

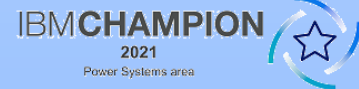
Date e ore *definizione e conversione*



Marco Riva



www.markonetools.it



Ultimo aggiornamento: 24/10/2021

Introduzione a date e ore

- Il tipo dati data/ora/timestamp è stato introdotto nella **V2R3** (1994), ma ancora oggi purtroppo sembra un tipo dati sconosciuto e “pericoloso” da usare
- Esistono numerosi metodi diversi per operare con le date: codici operativi e funzioni RPG, funzioni SQL, API, comandi di sistemi operativo: **almeno 98 metodi!**



2

Tutto (o quasi) su date e ore:

<https://www.markonetools.it/tutto-o-quasi-su-date-e-ore/>

Oggi parliamo di...

- ▶ definizione campi e variabili
- ▶ definizione del formato
- ▶ uso delle costanti data/ora
- ▶ reperimento data/ora correnti
- ▶ conversione tra tipi dati



3

Riepilogo definizioni date/ore

Description	RPG IV			DDS		SQL		Value	Format
	Type ¹	Length	Keyword	Type	Keyword	Type	Length		
DATE/TIME ²	Date	D <i>date</i>	6 8 10	datfmt(*JUL) ³ datfmt(*YMD) datfmt(*DMY) datfmt(*MDY) datfmt(*ISO) ⁴ datfmt(*EUR) datfmt(*USA) datfmt(*JIS)	L	DATFMT DATSEP	DATE ⁵	40/001 to 39/365, def. 40/001 40/01/01 to 39/12/31 def. 40/01/01 01/01/40 to 31/12/39 def. 01/01/40 01/01/40 to 12/31/39 def. 01/01/40 01/01/0001 to 31/12/9999 def. 01/01/0001 01.01.0001 to 31.12.9999 def. 01.01.0001 01/01/0001 to 12/31/9999 def. 01/01/0001 0001-01-01 to 9999-12-31 def. 0001-01-01	2 digits yy/ddd 2 digits yy/mm/dd 2 digits dd/mm/yy 2 digits mm/dd/yy 4 digits yyyy-mm-dd 4 digits dd.mm.yyyy 4 digits mm/dd/yyyy 4 digits yyyy-mm-dd
	Time	T <i>time</i>	8	timfmt(*HMS) ⁶ timfmt(*ISO) timfmt(*USA) timfmt(*EUR) timfmt(*JIS)	T	TIMFMT TIMSEP	TIME ⁷	00:00:00 to 24:00:00, def. 00:00:00 00.00.00 to 24.00.00, def. 00:00:00 00:00 AM to 12:00 AM, def. 00:00 AM 00.00.00 to 24.00.00, def. 00:00:00 00.00.00 to 24.00.00, def. 00:00:00	hh:mm:ss hh.mm.ss hh:mm AM or hh:mm PM hh.mm.ss hh:mm:ss
	Timestamp	Z <i>timestamp</i>	26-32 ⁸	*ISO	Z		TIMESTAMP ⁹	0001-01-01-00.00.00.000000 to 9999-12-31-24.00.00.000000 def. 0001-01-01-00.00.00.000000	YYYY-MM-DD- hh.mm.ss.mmmmmm ¹⁰

- ¹ In *brown italic font* the free-form syntax
- ² A 0 after format Indicates NO separator (ex. *ISO0)
- ³ Valid separator for 2 digits format: / - . , &
- ⁴ The default format for date, time e timestamp fields is *ISO
- ⁵ The internal representation is a string of 4 bytes that contains an integer (scaliger number)
- ⁶ Valid separator: . , &
- ⁷ The internal representation is a string of 3 bytes that contains two packed decimal digits.
- ⁸ From 7.2 the lenght can be until 32
- ⁹ The internal representation is a string of 3 of between 7 and 13 bytes
- ¹⁰ The fractional part is facultative. The length could be from 1 to 12. The default is 6.

IBM i 7.3



Poster tipi dati:

<https://www.markonetools.it/ce-ne-di-tutti-i-tipi/>

Definizione in SQL

- ▶ Esistono 3 tipi di dati per rappresentare i valori di data/ora: **date**, **time** e **timestamp (n)**, con *n* che rappresenta la parte frazionale dei secondi (da 0 a 12).
- ▶ La rappresentazione interna nel DB2 è una stringa
 - ▶ data (date) 4 bytes, il cosiddetto "scaliger number"
 - ▶ ora (time) 3 bytes
 - ▶ timestamp tra 7 e 13 bytes. I primi 4 bytes rappresentano la data, i successivi 3 l'ora e gli ultimi (da 0 a 6) la parte frazionale dei secondi.

Valore default:

Se nella definizione del campo si aggiunge la keyword **not null default** senza specificare nessun valore verrà assunto la data/ora/timestamp corrente al momento dell'operazione di insert.



Definizione con DDS

- ▶ Nel linguaggio DDS (per definire file fisici, video o printer) alla posizione 35 della specifica A si indica il tipo dati senza indicare la lunghezza (pos. 30-34)
 - ▶ L: data
 - ▶ T: ora
 - ▶ Z: timestamp
- ▶ Il formato si specifica con le keyword `DATEFMT` o `TIMEFMT`



Definizione in CLP

- ▶ I tipi data/ora/timestamp per le variabili in un programma CL non sono disponibili



7

Definizione in RPG

- ▶ data (`d - date`)
- ▶ ora (`t - time`)
- ▶ timestamp (`z - timestamp`)
- ▶ Non è necessario specificare la lunghezza del campo a parte il timestamp per il quale si può scegliere la precisione della frazione di secondo
- ▶ Il valore di default (nonché il valore minimo `*LOVAL`) è 1 gennaio 0001 ora 00.00.00.
Il valore massimo (`*HIVAL`) è 9999-12-31-24.00.00.
Ma bisogna porre attenzione ai formati!



Formati data/ora

- ▶ dal **formato** dipende come viene “esposto” il dato e non come viene memorizzato internamente dal DB2 o nella variabile RPG
- ▶ **dal formato** però **dipendono** anche le caratteristiche del contenuto ovvero **i valori di default e i limiti di valore minimo e massimo**
- ▶ esistono 57 combinazioni di formati data e 14 per le ore (!)
- ▶ in RPG la definizione del formato dipende (in ordine di priorità) da:
 1. specifiche di definizione delle variabili (D).
 2. specifiche di definizione file (F)
 3. keyword DATFMT, TIMFMT nella più recente direttiva /SET
 4. specifiche di controllo (H)
 5. *iso (default) -> aaaa-mm-gg



9

Riepilogo formati data/ora

Nome	Descrizione	Formato	Separatore ¹	Lungh.	Esempio	Minimo ² (*LOVAL)	Massimo (*HIVAL)	Default	Range anni
DATA									
*MDY	USA: Month/Day/Year	mm/dd/yy	/-,&	8	01/15/94	01/01/40	12/31/39	01/01/40	1940-2039
*DMY	European: Day/Month/Year	dd/mm/yy	/-,&	8	15/01/94	01/01/40	31/12/39	01/01/40	1940-2039
*YMD	Year/Month/Day	yy/mm/dd	/-,&	8	94/01/15	40/01/01	39/12/31	40/01/01	1940-2039
*JUL	Julian	yy/ddd	/-,&	6	94/015	40/001	39/365	40/001	1940-2039
*ISO	International Standards Organization	yyyy-mm-dd	-	10	1994-01-15	0001-01-01	9999-12-31	0001-01-01	0001-9999
*USA	IBM USA std.	mm/dd/yyyy	/	10	01/15/1994	01/01/0001	12/31/9999	01/01/0001	0001-9999
*EUR	IBM European std.	dd.mm.yyyy	.	10	15.01.1994	01.01.0001	31.12.9999	01.01.0001	0001-9999
*JIS ³	Japanese Industrial Standard	yyyy-mm-dd	-	10	1994-01-15	0001-01-01	9999-12-31	0001-01-01	0001-9999
*CYMD ⁴	Century ⁵ /Year/Month/Day	cy/mm/dd	/-,&	9	094/01/15	000/01/01 ⁶	999/12/31	000/01/01	1900-2899
*CMDY	Century/Month/Day/Year	cmm/dd/yy	/-,&	9	001/15/94	001/01/00	912/31/99	001/01/00	1900-2899
*CDMY	Century/Day/Month/Year	cdd/mm/yy	/-,&	9	015/01/94	001/01/00	931/12/99	001/01/00	1900-2899
*LONGJUL	Long Julian	yyyy/ddd	/-,&	8	1994/015	0001/001	9999/365	0001/001	0001-9999
*JOBRUN ⁷		DATFMT	DATSEP						
ORA									
*HMS	Ore/Minuti/Secondi	hh:mm:ss	.,&	8	14:00:00	00:00:00	24:00:00	00:00:00	
*ISO	International Standards Organization	hh.mm.ss	.	8	14.00.00	00.00.00	24.00.00	00.00.00	
*USA	IBM USA std.	hh:mm AM or hh:mm PM	:	8	02:00 PM	00:00 AM	12:00 AM	00:00 AM	
*EUR	IBM European std.	hh.mm.ss	.	8	14.00.00	00.00.00	24.00.00	00.00.00	
*JIS	Japanese Industrial Standard	hh:mm:ss	:	8	14:00:00	00:00:00	24:00:00	00:00:00	
*JOBRUN ⁷			TIMSEP						
TIMESTAMP									
		yyyy-mm-dd-hh.mm.ss.mmmmmm		26-32	1994-01-15-14.00.00.000000	0001-01-01-00.00.00	9999-12-31-24.00.00	0001-01-01-00.00.00	

in RPG i formati con suffisso 0 non prevedono l'uso dei separatori (da V3R7).
P.es. *iso0,
*ymd0...



¹ Separatore & indica che verrà inserito un blank.

Se il carattere separatore è diverso da quello di default del formato viene specificato sulla destra del formato, p.es. *YMD-

Se a destra del formato si aggiunge 0 significa che il separatore verrà soppresso, p.es. *ISO0

² Per le ore il valore minimo e massimo hanno il medesimo significato, ovvero la mezzanotte

³ Di fatto il formato *JIS è identico a *ISO

⁴ I formati che prevedono il secolo e il formato *LONGJUL non sono utilizzabili per memorizzare campi di tipo data in una tabella. Sono disponibili solo per consentire a chi già memorizzava le date in campi numerici con questo formato di convertirli in campi di tipo date

⁵ '0' per le date fino al 1999, '1' per le date dal 2000, ..., '9' dal 2800

⁶ Il range per i formati che prevedono il secolo è tra 1 gen 1900 e 31 dic 2899

⁷ Può essere specificato in fattore 1 di un'operazione di MOVE, MOVE1, TEST quando un campo carattere o numerico il cui formato corrisponde agli attributi correnti del job (DATFMT, DATSEP, TIMSEP) deve essere trasferito in un campo di tipo date/time

IBM i 7.3 MK1

Definizione formato in RPG

- ▶ keyword **datfmt**
specifica il formato interno per le costanti e variabili di tipo data. In assenza di questa keyword il valore di default è `*iso`
- ▶ keyword **datedit**
specifica il formato dei campi **numerici** con il codice editazione Y. Il separatore di default è `/`. I formati possono essere `*dmy`, `*mdy`, `*ymd`
→ Nella specifica H è **IMPORTANTE** quando si usano i campi predefiniti per la data del job (`*date`, `update...`)
- ▶ keyword **timfmt**
specifica il formato interno per le costanti e variabili di tipo ora. In assenza di questa keyword il valore di default è `*iso`



Formato data/ora e embedded SQL

- ▶ Nei programmi RPG embedded SQL i parametri `DATEFMT`, `DATSEP`, `TIMFMT`, `TIMSEP` possono essere specificati nel comando di creazione (`CRTSQLxxx`) oppure tramite l'istruzione `sql SET OPTION`

```
exec sql  
  set option DATEFMT = '*YMD';
```

ATTENZIONE!!!
`set option` va scritto all'**INIZIO** del sorgente subito dopo le specifiche di dichiarazione o comunque prima di qualsiasi altra istruzione SQL



Scrittura valori costanti in RPG

- ▶ I valori **costanti** (definiti nelle specifiche D o utilizzati nelle specifiche di calcolo) devono avere il formato definito nella specifica H o in mancanza della specifica H devono essere scritti in formato `*iso`
- ▶ **N.B.** Questa regola vale anche se il formato della variabile data è diverso da quello della specifica H.
- ▶ Il valore costante racchiuso tra apici deve essere preceduto da `d` (data), `t` (ora), `z` (timestamp)

```
Outdate=d'2010-12-31';  
OutTime=t'13.59.59';  
OutTimeStamp=z'2010-12-31-13.59.59.000000';
```

la porzione
frazionale è
facoltativa



Data/ora correnti in RPG/1

- ▶ E' possibile inizializzare una variabile di tipo data alla data di sistema con la keyword `inz(*sys)` oppure alla data del job con `inz(*job)`
- ▶ Un campo di tipo time può essere inizializzato all'ora di sistema con `inz(*sys)`
- ▶ Campi predefiniti per data del job:
 - ▶ UDATE: campo numerico di 6
 - ▶ UMONTH: campo numerico di 2
 - ▶ UYEAR: campo numerico di 2
 - ▶ UDAY: campo numerico di 2
 - ▶ C*DATE: campo alfanumerico di 8
 - ▶ ***DATE**: campo numerico di 8
 - ▶ *MONTH: campo numerico di 2
 - ▶ *YEAR: campo numerico di 4
 - ▶ *DAY: campo numerico di 2

il formato è
definito dalla
keyword
datedit nella
specifica H

ATTENZIONE!!!
NON sono la data corrente, ma la data di
avvio del job.
Nella maggior parte dei casi coincide con la
data corrente, ma non sempre



Data/ora correnti in RPG/2

- Un'altra possibilità è utilizzare la program status data structure (PSDS)
 - **data del job** (8 A): posizioni 191-198. Data in cui il job è stato immesso nel sistema. Formato identico a *date
 - **secolo** (2, 0 S): posizioni 199-200. Primi due caratteri dell'anno di 4 cifre.
 - **data del job** (6, 0 S): posizioni 270-275. Data in cui il job è stato immesso nel sistema. Formato identico a udate
 - **data di sistema** (6, 0 S): posizioni 276-281. Data di sistema. Formato identico a udate.
 - **ora di sistema** (6, 0 S): posizioni 282-287. Formato hhmss.

```
dcl-ds MySDS PSDS;  
ps_Program char(10);  
ps_Status char(5);  
...  
end-ds;
```



Data/ora correnti in RPG/3

- ▶ il codice operativo **TIME** restituisce un valore che dipende dalla definizione del campo risultato. In ogni caso si tratta di data/ora di **sistema** (non del job!)

definizione risultato	valore restituito
numerico 6, 0	ora
numerico 12, 0 o 14, 0	ora + data ¹
date	data
time	ora
timestamp	timestamp

ATTENZIONE!!!
solo per formato fisso

(1) data in formato ggmmaa o ggmmaaaa





Data/ora correnti in RPG/4

- Infine, sicuramente il metodo più comodo è utilizzare le BIF `%date()`, `%time()` e `%timestamp()` che restituiscono rispettivamente la data o l'ora di **sistema** (non del job!)
- La funzione `%timestamp(*sys)` restituisce il timestamp corrente di sistema con parte la frazionale dei secondi (solo le prime 6 cifre sono valorizzate). P.es. 2021-10-23-16.19.38.311156000000.
- Se non si desidera la parte frazionale dei secondi si può usare la funzione `%timestamp(*sys : 0)`. P.es. 2021-10-23-16.19.38.000000000000.

prima di 7.3/7.4 solo le prime 3 cifre erano valorizzate

cfr. RPG café:

https://www.ibm.com/support/pages/node/6127845?myns=swgrat&mynp=OCS56gQP&mync=E&cm_sp=swgrat--OCSS6gQP--E



17

Data/ora corrente in DDS

- ▶ Per valorizzare automaticamente un campo di un display/printer file con data/ora correnti si utilizzano le keyword `DATE (*JOB)`, `DATE (*SYS)` o `TIME`
- ▶ la keyword `DATE` prevede anche un secondo parametro per determinare la lunghezza del campo anno
 - ▶ `*Y`: 2 caratteri (default)
 - ▶ `*YY`: 4 caratteri

```
DATE (*JOB *YY)
```



Data/ora correnti in SQL



- ▶ In SQL ci sono i registri speciali:
 - ▶ `CURRENT DATE`: data corrente
 - ▶ `CURRENT TIME`: ora corrente (formato *hh.nn.ss*)
 - ▶ `CURRENT TIMESTAMP (n)`: ora corrente (formato *timestamp*). Con *n* che rappresenta la precisione della parte frazionale dei secondi.
- ▶ Oppure al medesimo risultato si arriva con le funzioni
 - ▶ `CURDATE` → `CURRENT DATE`
 - ▶ `CURTIME` → `CURRENT TIME`
 - ▶ `NOW (n)` → `CURRENT TIMESTAMP (n)`

ATTENZIONE!!!
Gli special registers o le altre funzioni se vengono utilizzati più di una volta nella stessa istruzione sono basati su **una sola lettura del clock di sistema.**



Riepilogo conversione tra tipi dati

da	a	metodo
data/ora/timestamp	alfanumerico	RPG: %char SQL: varchar_format
data/ora/timestamp	numerico	RPG: %dec RPG: %char + %uns SQL: varchar_format + dec SQL: dec
alfanumerico/numerico	data	RPG: %date
alfanumerico/numerico	ora	RPG: %time SQL: time
alfanumerico/numerico	timestamp	RPG: %timestamp SQL: timestamp SQL: timestamp_iso SQL: timestamp_format



Conversione da data legacy a data vera

conversione da **numerico** AAAAMMGG a data

```
DataTrue = %date(DataPacked:*iso);
```

conversione da **alfanumerico** AAAAMMGG a data

```
DataTrue = %date(CharDate:*iso0);
```



Conversione da campo data con RPG

conversione da data a **alfanumerico**

```
dcl-s DataAlfa char(10);  
dcl-s MiaData date inz('2019-08-29');  
  
DataAlfa = %char(MiaData: *iso0); // 20190829
```

conversione da data a **numerico**

```
dcl-s yyymmdd packed(8);  
dcl-s ddmmyy packed(6);  
dcl-s MiaData date(*USA) inz(d'2019-10-20');  
  
yyymmdd = %dec(MiaData : *iso);  
ddmmyy = %dec(MiaData : *dmy);  
yyymmdd = %uns(%char(MiaData : *iso0));
```



Conversione da campo data con SQL

conversione da data a **alfanumerico**

```
-- da data a campo alfanumerico di 8 cifre  
varchar_format(current date), 'YYYYMMDD')
```

conversione da data a **numerico**

```
dec(MyDate)  
dec(MyTime)  
dec(MyTimestamp)  
dec(MyTimestamp, 26, 12)  
dec(date(MyTimestamp))  
dec(time(MyTimestamp), 6, 0)
```



API QWCCVTD

- ▶ L'API QWCCVTD consente di effettuare diverse operazioni di conversione tra tipi dati data/ora e rappresentazioni alfanumeriche di data/ora
- ▶ Consente anche di convertire tra fusi orari diversi (cfr. <https://code400.com/2017/10/29/qwccvtd/>)

Prototipo con parametri obbligatori

```
dcl-pr QWCCVTD extpgm;  
  InpFormat char(10);  
  InpData char(16);  
  OutFormat char(10);  
  OutData char(20);  
  Error char(256);  
end-pr;
```

Data in input

Data in output

Formati data

- *CURRENT: The current system time
- *DTS: System time-stamp
- *JOB: The format given in the DATFMT job attribute
- *SYSVAL: The format given in the QDATFMT system value
- *YMD: YYMMDD (year, month, day) format
- *YYMD: YYYYMMDD (4-digit year, month, day) format
- *MDY: MMDDYY (month, day, year) format
- *MDYY: MMDDYYYY (month, day, 4-digit year) format
- *DMY: DDMMYY (day, month, year) format
- *DMYY: DDMMYYYY (day, month, 4-digit year) format
- *JUL: Julian format (YYDDD (year, day of year))
- *LONGJUL: Long Julian format (YYYYDDD (4-digit year, day of year))



24

<https://www.mcpressonline.com/programming/apis/the-api-corner-how-do-you-want-that-date-and-time>



Tabella conversione date/1

- E' una tabella "calendario" che contiene un record per ogni giorno in uno specifico range di date.
- Ogni record contiene la stessa data rappresentata in diversi modi (data giuliana, data "vera", giorno della settimana, data YYYYMMDD, data DDMMYY, excel...)



25



Tabella conversione date/2

LEGACY_ORDER_HEADER file

ORDER	CUST	ORDDAT	SHF DAT	SHPVIA
1168336	P8311	1022005	7232005	FedEx
7073849	I7472	1022005	9032005	Pick Up
1168336	P8311	1022006	7232006	FedEx
7073849	I7472	1022006	9032006	Pick Up
1168336	P8311	1022007	7232007	FedEx
7073849	I7472	1022007	9032007	Pick Up
1730932	E9747	1042005	10122005	UPS
8211860	P3751	1042005	8242005	Pick Up
1730932	E9747	1042006	10122006	UPS
8211860	P3751	1042006	8242006	Pick Up
1730932	E9747	1042007	10122007	UPS
8211860	P3751	1042007	8242007	Pick Up

Date conversion table

Join (RI constraints, SQL view, synonym join, report join)

DC_DATE	DC_JDE_JULIAN_DATE	DC_MDYY_DEC	DC_MDYY_CHAR	DC_YEAR	DC_DOW	DC_DOY	DC_WOY	DC_QOY	DC...
2006-01-01	106001	1012006	1012006	2006					
2006-01-02	106002	1022006	1022006	2006					
2006-01-03	106003	1032006	1032006	2006					
2006-01-04	106004	1042006	1042006	2006					
2006-01-05	106005	1052006	1052006	2006					
2006-01-06	106006	1062006	1062006	2006					
2006-01-07	106007	1072006	1072006	2006					
2006-01-08	106008	1082006	1082006	2006					
2006-01-09	106009	1092006	1092006	2006					
2006-01-10	106010	1102006	1102006	2006					
2006-01-11	106011	1112006	1112006	2006					
2006-01-12	106012	1122006	1122006	2006					

La logica di utilizzo della tabella di conversione è metterla in join con la tabella contenente la data "legacy" tramite il campo più appropriato (p.es. per una data legacy memorizzata in un campo packed 8, 0 in formato YMMDD si utilizza DC_YMDD_DEC) e quindi utilizzare nel resto della query il campo data "vero"

DC_DATE.

DC_DATE
123 DC_JDE_JULIAN_DATE
123 DC_MDYY_DEC
123 DC_MDYY_ZONED
ABC DC_MDYY_CHAR
123 DC_YMDD_DEC
123 DC_YMDD_ZONED
ABC DC_YMDD_CHAR
ABC DC_CC_CHAR
ABC DC_YY_CHAR
ABC DC_MM_CHAR
ABC DC_DD_CHAR
123 DC_YEAR
123 DC_DOW
123 DC_DOY
123 DC_WOY
123 DC_QOY
123 DC_CC
123 DC_YY
123 DC_YYYY
123 DC_MM
123 DC_DD
ABC DC_DAY_NAME
ABC DC_QUARTER_NAME
ABC DC_WEEKEND
ABC DC_SEASON
123 DC_FISCAL_YEAR
123 DC_FISCAL_QUARTER
ABC DC_MONTH_NAME
ABC DC_MONTH_ABRV
123 DC_JULIAN
123 DC_CYYMMDD
123 DC_EXCEL_DATE
DC_WEEK_STARTING_DATE
DC_WEEK_ENDING_DATE
DC_WEEK_ENDING_SUN_DATE
DC_SAME_DAY_LAST_YEAR
ABC DC_CURRENT_DAY
ABC DC_CURRENT_WEEK
ABC DC_CURRENT_MONTH
ABC DC_CURRENT_QUARTER
ABC DC_CURRENT_YEAR
ABC DC_CURRENT_DAY_LAST_YEAR
ABC DC_CURRENT_WEEK_LAST_YEAR
ABC DC_CURRENT_MONTH_LAST_YEAR
ABC DC_CURRENT_QUARTER_LAST_YEAR
ABC DC_CURRENT_YEAR_LAST_YEAR

Tabella conversione date/3

- Questa tabella è descritta redbook *SG24-7214-01 "Getting Started with DB2 Web Query for i" (appendix B p. 542-552)* ormai di difficile reperibilità sul web
- In allegato al pdf di questa presentazione trovate lo script per generarla (*Date_Conversion_table.sql*) corretto ed adattato alla versione italiana

Script di generazione:

- 1) creazione tabella `Date_Conversion`
- 2) creazione indici di uso più comune (`Date_Conversion_01I`, `Date_Conversion_02I`, `Date_Conversion_03I`)
- 3) creazione procedura `LOAD_DATE_CONVERSION_TABLE` per popolare la tabella
- 4) creazione procedura `UPDATE_DATE_CONVERSION_TABLE` per aggiornare alcuni campi variabili quotidianamente
- 5) esecuzione della procedura di popolamento della tabella (da eseguire una tantum)
- 6) *facoltativo*: schedulare giornalmente l'esecuzione della procedura di aggiornamento



Riferimenti



➤ E-mail aziendale: mriva@sirio-is.it



➤ Blog: www.markonetools.it



➤ E-mail blog: info@markonetools.it



➤ LinkedIn: www.linkedin.com/in/marcoriva-mk1



➤ Twitter: [@MarcoRiva73](https://twitter.com/MarcoRiva73)



➤ Facebook: <https://www.facebook.com/markonetools/>



➤ YouTube: <https://www.youtube.com/channel/UCb47YJQJCzU-5x4nnGzDu-w>

Power coffee - MK1



28