

## SELF

# *SQL Error Logging Facility*



Marco Riva



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



Ultimo aggiornamento: 21/07/2024

# SELF ?

## SQL Error Logging Facility

- ▶ aiuta a capire, analizzare e risolvere i problemi con gli errori e gli avvertimenti restituiti dalle istruzioni SQL
- ▶ SELF cattura i dettagli per categorie di errore o specifici codici di errore o avvertimenti SQL
- ▶ Può essere attivato per una specifica sessione SQL o per tutte le sessioni di tutti i job del sistema

<https://www.ibm.com/docs/en/i/7.5?topic=tools-sql-error-logging-facility-self>

da 7.4 TR7  
7.5 TR1

# Cosa fa?

- ▶ Quando viene attivato
  - ▶ controlla gli esiti SQLCODE delle istruzioni SQL
  - ▶ se SQLCODE coincide con uno di quelli che si è scelto di monitorare viene aggiunto o aggiornato un record nella tabella di log QSYS2 . **SQL\_ERROR** con le informazioni sull'errore e il contesto (istruzione SQL, stack chiamate, utente, ecc.)



3

# Come è fatto?

- ▶ Variabile globale **SYSIBMADM.SELFCODES**: contiene la lista degli sqlcode che si desidera controllare
  - ▶ L'impostazione di un valore di default nella creazione della variabile globale consente di attivare il monitoraggio per qualsiasi job
  - ▶ Per un monitoraggio in una specifica sessione SQL occorre eseguire l'istruzione SQL SET per modificare il valore della variabile globale

<https://www.ibm.com/docs/en/i/7.5?topic=services-selfcodes-global-variable>

- ▶ Vista **QSYS2.SQL\_ERROR\_LOG**: per interrogare tutto quanto è stato intercettato tramite SELF

<https://www.ibm.com/docs/en/i/7.5?topic=services-sql-error-log-view>



## Come si attiva?

- Occorre settare la variabile globale **SYSIBMADM.SELFCODES** con l'elenco degli SQLCODE che si desidera monitorare separati da , (virgola)
- Numero massimo di specifici SQLCODE monitorabili: **32**
- La funzione **VALIDATE\_SELF** consente di controllare se il valore settato in SELFCODES è sintatticamente corretto

```
-- attivazione controllo a livello globale  
create or replace variable SYSIBMADM.SELFCODES varchar(256)  
default (SYSIBMADM.VALIDATE_SELF('-913, -206'));
```

```
-- attivazione controllo nella sessione SQL corrente  
set SYSIBMADM.SELFCODES =  
(SYSIBMADM.VALIDATE_SELF('-913, -206'));
```

# Gruppi di errori

- Invece di elencare i singoli codici di errore è possibile settare la variabile globale SELFCODES con uno dei 3 valori speciali
  - **\*ERROR**: tutti i valori di sqlcode per gli errori
  - **\*WARN**: tutti i valori di sqlcode per gli avvertimenti
  - **\*ALL**: tutti i valori di sqlcode per errori e avvertimenti

```
-- solo errors
set SYSIBMADM.SELFCODES = '*ERROR';
-- solo warnings
set SYSIBMADM.SELFCODES = '*WARN';
-- entrambi
set SYSIBMADM.SELFCODES = '*ALL';
-- nessun controllo
set SYSIBMADM.SELFCODES = '*NONE';
```



È equivalente settare a 0 o null



6

## E' attivo?

- ▶ Per verificare se la registrazione SELF è attiva occorre visualizzare il contenuto della variabile globale **SYSIBMADM.SELFCODES**

```
-- verifica sqlcode monitorati  
values SYSIBMADM.SELFCODES;
```



7

# Come si legge il log?

► Interrogare la vista **SQL\_ERROR\_LOG**

```
select *  
from QSYS2.SQL_ERROR_LOG  
order by LOGGED_TIME desc;
```

Per la lista dei codici  
statement\_operation  
cfr. campo QQC21 in  
[Database Monitor view 1000](#)

123	LOGGED_SQLCODE	-104
ABC	LOGGED_SQLSTATE	42601
123	NUMBER_OCCURRENCES	1
	STATEMENT_TEXT	-- simulazione errore1SLECT *1 FROM EMPLOYEE
ABC	STATEMENT_OPERATION	EI
ABC	STATEMENT_OPERATION_DETAIL	EXECUTE IMMEDIATE
123	REASON_CODE	[NULL]
ABC	PROGRAM_LIBRARY	QSYS
ABC	PROGRAM_NAME	QZDASRV
ABC	PROGRAM_TYPE	*SRVPGM
ABC	MODULE_NAME	QZDASQL
	LOGGED_TIME	2024-07-20 10:47:39.204000
ABC	JOB_NAME	076525/QUSER/QZDASOINIT
123	THREAD_ID	0
ABC	ADOPTED_USER_NAME	

ABC	USER_NAME	
ABC	SYSTEM_USER_NAME	
ABC	CLIENT_ACCTNG	
ABC	CLIENT_APPLNAME	[NULL]
ABC	CLIENT_PROGRAMID	
ABC	CLIENT_USERID	[NULL]
ABC	CLIENT_WRKSTNNAME	[NULL]
ABC	RDB_NAME	
	INITIAL_LOGGED_TIME	2024-07-20 10:47:39.204000
ABC	INITIAL_JOB_NAME	076525/QUSER/QZDASOINIT
123	INITIAL_THREAD_ID	0
ABC	INITIAL_ADOPTED_USER_NAME	
	INITIAL_STACK	{'initial_stack':{{'ORD':11,'TYPE':'ILE','LIB':'QSYS','PGM':'QSQCUTE','MODULE':'QSQCUTE'}}



8



# Ripetizione della stessa istruzione

- ▶ Quando SELF intercetta un errore può inserire un nuovo record o aggiornare un record già esistente nella tabella SQL\_ERROR
- ▶ Il record viene aggiornato in base a questa **chiave**:
  - ▶ Libreria programma
  - ▶ Nome programma
  - ▶ Nome modulo
  - ▶ SQLCODE
  - ▶ Istruzione SQL
- ▶ In caso di aggiornamento viene incrementato il campo NUMBER\_OCCURRENCES
- ▶ I campi con prefisso INITIAL\_ si riferiscono alla prima registrazione del log



# Come si pulisce?

- Periodicamente è opportuno cancellare i record obsoleti dalla tabella `QSYS2.SQL_ERROR`

```
-- pulizia log più vecchi di 30 gg  
delete from QSYS2.SQL_ERROR  
  where LOGGED_TIME <= date(current date - 30 days);
```



# Come si usa?

## ► Caso 1) attivazione globale

è sufficiente settare il valore di default della variabile globale da questo momento SELF è attivo per tutti i job

## ► Caso 2) attivazione per job corrente

occorre settare la variabile globale nella sessione SQL corrente con istruzione SQL set

### **ATTENZIONE!**

Se in una sessione interattiva 5250 si apre l'interfaccia STRSQL per eseguire l'istruzione set per attivare SELF, appena si esce con F3 da strsql la sessione SQL termina e quindi anche il valore settato nella variabile globale SELF CODES ritorna al suo valore di default



# Attivazione dinamica SELF

- ▶ Se si desidera attivare SELF solo in un contesto specifico si può ricorrere a due metodi:
  1. Modificare il programma in esame aggiungendo l'istruzione set della variabile globale SELFCODES al momento in cui si necessita di attivare SELF
  2. Predisporre tutti i programmi embedded SQL ad eseguire l'istruzione set della variabile globale SELFCODES condizionata da una configurazione di attivazione dipendente dal job (p.es. con una variabile di ambiente) o dal contesto (p.es. un'area dati presente in lista librerie)



# Attivare SELF con una variabile di ambiente/1

- usiamo una variabile di ambiente (p.es. SELF\_ACTIVATION) settata a livello di job per informare i programmi che andranno in esecuzione se devono attivare o meno SELF

```
/* Imposto il valore della variabile di ambiente */  
ADDENVVAR ENVVAR(SELF_ACTIVATION) VALUE('*YES')  
LEVEL(*JOB) REPLACE(*YES)
```

```
// nel programma RPG dichiaro prototipo e variabili per reperire  
// il valore della variabile di ambiente  
dcl-pr getEnv pointer extproc('Qp0zGetEnvNoCCSID');  
  *n pointer value options(*string);  
end-pr;  
dcl-s pEnvVar pointer;  
dcl-s rtnEnvVarValue varchar(10) inz(*blanks);  
dcl-c EnvVarName 'SELF_ACTIVATION';  
dcl-s SelfStatus varchar(256);  
dcl-s Msg varchar(1000);
```



# Attivare SELF con una variabile di ambiente/2

```
pEnvVar = getEnv(EnvVarName);
if pEnvVar <> *null;
  rtnEnvVarValue = %str(pEnvVar);
endif;
select;
  when rtnEnvVarValue = '*YES';
    exec sql
      set SYSIBMADM.SELFCODES = '*ERROR';
  when rtnEnvVarValue = '*NO' or rtnEnvVarValue = *blanks;
    exec sql
      set SYSIBMADM.SELFCODES = '*NONE';
  when rtnEnvVarValue <> *blanks;
    exec sql
      set SYSIBMADM.SELFCODES = sysibmadm.validate_self(:rtnEnvVarValue);
endsl;
exec sql
  values SYSIBMADM.SELFCODES into :SelfStatus;
Msg = 'Stato di attivazione SELF: '+ SelfStatus;
snd-msg Msg;
```

reperisco valore variabile di ambiente

imposto il tipo di attivazione di SELF in base al valore della variabile di ambiente

informo nel joblog sullo stato di attivazione di SELF

Vedi esempio allegato TESTSELF .SQLRPGLE

# Best practice

- ▶ Porre attenzione ad attivare SELF su larga scala
- ▶ Per esempio settando il valore di default (= tutti i job) della variabile globale SELFCODES uguale a \*ALL (=tutti i possibili codici di errore e warning)
- ▶ Si potrebbe ottenere l'effetto di scrivere/aggiornare numerosi record da diversi job contemporaneamente nella tabella di log SQL\_ERRORRT con potenziali problemi di performance dovuti a lock concorrenti

- ▶ Usare ma non abusare



15

# 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



16