

# Creare un estrattore con Function Module

## 1. Creare struttura estrattiva

Per prima cosa vai in se11

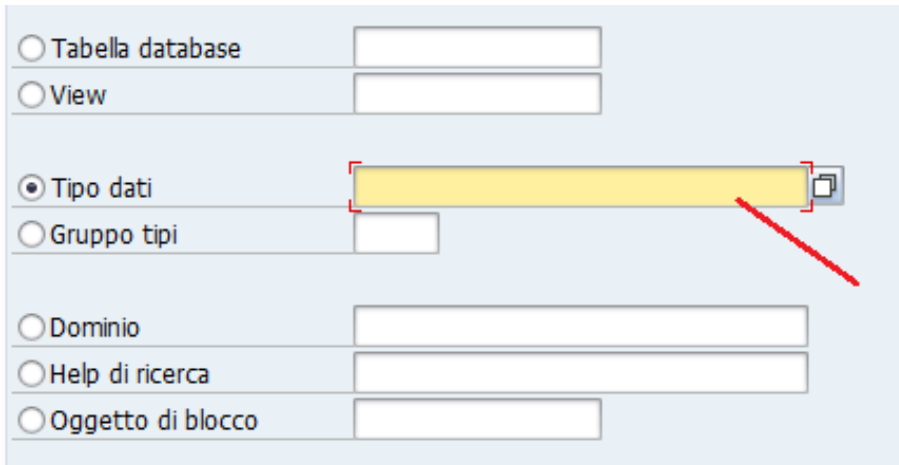


Tabella database

View

Tipo dati

Gruppo tipi

Dominio

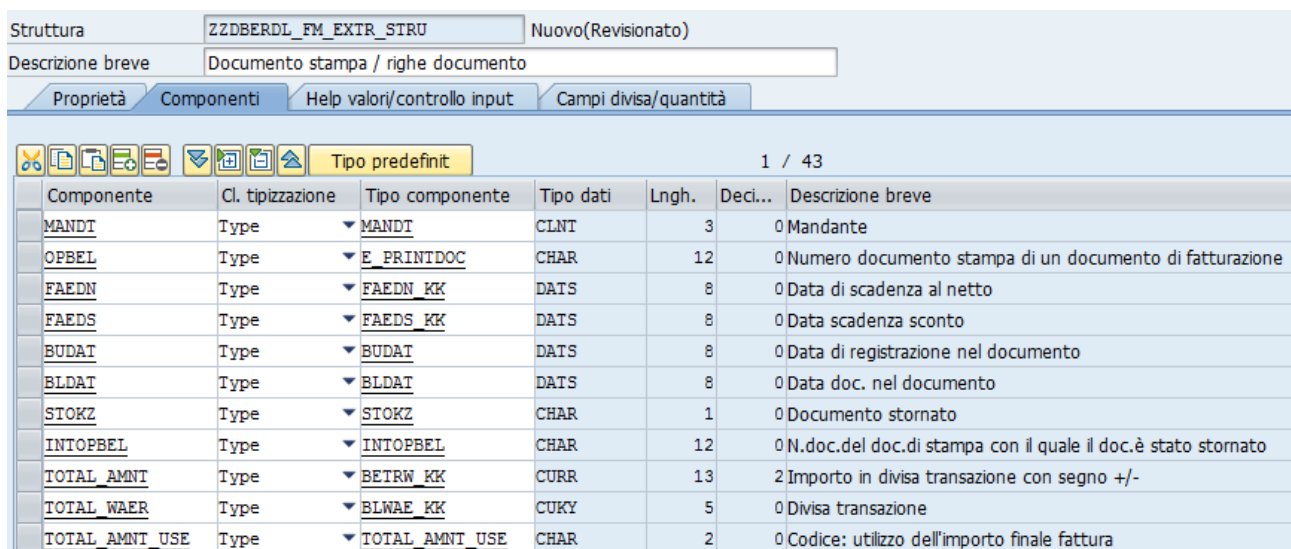
Help di ricerca

Oggetto di blocco

e metti un nome in Tipo Dati

(ovviamente un tipo dati che inizia con Z e che non esiste).

Poi fai crea e inizia a mettere i campi e i tipo campi che mi servono nella struttura.



Struttura: ZZDBERDL\_FM\_EXTR\_STRU Nuovo(Revisionato)

Descrizione breve: Documento stampa / righe documento

Proprietà Componenti Help valori/controllo input Campi divisa/quantità

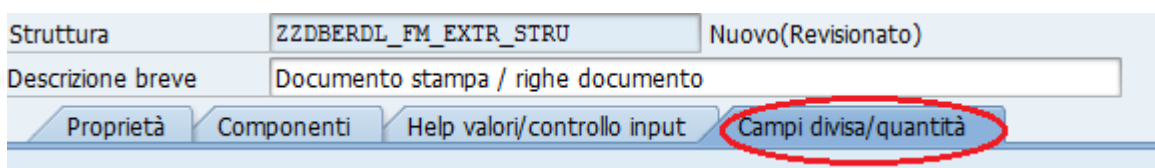
Tipo predefinit 1 / 43

| Componente     | Cl. tipizzazione | Tipo componente | Tipo dati | Lngh. | Deci... | Descrizione breve  |
|----------------|------------------|-----------------|-----------|-------|---------|--|
| MANDT          | Type             | MANDT           | CLNT      | 3     | 0       | Mandante   |
| OPBEL          | Type             | E_PRINTDOC      | CHAR      | 12    | 0       | Numero documento stampa di un documento di fatturazione      |
| FAEDN          | Type             | FAEDN_KK        | DATS      | 8     | 0       | Data di scadenza al netto                                    |
| FAEDS          | Type             | FAEDS_KK        | DATS      | 8     | 0       | Data scadenza sconto   |
| BUDAT          | Type             | BUDAT           | DATS      | 8     | 0       | Data di registrazione nel documento                          |
| BLDAT          | Type             | BLDAT           | DATS      | 8     | 0       | Data doc. nel documento                                      |
| STOKZ          | Type             | STOKZ           | CHAR      | 1     | 0       | Documento stornato   |
| INTOPBEL       | Type             | INTOPBEL        | CHAR      | 12    | 0       | N.doc.del doc.di stampa con il quale il doc.è stato stornato |
| TOTAL_AMNT     | Type             | BETRW_KK        | CURR      | 13    | 2       | Importo in divisa transazione con segno +/-                  |
| TOTAL_WAER     | Type             | BLWAE_KK        | CUKY      | 5     | 0       | Divisa transazione   |
| TOTAL_AMNT_USE | Type             | TOTAL_AMNT_USE  | CHAR      | 2     | 0       | Codice: utilizzo dell'importo finale fattura                 |

In questo caso ho preso quello fatto copiando i campi dalla tabella erdk .

### IMPORTANTE:

entrare in campi divisa/quantità



Struttura: ZZDBERDL\_FM\_EXTR\_STRU Nuovo(Revisionato)

Descrizione breve: Documento stampa / righe documento

Proprietà Componenti Help valori/controllo input Campi divisa/quantità

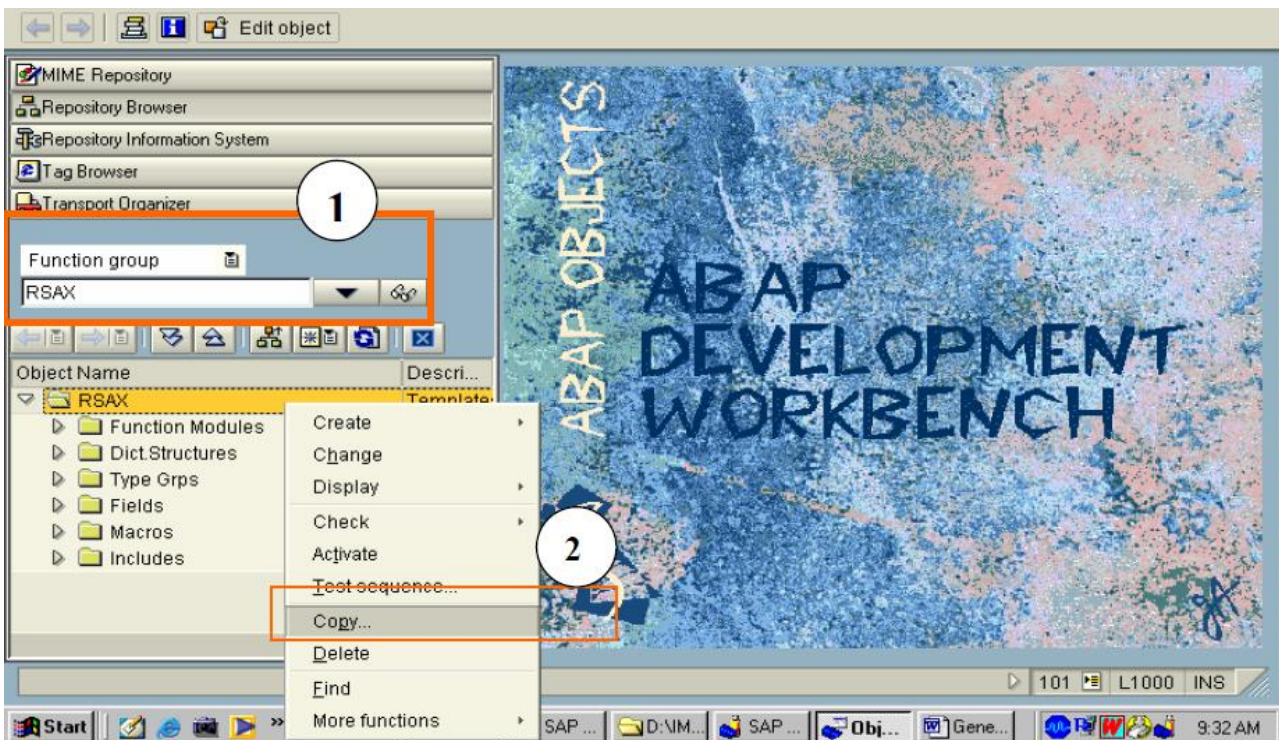
E associare ai campi divisa la tabella e il campo di riferimento

| Componente  | Cl. tipizzazione | Tp. componente | Tipo dati | Tab. di rif.       | Campo rif. | Descrizione breve  |
|-------------|------------------|----------------|-----------|--------------------|------------|--|
| SPARTE      | Type             | SPARTE         | CHAR      |                    |            | Settore merceologico   |
| VERTRAG     | Type             | VERTRAG        | CHAR      |                    |            | Contratto  |
| BUCHREL     | Type             | BUCHREL        | CHAR      |                    |            | Riga fattura rilevante per la registrazione                  |
| BRUTTOZEILE | Type             | BRUTTOZEILE    | CHAR      |                    |            | La riga fattura è una riga lordo                             |
| SHKZG       | Type             | SHKZG          | CHAR      |                    |            | Indicatore dare/avere  |
| XTOTAL AMNT | Type             | XTOTAL AMNT    | CHAR      |                    |            | Considerare partite nell'importo finale fattura              |
| NETTOBTR    | Type             | NETTOBTR       | CURR      | ZZDBERDL_FM_EXT... | TWAERS     | Importo netto riga fattura                                   |
| BETRW       | Type             | BETRW_KK       | CURR      | ZZDBERDL_FM_EXT... | TWAERS     | Importo in divisa transazione con segno +/-                  |
| AUGBW       | Type             | AUGBW_KK       | CURR      | ZZDBERDL_FM_EXT... | TWAERS     | Importo di pareggio in divisa di transazione (con segno +/-) |
| SBASW       | Type             | SBASW_KK       | CURR      | ZZDBERDL_FM_EXT... | TWAERS     | Imp. imponibile in divisa transazione(imp. base imponibile)  |
| SBETW       | Type             | SBETW_KK       | CURR      | ZZDBERDL_FM_EXT... | TWAERS     | Importo imposte in divisa transazione con segno +/-          |
| SKTOW       | Type             | SKTOW_KK       | CURR      | ZZDBERDL_FM_EXT... | TWAERS     | Importo sconto in divisa di transazione con segno +/-        |

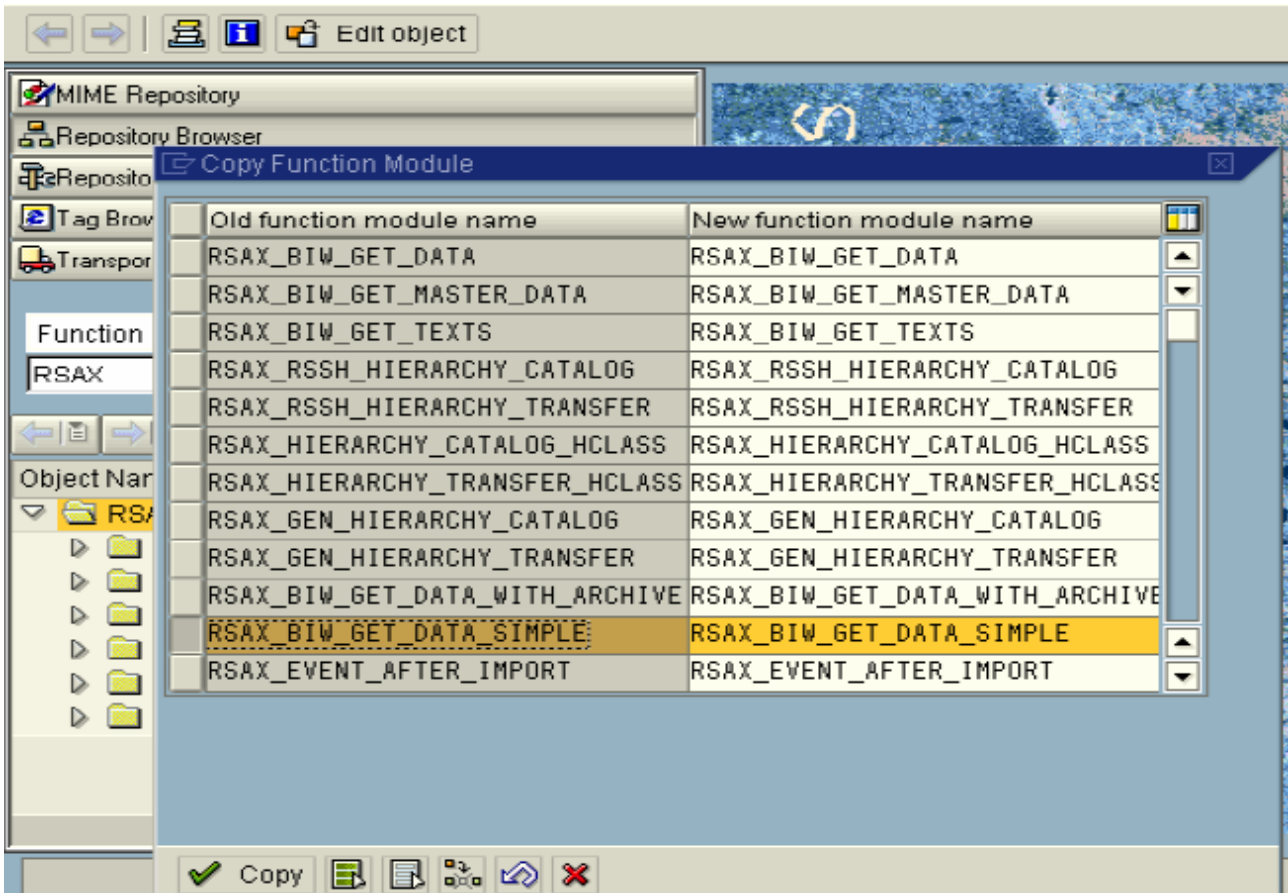
se no non funziona.

## 2. Creare il gruppo funzioni

Transazione **SE80** e vai su gruppo funzioni e selezioni RSAX e lo devi copiare



E chiamarlo con ZRSAX. Una volta copiato si aprirà questo:

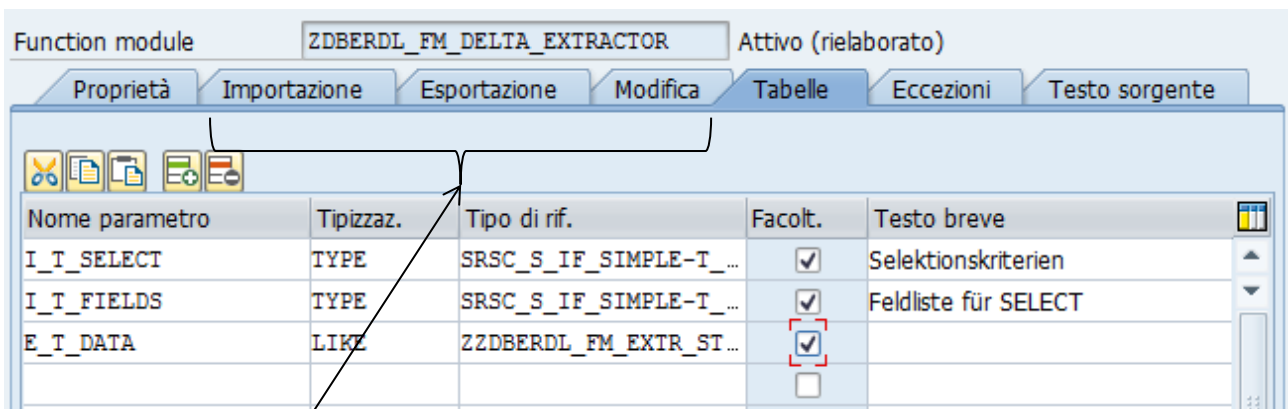


Selezionare SOLO quello evidenziato.

A questo punto posso creare il Function Module.

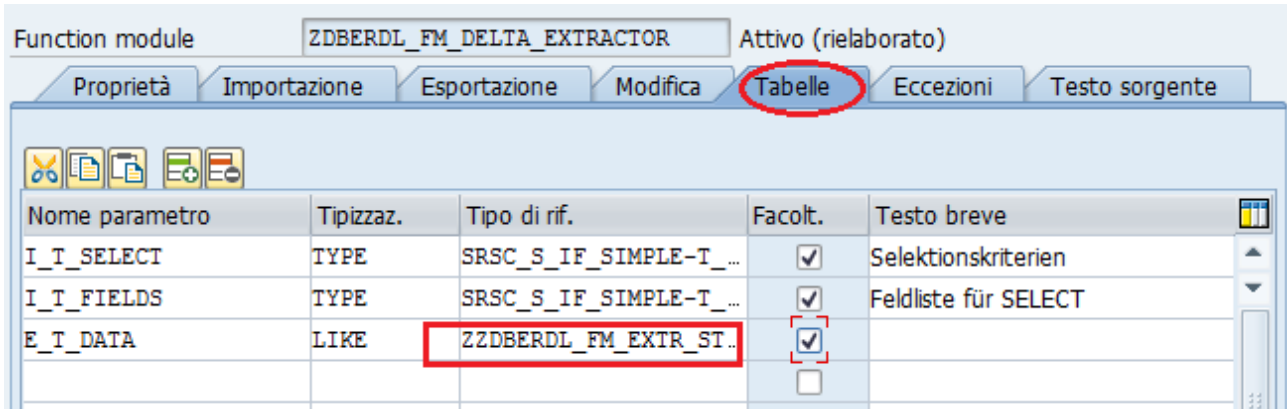
### 3. Function Module

Parto copiando il function module che ho in ZRSAX sotto la voce function module cambiandogli ovviamente il nome chiamandolo con il nome che gli voglio dare io.



Non cambiare niente!!!

IMPORTANTE



Ricordarsi di mettere in E\_T\_DATA la struttura estrattiva che abbiamo creato al passo 1.

Poi si va al testo sorgente e si implementa la logica per quello che deve fare l'estrattore:

Sotto riporto il codice dell'estrattore che sto facendo giusto per capire quali parti cambiare e quali lasciare com'è.

```

FUNCTION ZDBERDL_FM_DELTA_EXTRACTOR.
*-----
***"Interfaccia locale:
*  IMPORTING
*    VALUE(I_REQUNR) TYPE  SRSC_S_IF_SIMPLE-REQUNR
*    VALUE(I_DSOURCE) TYPE  SRSC_S_IF_SIMPLE-DSOURCE OPTIONAL
*    VALUE(I_MAXSIZE) TYPE  SRSC_S_IF_SIMPLE-MAXSIZE OPTIONAL
*    VALUE(I_INITFLAG) TYPE  SRSC_S_IF_SIMPLE-INITFLAG OPTIONAL
*    VALUE(I_READ_ONLY) TYPE  SRSC_S_IF_SIMPLE-READONLY OPTIONAL
*    VALUE(I_REMOTE_CALL) TYPE  SBIWA_FLAG DEFAULT SBIWA_C_FLAG_OFF
*  TABLES
*    I_T_SELECT TYPE  SRSC_S_IF_SIMPLE-T_SELECT OPTIONAL
*    I_T_FIELDS TYPE  SRSC_S_IF_SIMPLE-T_FIELDS OPTIONAL
*    E_T_DATA STRUCTURE  ZZDBERDL_FM_EXTR_STRU OPTIONAL
*  EXCEPTIONS
*    NO_MORE_DATA
*    ERROR_PASSED_TO_MESS_HANDLER
*-----

* Example: DataSource for table SFLIGHT
TABLES: ZZDBERDL_FM_EXTR_STRU, DBERDL.

* Auxiliary Selection criteria structure
DATA: L_S_SELECT TYPE SRSC_S_SELECT.
DATA: W_T_DATA LIKE LINE OF E_T_DATA.
DATA: ZZSTPRZ TYPE STEUERSATZ.

* Maximum number of lines for DB table
STATICS: S_S_IF TYPE SRSC_S_IF_SIMPLE,

* counter
S_COUNTER_DATAPAKID LIKE SY-TABIX,

* cursor
S_CURSOR TYPE CURSOR.

* Select ranges
RANGES: L_R_OPBEL FOR ZZERDK_FM_EXTR_STRU-OPBEL,
L_R_ERDAT FOR ZZERDK_FM_EXTR_STRU-ERDAT,
L_R_ZZDATE FOR ZZERDK_FM_EXTR_STRU-ZZDATE.

* Initialization mode (first call by SAPI) or data transfer mode
* (following calls) ?
IF I_INITFLAG = SBIWA_C_FLAG_ON.

*****
* Initialization: check input parameters
* buffer input parameters
* prepare data selection
*****

* Check DataSource validity
CASE I_DSOURCE.
WHEN 'ZDBERDL_FM_EXTR'
WHEN OTHERS.

```

Qua si mette il nome della struttura che abbiamo creato e le tabelle da cui faccio la select

Mi definisco le tabelle che saranno usate dopo

Qua definisco il range per cui lavora l'estrattore e questi range me li valorizzo con il loop

NON SI CAMBIA NIENTE

Qua si mette il nome del data source che ancora non abbiamo creato in realtà

```

        IF 1 = 2. MESSAGE E009(R3). ENDIF.
* this is a typical log call. Please write every error message like this
    LOG_WRITE 'E'                "message type
              'R3'              "message class
              '009'            "message number
              I_DSOURCE        "message variable 1
              ' '              "message variable 2
        RAISE ERROR_PASSED_TO_MESS_HANDLER.
    ENDCASE.

    APPEND LINES OF I_T_SELECT TO S_S_IF-T_SELECT.

* Fill parameter buffer for data extraction calls
    S_S_IF-REQUNR    = I_REQUNR.
    S_S_IF-DSOURCE  = I_DSOURCE.
    S_S_IF-MAXSIZE  = I_MAXSIZE.

* Fill field list table for an optimized select statement
* (in case that there is no 1:1 relation between InfoSource fields
* and database table fields this may be far from being trivial)
    APPEND LINES OF I_T_FIELDS TO S_S_IF-T_FIELDS.

ELSE.                "Initialization mode or data extraction ?

*****
* Data transfer: First Call      OPEN CURSOR + FETCH
*                               Following Calls FETCH only
*****

* First data package -> OPEN CURSOR
    IF S_COUNTER_DATAPAKID = 0.

* Fill range tables BW will only pass down simple selection criteria
* of the type SIGN = 'I' and OPTION = 'EQ' or OPTION = 'BT'.

```

NON SI CAMBIA NIENTE

```

    LOOP AT S_S_IF-T_SELECT INTO L_S_SELECT WHERE FIELDNM = 'OPBEL'.
        MOVE-CORRESPONDING L_S_SELECT TO L_R_OPBEL.
        APPEND L_R_OPBEL.
    ENDLOOP.

```

```

    LOOP AT S_S_IF-T_SELECT INTO L_S_SELECT WHERE FIELDNM = 'ERDAT'.
        MOVE-CORRESPONDING L_S_SELECT TO L_R_ERDAT.
        APPEND L_R_ERDAT.
    ENDLOOP.

```

Mi riempio una tabella interna che poi uso come condizione nella select ed è il range definito prima

```

    LOOP AT S_S_IF-T_SELECT INTO L_S_SELECT WHERE FIELDNM = 'ZZDATE'.
        MOVE-CORRESPONDING L_S_SELECT TO L_R_ZZDATE.
        L_R_ZZDATE-SIGN = 'I'.
        L_R_ZZDATE-OPTION = 'GE'.
        CLEAR L_R_ZZDATE-HIGH.
        APPEND L_R_ZZDATE.
    ENDLOOP.

```

Qua mi faccio il loop in cui gli dico di riempire le tabelle che mi so definito anche se non ho capito bene che fa sta parte

read per FETCH statement

Questa è la parte che gestirà il delta perché si crea l'intervallo di date

\* from input parameter I\_MAXSIZE. If there is a one to one relation  
\* between DataSource table lines and database entries, this is trivial.  
\* In other cases, it may be impossible and some estimated value has to  
\* be determined.

```
OPEN CURSOR WITH HOLD S_CURSOR FOR
SELECT ERDK~MANDT
       ERDK~OPBEL
       ERDK~FAEDS
       ERDK~BUDAT
       ERDK~BLDAT
       ERDK~STOKZ
       ERDK~INTOPBEL
       ERDK~TOTAL_AMNT
       ERDK~TOTAL_WAER
       ERDK~TOTAL_AMNT_USE
       ERDK~PARTNER
       ERDK~VKONT
       ERDK~ERDAT
       ERDK~ERNAM
       ERDK~AEDAT
       ERDK~STO_OPBEL
       ERDK~ICREASON
       ERDK~EXBEL
       ERDK~MAZAE_ABS
       ERDK~INVDOCNO
       DBERDL~PRINTDOCLINE
       DBERDL~BELZART
       DBERDL~BUKRS
       DBERDL~SPARTE
       DBERDL~VERTRAG
       DBERDL~BUCHREL
       DBERDL~BRUTTOZEILE
       DBERDL~SHKZG
       DBERDL~XTOTAL_AMNT
       DBERDL~NETTOBTR
       DBERDL~BETRW
       DBERDL~AUGBW
       DBERDL~SBASW
       DBERDL~SBETW
       DBERDL~SKTOW
       DBERDL~TWAERS
       DBERDL~MWSKZ
       DBERDL~HVORG
       DBERDL~TVORG
       DBERDL~SKTPZ
       DBERDL~AB
       DBERDL~BIS
       DBERDL~STPRZ
FROM ERDK INNER JOIN DBERDL ON ( ERDK~OPBEL = DBERDL~PRINTDOC )
WHERE ( ERDAT IN L_R_ZZDATE OR AEDAT IN L_R_ZZDATE )
AND OPBEL IN L_R_OPBEL
AND "ERDAT IN L_R_ERDAT AND
INVOICED = 'X' AND
TOBRELEASD NE 'X' AND SIMULATED NE 'X' ).
```

Qui mi faccio la select in  
con l'open cursor che mi  
modifica direttamente le  
tabelle assegnando i campi

Questa è la parte che gestirà il delta  
perché mmi ero riempito il range e  
gli faccio selezionare solo quello che  
appartiene al range

```
ENDIF.                                "First data package ?
```

```
* Fetch records into interface table.  
*   named E_T_'Name of extract structure'.
```

```
FETCH NEXT CURSOR S_CURSOR  
  APPENDING CORRESPONDING FIELDS  
  OF TABLE I_ZDBERDL  
  PACKAGE SIZE S_S_IF-MAXSIZE.
```

Il FETCH è simile alla SELECT però nei FM si usa questo

```
IF SY-SUBRC <> 0.  
  CLOSE CURSOR S_CURSOR.  
  RAISE NO_MORE_DATA.  
ENDIF.
```

```
IF I_ZDBERDL[] IS NOT INITIAL.
```

```
  LOOP AT I_ZDBERDL ASSIGNING <FS_ZDBERDL>.  
    IF <FS_ZDBERDL>-AEDAT IS INITIAL.  
      MOVE <FS_ZDBERDL>-ERDAT TO W_T_DATA-ZZDATE.  
    ELSE.  
      MOVE <FS_ZDBERDL>-AEDAT TO W_T_DATA-ZZDATE.  
    ENDIF.  
  
    MOVE-CORRESPONDING <FS_ZDBERDL> TO W_T_DATA.  
    MOVE <FS_ZDBERDL>-ZZSTPRZ TO W_T_DATA-ZZSTPRZ.  
    APPEND W_T_DATA TO E_T_DATA.  
  ENDLOOP.  
  CLEAR I_ZDBERDL.  
ENDIF.
```

Qui c'è la logica vera e propria del Function Module ossia gli sto dicendo che cosa fare

```
S_COUNTER_DATAPAKID = S_COUNTER_DATAPAKID + 1.
```

```
ENDIF.                                "Initialization mode or data extraction ?
```

```
ENDFUNCTION.
```

### 3.1 Dichiarazioni nella TOP

A questo punto bisogna dichiarare le varie tabella nella TOP del gruppo funzioni ossia nella parte in cui le dichiarazioni sono valide per tutti i function module che appartengono a quel gruppo funzioni:



Function module: ZDBERDL\_FM\_DELTA\_EXTRACTOR Attivo

Proprietà | Importazione | Esportazione | Modifica | Tabelle | Eccezioni | Testo sorgente

Classificazione

Gruppo funzioni: ZRSAX Templates for application APIs

Testo breve: FM Estrazione da Righe documento di stampa in delta

Modalità d'esecuzione

Function module standard

Modulo accessibile in remoto  basXML supportato

Modulo di registrazione

Avvio immed.

Lancio immed. - senza reg. succ.

Avvio differito

Es. collettiva

Dati generali

Responsabile: AGASPARRO

Autore ult. mod.: AGASPARRO

Data di modifica: 15.01.2016

Pacchetto: ZBW\_REP\_DI

Nome del programma: SAPLZRSAX

Nome Include: LZRSAXU02

Lingua originale: DE

Non rilasciato

Blocco editor

Globale

clickare due volte sul nome programma ( quello cerchiato). A questo punto si apre una schermata tipo ABAP in cui bisogna far doppio click su:

Gruppo funzioni: SAPLZRSAX Attivo

```

1 *****
2 * System-defined Include-files. *
3 *****
4 INCLUDE LZRSAXTOP. " Global Data
5 INCLUDE LZRSAXUXX. " Function Modules
6
7 *****
8 * User-defined Include-files (if necessary). *
9 *****
10 * INCLUDE LRSAXF... " Subprograms
11 * INCLUDE LRSAXO... " PBO-Modules
12 * INCLUDE LRSAXI... " PAI-Modules

```

Ossia sul TOP e qui si aprirà un programma ABAP in cui fare la definizione (è tipo le global definition delle start e end routine)

```

FUNCTION-POOL ZRSAX. "MESSAGE-ID ..

* Include LRSAXD01 can be directly referenced by application APIs !!
INCLUDE lrsaxd01.

* Include macros for "describe field"
INCLUDE rsaucmac.

```

Non cambiare

```

TYPES: BEGIN OF ZERDK,
        MANDT          TYPE MANDT,
        OPBEL          TYPE E_PRINTDOC,
        FAEDN          TYPE FAEDN_KK,
        FAEDS          TYPE FAEDS_KK,
        BUDAT          TYPE BUDAT,

```

```

BLDAT          TYPE BLDAT,
STOKZ          TYPE STOKZ,
INTOPBEL      TYPE INTOPBEL,
TOTAL_AMNT    TYPE BETRW_KK,
TOTAL_WAER    TYPE BLWAE_KK,
TOTAL_AMNT_USE TYPE TOTAL_AMNT_USE,
PARTNER       TYPE BU_PARTNER,
VKONT         TYPE VKONT_KK,
ERDAT         TYPE ERDAT,
ERNAM         TYPE ERNAM,
AEDAT         TYPE AEDAT,
STO_OPBEL     TYPE STO_OPBEL,
ICREASON      TYPE BCREASON,
EXBEL         TYPE EXBEL_KK,
MAZAE_ABS     TYPE MAZAE_KK,
ZZ_LIMIT      TYPE ZAISB_LIMIT,
ZZ_RATEIZZABILE TYPE ZAISB_RATEIZZABILE,
INVDOCNO      TYPE INVDOCNO_KK,
ZZSTPRZ       TYPE STEUERSATZ,
ZZDATE        TYPE ERDAT,
END OF ZERDK.

```

FIELD-SYMBOLS: <FS\_ZERDK> TYPE ZERDK.

DATA : I\_ZERDK TYPE STANDARD TABLE OF ZERDK.

```

TYPES: BEGIN OF ZDBERDL,
MANDT          TYPE MANDT,
OPBEL          TYPE E_PRINTDOC,
FAEDN          TYPE FAEDN_KK,
FAEDS          TYPE FAEDS_KK,
BUDAT          TYPE BUDAT,
BLDAT          TYPE BLDAT,
STOKZ          TYPE STOKZ,
INTOPBEL      TYPE INTOPBEL,
TOTAL_AMNT    TYPE BETRW_KK,
TOTAL_WAER    TYPE BLWAE_KK,
TOTAL_AMNT_USE TYPE TOTAL_AMNT_USE,
PARTNER       TYPE BU_PARTNER,
VKONT         TYPE VKONT_KK,
ERDAT         TYPE ERDAT,
ERNAM         TYPE ERNAM,
AEDAT         TYPE AEDAT,
STO_OPBEL     TYPE STO_OPBEL,
ICREASON      TYPE BCREASON,
EXBEL         TYPE EXBEL_KK,
MAZAE_ABS     TYPE MAZAE_KK,
ZZ_LIMIT      TYPE ZAISB_LIMIT,
ZZ_RATEIZZABILE TYPE ZAISB_RATEIZZABILE,
INVDOCNO      TYPE INVDOCNO_KK,
ZZSTPRZ       TYPE STEUERSATZ,
ZZDATE        TYPE ERDAT,
PRINTDOCLINE  TYPE BELZEILE,
BELZART        TYPE BELZART,
BUKRS         TYPE BUKRS,

```

```

SPARTE          TYPE SPARTE,
VERTRAG        TYPE VERTRAG,
BUCHREL        TYPE BUCHREL,
BRUTTOZEILE    TYPE BRUTTOZEILE,
SHKZG          TYPE SHKZG,
XTOTAL_AMNT    TYPE XTOTAL_AMNT,
NETTOBTR       TYPE NETTOBTR,
BETRW          TYPE BETRW_KK,
AUGBW          TYPE AUGBW_KK,
SBASW          TYPE SBASW_KK,
SBETW          TYPE SBETW_KK,
SKTOW          TYPE SKTOW_KK,
TWAERS         TYPE TWAERS,
MWSKZ          TYPE MWSKZ,
HVORG          TYPE HVORG_KK,
TVORG          TYPE TVORG_KK,
SKTPZ          TYPE SKTPZ_KK,
AB             TYPE ABZEITSCH,
BIS            TYPE BISZEITSCH,
END OF ZDBERDL.

```

```

FIELD-SYMBOLS: <FS_ZDBERDL> TYPE ZDBERDL.
DATA : I_ZDBERDL TYPE STANDARD TABLE OF ZDBERDL.

```

Definizione della tabella con tutti i campi . Con il BEGIN OF e END OF mi definisco la struttura poi mi definisco un FIELD-SIMBOL per quella struttura e poi una tabella del tipo della struttura che ho definito

```

* Data and macros for tracing of BI Content Extraction
DATA g_r_tracer TYPE REF TO if_rsap_extraction_tracer.
INCLUDE rsap_extraction_tracing_bice.

* Constants
CONSTANTS: rsax_c_flag_on VALUE 'X',
           rsax_c_flag_off VALUE ' '.

* Type-pools (TNS_220101)
TYPE-POOLS:
  rsaot,
  rsazt.

* Select ranges
RANGES:
  g_r_carrid FOR sbook-carrid,
  g_r_connid FOR sbook-connid,
  g_r_fldate FOR sbook-fldate,
  g_r_bookid FOR sbook-bookid.

* Global data for reading from archive.
DATA:
  g_extrmode TYPE rsazt_extrmode,
  g_step_extrmode TYPE rsazt_extrmode,
  g_handle TYPE i.

* ===== Required by RSAX_BIW_GET_SEGM
* General
DATA: g_no_more_data,

```

NON TOCCARE

```

g_s_params TYPE roextrprms.

* ---- requested fields
DATA: g_t_segfields TYPE TABLE OF rssegfdsel,
      g_t_fields1 TYPE sbiwa_t_fields,
      g_t_fields2 TYPE sbiwa_t_fields,
      g_t_fields3 TYPE sbiwa_t_fields,
      g_t_fields4 TYPE sbiwa_t_fields,
      g_t_fields5 TYPE sbiwa_t_fields,
      g_t_fields6 TYPE sbiwa_t_fields.

* ---- Length
DATA: BEGIN OF g_s_length,
      roosource TYPE i,
      roosourcet TYPE i,
      roosseg TYPE i,
      roosfield TYPE i,
      roohiecat TYPE i,
      roohiecom TYPE i,
      END OF g_s_length.

```

NON TOCCARE

Ora il Function Module è finito. Manca solo come gestire il Delta e in questo caso devo farlo su ZZDATE .

#### 4. Creazione Data source

Vado in RSO2 e metto il nome del datasource che voglio creare e poi faccio crea

e poi mi compilo la schermata

che si apre:

## Creare DataSource per Dati movimento: ZDBERDL\_FM\_EXTR

Delta generico

|                   |                          |                     |
|-------------------|--------------------------|---------------------|
| DataSource        | ZDBERDL_FM_EXTR          | Estrazione da view  |
| Comp. applicativo | FI-CA                    | Estrazione da query |
| Sincron. dati     | <input type="checkbox"/> | Estrazione med. FM  |
| Stato ogg.        | Nvo                      |                     |

### Testi

|                    |   |
|--------------------|---|
| Descrizione breve  | Posizioni DDS                             |
| Descrizione media  | Posizioni Documenti di Stampa             |
| Descrizione estesa | Posizioni Documenti di Stampa "Fatturone" |

### Estrazione da view DB

|                 |  |
|-----------------|--|
| View/tabella    |  |
| Struttura estr. |  |

### Estrazione da SAP Query

|         |  |
|---------|--|
| InfoSet |  |
|---------|--|

### Estrazione mediante function mod.

|                 |                            |
|-----------------|----------------------------|
| Function module | ZDBERDL_FM_DELTA_EXTRACTOR |
| Struttura estr. | ZZDBERDL_FM_EXTR_STRU      |

E poi impostare il delta nella stessa schermata premendo su delta generico. Cliccando si apre:

Delta generico

DataSource

Campo di determinazione delta

Nm.campo

Time stamp (UTC)  
 Timestamp (locale)  
 Giorno cal.




Parametrizzazione

Limite sup. intervallo sicurezza

Limite inf. intervallo sicurezza

Ab.tempo reale

Nuovo stato p.record modificati  
 Delta aggiunt.

Nel nome campo ci metto ZZDATE ci lascio Time Stamp perché così è definito il tipo campo (ZZDATE) e poi bisogna mettere 1 a limite superiore intervallo di sicurezza:

Delta generico

DataSource ZDBERDL\_FM\_EXTR

Campo di determinazione delta

Nm.campo ZZDATE

Time stamp (UTC)  
 Timestamp (locale)  
 Giorno cal.

Parametrizzazione

Limite sup. intervallo sicurezza 1

Limite inf. intervallo sicurezza

Ab. tempo reale

Nuovo stato p.record modificati  
 Delta aggiunt.

Quell'1 sta a indicare che l'estrattore tira fuori i dati fino al giorno prima. In realtà in questo caso non dovrebbe servire a niente perché questo l'ho messo già mediante codice con il GE del loop di ZZDATE e GE significa Greater equal ossia maggiore uguale e serve proprio a questo 1. Ora salvo e il data source è fatto!