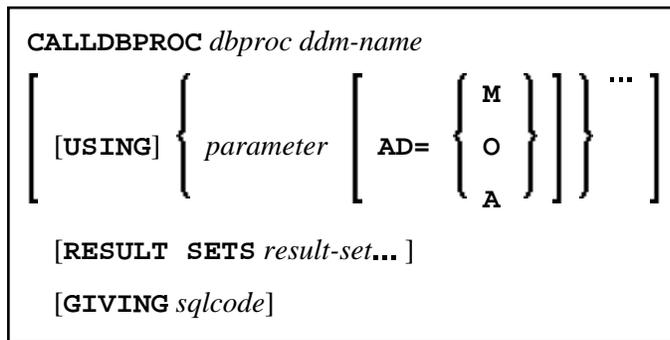


CALLDBPROC - SQL



Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiel

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Funktion

Das Statement CALLDBPROC dient dazu, eine Stored Procedure des SQL-Datenbanksystems, mit dem Natural verbunden ist, aufzurufen.

Die Stored Procedure kann entweder ein Natural-Subprogramm oder ein in einer anderen Programmiersprache geschriebenes Programm sein. Neben der Möglichkeit, Parameter zwischen dem aufrufenden Objekt und der Stored Procedure zu übergeben, unterstützt CALLDBPROC sogenannte Result Sets; mit diesen ist es möglich, größere Datenmengen von der Stored Procedure an das aufrufende Objekt zu übergeben, als dies mittels Parametern möglich wäre.

Die Result Sets sind von der Stored Procedure erzeugte temporäre Ergebnistabellen, die das aufrufende Objekt mittels eines READ RESULT SET-Statements lesen und verarbeiten können.

Anmerkung:

Im Prinzip ist der Aufruf einer Stored Procedure mit dem Aufruf eines Natural-Subprogramms vergleichbar: wenn das CALLDBPROC-Statement ausgeführt wird, wird die Kontrolle an die Stored Procedure übergeben; nach Verarbeitung der Stored Procedure wird die Kontrolle wieder an das aufrufende Objekt zurückgegeben, und die Verarbeitung wird mit dem nächsten Statement nach dem CALLDBPROC-Statement fortgesetzt.

Syntax-Beschreibung

<i>dbproc</i>	<p>Aufzurufende Stored Procedure:</p> <p>Als <i>dbproc</i> geben Sie den Namen der Stored Procedure an, die aufgerufen werden soll. Der Name kann entweder als alphanumerische Variable oder als Konstante in Apostrophen (') angegeben werden.</p> <p>Der Name muss den Regeln für Stored-Procedure-Namen des Ziel-Datenbanksystems entsprechen.</p> <p>Falls die Stored Procedure ein Natural-Subprogramm ist, darf der eigentliche Procedure-Name nicht länger als 8 Stellen sein.</p>						
<i>ddm-name</i>	<p>Name eines Natural-Datendefinitionsmoduls:</p> <p>Der Name eines DDMs muss angegeben werden, um die "Adresse" der Datenbank, die die Stored Procedure ausführt, bereitzustellen. Weitere Informationen siehe <i>ddm-name</i>.</p>						
[USING] <i>parameter</i>	<p>Zu übergebende Parameter:</p> <p>Hier können Sie Parameter angeben, die vom aufrufenden Objekt an die Stored Procedure übergeben werden sollen.</p> <p>Als <i>parameter</i> können Sie Folgendes angeben:</p> <ul style="list-style-type: none"> ● <i>host-variable</i> (optional mit INDICATOR- und LINDICATOR-Klauseln) ● eine Konstante oder ● das Schlüsselwort NULL. <p>Weitere Informationen siehe <i>host-variable</i>.</p>						
AD=	<p>Attribut-Definition:</p> <p>Wenn es sich bei dem <i>parameter</i> um eine <i>host-variable</i> handelt, können Sie ihn wie folgt markieren:</p> <table border="1" data-bbox="415 1318 1393 1671"> <tr> <td data-bbox="415 1318 594 1436">AD=O</td> <td data-bbox="594 1318 1393 1436">Nicht änderbar, siehe Session-Parameter AD=O. (Entsprechende Prozedur-Notation in DB2 for z/OS: IN.)</td> </tr> <tr> <td data-bbox="415 1436 594 1549">AD=M</td> <td data-bbox="594 1436 1393 1549">Änderbar, siehe Session-Parameter AD=M. (Entsprechende Prozedur-Notation in DB2 for z/OS: INOUT.)</td> </tr> <tr> <td data-bbox="415 1549 594 1671">AD=A</td> <td data-bbox="594 1549 1393 1671">Nur zur Eingabe, siehe Session-Parameter AD=A. (Entsprechende Prozedur-Notation in DB2 for z/OS: OUT.)</td> </tr> </table> <p>Wenn der <i>parameter</i> eine <i>constant</i> ist, kann AD nicht explizit angegeben werden. Für Konstanten gilt immer AD=O.</p>	AD=O	Nicht änderbar, siehe Session-Parameter AD=O. (Entsprechende Prozedur-Notation in DB2 for z/OS: IN.)	AD=M	Änderbar, siehe Session-Parameter AD=M. (Entsprechende Prozedur-Notation in DB2 for z/OS: INOUT.)	AD=A	Nur zur Eingabe, siehe Session-Parameter AD=A. (Entsprechende Prozedur-Notation in DB2 for z/OS: OUT.)
AD=O	Nicht änderbar, siehe Session-Parameter AD=O. (Entsprechende Prozedur-Notation in DB2 for z/OS: IN.)						
AD=M	Änderbar, siehe Session-Parameter AD=M. (Entsprechende Prozedur-Notation in DB2 for z/OS: INOUT.)						
AD=A	Nur zur Eingabe, siehe Session-Parameter AD=A. (Entsprechende Prozedur-Notation in DB2 for z/OS: OUT.)						

RESULT SETS <i>result-set</i>	Feld für Result-Set-Locator-Variable: <p>Als <i>result-set</i> geben Sie ein Feld an, in das der Result-Set-Locator zurückgegeben werden soll.</p> <p>Ein <i>result-set</i> muss eine Variable mit Format/Länge I4 sein.</p> <p>Der Wert einer <i>result-set</i>-Variablen ist lediglich eine Zahl, die den Result Set identifiziert und die in einem nachfolgenden READ RESULT SET-Statement referenziert werden kann.</p> <p>Die Reihenfolge der <i>result-set</i>-Werte entspricht der Reihenfolge der von der Stored Procedure zurückgegebenen Result Sets.</p> <p>Der Inhalt der Result Sets kann von einem nachfolgenden READ RESULT SET-Statement verarbeitet werden. Wenn kein Result Set zurückgegeben wird, enthält die betreffende <i>result-set</i>-Variable Null (0).</p> <p>Es kann nur ein Result-Set angegeben werden.</p>
GIVING <i>sqlcode</i>	GIVING <i>sqlcode</i>-Option: <p>Mit dieser Option können Sie den SQL-Code des SQL CALL-Statements erhalten, das die Stored Procedure aufruft.</p> <p>Wenn Sie diese Option angeben und der SQL-Code der Stored Procedure ist nicht Null (0), wird keine Natural-Fehlermeldung ausgegeben. In diesem Fall muss die als Reaktion auf den SQL-Codewert auszuführende Handlung im aufrufenden Natural-Objekt programmiert werden.</p> <p>Das <i>sqlcode</i>-Feld muss eine Variable von Format/Länge I4 sein.</p> <p>Wenn Sie die Option GIVING <i>sqlcode</i> nicht verwenden, gibt Natural eine Fehlermeldung aus, falls der SQL-Code der Stored Procedure nicht Null (0) ist.</p>

Beispiel

Das folgende Beispiel zeigt ein Natural-Programm, das die Stored Procedure DEMO_PROC aufruft, um alle zu einem gegebenen Bereich gehörenden Namen der Tabelle PERSON einzulesen.

Drei Parameter-Felder werden an DEMO_PROC übergeben: der erste und zweite Parameter übergeben jeweils Start- und Endwerte des Bereichs von Namen an die Stored Procedure, und der dritte Parameter nimmt einen Namen auf, der das Kriterium erfüllt.

In diesem Beispiel werden die Namen in einem Result Set zurückgegeben, der mit dem READ RESULT SET-Statement verarbeitet wird.

```

DEFINE DATA LOCAL
1 PERSON VIEW OF DEMO-PERSON
  2 PERSON_ID
  2 LAST_NAME
1 #BEGIN      (A2) INIT <'AB'>
1 #END        (A2) INIT <'DE'>
1 #RESPONSE  (I4)

```

```
1 #RESULT    (I4)
1 #NAME (A20)
END-DEFINE

...

CALLDBPROC 'DEMO_PROC' DEMO-PERSON #BEGIN (AD=0) #END (AD=0) #NAME (AD=A)
    RESULT SETS #RESULT
    GIVING #RESPONSE

READ RESULT SET #RESULT INTO #NAME FROM DEMO-PERSON
    GIVING #RESPONSE
    DISPLAY #NAME
END-RESULT

...

END
```