 - Pode ser utilizado em diversos programas;
 - Facilita a manutenção dos programas.

205

ABAP Report - Características

- J: Pool de Interface: As interfaces são administradas pelo gerador de classes; a categoria de programa J não pode ser modificada nos atributos.
- K: Pool de classes: Contém classes. As classes são administradas pelo gerador de classes; a categoria de programa J não pode ser modificada nos atributos.
- M: Pool de módulos: Declarar o programa como Module Pool, significa que, este programa necessita da criação de transações ou menus para ser executado. Exemplo: Programas Diálogo.
- S: Pool de sub-rotinas: contém partes de programa (FORM) que são executáveis através das instruções PERFORM externas.

206

ABAP Report - Características

- Status
 - Descreve o status do programa. Exemplo: T Programa de testes
- Aplicação
 - Campo para alocação de programa em uma determinada área de negócio. Exemplo: Se o programa for para Financial Account, escolhe-se F.
- Grupo de Autorizações
 - Autorização para os usuários executarem o programa: S_DEVELOP e para manter o programa S_PROGRAM.

207

ABAP Report - Características

- Pacote
 - Os pacotes são importantes para transportes entre os sistemas de produção e desenvolvimento.
 - São combinados todos os objetos pertencentes ao desenvolvimento em uma classe de desenvolvimento em uma change request
 - Todos os programas associados a classe de desenvolvimento \$TMP são locais e não são transportados.
 - A classe de desenvolvimento é escolhida na criação do programa

208

ABAP Report - Características

- Banco de dados lógico
 - Somente para programas executáveis (Tipo 1). Este campo determina o Banco de Dados Lógico que será utilizado por este programa.
- Versão tela seleção
 - Versão de tela de seleção do programa de banco de dados
- Bloqueio do Editor
 - Quando marcado, permite somente o proprietário atualizar, apagar, renomear o programa.
 - Somente o proprietário poderá modificar e desbloquear o programa

209

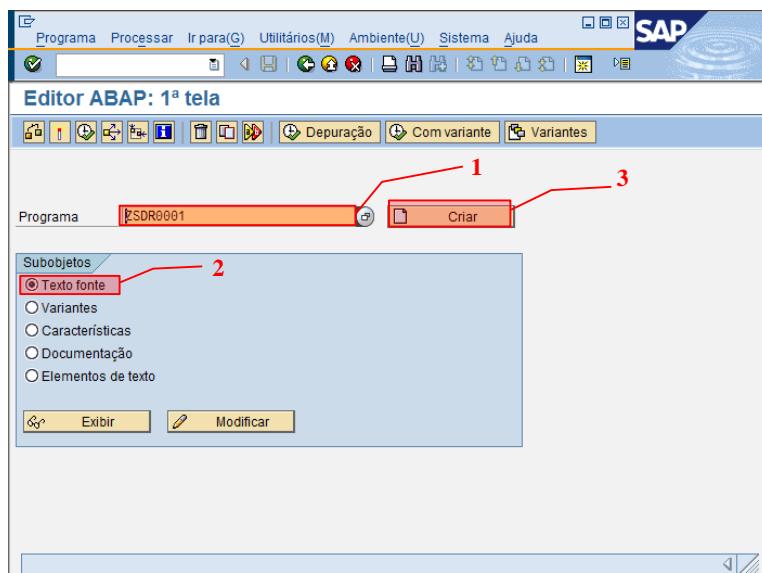
ABAP Report - Características

- Verificação de unicode ativas
 - Realiza a verificação de unicode
- Aritmética em ponto fixo
 - Se marcar este campo, o sistema arredonda os campos numéricos (tipo P) de acordo com o número de casas decimais especificados no programa, ou preenche-as com zeros, neste caso o sinal decimal é o ponto (.).
- Início via variante
 - Somente para programas executáveis. Informa que o programa só pode ser executado com uma Variant.

210

Report - Criação

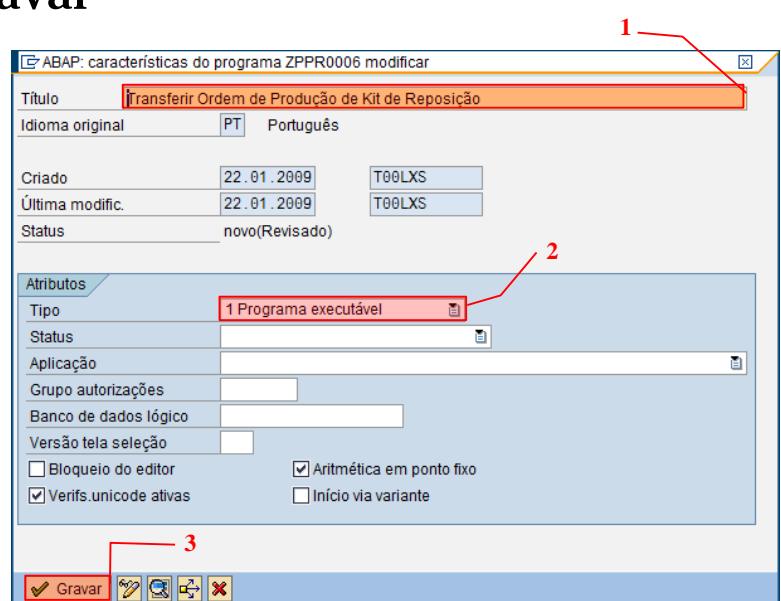
1. Inserir o nome do programa
2. Selecionar “Texto Fonte”
3. Clicar no botão “Criar”



211

Report - Criação

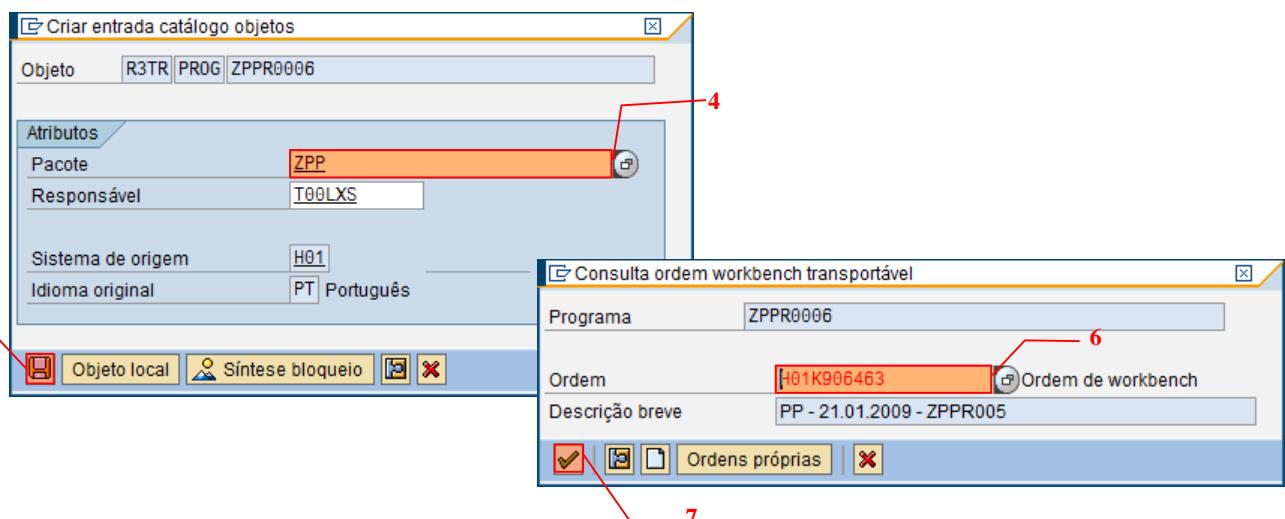
1. Inserir uma descrição breve do programa
2. Determinar o tipo de programa
3. Clicar no botão “Gravar”



212

Report - Criação

4. Inserir o pacote
5. Clicar no botão “Salvar”
6. Inserir o número da change request
7. Clicar no botão “Confirmar”



213

Report - Editor ABAP

The image shows the SAP ABAP editor interface with the title 'Editor ABAP: Report ZPPR0006 modificar'. The report code 'ZPPR0006' is selected in the 'Report' dropdown. The code area contains the following ABAP code:

```

REPORT  ZPPR0006.

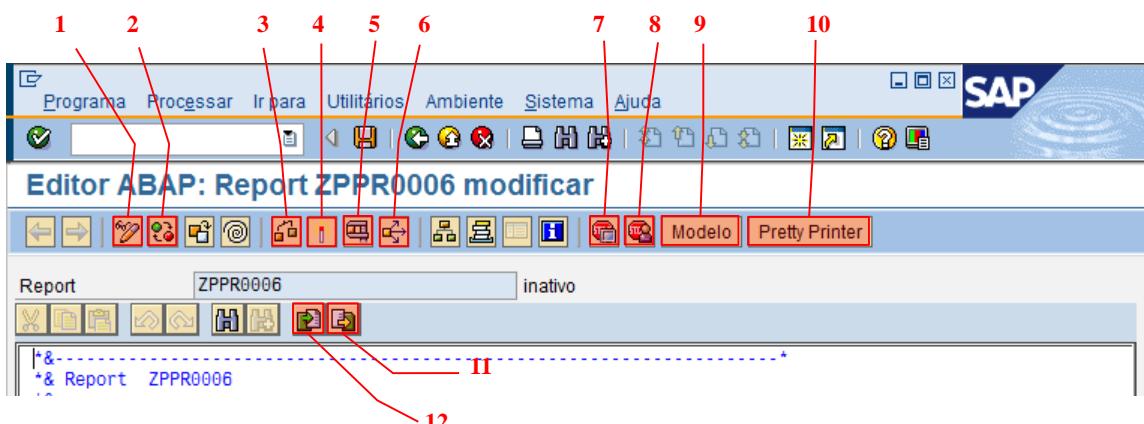
```

The status bar at the bottom indicates 'Linha 1, Coluna 1' and 'Linha 1 - Linha 10 de 10 linhas'.

214

Editor ABAP

- | | | | |
|----|--|-----|--|
| 1. | Exibir/Modificar (CTRL+F1) | 7. | Definir/Eliminar ponto de parada de sessão |
| 2. | Ativar/Desativar o programa (SHIFT+F8) | 8. | Definir/Eliminar ponto de parada externo |
| 3. | Verificar o código (CTRL+F2) | 9. | Modelo |
| 4. | Ativar o programa (CTRL+F3) | 10. | Pretty Printer |
| 5. | Executar o programa (F8) | 11. | Exportar o código |
| 6. | Lista de Utilização (CTRL+SHIFT+F3) | 12. | Importar o código |



215

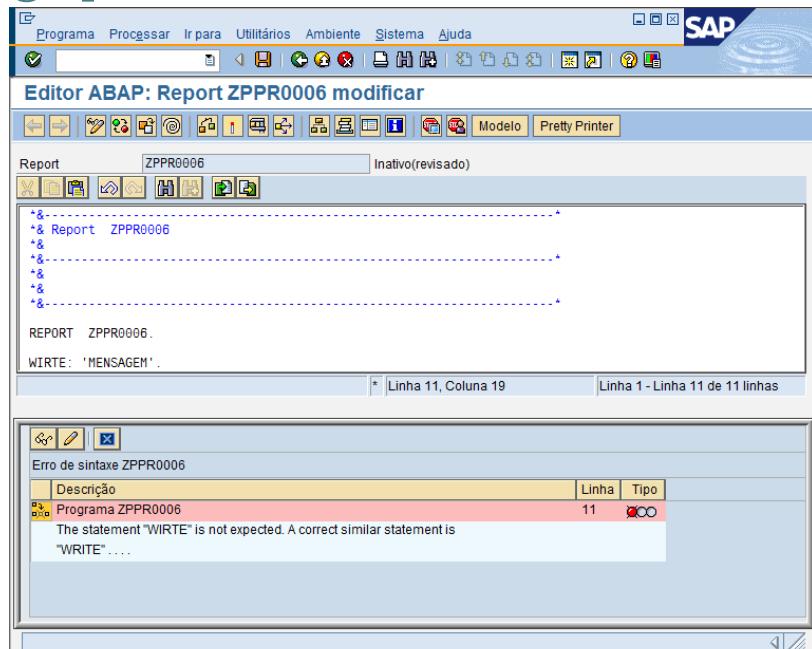
Editor ABAP - Funcionalidades

- Exibir/Modificar (CTRL+F1)
 - Alterna o editor para o modo de visualização e edição do código
- Ativar/Desativar o programa (SHIFT+F8)
 - Altera o status do programa para ativo ou inativo
- Ativar o programa (CTRL+F3)
 - Ativa o programa tornado-o possível para a execução do usuário. Na ativação, o código é verificado a procura de erros

216

Editor ABAP - Funcionalidades

- Verificar o código (CTRL+F2)
 - Verifica se o código possui erro de sintaxe



217

Editor ABAP - Funcionalidades

- Executar o programa (F8)
 - Executa o programa, desde que o mesmo seja do tipo 1
 - Programa Executável e o programa esteja ativo.
- Lista de Utilização (CTRL+SHIFT+F3)
 - Retorna uma lista informando quais os programas, transações, classes e ampliações utilizam o programa pesquisado.
- Definir/Eliminar ponto de parada de sessão
 - Insere um ponto de parada para que o programa seja interrompido em modo de depuração para a solução de problemas. O ponto de parada é utilizado em todas as sessões do logon atual.

218

Editor ABAP - Funcionalidades

- **Definir/Eliminar ponto de parada externo**
 - Insere um ponto de parada válido para todos os usuários, do servidor e sistema atual.
 - Os pontos de parada externos são úteis para debugar programas BSP ou WebDynpro. Nesse caso, o logon não é realizado através do SAP GUI, mas sim do browser, por exemplo. Entretanto, os pontos de parada devem ser ativados antes de logar.
 - O uso do ponto de parada externo é válido pelo período de 2 horas.
- **Modelo**
 - O botão modelo é uma ferramenta para inserir determinados comandos como modelo, a modo de facilitar a codificação pelo desenvolvedor.

219

Editor ABAP - Funcionalidades

- **Pretty Printer**
 - O botão “Pretty Printer” é utilizado para uniformizar o layout do programa, tal como endentação e capitulação.
- **Importar/Exportar código**
 - Importa/Exporta o código fonte de/para um arquivo texto.

220

ABAP Report

Construção de um programa ABAP Report

Comando REPORT

- Todo programa ABAP Report é iniciado com o comando REPORT.
- A sintaxe do comando REPORT é:

```
REPORT rep [list-options]
           [MESSAGE-ID mid]
           [DEFINING DATABASE ldb]
```

- As opções de list-options são:
 - NO STANDARD PAGE HEADING
 - Esta opção suprime a saída do cabeçalho padrão da página.

Comando REPORT

- LINE-SIZE *largura*

- Esta opção define a largura da linha da listagem com a quantidade de caracteres definido em *largura*.
- A largura máxima é de 1023 caracteres.
- Quando a opção LINE-SIZE não é especificada, a largura da listagem se ajustará automaticamente ao tamanho da janela.

223

Comando REPORT

- LINE-COUNT *page_lines* [(*footer_lines*)]

- Esta opção define a altura da página em número de linhas.
- Se a opção LINE-COUNT não for especificada, a altura é definida internamente em 60.000 linhas
- O parâmetro *footer_lines* define o número de linhas reservadas para o rodapé da página.

224

Comando Report

- **MESSAGE_ID**
 - A opção MESSAGE-ID é para especificar a classe de mensagens, a qual permite facilitar o uso do comando MESSAGE, na qual somente o tipo e o número da mensagem são especificados.
- **DEFINING DATABASE ldb**
 - A opção DEFINING DATABASE ldb apresenta o programa atual como um programa de banco de dados do banco de dados lógico ldb.
 - O banco de dados lógico são definidos na transação SE36.

225

Comando Report - Exemplos

Define a classe de mensagens SD, com a largura da lista de 100 caracteres e sem cabeçalhos

```
REPORT zsdr0001 MESSAGE-ID SD  
      LINE-SIZE 100  
      NO STANDARD PAGE HEADING.
```

Define o banco de dados lógico F1S

```
REPORT zfir0001 DEFINING DATABASE F1S.
```

A altura da página é de 65 linhas, sendo 8 reservados para o rodapé da página. A largura da página é 132 caracteres.

```
REPORT zmmr0001 LINE-COUNT 65(8) LINE-SIZE 132.
```

226

ABAP Report

Tela de Seleção

Tela de seleção

- A tela de seleção fornece uma interface padronizada para os usuários.
- Os usuários podem inserir valores simples ou complexos.
- As opções de entrada são campos, radio-buttons e check-boxes.

Tela de seleção - Definição

- Existem três comando ABAP para definir a tela de seleção:
 - PARAMETERS para campos simples
 - SELECT-OPTIONS para campos complexos
 - SELECTION-SCREEN para formatar a tela de seleção
- Por padrão, quando o programa do tipo Report é definido, o sistema cria automaticamente a tela 1000, quando a tela de seleção é especificada.

229

Tela de seleção - Parameters

- O comando PARAMETERS permite a criação de um campo de entrada de um único valor.
- Cada comando PARAMETERS declarado no programa cria um campo de entrada na tela de seleção.

230

Tela de seleção - Parameters

- A sintaxe do comando PARAMETERS é:

```
PARAMETERS <p>[(<length>)] [TYPE <type>|LIKE <obj>] [DECIMALS <d>].
```

- Este comando criar o parameter <p>. O nome do parameter está limitado a oito posições.
- Os parâmetros LENGTH, TYPE/LIKE, DECIMALS é o mesmo que o comando de declaração DATA.
- Os dados de declaração dos parameters inclui todos os tipos elementares do ABAP com exceção do tipo F, referência e tipos agregados.

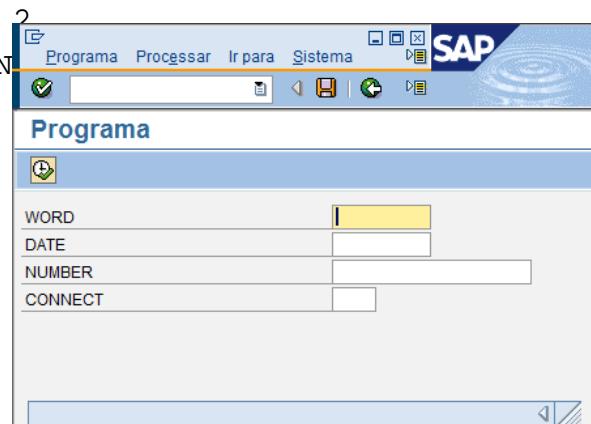
231

Tela de seleção - Parameters

Exemplo:

```
PARAMETERS: WORD(10) TYPE C,
            DATE TYPE D,
```

```
NUMBER TYPE P DECIMALS 2
CONNECT TYPE SPFLI-CONN
```



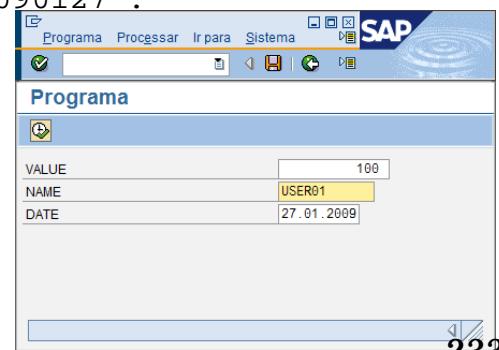
232

Tela de seleção - Parameters

- Valores padrão do Parameters
 - Para atribuir um valor padrão ao Parameters, é necessário utilizar o parâmetro DEFAULT

Exemplo

```
PARAMETERS: VALUE TYPE I DEFAULT 100,  
           NAME LIKE SY-UNAME DEFAULT SY-UNAME,  
           DATE LIKE SY-DATUM DEFAULT '20090127'.
```



233

Tela de seleção - Parameters

- Valores em maiúsculo e minúsculo
 - Para alterar o valor dos parameters para maiúsculo ou minúsculo, é necessário utilizar o parâmetro LOWER CASE.
 - Se a opção não for especificada o valor será convertido automaticamente para maiúsculo

Tela de seleção - Parameters

- Reduzir o tamanho visível do Parameters
 - Por padrão, o tamanho do campo de entrada na tela de seleção é o mesmo do parameter do programa ABAP. Entretanto, é possível diminuir o tamanho visível do campo através o parâmetro VISIBLE LENGTH

Exemplo

```
PARAMETERS: p1(10) TYPE c VISIBLE LENGTH 1,  
           p2(10) TYPE c VISIBLE LENGTH 5,  
           p3(10) TYPE c VISIBLE LENGTH 10.
```

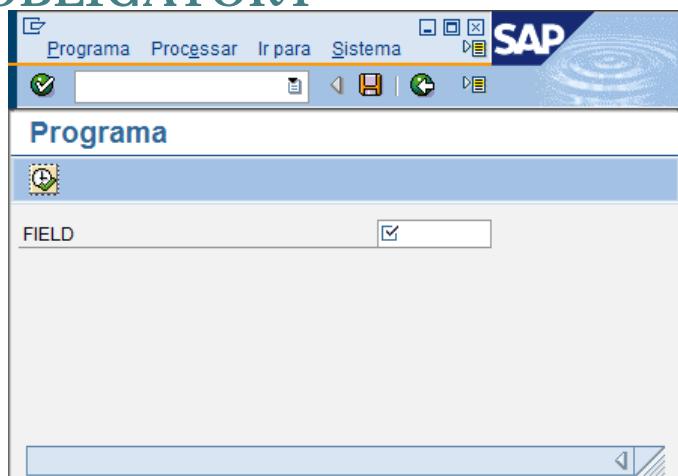
235

Tela de seleção - Parameters

- Tornar o campo obrigatório
 - O campo pode ser definido como obrigatório através do parâmetro OBLIGATORY

Exemplo

```
PARAMETERS FIELD(10) OBLIGATORY.
```



236

Tela de seleção - Parameters

- Definir a ajuda de pesquisa para o Parameters
 - Para definir a ajuda de pesquisa no Parameters, será associado a ajuda de pesquisa do dicionário de dados. A sintaxe é:

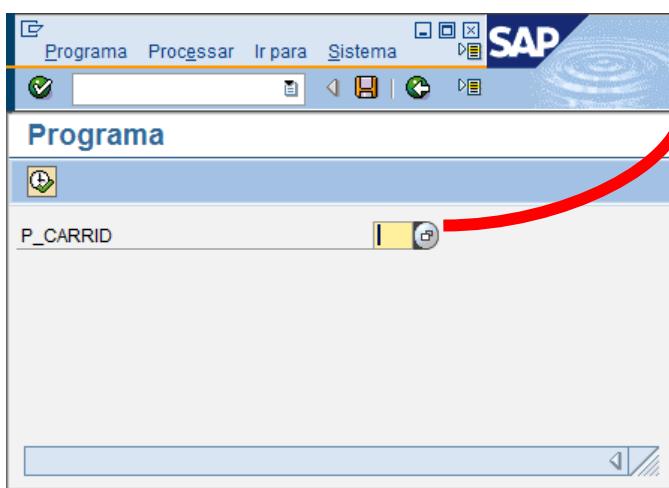
```
PARAMETERS <p> ... MATCHCODE OBJECT <search_help>.
```

237

Tela de seleção - Parameters

Exemplo

```
PARAMETERS p_carrid TYPE s_carr_id
           MATCHCODE OBJECT demo_f4_de.
```



F4



238

Tela de seleção - Parameters

- Verificar o conteúdo do campo
 - Para verificar o conteúdo do campo, é necessário utilizar o parâmetro VALUE CHECK
 - O parameter deve estar associado a um campo de uma tabela, visão e estrutura
 - Se o valor inserido não estiver inserido na tabela, por exemplo, o sistema retorna uma mensagem de erro e o programa é impedido de continuar.

Exemplo

```
PARAMETERS P_CARR LIKE SPFLI-CARRID OBLIGATORY VALUE CHECK.
```

239

Tela de seleção - Parameters

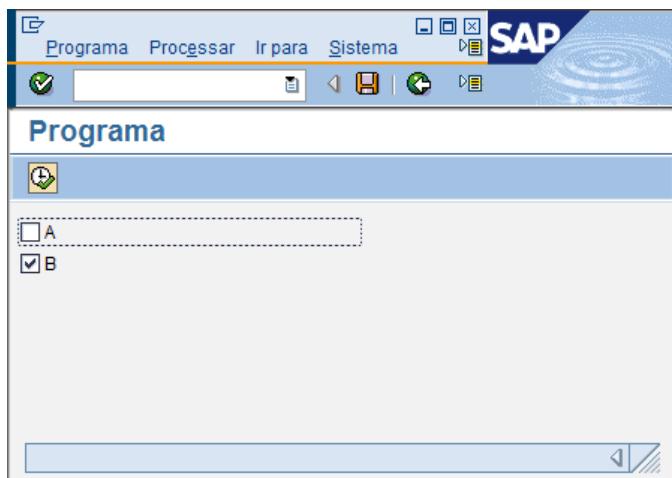
- Checkbox
 - Para definir um campo como checkbox é necessário utilizar o parâmetro AS CHECKBOX
 - O parâmetro é criado com o tipo C e tamanho 1. Neste caso, não pode ser utilizados os comandos TYPE e LIKE.
 - Os valores do Parameters são ‘‘ (vazio) e X.

240

Tela de seleção - Parameters

Exemplo

PARAMETERS: A AS CHECKBOX,
B AS CHECKBOX DEFAULT 'X'.



241

Tela de seleção - Parameters

- **Radio Buttons**

- Para definir um Parameters como radio button, é necessário utilizar o parâmetro **RADIOBUTTON GROUP**.

PARAMETERS <p> RADIOBUTTON GROUP <radi>.....

- O parameters <p> é criado com o tipo C e tamanho 1, é atribuído ao grupo <radi>. O tamanho máximo da string <radi> é 4. Os parâmetros de atribuição TYPE e LIKE podem ser utilizados, mas deve referir-se a um campo do tipo C de tamanho 1.
- Deve ser atribuído no mínimo 2 campos para cada grupo <radi>.

242

Tela de seleção - Parameters

Exemplo

```
PARAMETERS: R1 RADIobutton GROUP RAD1,  
R2 RADIobutton GROUP RAD1 DEFAULT 'X',  
R3 RADIobutton GROUP RAD1,  
S1 RADIobutton GROUP RAD2,  
S2 RADIobutton GROUP RAD2,  
S3 RADIobutton GROUP RAD2 DEFALUT 'X'.
```



243

Tela de seleção - Parameters

- Ocultando campo de entrada
 - Para ocultar o campo de entrada na tela de seleção, é necessário utilizar o parâmetro NO-DISPLAY.

244

Tela de seleção - Select-Options

- O comando SELECT-OPTIONS cria um campo que permite ao usuário inserir um conjunto de valores e inserir os critérios de seleção.
- Ao contrário do Parameters em que o campo da tela está associado a uma variável, o campo do select-options está associado a uma tabela interna especial chamada tabela de seleção.

245

Tela de seleção - Select-Options

- A sintaxe para criar o select options é:

```
SELECT-OPTIONS <seltab> for <f>
```

- É declarado uma tabela de seleção<seltab> que está associado a couna <f> de uma tabela de banco de dados.
- A tabela de seleção é utilizada de vários modos, mas o propósito principal é traduzir diretamente o critério de seleção nas seleções de banco de dados usando a cláusula WHERE do comando SQL

246

Tela de seleção - Select-Options

- Estrutura da tabela de seleção
 - O tipo de linha de uma tabela de seleção é uma estrutura que possui os seguintes componentes:
 - SIGN
 - O tipo de dados do campo SIGN é C de tamanho 1. O conteúdo do campo SIGN determina para cada linha se o resultado da linha é para ser incluída ou excluída. Os valores possíveis são I (Inclusive) e E (exclusive)

247

Tela de seleção - Select-Options

- Option
 - O tipo de dados do campo Option é C de tamanho 2. Option contém o operador de seleção e os valores disponíveis são: EQ, NE, GT, LE, LT, CP e NP caso o campo HIGH não esteja preenchido. Se o campo HIGH estiver preenchido os operandos disponíveis são: BT e NB.
- LOW
 - O tipo de dados do campo LOW é do mesmo tipo do tipo de coluna da tabela de banco de dados a qual a tabela de seleção está associada.
 - Se o campo HIGH estiver vazio, o conteúdo do campo LOW define uma simples comparação de campo. Em combinação com o operador especificado no campo OPTION, é especificado uma condição para a seleção do banco de dados.

248

Tela de seleção - Select-Options

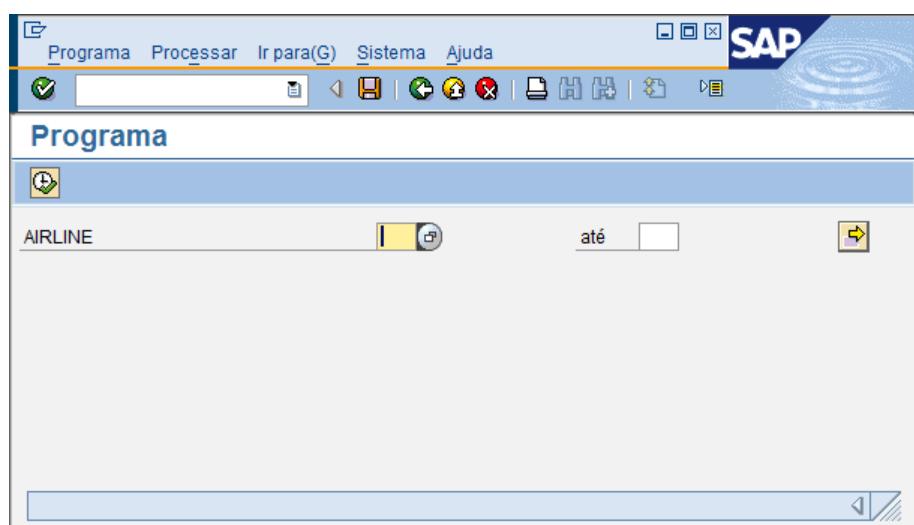
- Se o campo HIGH conter valor, o conteúdo dos campos LOW e HIGH definem os limites inferior e superior de um intervalo. Em combinação com o operador do campo OPTION, o intervalo especifica uma condição de seleção do banco de dados.

249

Tela de seleção - Select-Options

Exemplo:

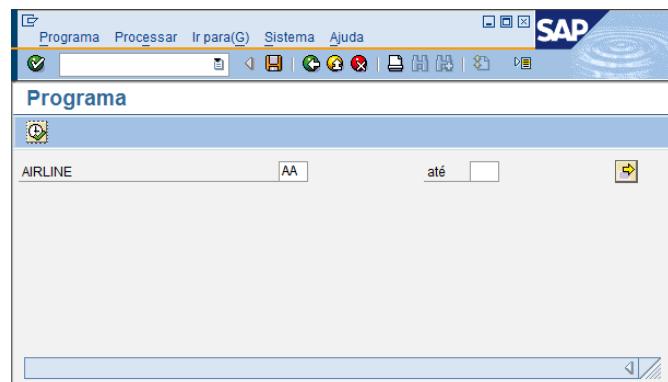
```
DATA WA_CARRID TYPE SPFLI-CARRID.  
SELECT-OPTIONS AIRLINE FOR WA_CARRID.
```



250

Tela de seleção - Select-Options

Se o usuário entrar com o valor abaixo:



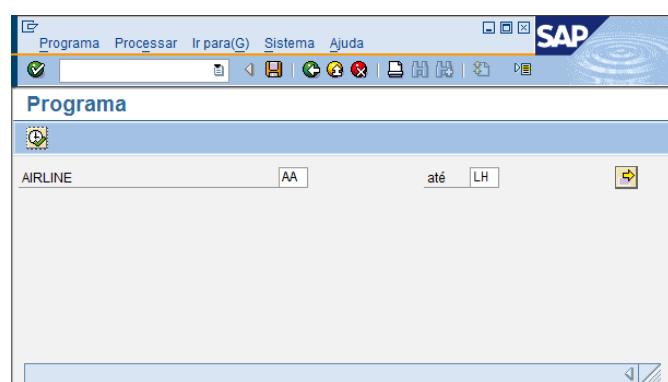
SIGN	OPTION	LOW	HIGH
I	EQ	AA	

O conteúdo da tabela de seleção será:

251

Tela de seleção - Select-Options

Se o usuário entrar os valores abaixo:



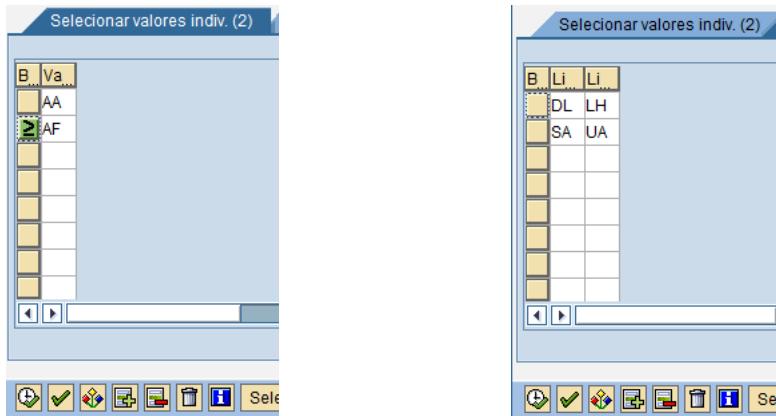
SIGN	OPTION	LOW	HIGH
I	EQ	AA	LH

O conteúdo da tabela de seleção será:

252

Tela de seleção - Select-Options

Se o usuário clicar no botão preencher as telas abaixo:



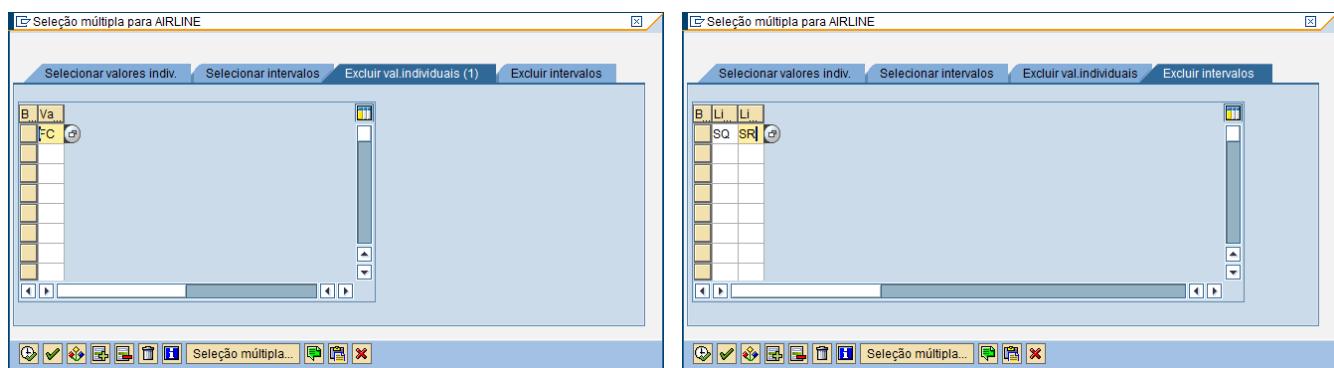
O conteúdo da tabela de seleção será:

SIGN	OPTION	LOW	HIGH
I	EQ	AA	
I	GE	AF	
I	BT	DL	LH
I	BT	AS	UA

253

Tela de seleção - Select-Options

Se o usuário clicar no botão preencher as telas abaixo:



O conteúdo da tabela de seleção será:

SIGN	OPTION	LOW	HIGH
E	EQ	FC	
E	BT	SQ	SR

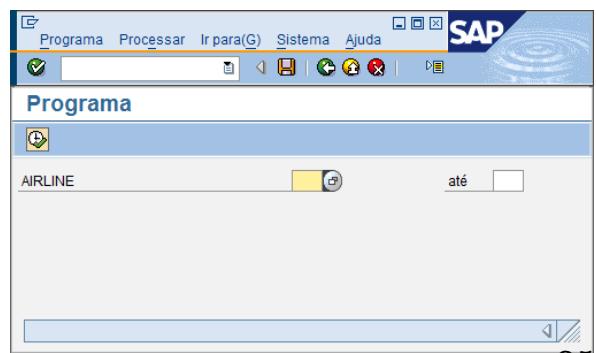
254

Tela de seleção - Select-Options

- Restringir a seleção a uma linha
 - Para permitir o usuário processar somente a primeira linha da tabela de seleção na tela de seleção, é necessário utilizar o parâmetro NO-EXTENSION.

```
DATA WA_SPFLI TYPE SPFLI.
```

```
SELECT-OPTIONS AIRLINE FOR WA_SPFLI-CARRID NO-EXTENSION.
```

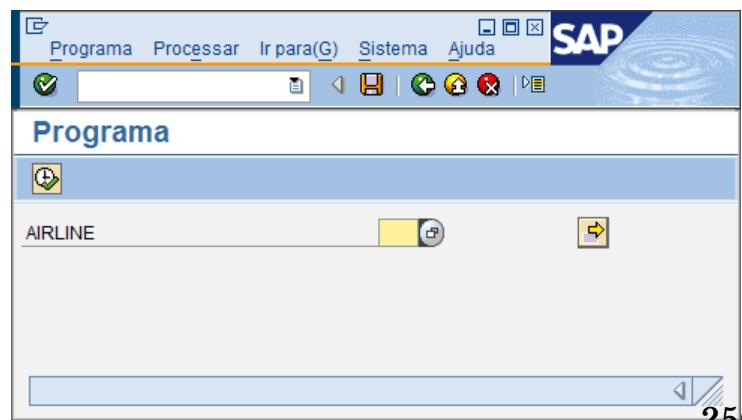


Tela de seleção - Select-Options

- Restringir a entrada a um único campo
 - Para permitir ao usuário processar somente um único campo da tela de seleção, é necessário utilizar o parâmetro NO INTERVALS.

```
DATA WA_SPFLI TYPE SPFLI.
```

```
SELECT-OPTIONS AIRLINE FOR WA_SPFLI-CARRID NO INTERVAL.
```



Tela de seleção - Selection-Screen

- O comando SELECTION-SCREEN possui opções de formatação que podem ser utilizadas para definir o layout da tela de seleção.
- Podem ser definidos o layout dos parameters e critério de seleção e mostrar linhas de comentários e underlines na tela de seleção.

257

Tela de seleção - Selection-Screen

- Linhas em branco
 - O comando abaixo cria <n> linhas em branco, onde <n> pode ser um valor de 1 a 9. Se o valor <n> for omitido, será criada somente uma linha em branco.
`SELECTION-SCREEN SKIP [<n>].`

258

Tela de seleção - Selection-Screen

- Underlines

- O comando abaixo cria uma underline. Se a opção <pos(len)> não for utilizada, uma nova linha é criada abaixo da linha atual. Se a opção <pos(len)> for utilizada, o underline começa na posição <pos> na linha atual e continua com o tamanho <len> de posições. Com vários elementos em uma linha a opção (<LEN>) pode ser utilizada sem o <pos>. A barra produz um line feed.
- O MODIF ID tem o mesmo comportamento do comando PARAMETERS. Pode ser usado para modificar a linha antes da tela de seleção ser chamada.

```
SELECTION-SCREEN ULINE [ [ / ]<pos(len)> ] [MODIF ID <key>].
```

259

Tela de seleção - Selection-Screen

- Comentários

- O comando abaixo insere um comentário na tela de seleção. Para <comm>, pode ser especificado um símbolo de texto ou um nome de campo com o tamanho máximo de oito posições. O campo não pode ser declarado com o comando DATA, mas é gerado automaticamente com o tamanho <len>. O campo deve ser preenchido antes da chamada da tela de seleção.
- A opção <pos(les)> deve ser sempre especificada, se não houver mais de um elemento na mesma linha.
- A barra (/) insere o comentário na próxima linha.
- A opção FOR FIELD deve ser utilizada para atribuir o campo ao comentário.

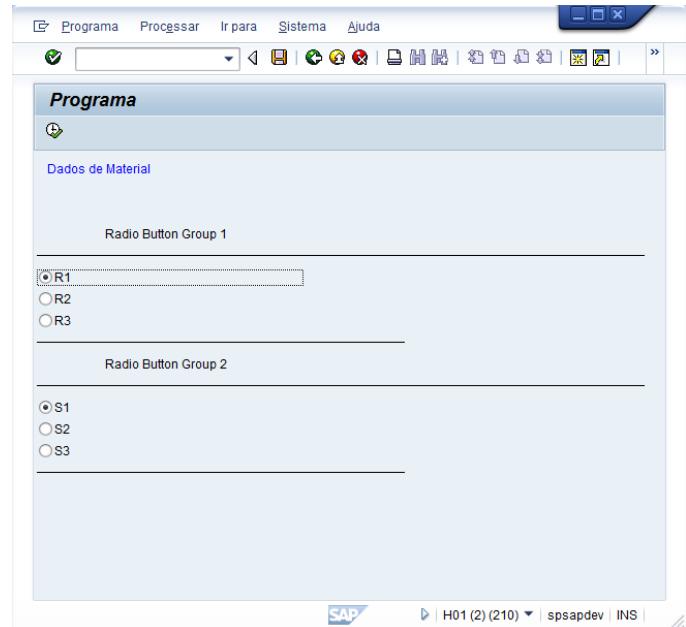
```
SELECTION-SCREEN COMMENT [ / ]<pos(len)> <comm> [FOR FIELD <f>]  
[MODIF ID <key>].
```

260

Tela de seleção - Selection-Screen

- Exemplo

```
SELECTION-SCREEN COMMENT /2(50) TEXT-001 MODIF ID SC1.
SELECTION-SCREEN SKIP 2.
SELECTION-SCREEN COMMENT /10(30) COMM1.
SELECTION-SCREEN ULINE.
PARAMETERS: R1 RADIobutton GROUP RAD1,
R2 RADIobutton GROUP RAD1,
R3 RADIobutton GROUP RAD1.
SELECTION-SCREEN ULINE /1(50).
SELECTION-SCREEN COMMENT /10(30) COMM2.
SELECTION-SCREEN ULINE.
PARAMETERS: S1 RADIobutton GROUP RAD2,
S2 RADIobutton GROUP RAD2,
S3 RADIobutton GROUP RAD2.
SELECTION-SCREEN ULINE /1(50).
INITIALIZATION.
COMM1 = 'Radio Button Group 1'.
COMM2 = 'Radio Button Group 2'.
LOOP AT SCREEN.
IF SCREEN-GROUP1 = 'SC1'.
SCREEN-INTENSIFIED = '1'.
MODIFY SCREEN.
ENDIF.
ENDLOOP.
```



261

Tela de seleção - Selection-Screen

- Vários elementos em uma linha

- Para posicionar um conjunto de parametros ou comentários em uma linha na tela de seleção, os elementos devem ser declarados em um bloco através dos seguintes comandos:

SELECTION-SCREEN BEGIN OF LINE.

...
SELECTION-SCREEN END OF LINE.

262

Tela de seleção - Selection-Screen

- O texto de seleção não é apresentado quando está opção é utilizada. Para descrever os elementos de texto, deve ser utilizado o parâmetro COMMENT do comando SELECTION-SCREEN.
- Para determinar a posição de um elemento em uma linha, deve ser utilizado o seguinte comando:

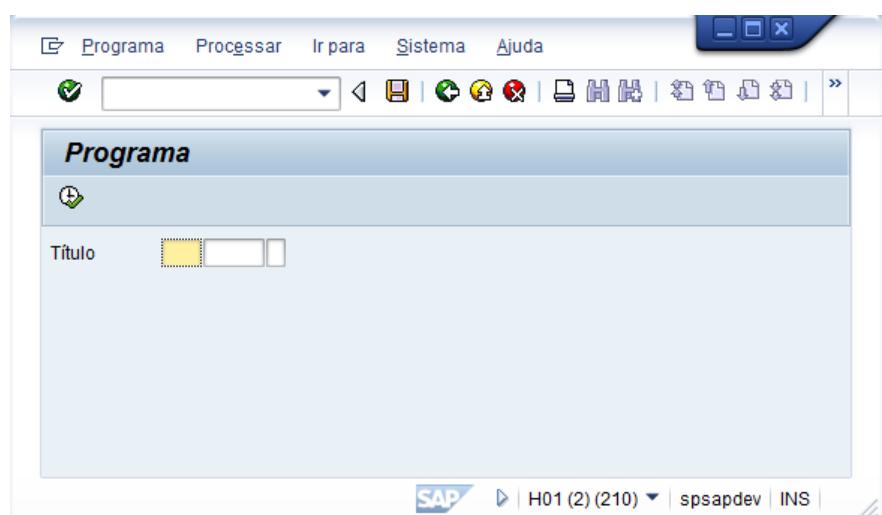
```
SELECTION-SCREEN POSITION <pos>.
```

263

Tela de seleção - Selection-Screen

- Exemplo 1

```
SELECTION-SCREEN BEGIN OF LINE.  
SELECTION-SCREEN COMMENT 1(10) TEXT-001.  
PARAMETERS: P1(3), P2(5), P3(1).  
SELECTION-SCREEN END OF LINE.
```



264

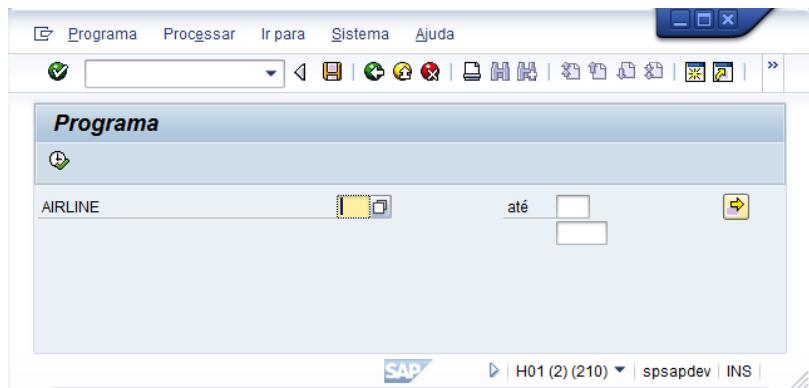
Tela de seleção - Selection-Screen

- Exemplo 2

```

REPORT DEMO.
DATA WA_SPFLI TYPE SPFLI.
SELECT-OPTIONS AIRLINE FOR WA_SPFLI-CARRID.

SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN POSITION POS_HIGH.
PARAMETERS FIELD(5).
SELECTION-SCREEN END OF LINE.
```



265

Tela de seleção - Selection-Screen

- Bloco de elementos

- Para criar um bloco lógico de elementos na tela de seleção, utilize o comando abaixo:

```

SELECTION-SCREEN BEGIN OF BLOCK <block>
[WITH FRAME [TITLE <title>]]
[NO INTERVALS].
...
SELECTION-SCREEN END OF BLOCK <block>.
```

- Deve ser definido um nome <block> para cada bloco. Os elementos de bloco podem ser aninhados.

266

Tela de seleção - Selection-Screen

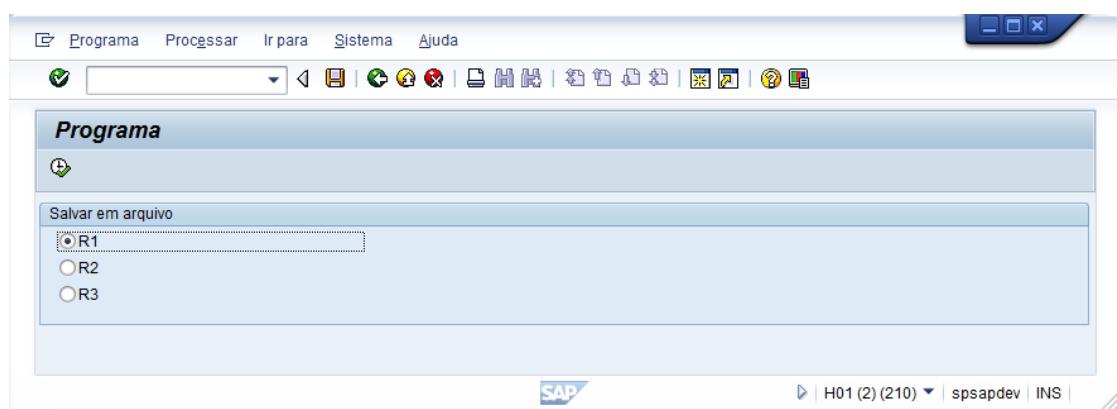
- Se a opção WITH FRAME for utilizado um quadro é desenhado em volta do bloco.
- Se o TITLE for utilizado, é apresentado o título para cada bloco.
- Na opção NO INTERVALS, todos os campos definidos pelo comando SELECT-OPTIONS são tratados para restringir os dados como valores únicos, sem intervalos.

267

Tela de seleção - Selection-Screen

- Exemplo

```
SELECTION-SCREEN BEGIN OF BLOCK RAD1
WITH FRAME TITLE TEXT-002.
PARAMETERS R1 RADIOPUSHBUTTON GROUP GR1.
PARAMETERS R2 RADIOPUSHBUTTON GROUP GR1.
PARAMETERS R3 RADIOPUSHBUTTON GROUP GR1.
SELECTION-SCREEN END OF BLOCK RAD1.
```

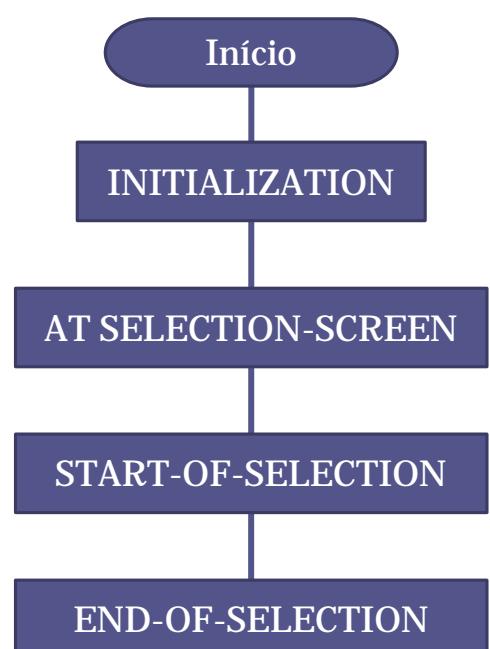


ABAP Report

Eventos

Eventos

- Quando um programa executável é acionado, o fluxo do programa é controlado pelos eventos externos no ambiente de execução ABAP.
- O diagrama apresenta a seqüência dos eventos:



Eventos

- O evento AT SELECTION-SCREEN só é processado se uma tela de seleção é definido no programa. Os outros eventos são acionados quando qualquer programa executável é acionado.
- Bem como esses eventos, há outros eventos que podem ser acionados quando quando uma listagem é criada (TOP-OF-PAGE, END-OF-PAGE) e em listas interativas (AT LINE-SELECTION, AT USER-COMMAND)

271

Eventos - Descrição

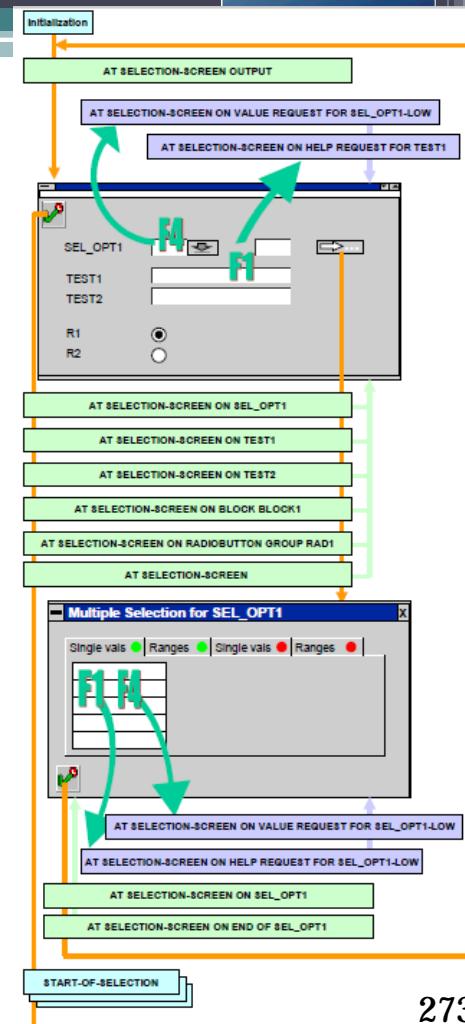
- INITIALIZATION
 - Esse evento é acionado antes do processamento da tela de seleção. Pode ser utilizado para inicializar os valores dos parameters e select-options da tela de seleção.

272

Eventos - Descrição

- AT SELECTION-SCREEN

- Esse evento é a forma básica de toda uma série de eventos que ocorrem enquanto a tela de seleção está sendo processada
- A tela de seleção em um programa executável é automaticamente chamado entre os eventos INITIALIZATION e START-OF-SELECTION. Quando a tela de seleção é chamada e o usuário interage com ela, o ambiente de execução ABAP cria os eventos de seleção, os quais ocorrem entre os eventos INITIALIZATION e START-OF-SELECTION.



273

Eventos - Descrição

TRAINNING
Education Services

- START-OF-SELECTION

- Este evento é executado depois do processamento da tela de seleção. Pode ser utilizado para preparar a leitura e preparação dos dados para a criação de uma lista.

- END-OF-SELECTION

- Este é o último evento chamado pelo ambiente de execução. É ativado depois que todos os dados tenham sido lidos e processados e antes que o processador de listagem seja iniciado.

274

Eventos - Sair do evento

- Para sair do evento, basta utilizar um dos comandos abaixo:
 - **STOP**
 - O comando STOP, só pode ser utilizado nos eventos AT SELECTION-SCREEN (sem parâmetros) e START-OF-SELECTION. Ao usar o comando STOP, o evento atual é finalizado e o evento END-OF-SELECTION é ativado.
 - **EXIT**
 - Ao utilizar o comando EXIT, o ambiente de execução não chama mais nenhum evento e chama o processador de listagem para apresentar a listagem básica.

275

Eventos - Sair do evento

- Para sair do evento, basta utilizar um dos comandos abaixo:
 - **RETURN**
 - O comando return tem o mesmo procedimento do comando EXIT.
 - **CHECK**
 - O comando check é utilizado junto com uma lógica de teste. Se a condição for falsa, o evento é finalizado.

276

Curso Avançado de ABAP

Module Pool

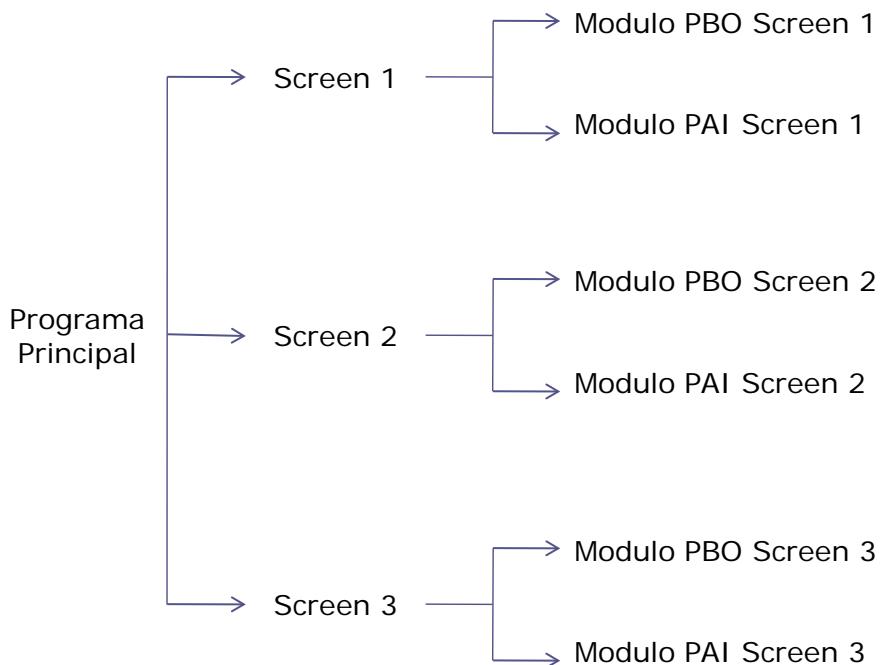


Module Pool



- **Module Pool** são programas que funcionam on-line, ou seja, instantaneamente
- Maior capacidade de customização de telas e maior flexibilidade de criação de interface para o usuário
- Útil na criação de aplicações que necessitam de subtelas ou fornecer informação de forma diferente de um relatório.

Estrutura do programa



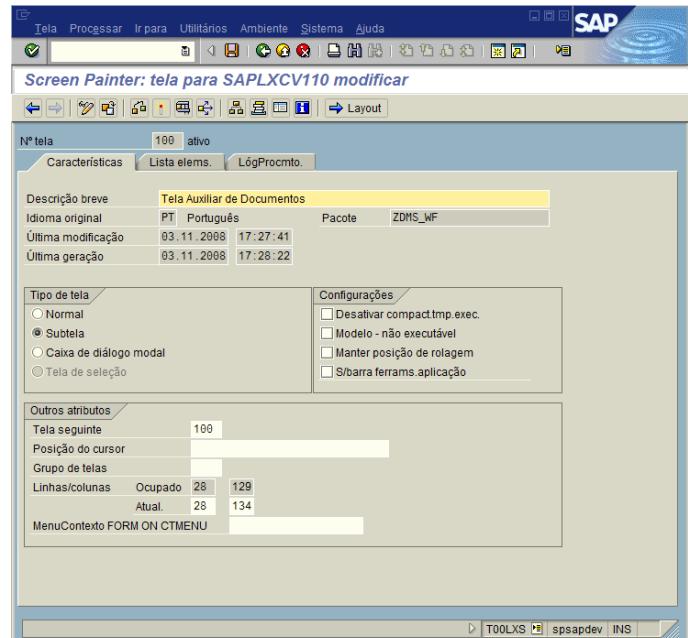
279

Screen

- Desenho da Tela
- Lista de campos para tratamento de erros
- Definição dos módulos PBO e PAI relacionados com a tela

Screen - Características

- Definição do nome da tela, tipo de tela e tamanho



281

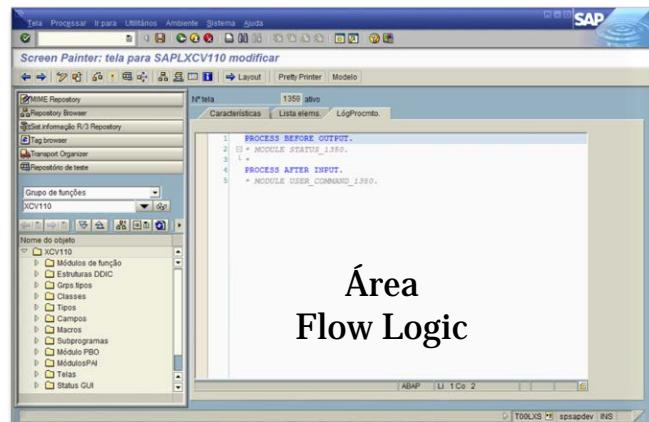
Status GUI e Title GUI

- Definição dos comandos de botão e menus disponíveis
 - Definição do título de tela
 - Habilitação dos comandos standards existentes
 - Criação de novos comandos de botão, menus e teclas de atalho

282

Flow Logic

- O Flow Logic é a área responsável pelo controle do fluxo da tela
- Nessa área é especificado os eventos PBO, PAI e POV



283

Flow Logic

- Comandos do Flow Logic
 - O flow logic só aceita os comandos abaixo

CALL	Chama uma subtela
CHAIN	Inicia o processamento chain.
ENDCHAIN	Finaliza o processamento chain.
ENDLOOP	Finaliza o processamento em LOOP
FIELD	Referência a um campo. É possível combinar com os comandos MODULE e SELECT
LOOP	Inicia o processamento em LOOP
MODIFY	Modifica uma tabela
MODULE	Identifica um módulo de processamento
ON	Utilizado com o comando FIELD
PROCESS	Define um evento de processamento
SELECT	Verifica um entrada em uma tabela
VALUES	Define os valores de entrada permitido

284

Flow Logic

```
PROCESS BEFORE OUTPUT.
```

```
  MODULE STATUS_0100.
```

```
PROCESS AFTER INPUT.
```

```
  MODULE USER_COMMAND_0100.
```

```
PROCESS ON VALUE-REQUEST.
```

```
  MODULE USER_COMMAND_0100.
```

285

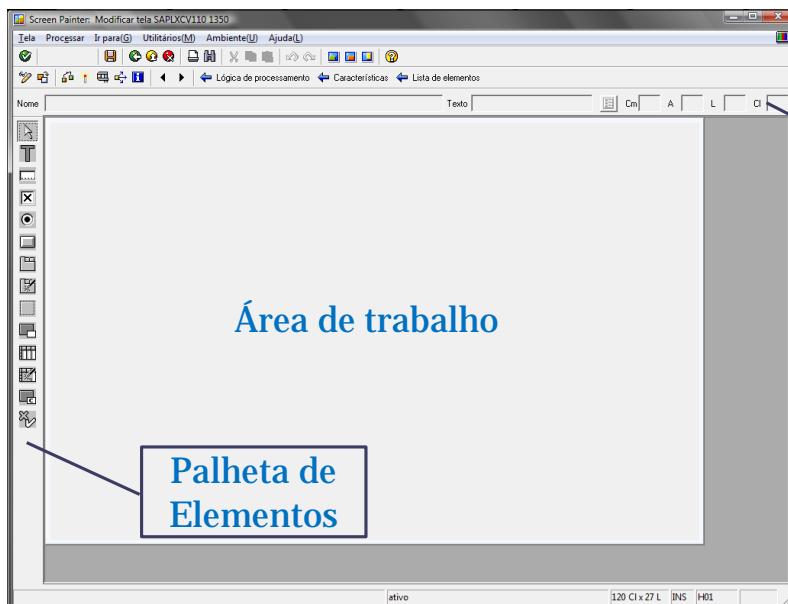
Screen Painter

- O screen-painter é uma ferramenta do ABAP Workbench que permite a criação de telas para uma transação
- É utilizado para criar telas com campos ou outro elemento gráfico e codificar a lógica do fluxo por trás da tela

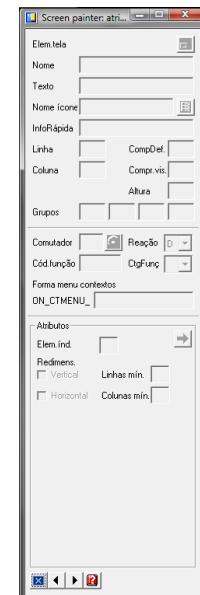
286

Screen Painter

Tela do Screen painter



Janela de Atributos

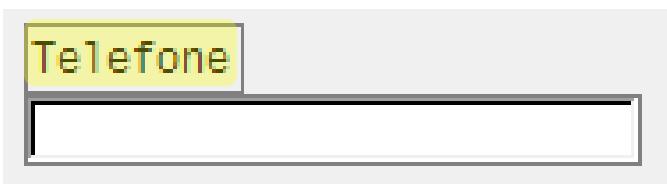


287

Palheta de Elementos

T Campo Texto

- Insere um campo de saída de texto, o qual é utilizado como título de um objeto como um campo de entrada e saída



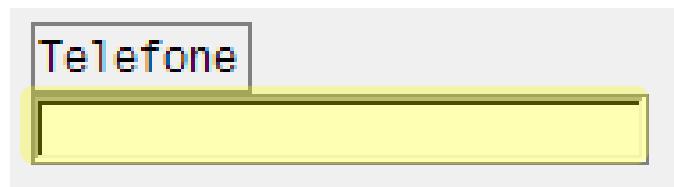
288

Palheta de Elementos



Campo de Entrada/Saída

- É um campo utilizado para a entrada de dados pelo usuário



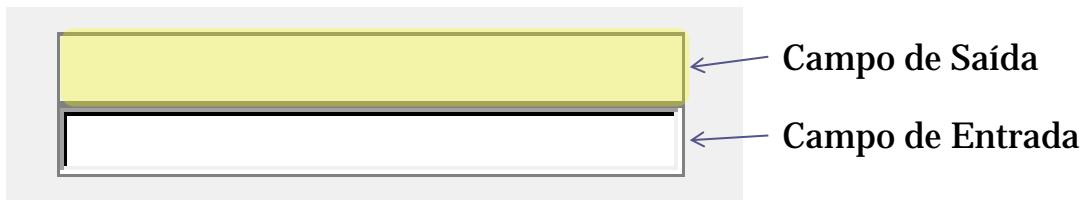
289

Palheta de Elementos



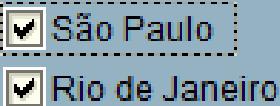
Campo de Entrada/Saída

- Pode ser utilizado como um campo de texto dinâmico, ou seja, através de programação o valor do texto é atribuído dinamicamente.



290

Palheta de Elementos

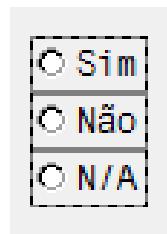


Campo de Seleção

- Campo utilizado para o usuário escolher uma ou mais opção
- Quando o campo estiver marcado, o valor transferido à variável atribuída ao campo de seleção é 'X'. O tipo de variável é CHAR de uma posição.

291

Palheta de Elementos



Botão de Rádio

- Utilizado para fornecer ao usuário opções. Ex.: Sim/Não, Aprovado/Reprovado
- O usuário só pode escolher uma opção
- Os campos devem ser agrupados para que o botão funcione
- O tipo de dado da variável atribuída ao campo é CHAR de uma posição

292

Palheta de Elementos

 Abrir

Botão

- O botão é um objeto para a execução de uma rotina
- Para que o processador ABAP reconheça que um botão foi pressionado, o botão tem uma propriedade chamada código de função.
- O código de função é transferido para a variável de sistema SY-UCOMM para que o programador possa criar uma rotina
- O evento acionado pelo botão é o evento PAI da tela onde o botão foi criado

293

Palheta de Elementos



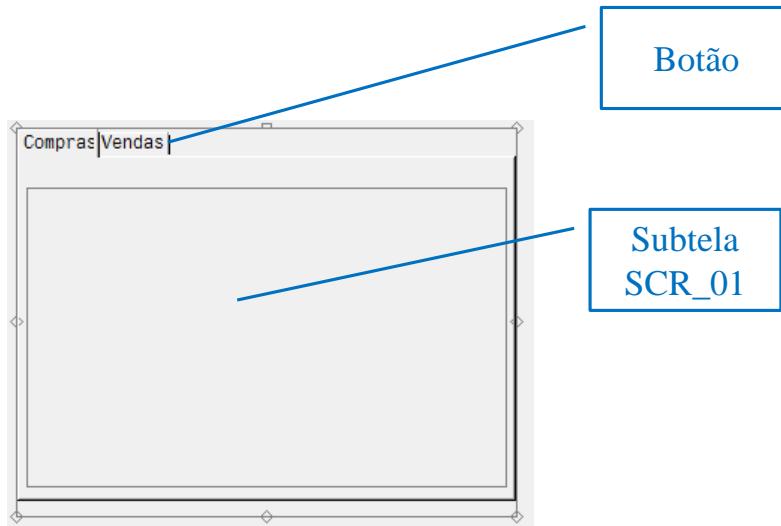
Controle de Fichas

- Os campos podem ser organizados por fichas(abas, guias)
- O único objeto que pode ser inserido nas fichas é subtela
- Cada ficha é um botão, o qual assume as mesmas características do botão (código de função), com a adição do campo de referência, que é o nome técnico da subtela a ser adicionada na ficha

294

Palheta de Elementos

Controle de Fichas



295

Palheta de Elementos

Controle de Fichas (Wizard)

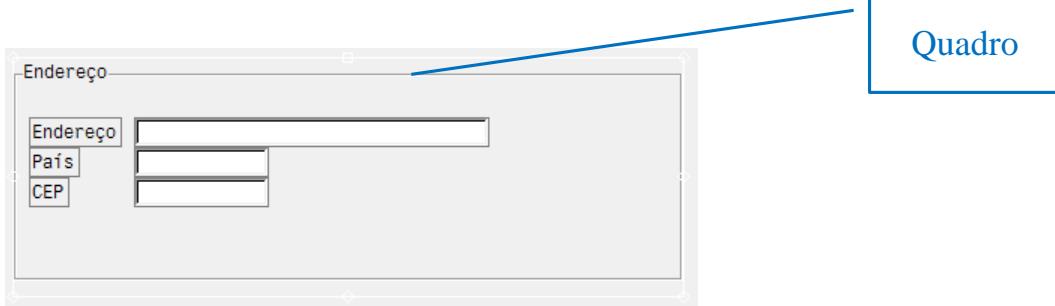
- O controle de fichas pode ser criado através de um assistente
- De acordo com as opções escolhidas pelo desenvolvedor, o sistema irá criar automaticamente as rotinas nos eventos PAI e PBO

296

Palheta de Elementos

□ Quadro

- O quadro ou moldura é uma ferramenta para que o usuário possa organizar a tela
- O quadro pode ter título ou não



Quadro

297

Palheta de Elementos

□ Área de Subtela

- Área de subtelas são áreas que podem conter uma determinada subtela
- A vantagem é a reutilização de subtelas entre várias telas

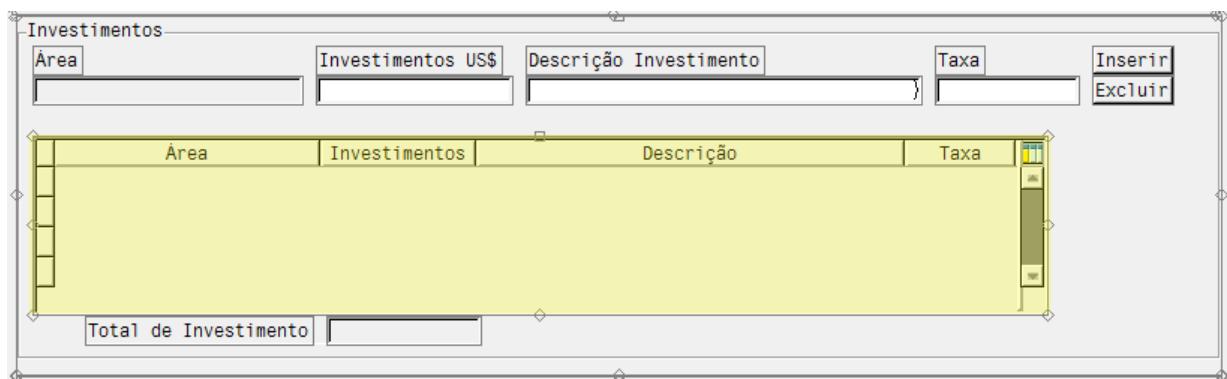
298

Palheta de Elementos



Table Control

- Table Control é uma tabela que permite que os dados sejam mostrados e manipulados na forma de uma tabela



299

Palheta de Elementos



Table Control (Wizard)

- O table control pode ser criado através de um assistente
- De acordo com as opções escolhidas pelo desenvolvedor, o sistema irá criar automaticamente as rotinas nos eventos PAI e PBO

300

Palheta de Elementos



Container

- Container é uma área de tela onde pode ser adicionado uma caixa de texto ou uma tabela ALV através de ABAP Objects

The screenshot shows a SAP application window titled "Detalhes da Alteração". At the top, there is a toolbar with various icons. Below the toolbar is a header row with columns: "Acrescido/Revisado", "Seqüência", "Revisão", "Quantidade", "Desenho", and "Denom.". A tree view on the left shows a folder structure with node 1200213763 expanded, revealing sub-nodes 77881999111 and 988771661. The main table area contains three rows of data corresponding to these nodes. The first row (node 1200213763) has values: Seqüência 10 01, Revisão 1,000, Desenho checked, Denom. MOTOR. The second row (node 77881999111) has values: Seqüência 11 02, Revisão 1,000, Desenho checked, Denom. CABECOTE DO MOTOR. The third row (node 988771661) has values: Seqüência 12 01, Revisão 1,000, Desenho checked, Denom. ALTERNADOR COMPLETO.

Acrescido/Revisado	Seqüência	Revisão	Quantidade	Desenho	Denom.
1200213763	10 01	1,000	<input type="checkbox"/>	MOTOR	
	77881999111	11 02	<input type="checkbox"/>	CABECOTE DO MOTOR	
	988771661	12 01	<input type="checkbox"/>	ALTERNADOR COMPLETO	

301

Palheta de Elementos

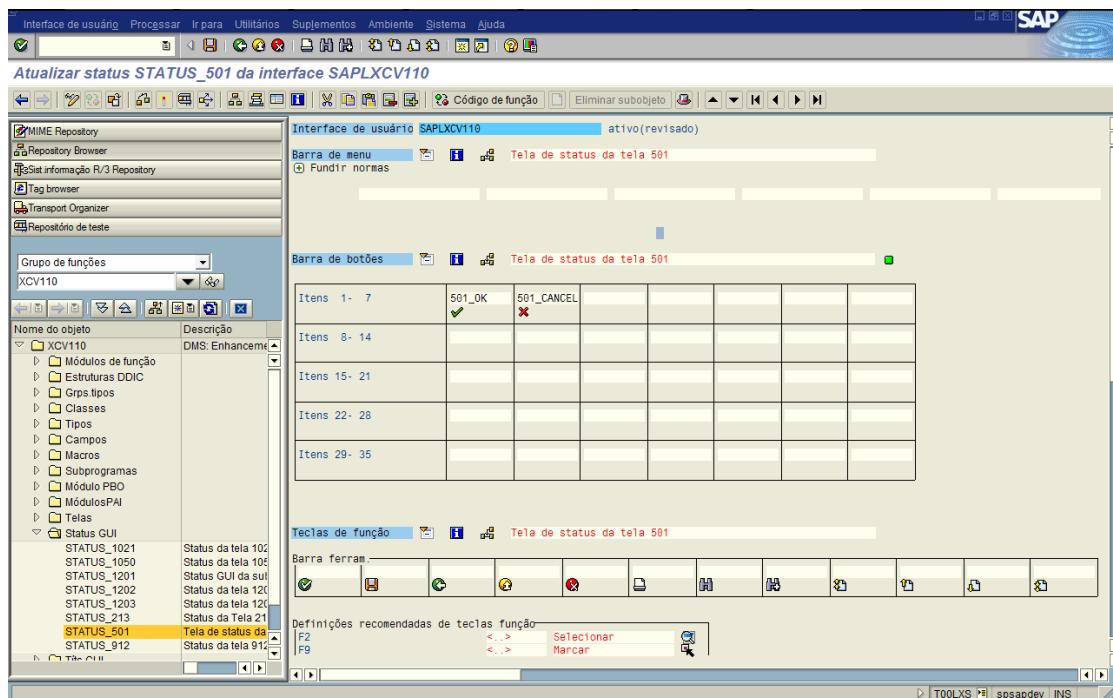


Ícone Status

- É um elemento de tela utilizado para representar o estado de um programa graficamente.
- O nome técnico deve ser o mesmo do texto do tipo ICONS do dicionário de dados

302

Status GUI



303

Title GUI



304

Evento PBO

- Evento executado antes da exibição da tela para o usuário:
 - Definição do GUI Status e GUI Title para a tela correspondente
 - Seleção inicial de dados e atribuição de valores a serem exibidos na tela

305

Evento PBO

```
*&-----*  
*&  Module STATUS_0100 OUTPUT  
*&-----*  
*  Módulo PBO  
*-----*  
MODULE STATUS_0100 OUTPUT.  
SET PF-STATUS 'TELA100'.  
SET TITLEBAR 'Z10'.  
  
ENDMODULE.           " STATUS_0100 OUTPUT
```

306

Evento PAI

- Evento executado após um comando do usuário (enter, click de um botão, etc...)
 - Verificação do código de comando executado (definido no GUI Status)
 - Código ABAP para execução da lógica do comando

307

Evento PAI

```
*&-----*  
*&  Module USER_COMMAND_0100 INPUT  
*&-----*  
*  Módulo PAI  
*-----*  
MODULE USER_COMMAND_0100 INPUT.  
CASE SY-UCOMM.  
  WHEN 'EXIT'.  
    SET SCREEN 0.  
    LEAVE SCREEN.  
  ENDCASE.  
ENDMODULE.          " USER_COMMAND_0100 INPUT
```

308

Evento POV e POH

- Process On Value-request (POV)
 - O evento POV é ativado quando o usuário solicita os possíveis valores (F4) de um determinado campo
- Process On Help-request (POH)
 - O evento POH é ativado quando o usuário solicita ajuda (F1) de um determinado campo

309

316

Curso Avançado de ABAP

Batch-Input

O que é batch Input

- O batch-input é uma ferramenta utilizada para transferir um conjunto de dados para o sistema R/3
- O fluxo da transação é simulado e o dado é transferido como se fosse inserido de forma on-line
- A vantagem é que todas as verificações da transação são executadas, garantido assim a consistênciados dados

317

Processo

- O processo de batch input é dividido em duas partes:
 - O programa de transferência de dados cria uma sessão batch input que possui todos os dados relevantes
 - A sessão batch input é processada e o dado é transferido para o sistema R/3

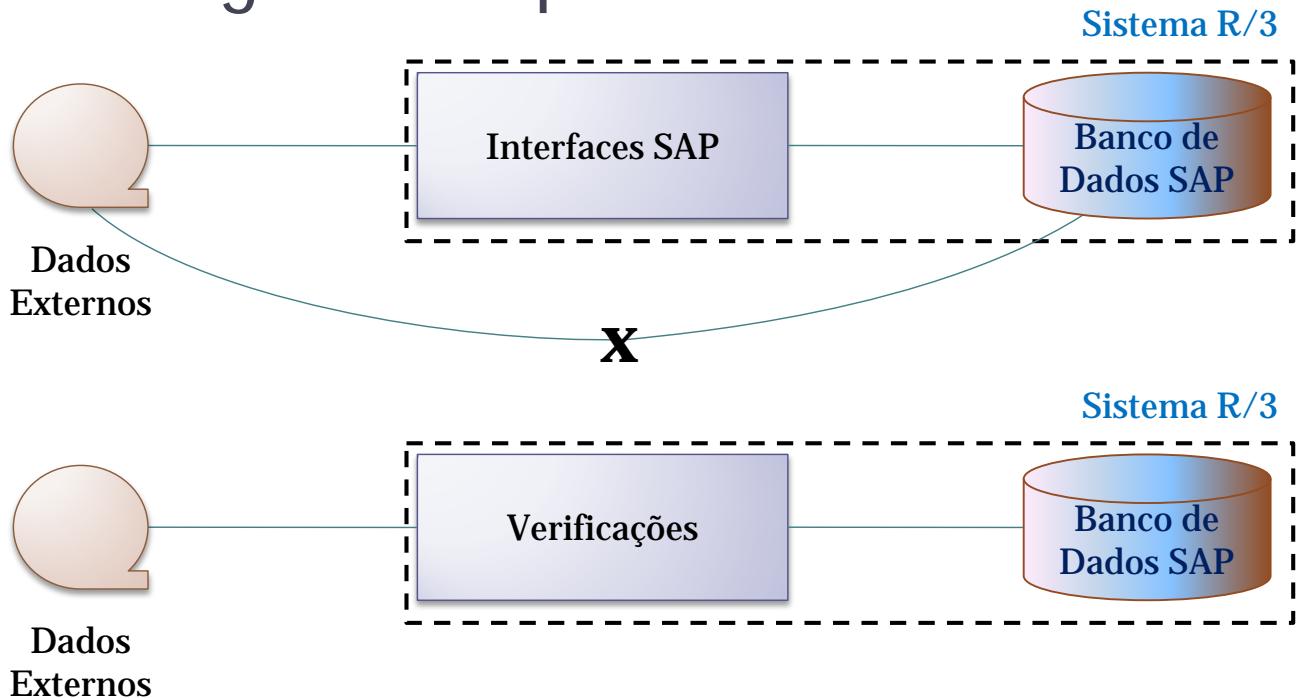
318

Tipo de processamento

- A sessão de batch input pode ser processada de três tipos:
 - Primeiro plano (foreground)
 - Plano de fundo (background)
 - Em plano de fundo, mas apresenta a tela com o erro ocorrido

319

Visão geral do problema

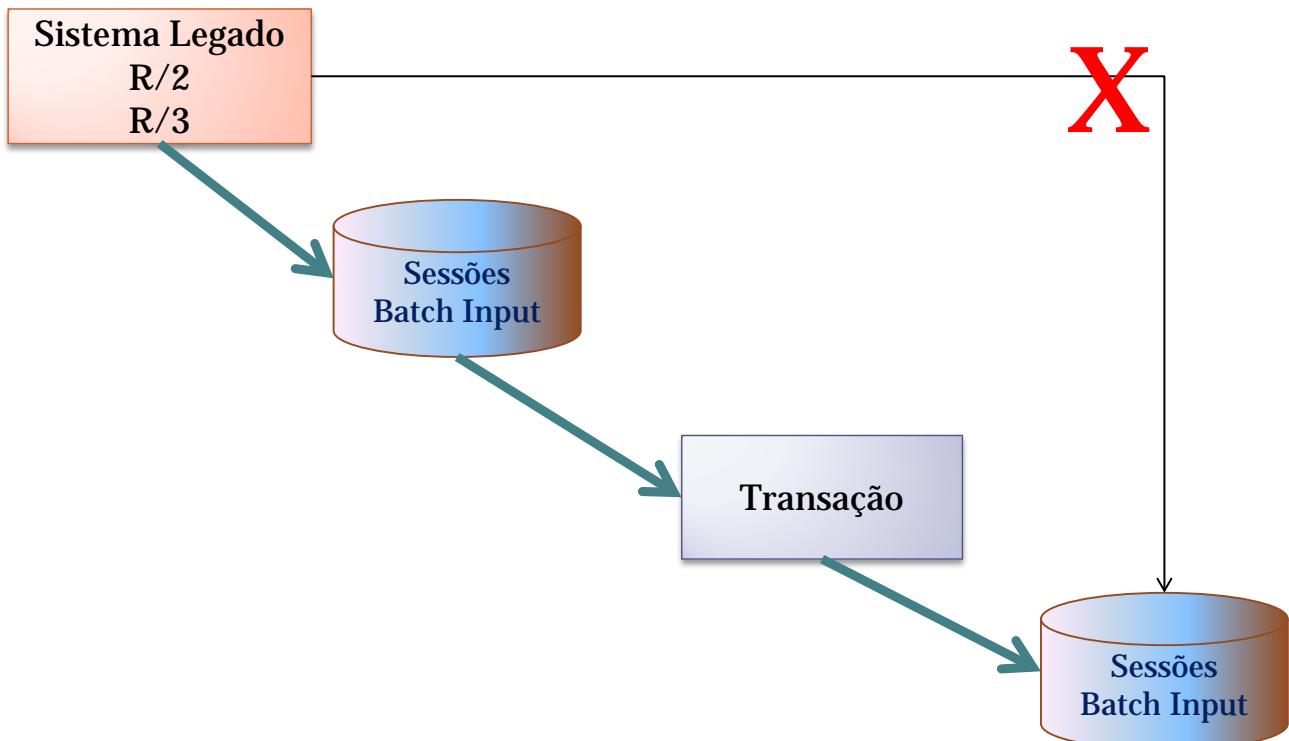


Sistema R/3

Sistema R/3

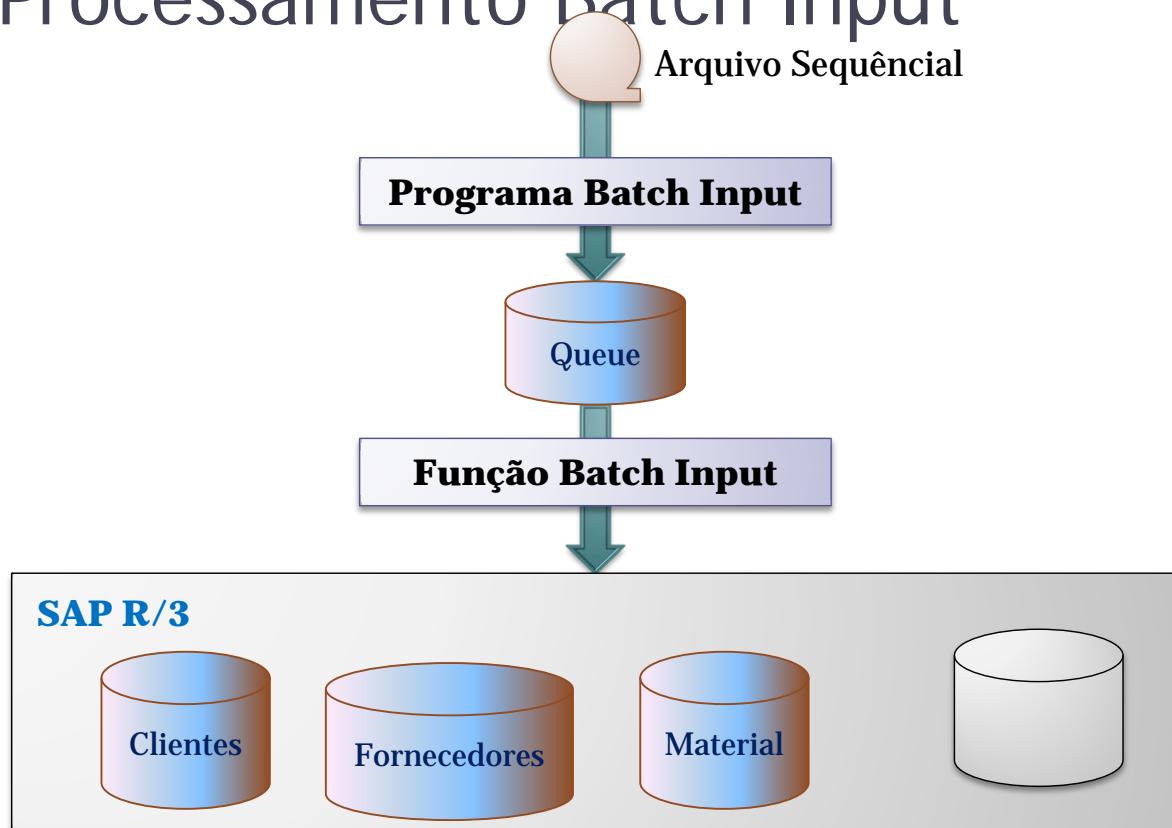
320

Visão geral do problema



321

Processamento Batch Input

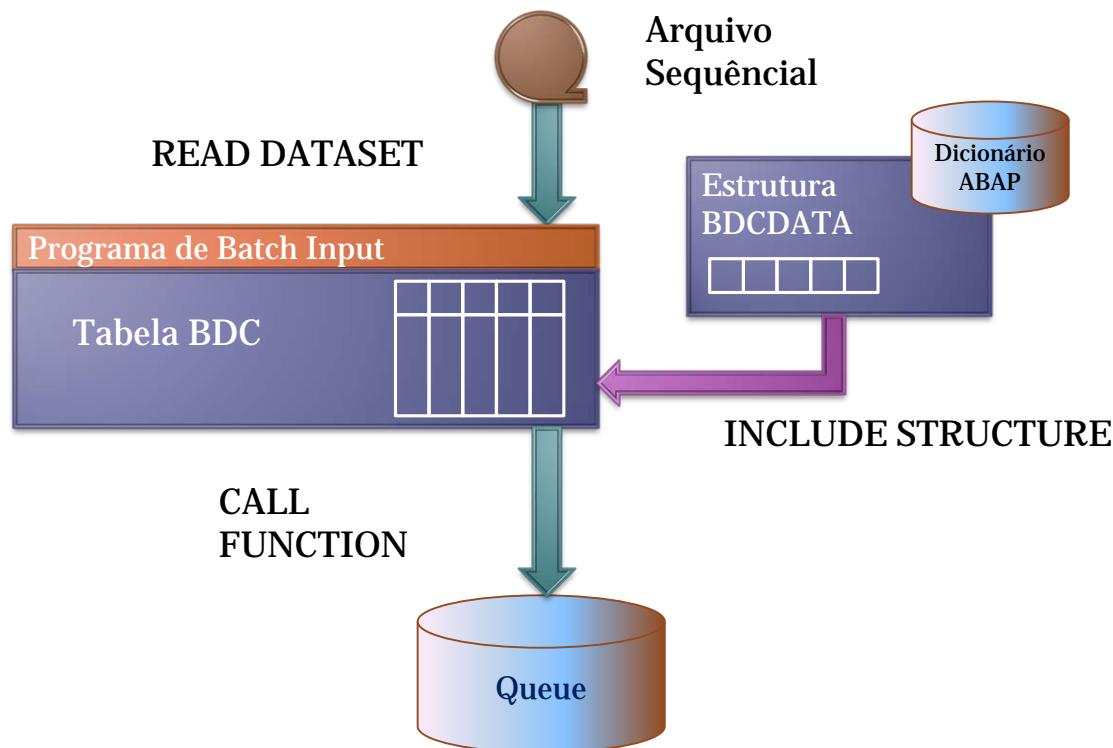


322

Programa Batch Input

TRAINNING

Education Services

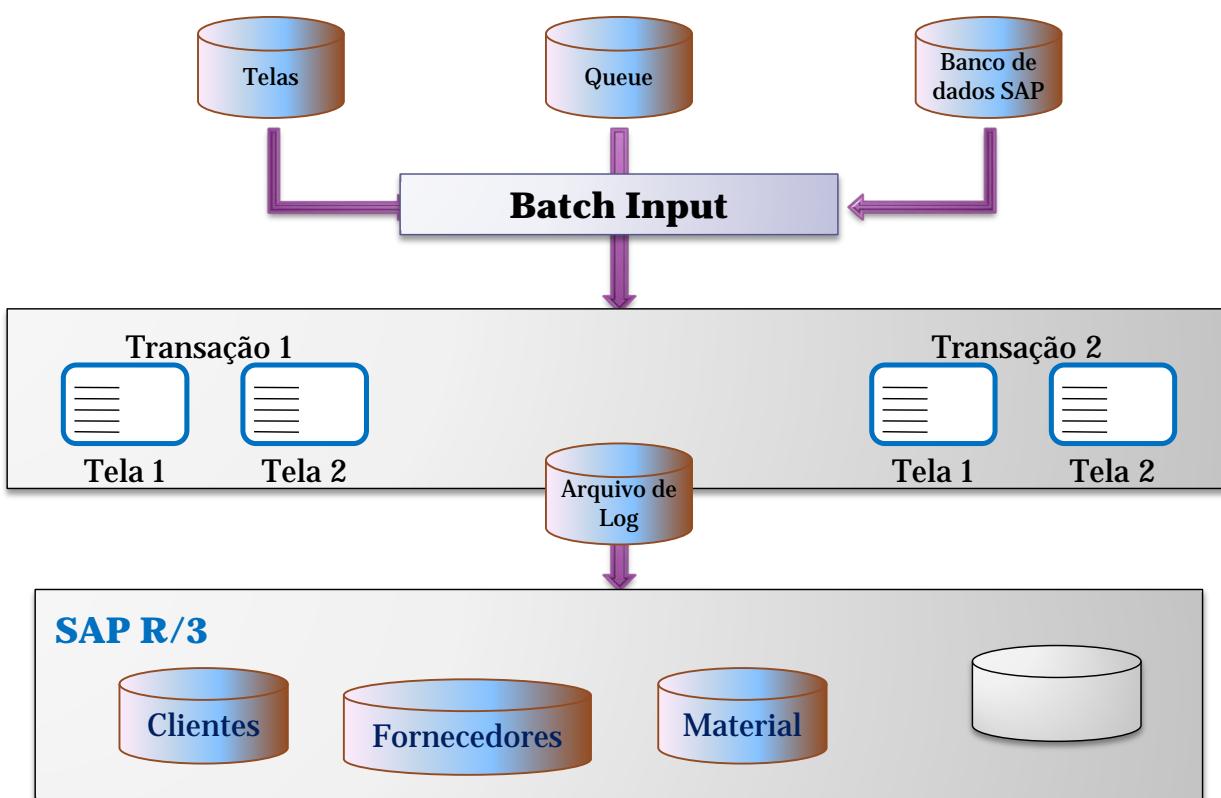


323

Batch Input Queue

TRAINNING

Education Services



324

Formato da Sessão Batch Input

- Usuário que criou a sessão
- Mandante na qual a sessão deve ser processada
- Nome da sessão
- Usuário Batch Input
- Data de quando a sessão deve ser processada (data de bloqueio)
- Remover a sessão após o processamento (manter a função)

Cabeçalho da Sessão

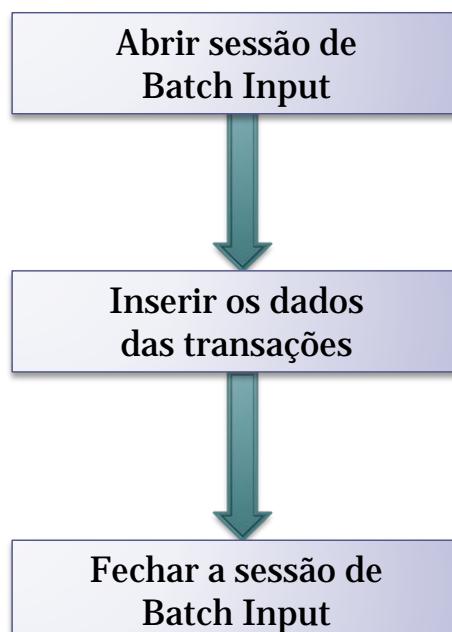
Transação



Fechar Sessão

325

Criação de uma Sessão Batch Input



326

Módulos de Função de Batch Input

BDC_OPEN_GROUP

Mandante	CLIENT
Nome da Sessão	GROUP
Data de Bloqueio	HOLD
Remover a sessão após processamento	KEEP
Usuário BDC	USER

BDC_INSERT

Código da Transação Tabela BDC	TCODE DYNPROTAB
-----------------------------------	--------------------

BDC_CLOSE_GROUP

327

Módulos de Função de Batch Input

- O módulo de função para abrir a sessão de batch input é BDC_OPEN_GROUP
 - Os parâmetros obrigatórios são:
 - CLIENT - Mandante
 - GROUP - nome da sessão
 - USER - usuário da sessão Batch Input
 - Os parâmetros abaixo são opcionais
 - HOLDDATE - a sessão só poderá ser executada a partir da data especificada
 - Keep – Flag informando se a sessão deverá ser removida após o processamento ou não (KEEP = 'X' não apaga a sessão)

328

Módulos de Função de Batch Input

- O módulo de função para inserir os dados de UMA transação é **BDC_INSERT**
 - Para transferir os dados para a função, será necessário que a tabela BDC esteja preenchida
 - Os seguintes parâmetros deverão ser preenchidos:
 - TCODE – Código da transação (ex.: MM01, XK01)
 - DYNPROTAB (Tabela interna BDC)

329

Tabela BDC

Nome do programa	Número da tela	ID Início	Nome do campo	Conteúdo do campo
PROGRAM	DYNPRO	DYNBEGIN	FNAM	FVAL
SAPLMGMM	0060	X		
			BDC_CURSOR	RMMG1-MATNR
			BDC_OKCODE	/00
			RMMG1-MATNR	MH00046468111
			RMMG1-MBRSH	M
			RMMG1-MTART	ERSA
SAPLMGMM	0070	X		
			BDC_CURSOR	MSICHTAUSW-DYTXT(17)
			BDC_OKCODE	=ENTR
			MSICHTAUSW-KZSEL(01)	X
			MSICHTAUSW-KZSEL(04)	X

330

Definição da Tabela BDC

- A tabela interna deve ser criada com referência a estrutura BDCDATA.

```
DATA: TI_BDCDATA          TYPE BDCDATA OCCURS 0 WITH HEADER LINE.  
      Ou  
DATA: TI_BDCDATA          TYPE TABLE OF BDCDATA.  
      Ou  
DATA: BEGIN OF TI_BDCDATA OCCURS 0 .  
      INCLUDE STRUCTURE BDCDATA.  
DATA: END OF TI_BDCDATA.
```

331

Estrutura BDCDATA

The screenshot shows the SAP Dictionary interface with the following details:

- Toolbar:** Includes Estrutura, Processar, Ir para, Utilitários, Suplementos, Ambiente, Sistema, Ajuda, and various icons for search, refresh, and navigation.
- Title Bar:** Dictionary: exibir estrutura
- Search Bar:** Estrut. BDCDATA ativo
- Description Bar:** Descrição breve: Batch input nova estrutura do campo de tabela
- Tab Bar:** Características, Componentes (selected), Entrs.possíveis/verificação, Campos moeda/quantidade
- Table View:** Shows the components of the BDCDATA structure. The table has columns: Componente, Tp..., Tipo componente, Categoria d..., Compr, Casa..., and Descrição breve. The data is as follows:

Componente	Tp...	Tipo componente	Categoria d...	Compr	Casa...	Descrição breve
PROGRAM	<input type="checkbox"/>	BDC_PROG	CHAR	40	0	Pool de módulos BDC
DYNPRO	<input type="checkbox"/>	BDC_DYNR	NUMC	4	0	Número de tela BDC
DYNBEGIN	<input type="checkbox"/>	BDC_START	CHAR	1	0	Início BDC de uma tela
FNAM	<input type="checkbox"/>	FNAM_4	CHAR	132	0	Nome do campo
FVAL	<input type="checkbox"/>	BDC_FVAL	CHAR	132	0	Valor do campo BDC

332

Preenchimento da tabela BDC

```
FORM <fill_BDC_table>.  
  REFRESH <BDC_table>.  
  CLEAR <BDC_table>.  
  MOVE: <program_name> TO <BDC_table>-PROGRAM,  
        <number1> TO <BDC_table>-DYNPRO,  
        'X'      TO <BDC_table>-DYNBEGIN.  
  APPEND <BDC_table>.  
  CLEAR <BDC_table>.  
  MOVE: <field11> TO <BDC_table>-FNAM,  
        <value11> TO <BDC_table>-FVAL.  
  APPEND <BDC_table>.  
  CLEAR <BDC_table>.  
  MOVE: <field12> TO <BDC_table>-FNAM,  
        <value12> TO <BDC_table>-FVAL.  
  APPEND <BDC_table>.  
  ...  
  CLEAR <BDC_table>.  
  MOVE: <program_name> TO <BDC_table>-PROGRAM,  
        <number2> TO <BDC_table>-DYNPRO,  
        'X'      TO <BDC_table>-DYNBEGIN.  
  APPEND <BDC_table>.  
  ...  
ENDFORM.
```

333

Módulos de Função de Batch Input

- O módulo de função para fechar a sessão de batch input é **BDC_CLOSE_GROUP**
 - O módulo de função **BDC_CLOSE_GROUP** não possui parâmetros de entrada, mas retorna as exceções que podem ser tratadas no programa de batch input.

334

SHDB - Transaction Recorder

- Assim uma determinada transação pode ser mapeada com a finalidade de ser utilizada em um programa de batch input
- A transação de mapeamento de transação é SHDB
- Na transação SHDB é possível criar registros de mapeamento e a partir dela, criar programas, módulo de função e sessão batch input.

335

SHDB - Transaction Recorder

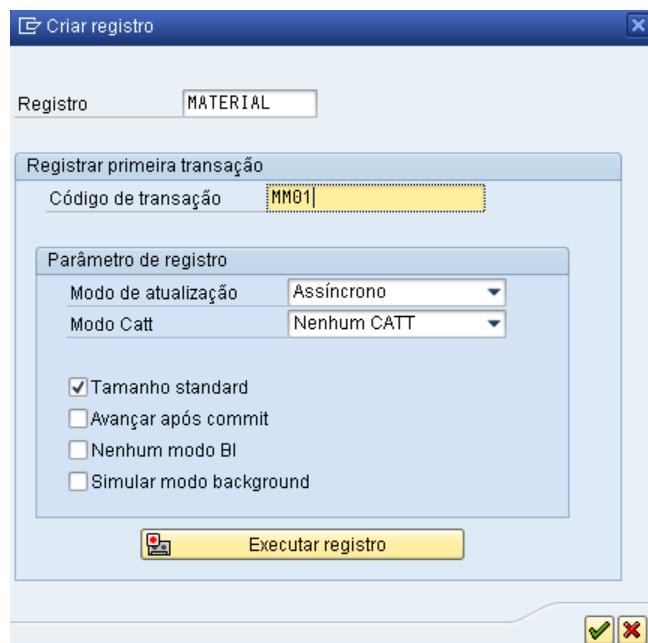
- Entrar na transação SHDB
 1. clicar no botão “Nova gravação”



336

SHDB - Transaction Recorder

- Inserir os dados tais como o nome do registro da gravação, a transação a ser executada e as opções de execução da transação.



337

SHDB - Transaction Recorder

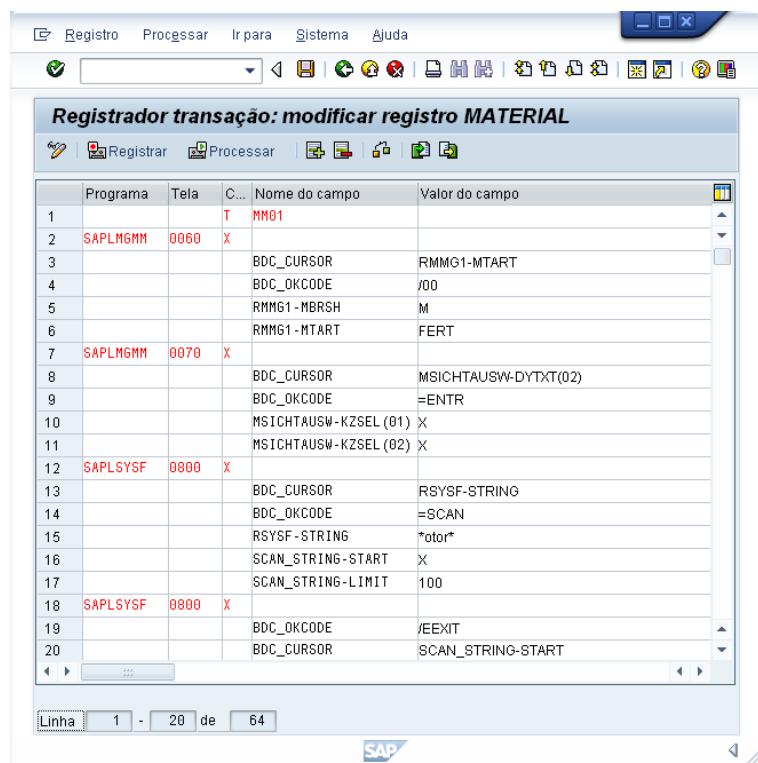
- Inserir os dados necessários nas telas conforme a regra de negócio.

338

SHDB - Transaction Recorder

TRAINNING
Education Services

- Ao terminar o mapeamento da tela, o código de BDC é apresentado.

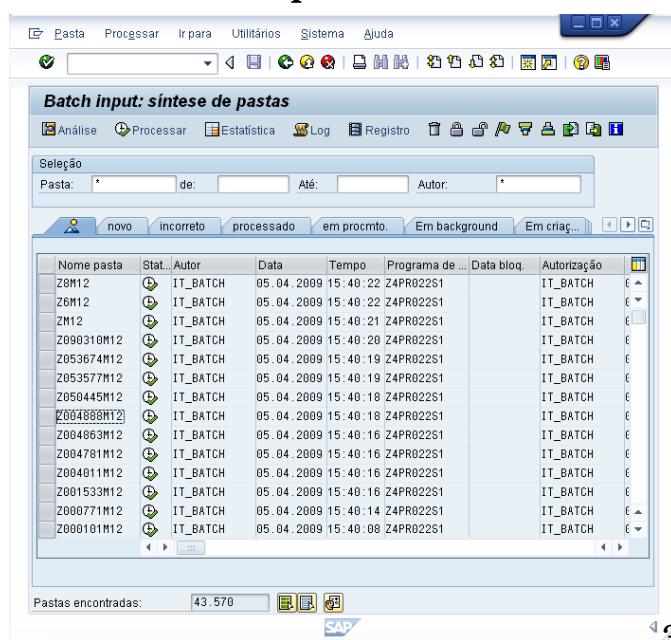


339

SM35 - Gerência de Batch Input

TRAINNING
Education Services

- As sessões criadas pelo programa de batch input são administradas na transação SM35.
- A transação SM35 permite processar as sessões (pastas) e visualizar o log das sessões e estatísticas do processamento.



340

SM35 - Status das sessões

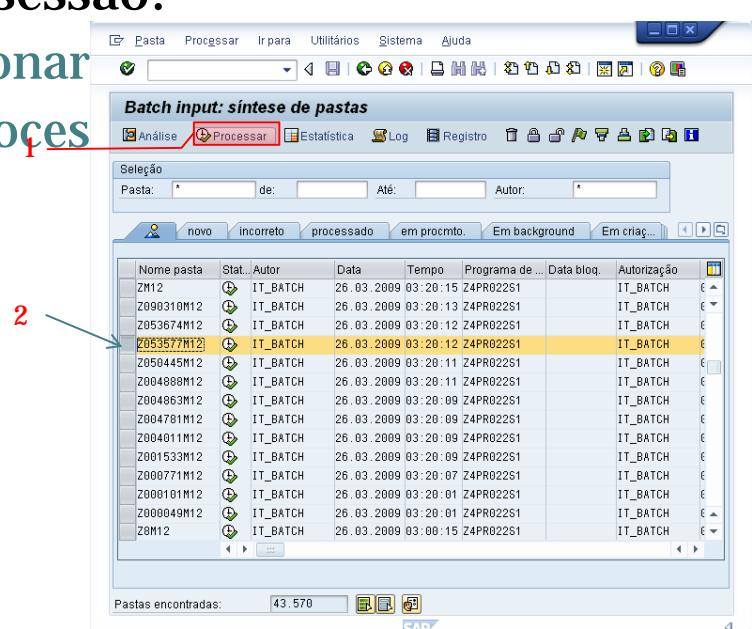
Processada	Sessão processada com sucesso
Defeituosa	Sessão processada mas contendo transações com erros
A ser processada	Sessão foi gravada mas ainda não foi processada
Em criação	Sessão está a ser gravada
Em processamento	Sessão está a ser processada
Background	Sessão agendada para processamento em background
Bloqueado	Sessão bloqueada para processamento até determinada data

341

SM35 - Processar Sessão

- Para processar uma sessão:

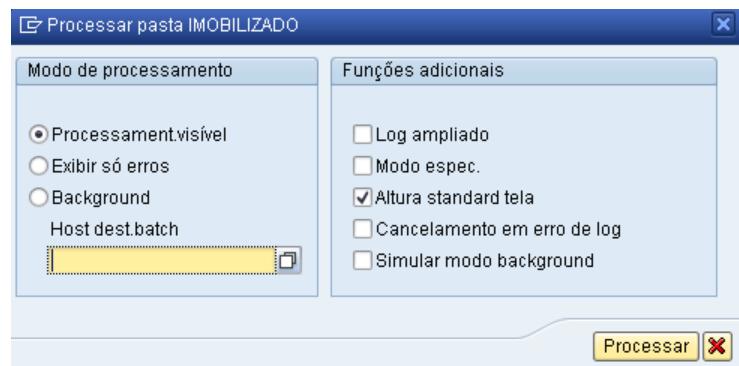
1. é necessário selecionar
2. clicar no botão “Proces



342

SM35 – Processar Sessão

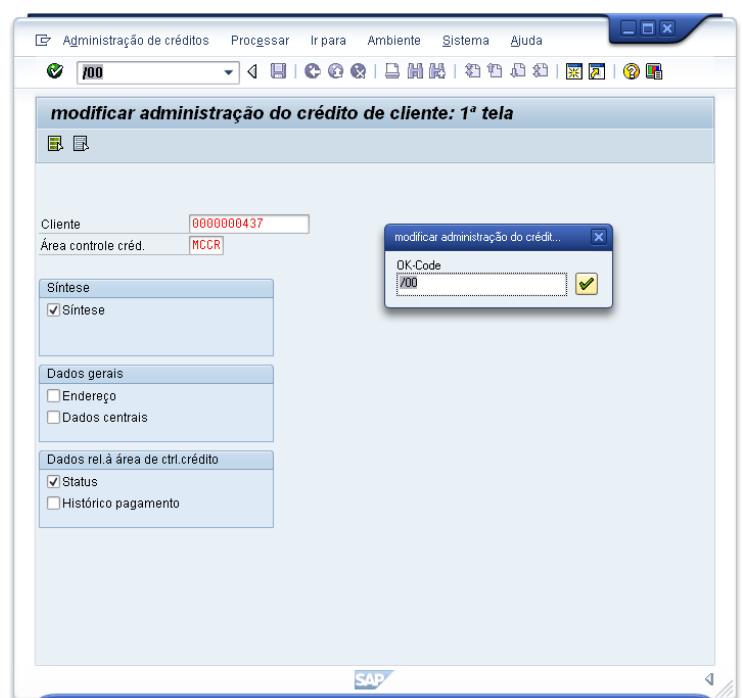
- Ao clicar no botão processar, é aberta uma janela onde será definido os parâmetros de processamento.



343

SM35 – Processar Sessão

- Os campos com os valores em vermelho são preenchidos automaticamente pela sessão.
- A janela OK_CODE recebe o código de função.



344

SM35 - Códigos de Função

Função	OK code
Terminar a transação de batch input corrente e marcar como incorreta	/n
Apagar a transação de batch input corrente da sessão	/bdel
Terminar processamento batch input e marcar sessão como incorreta	/bend
Alterar modo de processamento, passar a visualizar telas em vez de ver apenas erros	/bda
Alterar modo de processamento , passar a visualizar apenas erros em vez de ver as telas	/bde

345

Sumário

- Batch Input é um procedimento automático adequado para processamento de dados de massa. É usando primariamente na instalação do sistema SAP e na transferência do sistema legado (sistema externo).
- No desenvolvimento, o dado a ser transferido deve existir no formato apropriado da tela. Também é necessário utilizar os módulos de função apropriados para criar uma sessão executável em um arquivo intermediário (queue).
- As sessões são executadas através da transação SM35.

346

Curso Avançado de ABAP

Call Transaction



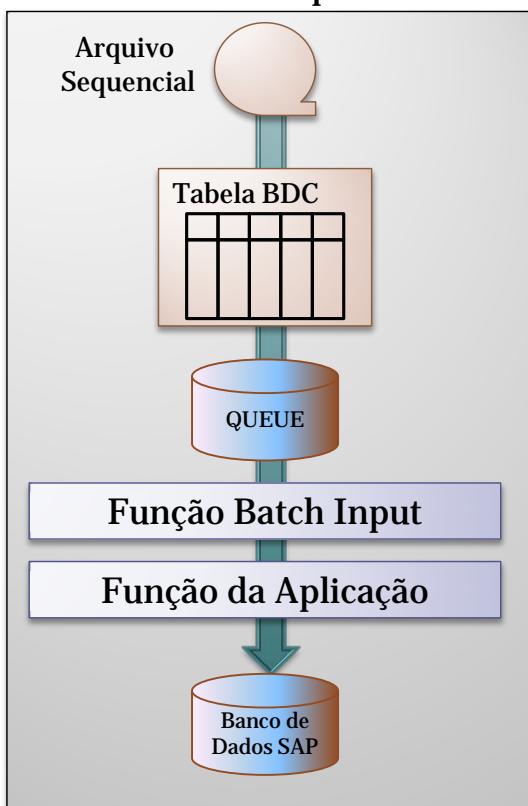
O que é Call Transaction



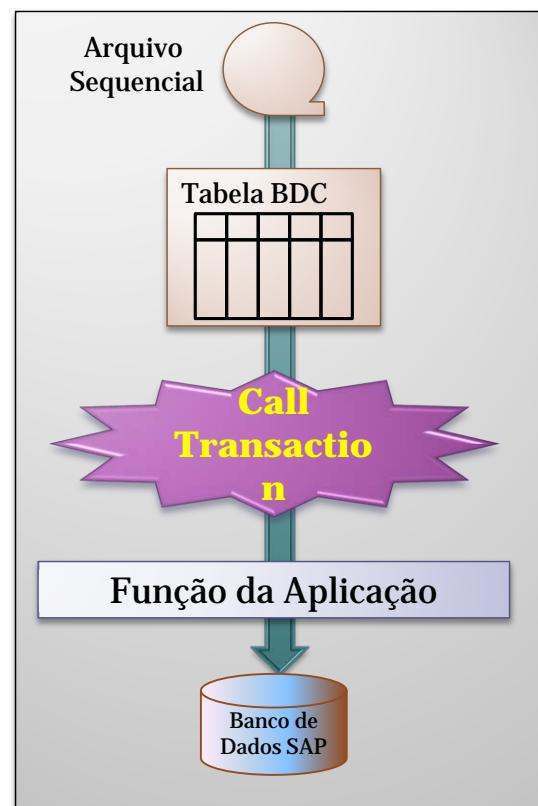
- O Call Transaction é uma ferramenta utilizada para transferir um conjunto de dados para o sistema R/3
- Diferentemente do batch input, o call transaction permite transferir os dados diretamente para a interface de diálogo (sem usar o queue).

Overview

Batch Input



Call Transaction



349

Comando CALL TRANSACTION

CALL TRANSACTION <código da transação>

USING <Tabela BDC>
MODE <Modo de apresentação>
UPDATE <técnica de
atualização>
MESSAGES INTO <Tabela de
mensagem>

Modo de Apresentação

A	Mostrar todas as telas
E	Mostrar a tela se ocorrer erro
N	Mostrar a tela se ocorrer erro
P	Permite o debug

Técnica de Atualização

S	Síncrono - Não continua o processamento até que a atualização seja finalizada
A	Assíncrono - Continua o processamento imediatamente

350

Código de retorno e campos de sistema

Código de retorno – SY-SUBRC

Valor	Significado
0	Sucesso
<1000	Erro no programa de diálogo
>1000	Erro no processamento batch input (verificar campos de sistema)

Campos de sistema

Nome do campo	Significado
SY-MSGID	ID da mensagem
SY-MSGTY	Tipo da Mensagem
SY-MSGNO	Número da Mensagem
SY-MSGV1	Variável da mensagem 1
SY-MSGV2	Variável da mensagem 2
SY-MSGV3	Variável da mensagem 3
SY-MSGV4	Variável da mensagem 4

351

Saída da mensagem

```
REPORT ZMMR0001.  
...  
  
CALL TRANSACTION 'MM01'  
    USING TI_BDCDATA  
    MODE 'N'  
    UPDATE 'S'  
    MESSAGES INTO TI_MESSAGES.  
  
MESSAGE ID SY-MSGID  
    TYPE SY_MSGTY  
    NUMBER SY-MSGNO  
    WITH SY-MSGV1 SY-MSGV2  
        SY-MSGV3 SY-MSGV4.  
...
```

352

Sumário

- O call transaction é uma alternativa ao batch input. A transferência de dados é particularmente mais rápido pois o arquivo de queue é evitado (sem geração de sessão).



---- DESTACAR ESTA FOLHA E ENTREGAR AO INSTRUTOR ----

Data: _____

NOME: _____

CURSO: _____

INSTRUTOR: _____

AVALIAÇÃO FINAL DE TREINAMENTO

Marque com um X na lacuna ao lado da figura na opção desejada.

Conteúdo Programático do Curso	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Tempo de Realização do Curso	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Referência Final ao Treinamento	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Material Didático	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Didática Instrutor	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Conhecimento Técnico do Instrutor	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Instalações Físicas (Mesa, Cadeira, Quadro, Projetor, Computador)	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Atendimento Comercial e Administrativo	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼
Conteúdo, Material Adicional e Informações no site da Trainning	<input type="checkbox"/> ☹	<input type="checkbox"/> ☺	<input type="checkbox"/> ☻	<input type="checkbox"/> ☼

COMENTÁRIOS

Espaço livre para comentários, críticas e sugestões sobre o treinamento realizado.

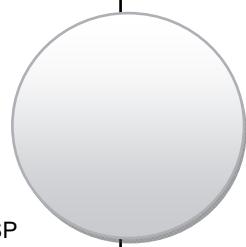
INDICAÇÕES

Indique 1 amigo e Ganhe Desconto no próximo treinamento.

☹ REGULAR

☺ SATISFATÓRIO

☻ ÓTIMO



Trainning Education Services ® - Sede: Av. Paulista, 777, 15º andar – SP

São Paulo (11) 3323-1676 - Rio de Janeiro (21) 4063-6518 - Brasília (61) 4063-7766 - Belo Horizonte (31) 4063-8187
Vitória (27) 4062-9192 - Curitiba (41) 4063-8083 - Porto Alegre (51) 4063-9474 - Recife (81) 3301-1600 - Salvador (71) 4062-9323
www.trainning.com.br - comercial@trainning.com.br

AVALIAÇÃO FINAL DE TREINAMENTO