

SQL ci parla

SQLDA

descriptor area

parte II



Marco Riva



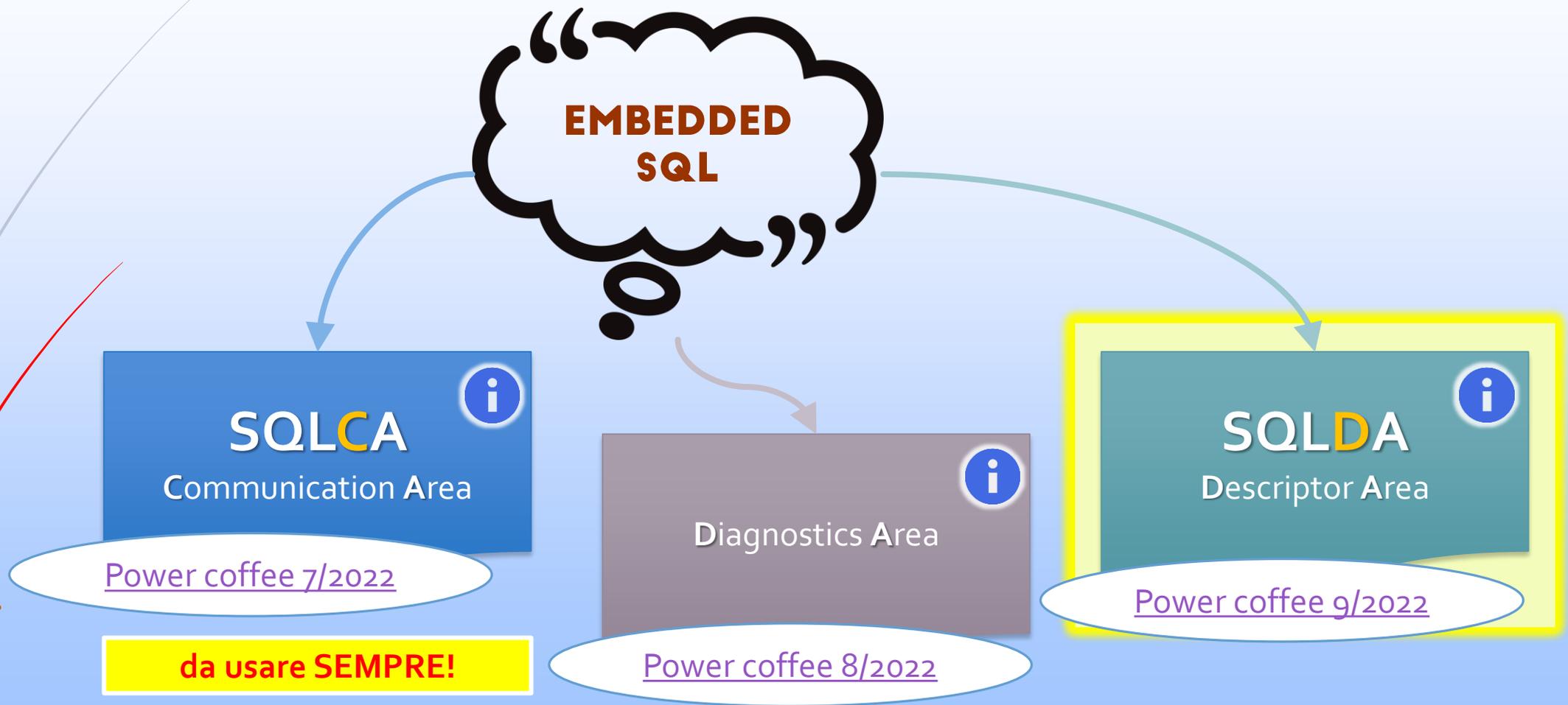
www.markonetools.it

IBM CHAMPION

2021-2022
Power area



Come ci parla l'SQL?



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



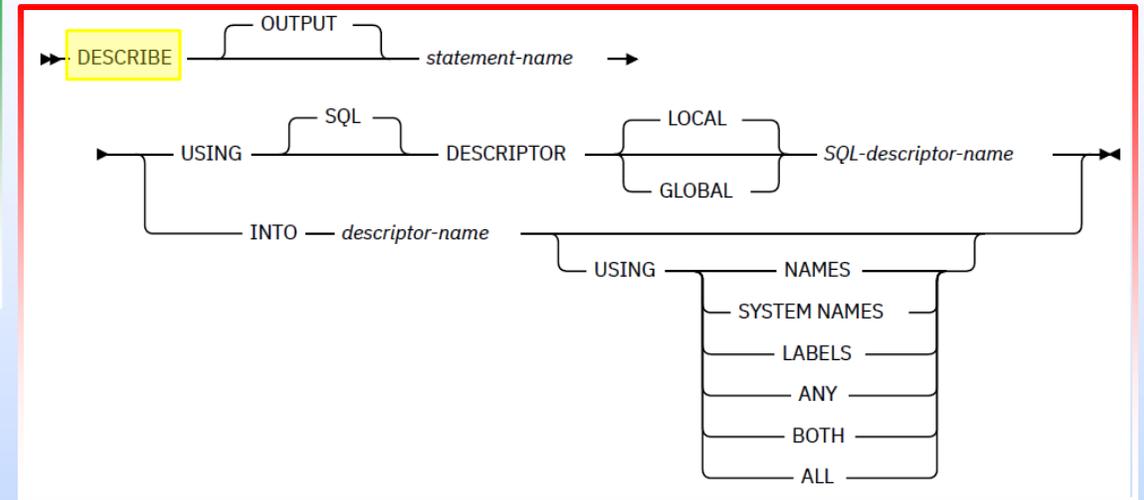
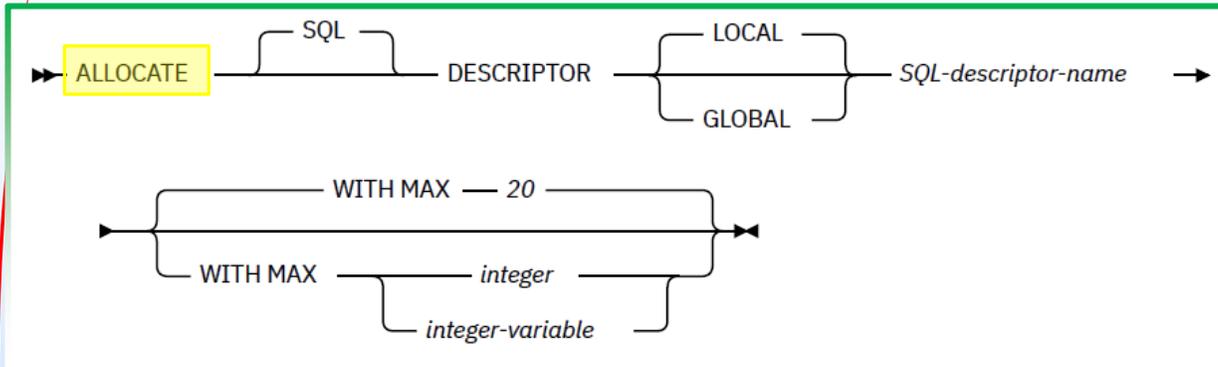
3

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, impostazione e lettura descrittore



get-item-info

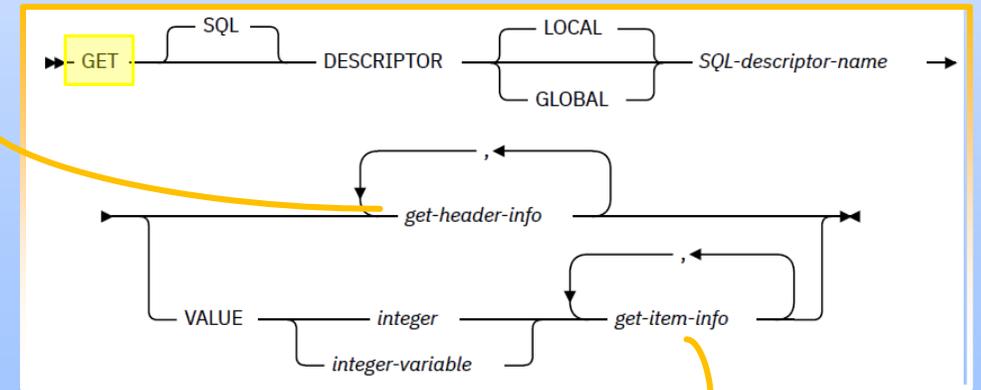
variable-2

| |
|----------------------------|
| CARDINALITY |
| DATA |
| DATETIME_INTERVAL_CODE |
| DB2_BASE_CATALOG_NAME |
| DB2_BASE_COLUMN_NAME |
| DB2_BASE_SCHEMA_NAME |
| DB2_BASE_TABLE_NAME |
| DB2_CCSID |
| DB2_COLUMN_CATALOG_NAME |
| DB2_COLUMN_GENERATED |
| DB2_COLUMN_GENERATION_TYPE |
| DB2_COLUMN_HIDDEN |
| DB2_COLUMN_NAME |
| DB2_COLUMN_ROW_CHANGE |
| DB2_COLUMN_SCHEMA_NAME |
| DB2_COLUMN_TABLE_NAME |
| DB2_COLUMN_UPDATABILITY |
| DB2_CORRELATION_NAME |

get-header-info

variable-1

| |
|--------------------------|
| COUNT |
| DB2_CURSOR_HOLDABILITY |
| DB2_CURSOR_RETURNABILITY |
| DB2_CURSOR_SCROLLABILITY |
| DB2_CURSOR_SENSITIVITY |
| DB2_CURSOR_UPDATABILITY |
| DB2_MAX_ITEMS |
| DB2_RESULT_SETS_COUNT |
| DYNAMIC_FUNCTION |
| DYNAMIC_FUNCTION_CODE |
| KEY_TYPE |



5

Resultset variabile/1

- Un caso tipico in cui il descrittore SQL ci viene in aiuto con un'istruzione costruita dinamicamente è quando l'elenco delle colonne del resultset è variabile
 - quante variabili host ho bisogno per la clausola INTO?
 - come definisco le variabili host in modo che siano congruenti con il tipo dati del resultset?

```
Ufficio rif.:      ___  
Sequenza colonne: Codice 0 Descrizione 1 Uff.rif. 3 Resp. 2  
(0 per non visualizzare)
```

```
XXXXXXXXXXXXXXXXXX  XXX  XXXXXX  
XXXXXXXXXXXXXXXXXX  XXX  XXXXXX  
. . .
```

filtro sui record

ordine in cui visualizzare le colonne del resultset

```
select DEPTNAME, MGRNO, ADMRDEPT  
from DEPARTMENT  
where ADMRDEPT = ?  
order by DEPTNO;
```

Resultset variabile/2

- ▀ allocazione del descrittore

```
exec sql  
  allocate descriptor local :Desc with max 4;
```

```
dcl-s Desc varchar(20)  
  inz('dCurs1');
```

- ▀ costruisco la clausola select in base alle scelte dell'utente. Il codice può essere più o meno complesso in base a come viene proposto all'utente la possibilità di scelta (elenco di campi preimpostato di cui scegliere l'ordine, digitazione libera dei nomi campi, ecc.)

```
dcl-s sqlstm  
  varchar(500);
```

```
sqlstm = 'select DEPTNAME, MGRNO, DEPTNO';
```

- ▀ a cui aggiungo le clausole from e where

```
sqlstm = sqlstm + ' from DEPARTMENT +  
  where ADMRDEPT = ?';
```



7

Esempio allegato:
DESC02.SQLRPGLE

Resultset variabile/3

- Una volta costruita nella variabile `sqlstm` l'istruzione SQL eseguo la preparazione

```
exec sql  
  prepare pCurs1 from :sqlstm;
```

```
exec sql declare Curs1  
  cursor for pCurs1;
```

- Tramite il descrittore reperisco le informazioni sull'istruzione appena preparata

```
exec sql  
  describe pCurs1 using sql descriptor local :Desc;
```

- Reperisco il numero di colonne del resultset

```
exec sql  
  get descriptor :Desc :numCol = COUNT;
```

```
dcl-s numCol int(10);
```

SQLDA

intestazione

descrizione
colonna 1

descrizione
colonna 2

descrizione
colonna ...



8

Resultset variabile/4

- ▶ Apro il cursore

```
exec sql  
  open Curs1 using :WSDEPT;
```

- ▶ Leggo il cursore indirizzando l'output al descrittore SQL

```
exec sql  
  fetch next from Curs1 into sql descriptor :Desc;
```

- ▶ Per ogni colonna leggo dati e informazioni dal descrittore

```
// reperimento nome della prima colonna della tabella  
exec sql  
  get descriptor :Desc value 1 :MyCol = DB2_SYSTEM_COLUMN_NAME;  
// reperimento dati della prima colonna della tabella  
exec sql  
  get descriptor :Desc value 1 :MyVar = DATA;
```

sqlcode = -76

altrimenti...

ATTENZIONE all'esatta
corrispondenza del tipo dati

Resultset variable/5

➤ Altre informazioni reperibili dal descrittore con **get descriptor**:

➤ *header*:

➤ DB2_CURSOR_UPDATABILITY: 1 se cursore aggiornabile

➤ *item*:

➤ TYPE, LENGTH, PRECISION, SCALE: tipo dati, lunghezza, precisione e scala della colonna

➤ DATETIME_INTERVAL_CODE: specificazione del tipo dati data (1 date, 2 time, 3 timestamp)

➤ DB2_COLUMN_UPDATABILITY: 1 se la colonna è aggiornabile

➤ DB2_LABEL: etichetta della colonna

type

1 = character

12 = varchar

2 = numeric

3 = decimal

4 = integer

5 = smallint

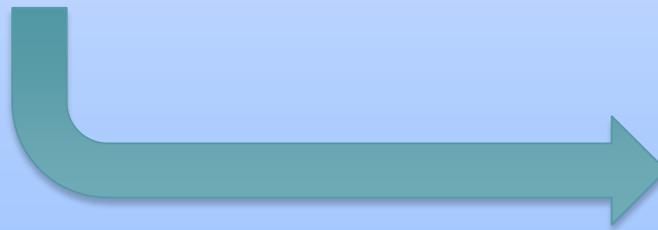
9 = date/time



10

Parameter markers variabili + resultset variabile?/1

- ▶ Nell'esempio del [Power coffee 9/2022](#) abbiamo utilizzato il descrittore per governare il numero variabile di parameter markers
- ▶ Ora abbiamo utilizzato il descrittore per poter gestire un resultset variabile
- ▶ E' possibile combinare le due caratteristiche per gestire un SQL dinamico che abbia un numero variabile di parameter markers e un numero variabile di colonne nel resultset?



Parameter markers variabili + resultset variabile?/2

- ▶ riprendiamo l'esempio visto in precedenza aggiungendo la possibilità che il valore *blanks* nel campo "Ufficio rif." indichi di non filtrare i records

ordine in cui visualizzare le colonne del resultset

```
Ufficio rif.:      ___ (blanks = tutti)
Sequenza colonne: Codice 0 Descrizione 1 Uff.rif. 3 Resp. 2
                  (0 per non visualizzare)
```

```
XXXXXXXXXXXXXXXXX  XXX  XXXXXX
XXXXXXXXXXXXXXXXX  XXX  XXXXXX
. . .
```

filtro sui record
opzionale

```
select DEPTNAME, MGRNO, ADMRDEPT
from DEPARTMENT
where ADMRDEPT = ?
order by DEPTNO;
```



Parameter markers variabili + resultset variabile?/3

► allocazione di due descrittori

```
// descrittore per i parameter markers
exec sql
  allocate descriptor local :DescA with max 1;
// descrittore per il resultset
exec sql
  allocate descriptor local :DescB with max 4;
```

```
dcl-s DescA varchar(20)
      inz('dCurs1A');
```

```
dcl-s DescB varchar(20)
      inz('dCurs1B');
```

► se uso il parameter marker imposto il descrittore dCurs1A

```
exec sql
  set descriptor :descA
    value 1 type = 1, length = 3, data = :pmDept;
exec sql
  set descriptor :descA count = 1;
```

► altrimenti imposto il numero elementi a 0

```
exec sql
  set descriptor :descA count = 0;
```



13

Esempio allegato:
DESC03.SQLRPGLE

Parameter markers variabili + resultset variabile?/4

- ▶ l'altro descrittore `dCurs1B` lo utilizzo per reperire le informazioni sul resultset

```
exec sql
  prepare pCurs1 from :sqlstm;
exec sql
  describe pCurs1 using sql descriptor local :DescB;
exec sql
  get descriptor :DescB :numCol = COUNT;
```

- ▶ per l'apertura del cursore uso il descrittore `dCurs1A`

```
exec sql
  open Curs1 using SQL descriptor :descA;
```

- ▶ per la lettura del cursore uso il descrittore `dCurs1B`

```
exec sql
  fetch next from Curs1 into sql descriptor :DescB;
```



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



15