

SQL ci parla

SQLDA

descriptor area

parte I



Marco Riva



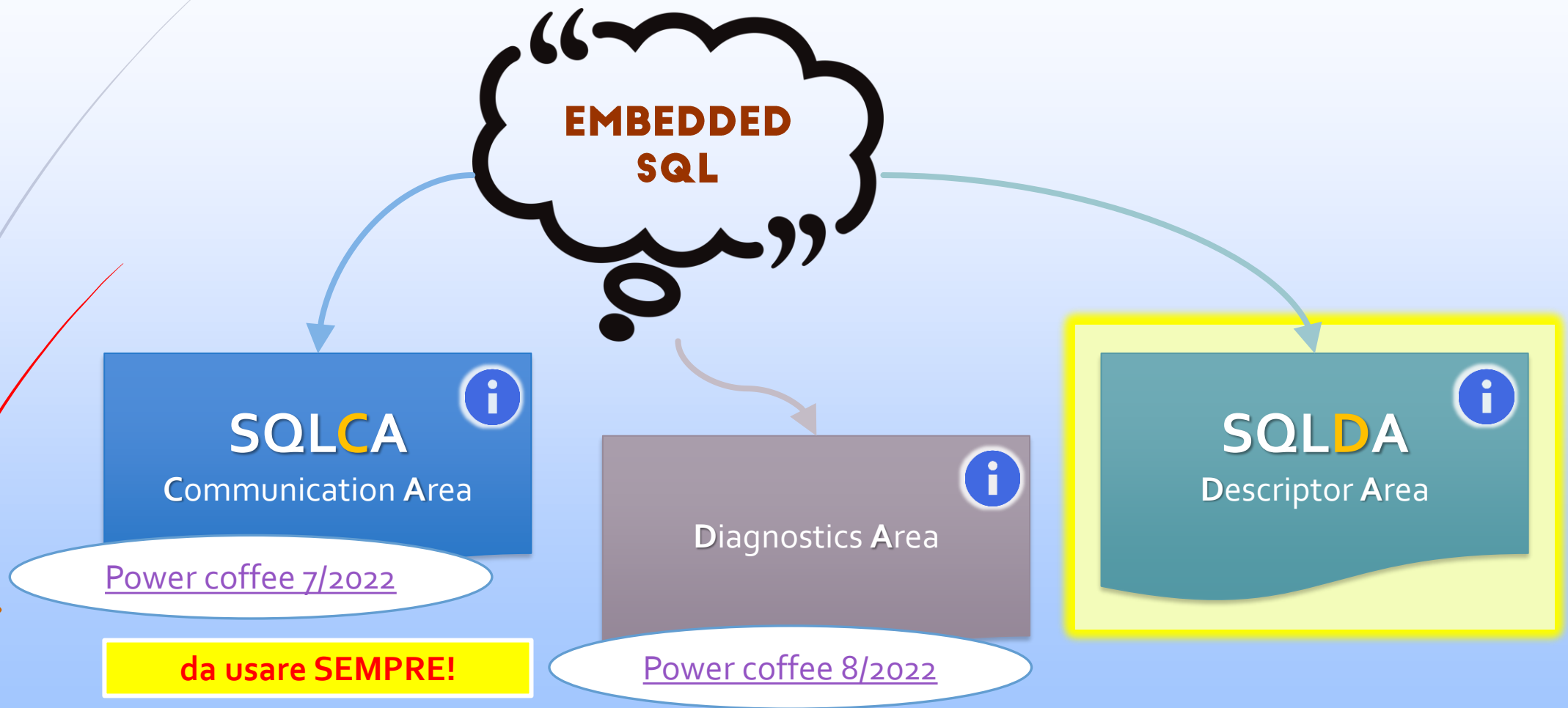
www.markonetools.it

IBM CHAMPION

2021-2022
Power area



Come ci parla l'SQL?



SQLCA – Communication Area



- ▶ E' la **struttura dati** fondamentale con cui SQL ci restituisce informazioni sull'esito di qualsiasi istruzione SQL
- ▶ In RPG è definita dal precompilatore SQL
- ▶ Viene valorizzata dopo ogni istruzione SQL
- ▶ E' da utilizzare *obbligatoriamente* per conoscere l'esito dell'istruzione (tranne istruzioni `DECLARE`). Embedded SQL non restituirà mai un'eccezione RPG come con I/O nativo
- ▶ In ogni programma esiste **una sola** SQLCA



3

SQL Diagnostics Area



- E' un insieme di informazioni aggiuntive rispetto a quanto restituito in SQLCA
- Vengono reperite tramite l'istruzione SQL **get diagnostics**



4

SQLDA – SQL Descriptor Area



- E' un insieme di variabili utilizzate nell'istruzione `describe` oppure dalle istruzioni `prepare`, `open`, `call`, `fetch` e `execute`
- In RPG l'SQLDA è definita dal precompilatore SQL
- L'SQLDA utilizzata in `describe` o `prepare` restituisce informazioni sull'istruzione `select` che è stata preparata



5

SQLDA - definizione



- ▶ Si definisce con l'istruzione SQL **include sqlda**
 - ▶ crea le strutture dati **SQLDA**, **SQLVAR**, **SQLVAR2**
- ▶ Il sottocampo **SQL_VAR** di **SQLDA** è un array con numero elementi **SQL_NUM**
- ▶ **SQL_NUM** va definito come una costante numerica con la dimensione richiesta per l'array **SQL_VAR**



SQLDA - header

- Nella parte di intestazione dell'SQLDA ci sono 4 variabili

campo	tipo	describe/prepare (impostato dal DB2 tranne SQLN)	fetch, open, call, execute (impostato da utente prima dell'esecuzione)
SQLDAID	char(8)	'SQLDA'	7° byte: '2', '3', '4' numero di occorrenze di SQLVAR per ogni colonna del resultset
SQLDABC	int	lunghezza SQLDA	da settare con valore \geq $16 + \text{SQLN} * 80$
SQLN	smallint	da settare con valore \geq 0 prima dell'esecuzione	occorrenze di SQLVAR
SQLD	smallint	numero colonne descritte da SQLVAR	occorrenze di SQLVAR



7

SQLDA – SQLVAR occurrences

- ▶ Ogni occorrenza contiene 5 variabili
- ▶ Lunghezza totale di ogni occorrenza: 80 bytes
- ▶ Per le istruzioni `prepare` e `describe` ogni occorrenza descrive una colonna del resultset



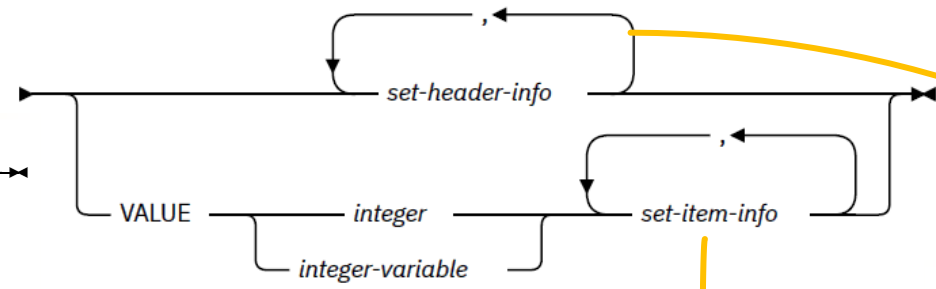
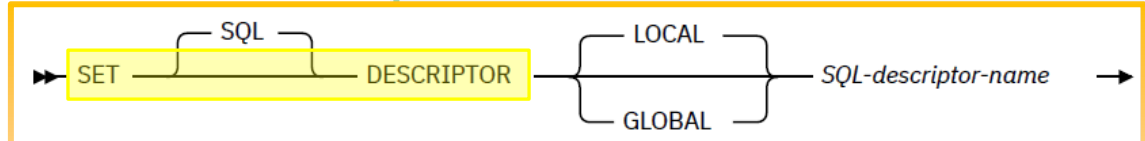
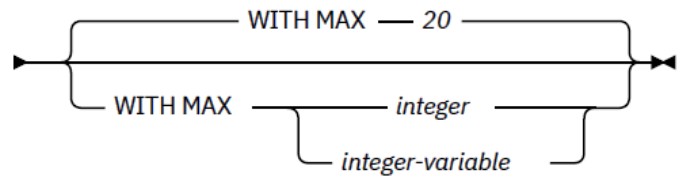
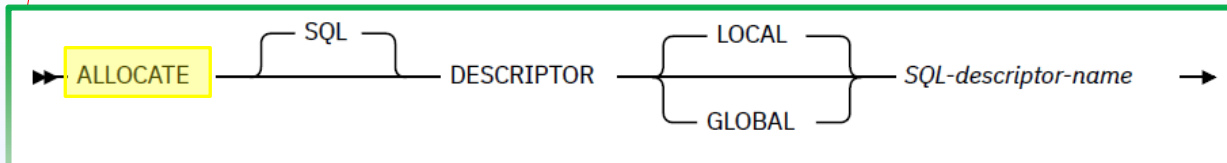
8

Manipolare il descrittore

- A partire dalla **V5R4** il DB2 agevola l'utilizzo della struttura SQLDA tramite l'uso dell'SQL descriptor
- allocate descriptor, deallocate descriptor
- get descriptor, set descriptor reperisce o imposta le informazioni di un descrittore
- using sql descriptor clausola da aggiungere su istruzione open per utilizzare il descrittore precedentemente impostato
- into sql descriptor clausola da aggiungere su istruzione fetch per indirizzare l'output al descrittore precedentemente impostato
- describe istruzione per reperire le informazioni di definizione di una tabella o del resultset di un'istruzione sql di select
- queste istruzioni possono essere utilizzate solo in embedded SQL

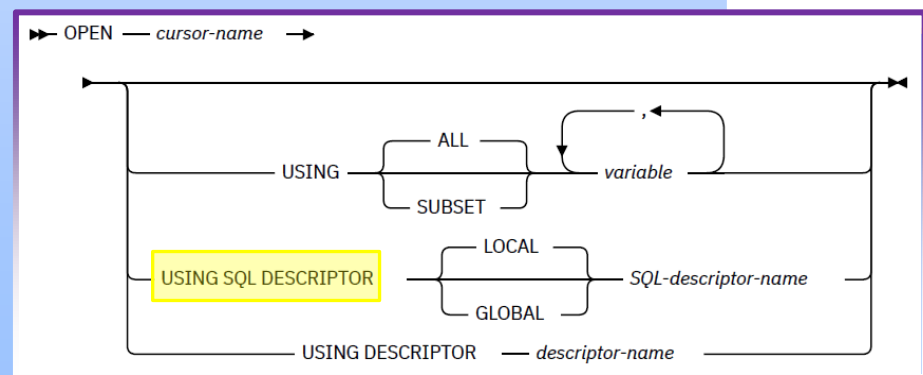
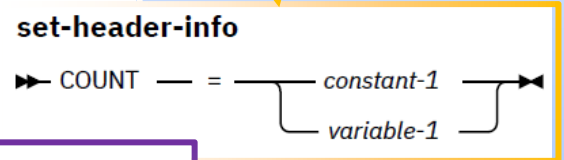


Allocazione e impostazione descrittore



set-item-info

CARDINALITY	=	constant-2
DATA		variable-2
DATETIME_INTERVAL_CODE		
DB2_CCSID		
INDICATOR		
LENGTH		
LEVEL		
PRECISION		
SCALE		
TYPE		
USER_DEFINED_TYPE_CATALOG		
USER_DEFINED_TYPE_NAME		
USER_DEFINED_TYPE_SCHEMA		



Descrittore e SQL dinamico/1

```
Codice da:  _____
Ufficio:   ___
(blank = tutti)

XXX  XX  XXXXXXXX
XXX  XX  XXXXXXXX
. . .
```

```
Codice da:  000010
Ufficio:   ___
(blank = tutti)

XXX  XX  XXXXXXXX
XXX  XX  XXXXXXXX
. . .
```

```
Codice da:  _____
Ufficio:   D11
(blank = tutti)

XXX  XX  XXXXXXXX
XXX  XX  XXXXXXXX
. . .
```

```
Codice da:  000010
Ufficio:   D11
(blank = tutti)

XXX  XX  XXXXXXXX
XXX  XX  XXXXXXXX
. . .
```

```
select *
from EMPLOYEE
order by EMPNO;
```

```
select *
from EMPLOYEE
where EMPNO >= :WSEMPNO
order by EMPNO;
```

```
select *
from EMPLOYEE
where WORKDEPT = :WSDEPT
order by EMPNO;
```

```
select *
from EMPLOYEE
where EMPNO >= :WSEMPNO
and WORKDEPT = :WSDEPT
order by EMPNO;
```

```
select *
from EMPLOYEE
where EMPNO >= :WSEMPNO
and WORKDEPT between :WKDEPT1 and :WKDEPT2
order by EMPNO;
```

unica istruzione ma con probabili decadenza di performance

Descrittore e SQL dinamico/2

- ▶ La soluzione con SQL dinamico avrebbe il vantaggio di ottimizzare la scrittura dell'istruzione per ottenere le migliori performance
- ▶ comporterebbe una maggiore attenzione nella scrittura del codice per concatenare correttamente costanti e contenuto delle variabili dei campi a video (e relativi apici!)

```
dcl-s sqlstm varchar(1000);
dcl-s sqlstmW like(sqlstm);
dcl-c AP const('');

sqlstm = 'select * from EMPLOYEE '
if WSEMPNO <> *blanks;
  sqlstmW = ' where EMPNO >= ' + AP + %trim(WSEMPNO) + AP;
endif;
if WSDEPT <> *blanks;
  if sqlstmW = *blanks;
    sqlstmW = ' where';
  endif;
  sqlstmW = sqlstmW + ' and WORKDEPT = ' + AP + %trim(WSDEPT) + AP;
endif;
sqlstm = sqlstm + ' order by EMPNO';
```

ATTENZIONE A:

- concatenazioni apici e doppi apici
- conversione variabili numeriche e data/ora in alfanumeriche



Descrittore e SQL dinamico/3

- ▶ E' sicuramente preferibile usare SQL dinamico con i parameter markers (?).
 - ▶ quanti sono i parameter markers da sostituire con using all'apertura del cursore?
 - ▶ a quali tipi di variabili corrispondono i parameter marker?
- ▶ ci aiuta l'utilizzo del **descrittore**:
 - ▶ allocazione descrittore
 - ▶ aggiunta condizione where ove richiesto con parameter marker
 - ▶ aggiunta del valore di confronto di ogni parametro al descrittore
 - ▶ apertura cursore con il descrittore

nel medesimo ordine posizionale

```
select *  
from EMPLOYEE  
order by EMPNO;
```

Power coffee - MK1

```
select *  
from EMPLOYEE  
where EMPNO >= ?  
order by EMPNO;
```

```
select *  
from EMPLOYEE  
where WORKDEPT = ?  
order by EMPNO;
```

```
select *  
from EMPLOYEE  
where EMPNO >= ?  
and WORKDEPT = ?  
order by EMPNO;
```



Descrittore e SQL dinamico/4

➤ allocazione del descrittore

```
exec sql  
  allocate descriptor local 'dCurs1' with max 2;
```

numero max
dei parameter
markers

nome del
descrittore

➤ preparazione per costruzione istruzione SQL

```
dcl-s sqlstm varchar(1000);  
dcl-s sqlstmW like(sqlstm);  
dcl-s NumParm uns(3);  
  sqlstm = 'select EMPNO, LASTNAME, FIRSTNME +  
           from EMPLOYEE ';
```

ATTENZIONE agli
apici e al **case-
sensitive!**
E' meglio usare una
variabile host

➤ aggiunta filtri di selezione

```
if WSEMPNO <> *blanks;  
  sqlstmW = ' where EMPNO >= ?';  
  NumParm += 1;  
  exec sql  
    set descriptor 'dCurs1'  
      value :NumParm type = 1, length = 6, data = :WSEMPNO;  
endif;
```

type
1 = character
2 = numeric
3 = decimal
9 = date/time
12 = varchar

local: ambito
interno al
programma
global: ambito
della sessione SQL
quindi può essere
usato anche in un
altro programma



14

Esempi allegati:
DESC01.SQLRPGLE,
DESC01B.SQLRPGLE

Descrittore e SQL dinamico/5

- ▶ imposto il numero di parameter marker effettivamente inseriti nell'istruzione SQL

```
exec sql
  set descriptor 'dCurs1' count = :NumParm;
```

- ▶ preparo l'istruzione SQL (prepare)

- ▶ apro il cursore utilizzando il descrittore

```
exec sql
  open Curs1 using SQL descriptor 'dCurs1';
```

- ▶ leggo il cursore (`fetch`) e chiudo il cursore (`close`) e poi disalloco il descrittore. Il descrittore viene *implicitamente* disallocato insieme alla chiusura del cursore.

```
exec sql
  deallocate descriptor local 'dCurs1';
```

ATTENZIONE!

Se si usa una costante come nome descrittore e si digitano caratteri non corrispondenti nelle varie istruzioni si riceve errore SQL0077
"Nome descrittore non corretto"



Un passo avanti...

- ▶ Questo è uno dei possibili modi per utilizzare il descrittore SQL.
- ▶ Nei prossimi Power Coffee vedremo altri esempi



16

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



17