

Praktikum Datenbanksysteme

an der

Universität Trier

Leitung: Prof. Bernd Walter, Dipl. Inf. Alexander Weber

Arbeiten mit IBM[®] DB2[®]

Abschlußbericht zum Praktikum (SS 2002)

Jörg Follmann, Matr. Nr.: 601561

1 Inhaltsverzeichnis zum Abschlussbericht

<u>1 INHALTSVERZEICHNIS ZUM ABSCHLUSSBERICHT</u>	<u>2</u>
<u>2 AUFGABENSTELLUNG</u>	<u>3</u>
<u>3 SYSTEMVORAUSSETZUNGEN.....</u>	<u>3</u>
3.1 HARDWARE.....	3
3.2 SOFTWARE	4
<u>4 ARBEITEN MIT DB2</u>	<u>4</u>
4.1 INSTALLATION.....	4
4.2 EINFÜGEN DER TESTDATEN.....	5
4.3 BEISPIEL – ABFRAGEN	6
4.4 ERGEBNISSE UND FOLGERUNGEN AUS DEN ABFRAGEN	8
4.4.1 ALLGEMEINES ZUM DB2® ZUGRIFFSPLAN	8
4.4.2 BEISPIEL EINES ERSTELLTEN ZUGRIFFSPLANS	8
4.4.3 WEITERE ERGEBNISSE.....	10
<u>5 XML – EXTENDER.....</u>	<u>10</u>
5.1 INSTALLATION UND EINRICHTEN DES XML – EXTENDERS	11
5.2 ARBEIT MIT DEM XML – EXTENDER.....	13
<u>6 FAZIT</u>	<u>14</u>
<u>7 ABBILDUNGS – UND QUELLENVERZEICHNIS</u>	<u>15</u>

2 Aufgabenstellung

Das Ziel des Praktikums ist es, den Umgang mit dem System IBM® DB2® zu erlernen und dabei unterschiedliche Messungen vorzunehmen. Die erzielten Ergebnisse der praktischen Arbeit sind in diesem Bericht festgehalten.

Detaillierter sind folgende inhaltliche Punkte gefordert:

- (1) Installation von IBM® DB2® auf einem bereitgestellten Testsystem. Seine Spezifikation findet sich im Kapitel 3. Hierbei sollen insbesondere Auffälligkeiten und entstehende Probleme während der Installation untersucht werden. Diese sind in Kapitel 4.1 beschrieben.
- (2) Einrichten einer einfachen Beispiel – Datenbank, hier eine mit zufälligen Einträgen gefüllte Datenbank „UNIDB“, näheres zur Struktur und Vorgehensweise siehe Kapitel 4.2.
- (3) Zeitmessungen von unterschiedlichen Abfragen in der erstellten Beispiel – Datenbank. Die Vorgehensweise hierzu und die Beispiele werden in Kapitel 4.3 und Kapitel 4.4 vorgestellt.
- (4) Installation des XML – Extenders für DB2®. Auch hier sind Probleme bei der Installation zu untersuchen und erste Versuche mit selbst gebastelten Beispiel – Datenbanken zum Kennen lernen des Systems zu führen.
- (5) Einlesen der XML – Datei der DBLP¹ – Datenbank in eine selbst zu erstellende Datenbank.
Kapitel 5 beschäftigt sich mit den bei der Arbeit mit dem XML – Extender erzielten Ergebnissen.

3 Systemvoraussetzungen

3.1 Hardware

Der zur Verfügung stehende Rechner hat folgende Ausstattung:

¹ (DataBase systems and Logic Programming) Internet – Bibliographie - Suchmaschine an der Universität Trier: <http://dblp.uni-trier.de/>

- Intel Pentium III 500 MHz
- 256 KB Arbeitsspeicher
- Festplatte Fujitsu MPD 3130 AT, 13GB, 5400 U/min²

und das zur Installation notwendige CD – Rom – Laufwerk.

3.2 Software

Als Betriebssystem auf obigem Rechner wird SuSE Linux Version 8.0 verwendet. Alle Installationsroutinen, Datenbankoperationen und Messungen werden unter KDE Version 3.0 durchgeführt. Für die Erstellung einfacher Programme zur Abfrage oder Einfügen von Daten in die Datenbank wird im Laufe des Praktikums auf kleinere JAVATM – Programme zurückgegriffen. Hierfür reicht die installierte Version SDK 1.2 aus. Die verwendete Programme sind diesem Bericht auf einer Diskette im Anhang beigelegt. Für die Ausführung der DB2[®] – Steuerzentrale ist die Installation der IBM eigenen JAVA – Version IBM JAVA118 – SDK – 1.1.8-5.0 notwendig.

4 Arbeiten mit DB2

4.1 Installation

Die Installation von IBM[®] DB2[®] in der ausgelieferten Version 7.1 scheiterte. Das System meldete Speicherzugriffsfehler (Segmentation fault), die nicht behoben werden konnten. Die Installation von IBM[®] DB2[®] Version 7.1.0-40 verlief dagegen problemlos. Das Problem ist bei SuSE³ bekannt. Ursache ist ein Fehler in der Installationsroutine von IBM[®] DB2[®]. Diese nutzt ein zu kleines Array zum Auslesen der Datei „/etc/services“ des Betriebssystems.

Das Problem kann behoben werden, indem die Datei „/etc/services“ vor der Installation durch eine kleinere Datei ersetzt wird und dieser Schritt nach der Installation wieder rückgängig gemacht wird. Dann müssen in der „alten“ „/ect/services“ die von der Installationsroutine von IBM[®] DB2[®] hinzugefügten Services ergänzt werden. Eine genaue Beschreibung des Verfahrens ist auf der angegebenen Web – Site zu finden.

² Quelle[3]

³ http://sdb.suse.de/de/sdb/html/db2_suli8.html

4.2 Einfügen der Testdaten

Mit Hilfe von Programm 1 der Anlage⁴ wurde eine zuvor angelegte Beispiel – Datenbank „UNIDB“ mit zufälligen Daten gefüllt. An Hand dieser Datenbank wurden die in Kapitel 4.3 dokumentierten Abfragen auf ihre zeitlichen Parameter getestet. Das Schema ist folgender UML – Spezifikation zu entnehmen:



Abb. 1: UML – Spezifikation der Datenbank UNIDB.

Folgende Tabelle zeigt die Anzahl der Datensätze pro Tabelle:

⁴ Source – Code: Dbase2.java im Verzeichnis Programm1 auf der beigelegten Diskette

TABELLENNAME	ENTHALTENE DATENSÄTZE
ASSISTENTEN	10.000
HOEREN	49.982
PROFESSOREN	10.000
PRUEFEN	8.812
STUDENTEN	10.000
VORLESUNGEN	10.000
VORLESUNGEN	10.000

Tabelle 1: Anzahl der enthaltenen Datensätze

4.3 Beispiel – Abfragen

Zur Untersuchung des Laufzeitverhaltens von DB2[®] wurden beispielhaft 8 Abfragen mit Hilfe des DB2[®] – Visual Explain, als Tool integriert in die Steuerzentrale erstellt.

LFD. NR.	ABFRAGE / ERLÄUTERUNG	BERECHNETE DAUER ⁵
1	<pre>SELECT Q3.NAME AS "NAME" FROM DB2INST1.VORLESUNGEN AS Q1, DB2INST1.HOEREN AS Q2, DB2INST1.STUDENTEN AS Q3 WHERE (Q1.TITEL = 'vor11212') AND (Q3.MATRNR = Q2.MATRNR) AND (Q2.VORLNR = Q1.VORLNR)</pre>	
	Suche alle Studentennamen, die die Vorlesung „vor11212“ hören	1.044,81
2	<pre>SELECT Q3.TITEL AS "TITEL", Q1.TITEL AS "TITEL" FROM DB2INST1.VORLESUNGEN AS Q1, DB2INST1.VORLESUNGEN AS Q3 WHERE (Q2.VORGAENGER = Q1.VORLNR) AND (Q3.VORLNR = Q2.NACHFOLGER)</pre>	
	Suche alle direkten Vorgänger aller Vorlesungen	1.531,17
3	<pre>SELECT DISTINCT Q8.TITEL AS "TITEL" FROM DB2INST1.VORLESUNGEN AS Q1, (SELECT Q3.\$C0, Q2.NACHFOLGER FROM DB2INST1.VORLESUNGEN AS Q2, Q7 AS Q3</pre>	

⁵ Die Angabe der Dauer bezieht sich auf die in Visual Explain verwendeten virtuellen Kosten. Siehe hierzu auch Kapitel 4.4.1

	<pre>WHERE (Q3.\$C1 = Q2.VORGAENGER) UNION ALL SELECT Q5.VORGAENGER, Q5.NACHFOLGER FROM DB2INST1.VORAUSSETZEN AS Q5) AS Q7, DB2INST1.VORLESUNGEN AS Q8 WHERE (Q1.TITEL = 'vorl101') AND (Q7.\$C1 = Q1.VORLNR) AND (Q8.VORLNR = Q7.\$C0)</pre>	
	<p>Alle Vorgänger der Vorlesung „vorl101“ bestimmen (suche über TITEL)</p>	10.121,93
4/5	<pre>SELECT Q1.NAME AS "NAME" FROM DB2INST1.STUDENTEN AS Q1 WHERE (9 <= Q1.SEMESTER) [ORDER BY Q1.SEMESTER]</pre>	
	<p>Alle Studenten, die mindestens im 9. Semester sind [geordnet nach Semester]</p>	326,61 [329,1]
6	<pre>SELECT Q3.NAME AS "NAME" FROM DB2INST1.VORLESUNGEN AS Q1, DB2INST1.HOEREN AS Q2, DB2INST1.STUDENTEN AS Q3 WHERE (Q1.TITEL = 'vorl3425') AND (Q2.VORLNR = Q1.VORLNR) AND (Q3.MATRNR = Q2.MATRNR) AND (9 <= Q3.SEMESTER)</pre>	
	<p>Alle Studenten, die mindestens im 9. Semester sind und Vorlesung „vorl3425“ hören</p>	1.794,19
7	<pre>SELECT Q2.NAME AS "NAME" FROM DB2INST1.PRUEFEN AS Q1, DB2INST1.STUDENTEN AS Q2 WHERE (Q2.MATRNR = Q1.MATRNR) AND (Q1.NOTE = 1)</pre>	
	<p>Alle Namen der Studenten, deren Note 1 in einer Prüfung war</p>	571,14
8	<pre>UPDATE STUDENTEN AS Q1 SET Q1.SEMESTER = Q1.SEMESTER + 1</pre>	
	<p>Alle Studenten werden um ein Semester höher gesetzt.</p>	127.987,08

Tabelle 2: Beispiel – Abfragen

Alle Screenshots zu den aufgeführten Abfragen findet der interessierte Leser in der Anlage auf Diskette⁶. Als Beispiel wird im nächsten Kapitel der Graphen zu Abfrage 3 erläutert, der alle wesentlichen Elemente enthält.

4.4 Ergebnisse und Folgerungen aus den Abfragen

Zur Erstellung dieses Kapitels wurde auf Quelle[1] zurückgegriffen.

4.4.1 Allgemeines zum DB2[®] Zugriffsplan

Bei der Verwendung des „Explain – Tools“ werden virtuelle Kosten ermittelt. Sie setzen sich aus den geschätzten Zeiten der CPU, sowie denen für notwendige I/O – Operationen zusammen. Diese virtuellen Kosten lassen eine Abschätzung der Komplexität unterschiedlicher Abfragen gegeneinander zu, so dass sich in den in dieser Arbeit aufgeführten Untersuchungen zeitliche Vergleiche jeweils auf diese Einheit beziehen.

4.4.2 Beispiel eines erstellten Zugriffsplans

Der nachfolgende Zugriffsplan bezieht sich auf Beispiel 3 der Tabelle 2 in Kapitel 4.3.

⁶ Alle Screenshots finden sich im Verzeichnis Screenshots im pdf – Format auf der beigelegten Diskette. Der Name der jeweiligen Datei setzt sich dabei aus „bsp“ gefolgt von der in der Tabelle verwendeten Nummer zusammen.

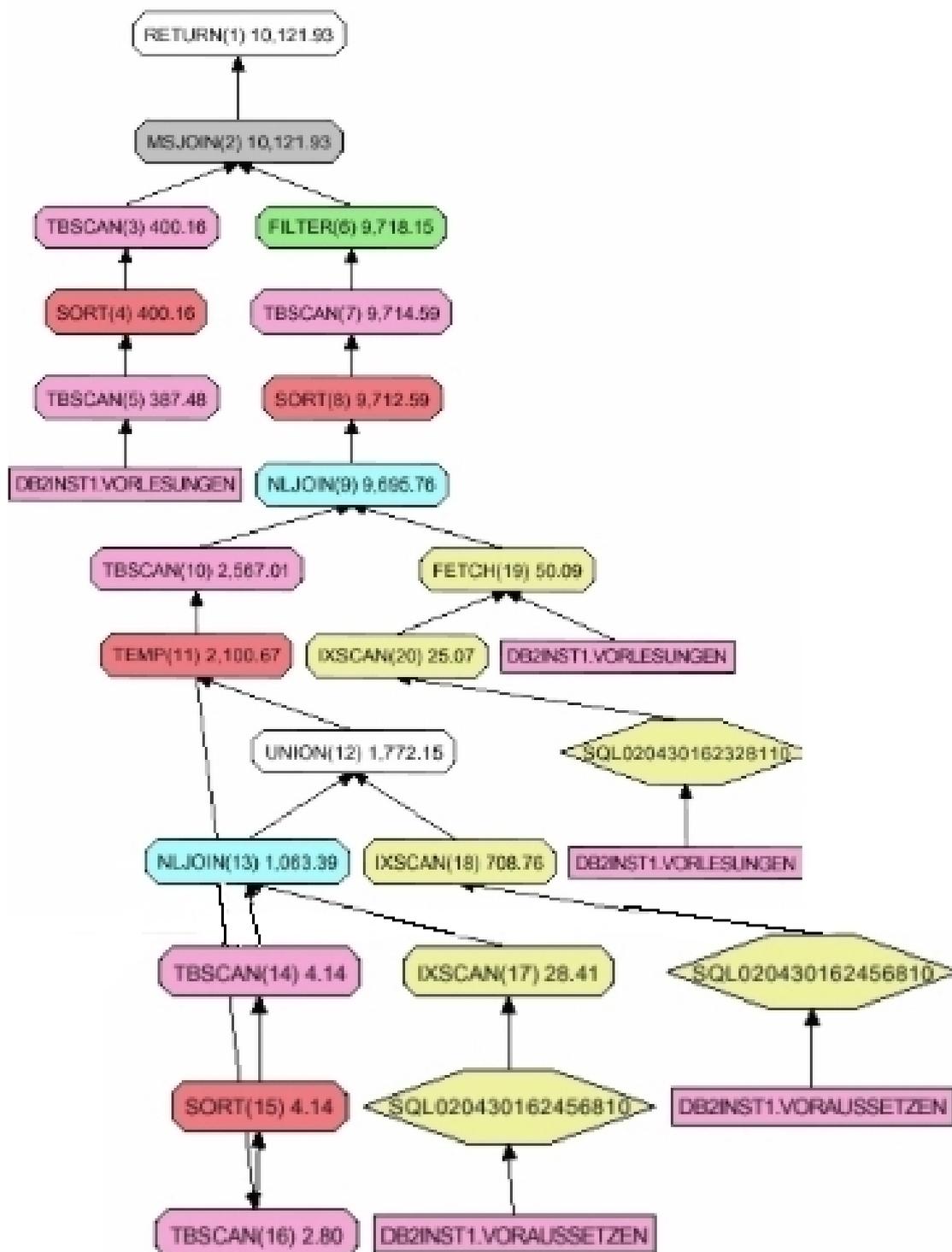


Abb. 2: DB2® – Zugriffssplan erstellt mit DB2® – Visual Explain

Am Beispiel obiger Abbildung sollen einige Elemente des Zugriffssplans kurz vorgestellt und ihre Funktion gegeneinander abgegrenzt werden.

Die beiden wichtigsten und teuersten Funktionen sind die „Scan“ – Funktionen „TBSCAN“ [Direktes Lesen der Daten] und „IXSCAN“ [Indexsuche]. Die erste Methode

wird bevorzugt verwendet. Der Datenzugriff über „IXSCAN“ erfolgt nur falls Indexsuche nicht aufwendiger ist als das Lesen der gesamten Tabelle. Bei kleinen Tabellen oder Verwendung eines Großteils der Tabelle verwendet das DB2® – Optimierungsprogramm für Abfragen „TBSCAN“.

Die Abfrage – Operationen optimieren ihre Ergebnisse für weitere folgende Operationen. Für die Betrachtung des „SORT“ – Vorgangs soll hier nur auf Kapitel 4.4.3 verwiesen werden.

Auch die in der Theorie als teuer bekannten Operationen „JOIN“ und „UNION“ werden von den „Scan“ – Funktionen vorbereitet. Deshalb sind „MSJOIN“ [Merge und Scan – Join; Hierfür müssen die zu verbindenden Tabellen bereits anhand des Element über das die „JOIN“ – Operation ausgeführt wird sortiert sein. Die Elemente der äußeren Tabelle werden mit denen der inneren Tabelle verglichen. Analog zu „Merge – Sort“ genügt ein Durchlauf für die Misch – Funktion.] und „UNION“ fast ohne zusätzliche Kosten – Einheiten zu berechnen (siehe Operationen 2,12 aus Abbildung 2).

Für die Berechnung von „NLJOIN“ [Nested Loop Join; verschachtelte Suche. Die innere Tabelle wird für jedes Element der äußeren Tabelle einmal durchsucht] ist in ihren Kosten viel teurer als „MSJOIN“.

4.4.3 Weitere Ergebnisse

Aus dem Vergleich der Kosten aus den Abfragen 4 und 5 lässt sich die Effizienz der Abfragen mittels DB2® erkennen. Obwohl das Ergebnis in Abfrage 5 zusätzlich sortiert wird, erhöhen sich die Kosten für die Ausführung dieses Statements nur gering im Vergleich zum unsortierten Ergebnis. Die Abfrage – Operationen bereiten also eine schnelle Sortierung nach dem Schlüssel „SEMESTER“ bereits vor.

Update – Operationen benötigen, was wegen der notwendigen Änderung des Datenbestands auf der Festplatte als logisch und hardware – bedingt gesehen werden muss, viel mehr Zeit als Abfrage – Anfragen.

5 XML – Extender

Zum Umgang mit dem XML – Extender wurde auf Quelle [2] zurückgegriffen.

5.1 Installation und Einrichten des XML – Extenders

Die Installation des XML – Extenders Version 7.2.0-1 verlief problemlos bei Verwendung der rpm – Routine. Zum Test des XML – Extenders mittels der DBLP – Daten wurden zunächst folgende Schritte zur Einrichtung einer entsprechenden Datenbank durchgeführt:

1. Die Datenbank „xmldata“ neu erstellen.
2. Die neue Datenbank muss nun zur Benutzung der XML – Routinen und – Datenstrukturen mittels Eingabe der folgenden Zeilen gebunden und aktiviert werden werden:

```
db2 connect to xmldata
cd $HOME/dxx/bnd
db2 bind @dxxbind.lst
cd $HOME/sqllib/bnd
db2 bind @db2cli.lst
dxxadm enable_db xmldata
```

3. Die zum DBLP gehörende DTD – Datei „/home/db2inst1/db2inst1/dblp.dtd“ enthält die DTD – Spezifikation für die gespeicherten Daten. (Die befindet sich auf der beigelegten Diskette im Verzeichnis XMLExtender⁷) Zum Einfügen der Daten in die Datenbank muss die DTD im DTD - Repository gespeichert werden. Dies erledigt folgendes Kommando[db2inst1 ist der angemeldete Benutzer des Systems]:

```
insert into db2xml.dtd_ref values('/home/db2inst1/db2inst1/dblp.dtd',
db2xml.XMLClobFromFile('/home/db2inst1/db2inst1/dblp.dtd'),0,
'db2inst1', 'db2inst1', 'db2inst1')
```

4. Die Spezifikation der zu übernehmenden Daten erfolgt in einer DAD – Datei. Die ist zum jetzigen Zeitpunkt zu erstellen. Die in diesem Fall verwendete DAD – Datei („dblp.dad“) findet sich auf der beigelegten Diskette⁸.
5. Anschließend muss eine XML - Tabelle erstellt werden, in die die Daten übergeben werden:

```
db2 „create table xml_files(„id“ integer not null“, „xmldata“
DB2XML.XMLFiLE, primary key (id))“
```

6. Nun müssen die Spalten der Tabelle für den XML – Extender aktiviert werden:

```
dxxadm enable_column xmldata xml_files xml_data getstart.dad
```

⁷ XMLExtender/dblp.dtd

Der XML – Extender legt durch diesen Befehl automatisch die benötigten und in der DAD – Datei spezifizierten Seitentabellen, die im diesem Beispiel aus Abbildung 3 zu erkennen sind, an. Beim Import von Daten werden diese dann gefüllt.



Abb 3: Die Datenbank XMLDATA mit den vom XML – Extender erstellten Seitentabellen

⁸ XMLExtender/dblp.dad

5.2 Arbeit mit dem XML – Extender

Bevor in diesem Kapitel die Daten – Eingabe dokumentiert wird, zuerst noch einige allgemeine Bemerkungen zur Arbeitsweise. In diesem Fall ist das Arbeiten mit XML - Spalten, weil die XML – Dokumente bereits vorhanden sind, nicht aktualisiert, sondern nur geprüft werden, anstatt der Verwendung von XML – Objektgruppen, sinnvoll.

Da alle Daten in einer Datei gespeichert waren, und somit Argumente von XML – Extender nicht eindeutig zuzuordnen sind, wurde die XML – Datei in kleinere jeweils einen Datensatz enthaltene Dateien aufgeteilt (mittels Programm 2 aus der beigelegten Diskette⁹).

Die Spezifikation „multi_occurrence = YES“ in der DAD –Datei erzwingt das erstellen einer eigen Tabelle für dieses Merkmal, beispielsweise wird für die Autoren der Artikel hierfür die Tabelle „Authors_art“ bereitgestellt.

Die Daten wurden mit folgendem Shell – Script in die Datenbank übernommen:

```
#!/bin/bash
db2 connect to xmldata

for (( i=0; $i < 99999; i=$((i+1))
do
    db2 "insert into xml_files
        values('$i',('/home/db2inst1/db2inst1/xml_docs/$i.xml'),' $i'"
done

db2 disconnect xmldata
```

Beim Einlesen der Daten fiel folgender Fehler auf:

Traten Sonderzeichen wie ü in einem Datensatz auf, wurde der entsprechende String vor dem „&“ abgeschnitten und nur der Teil – String in die Datenbank übernommen. Dies galt für alle XML – Sonderzeichen beginnend mit „&“.

Für das Einlesen von 2310 Dokumenten benötigte das System 30 Minuten und 42 Sekunden. Aufgrund der zahlreichen Datenbank – und Festplatten – Operationen scheint dies aber gerechtfertigt. Die Datenbank läuft mit dieser Anzahl an Dokumenten jedoch reibungslos und unterstützte die durchgeführten Anfragen problemlos.

⁹ xml_parse.java im Ordner Programm2

6 Fazit

Das DB2[®] – Command – Center benötigt eine alte JAVA[™] – Version: IBM JAVA118 – SDK – 1.1.8-5.0. Diese ist inkompatibel zu den Implementierungen von SUN und muss vor dem Starten des Command – Centers in den Einstellungen des Linux – Systems gesetzt werden. Besonders beim Einsatz neuer JAVA – Programme führt dies zu unnötigem Arbeitsaufwand.

Das Eingeben von Statements an der Shell, bei gestartetem DB2[®] ist etwas umständlich. Das sonst in Linux übliche Kommando ↑ [Pfeil – nach – oben – Taste] zum Wiederholen des vorherigen Befehls funktioniert bei gestartetem DB2[®] nicht. Deshalb sollten Befehle an der Shell ohne den vorherigen Aufruf von DB2[®] ausgeführt werden, dann mit vorgestelltem *db2* „select ...“. Dies vereinfacht den Umgang mit Tippfehlern, da der gesamte Befehl nicht wieder neu getippt werden muss.

Besonders bei dem Umgang mit dem XML – Extender fallen die oft nicht eindeutigen Fehlermeldungen des SQL – Compilers auf.

DB2[®] empfiehlt sich für den Umgang und die Verwaltung großer Datenmengen. Braucht man nicht den vollen Funktionsumfang und reichen eingeschränkte Rechtevergaben aus, sollte man auch angesichts des Preises von 1690 €¹⁰ (Stand 1.10.2002) eher das kostenlose MySQL bevorzugen, dass für kleinere Anwendungen durchaus ausreicht und eine bessere Übersichtlichkeit in der Datenbank – Verwaltung bietet.

¹⁰ Preis für Suse Linux Database Server mit IBM[®] DB2[®] Universal Database – Workgroup Edition in deutscher Ausgabe. Siehe auch http://www.suse.de/de/business/products/suse_business/database_server/prices.html und http://www.suse.de/de/business/products/suse_business/database_server/content.html
Der angegebene Preis versteht sich in Deutschland zuzüglich 16% Mehrwertsteuer.

7 Abbildungs – und Quellenverzeichnis

- [1] Aus der Interpräsenz der FH Salzburg:
http://www.fh-sbg.ac.at/~ulamec/sql/sql_explain.htm

- [2] Unter dem folgenden Link kann die vom IBM[®] herausgegebene Dokumentation „XML Extender Administration and Programming in deutscher Sprache im pdf - Format heruntergeladen werden. Die Quelle findet der interessierte Leser unter dem Punkt Administration.
http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v7de_pubs.d2w/de_main

- [3] Daten des Hersteller Fujitsu:
<http://fdg.fujitsu.com/home/product.asp?L=de&PID=169>

Abbildungen:

- Abb. 1: UML – Spezifikation der Datenbank UNIDB

- Abb. 2: DB2[®] – Zugriffsplan erstellt mit DB2[®] – Visual Explain

- Abb. 3: Die Datenbank XMLDATA mit den vom XML – Extender erstellten Seitentabellen