

IBM DB2 Web Query – Verwendung von Stored Procedures

Im letzten Teil dieser Reihe wurde der Einsatz von SQL-Views gezeigt. Der aktuelle Artikel führt das dort erläuterte Beispiel weiter und zeigt, wie auch Stored Procedures mit DB2 Web Query verwendet werden können.

Voraussetzungen

Für die Beispiele verwende ich Tabellen, die in einer Beispielbibliothek des System i-Navigators enthalten sind.

- ORG – Organisation, 8 Zeilen bzw. Sätze
- STAFF – Mitarbeiter 36 Zeilen bzw. Sätze

Die Tabelle STAFF wurde um eine Spalte USRPRF erweitert.

Die Verwendung von Stored Procedures (Gespeicherte Prozeduren)

Mit DB2 Web Query ist es auch möglich, Synonyme zu erstellen, die auf *Stored Procedures* basieren. Eine Stored Procedure ist ein Programm oder eine Funktion in einem Serviceprogramm, die innerhalb einer Anwendung mit der SQL-Anweisung **CALL** aufgerufen werden kann. Stored Procedures können mit Hilfe der prozeduralen SQL-Sprache geschrieben werden. Ebenso können bereits vorhandene Programme oder Prozeduren aus Serviceprogrammen dazu verwendet werden. Es spielt dabei keine Rolle, in welcher Sprache diese Programme erstellt wurden (RPG, COBOL, CL, Java usw.).

Wenn eine Stored Procedure eine Ergebnismenge (= Result Set) zurückgibt, kann diese von DB2 Web Query als Quelle für einen Bericht verarbeitet werden. Für die Rückgabe der Ergebnismenge in einer Stored Procedure wird normalerweise ein Cursor verwendet. Anstelle des Cursors könnte allerdings auch eine SQL-View stehen (siehe auch den letzten Artikel dieser Reihe). Allerdings gibt es hier einige Unterschiede und Vorteile, die die Verwendung von Stored Procedures bietet:

- Bereits existierende Programme können verwendet werden. Diese Programme können auch dahingehend angepasst werden, dass sie eine Ergebnismenge zurück geben. Dies könnte z.B. durch Auskommentieren der Überschriftszeilen und Modifizieren der Detailausgaben geschehen. Anstatt die Ausgabe auf einen Drucker zu legen, könnte diese in eine temporäre Datei oder Ähnliches erfolgen. Dadurch könnte die manchmal komplexe Programmlogik erhalten bleiben und gleichzeitig die Darstellung der Ausgabe in einer modernen Art erfolgen.
- An Stelle von SQL können für das Lesen native Operationen wie READ und CHAIN verwendet werden.
- Es können andere Programme, Befehle und System APIs aufgerufen werden. Ebenso können Informationen von anderen Systemobjekten wie Nachrichtenwarteschlangen, Datenwarteschlangen oder Datenbereiche ausgelesen werden.

Das folgende Beispiel zeigt die Verwendung einer Stored Procedure. Neben der Erstellung eines Berichtes mit Hilfe einer SQL-View, die im letzten Artikel besprochen wurde, wird noch eine Überwachungsfunktion hinzugefügt. Dies bedeutet, dass jedes Mal, wenn der Bericht ausgeführt wird, auch ein Satz in eine Log-Datei mit einer Zeitmarke, dem Berichtsnamen und dem Namen der Benutzers geschrieben wird.

Die Anweisungen für die Erstellung der Log-File ist in Abbildung 1 dargestellt.

```
/* Tabelle für Auditing erstellen */
CREATE TABLE rptaudlog (
    rpttimst TIMESTAMP,
    rptname CHAR(25),
    rptusrprf CHAR(10));
```

Abbildung 1: Erstellen der Tabelle zur Überwachung

```

/* Stored Procedure erstellen */
drop procedure secure_departments_proc;
CREATE PROCEDURE secure_departments_proc
DYNAMIC RESULT SETS 1
LANGUAGE SQL
NOT DETERMINISTIC
MODIFIES SQL DATA
specific secure_departments_proc
P1 : BEGIN
    DECLARE c1 CURSOR WITH RETURN TO CLIENT FOR      (1)
        SELECT division, deptname, lname, fname, salary
        FROM secure_departments;                      (2)
        INSERT INTO rptaudlog VALUES(CURRENT TIMESTAMP,   (3)
        'SECURE_DEPARTMENTS', SESSION_USER);
        OPEN c1 ;                                     (4)
END P1 ;

```

Abbildung 2: Erstellen der Stored Procedure

Erläuterungen zur Stored Procedure:

- (1) Standardmäßig wird für die Deklaration eines Cursors die Anweisung WITH RETURN TO CALLER verwendet. Dabei kann es zu Konflikten kommen, wenn eine Verschachtelung von Stored Procedures vorliegt. Der Cursor wird deshalb deklariert mit WITH RETURN TO CLIENT. Dadurch wird sicher gestellt, dass die Ergebnismenge an die Client-Anwendung zurück gegeben wird.
- (2) Hier wird der SQL-View aus der letzten Folge dieser Reihe eingesetzt.
- (3) Die SQL-Anweisung INSERT fügt bei jedem Aufruf eine Zeile in die Log-Datei.
- (4) Der Cursor wird und bleibt geöffnet. Dabei wird die Ergebnismenge an die Clientanwendung, nämlich DB2 Web Query, zurück gegeben.

Zusätzlich zum kompletten Namen `secure_departments_proc` vergibt das System einen internen 10-stelligen Namen. Dieser wird dann verwendet, um die Stored Procedure mit der SQL-Anweisung **CALL** aufzurufen:

`call secur00001`

Erstellen des Synonyms für die *Gespeicherte Prozedur (Stored Procedure)*

Damit eine Stored Procedure mit DB2 Web Query verwendet werden kann, ist es erforderlich, das entsprechende Synonym zu erstellen. Dies geschieht auf die bereits mehrfach gezeigte Art der Erzeugung von Metadaten.

Bei *Objekttyp beschränken auf* wird nun "Gespeicherte Prozeduren" ausgewählt (siehe Abbildung 3).

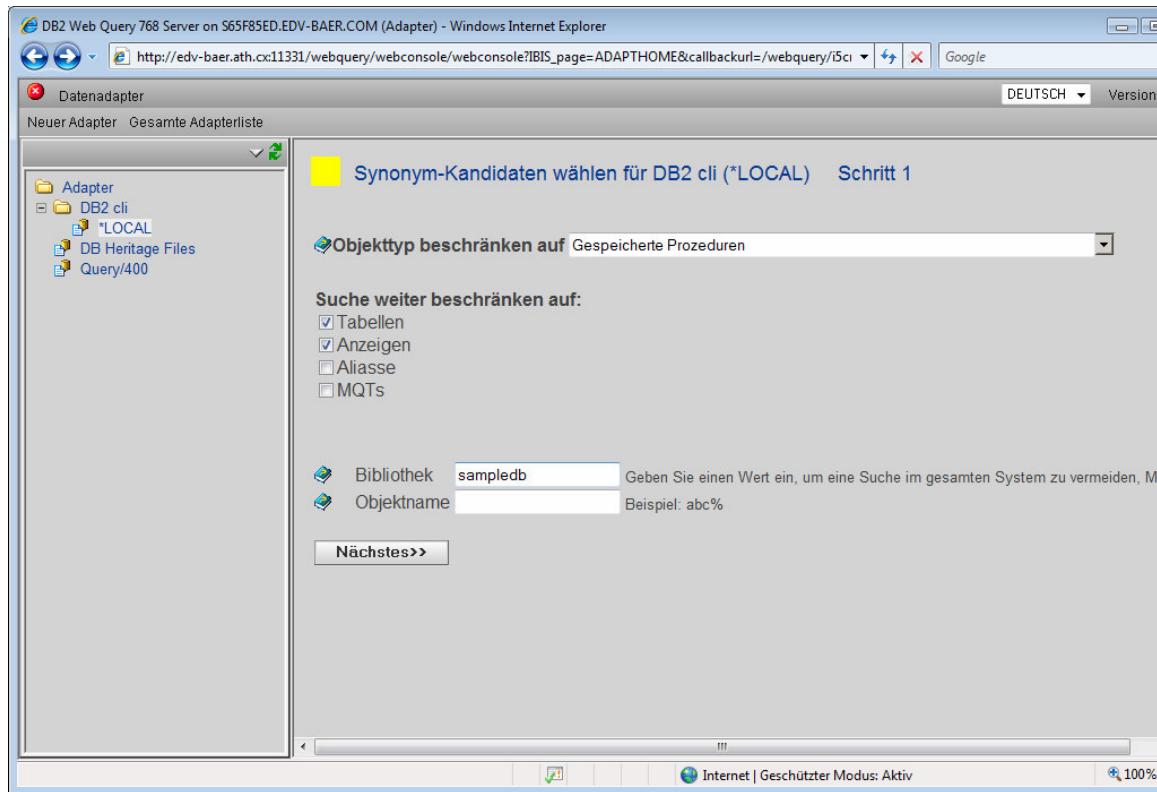


Abbildung 3: Auswahl "Gespeicherte Prozeduren" (Stored Procedures)

Damit nicht alle Stored Procedures des gesamten Systems angezeigt werden, wird noch die Bibliothek angegeben ("SAMPLEDB"), die die oben erzeugte Prozedur enthält.

Im nächsten Fenster werden alle in der ausgewählten Bibliothek gefundenen Stored Procedures angezeigt. Hier kann diejenige ausgewählt werden, für die das Synonym erstellt werden soll (Abbildung 4).

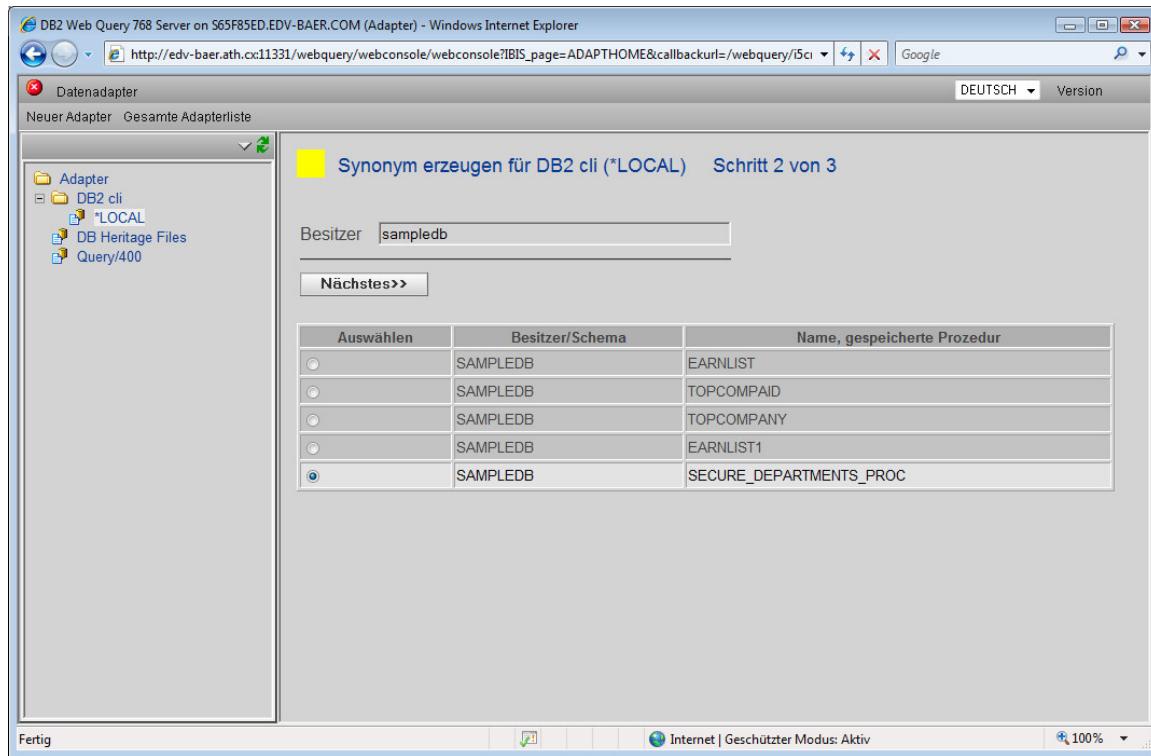


Abbildung 4: Auswahl der Stored Procedure für die Synonymerstellung

Um die Default-Einstellungen für die Datentyp-Einstellungen zu ändern, wird die entsprechende Checkbox markiert. Die anpassbaren Zuordnungen werden angezeigt (Abbildung 5).

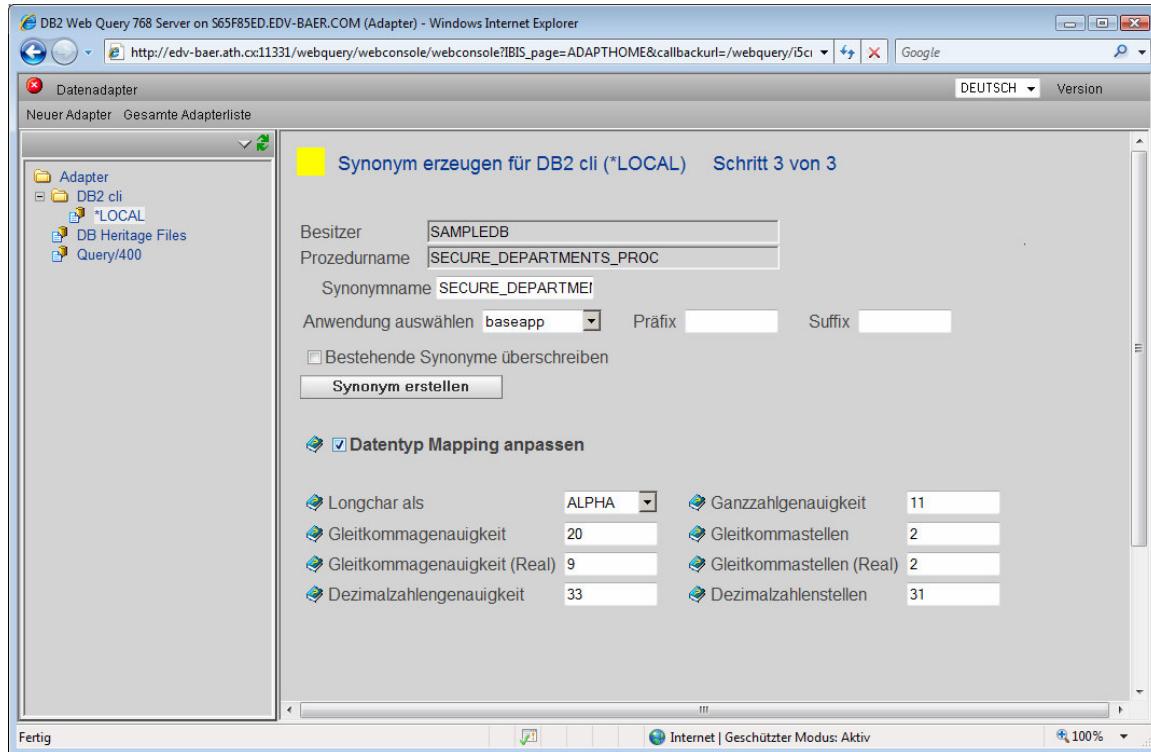


Abbildung 5: Datentypen anpassen

Nach Fertigstellung kann die ursprüngliche Stored Procedure jetzt mit DB2 Web Query verwendet werden.

Vorschau für die nächste Folge:

In der nächsten Folge werde ich weitere Funktionen von DB2 Web Query beschreiben.

Unter anderem wird erläutert, wie die oft große Anzahl von Tabellenspalten mit Hilfe von **Business Views** reduziert und damit die Übersichtlichkeit von Synonymen verbessert werden kann.

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