

I servizi SQL di IBM i *analisi dei programmi embedded SQL*



Marco Riva



www.markonetools.it



Ultimo aggiornamento: 20/07/2022

IBM i services – home page

► <https://www.ibm.com/support/pages/node/1119123>

IBM i Services (SQL)

Content

You are in: [IBM i Technology Updates](#) > [Db2 for i - Technology Updates](#) > IBM i Services (SQL)

IBM i Service	Type of Service	IBM i 7.5	IBM i 7.4	IBM i 7.3
Application Services				
QSYS2.ACTIVATION_GROUP_INFO()	Table function	Base	SF99704 Level 20	SF99703 Level 28
QSYS2.ADD_USER_INDEX_ENTRY()	Procedure	Base	SF99704 Level 20	Not Supported
QSYS2.ADD_USER_INDEX_ENTRY_BINARY()	Procedure	Base	SF99704 Level 20	Not Supported
QSYS2.BINDING_DIRECTORY_INFO	View	Base	SF99704 Level 20	SF99703 Level 28
QSYS2.BOUND_MEMBER_INFO	View	Base	SF99704	SF99703 Level 28

Power coffee - MK1



2

Reperire informazioni sui programmi

► PROGRAM INFO

restituisce informazioni sugli oggetti programma

► BOUND MODULE INFO

restituisce informazioni sugli oggetti modulo combinati in programmi o programmi di servizio ILE



3

Reperire informazioni sulle istruzioni SQL

► SYSPROGRAMSTAT

restituisce un record per ogni istruzione SQL contenuta in un programma, modulo o programma di servizio

► SYSPROGRAMTMTSTAT

restituisce un record per ogni istruzione SQL contenuta in un programma, modulo o programma di servizio

► PARSE STATEMENT

elenca tutti gli oggetti e nomi di colonna (campo) utilizzati in un'istruzione SQL



Proprietà generali di un programma/1

```
select PROGRAM_NAME "Nome", OBJECT_TYPE "Tipo",  
trim(BOUND_MODULE_LIBRARY) concat '/' concat trim(BOUND_MODULE) "Modulo",  
MODULE_ATTRIBUTE "Tipo mod.", timestamp(MODULE_CREATE_TIMESTAMP, 0)  
"Data/ora creaz.mod.", trim(SOURCE_FILE_LIBRARY) concat '/' concat  
trim(SOURCE_FILE) concat '(' concat trim(SOURCE_FILE_MEMBER) concat ')'  
"Sorgente", timestamp(SOURCE_CHANGE_TIMESTAMP, 0) "Data/ora src",  
DEBUG_DATA "Dati debug", OPTIMIZATION_LEVEL "Liv.ottimizzazione",  
TARGET_RELEASE "Rel.min.", NUMBER_PROCEDURES "Num.procedure",  
SQL_STATEMENT_COUNT "Num.istruzioni SQL", SQL_COMMITMENT_CONTROL  
"Contr.sincronia", SQL_NAMING "SQL Naming", SQL_DATE_FORMAT "SQL formato  
data", SQL_DATE_SEPARATOR "SQL sep.data", SQL_CLOSE_SQL_CURSOR "Chiusura  
cursori", SQL_DELAY_PREPARE "Ritardo prep. SQL din."  
from BOUND_MODULE_INFO  
where PROGRAM_LIBRARY = :Libreria  
and SQL_STATEMENT_COUNT > 0 -- solo programmi embedded SQL  
order by PROGRAM_NAME;
```



Proprietà generali di un programma/2

ABC Nome	ABC Tipo	ABC Modulo	ABC Tipo mod.	Data/ora creaz.mod.	ABC Sorgente	Data/ora src	ABC Dati debug	123 Liv.o
CRTALIAS	*PGM	QTEMP/CRTALIAS	RPGLE	2011-07-25 11:59:09.000	MK1SQL/QSQLSRC(CRTALIAS)	2011-07-25 11:59:07.000	*YES	
CURS01	*PGM	QTEMP/CURS01	RPGLE	2021-03-17 10:37:21.000	MK1SQL/QCORSO(CURS01)	2021-03-17 10:22:13.000	*YES	
CURS01B	*PGM	QTEMP/CURS01B	RPGLE	2011-10-24 14:54:56.000	MK1SQL/QCORSO(CURS01B)	2011-10-24 14:41:05.000	*YES	
CURS01B_B	*PGM	QTEMP/CURS01B_B	RPGLE	2011-10-25 16:10:36.000	MK1SQL/QCORSO(CURS01B_B)	2011-10-25 16:10:35.000	*YES	
CURS01C	*PGM	QTEMP/CURS01C	RPGLE	2021-03-17 17:26:52.000	MK1SQL/QCORSO(CURS01C)	2021-03-17 17:26:17.000	*YES	
CURS01D	*PGM	QTEMP/CURS01D	RPGLE	2011-10-26 14:13:30.000	MK1SQL/QCORSO(CURS01D)	2011-10-26 14:13:30.000	*YES	

Liv.ottimizzazione	ABC Rel.min.	123 Num.procedure	123 Num.istruzioni SQL	ABC Contr.sincronia	ABC SQL Naming	ABC SQL formato data	ABC SQL sep.data
10	V5R2M0	4	2	*NONE	*SYS	*DMY	/
10	V7R3M0	5	4	*NONE	*SYS	*DMY	/
10	V5R3M0	4	4	*NONE	*SYS	*DMY	/
10	V5R3M0	4	4	*NONE	*SYS	*DMY	/
10	V7R3M0	5	5	*NONE	*SYS	*DMY	/
10	V5R3M0	4	6	*NONE	*SYS	*DMY	/

ABC Chiusura cursori	ABC Ritardo prep. SQL din.
*ENDACTGRP	*NO
*ENDACTGRP	*NO
*ENDACTGRP	*YES
*ENDACTGRP	*YES
*ENDACTGRP	*NO
*ENDACTGRP	*YES



Proprietà istruzioni SQL embedded/1

```
select PROGRAM_NAME "Programma",  
       (select TEXT_DESCRIPTION from PROGRAM_INFO where  
PROGRAM_LIBRARY = SYSTEM_PROGRAM_SCHEMA and PROGRAM_NAME =  
SYSTEM_PROGRAM_NAME) as "Descrizione",  
MODULE_NAME "Modulo", PROGRAM_TYPE "Tipo", STATEMENT_NUMBER  
"Num.istruzione", EARLIEST_POSSIBLE_RELEASE "Rel.min.",  
SQL_DB2_GROUP_LEVEL "DB2 prereq.", SERVICES_DB2_GROUP_LEVEL  
"DB2 service prereq.", STATEMENT_TEXT "Istruzione", ISOLATION  
"Contr.sincronia", NUMBER_HOST_VARIABLES "Num.variabili host  
(tot)", NUMBER_INPUT_HOST_VARIABLES "Num.variabili host  
(input)"  
from SYSPROGRAMSTMTSTAT  
where PROGRAM_SCHEMA = :Libreria  
order by PROGRAM_NAME, MODULE_NAME, STATEMENT_NUMBER;
```



Proprietà istruzioni SQL embedded/2

ABC Programma	ABC Descrizione	ABC Modulo	ABC Tipo	123 Num.istruzione	ABC Rel.min.	123 DB2 prereq.	123 DB2 service prereq.
CURS07B	SQL statico scrollable cursor sensitive	CURS07B	*PGM	1	V5R3M0	[NULL]	[NULL]
CURS07B	SQL statico scrollable cursor sensitive	CURS07B	*PGM	2	[NULL]	[NULL]	[NULL]
CURS07B	SQL statico scrollable cursor sensitive	CURS07B	*PGM	3	[NULL]	[NULL]	[NULL]
CURS07B	SQL statico scrollable cursor sensitive	CURS07B	*PGM	4	[NULL]	[NULL]	[NULL]
CURS07B	SQL statico scrollable cursor sensitive	CURS07B	*PGM	5	[NULL]	[NULL]	[NULL]
CVTDATE	SQL FUNCTION CVTDATE	CVTDATE	*SRVPGM	3	[NULL]	[NULL]	[NULL]

Istruzione	ABC Contr.sincronia	123 Num.variabili host (tot)	123 Num.variabili host (input)
DECLARE CURS07 SENSITIVE SCROLL CURSOR FOR SELEC	[NULL]	5	5
OPEN CURS07	[NULL]	5	5
FETCH NEXT FROM CURS07 INTO : H ;: H ;: H ;: H	[NULL]	5	0
FETCH BEFORE FROM CURS07	[NULL]	0	0
CLOSE CURS07	[NULL]	0	0
SET : H ;: H = *DATE (LEFT (CHAR (: H ;: H), 4) CONCAT 'NO	[NULL]	8	



Dichiarazioni cursori SQL

```
select PROGRAM_NAME "Programma",  
       (select TEXT_DESCRIPTION from PROGRAM_INFO where PROGRAM_LIBRARY =  
        SYSTEM_PROGRAM_SCHEMA and PROGRAM_NAME = SYSTEM_PROGRAM_NAME) as  
       "Descrizione",  
       MODULE_NAME "Modulo", PROGRAM_TYPE "Tipo", STATEMENT_NUMBER  
       "Num.istruzione", EARLIEST_POSSIBLE_RELEASE "Rel.min.", STATEMENT_TEXT  
       "Istruzione", ISOLATION "Contr.sincronia"  
from SYSPROGRAMSTMTSTAT  
where PROGRAM_SCHEMA = :Libreria  
       and STATEMENT_TEXT like 'DECLARE%CURSOR%'  
order by PROGRAM_NAME, MODULE_NAME, STATEMENT_NUMBER;
```

Programma	Descrizione	Modulo	Tipo	123 Num.istruzione	Rel.min.	Istruzione
CURS01	SQL statico serial cursor	CURS01	*PGM	1	[NULL]	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , C
CURS01B	SQL statico serial cursor (senza DS dien00f)	CURS01B	*PGM	1	[NULL]	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , C
CURS01B_B	SQL statico serial cursor (senza DS dien00f)	CURS01B_B	*PGM	1	[NULL]	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , C
CURS01C	SQL statico serial cursor read-only	CURS01C	*PGM	1	[NULL]	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , C
CURS01D	SQL statico serial cursor con update	CURS01D	*PGM	1	[NULL]	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , C
CURS02	SQL dinamico serial cursor	CURS02	*PGM	1	[NULL]	DECLARE CURS02 CURSOR FOR SELECT CLCCLI , C

Dichiarazioni cursori SQL (dinamici)/1

```
with
  T_DECLARE as
    (select PROGRAM_SCHEMA, PROGRAM_NAME, MODULE_NAME, PROGRAM_TYPE, STATEMENT_NUMBER,
     EARLIEST_POSSIBLE_RELEASE, STATEMENT_TEXT, substr(STATEMENT_TEXT, position('CURSOR FOR',
     STATEMENT_TEXT) + 11) "STM_NAME"
    from SYSPROGRAMSTMTSTAT
    where PROGRAM_SCHEMA = :Libreria and STATEMENT_TEXT like 'DECLARE%CURSOR%'),
  T_PREPARE as
    (select PROGRAM_SCHEMA, PROGRAM_NAME, MODULE_NAME, PROGRAM_TYPE, STATEMENT_NUMBER,
     EARLIEST_POSSIBLE_RELEASE, STATEMENT_TEXT, substr(STATEMENT_TEXT, position('PREPARE', STATEMENT_TEXT) +
     8, position('FROM', STATEMENT_TEXT) - (position('PREPARE', STATEMENT_TEXT) + 8)) "STM_NAME"
    from SYSPROGRAMSTMTSTAT
    where PROGRAM_SCHEMA = :Libreria and STATEMENT_TEXT like 'PREPARE%FROM%')
select D.PROGRAM_NAME "Programma", D.MODULE_NAME "Modulo", D.PROGRAM_TYPE "Tipo",
(select TEXT_DESCRIPTION from PROGRAM_INFO where PROGRAM_LIBRARY = D.PROGRAM_SCHEMA and PROGRAM_NAME =
D.PROGRAM_NAME) as "Descrizione", D.EARLIEST_POSSIBLE_RELEASE "Rel.min.", D.STM_NAME "Nome istruz.",
D.STATEMENT_TEXT "Istruz.declare", P.STATEMENT_TEXT "Istruz.prepare"
from T_DECLARE as D
  inner join T_PREPARE as P
    on (D.PROGRAM_SCHEMA, D.PROGRAM_NAME, D.MODULE_NAME, D.STM_NAME) =
      (P.PROGRAM_SCHEMA, P.PROGRAM_NAME, P.MODULE_NAME, P.STM_NAME)
order by D.PROGRAM_SCHEMA, D.PROGRAM_NAME, D.MODULE_NAME, D.STM_NAME;
```



Dichiarazioni cursori SQL (dinamici)/2

ABC Programma	ABC Modulo	ABC Tipo	ABC Descrizione	ABC Rel.min.	Nome istruz.
CURS02	CURS02	*PGM	SQL dinamico serial cursor	[NULL]	STM01
DESC01	DESC01	*PGM	SQL descriptor & parameter markers	[NULL]	PCURS1
DESC01B	DESC01B	*PGM	SQL descriptor & parameter markers	[NULL]	PCURS1
DESC02	DESC02	*PGM	SQL descriptor & resultset variabile	[NULL]	PCURS1
DESC03	DESC03	*PGM	SQL descriptor & par.marker + resultset variabile	[NULL]	PCURS1
DESC03B	DESC03B	*PGM	SQL descriptor & par.marker + resultset variabile	[NULL]	PCURS1

Istruz.declare	Istruz.prepare
DECLARE CURS02 CURSOR FOR STM01	PREPARE STM01 FROM : H
DECLARE CURS1 CURSOR FOR PCURS1	PREPARE PCURS1 FROM : H
DECLARE CURS1 CURSOR FOR PCURS1	PREPARE PCURS1 FROM : H
DECLARE CURS1 CURSOR FOR PCURS1	PREPARE PCURS1 FROM : H
DECLARE CURS1 CURSOR FOR PCURS1	PREPARE PCURS1 FROM : H
DECLARE CURS1 CURSOR FOR PCURS1	PREPARE PCURS1 FROM : H



Dettagli istruzione SQL/1

```
with
  PGM_STAT as
    (select [ . . . ]
      from SYSPROGRAMSTAT as A inner join SYSPROGRAMSTMTSTAT as B
        on (A.PROGRAM_SCHEMA, A.PROGRAM_NAME, A.MODULE_NAME) = (B.PROGRAM_SCHEMA, B.PROGRAM_NAME,
B.MODULE_NAME)
      where A.PROGRAM_SCHEMA = :Libreria),
  TAB_STAT as
    (select SYSTEM_PROGRAM_NAME, SQL_STATEMENT_TYPE, USAGE_TYPE, STATEMENT_NUMBER,
      listagg(ltrim((case when C.SCHEMA is null then ' ' else C.SCHEMA concat '/' end) concat C.NAME), ', ')
within group(order by C.NAME) as TABLES
      from PGM_STAT,
        table(PARSE_STATEMENT(SQL_STATEMENT => STATEMENT_TEXT, NAMING => NAMING, DECIMAL_POINT =>
DECIMAL_POINT, SQL_STRING_DELIMITER => SQL_STRING_DELIMITER)) as C
      where C.NAME_TYPE = 'TABLE'
      group by SYSTEM_PROGRAM_NAME, SQL_STATEMENT_TYPE, USAGE_TYPE, STATEMENT_NUMBER),
  COL_STAT as
    (select [ . . . ]
      where C.NAME_TYPE = 'COLUMN'
      group by SYSTEM_PROGRAM_NAME, SQL_STATEMENT_TYPE, USAGE_TYPE, STATEMENT_NUMBER) . . .
```



12

Dettagli istruzione SQL/2

```
...
select P.SYSTEM_PROGRAM_NAME "Programma", P.PROGRAM_TYPE "Tipo", T.TABLES "File", C.COLUMNS "Campi",
C.USAGE_TYPE "Uso", T.SQL_STATEMENT_TYPE "Tipo istruz.", P.STATEMENT_TEXT "Istruz."
from PGM_STAT as P
    inner join TAB_STAT as T
        on (P.SYSTEM_PROGRAM_NAME, P.STATEMENT_NUMBER) = (T.SYSTEM_PROGRAM_NAME, T.STATEMENT_NUMBER)
    left join COL_STAT as C
        on (P.SYSTEM_PROGRAM_NAME, P.STATEMENT_NUMBER) = (C.SYSTEM_PROGRAM_NAME, C.STATEMENT_NUMBER)
-- where C.COLUMNS like '%CLDRSO%' -- eventuale filtro sull'utilizzo di uno specifico campo
order by P.SYSTEM_PROGRAM_NAME, T.TABLES, T.SQL_STATEMENT_TYPE;
```

ABC Programma	ABC Tipo	ABC File	ABC Campi	ABC Uso	ABC Tipo istruz.	Istruz.
CRTALIAS	*PGM	STATI10	STCDEP, STNBOL, STNFAT	QUERY	QUERY	SELECT STCDEP INTO : H FROM STATI10 WHERE STNFAT = 2471 AND STNBOL = 2471
CURS01	*PGM	CLIEN00F	CLCCLI, CLCCME, CLDRSO	QUERY	DECLARE CURSOR	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , CLDRSO FROM CLIEN00F
CURS01B	*PGM	CLIEN00F	CLCCLI, CLCCME, CLDRSO	QUERY	DECLARE CURSOR	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , CLDRSO FROM CLIEN00F
CURS01B_B	*PGM	CLIEN00F	CLCCLI, CLCCME, CLDRSO	QUERY	DECLARE CURSOR	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , CLDRSO FROM CLIEN00F
CURS01C	*PGM	CLIEN00F	CLCCLI, CLCCME, CLDRSO	QUERY	DECLARE CURSOR	DECLARE CURS01 CURSOR FOR SELECT CLCCLI , CLDRSO FROM CLIEN00F



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



14