

## Debug funzioni e procedure SQL con RDi



Marco Riva



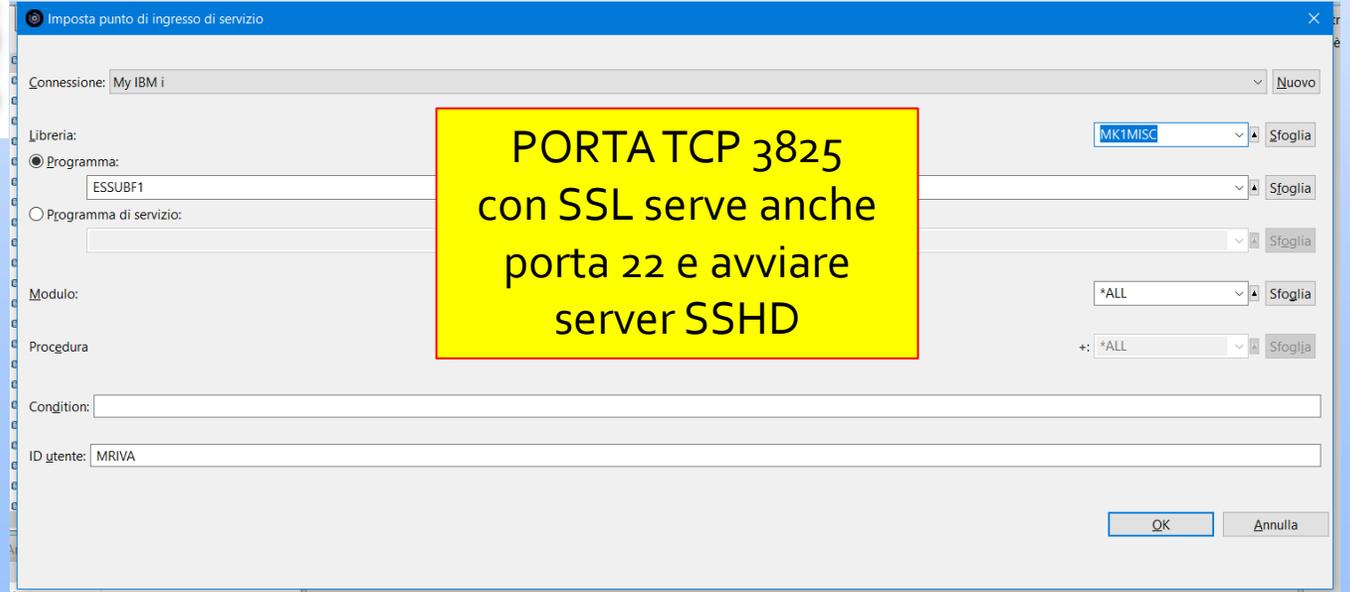
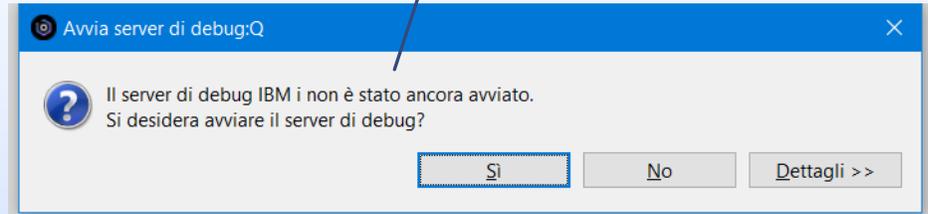
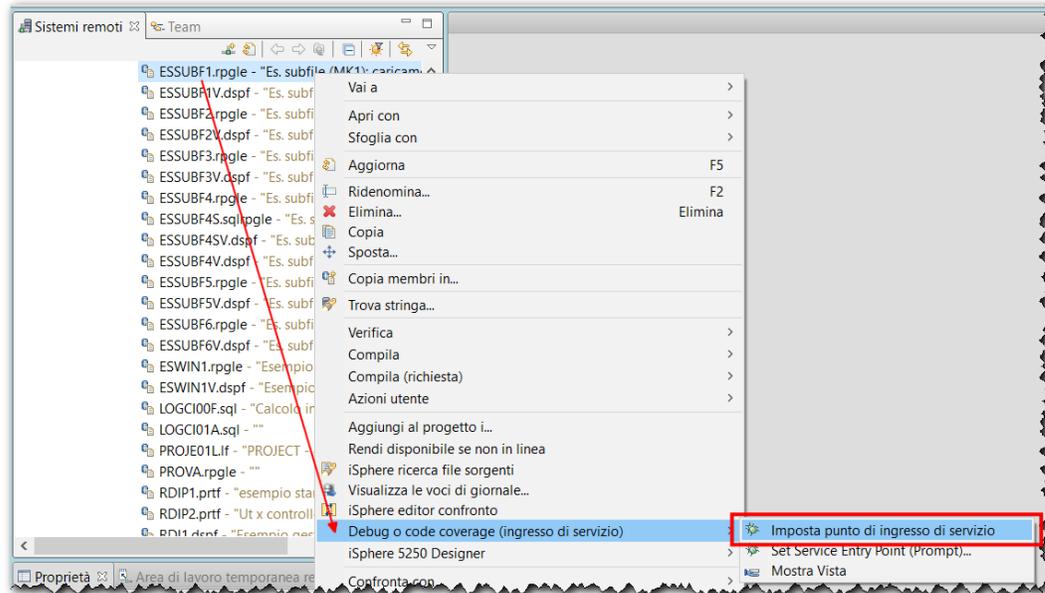
[www.markonetools.it](http://www.markonetools.it)



Ultimo aggiornamento: 24/03/2023

# Debug RDi – punti ingressi servizio

una volta avviato il server di debug è disponibile per tutti



Power coffee - MK1

# Cosa è una funzione/procedura SQL?

- ▶ Quando si crea una funzione o una procedura in linguaggio SQL  
`create function...`  
`language SQL...`  
viene tradotta in embedded SQL C e  
compilata generando un programma di  
servizio di tipo **\*SRVPGM-CLE** (per le function) o di tipo **\*PGM-CLE** (per le procedure)

## Info oggetto

Nome: CURDATLEG

Libreria: MR731

Tipo: \*SRVPGM

Attributo: CLE

Testo: SQL FUNCTION CURDATELEGACY



3

## Prepararsi al debug

- ▶ aggiungere  
`set option dbgview=*source`  
prima del corpo della routine
- ▶ viene creato il membro sorgente nel file QSQDSRC nella stessa libreria dove viene creata la funzione/procedura.  
Il membro contiene le righe di codice che verranno mostrate durante il debug

```
-- [11] funzione scalare conversione data leggibile
create or replace function CvtDate
(pData decimal(8, 0) default 99991231)
returns date
language sql
specific CvtDateP
deterministic -- il risultato è sempre lo stesso
modifies sql data

set option dbgview = *source

begin
-- se data non valida restituisce il valore 99991231
-- se data non valida restituisce il valore 99991231
```

# Attivare il debug

- da RDi si può impostare un punti di ingresso di servizio sul programma di servizio generato da `create function` o sul programma generato da `create procedure`

Modifica punto di ingresso di servizio

Connessione:  Nuovo

Libreria:  Sfogli

Programma:  Sfogli

Programma di servizio:  Sfogli

Modulo:  Sfogli

Procedura:  Sfogli

Condition:

ID utente:

OK Annulla



5

# Debug in corso

- ▶ ogni istruzione SQL in realtà è tradotta in diverse righe di codice C, quindi il debug passo-passo non è comodo, si consiglia di utilizzare i punti di interruzione

- ▶ nel debug le variabili da monitorare hanno nome preceduto dal prefisso %%

The screenshot displays the IBM i debug environment. The top window shows the 'Variabili' (Variables) pane with two entries: '%%PDATA = 99991231.' and 'PDATA = SPP:C1B08A775B001566'. A red arrow points to the first entry. Below this, the 'CVTDATEP.MBR' source code is visible. The code is as follows:

```
Riga 6      Colonna 1      Sostituisci      Sfogliare
-----1-----2-----3-----4-----5-----6-----7-----+
000001 CREATE OR REPLACE FUNCTION CVTDATE ( PDATA DECIMAL ( 8 , 0 ) DEFAULT 99991231
000002      -- il risultato è sempre lo stesso a parità di parametri di input MODIF
000003 BEGIN
000004 DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
000005 RETURN DATE ( '0001-01-01' );
000006 RETURN DATE ( DIGITS ( PDATA ) CONCAT '000000' );
000007 END;
```

A red arrow points to the line 'RETURN DATE ( DIGITS ( PDATA ) CONCAT '000000' );' in the source code.

# Prepararsi al debug del codice C

- ▶ Se si aggiunge  
`set option dgbview=*list`  
prima del corpo della routine
- ▶ Non viene creato il membro sorgente nel file QSQDSRC
- ▶ Durante il debug verrà mostrato il codice del programma in C generato dal DB2



7

# Debug in corso del codice C

► I parametri sono sottocampi di una struttura dati con nome identico alla funzione (nome SQL)

► Le variabili sono sottocampi della struttura dati SQLP\_L2

The screenshot shows a debugger interface with two main panels. The top panel displays the 'Debug' window with a tree view showing the current thread and its stack frames. The bottom panel shows the 'CNTEMP.LISTING' window with a code listing. The code listing is as follows:

Riga	Colonna	Inserisci	Sfogli
193		/**/\$\$	
194		EXEC SQL END DECLARE SECTION	
195		\$\$\$**/	
196	3	SQLCODE = 0;	
197	4	memset(SQLSTATE, '0', 5);	
198	5	SQL_JobCCSID = SQLRetrieveJobCCSIDOpt();	
199	6	sqlcontrol = 0x00;	
200	7	SQL_RETURN.SQLP_I2 = SQLP_NULLIND;	

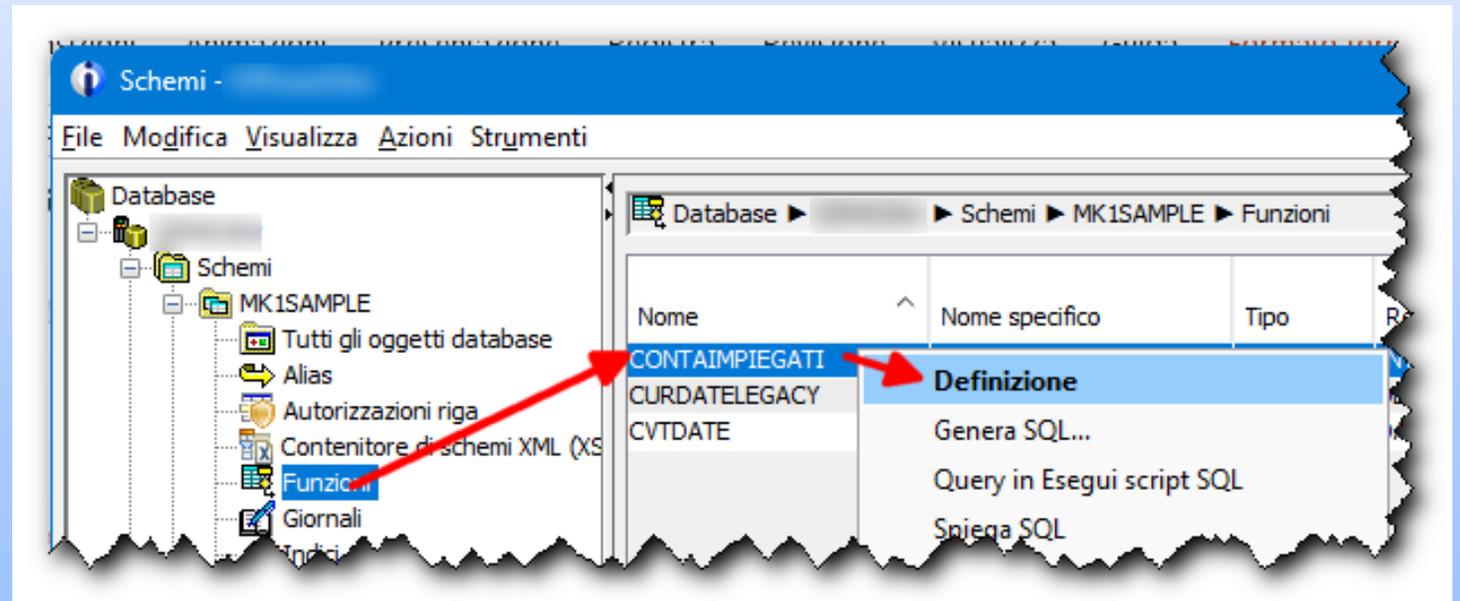
The 'Variabili' window on the right shows the following variables:

- CONTAIMPIEGATI
- PDEPT =
- SQLP\_I1 = 0
- SQLP\_L2
- NUMREC = 0

Red arrows point from the 'CONTAIMPIEGATI' and 'SQLP\_L2' variables to the corresponding code lines in the listing.

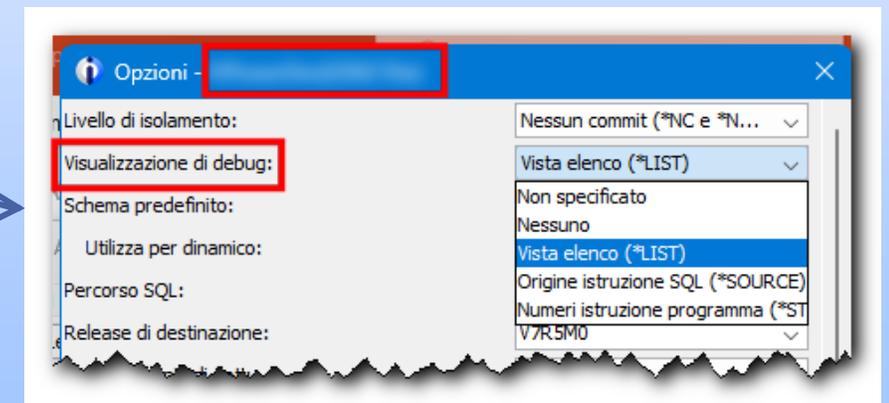
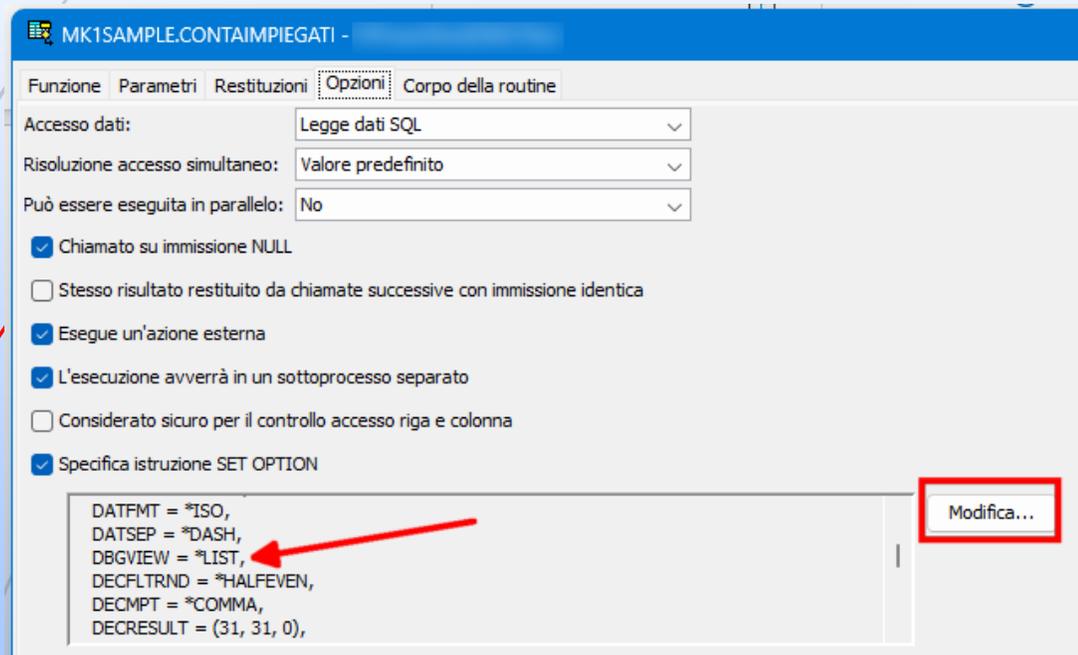
# Modificare impostazione debug/1

- ▶ Per modificare l'impostazione del debug non è necessario ricreare la funzione/procedura
- ▶ Dalla gestione schemi di ACS si può aprire la definizione



# Modificare impostazione debug/2

- Nella scheda Opzioni cliccare sul pulsante Modifica della sezione SET OPTION



10

# Riferimenti



➤ E-mail aziendale: [marco.riva@ivolution.it](mailto:marco.riva@ivolution.it)



➤ Blog: [www.markonetools.it](http://www.markonetools.it)



➤ E-mail blog: [info@markonetools.it](mailto:info@markonetools.it)



➤ LinkedIn: [www.linkedin.com/in/marcoriva-mk1](http://www.linkedin.com/in/marcoriva-mk1)



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



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



➤ YouTube: [youtube.com/@markonetools](https://youtube.com/@markonetools)

Power coffee - MK1



11