

Sapprofession

SAP ABAP in Pratica

Guida Step-by-Step

Sapprofession

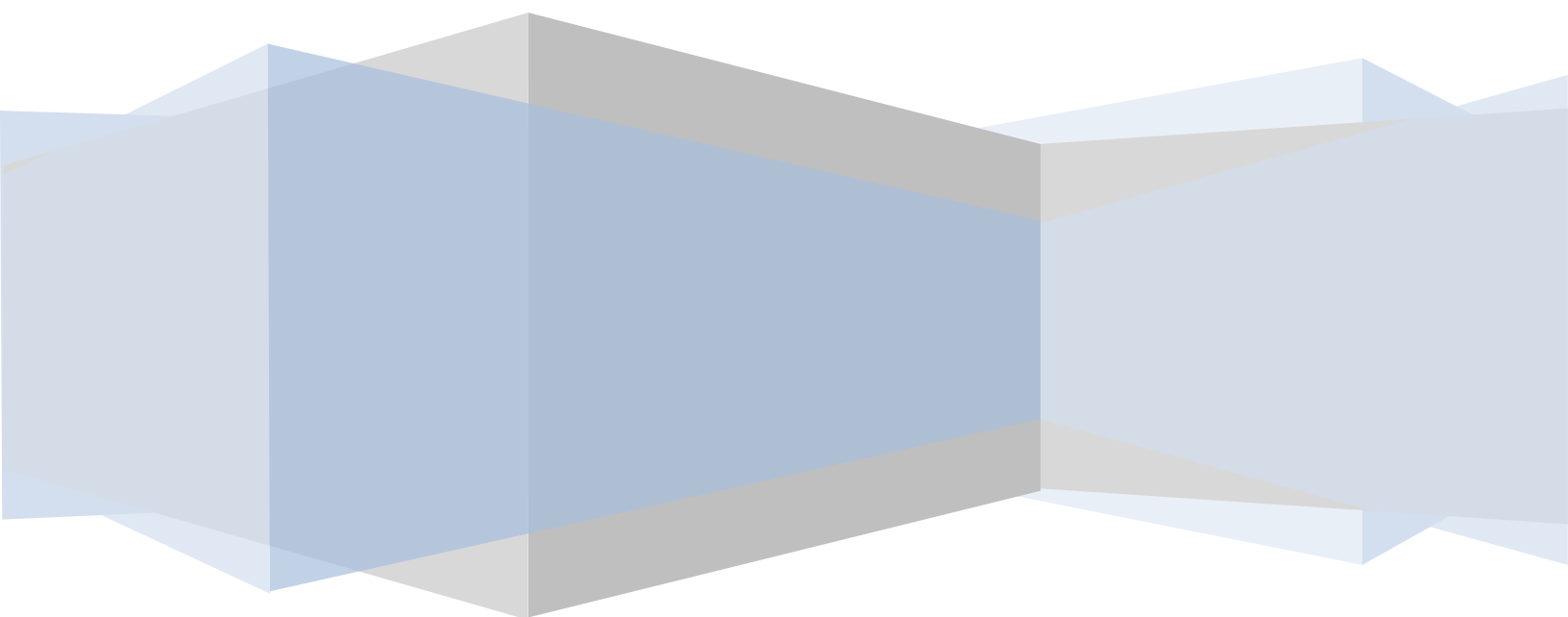


Table of Contents

Table of Contents.....	2
Capitolo 1: Introduzione.....	8
I Tipi di dati	8
Istruzioni di comando.....	10
Operatore logico	11
Editor ABAP/4.....	12
Capitolo 2: Il Dizionario dati.....	12
Domini	13
Dati.....	14
Tabelle.....	15
Strutture.....	15
Oggetti aggregati del dizionario ABAP	16
Visualizzazioni	17
Cerca in Help	18
Blocca gli oggetti.....	19
Capitolo 3: Macro, include, subroutine, moduli e gruppi di funzione	20
Macro SAP-ABAP	20
Programmi Include.....	21
Subroutine	22
Moduli di funzione.....	23
Gruppi di funzioni	24
Capitolo 4: SQL nativo e aperto	Error! Bookmark not defined.
Comandi SQL aperti di base	Error! Bookmark not defined.
Codici di ritorno SQL aperti	Error! Bookmark not defined.
Capitolo 5: tabelle interne SAP	Error! Bookmark not defined.
Che cosa è una Tabella Interna?.....	Error! Bookmark not defined.
Che cosa è una Work Area?	Error! Bookmark not defined.

Differenza tra tavola interna e area di lavoro	Error! Bookmark not defined.
Tipi di tabelle interne	Error! Bookmark not defined.
Creazione delle tabelle interne	Error! Bookmark not defined.
Riempire le tabelle interne	Error! Bookmark not defined.
Copia delle tabelle interne	Error! Bookmark not defined.
Lettura delle tabelle interne	Error! Bookmark not defined.
Cancellazione delle tabelle interne	Error! Bookmark not defined.
Capitolo 6: comandi di tabella ABAP	Error! Bookmark not defined.
Capitolo 7: ABAP programmazione dei report	Error! Bookmark not defined.
Schermata di selezione	Error! Bookmark not defined.
Eventi in un programma di report ABAP	Error! Bookmark not defined.
Formattare il report	Error! Bookmark not defined.
Programmazione interattiva	Error! Bookmark not defined.
Database logici	Error! Bookmark not defined.
Capitolo 8: programmazione di dialogo ABAP	Error! Bookmark not defined.
Differenza fra programmi di report e di dialogo	Error! Bookmark not defined.
Un esempio di elaborazione delle transazioni nella programmazione di dialogo	Error! Bookmark not defined.
Bookmark not defined.	
I componenti del programma di dialogo	Error! Bookmark not defined.
Capitolo 9: Subscreens	Error! Bookmark not defined.
Capitolo 10: Processo su Value & Process su Help	Error! Bookmark not defined.
Processo su Help-Request (POH): F1 Help	Error! Bookmark not defined.
Processo su Value (POV): F4	Error! Bookmark not defined.
Capitolo 11: ALV -- ABAP List Viewer Programming ..	Error! Bookmark not defined.
Report semplice	Error! Bookmark not defined.
Block Report	Error! Bookmark not defined.
Report gerarchici	Error! Bookmark not defined.
Display Variants	Error! Bookmark not defined.
Capitolo 12: Tutto sugli script SAP	Error! Bookmark not defined.

I componenti dello script SAP	Error! Bookmark not defined.
Insieme di layout	Error! Bookmark not defined.
Control Command	Error! Bookmark not defined.
Print Program	Error! Bookmark not defined.
Tipi di output	Error! Bookmark not defined.
Testi e grafica standard	Error! Bookmark not defined.
Capitolo 13: Smart Form	Error! Bookmark not defined.
Confronto tra Smart Form e SapScripts.....	Error! Bookmark not defined.
Vantaggi degli Smart Form	Error! Bookmark not defined.
Architettura degli Smart Form SAP	Error! Bookmark not defined.
Guida agli Smart Form	Error! Bookmark not defined.
Finestre negli Smart Form	Error! Bookmark not defined.
Flusso di programmazione degli Smart Form	Error! Bookmark not defined.
Template	Error! Bookmark not defined.
Capitolo 14: Customer exit e user exit	Error! Bookmark not defined.
Tipi di customer exit.....	Error! Bookmark not defined.
Esempi di customer exit	Error! Bookmark not defined.
Individuare i customer exit	Error! Bookmark not defined.
Crea una customer exit.....	Error! Bookmark not defined.
Che cosa è un USER EXIT?	Error! Bookmark not defined.
Capitolo 15: BADI	Error! Bookmark not defined.
Caratteristiche:	Error! Bookmark not defined.
Definire e implementare un BAPI.....	Error! Bookmark not defined.
Capitolo 16: Tutto sulle query ABAP.....	Error! Bookmark not defined.
Query.....	Error! Bookmark not defined.
InfoSets	Error! Bookmark not defined.
User groups.....	Error! Bookmark not defined.
Componente Translation/QUERY	Error! Bookmark not defined.
Elaborazione dati nelle Query	Error! Bookmark not defined.

CREARE UNA QUERY	Error! Bookmark not defined.
Capitolo 17: SAP BDC -- Batch Data Communication	Error! Bookmark not defined.
Introduzione all'input Batch	Error! Bookmark not defined.
Metodi input batch	Error! Bookmark not defined.
Procedure di input batch	Error! Bookmark not defined.
Scrittura del programma BDC.....	Error! Bookmark not defined.
Creazione della sessione batch di immissione	Error! Bookmark not defined.
Registratore batch di immissione	Error! Bookmark not defined.
Capitolo 18: EDI, ALE e IDOC.....	Error! Bookmark not defined.
Che cosa è un ALE?.....	Error! Bookmark not defined.
Differenza fra ALE e EDI?	Error! Bookmark not defined.
Che cosa è IDOC?.....	Error! Bookmark not defined.
Accorda caratteristiche	Error! Bookmark not defined.
Struttura di un IDOC	Error! Bookmark not defined.
Tipi di IDOC	Error! Bookmark not defined.
Che cosa è un Segment?	Error! Bookmark not defined.
Che cosa è il tipo Extension IDOC?	Error! Bookmark not defined.
Documentazione	Error! Bookmark not defined.
Tipo di messaggio	Error! Bookmark not defined.
Viste IDOC	Error! Bookmark not defined.
Associa profili	Error! Bookmark not defined.
Porte.....	Error! Bookmark not defined.
Questo è so CONFUSING!.....	Error! Bookmark not defined.
Il processo in uscita	Error! Bookmark not defined.
Il processo in arrivo	Error! Bookmark not defined.
Capitolo 19: BAPI	Error! Bookmark not defined.
Come creare una BAPI.....	Error! Bookmark not defined.
Test della BAPI	Error! Bookmark not defined.
Rilasciare e congelare la BAPI.....	Error! Bookmark not defined.

Capitolo 20: RFC	Error! Bookmark not defined.
Che cosa è una RFC?	Error! Bookmark not defined.
Tipi di RFC.....	Error! Bookmark not defined.
Come attribuire un RFC?.....	Error! Bookmark not defined.
Debug di Remote Function Call	Error! Bookmark not defined.

SAP ABAP in Pratica

By Sapprofession

Copyright 2017 – Tutti I Diritti Riservati – Sapprofession

ALL RIGHTS RESERVED. Nessuna parte di questo documento può essere riprodotta o distribuita sotto qualsiasi forma, elettronica o meccanica, inclusa la fotocopia, la registrazione e qualsiasi altro mezzo di memorizzazione, senza il permesso scritto dell'autore.

Capitolo 1: Introduzione

ABAP - Advanced Business Application Programming. It è un linguaggio di programmazione per applicazioni per il sistema SAP R/3. L'ultima versione di ABAP è chiamata ABAP Objects e supporta la programmazione orientata agli oggetti.

SAP eseguirà sia applicazioni scritte utilizzando ABAP/4, la versione precedente, come pure le applicazioni ABAP in ABAP Objects.

Nota che questo tutorial non entrerà in dettagli estesi su costrutti del linguaggio ABAP (che diventano molto noiosi da leggere) ma introdurrà velocemente concetti chiave per farti iniziare cosicché tu possa concentrare la tua attenzione su argomenti più importanti.

I Tipi di dati

La Sintassi per dichiarare una variabile in ABAP e' la seguente -

Data Variable_Name Type Variable_Type

Esempio:

DATA employee_number Type I.

La seguente è riportata una lista di Tipi di Dati supportati da ABAP

Tipi di Dati	Lunghezza iniziale del campo	Lunghezza valida del campo	Valore iniziale	Significato
Tipi numerici				
I	4	4	o	Integer (numero intero)
F	8	8	o	Numero a virgola mobile
P	8	1 -- 16	o	Numero imballato
Tipi di carattere				
C	1	1 -- 65535	'... '	Campo di testo (caratteri alfanumerici)
D	8	8	'00000000 '	Campo data (formato): (YYYYMMDD)
N	1	1 -- 65535	'o... o '	Campo numerico di testo (caratteri numerici)
T	6	6	'000000 '	Campo del tempo (formato): (HHMMSS)
Tipo esadecimale				
X		1 -- 65535	X 'o... o '	Campo esadecimale

Elaborazione dei dati – Assegnazione dei valori

a = 16.

move 16 to a.

write a to b.

- Operazioni aritmetiche

Calcola $a = a * 100$.

Istruzioni di comando

E' importante prendere confidenza con le seguenti istruzioni di comando per costruire i vostri programmi -

- If... EndIf Loop

if [not] exp [and /or [not] exp].

.....

[elseif exp.

.....]

[else.

.....]

Endif.

- Istruzione in base al caso

caso variabile.

when value1.

.....

when value2.

.....

[when others.

.....]

Endcase.

Do.

-While Loop

While <espressione logica>.

.....

.....

Endwhile.

- Do Loop

Fai <n> volte.

.....

.....

Enddo.

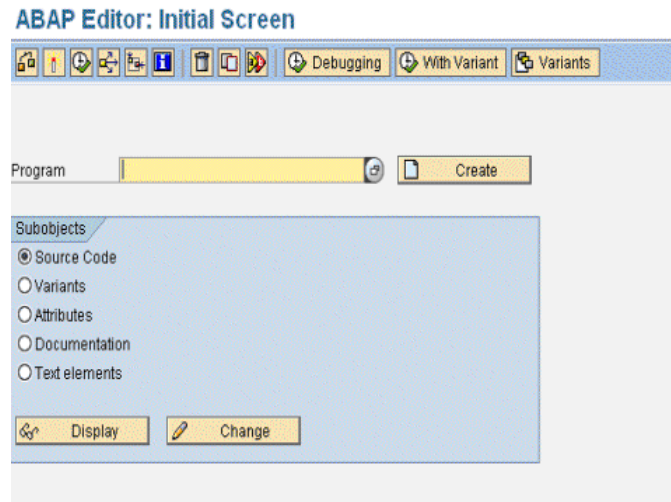
Operatore logico

Di seguito lista di operatori logici per effettuare operazioni:

- GE o >=
- GT o >
- LE o <=
- LT o <
- EQ o =
- NE o <>

Editor ABAP/4

Ed ecco la transazione principale dove passerete la maggior parte del tempo come sviluppatori che creano/modificano programmi. La Transazione **SE38**



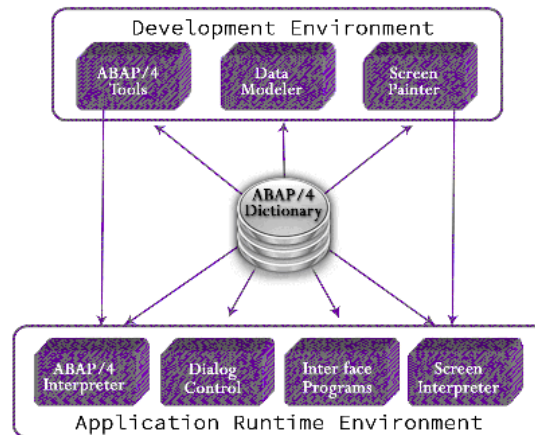
Capitolo 2: Il Dizionario dati

Un dizionario dati è una fonte centrale di informazioni per i dati in un sistema di gestione di informazioni. La sua funzione principale è per sostenere la creazione e la gestione delle definizioni dei dati (o "metadata").

Per che cosa è utilizzato un dizionario dati?

- Gestione delle definizioni dei dati
- Fornitura di informazioni per valutazioni
- Sostegno per sviluppo del software
- Supporto alla documentazione
- Garantire che le definizioni dati siano flessibili e aggiornate

INTEGRATION INTO ABAP/4 WORKBENCH



Gli oggetti nell'ABAP Dictionary risiedevano in tre livelli che supportano la loro capacità di essere riutilizzabili. Questi livelli sono:

1. Tavole e strutture
2. Dati
3. Domini

Esaminiamoli in dettaglio -

Domini

- Descrivono le caratteristiche tecniche di un campo di tabella
- Specificano una gamma di valori che descrive i valori dei dati consentiti nei campi
- I campi che si riferiscono allo stesso dominio (tramite i dati assegnati a loro) vengono modificati quando viene apportata una modifica al dominio
- Assicurano la coerenza

Es. Acquisto del numero di documento (EBELN)

Dictionary: Display Domain

Domain: EBELN Active

Short Description: Purchasing document number

Attributes Definition Value range

Formatting

Data type: CHAR Character string

No. characters: 10

Decimal places: 0

Output characteristics

Output length: 10

Convers. routine: ALPHA

☐ Sign

☐ Lowercase

Dati

- Il campo dati descrive il ruolo svolto da un campo in un contesto tecnico
- I campi dello stesso significato semantico possono riferirsi allo stesso dato
- Contiene le informazioni sul campo

Es. Acquisto del numero di documento (EBELN)

Dictionary: Display Data Element

Data element: EBELN Active

Short Description: Purchasing Document Number

Attributes Data Type Further Characteristics Field label

☒ Elementary type

☒ Domain

EBELN Purchasing document ...

Data Type: CHAR Character string

Length: 10 Decimal Places: 0

☐ Built-in type

Data type:

Length: 0 Decimal places: 0

☐ Reference type

☐ Name of Ref. Type

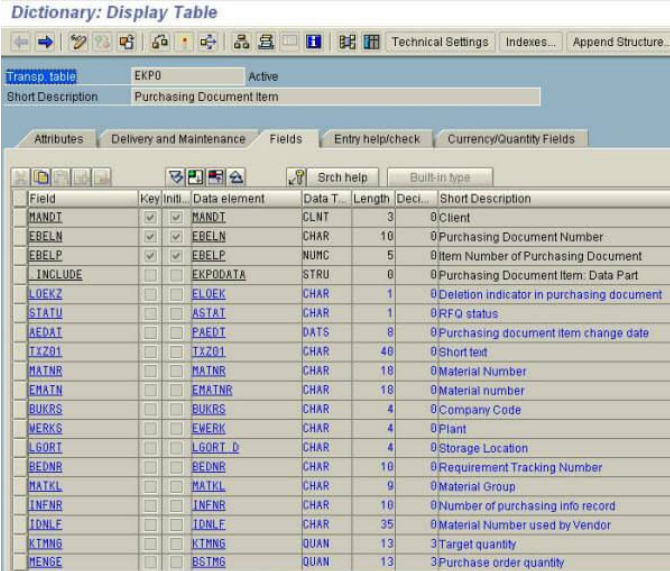
☐ Reference to Predefined Type

Data Type

Length: 0 Decimal Places: 0

Tabelle

- Rappresenta le tabelle di Database in cui risiedono i dati.
- Le tavole possono essere definite indipendentemente dal database nel dizionario ABAP.
- I campi della tavola sono definiti con i loro tipi di dati e con le loro lunghezze SAP ABAP (indipendenti dal database)



Field	Key	Init.	Data element	Data T...	Length	Deci...	Short Description
MANDT	☒	☒	MANDT	CLNT	3		0 Client
EBELN	☒	☒	EBELN	CHAR	10		0 Purchasing Document Number
EBELP	☒	☒	EBELP	NUMC	5		0 Item Number of Purchasing Document
INCLUDE	☐	☐	EKPODATA	STRU	0		0 Purchasing Document Item: Data Part
LOEKZ	☐	☐	ELOEK	CHAR	1		0 Deletion indicator in purchasing document
STATU	☐	☐	ASTAT	CHAR	1		0 RFQ status
AEDAT	☐	☐	PAEDT	DATS	8		0 Purchasing document item change date
TXZ01	☐	☐	TXZ01	CHAR	40		0 Short text
MATNR	☐	☐	MATNR	CHAR	18		0 Material Number
EHATN	☐	☐	EHATNR	CHAR	18		0 Material number
BUKRS	☐	☐	BUKRS	CHAR	4		0 Company Code
WERKS	☐	☐	EWERK	CHAR	4		0 Plant
LGORT	☐	☐	LGORT_D	CHAR	4		0 Storage Location
BEDNR	☐	☐	BEDNR	CHAR	10		0 Requirement Tracking Number
MATKL	☐	☐	MATKL	CHAR	9		0 Material Group
INFNR	☐	☐	INFNR	CHAR	10		0 Number of purchasing info record
IDNLF	☐	☐	IDNLF	CHAR	35		0 Material Number used by Vendor
KTMNG	☐	☐	KTMNG	QUAN	13		3 Target quantity
MENGE	☐	☐	RSTMG	QUAN	13		3 Purchase order quantity

Strutture

- Sono dichiarazioni di registrazione non corrispondono a una tabella di Database.
- Esattamente come il tipo dati definiti dall'utente.
- Definito come una tabella e può essere quindi reindirizzato dai programmi ABAP.
- Le strutture contengono dati solo durante il runtime di un programma.

Dictionary: Display Structure

Structure: BEKPO Active
Short Description: Transfer Structure Items for Purchasing Documents

Attributes Components Entry help/check Currency/quantity fields

1 / 256

Component	RT	Component type	Data Type	Length	Deci.	Short Description
INCLUDE		EKPD	NUMC	0		0 Purchasing Document Item
MANDT		MANDT	CLNT	3		0 Client
EBELN		EBELN	CHAR	10		0 Purchasing Document Number
EBELP		EBELP	NUMC	5		0 Item Number of Purchasing Document
INCLUDE		EKPODATA	NUMC	0		0 Purchasing Document Item: Data Part
LOEKZ		ELOEK	CHAR	1		0 Deletion indicator in purchasing document
STATU		ASTAT	CHAR	1		0 RFQ status
AEDAT		PAEDT	DATS	8		0 Purchasing document item change date
TXZ01		TXZ01	CHAR	40		0 Short text
MATNR		MATNR	CHAR	18		0 Material Number
EMATN		EMATNR	CHAR	18		0 Material number
BUKRS		BUKRS	CHAR	4		0 Company Code
WERKS		EWERK	CHAR	4		0 Plant
LGORT		LGORT_D	CHAR	4		0 Storage Location
BEDNR		BEDNR	CHAR	10		0 Requirement Tracking Number
MATKL		MATKL	CHAR	9		0 Material Group
INFNR		INFNR	CHAR	10		0 Number of purchasing info record
TMN F		TMN F	CHAR	15		0 Material Number used by Vendor

Oggetti aggregati del dizionario ABAP

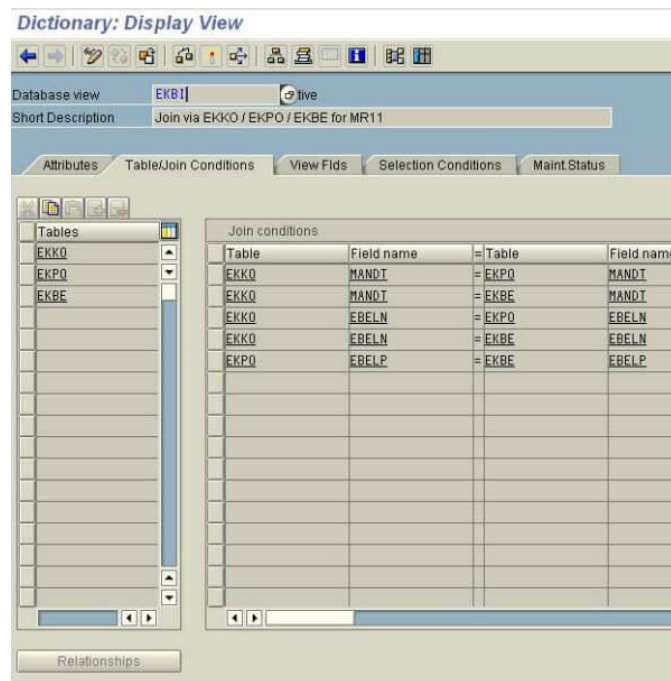
Mezzi aggregati che sono formati da numerosi componenti. Nel dizionari ABAP, gli oggetti aggregati sono oggetti che vengono da varie tabelle trasparenti.

1. Visualizzazioni
2. Cerca in Help
3. Blocca gli oggetti

Esaminiamoli in dettaglio

Visualizzazioni

- Sono utilizzate in SAP _ ABAP per riassumere dati distribuiti tra numerose tavole
- I dati di una visualizzazione non sono effettivamente memorizzati fisicamente. I dati di una visualizzazione sono derivati invece da una o più tabelle
- È adattato alle esigenze di un'applicazione specifica



Cerca in Help

- Un ricerca “help” è uno strumento che consente di cercare i record di dati nel sistema
- Una ricerca efficiente e facile da utilizzare aiuta utenti che non conoscono la chiave di un record

Dictionary: Display Search Help

Collective srch.hlp: S_MAT1 Active

Short description: Find Material Number

Attributes Definition Included search helps

Srch. help exit

Parameter

Search help parameter	Imp	Ex...	Data element	Default value
MATNR	☒	☒	MATNR	
WERKS	☒	☒	WERKS_D	
MATNR_B	☒	☒	MATNR	
PLNTY	☒	☒	CC_PLNTY	
PLNNR	☒	☒	CC_PLNNR	
PLNAL	☒	☒	CC_PLNAL	
MATNR_HTN	☒	☒	EMATN	
VKORG	☒	☒	VKORG	
VTWEG	☒	☒	VTWEG	

Blocca gli oggetti

- L'accesso simultaneo dello stesso record di dati da parte di due utenti nel sistema SAP viene sincronizzato da un meccanismo di blocco.
- I blocchi vengono impostati e rilasciati chiamando alcuni moduli di funzione. Questi moduli di funzione vengono generati automaticamente dalla definizione di cosiddetti oggetti di blocco nel dizionario ABAP/4

Moduli di funzione: Enqueue_<obj name>, per bloccare la tabella
dequeue_<obj name>, per rilasciare il blocco



Operazioni importanti

- **SE11:** schermata iniziale del dizionario dei dati (solo sul display SE12)
- **SE13:** dizionario ABAP: impostazioni tecniche
- **SE14:** utilità dei database
- **SE15:** sistema informativo sui repository
- **SE16:** visualizzatore dei dati
- **SE17:** tabella display generale
- **SE55:** manutenzione tabella visualizzazione
- **SM30:** manutenzione tabella

Capitolo 3: Macro, include, subroutine, moduli e gruppi di funzione

Quando si modula il codice sorgente, si inserisce una sequenza di istruzioni ABAP in un modulo. Quindi, invece di mettere tutte le dichiarazioni nel programma principale, basta chiamare il modulo. Quando il programma viene generato, il codice sorgente nell'unità di modularizzazione viene trattato come se fosse effettivamente fisicamente presente nel programma principale.

Necessità di modularizzazione

- migliora la struttura del programma.
- Facilita lettura del codice
- Facilita mantenimento del codice
- Evita la ridondanza e promuove il riutilizzo del codice

Diverse tecniche di modularizzazione

- Uso delle macro
- Uso dei file di inclusione
- Subroutine
- Moduli di funzione

Esaminiamoli in dettaglio:

Macro SAP-ABAP

Se si desidera riutilizzare lo stesso insieme di istruzioni più di una volta in un programma, è possibile includerle in una macro.

E' possibile utilizzare solo una macro all'interno del programma in cui è definita e può essere richiamata solo in righe del programma in seguito alla sua definizione.

Le macro possono essere utili per lunghi calcoli o istruzioni WRITE complesse.

Sintassi

DEFINE <macro_name>

'Macro statements

END-OF-DEFINITION

Le macro possono utilizzare Parametri &N dove N = 1,2,3...

Esempio:-

DATA: number1 TIPO I VALUE 1.

DEFINE increment.

ADD 1 to &1.

WRITE &1.

END-OF-DEFINITION.

Increment number1.

WRITE number1.

Output: 2

Programmi Include

I programmi Include sono esclusivamente per la modularizzazione del codice sorgente e non hanno alcuna interfaccia di parametro. I programmi Include permettono di utilizzare lo stesso codice sorgente in vari programmi. Essi possono essere utili se si dispone di lunghe istruzioni dei dati che si desidera utilizzare in diversi programmi.

Sintassi

Include <include Program Name>

Nota

- i programmi Include non possono chiamare se stessi.
- i programmi Include devono contenere dichiarazioni complete.

Esempio:

INCLUDE ZILX0004.

WRITE: /'User', SY-UNAME, / 'date', SY-DATUM.

=====

PROGRAM ZRPM0001.

INCLUDE ZILX0004.

Subroutine

Le subroutine sono procedure che è possibile definire in qualsiasi programma ABAP e chiamare anche da qualsiasi programma. Le subroutine vengono generalmente chiamate internamente, cioè, contengono sezioni di codice o algoritmi che sono utilizzati frequentemente localmente. Se si desidera che una funzione sia riutilizzabile per tutto il sistema, utilizzare un modulo di funzione.

Sintassi-

FORM <Subroutine> [<pass>].

<Statement Block>.

ENDFORM.

<Subroutine> = Nome della subroutine

<pass> = i parametri sono passati

Tipi di subroutine

1. Interna
 - Subroutine definita nello stesso programma che viene chiamato.
 - Può accedere a tutti i dati objects dichiarati nel programma principale ABAP/4.
2. Esterna
 - La subroutine viene definita al di fuori del programma che viene chiamato.
 - E' necessario utilizzare l'opzione <pass> o dichiarare i dati objects in parti comuni di memoria.

Chiamare una subroutine Subroutine Interna

PERFORM <subroutine> [<pass>]

<subroutine> = Nome della subroutine

<pass> = i parametri sono passati

Sono automaticamente disponibili i dati dichiarati nel programma principale.

Subroutine esterna

PERFORM <subroutine>(<Program>) [<pass>].

PERFORM <subroutine> (<Program>) [<pass>] [se trovato].

PERFORM (<subroutine>) IN PROGRAM (<Program>) [<pass>] [IF FOUND].

PERFORM <index> OF <subroutine1> <subroutine2> <subroutine3> [<pass>].

Note

- Le chiamate nidificate sono permesse in subroutine (ad es PERFORM all'interno di un FORM...) (ENDFORM).
- Sono possibili anche le chiamate ricorsive.
- Per definire dati locali, utilizzare l'istruzione DATA dopo FORM. Ogni volta che si immette la subroutine, i dati vengono ricreati (con un valore iniziale) e staccati alla fine (dalla pila).
- Per definire dati globali utilizzati all'interno di una subroutine, utilizzate l'istruzione LOCAL dopo FORM. I valori vengono salvati quando si immette la subroutine e poi rilasciati alla fine (dalla pila)

Moduli di funzione

I moduli di funzione sono routine generiche ABAP/4 che chiunque può utilizzare. Infatti, esistono molti moduli di funzioni standard.

I moduli di funzione sono organizzati per gruppi di funzione: raccolte di funzioni logicamente collegate. Un modulo di funzione appartiene sempre ad un gruppo di funzione.

Sintassi-

FUNCTION <function module>

<Statements>

ENDFUNCTION.

Informazioni importanti associate al modulo di funzione

- Amministrazione
- Importazione / modifica / esportazione dei parametri.
- Parametri della tabella / eccezioni.
- Documentazione
- Codice sorgente -- L<fgrp>U01 . <fgrp> è il Gruppo di Funzione
- Dati globali -- dati L<fgrp>TOP . Dati globali per il gruppo di funzione - Accessibile attraverso moduli di funzione nel gruppo di funzioni.
- Programma principale -- SAPL<fgrp> . Contiene la lista di tutti i file di inclusione per quel gruppo di funzioni

Chiamare un modulo di funzione

Per chiamare un modulo di funzione, utilizzare l'istruzione CALL FUNCTION:

CALL FUNCTION <module>

[EXPORTING f1 = un 1...]. [f n = un n]

[IMPORTING f1 = un 1...]. [f n = un n]

[CHANGING f1 = un 1...]. [f n = un n]

[TABLES f1 = un 1...]. [f n = un n]

[EXCEPTIONS e1 = r 1...]. E n = r n [ERROR_MESSAGE = r E]

[OTHERS = ro]].

Gruppi di funzioni

I gruppi di funzioni sono contenitori per moduli di funzione. Infatti, ci sono molti gruppi di funzione standard. Tutti i moduli di funzione in un gruppo di funzioni possono accedere ai dati globali del gruppo.

Come i programmi eseguibili (tipo 1) e gruppi di moduli (tipo M), i gruppi di funzioni possono contenere schermate, schermate di selezione ed elenchi.

Note

- Impossibile eseguire i gruppi di funzione
- Il nome di un gruppo di funzioni può essere lungo fino a 26 caratteri.
- Quando si crea un gruppo di funzioni o un modulo di funzione, il programma principale e i programmi Include vengono generati automaticamente.
- I gruppi di funzione incapsulano i dati.

Come creare un Gruppo di Funzione

1. Goto Transaction **SE80**.
2. Scegli il programma nel DropDown.
3. Scrivere il nome del Gruppo di Funzione che si vuole creare. Generalmente i gruppi di funzione creati dall'utente iniziano con "Z". Ad es.
<Z_FUNCTION_GROUP_NAME>. Premere Invio.
4. Osservare che l'operazione TOP Include è creata per default se l'utente controlla l'opzione creare un TOP Include.

Come creare un Modulo di Funzione

1. Creare una funzione Gruppo (dire " ZCAL").
2. Creare un modulo di funzione, impostare gli attributi come (Function Group, Application, Short Text e Process Type) e salvare.
3. Il file Include "LZCALU01" avrà il codice sorgente del primo modulo di funzione.
4. Il file Include "LZCALTOP" avrà dati globali.
5. Il programma principale "SAPLZCAL" contiene
 - Dati globali file Include "LZCALTOP"
 - I moduli di funzione includono il file "LZCALUXX"
 - File Include definiti dall'utente "LZCALF.", LZCALO. "" e "LZCALI".
6. Definire i parametri di interfaccia e le eccezioni
7. Scrivere il codice sorgente
8. Attivare modulo di funzione
9. Testare il modulo di funzione -- Single Test & Debugging
10. Documentazione e rilascio di un modulo di funzione

Tutto questo per Modularità in ABAP.