

Mehrstufige Query/400-Auswertungen

Die Übernahme von einfachen Query/400-Auswertungen nach DB2 Web Query kann problemlos mit dem Query/400-Adapter erfolgen. Dieser ist im Browser basierten Synonym-Editor integriert. Aber was ist mit Query/400-Auswertungen, die sich über mehrere Stufen erstrecken? Dabei wird das Ergebnis der ersten Stufe für die Auswertung auf der zweiten Stufe verwendet. Das Ergebnis der zweiten Stufe wird für die Auswertung der dritten Stufe usw. verwendet. Um diesen Ablauf mit DB2 Web Query abzubilden, wende ich einen Trick an, den ich Ihnen hier beschreibe.

Übernahme von mehrstufigen Query/400

Mit Query/400 und CL ist es möglich, auch mehrstufige Auswertungen zu erstellen. Dabei greift das jeweils nachfolgende Query auf eine von dem vorherigen Query erstellte Datei zu. Dies kann über beliebig viele Stufen durchgeführt werden.

Mit DB2 Web Query ist dieser Ablauf nicht eins-zu-eins umsetzbar, etwa mit dem Synonym-Editor. Dazu müsste jede einzelne Stufe manuell aufgerufen und ausgeführt werden. Zu überlegen wäre auch, ob die letztendlich erzielte Auswertung nicht über einen anderen Ansatz erreicht werden kann. Vielleicht wäre es sinnvoll, hierfür eine MQT (= Materialized Query Table) zu verwenden.

Hier beschreibe ich Ihnen eine Lösung, die mit Hilfe einer Stored Procedure (= Gespeicherte Prozedur) erzielt wird.

Die beiden Query/400-Anwendungen

Für meine Erläuterungen verwende ich als Beispiel zwei Query/400-Auswertungen, die auch in der Bibliothek QWQCENT enthalten sind. Die erste Auswertung CHAINQRY1 verwendet die beiden Tabellen ORDERS und INVENTORY aus der Bibliothek QWQCENT. Das Ergebnis wird in eine Datenbankdatei CHAINFILE ausgegeben. Diese wiederum dient für die zweite Auswertung CHAINQRY2 als Eingabedatei. Die beiden Query/400-Anwendungen werden in einem CL-Programm CHAINQRY nacheinander ausgeführt.

```
PGM

RUNQRY      QRY (QWQCENT/CHAINQRY1)
RUNQRY      QRY (QWQCENT/CHAINQRY2)

ENDPGM
```

Abbildung 1: CL-Programm zum Ausführen von zwei Query/400-Auswertungen

CHAINQRY1 führt folgendes aus:

1. Verknüpft und verarbeitet die beiden Tabellen ORDERS und INVENTORY.
2. Sortiert nach Produkttyp und Produktkategorie und führt dafür einen Gruppenwechsel durch.
3. Die Differenz zwischen Umsatz und Kosten wird als Profit berechnet.
4. Das Ergebnis wird in eine Datenbankdatei CHAINFILE in der Bibliothek QTEMP ausgegeben.

Das zweite Query/400-Programm CHAINQRY2 verarbeitet die soeben erstellte Datei wie folgt:

1. Die Datei CHAINFILE in QTEMP wird eingelesen.
2. Die Gewinnspanne wird anhand von Profit und Umsatz ermittelt.
3. Die Sortierung erfolgt nach Gewinnspanne, Produkttyp und Produktkategorie.
4. Ein Gruppenwechsel erfolgt bei Produkttyp und Produktkategorie.
5. Das Ergebnis wird am Bildschirm angezeigt.

```

Ergebnisfelder definieren

Feldnamen oder Konstanten und Operatoren eingeben und Eingabetaste drücken.
Operatoren: +, -, *, /, SUBSTR, !!, DATE...

Feld      Ausdruck      Spaltenüberschrift      Länge Dez.
PFTMARGIN (grosspro01/revenue01)*100      Gross Profit      Margin

```

Abbildung 2: Ermitteln der Gewinnspanne

Das Ergebnis dieser mehrstufigen Auswertung stellt sich wie folgt dar:

```

Bericht anzeigen
Abfrage . . . : QWQCENT/CHAINQRY2      Breite des Berichts . . : 76
Anfang auf Zeile . . . . .      Anfang in Spalte . . . . .
Zeile . . . . .1. . . . .2. . . . .3. . . . .4. . . . .5. . . . .6. . . . .7. . . . .
      Gross Profit      Product Type      Product Category
      Margin
000001      70,87      Audio      Speakers
000002
000003      57,68      Office      Organizers
000004
000005      52,17      Camcorders      Digital8 Camcorders
000006
000007      39,26      Audio      Amplifiers/PreAmps/Tuners
000008
000009      39,21      Audio      MP3
000010
000011      35,95      Audio      Receivers
000012
000013      33,78      Camcorders      MiniDV Camcorders
000014
000015      32,75      Audio      Audio Systems

```

Abbildung 3: Das Ergebnis der mehrstufigen Query/400-Auswertung

Anpassung für die Umsetzung nach DB2 Web Query

Damit dieser Ablauf auch nach Web Query übernommen und mit Hilfe einer Stored Procedure dargestellt werden kann, sind einige Änderungen erforderlich. Zunächst muss die Ausgabe von CHAINQRY2 in eine temporäre Ausgabedatei anstelle eines Bildschirms erfolgen. Die dann zu erstellende Stored Procedure soll die Zeilen dieser temporären Datei in einen Web Query-Report ausgeben.

Dazu sind die folgenden Schritte notwendig:

1. Das CL-Programm CHAINQRY wird so geändert, dass das Endergebnis in eine temporäre Datei RTNCHAIN erfolgt und nicht auf einen Bildschirm. Die durchgeführten Änderungen sind gelb hinterlegt.

```

PGM
DLTF      FILE (QTEMP/RTNCHAIN)
MONMSG      MSGID (CPF2105)
RUNQRY      QRY (QWQCENT/CHAINQRY1)
RUNQRY      QRY (QWQCENT/CHAINQRY2) OUTTYPE (*OUTFILE) +
      OUTFILE (QTEMP/RTNCHAIN)

```

Abbildung 4: Geändertes CL-Programm CHAINQRY**Beachten Sie:**

Wenn Ihr CL-Programm einen Parameter enthält und der Benutzer dadurch zur Eingabe einer Satzauswahl aufgefordert wird (RUNQRY mit RCDSL(*YES)), so muss dieser Parameter entfernt werden. Das Filtern kann letztendlich im Web Query-Report erfolgen.

2. Wandeln Sie das CL-Programm um.
3. Als nächstes wird ein RPG-Programm erstellt, welches als Eingabeparameter den Namen und die Bibliothek des CL-Programms erwartet.
Das RPG-Programm ruft das CL-Programm auf, liest die dabei erstellte temporäre Datei QTEMP/RTNCHAIN ein, deklariert einen Cursor für diese Datei und gibt das Ergebnis als SQL-Ergebnismenge zurück. Dieses Programm wird dann als Stored Procedure (Gespeicherte Prozedur) registriert. Daraus wird mit DB2 Web Query ein Synonym erstellt, welches die Voraussetzung für die Erstellung von Reports mit DB2 Web Query liefert.

Hinweis:

Grundsätzlich ist es nicht erforderlich, den Namen und die Bibliothek des CL-Programms als Eingabeparameter zu deklarieren. Beides könnte auch fest im RPG-Programm kodiert werden. Mit der Lösung als Eingabeparameter hat man eine flexible Anwendung, die für alle mehrstufigen Query/400-Anwendungen verwendet werden kann. Allerdings muss dann für jede eindeutige SQL-Ergebnismenge ein eigenes Synonym erstellt werden.

```
H option(*srcstmt : *nodebugio)
D PGMNAME          S              10
D LIBNAME          S              10
D PGMLIB           S              21
C   *ENTRY          plist
C                   PARM          PGMNAME
C                   PARM          LIBNAME
/free
    if pgmname = *blanks;
        return;
    endif;
    if libname = *blanks;
        libname = '*LIBL';
    endif;

    pgmlib = %trim(libname) + '/' + %trim(pgmname);
/end-free
C                   CALL          PGMLIB
/free
EXEC SQL
    DECLARE c1 CURSOR WITH RETURN TO CLIENT FOR
    SELECT *
    FROM qtemp/rtnchain;
EXEC SQL
    OPEN c1;
```

Abbildung 5: RPG-Programm als Grundlage für eine Stored Procedure

4. Wandeln Sie das Programm um mit:

```
CRTSQLRPGI OBJ(LibName/WQ_CHAIN) SRCFILE(LibName/QRPGLESRC) COMMIT(*NONE)
```

Anstelle von **LibName** setzen Sie Ihren Bibliotheksnamen.

5. Erstellen Sie für dieses Programm eine Stored Procedure mit:

```
CREATE PROCEDURE LibName.WQ_CHAIN (  
  IN PGMNAME CHAR(10) ,  
  IN LIBNAME CHAR(10) )  
  LANGUAGE RPGLE  
  SPECIFIC LibName.WQ_CHAIN  
  NOT DETERMINISTIC  
  READS SQL DATA  
  CALLED ON NULL INPUT  
  EXTERNAL NAME 'LibName/WQ_CHAIN '  
  PARAMETER STYLE GENERAL ;
```

Abbildung 6: Stored Procedure für das RPG-Programm

6. Für diese Stored Procedure wird ein Synonym erstellt.
Der Objekttyp wird auf *Gespeicherte Prozeduren* beschränkt.

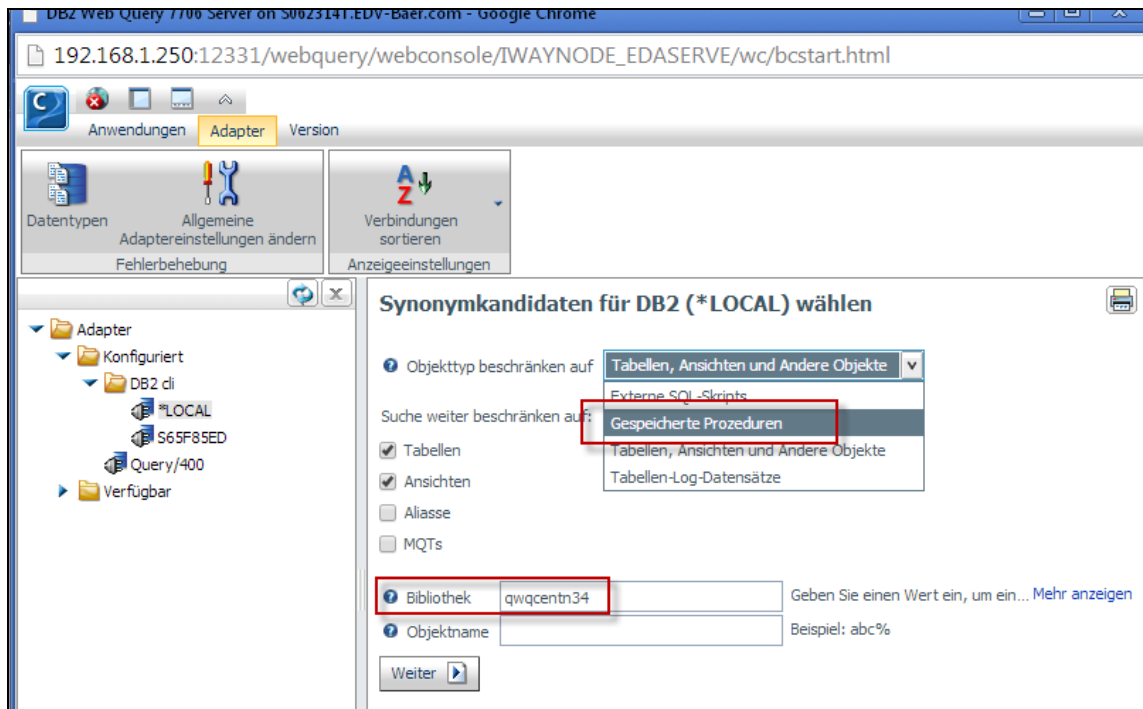


Abbildung 7: Auswahl *Gespeicherte Prozeduren* (Stored Procedure) für Synonym-Erstellung
Die oben erstellte Stored Procedure wird ausgewählt.

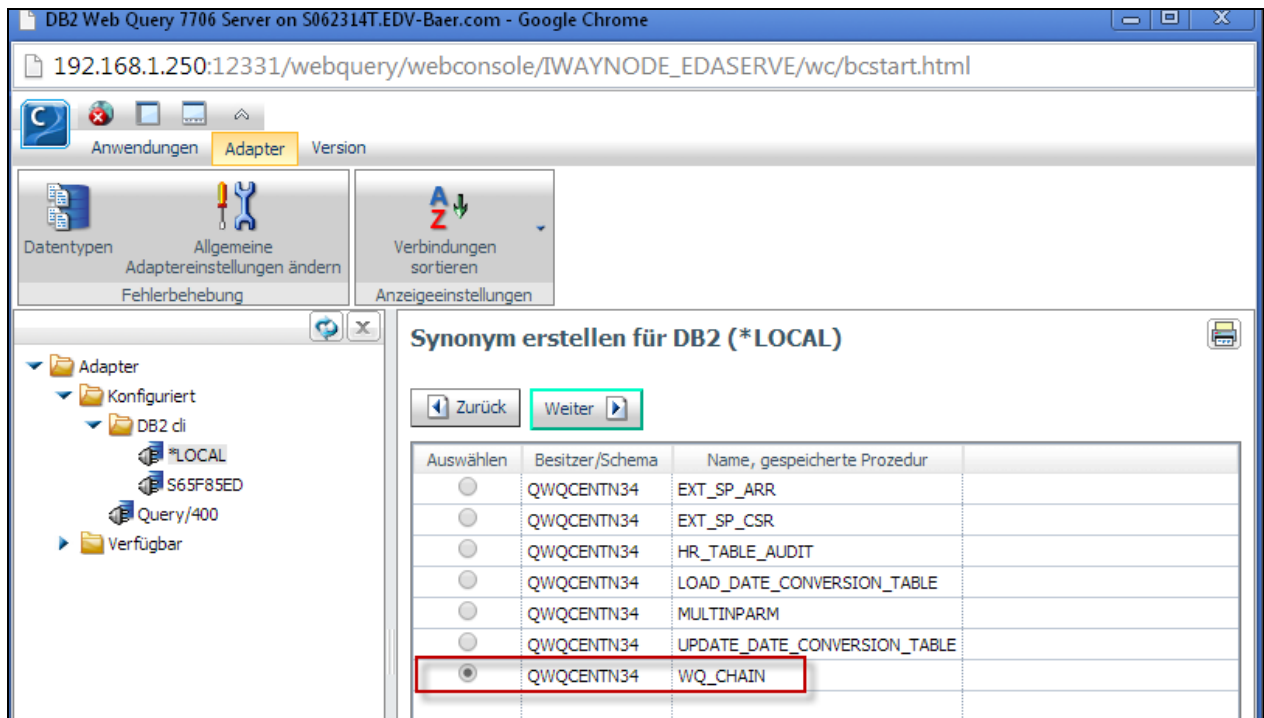


Abbildung 8: Oben erstellte Stored Procedure *WQ_CHAIN* auswählen, für die ein Synonym erstellt werden soll

Der Synonym-Editor erkennt, dass zwei Input-Parameter vorhanden sind. Geben Sie bitte für diese jeweils einen gültigen Wert ein.

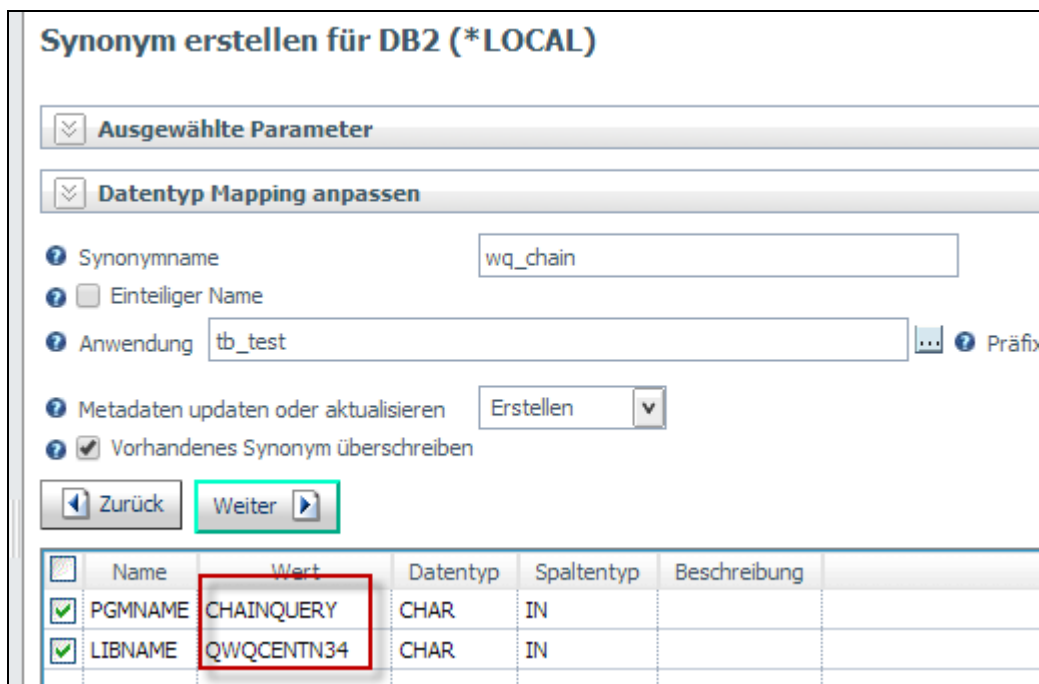


Abbildung 9: Werte für die Eingabeparameter angeben

7. Mit diesem Synonym *WQ_CHAIN* wird nun ein Report erstellt.

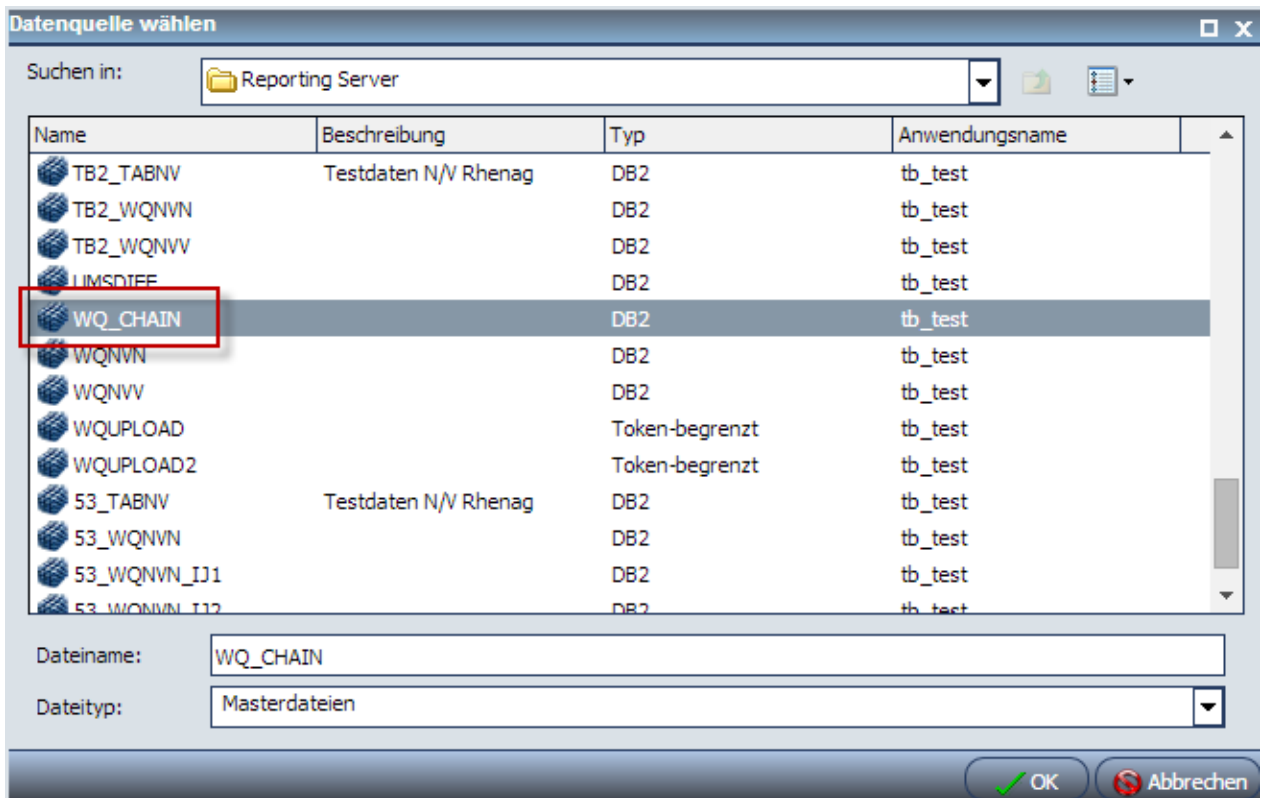


Abbildung 10: Auswahl des Synonyms der Stored Procedure für die Erstellung eines Reports

Für den Report müssen die beiden Eingabeparameter als Filter und die anderen Felder als Datenfelder definiert werden.

The screenshot shows the SAP WebFOCUS interface. On the left, the 'Daten' pane shows a data source 'WQ_CHAIN' with dimensions 'PGMNAME', 'LIBNAME', 'Product Type', and 'Product Category', and a measure 'Gross Profit, Margin'. Below this, the 'Filter' pane shows 'PGMNAME Gleich Einfacher Parameter' and 'LIBNAME Gleich Einfacher Parameter' selected. The 'Abfrage' pane shows a report configuration for 'Report (WQ_CHAIN)' with a 'Summe' function, grouped by 'Product Type' and 'Product Category', and displayed across columns. The top right pane shows the 'Parameter' section with 'PGMNAME: CHAINQRY' and 'LIBNAME: QWQCEN34', and buttons for 'Ausführen', 'Zurücksetzen', and 'Ausgabe löschen'. The main area displays a table of results with columns 'Product Type', 'Product Category', and 'Gross Profit Margin'.

Product Type	Product Category	Gross Profit Margin
.	.	27,71
Audio	Amplifiers/PreAmps/Tuners	39,26
	Audio Systems	32,75
	CD Players and Recorders	29,73
	MP3	39,21
	Receivers	35,95
	Speakers	70,87
Camcorders	Digital8 Camcorders	52,17
	DVD Camcorders	20,82
	MiniDV Camcorders	33,78
Cameras	Digital Cameras	27,58
Office	Handheld and PDA	24,10
	Organizers	57,68
Video	DVD	24,59
	TV	10,68
	VCR	24,98

Abbildung 11: Report, der anhand der Stored Procedure erstellt wurde

Hinweis:

Schicken Sie mir eine E-Mail, falls Sie dazu weitere Informationen benötigen.

Vorschau:

Im nächsten Artikel werde ich Ihnen weitere Tipps+Tricks zeigen.

Bis dahin wünsche ich Ihnen weiterhin viel Spaß beim Vermehren Ihrer Fertigkeiten.

Den Autor Theo Bär erreichen Sie unter EDV-Beratung Theo Bär - Ringmauerweg 1 - 69250 Schönau -
Tel. (+49) 6228 912 630 - e-Mail info@edv-baer.com