



SQL:2003 – Neue Konzepte

Universität Bonn

Seminar „Intelligente Datenbanken“

Prof. Dr. R. Manthey / Andreas Behrend

14. Juni 2005

Gereon Schüller

Email: X@Y mit

X=„schuelle“ Y=„informatik.uni-bonn.de“

Quellen

- Eisenberg, Andrew et al., „SQL:2003 Has Been Published“, SIGMOD Record, Vol.33, No. 1, pp. 119-126, March 2004
- iAnywhere, Dokumentation zu „SQL Anywhere 9.0.2“, http://www.iAnywhere.com/developer/product_manuals/sqlanywhere/0902/de/html/index.html
*(besonders umfangreiches Material zur Syntax von SQL)**
- Kulkarni, Krishna, „Overview of SQL:2003“, Silicon Valley Laboratory of the IBM Corporation, San Jose, 2003-11-06
- Türker, Can, „SQL:2003 – Was dürfen wir erwarten?“, Datenbank Spektrum 4/2002
- Türker, Can „SQL:1999 & SQL:2003“, dpunkt.verlag, Heidelberg 2003 *(auch gute Referenz für SQL 1999)**
- W3C, „XQuery 1.0: An XML Query Language“, W3C Working Draft 04 April 2005 <http://www.w3.org/TR/xquery>*

* eigene Quellen

SQL:2003 – Ausgewählte Neuheiten

- Identitätsspalten u. Sequenzgeneratoren
- Generierte Spalten
- Merging
- tabellenwertige Funktionen
- Multimengen als Datentyp
- SQL/XML

[Identitätsspalten]

- künstlich erzeugter Schlüssel
- unter Access als „AutoWert“ bezeichnet
- Startwert, Schrittweite und Endwert frei wählbar
- Datentyp muss numerisch sein
- Spaltenwert trotzdem manuell änderbar (auf Wunsch)
- wird vom System konsistent gehalten

Definieren von Identitätsspalten

```
CREATE TABLE Vorlesungen (  
  Vorlesungs_ID INTEGER  
    GENERATED ALWAYS AS IDENTITY  
    START WITH 1000  
    INCREMENT 10  
    MINVALUE 1000  
    NO MAXVALUE  
    NO CYCLE,  
  Titel VARCHAR(50),  
  Dozent VARCHAR(50))
```

Spalte wird automatisch generiert als Identität

mit 1000 starten, um 10 hochzählen, Mindesteingabe 1000

kein Höchstwert, nicht zyklisch weiterzählen

<u>Veranstaltungs_ID</u>	Titel	Dozent
1000	Informatik I	Meier
1010	Analysis	Heine
1020	Analysis	Borel

Sequenzgeneratoren

Seminare		Praktika	
Seminar_ID	Titel	Praktikum_ID	Titel
1000	SQL	1010	Java

```
CREATE Sequence Veranstaltungs_Seq
  START WITH 1000
  INCREMENT 10
  MINVALUE 1000
  NO MAXVALUE
  NO CYCLE
```

```
INSERT INTO Seminare(Seminar_ID, Titel) VALUES (NEXT
VALUE FOR Veranstaltungs_Seq, 'SQL:2003')
```

```
INSERT INTO Praktika(Praktikum_ID, Titel) SELECT NEXT
VALUE FOR Veranstaltungs_Seq, Titel FROM Vorschlaege
```

Generierte Spalten: Motivation

- Spalten werden oft aus anderen Spalten berechnet
- Neuberechnung durch Abfrage nicht immer sinnvoll

=Nettopreis*1,16

Artikelname	Nettopreis	Bruttopreis
Motherboard	100	116
Drucker	110	127,60
Tastatur	10	11,60

Generierte Spalten

- in SQL:2003 lassen sich Spalten als „generiert“ definieren
- Attribute von generierten Spalten dürfen nur von Attributen des eigenen Tupels abhängen
- Attribut wird dann bei jeder Datensatzänderung automatisch neu berechnet

```
CREATE TABLE Artikel (  
  ArtikelName AS VARCHAR(50),  
  Nettopreis DECIMAL(7,2),  
  BruttoPreis ALWAYS GENERATED AS (Nettopreis * 1.16))
```

Kein SELECT-FROM-WHERE
Block erlaubt

[UPDATE oder INSERT?]

- beim Zusammenfügen von Tabellen können Tupel mit gleichem Schlüssel in beiden Tabellen vorhanden sein

Lagerbestand	
<u>Artikelnummer</u>	Anzahl
4711	17
0815	134

Addition nötig

Lieferung	
<u>Artikelnummer</u>	Anzahl
2891	7
4711	19

Merging

- im Falle gleicher Schlüssel ist kein Insert, aber ggf. ein Update notwendig
- komplizierte Transaktion notwendig
- Vereinfachung durch neues Statement:
- SQL stellt mit MERGE eine Fallunterscheidung für jedes Tupel bereit

```
MERGE INTO Lager
USING (SELECT ArtikelNummer, Anzahl) FROM Lieferung)
  ON Lager.ArtikelNummer=Lieferung.ArtikelNummer)
WHEN MATCHED THEN
  UPDATE SET Anzahl=Lager.Anzahl+Lieferung.Anzahl
WHEN NOT MATCHED THEN
  INSERT (ArtikelNummer, Anzahl) VALUES
    (Lieferung.ArtikelNummer,Lieferung.Anzahl)
```

[Tabellenwertige Funktionen]

- Aufruf als Tabellenreferenz:
`SELECT a,b FROM meineFunktion(parameter);`
- zwei Sorten:
 1. interne SQL-Funktionen (bisher in SQL/PLI)
 2. externe Funktionen in anderen Programmiersprachen
- Abbildung von Werten auf eine virtuelle Tabelle
- ggf. Abbildung von externer Funktion auf SQL-Tabelle

Interne tabellenwertige Funktion

```
CREATE FUNCTION StudentenImFS(Semesterzahl INTEGER)
```

```
  RETURNS TABLE (  
    MatrNr INTEGER,  
    Name VARCHAR(15),  
    Vorname VARCHAR(15))
```

```
  LANGUAGE SQL  
  READS SQL DATA  
  DETERMINISTIC
```

```
  RETURN TABLE(  
    SELECT Matrikelnummer, Name, Vorname  
    FROM Studenten  
    WHERE Studenten.FS=Semesterzahl);
```

Aufrufdefinition

Rückgabedefinition

Zusicherungen

Funktionsrumpf

Externe tabellenwertige Funktion

- Funktionsprototyp wird in SQL definiert
- Funktion selbst in anderer Sprache geschrieben

```
CREATE FUNCTION BlockStatus()  
  RETURNS TABLE (  
    Block_Nummer BYTE,  
    Leistung_KW INTEGER,  
    Temperatur_C DECIMAL(3,2),  
    FehlerCode INTEGER)  
  NOT DETERMINISTIC  
  NO SQL  
  LANGUAGE JAVA  
  EXTERNAL  
  PARAMETER STYLE SQL;
```

Multimengen in SQL:2003

- Arrays haben feste Ordnung und feste Größe
- Multimengen sind Arrays ähnlich
- Multimengen haben jedoch keine feste Ordnung und beliebige Größe
- lassen sich von jedem Datentyp erstellen
- Definition mittels Schlüsselwort **MULTISET**

```
CREATE TABLE Studenten (  
    Name VARCHAR(15),  
    ...  
    Sprachen VARCHAR(15) MULTISET);
```

Funktionen für Multimengen

- Konstruktor `MULTISET(e1,e2,...,en)`:
Erzeugt eine Multimenge durch Aufzählung:
`UPDATE Studenten Sprachen=MULTISET('En','De','Fr')`
- Konstruktor `MULTISET(SELECT ...)`:
Erzeugt eine Multimenge per Abfrage, bei mehr als einer Spalte jedes Element vom Typ ROW
`UPDATE Studenten SET Sprachen=MULTISET(SELECT * FROM Sprachen)`
- `COLLECT` erzeugt Multimengen aus einer Spalte
- `FUSION` und `INTERSECTION` vereinigen oder schneiden Multimengen als Aggregatfunktion
- daneben gängige Operatoren wie `UNION`, `EXCEPT`, `INTERSECT`, `MEMBER OF`
- `IS A SET` prüft auf Duplikate, Funktion `SET` entfernt Duplikate
- tabellenwertige Funktion `UNNEST` „entpackt“ Multimengen zu einer Tabelle

[SQL/XML]

- völlig neues Kapitel im Standard, sog. „Part 14“
- Funktionen zum Import und Export zwischen SQL und XML

Aufbau eines XML-Dokuments

```
<?xml version='1.0' encoding='UTF-8'?>
<Studenten>
  <Student MatrikelNummer="185934">
    <Vorname>Eliza</Vorname>
    <Name>Day</Name>
    <Anschriften>
      <Anschrift>Rosengasse 7, 53117 Bonn</Anschrift>
    </Anschriften>
  </Student>
  <Student MatrikelNummer="193429">
    <Vorname>Hans</Vorname>
    <Name>Gl&uuml;ck</Name>
    <Anschriften>
      <Anschrift>Goldgasse 9, 51345 Much</Anschrift>
    </Anschriften>
  </Student>
</Studenten>
<!--Hier k&ouml;nnte die Datei enden-->
```

Vorspann

Element

Elementattribut

Elementinhalt

Kommentar

Entitätsreferenz

Der Datentyp XML und XQuery

- neuer Basisdatentyp XML zum Abspeichern von XML-Daten
- Erzeugung aus String mittels XMLPARSE

XQuery

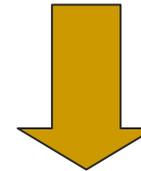
- XML-Anfragesprache
- Ausführung in SQL mittels XMLGEN

```
for $b in document („studenten.xml“)/Studenten/Student
where
    data($b/@MatrikelNummer) lt 190000
return
    <Student Vorname="{data($b/Vorname)}">...</Student>
```

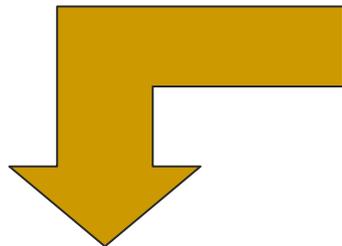
Funktionen zum Erzeugen von XML

- XMLAGG aggregiert SQL-Werte zum XML-Wald

MatrNr	Vorname	Name
185934	Eliza	Day
193429	Hans	Glück
184158	Doris	Day



```
SELECT Nachname, XMLAGG(Vorname) AS XMLGruppe FROM Studenten GROUP BY Nachname
```



Nachname	XMLGruppe
Day	<VORNAME>Eliza</VORNAME><VORNAME>Doris</VORNAME>
Glück	<VORNAME>HANS</VORNAME>

SQL:2003 - Zusammenfassung

- Identitätsspalten stellen automatisch hochgezählte Werte bereit
 - Sequenzgeneratoren zählen Werte bei Aufruf hoch
 - generierte Spalten speichern Werte, die sich aus anderen Spalten berechnen lassen
 - Merging fügt zwei Tabellen mit Sonderbehandlung schon vorhandener Datensätze zusammen
 - tabellenwertige Funktionen geben ganze Tabellen aus vordefinierten Abfragen oder externen Programmen zurück
 - Multimengen speichern eine beliebige Anzahl von Werten mit Duplikaten
 - SQL/XML ermöglicht eine Verknüpfung von XML und SQL
- ...und noch viele kleinere Neuerungen in SQL:2003, mehr dazu in der Ausarbeitung und der angegebenen Literatur