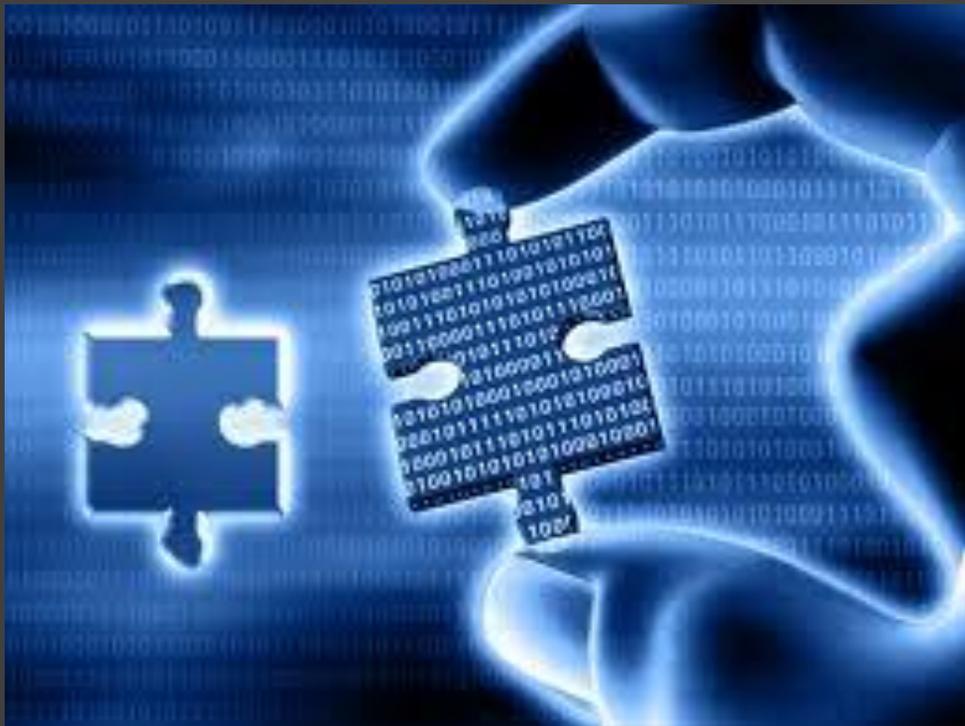


LA PROGRAMMAZIONE IN SAP CON IL LINGUAGGIO ABAP/4



- COMPLETO -

INDICE

Struttura delle Tabelle SAP	pag. 7
Navigazione nelle Tabelle SAP	pag. 12
Creazione di un Dominio	pag. 16
Creazione di un Tipo Dati	pag. 22
Creazione di una Tabella personalizzata	pag. 28
Creazione di una Chiave Esterna in una Tabella	pag. 38
Creazione di un Help di Ricerca in una Tabella	pag. 42
Creazione di un Programma Report	pag. 49
Dichiarazione di Tabelle e Variabili in un programma	Pag. 57
Elenco delle Variabili di Sistema	pag. 61
Variabili: elenco delle varie tipologie	pag. 66
Creazione dei Parametri di Selezione in un programma	pag. 69
L'istruzione SELECT	pag. 77
Le Tabelle Interne e relative istruzioni	pag. 96
Le Condizioni	pag. 106
I Cicli	pag. 109
Stampa a Video dei dati di output di un programma	pag. 112

INDICE

Caricamento di un File sul Server SAP in ABAP	pag. 119
Lettura di un File dal Server SAP in ABAP	pag. 124
Le istruzioni FORM e PERFORM	pag. 128
L'istruzione INCLUDE	pag. 131
Il Debug	pag. 134
Creazione di una Variante	pag. 145
Creazione di un programma ALV	pag. 151
Schedulazione di un Programma tramite Job	pag. 162
Functions e BAPI – Analisi	Pag. 167
Functions e BAPI – Utilizzo	pag. 178
Creazione di un Enhancements	pag. 187
Creazione di un Batch Input	pag. 193
Creazione di uno Smartforms – Layout	pag. 213
Creazione di uno Smartforms – Stili	pag. 249
Creazione di uno Smartforms – il programma ABAP	pag. 259
Utility: attivare l'Editor SAP negli Smartforms	pag. 266
Creazione di un Module Pool	pag. 270

INDICE

Creazione di un Job	pag. 296
Visualizzazione di un Job	pag. 305
Utility: caricamento rapido di un file sul Server SAP	pag. 308
Utility: scaricamento rapido di un file dal Server SAP	pag. 311
Utility: caricamento di immagini su SAP	pag. 314

Il linguaggio di Programmazione ABAP/4

Abbiamo scelto come argomento di questa demo alcune delle attività più importanti che vengono svolte in ambito programmazione ABAP/4.

Ci auguriamo che questa demo ti possa permettere di verificare, non solo l'utilità dei nostri e-book, ma anche la qualità con cui essi sono stati implementati.

Il linguaggio di Programmazione ABAP/4

L'ABAP/4 è il linguaggio di programmazione proprietario di SAP per creare e personalizzare il Sistema tramite programmi custom. Di base, l'ABAP sfrutta le potenzialità del linguaggio SQL migliorandone le performance (ovviamente se le istruzioni sono ben implementate). In questo E-Book vengono utilizzati esempi specifici per ogni tematica, partendo quindi dall'SQL SAP (chiamato Open SQL) arrivando fino alle Utility più complesse come i Batch Input, Smartforms e Module Pool. L'E-Book è stato pensato per essere un valido supporto durante l'attività di studio o l'attività lavorativa e vista la molteplicità delle operazioni disponibili in ABAP vengono forniti esempi schematici nel modo più chiaro possibile.

L'ISTRUZIONE "SELECT" (TRANSAZIONE SE38)

L'istruzione "SELECT"

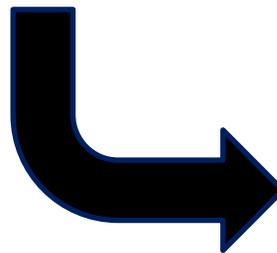
L'istruzione "SELECT" permette al programma di prelevare dei dati da una tabella SAP e poi elaborarli a seconda delle esigenze del programmatore. Possono ad esempio essere stampati a video, trasportati in altre tabelle o combinati per avere calcoli o output particolari.

Esistono varie tipologie di SELECT ed ognuna deve avere obbligatoriamente la tabella di riferimento (istruzione FROM) ed in modo facoltativo la condizione di selezione (istruzione WHERE). Gli esempi di SELECT sono basati sulla tabella MARA (anagrafica materiale).

SELECT *.

Questa SELECT permette la selezione di tutti i campi di una tabella senza che il programmatore debba elencarli tutti. Per scrivere in output il campo tabella si utilizza l'istruzione "WRITE". Se dopo l'istruzione è presente il carattere / vuol dire il record successivo viene scritto direttamente a capo.

```
SELECT * FROM MARA.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```



```
1300-300  
1300-310  
1300-311  
1300-312  
1300-320  
1300-321  
1300-330  
1300-331  
1300-340  
1300-341  
1300-342  
1300-350  
350-130  
350-140  
350-150  
350-160  
350-170  
350-180
```

L'istruzione "SELECT"

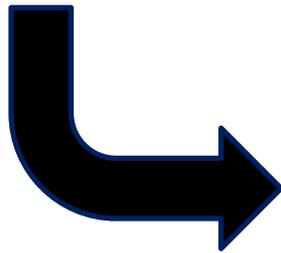
SELECT campo1, campo2, ecc.

Questa SELECT permette la selezione dei singoli campi di una tabella indicati dal programmatore. Anche qui per scrivere in output il campo tabella si utilizza l'istruzione "WRITE".

I dati che vengono estratti dalla tabella devono essere salvati in altrettante variabili di appoggio.

```
TABLES: MARA.
DATA: CODICE_MATERIALE LIKE MARA-MATNR.
DATA: TIPO_MATERIALE LIKE MARA-MTART.

SELECT MATNR MTART FROM MARA INTO (CODICE_MATERIALE, TIPO_MATERIALE).
  WRITE: / CODICE_MATERIALE, TIPO_MATERIALE.
ENDSELECT.
```



1300-300	HALB
1300-310	ROH
1300-311	ROH
1300-312	ROH
1300-320	ROH
1300-321	ROH
1300-330	HALB
1300-331	HALB
1300-340	ROH
1300-341	ROH
1300-342	ROH
1300-350	ROH
350-130	HAWA
350-140	HAWA
350-150	HAWA
350-160	HAWA
350-170	HAWA
350-180	HAWA
350-190	HAWA
40-100C	HALB
40-100F	HALB
40-100D	HALB

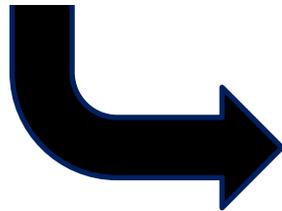
L'istruzione "SELECT"

SELECT con CONDIZIONE

Qualsiasi SELECT, se seguita dalla condizione "WHERE", permette di filtrare il risultato secondo parametri decisi dal programmatore.

Questa istruzione coinvolge anche simboli di confronto tra oggetti, analizzati in questa sezione.

```
SELECT * FROM MARA WHERE MTART = 'HAWA'.
WRITE: / MARA-MATNR, MARA-MTART.
ENDSELECT.
```

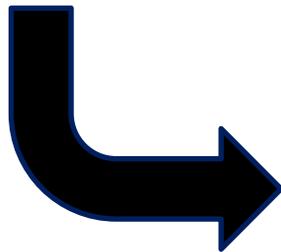


350-130	HAWA
350-140	HAWA
350-150	HAWA
350-160	HAWA
350-170	HAWA
350-180	HAWA
350-190	HAWA
1300-900	HAWA
1400-100	HAWA
1400-200	HAWA
1400-300	HAWA
1400-310	HAWA
1400-315	HAWA
1400-400	HAWA
1400-715	HAWA
1417	HAWA
1427	HAWA
1400-740	HAWA

L'istruzione "SELECT"

Se si hanno più condizioni, esse devono essere legate tra di loro. Nel caso in cui tutte queste debbano essere soddisfatte si utilizza l'istruzione "AND".

```
SELECT * FROM MARA WHERE MTART = 'HAWA' AND ERSDA > '20130101'.
WRITE: / MARA-MATNR, MARA-MTART, MARA-ERSDA.
ENDSELECT.
```

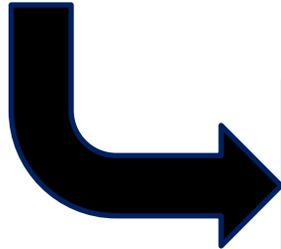


8124	HAWA	25.02.2013
8125	HAWA	25.02.2013
8126	HAWA	25.02.2013
8127	HAWA	25.02.2013
8128	HAWA	25.02.2013
8138	HAWA	25.02.2013
8141	HAWA	25.02.2013
8142	HAWA	25.02.2013
8145	HAWA	26.02.2013
7833	HAWA	14.02.2013
7832	HAWA	14.02.2013
7908	HAWA	15.02.2013
7909	HAWA	18.02.2013
7923	HAWA	18.02.2013
7920	HAWA	18.02.2013
7919	HAWA	18.02.2013
7781	HAWA	13.02.2013
7836	HAWA	14.02.2013
POLVERE D'ORO	HAWA	14.02.2013
7929	HAWA	18.02.2013

L'istruzione "SELECT"

Se invece l'output deve dipendere dal fatto che non tutte le condizioni, ma solo alcune vengano soddisfatte, si utilizza l'istruzione "OR".

```
SELECT * FROM MARA WHERE MTART = 'HAWA' OR ERSDA > '20130101'.
WRITE: / MARA-MATNR, MARA-MTART, MARA-ERSDA.
ENDSELECT.
```



350-130	HAWA	10.09.2003
350-140	HAWA	10.09.2003
350-150	HAWA	10.09.2003
350-160	HAWA	10.09.2003
350-170	HAWA	10.09.2003
350-180	HAWA	10.09.2003
350-190	HAWA	10.09.2003
1300-900	HAWA	11.05.1995
1400-100	HAWA	19.02.1996
1400-200	HAWA	19.02.1996
1400-300	HAWA	10.12.1996
1400-310	HAWA	10.12.1996
1400-315	HAWA	15.04.2003
1400-400	HAWA	19.02.1996
1400-715	HAWA	10.12.1996
1417	HAWA	07.12.2006
1427	HAWA	12.01.2007
1400-740	HAWA	19.02.1996
1400-750	HAWA	19.02.1996
400-510	HAWA	24.05.1995

L'istruzione "SELECT"

Tra il campo tabella ed il valore da comparare si hanno simboli di confronto.

Per l'uguaglianza si usa il simbolo "=" o "EQ". EQ significa Equal to.

```
SELECT * FROM MARA WHERE MTART EQ 'HAWA'.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```

```
SELECT * FROM MARA WHERE MTART = 'HAWA'.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```

Per il simbolo di maggiore si usa il simbolo "<>" o "NE". NE significa Not equal.

```
SELECT * FROM MARA WHERE MTART NE 'HAWA'.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```

```
SELECT * FROM MARA WHERE MTART <> 'HAWA'.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```

Per il simbolo di minore si usa il simbolo "<" o "LT". LT significa Less than.

```
SELECT * FROM MARA WHERE MTART LT 'HAWA'.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```

```
SELECT * FROM MARA WHERE MTART < 'HAWA'.  
WRITE: / MARA-MATNR.  
ENDSELECT.
```

L'istruzione "SELECT"



Per il simbolo di maggiore si usa il simbolo ">" o "GT". GT significa Greater than.

```
SELECT * FROM MARA WHERE MTART GT 'HAWA'.  
  WRITE: / MARA-MATNR.  
ENDSELECT.
```

```
SELECT * FROM MARA WHERE MTART > 'HAWA'.  
  WRITE: / MARA-MATNR.  
ENDSELECT.
```

Per il simbolo di minore-uguale si usa il simbolo "<=" o "LE". LE significa Less equal.

```
SELECT * FROM MARA WHERE MTART LE 'HAWA'.  
  WRITE: / MARA-MATNR.  
ENDSELECT.
```

```
SELECT * FROM MARA WHERE MTART <= 'HAWA'.  
  WRITE: / MARA-MATNR.  
ENDSELECT.
```

Per il simbolo di minore-uguale si usa il simbolo ">=" o "GE". GE significa Greater equal.

```
SELECT * FROM MARA WHERE MTART GE 'HAWA'.  
  WRITE: / MARA-MATNR.  
ENDSELECT.
```

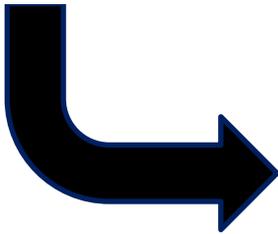
```
SELECT * FROM MARA WHERE MTART >= 'HAWA'.  
  WRITE: / MARA-MATNR.  
ENDSELECT.
```

L'istruzione "SELECT"

SELECT SINGLE

Questa SELECT permette la selezione del primo record che corrisponde alle condizioni impostate. Se esiste più di un record che corrisponde al criterio di scelta viene infatti considerato solo il primo di questi in ordine di lettura. Notare che la SELECT SINGLE non richiede l'uso della ENDSELECT.

```
SELECT SINGLE * FROM MARA WHERE MTART EQ 'HAWA'.
WRITE: / MARA-MATNR, MARA-MTART.
```



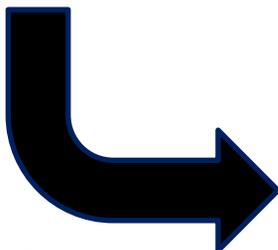
350-130	HAWA
---------	------

SELECT MIN / SELECT MAX

Queste SELECT permettono la selezione del record con il valore minimo o massimo che corrisponde alle condizioni impostate. Esse richiedono il salvataggio del valore in corrispettive variabili d'appoggio. Le SELECT MIN e SELECT MAX non richiedono l'istruzione ENDSELECT.

```
TABLES: MARA.
DATA: MAX LIKE MARA-MATNR.
DATA: MIN LIKE MARA-MATNR.
```

```
SELECT MAX( MATNR ) MIN( MATNR ) FROM MARA INTO (MAX, MIN).
WRITE: MAX, MIN.
```



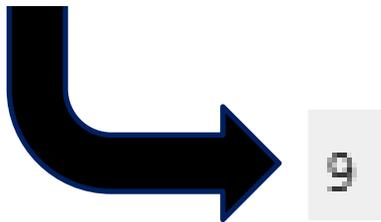
0	23
---	----

L'istruzione "SELECT"

SELECT COUNT

Questa SELECT permette il conteggio dei record che soddisfano la condizione imposta dal programmatore . Essa richiede il salvataggio del valore in una corrispettiva variabile d'appoggio. La SELECT COUNT non richiede l'istruzione ENDSELECT.

```
DATA: CONTATORE TYPE N.
SELECT COUNT (*) FROM MARA INTO CONTATORE WHERE MTART = 'HAWA' .
WRITE: CONTATORE.
```

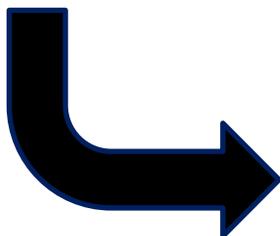


9

SELECT ANNIDATE

Una SELECT può a sua volta contenere un'altra SELECT. In questo caso, la SELECT più interna legge i dati dalla seconda tabella a fronte del record letto nella SELECT più esterna.

```
SELECT * FROM MARA WHERE MTART = 'HAWA' .
  SELECT * FROM MAKT WHERE MATNR = MARA-MATNR.
  WRITE: / MARA-MATNR, MARA-MTART, MAKT-MAKTX.
  ENDSELECT.
ENDSELECT.
```



350-130	HAWA Window
350-130	HAWA Window
350-130	HAWA Window
350-130	HAWA Farbe
350-130	HAWA Window
350-130	HAWA Couleur
350-130	HAWA Window
350-130	HAWA 塗料
350-130	HAWA Tinta / cor
350-130	HAWA Kpacka
350-130	HAWA Color

L'istruzione "SELECT"

SELECT E TRASPORTO DATI IN TABELLA INTERNA

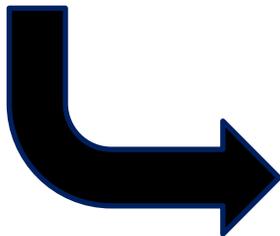
Questa SELECT permette di estrarre tutti i contenuti o parte dei contenuti da una tabella per inserirli direttamente in una tabella interna. Il requisito fondamentale per eseguire correttamente la SELECT è che la Tabella Interna deve essere dichiarata con la dicitura "OCCURS n WITH HEADER LINE". Notare che per la scrittura in output della Tabella Interna è necessario il ciclo LOOP. . La SELECT non richiede l'istruzione ENDSELECT.

```
TABLES: MARA.
```

```
DATA: COPIA_MARA LIKE MARA OCCURS 0 WITH HEADER LINE.
```

```
SELECT * FROM MARA INTO TABLE COPIA_MARA WHERE MTART = 'HAWA'.  
] LOOP AT COPIA_MARA.
```

```
WRITE: / COPIA_MARA-MATNR, COPIA_MARA-MTART.  
ENDLOOP.
```



350-130	HAWA
350-140	HAWA
350-150	HAWA
350-160	HAWA
350-170	HAWA
350-180	HAWA
350-190	HAWA
1300-900	HAWA
1400-100	HAWA
1400-200	HAWA
1400-300	HAWA
1400-310	HAWA
1400-315	HAWA
1400-400	HAWA
1400-715	HAWA
1417	HAWA
1427	HAWA
1400-740	HAWA
1400-750	HAWA
400-510	HAWA
400-511	HAWA

L'istruzione "SELECT"

SELECT E TRASPORTO DATI IN TABELLA INTERNA CON CAMPI UGUALI

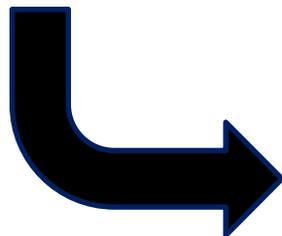
Questa SELECT permette di estrarre tutti i contenuti o parte dei contenuti da una tabella per inserirli direttamente in una tabella interna che ha gli stessi identici campi della tabella sorgente.

L'uguaglianza dei campi non è riferita al numero, ma al nome campo. Il requisito fondamentale per eseguire correttamente la SELECT è che la Tabella Interna deve essere dichiarata con la dicitura "OCCURS n WITH HEADER LINE". Notare che per la scrittura in output della Tabella Interna è necessario il ciclo LOOP. La SELECT non richiede l'istruzione ENDSELECT.

```
TABLES: MARA.
```

```
DATA: COPIA_MARA LIKE MARA OCCURS 0 WITH HEADER LINE.
```

```
SELECT * FROM MARA INTO CORRESPONDING FIELDS OF TABLE COPIA_MARA WHERE MTART = 'HAWA'.
LOOP AT COPIA_MARA.
  WRITE: / COPIA_MARA-MATNR, COPIA_MARA-MTART.
ENDLOOP.
```



350-130	HAWA
350-140	HAWA
350-150	HAWA
350-160	HAWA
350-170	HAWA
350-180	HAWA
350-190	HAWA
1300-900	HAWA
1400-100	HAWA
1400-200	HAWA
1400-300	HAWA
1400-310	HAWA
1400-315	HAWA
1400-400	HAWA
1400-715	HAWA
1417	HAWA
1427	HAWA
1400-740	HAWA
1400-750	HAWA
400-510	HAWA
400-511	HAWA

L'istruzione "SELECT"

SELECT E TRASPORTO DATI IN CODA A TABELLA INTERNA CON CAMPI UGUALI

Questa SELECT permette di estrarre tutti i contenuti o parte dei contenuti da una tabella per inserirli direttamente in una tabella interna in coda ai contenuti già presenti. La tabella interna deve avere gli stessi identici campi della tabella sorgente. L'uguaglianza dei campi non è riferita al numero, ma al nome campo. Il requisito fondamentale per eseguire correttamente la SELECT è che la Tabella Interna deve essere dichiarata con la dicitura "OCCURS n" ed è a discrezione del programmatore se dichiararla anche con l'istruzione "WITH HEADER LINE". Notare che per la scrittura in output della Tabella Interna è necessario il ciclo LOOP. La SELECT non richiede l'istruzione ENDSELECT.

```
TABLES: MARA.
```

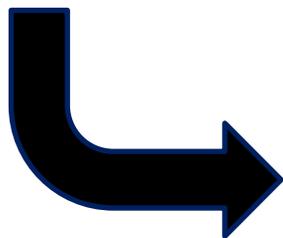
```
DATA: COPIA_MARA LIKE MARA OCCURS 0 WITH HEADER LINE.
```

```
SELECT * FROM MARA APPENDING CORRESPONDING FIELDS OF TABLE COPIA_MARA
WHERE MTART = 'HAWA'.
```

```
LOOP AT COPIA_MARA.
```

```
WRITE: / COPIA_MARA-MATNR, COPIA_MARA-MTART.
```

```
ENDLOOP.
```



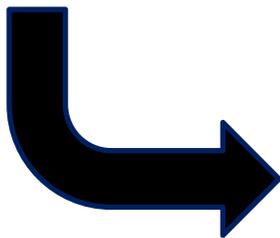
1400-700	HAWA
400-510	HAWA
400-511	HAWA
400-512	HAWA
400-513	HAWA
400-520	HAWA
400-521	HAWA
400-522	HAWA
400-523	HAWA
1408	HAWA
50-5000	HAWA
100-250	HAWA
100-251	HAWA
100-260	HAWA
100-261	HAWA

L'istruzione "SELECT"

SELECT IN UN RANGE DI VALORI

Questa SELECT permette di estrarre tutti i contenuti inclusi in un range definito nella condizione "WHERE". L'istruzione adatta per questo tipo di SELECT è "BETWEEN". Qui l'istruzione "ENDSELECT" è obbligatoria

```
SELECT * FROM MARA WHERE ERSDA BETWEEN '20130101' AND '20130228'.
WRITE: / MARA-MATNR, MARA-MTART, MARA-ERSDA.
ENDSELECT.
```



7866	ROH	15.02.2013
7868	ROH	15.02.2013
7869	ROH	15.02.2013
7870	ROH	15.02.2013
7871	ROH	15.02.2013
7872	ROH	15.02.2013
7877	ROH	15.02.2013
7876	ROH	15.02.2013
7875	ROH	15.02.2013
7879	ROH	15.02.2013
7878	ROH	15.02.2013
7882	ROH	15.02.2013
8124	HAWA	25.02.2013
8125	HAWA	25.02.2013
8126	HAWA	25.02.2013
8127	HAWA	25.02.2013
8128	HAWA	25.02.2013
8134	FERT	25.02.2013
8135	FERT	25.02.2013
8137	FERT	25.02.2013
8138	HAWA	25.02.2013
8141	HAWA	25.02.2013

L'istruzione "SELECT"

SELECT VALORI DA UNA PARTE DI CHIAVE

Questa SELECT permette di estrarre tutti i contenuti che iniziano o terminano con la dicitura specificata nella condizione WHERE.

L'istruzione adatta per questo tipo di SELECT è "LIKE" e la porzione di chiave deve essere delimitata dal simbolo percentuale "%" all'inizio o fine a seconda delle esigenze. In base, quindi, alla posizione in cui è messa la percentuale, il Sistema dà in output tutti i record che iniziano, contengono o terminano per la porzione di chiave inserita.

```
SELECT * FROM MARA WHERE MATNR LIKE '%5'.
WRITE: / MARA-MATNR.
ENDSELECT.
```



```
645
955
1015
1315
```

```
SELECT * FROM MARA WHERE MATNR LIKE '5%'.
WRITE: / MARA-MATNR.
ENDSELECT.
```



```
50-5000
500-100
500-101
500-102
```

```
SELECT * FROM MARA WHERE MATNR LIKE '5%0'.
WRITE: / MARA-MATNR.
ENDSELECT.
```



```
50-5000
500-100
500-110
500-120
```


L'istruzione "SELECT"

L'uso della SELECT JOIN spesso risulta dispendioso a livello di prestazioni SAP. E' infatti consigliabile limitare il più possibile l'uso della JOIN sostituendola con l'uso delle SELECT ANNIDATE. Il risultato è lo stesso, ma a livello di performance questa risulta più veloce nell'elaborazione.

```
SELECT X~MATNR X~MTART Y~MAKTX
  INTO (CODICE_MATERIALE, TIPO_MATERIALE, DESCRIZIONE_MATERIALE)
  FROM MARA AS X JOIN MAKT AS Y
  ON X~MATNR = Y~MATNR.
  WRITE: / CODICE_MATERIALE, TIPO_MATERIALE, DESCRIZIONE_MATERIALE.
ENDSELECT.
```



1300-1800	FERT MSI 1200 cc Custom Racing Motorcycle
1300-1800	FERT MSI 1200 cc Custom Racing Motorcycle
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY Fahrwerk
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY Châssis
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD グラッドボーイシャーシ
1300-200	HALB HD GLAD BOY chassi
1300-200	HALB HD GLAD BOY Шасси
1300-200	HALB HD GLAD BOY Chasis

```
SELECT MATNR MTART
  INTO (CODICE_MATERIALE, TIPO_MATERIALE) FROM MARA.
  SELECT MAKTX
  INTO (DESCRIZIONE_MATERIALE) FROM MAKT
  WHERE MATNR = CODICE_MATERIALE.
  WRITE: / CODICE_MATERIALE, TIPO_MATERIALE, DESCRIZIONE_MATERIALE.
ENDSELECT.
```



```
ENDSELECT.
```

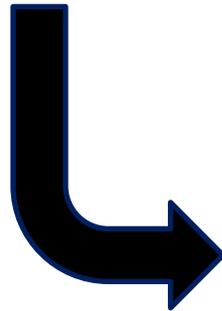
1300-1800	FERT MSI 1200 cc Custom Racing Motorcycle
1300-1800	FERT MSI 1200 cc Custom Racing Motorcycle
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY Fahrwerk
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD GLAD BOY Châssis
1300-200	HALB HD GLAD BOY chassis
1300-200	HALB HD グラッドボーイシャーシ
1300-200	HALB HD GLAD BOY chassi
1300-200	HALB HD GLAD BOY Шасси
1300-200	HALB HD GLAD BOY Chasis

L'istruzione "SELECT"

ORDINAMENTO RISULTATO DELLA SELECT

Ogni risultato di una SELECT può essere ordinato secondo criteri decisi dal programmatore. L'istruzione adatta a ciò è la "ORDER BY". Tutti i campi successivi all'istruzione sono interpretati da SAP in modo sequenziale.

```
SELECT * FROM MARA ORDER BY MTART.
WRITE: / MARA-MATNR, MARA-MTART.
ENDSELECT.
```



3308	ABF
IDES_WA_0001	ABF
IDES_WA_0002	ABF
IDES_WA_0003	ABF
P182_WST	ABF
P182_WST_PP	ABF
P182_WST_RQ	ABF
1117	AEM
3406	AEM
MSA-3100	AEM

L'Ordinamento può essere Crescente o Decrescente. Per impostare ciò, basta aggiungere la dicitura "ASCENDING" o "DESCENDING" subito dopo il campo oggetto dell'ordinamento.

Se, invece, si vuole ordinare automaticamente per Chiave Primaria, basta inserire l'istruzione "ORDER BY PRIMARY KEY".

```
SELECT * FROM MARA ORDER BY MATNR ASCENDING.
ENDSELECT.
```

```
SELECT * FROM MARA ORDER BY MATNR DESCENDING.
ENDSELECT.
```

```
SELECT * FROM MARA ORDER BY PRIMARY KEY.
ENDSELECT.
```

L'istruzione "SELECT"

RAGGRUPPAMENTO RISULTATO DELLA SELECT

Ogni risultato di una SELECT può essere raggruppato secondo criteri decisi dal programmatore. L'istruzione adatta a ciò è la "GROUP BY". Tutti i campi successivi all'istruzione sono interpretati da SAP in modo sequenziale. Questa istruzione richiede l'uso di variabili di appoggio.

```
SELECT MATNR MTART INTO (CODICE_MATERIALE, TIPO_MATERIALE)
  FROM MARA GROUP BY MTART MATNR.
  WRITE: / CODICE_MATERIALE, TIPO_MATERIALE.
ENDSELECT.
```



1300-300	HALB
1300-310	ROH
1300-312	ROH
1300-330	HALB
350-170	HAWA
ACC_FEES	ZSER
ACQ_WATERPUMP	FERT
ACT-BCD-03	ROH
ACT-BCD-04	ROH
1300-361	ROH
1300-372	ROH
1300-400	HALB
1300-401	ROH
1300-410	ROH
1300-420	ROH
40-110R	ROH

DEMO

Si-Soft
Informatica srl

Questo documento è di proprietà
della Si-Soft Informatica e tutti i
diritti sono riservati.