**ABAP 7.40 and above release**

**1. Escape Character for Host Variables**

ABAP data objects used in Open SQL statements (usually variables) are now interpreted as host variables.

From Release 7.40, SP05, host variables can and should be prefixed with the escape character @.

**EXAMPLE:**

SELECT VBELN ,

POSNR ,

MATNR

FROM VBAP

INTO TABLE @LIT\_VBAP

WHERE VBELN = @P\_VBELN.

**2. String Templates**

The purpose of a string template is to create a new character string out of literal texts and embedded expressions.

|  |  |  |
| --- | --- | --- |
| A string template is defined by using the | (pipe) symbol at the beginning and end of a template. **Description** | **Before 7.40** | **After 7.40** |
| String Template Normal Assignment | DATA: lv\_character\_string TYPE string.  lv\_character\_string = `This is a literal text. `. | DATA: lv\_character\_string TYPE string.  lv\_character\_string = **|**This is a literal text.**|**. |

**Examples:**

Before ABAP 7.40

****

After ABAP 7.40



**3. Inline Declaration**

|  |  |  |
| --- | --- | --- |
| **Description** | **Before 7.40** | **After 7.40** |
| Data Declaration in Normal Scenario and in Call Method | DATA: LV\_TEXT TYPE STRING.  LV\_TEXT = 'ABC'. | **DATA(TEXT)** = 'ABC'. |
| Select Into Table Declaration **(ONLY POSSIBLE IN ABAP 7.50 RELEASE)** | DATA: ITAB TYPE TABLE OF VBAK,  LV\_VBELN TYPE VBAK-VBELN.  SELECT \* FROM VBAK INTO TABLE ITAB WHERE VBELN = LV\_VBELN. | DATA: LV\_VBELN TYPE VBAK-VBELN.  SELECT \* FROM VBAK INTO TABLE **@DATA(ITAB)** WHERE VBELN = **@LV\_VBELN**. |
| Select Single Into Declaration**(ONLY POSSIBLE IN ABAP 7.50 RELEASE)** | PARAMETERS: P\_VBELN TYPE VBAK-VBELN.  DATA: LV\_VBELN TYPE VBAK-VBELN, LV\_AUART TYPE VBAK-AUART.  SELECT SINGLE VBELN AUART FROM VBAK INTO ( LV\_VBLEN, LV\_AUART) WHERE VBELN = P\_VBELN.  IF SY-SUBRC = 0.  WRITE : / LV\_VBELN , LV\_AUART.  ENDIF. | PARAMETERS: P\_VBELN TYPE VBAK-VBELN.  SELECT SINGLE VBELN,  AUART FROM VBAK INTO **@DATA(LWA\_VBAK)** WHERE VBELN = **@P\_VBELN**.  IF SY-SUBRC = 0.  WRITE : / LWA\_VBAK-VBELN , LWA\_VBAK-AUART.  ENDIF. |
| **4. Table Expression**   Table expression is a new way to read data from internal table.  Index access | READ TABLE lit\_vbak INTO lwa\_vbak INDEX 5. | LWA\_VBAK = LIT\_VBAK [ 5 ]. |
| Index access using a secondary index | READ TABLE itab INDEX idx USING KEY key INTO lwa. | LWA = itab [ KEY key INDEX idx ]. |
| Access using a free key | READ TABLE LIT\_VBAK INTO LWA\_VBAK  WITH KEY VBELN = P\_VBELN AUART = P\_AUART. | LWA\_VBAK = LIT\_VBAK [ VBELN = P\_VBELN AUART = P\_AUART ]. |
| Key access using a table key | SYNTAX:  READ TABLE itab WITH TABLE KEY key COMPONENTS col1 = ... col2 = ... INTO lwa. Example: TYPES: BEGIN OF TY\_VBAP, VBELN TYPE VBAP-VBELN, POSNR TYPE VBAP-POSNR, MATNR TYPE VBAP-MATNR, WERKS TYPE VBAP-WERKS, END OF TY\_VBAP. DATA: LIT\_VBAP TYPE STANDARD TABLE OF TY\_VBAP WITH KEY VBELN POSNR WITH NON-UNIQUE SORTED KEY  MAT\_PLANT COMPONENTS MATNR WERKS, LWA\_VBAP TYPE TY\_VBAP. READ TABLE LIT\_VBAP WITH TABLE KEY MAT\_PLANT COMPONENTS MATNR = P\_MATNR  WERKS = P\_WERKS INTO LWA\_VBAP. | SYNTAX:  lwa = itab [KEY key col1 = ... col2 = ...]. OR lwa = itab [KEY key COMPONENTS col1 = ... col2 = ...]. Example: LWA\_VBAP = LIT\_VBAP [ KEY MAT\_PLANT MATNR = P\_MATNR WERKS = P\_WERKS ]. OR LWA\_VBAP = LIT\_VBAP [ KEY MAT\_PLANT COMPONENTS MATNR = P\_MATNR WERKS = P\_WERKS ]. |
| Does Record Exist? | READ TABLE lit\_vbak WITH KEY vbeln = p\_vbeln auart = p\_auart TRANSPORTING NO FIELDS.  IF SY-SUBRC = 0. ………………  ENDIF. | IF LINE\_EXISTS(LIT\_VBAK[ VBELN = P\_VBELN  AUART = P\_AUART ] ).  "Record Found.  ELSE.  "Record not Found"  ENDIF |
| Value Operator Structure Example | TYPES: BEGIN OF TY\_STRUCT,  FIRST(4) TYPE C,  SEC(4) TYPE C,  LAST(4) TYPE C,  END OF TY\_STRUCT.  DATA: LWA\_STRUCT TYPE TY\_STRUCT. LWA\_STRUCT-FIRST = 'ONE' .  LWA\_STRUCT-SEC = 'TWO'.  LWA\_STRUCT-LAST = 'THREE' .  CL\_DEMO\_OUTPUT=>DISPLAY( LWA\_STRUCT ). | TYPES: BEGIN OF TY\_STRUCT,  FIRST(4) TYPE C,  SEC(4) TYPE C,  LAST(4) TYPE C,  END OF TY\_STRUCT.  DATA: LWA\_STRUCT TYPE TY\_STRUCT. LWA\_STRUCT = VALUE #( FIRST = 'ONE' SEC = 'TWO' LAST = 'THREE' ).  CL\_DEMO\_OUTPUT=>DISPLAY( LWA\_STRUCT ). |

|  |  |  |
| --- | --- | --- |
| Value Operator Table Example | TYPES : BEGIN OF TY\_STRUCT,  FIRST(4) TYPE C,  SEC(4) TYPE C,  LAST(4) TYPE C,  END OF TY\_STRUCT.  DATA: LWA\_STRUCT TYPE TY\_STRUCT, LIT\_STRUCT TYPE TABLE OF TY\_STRUCT. LWA\_STRUCT-FIRST = 'ONE' . LWA\_STRUCT-SEC = 'TWO'.  LWA\_STRUCT-LAST = 'THREE' .  APPEND LWA\_STRUCT TO LIT\_STRUCT.  LWA\_STRUCT-FIRST = '1' .  LWA\_STRUCT-SEC = '2'.  LWA\_STRUCT-LAST = '3' .  APPEND LWA\_STRUCT TO LIT\_STRUCT.  CL\_DEMO\_OUTPUT=>DISPLAY( LIT\_STRUCT ). | TYPES : BEGIN OF TY\_STRUCT,  FIRST(4) TYPE C,  SEC(4) TYPE C,  LAST(4) TYPE C,  END OF TY\_STRUCT.  DATA: LIT\_STRUCT TYPE TABLE OF TY\_STRUCT.  \* Appending entries in internal table  LIT\_STRUCT = VALUE #( ( FIRST = 'ONE' SEC = 'TWO' LAST = 'THREE' )  ( FIRST = '1' SEC = '2' LAST = '3' ) ).  CL\_DEMO\_OUTPUT=>DISPLAY( LIT\_STRUCT ). |

Example of Index.



Example of Access using Free Key:



Example of Record exist in Table:

