

Studienarbeit

**Installation und Konfiguration
eines DB2Connect Enterprise
Edition Servers**

Katharina Denking

Inhaltsverzeichnis

Einleitung	3
1 Client/Server Architekturen	3
1.1 Client/Server 2-Tier	3
1.4 Client/Server 3-Tier / Multitier Architektur	4
2 Hardware und Betriebssysteme	5
2.1 Client	6
2.2 Middle-Tier (DB2Connect Server)	6
2.3 Host	6
Ein-/Ausgabe Architektur	6
OS/390-Betriebssystem	6
3 Connectoren	7
3.1 Allgemein	7
3.2 DB2Connect	8
3.2.1 Was ist DB2Connect?	8
3.2.2 Editionen im Überblick	9
3.2.3 DB2Connect Personal Edition (DB2Connect PE)	9
3.2.4 DB2Connect Enterprise Edition (DB2Connect EE)	10
3.2.5 DB2 Connect Application Server Edition (DB2 Connect ASE)	12
3.2.6 DB2Connect Unlimited Edition(DB2 Connect UE)	12
3.2.7 Softwarepaket Informationen	13
3.3 HiT JDBC Treiber	13
3.4 Fazit	14
4 Installation	15
4.1 Software	15
4.1.1 Installation von DB2Connect EE	15
4.1.2 Konfiguration von DB2Connect EE	17
5 Anwendungsbeispiel	20
5.1 TestInstall.java	20
5.1 JDBCTestPrg.java	22
6 Literaturverzeichnis	24

Einleitung

Diese Studienarbeit befasst sich mit der Installation und Konfiguration eines DB2Connect Servers. Dieser Server soll im Rahmen des Client/Server Praktikums für Zugriffe auf eine DB2 Datenbank auf einem OS/390 Großrechner verwendet werden. Der OS/390 Server ist unter jedi.informatik.uni-leipzig.de im Internet zu erreichen.

Die Software sollte nach folgenden Kriterien ausgewählt werden:

- Erreichbarkeit der DB2 Datenbank von verschiedenen Betriebssystemen aus (min. Solaris und Windows)
- Keine zusätzliche Software Installation auf den Arbeitsplatzrechnern ist notwendig

Durch die oben genannten Auswahlkriterien ist die Verwendung einer 3-Tier Umgebung die beste Wahl. Deshalb werden in Kapitel 1 die Merkmale einer 2- bzw. 3-Tier Umgebung erläutert.

Die verwendete Hardware und deren Betriebssysteme werden in Kapitel 2 beschrieben.

In Kapitel 3 werden die grundlegenden Eigenschaften von Connectoren und die verschiedenen Connector-Typen erklärt. Dabei wird insbesondere auf die verschiedenen DB2Connect Versionen eingegangen.

In Kapitel 4 wird die Installation und Konfiguration der ausgewählten Software erläutert.

Anhand zweier Anwendungsbeispiele wird in Kapitel 5 die Installation und Konfiguration verifiziert.

Das Literaturverzeichnis befindet sich in Kapitel 6.

1 Client/Server Architekturen

Hier wird eine Übersicht über die 2- und 3-Tier Client/Server Architekturen gegeben, da sie bei der Wahl der geeigneten Software eine wichtige Rolle spielen.

Weiterführende Informationen findet man unter [CSA] und [SPR].

1.1 Client/Server 2-Tier.

Heute trennt man den Anwendungsprozess in zwei Komponenten [SPR]:

- Die *Anwendungs-/Business-Logik* verarbeitet die Eingabedaten und erzeugt Ausgabedaten für den Endbenutzer.
- Die *Präsentations-Logik* übernimmt diese Datenstruktur und setzt sie in eine für den Endbenutzer ansprechende graphische Bildschirmdarstellung um.

Die Präsentations-Logik kann somit im hohen Maße konfiguriert werden (Templates, Cascading Style Sheets), ohne dass es Auswirkungen auf den ganzen Entwicklungsprozess hat. Des weiteren wird so die Softwareentwicklung besser unterstützt, da die einzelnen Programmteile getrennt entwickelt und über vordefinierte Schnittstellen verbunden werden können. Die Präsentationslogik kann durch Grafiker und Designer benutzerfreundlich gestaltet werden. Für die Intelligenz der Applikation werden dagegen erfahrene Programmierer eingesetzt.

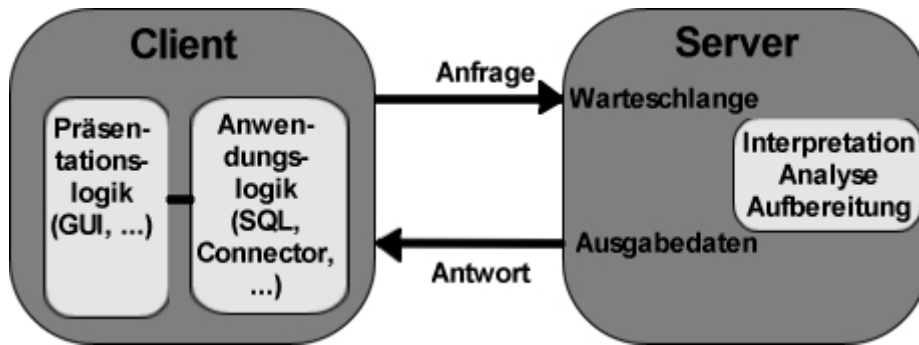


Abbildung 1 - 2-Tier Host Client Verbindung

Eine wie in Abbildung 1 dargestellte Architektur kann nur unter bestimmten Voraussetzungen eingesetzt werden [SPR]:

- einfache Transaktionen,
- < 200 Klienten (bei komplexen Transaktionen evtl. nur < 10 Klienten),
- < 100 000 Transaktionen / Tag (deutlich weniger bei komplexen Transaktionen),
- LAN-Umgebung,
- einen oder wenige Server,
- mäßige Sicherheitsanforderungen.

Die Nachteile überwiegen stark, sobald mehr Rechner angebunden werden:

- lokale Installation der Anwendung auf jedem Client bzw. Einsatz eines File-Servers
- höheres Datenvolumen im Netz als bei einer 3-Tier Architektur
- geringe Sicherheitsvorkehrungen

1.4 Client/Server 3-Tier / Multitier Architektur

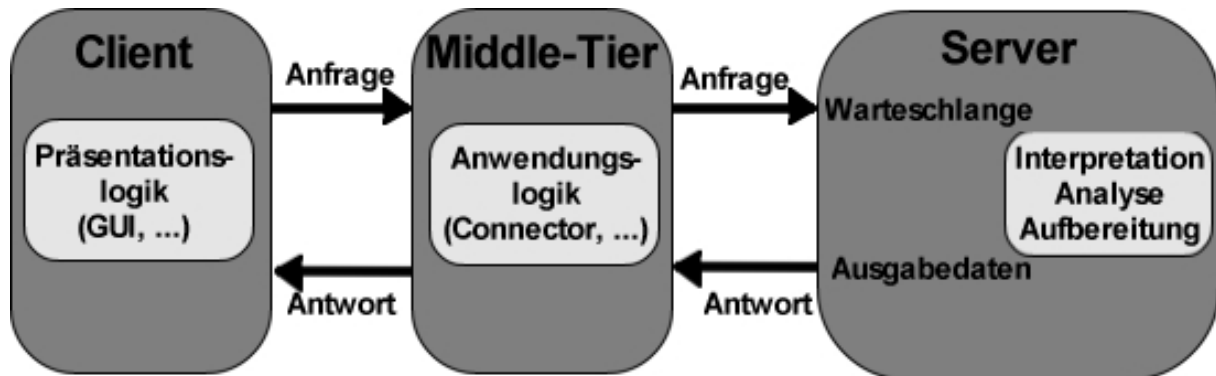


Abbildung 2 - 3-Tier Host Client Verbindung

Die 3-Tier Client/Server Architektur trennt streng die Präsentations- und Anwendungs-Logik und die Datenhaltung (s. Abbildung 2).

Der Client, auch als Frontend bezeichnet, führt die Präsentations-Logik aus. Es handelt sich dabei nicht zwingend um einen PC. Beispiele für weitere Clients wären PDAs und Mobiltelefone. Wichtige Größen sind zum einen die Geschwindigkeit und zum anderen die Stabilität des Clients. Die Präsentationslogik präsentiert die Daten z.B. in Form einer GUI oder eines Browsers.

Sowohl das Middle-Tier als auch der Server, auch als Backend bezeichnet, müssen über eine hohe Skalierbarkeit, Performance und Ausfallsicherheit verfügen. Bei Backendsystemen, auf denen die Daten in Form von Datenbanken zur Verfügung stehen, wird dies oft durch Cluster und/oder Hochverfügbarkeitssoftware erreicht.

Das Middle-Tier führt die Anwendungslogik aus. Für ausreichende Performance ist hierbei eine hohe Rechenleistung erforderlich. Bei Backendsystemen dagegen ist vor allem die I/O-Leistung ausschlaggebend.

Die Vorteile sind:

- nur eine Installation der Anwendungslogik pro Middle-Tier (auf den mehrere Clients zugreifen können).
- Geringeres Datenvolumen im Netz.
- Das Middle-Tier kann als Gateway fungieren.
- Die Zugriffe können deutlich besser kontrolliert werden.

Zusammenfassend kann man sagen, dass die 3-Tier Architektur eine sehr robuste und flexible Lösung darstellt.

Die Nachteile sind:

- üblicherweise hohe Softwarekosten.
- höherer Programmieraufwand.

Sobald die Anforderungen die Performancegrenzen einer 2-Tier Architektur sprengen, ist der Mehraufwand für eine 3-Tier Architektur unabdingbar. Die hohen Kosten werden durch die verbesserte Performance wieder ausgeglichen.

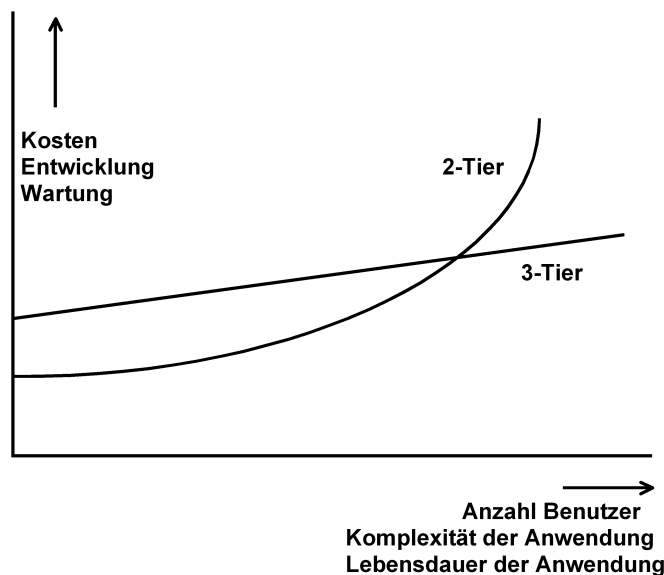


Abbildung 3 - 3-Tier vs. 2-Tier Host Client Verbindung

2 Hardware und Betriebssysteme

Dieses Kapitel dient einer kurzen Übersicht über die verwendete Hardware inklusive der darauf installierten Betriebssysteme.

Es wurde eine 3-Tier Architektur verwendet. Folglich muss zwischen dem Client, der ein Unix oder Windows Rechner sein kann, dem Middle-Tier welcher als DB2Connect Server unter Win2000 läuft und dem Host, einem OS/390 Mainframe, unterschieden werden.

Die Abschnitte zum Client und dem Middle-Tier sind im Vergleich zu dem Abschnitt über den Host deutlich kürzer, da zu Windows und Unix Rechnern ausführliche Literatur existiert.

2.1 Client

Beim Client ist man nicht auf ein bestimmtes Betriebssystem festgelegt. Er muss nur eine Internetverbindung zu dem DB2Connect Server erstellen und entsprechende Datenbankabfragen formulieren können.

In dem Client/Server Praktikum werden zum einen die Solarisrechner aus dem Rechnerpool und zum anderen private Rechner von Studenten, die von zu Hause aus arbeiten, verwendet. Auf diesen Rechnern werden mit eclipse (eine Entwicklungsumgebung für Java [ECL]) Java Servlets geschrieben, die über den DB2Connect Server Anfragen an die DB2 Datenbank stellen.

2.2 Middle-Tier (DB2Connect Server)

Der DB2Connect Server ist ein herkömmlicher Windows 2000 Server. Er steht an der Universität Leipzig und ist unter yoda.informatik.uni-leipzig.de erreichbar. Auf ihm wurde über einen VNC Client (Virtual Network Computing) die DB2Connect Enterprise Edition installiert. Die Installationen und Konfigurationen des Rechners sind das zentrale Thema dieser Arbeit und werden in Kapitel 4 beschrieben.

2.3 Host

Der Host ist eine S390 unter OS/390 in Leipzig. Auf ihm läuft eine DB2 Datenbank die unter anderem für den Praktikumbetrieb verwendet wird. Um sich noch umfassend darüber zu informieren kann man in [SPR] nachlesen.

Ein-/Ausgabe Architektur

S/390-Rechner und somit auch die zSeries, verfügen über eine überdurchschnittliche Ein-/Ausgabe-Leistung.

Bei Zugriffen auf Datenbanken und bei der Transaktionsverarbeitung wird das Leistungsverhalten eines Rechners nicht vorrangig durch die Leistungsfähigkeit der CPU bestimmt, sondern wie effektiv der Zugriff auf die gespeicherten Daten erfolgen kann.

Ein weiterer großer Vorteil der S/390-Ein-/Ausgabe-Ansteuerung ist, dass sie einem strengen Architektur-Regelwerk unterliegt wodurch diese Rechner sehr kompatibel mit anderen Rechnerarten sind.

OS/390-Betriebssystem

OS/390 stellt ein sehr komplexes Betriebssystem mit einer Vielzahl von Software-Komponenten dar. Es handelt sich hierbei um ein Server-Betriebssystem, das für den Multi User-Betrieb optimiert ist.

Auf diesem Betriebssystem läuft das OS/390 (objekt-) relationale Datenbank-Produkt DB2 Universal Database (auf dem Server in der Version 5 installiert). Es ist in einer identischen Implementierung für alle UNIX-, Linux- und Windows-Betriebssysteme verfügbar.

DB2 ist eine Server-Anwendung, die grundsätzlich in einem getrennten Adressraum läuft. Wie bei allen Server-Anwendungen ist ein Klient erforderlich, um auf einen DB2-Server zuzugreifen. Der Zugriff kann mit Hilfe von SQL-Statements erfolgen, die z.B. in einem Java-Anwendungsprogramm oder einem Java-Servlet eingebettet sind.

3 Connectoren

3.1 Allgemein

Ein Connector ist ein Softwareprogramm über das (Web-) Anwendungen, das sogenannte Frontend, auf Backend-Applikationen, wie z.B. Datenbanken oder Transaktionsmonitore, zugreifen.

Die Wahl des Connectors wird zum einen vom verwendeten Front- bzw. Backend, zum anderen von Performance und Sicherheitsaspekten beeinflusst. Häufig muss auch anhand der vorhandenen Client/Server Architektur (2-Tier oder 3-Tier Konfiguration) der richtige Connector ausgewählt werden.

Connectoren werden üblicherweise in 3 verschiedene Typen aufgeteilt [SPR]:

API Connectoren verwenden kein Java. Sie sind seit längerer Zeit verfügbar und haben in der Regel eine schlechtere Portabilität und ein besseres Leistungsverhalten als die Connectoren der beiden anderen Klassen. Existierende OS/390-Beispiele sind: *CICS Web Support (CWS)*, *IMS TCP/IP OTMA Connection (IMS TOC)*, *Net.Data* sowie *CWS mit 3270 Bridge*.

Java Connectoren verwenden Java. Für sie liegen ebenfalls Erfahrungen im industriellen Einsatz vor. Für relationale Datenbanken existieren die *JDBC*- und *SQLJ*-Connectoren. CICS und IMS haben das *CICS Transaction Gateway (CTG)* und den *IMS TOC Connector for Java* als Teil des *Common Connector Framework (CCF)* entwickelt. Dieses Framework wird von den WebSphere Java-Anwendungs-Entwicklungswerkzeugen unterstützt, spezifisch *VisualAge for Java. Host On-Demand* ist eine weitere Lösung für 3270-Anwendungen.

Enterprise JavaBeans Connectoren bauen auf der *Java 2*-Protokoll-Familie auf, z.B. *JMS*, *JDBC 2.0*, und stellen eine Evolution der Java-Connectoren dar. Sie verfügen über zusätzliche Transaktions- und Sicherheitseigenschaften auf der Thread-Ebene sowie Daten-Persistenz als Teil des Programmiermodells. Sie sind derzeit in der Entwicklung; eine Reihe offener Fragen müssen noch geklärt werden. Sie sind als Teil des *J2EE*-Standards in der *Java Connector Architecture (JCA)* Spezifikation definiert.

Java-Connectoren haben gegenüber API-Connectoren den Vorteil, dass sie sich nahtlos in ein Java orientiertes *Model-View-Controller-Konzept* einfügen. Beispielsweise sind folgende Java-Connectoren für die IBM WebSphere verfügbar:

- JDBC
- SQLJ
- DB2
- Oracle
- Adabas
- CICS
- IMS
- MQSeries
- SAP R/3
- Lotus Domino

3.2 DB2Connect

3.2.1 Was ist DB2Connect?

DB2Connect ermöglicht es, Applikationen auf eine schnelle, robuste und hoch skalierbare Weise Verbindungen zwischen Web, Windows, UNIX, Linux, Mobilien Applikationen und Datenbankdaten auf Mainframes herzustellen. Eine Alternative wäre der HiT-Treiber der Firma HiTSoftware, welcher zur Gruppe der Java Connectoren gehört.

Es kann auf einer Vielzahl von Betriebssystemen und Anwendungen operieren wie z.B. OS/390, z/OS, CICS, IMS (via DB2 stored procedures), DB2 Server für VSE & VM, AS/400 und iSeries u.v.m.. Der vollständige Support wird jedoch nur für IBM SQL, DB2 Universal Database für OS/390 und AS/400 und DB2 für MVS/ESA/ VSE & VM garantiert.

DB2Connect unterstützt fast alle APIs (Application Programming Interfaces) die den Zugriff auf Daten steuern. Beispiele hierfür sind JDBC, SQLJ, ODBC, OLE DB, DB2 CLI und Embedded SQL APIs. Über die Datenbankzugriffe hinaus werden auch Zugriffe auf nicht Datenbank Ressourcen wie CICS, WebSphere, IMS oder VSAM ermöglicht, was jedoch in dieser Studienarbeit nicht aufgegriffen wird.

Als Verbindungsprotokolle stehen TCP/IP und SNA zur Verfügung wobei TCP/IP stark empfohlen wird.

Die Einbindung der Distributed Relational Database Architektur (DRDA AR) in DB2Connect stellt eine bewährte Lösung dar. Bei DRDA handelt es sich um auf der OS/390 Seite verwendete Protokolle, die zur Verbindung zu Relationalen Datenbank Management Systemen verwendet wird. DB2Connect ist deutlich performanter als (Java-) Connectoren, wie z.B. dem HiT JDBC Connector.

DB2Connect benutzt das 2-Phasen-Commit um die Datenkonsistenz zu gewährleisten. Es kann sowohl für Applikationen mit SQL-Anfragen, als auch für solche, die einen Transaktionsmonitor benutzen, verwendet werden.

Ein weiteres Feature sind die verteilten Anfragen (Distributed Request). Sie ermöglichen es, SQL Statements auszuführen, die aus zwei oder mehreren Datenbank Management Systemen (DBMS) oder Datenbeständen stammen. Beispiele hierfür sind:

- Join zwischen Tabellen aus 2 verschiedenen DB2 Datenbanken für OS/390 Subsystemen.
- UNION Operation zwischen einer DB2Tabelle und einem Oracle View.

Diese verteilten Anfragen lösen somit das Problem, dass die DBMS nie alle SQL Dialekte und Anfrageoptimierungen unterstützen.

Umfassende Daten und Fakten findet man unter [IBM].

3.2.2 Editionen im Überblick

DB2Connect gibt es in 4 verschiedenen Versionen:

- DB2 Connect Personal Edition (DB2 Connect PE)
- DB2 Connect Enterprise Edition (DB2 Connect EE)
- DB2 Connect Application Server Edition (DB2 Connect ASE)
- DB2 Connect Unlimited Edition (DB2 Connect UE)

Abbildung 4 veranschaulicht, wie die Produkte miteinander in Verbindung stehen:

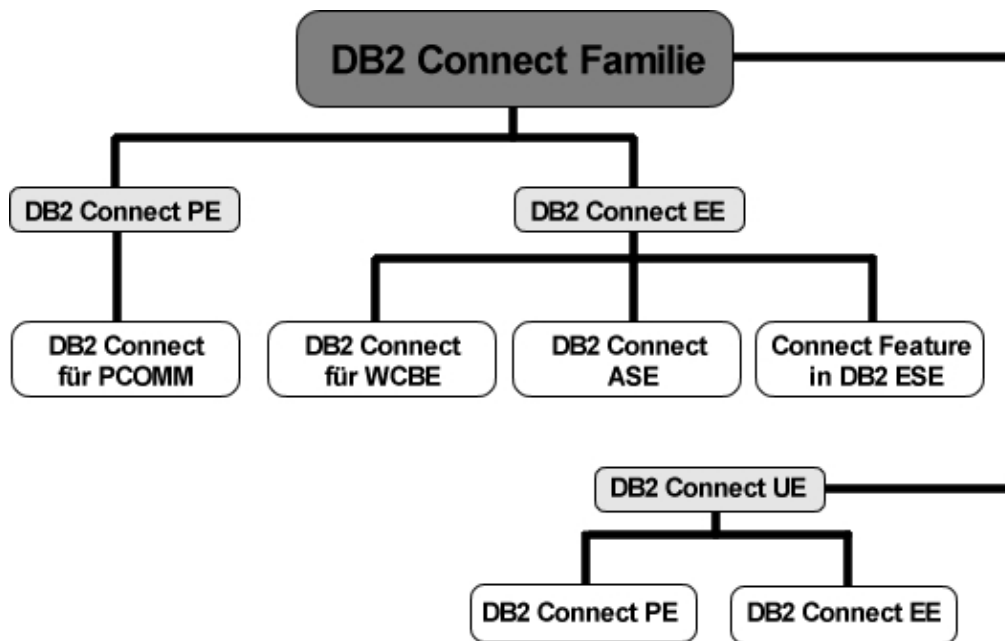


Abbildung 4 - DB2 Connect Familie

3.2.3 DB2Connect Personal Edition (DB2Connect PE)

Die Personal Edition [PED] erlaubt den Zugriff von einem Linux oder Windows PC über TCP/IP oder SNA auf Daten die auf Systemen wie z.B. DB2 für OS/390, MVS/ESA, VSE, VM, OS/400 oder DRDA kompatiblen Datenbank Servern gespeichert sind.

Dabei werden auch komplexe SQL Anfragen, die umfangreiche Ergebnisse liefern, performant beantwortet.

Die Installation erfolgt in einer 2-Tier-Architektur (s. Abbildung 1). Jedoch ist es in dieser Version nicht möglich DB2Connect als Server für mehrere Entwicklungsrechner zu verwenden.

Folgende Plattformen werden unterstützt:

Linux, Windows NT, Windows 98, Windows 2000, Me.

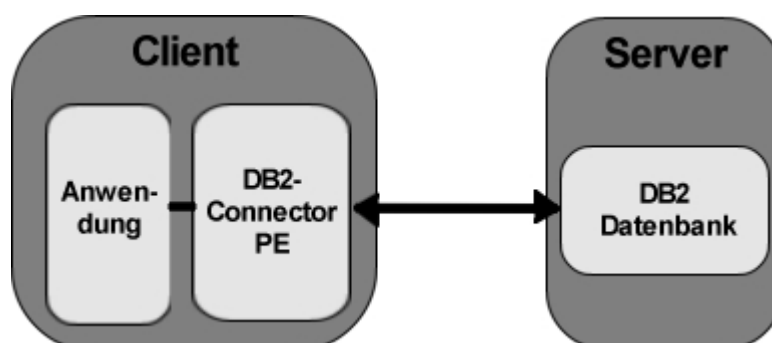


Abbildung 5 - DB2Connect PE Installation

Um die Software auf vielen Rechnern mit wenig Aufwand zu betreiben kann man DB2Connect PE auch auf einem File-Server installieren. Die Clients werden dann als *DB2 Thin-Connect Workstation* bezeichnet. Eine *DB2 Thin-Connect Workstation* ist eine spezielle Installation von DB2Connect PE in welcher der Code in einem gemeinsamen Netzlaufwerk auf einem File-Server, auch *DB2Connect Code Server* genannt, installiert ist.

Diese Methode bietet keinerlei Nachteile gegenüber der lokalen Installation. In Abbildung 6 ist eine *DB2 Thin-Connect Workstation* dargestellt, die eine Verbindung aufbaut und Daten herunterlädt.

Der Ablauf ist folgendermaßen:

- Der Client lädt die DB2Connect DLLs und anderen Code von einem Netzlaufwerk (anstatt von der lokalen Festplatte) herunter. Der gestrichelte Pfeil symbolisiert diesen Ladevorgang.
- Der Client öffnet eine Verbindung zum DB2 Server und holt sich die entsprechenden Daten (durchgezogene Pfeile). Der Code Server agiert dabei nicht als Gateway sondern stellt nur Plattenplatz zur Verfügung.
- Es muss sicher gestellt werden, dass nie mehr *DB2 Thin-Connect Workstations* aktiv sind als Lizenzen vorhanden sind.
- Ist die *DB2 Thin-Connect Workstation* fertig, gibt sie ihre Lizenz wieder für andere frei.

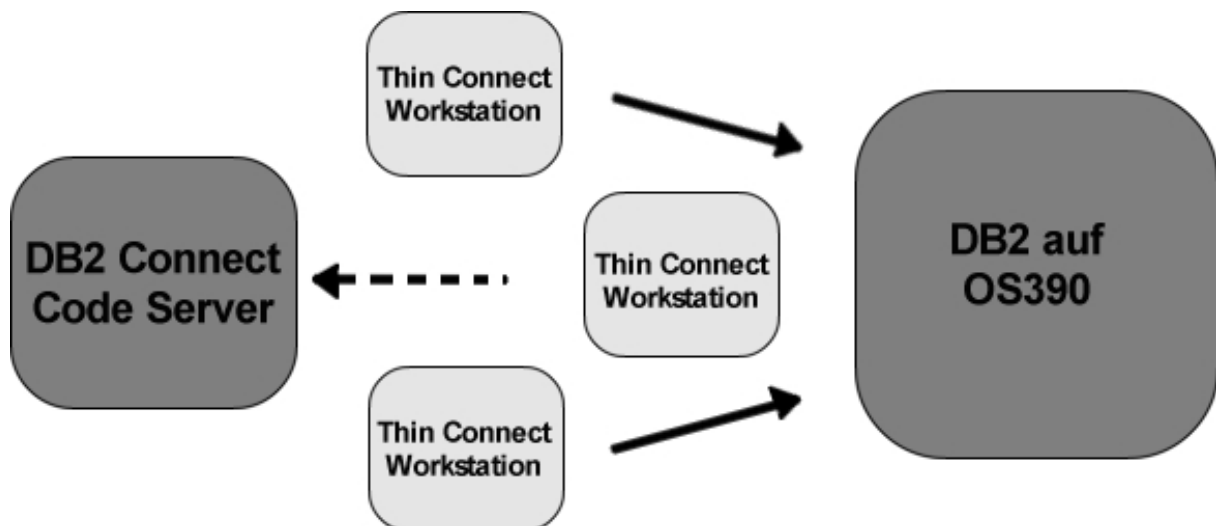


Abbildung 6 - DB2 Thin-Connect Workstation lädt Code vom Code-Server und erstellt eine Verbindung zum Server

Zusammenfassend:

DB2Connect PE ist eine gute Wahl um eine direkte Verbindung zu Mainframes und iSeries Datenbankservern in einer 2-Tier Client/Server Umgebung von PCs aus zu realisieren. Das Produkt ist nicht für Entwicklungen auf Multi-User Systemen oder Application-Servern gedacht und nur für eine begrenzte Anzahl an gleichzeitigen Zugriffen geeignet.

3.2.4 DB2Connect Enterprise Edition (DB2Connect EE)

DB2Connect EE ist eine erweiterte Version der Personal Edition. Sie ist zusätzlich für Hostzugriffe über Gateways und, da alle DB2 und DB2Connect Server Produkte einen integrierten DB2 Client haben, für direkte Zugriffe auf einen entfernten DB2 Server für 2- und 3-Tier Konfigurationen geeignet.

Somit ist die DB2Connect EE Version eine gute Wahl für:

- Web Applikationen z.B. Microsoft IIS, Net.Data™, WebSphere und NetDynamics
- Java Applets
- TP Monitor Applikationen (CICS, MS Transactions Server und MQ Series)
- Host Access für Internet/Intranet Benutzer durch spezielle *web-enabling* Funktionen
- 3-Tier Applikationen mit mehreren Clients,

bei denen die Anzahl der Benutzer leicht einzuschränken ist.

Folgende Plattformen werden unterstützt:

Server: Windows NT, Windows 2000, AIX, HP-UX (32 und 64 bit), Solaris (32 und 64 bit), Linux, Linux/390.

Client: Windows 98, Windows ME, Windows XP (32-bit) Home Edition und Professional Edition, Windows NT, Windows 2000, AIX, HP-UX (32 und 64 bit), Solaris (32 und 64 bit), Linux, Linux 390

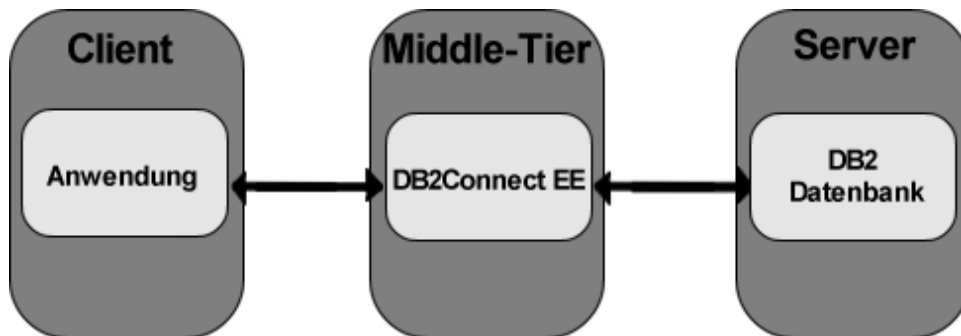


Abbildung 7 - DB2Connect EE Installation

Man unterscheidet bei der DB2Connect EE zwischen zwei Arten von Benutzern:

- Ein *registered user* kann durch einen Namen identifiziert werden und einen oder mehrere DB2Connect EE Server benutzen.
- Die *concurrent user* werden nur als eine Maßeinheit dafür benutzt wie viele Benutzer zur gleichen Zeit den gleichen Server benutzen um sicher zu stellen, dass nicht mehr Benutzer den Server verwenden, als für diesen lizenziert sind.

Es ist viel zu teuer jedem *concurrent user* der gerade auf einen Server zugreifen will der keine freien Lizenzen mehr besitzt eine eigene Lizenz zu kaufen. Somit kann diese Installation nicht für Web Technologien, wie z.B. Internet bzw. großes Intranet, genutzt werden.

DB2Connect EE Feature ESE

Die DB2 UDB Enterprise Server Edition (DB2Connect ESE) beinhaltet eine DB2Connect Komponente für gelegentliche Mainframe Verbindungen für die Datenreproduktion und Server Administration.

Sie erlaubt maximal 5 lizenzierten Benutzern auf S390/zSeries und iSeries Server zuzugreifen. Die Anzahl der Benutzer kann bei dieser Installation nicht erhöht werden.

Zusammenfassend:

Die DB2Connect Enterprise Edition soll benutzt werden, wenn man

- a) bei einer 2-Tier Applikation bei der das Ressourcen Management und die Administrations-Fähigkeit, die in der PE-Version nicht enthalten sind, benötigt werden
- b) eine 3-Tier Architektur erwünscht ist.
- c) die Anzahl der Benutzer leicht einzuschränken ist, da die Lizenzgebühr nach Anzahl der Benutzer ausgelegt ist.

3.2.5 DB2 Connect Application Server Edition (DB2 Connect ASE)

DB2Connect Application Server Edition [ASE] ist von der Funktionalität her identisch mit der DB2Connect Enterprise Edition.

Der Unterschied besteht in den Lizenzbedingungen, die auf die Anforderungen von 3-Tier Client/Server Applikationen und Web Technologien abgestimmt sind.

Die Lizenzgebühren werden daran ausgerichtet, mit wie vielen Prozessoren DB2Connect ASE läuft. Beispiele für solche Applikationen wären WebSphere, BEA Weblogic, Microsoft IIS, CICS, Microsoft MTS, Microsoft COM+, Ascential DataStage, Brio, Business Objects, u.v.m..

Folgende Plattformen werden unterstützt:

Server: Windows NT, Windows 2000, AIX, HP-UX (32 und 64 bit), Solaris (32 und 64 bit), Linux, Linux/390.

Client: Windows 98, Windows ME, Windows XP (32-bit) Home Edition und Professional Edition, Windows NT, Windows 2000, AIX, HP-UX (32 und 64 bit), Solaris (32 und 64 bit), Linux, Linux 390

Zusammenfassend:

Die DB2Connect Application Server Edition ist für die Verbindung von Applikations-Servern (Middle-Tier) zu zSeries oder iSeries Daten in 3-Tier Umgebungen geeignet.

Die Lizenzbedingungen sind für Benutzer mit einer begrenzten Anzahl an Applikations-Servern, die auf Mainframes zugreifen, gemacht. Die Lizenzen werden jedoch nicht nach Anzahl der Rechner, sondern nach Anzahl der Prozessoren in den Rechnern vergeben.

3.2.6 DB2Connect Unlimited Edition(DB2 Connect UE)

Die DB2Connect Unlimited Edition [UED] hat wie die DB2Connect ASE, die gleiche Funktionalität wie DB2Connect EE und unterscheidet sich auch nur durch die Nutzungs- und Lizenzbedingungen.

Um großen OS/390, z/OS und VSE/VM Umgebungen gerecht zu werden, wird anstatt der Benutzer- oder Prozessoranzahl ein Management Services Unit Modell (MSU) eingeführt das unbegrenzten Benutzerzugriff gewährt. Ist der Host in einem Parallel Sysplex Cluster ([SPR] Kapitel 3.3.4), basieren die MSUs auf den DB2 Ansprüchen des gesamten Clusters. D.h. eine MSU darf nicht nur auf eine DB2 Datenbank, auch wenn sie nur auf eine zugreift, ausgelegt sein, sondern muss den Ansprüchen aller Datenbanken im Cluster genügen.

Die Lizenzgebühren richten sich nach der Größe der Datenbank, für die die Funktionalität bereit gestellt werden soll und der Anzahl der eingerichteten MSUs. Somit ist es dem Benutzer erlaubt, eine unbegrenzte Anzahl an DB2Connect EE oder PE Server zu betreiben, die auf einen DB2 UDB Server für OS/390 bzw. z/OS oder DB2 UDB für VSE & VM Datenbank Server zugreifen.

Zusammenfassend:

DB2Connect UE ist somit das umfangreichste Paket für sehr große Anwendung bei denen viele Clients und Applikations-Server mit DB2 Datenbanken verbunden werden sollen.

3.2.7 Softwarepaket Informationen

Die Softwarepakete der verschiedenen DB2Connect Versionen enthalten außer DB2Connect selbst noch zusätzliche Softwareprodukte. Welche Softwareprodukte in welcher Version von DB2Connect enthalten sind zeigt Tabelle 1:

	PE	EE/ESE	ASE	UE
Application Development Client	X	X	X	X
Administration Client	X	X	X	X
Run-Time Client	X	X	X	X
WebSphere Studio Site Developer			X	X
WebSphere + MQSeries			X	X

Tabelle 1 – Zusatzsoftware

Der Funktionsumfang der einzelnen Softwareprodukte ist im folgenden genauer erläutert:

Application Development Client:

stellt die Tools und die Umgebung zum Entwickeln von Anwendungen für den Zugriff auf DB2-Server und Anwendungs-Server bereit, die DRDA implementieren.

Administration Client:

bietet Workstations auf einer Vielzahl von Plattformen die Möglichkeit, auf DB2-Datenbanken zuzugreifen und diese zu verwalten.

Der DB2 Administration Client verfügt über alle Funktionen von DB2 Run-Time Client sowie über alle DB2-Verwaltungs-Tools, ihre Dokumentation und die Unterstützung für Thin Clients.

Darüber hinaus sind die Client-Komponenten für DB2 Query Patroller, ein umfangreiches Tool für die Abfrageverwaltung und zur Verteilung der Auslastung enthalten. Um den Query Patroller verwenden zu können, muss ein Query Patroller-Server installiert sein.

Run-Time Client:

Er bietet Workstations auf einer Vielzahl von Plattformen die Möglichkeit zum Zugriff auf DB2-Datenbanken.

WebSphere Studio Site Developer:

Java-Plattform für die Erstellung von Webanwendungen.

WebSphere:

WebSphere ist ein Web Applikation Server von IBM. Er stellt eine performante Laufzeitumgebung für Servlets zur Verfügung um dynamische Webseiten zu präsentieren [SPR].

MQSeries:

MQSeries ist ein sehr erfolgreiches Middleware-Produkt der Firma IBM für kommerzielles Messaging und Queuing. Es gehört zu der WebSphere-Familie und ermöglicht den Datenaustausch zwischen Anwendungen auf mehr verschiedenen Systemplattformen [SPR].

3.3 HiT JDBC Treiber

Es gibt verschiedene Möglichkeiten von einer Workstation aus auf eine DB2 Datenbank unter OS/390 zuzugreifen. Neben DB2Connect wird von IBM noch der Hit JDBC Treiber der Firma HiT-Software empfohlen.

Zu jedem Produkt der DB2Connect Familie bietet die Firma HiT ein ähnliches Produkt an. In Tabelle 2 werden die entsprechenden Lizenzmodelle gegenüber gestellt:

HiT JDBC Treiber	DB2Connect
Client Lizenz	DB2Connect PE
Entwickler Lizenz (für einen Arbeitsplatzrechner)	DB2Connect EE ohne Server Funktionalität
Limited Connection Server Lizenz: begrenzte Anzahl gleichzeitiger Verbindungen zwischen einem einzelnen Plattform-Server und mehreren DB2 Servern	DB2Connect ESE (für 5 Benutzer) bzw. DB2Connect EE mit Server Funktionalität für mehr als 5 registrierten Benutzer
Open Connection Cluster Server Lizenz	DB2Connect ASE
Open Connection Web Service Server Lizenz	DB2Connect UE

Tabelle 2 - Lizenzübersicht

3.4 Fazit

Für diese Studienarbeit wird DB2Connect von IBM verwendet. Auf Grund der ähnlichen Leistungsmerkmale könnte auch der HiT Treiber in einer geeigneten Lizenzvariante verwendet werden. Ein Leistungsvergleich von zwei gleichen Installationen wäre eine interessante Weiterführung dieser Arbeit.

In der Tabelle 3 sind die für die Auswahl wichtigsten Merkmale der unterschiedlichen DB2Connect Versionen zusammengefasst [DB2]:

	Personal Edition	Enterprise Edition ESE	Enterprise Edition	Application Server Edition	Unlimited Edition
Lizenz basiert auf:					
Benutzeranzahl		5 Stück	X		
Installationsanzahl	X				
Anzahl der Prozessoren der Middle-Tiere				X	
Größe der Datenbank(en) und MSUs					X
Sonstige Merkmale:					
2-Tier	X	X	X		
3-Tier (Middle-Tier als Gateway)		X	X	X	X
Windows 2000	X	X	X		

Tabelle 3 - Übersicht DB2Connect

Auf Grund der für das Praktikum entscheidenden Voraussetzungen ist die DB2Connect EE die richtige Wahl, da es aus lizentechnischen Gründen wichtig ist, dass Studenten von verschiedenen Rechnern mit verschiedenen Betriebssystemen ohne eine separate Installation eines Connectors von zu Hause und von der Universität aus auf die OS/390 DB2 Datenbank zugreifen können. Die Performance bezüglich hoher Benutzerzahlen spielte bei der Auswahl keine Rolle.

4 Installation

4.1 Software

4.1.1 Installation von DB2Connect EE

Für die Installation und Konfiguration wurde das Tutorial [TUT] benutzt. Darin wird die Installation und Konfiguration der DB2Connect PE detailliert beschrieben. Die Unterschiede zur Installation von DB2Connect EE sind nicht gravierend.

Da der Server in Leipzig steht, wurde die VNC Software benutzt um remote von Tübingen aus arbeiten zu können. Auf yoda.informatik.uni-leipzig.de war der VNC Server bereits installiert. Somit war nur noch der VNC Client auf einem Arbeitsplatzrechner zu installieren. Nach der erfolgreichen Installation öffnet man den VNCviewer. Es erscheint ein Fenster in das man zuerst den Rechnernamen bzw. die IP eingeben muss. Danach erfolgt die Benutzeranmeldung auf dem Server (hier: yoda.informatik.uni-leipzig.de) durch die Eingabe des Passworts.

In dem sich öffnenden Fenster wird die Bildschirmausgabe des Zielrechners dargestellt. Man kann so mit der eigenen Maus und Tastatur mit einer geringen Verzögerung auf dem anderen Rechner arbeiten.

Über den VNC Client wurde die gesamte Installation von DB2Connect EE durchgeführt. Zuerst werden wie in Abbildung 8 alle Komponenten ausgewählt

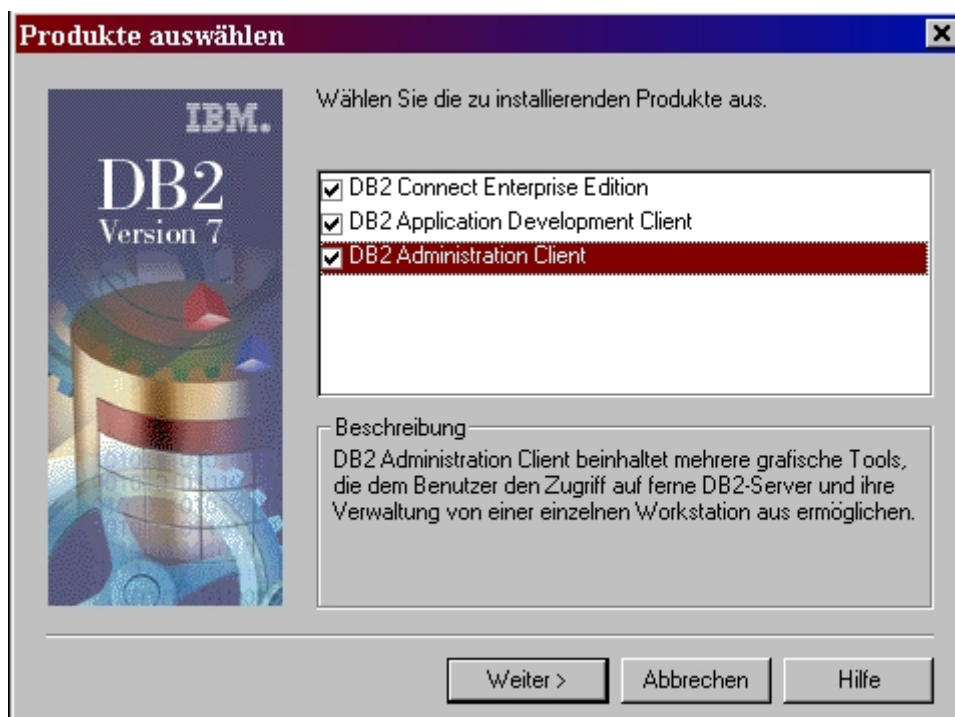


Abbildung 8 - Installationsschritt 1

Danach wird die Standard Installation selektiert (s. Abbildung 9).



Abbildung 9 - Installationsschritt 2

Als nächstes muss noch festgelegt werden, unter welcher Benutzerkennung DB2Connect beim Systemstart automatisch gestartet werden soll. Hier wurde die Administratorkennung von Yoda mit dem entsprechenden Passwort verwendet.

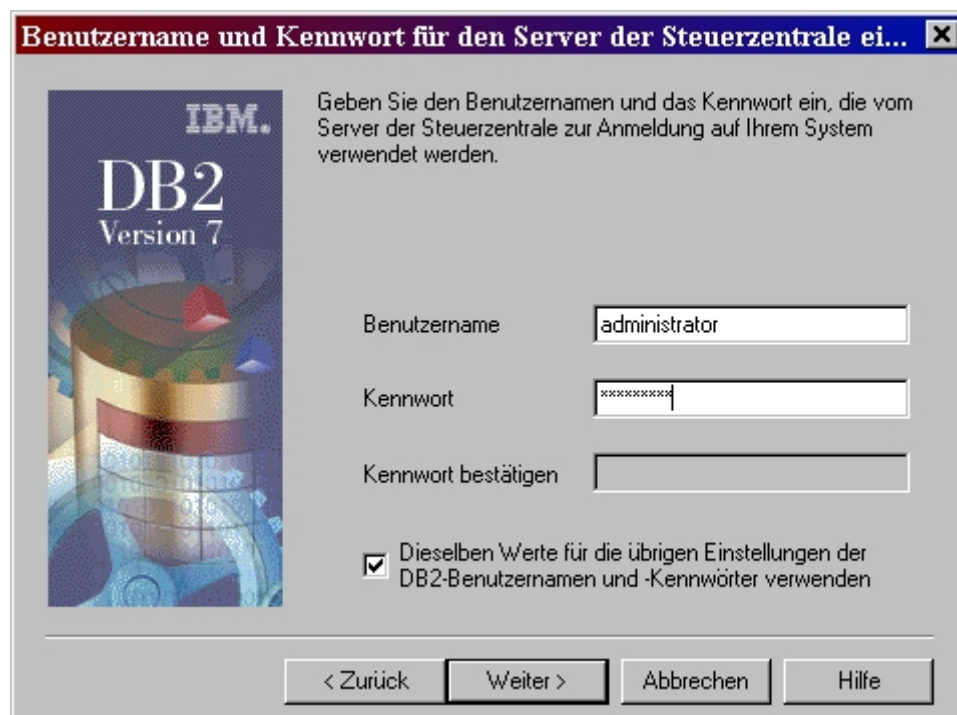


Abbildung 10 – Installationsschritt 3

Nach einem Neustart kann die Verbindung zur DB2 Datenbank konfiguriert werden.

4.1.2 Konfiguration von DB2Connect EE

Um DB2Connect für den Praktikumbetrieb zu verwenden, muss eine Verbindung zur DB2 Datenbank auf jedi angelegt werden. Dies geschieht über die Client-Konfiguration-Unterstützung, welche nun unter

Start → Programme → IBM DB2 → Client-Konfiguration-Unterstützung

zu finden ist.

Durch die Auswahl des Buttons “Hinzufügen“ wird eine neue Verbindung zu einer Datenbank erstellt.

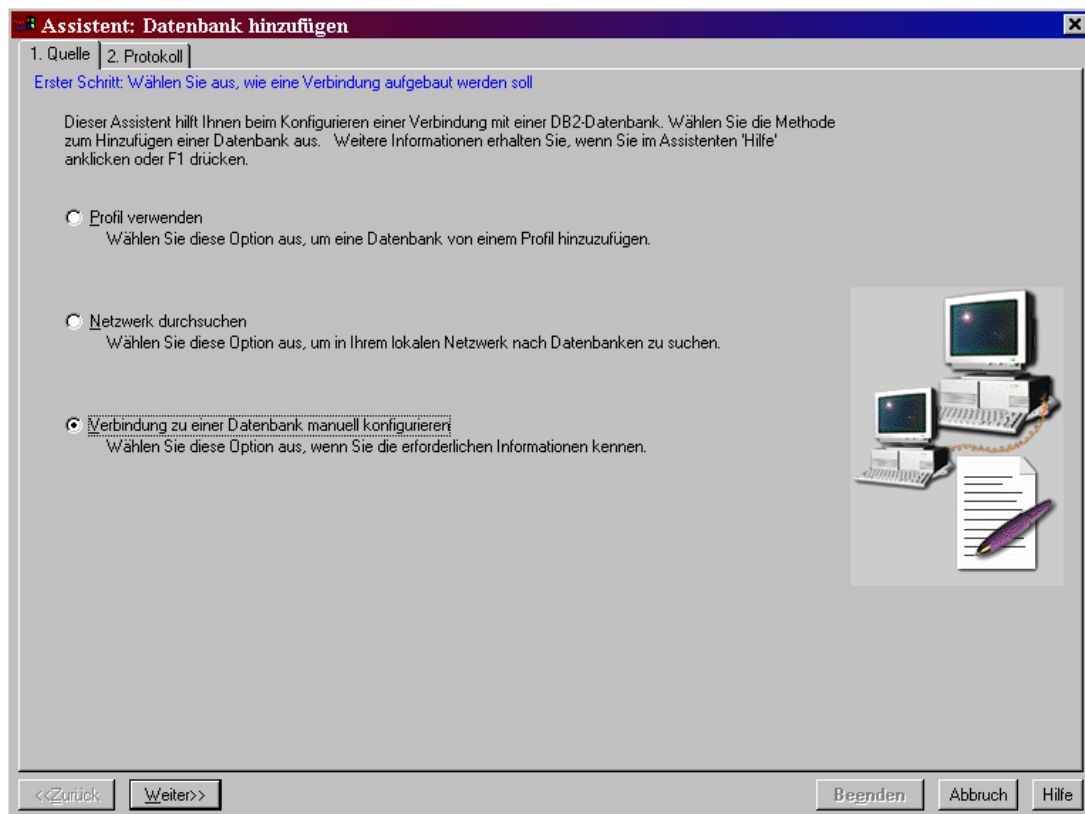


Abbildung 11 – Konfigurationsschritt 1

Wie in Abbildung 11 muss nun „Verbindung zu einer Datenbank manuell konfigurieren“ ausgewählt werden. Mit „Weiter“ wird danach zum nächsten Fenster gewechselt .

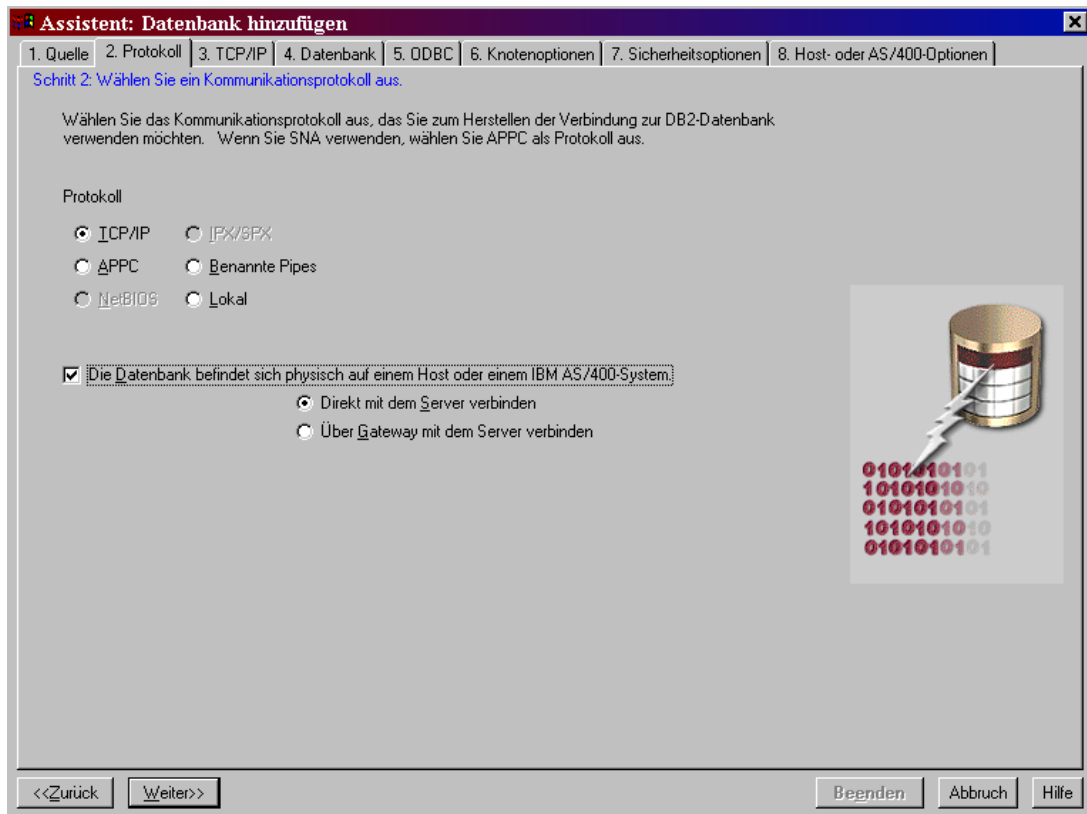


Abbildung 12 - Konfigurationsschritt 2

Als Protokoll wurde, wie in Kapitel 3.2.1 empfohlen, TCP/IP ausgewählt. Außerdem wird die Checkbox und „Die Datenbank direkt mit einem Server verbinden,“ selektiert.

Durch das Ausfüllen der Karteikarte „2.Protokoll“ werden die Karteireiter 3-8 geöffnet, in welche folgende Werte eingetragen wurden:

3.TCP/IP

Host-Name: 139.18.4.97 (IP von jedi.informatik.uni-leipzig.de)
Anschlussnummer: 446

4. Datenbank

Datenbankname: DBA1
Aliasname: DBA1

5. ODBC

alles deselektieren

6. Knotenoptionen

„Knotenoptionen konfigurieren (Wahlfrei)“ selektieren
als Betriebssystem „MVS/ESA, OS390“ wählen

7. Sicherheitsoptionen

„Sicherheitsoptionen konfigurieren (Wahlfrei)“ selektieren
Host- oder AS/400-Authentifizierung (DCS) auswählen aber die
Verschlüsselung **deaktivieren**

8. Host- oder AS/400-Optionen

„Host- oder AS/400-Optionen konfigurieren (Wahlfrei)“ und
„Trennen, wenn der Client eine Unterbrechung generiert ...“ selektieren

Nachdem alle Einstellungen vorgenommen wurden, wird durch einen Klick auf „Beenden“ die Konfiguration übernommen.

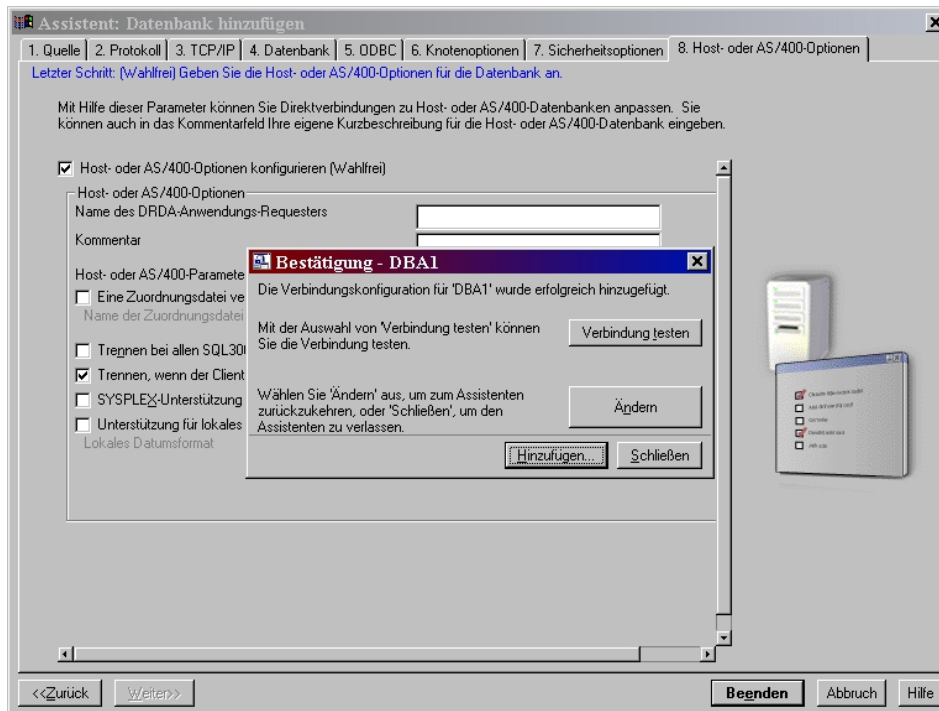


Abbildung 13 - Meldung "erfolgreiche Verbindungskonfiguration" und Aufforderung die Verbindung zu testen

Um sicherzustellen, dass alle Einstellungen richtig übernommen wurden, sollte die Verbindung gleich mit DB2Connect getestet werden.

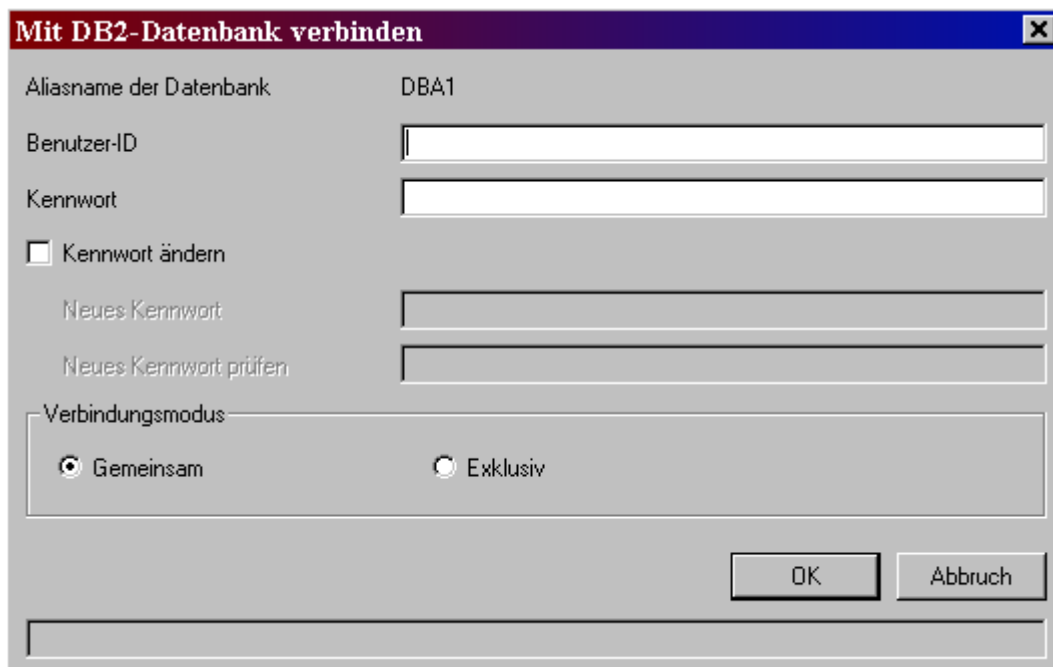


Abbildung 14 - Test von DB2Connect EE

Nach der Eingabe einer gültigen Benutzererkennung und des entsprechenden Passwortes erscheint die Meldung, dass die Testverbindung erfolgreich war (s. Abbildung 15).

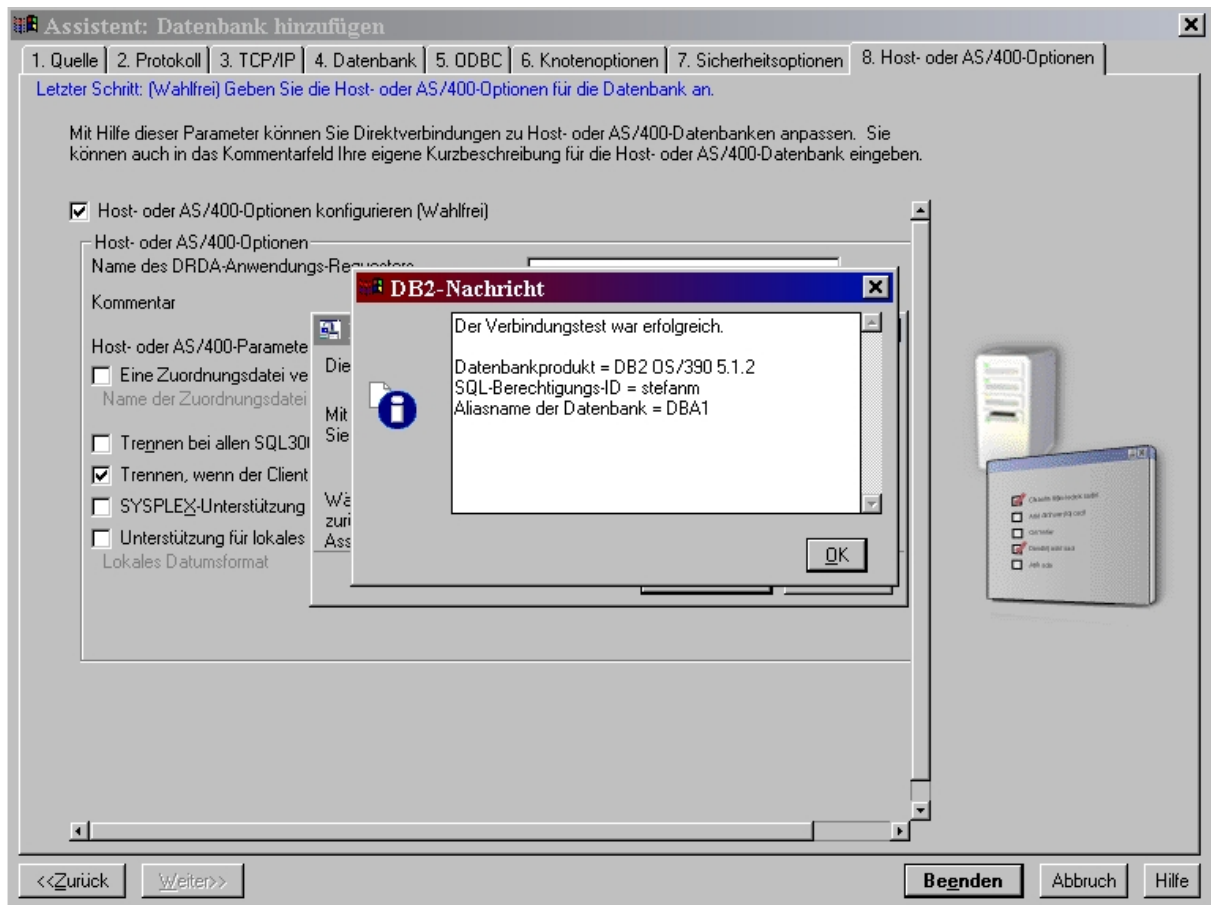


Abbildung 15 – Erfolgsmeldung des Verbindungstests

Falls die Verbindung fehlschlägt, sollte man auf jeden Fall die Benutzererkennung durch einen Login mit einem 3270 Emulator auf dem OS/390 Server überprüfen.

5 Anwendungsbeispiel

In diesem Kapitel wird die durchgeführte Installation ausführlich getestet. Bisher wurde nur die Verbindung zwischen Middle-Tier und Host überprüft, nun soll der vollständige Datenbankzugriff vom Client bis zu Host verifiziert werden. Dafür wird in Kapitel 5.1 der JUnit Installationstest aus [MUN] angepasst. Dabei handelt es sich um eine Java Klasse die zusammen mit dem Junit Testframework [JUN] die Datenbankverbindung überprüft. In Kapitel 5.2 wird das Java Programm aus [MUN] entsprechend verändert.

5.1 TestInstall.java

Die Klasse TestInstall.java aus Abbildung 16 testet die Verbindung zu der in der Konfiguration angegebenen Datenbank DBA1 von einem lokalen PC über Yoda. Das einzige was zu ändern ist, wäre das Passwort, Login, DB2Driver und die URL.

```

15// *****
16// Login und Pwd definieren
17// *****
18 static String user = "prak20";
19 static String password = "prak20";
20// *****
21// Ende Login und Pwd definieren
22// *****
23
24 public TestInstall(String name) {
25     super(name);
26 }
27
28 public void setUp() {
29
30 }
31
32 public void testDrivers() {
33     // Windows => teste beide Connectoren , hit + DB2Connect
34     assertTrue(
35         "Hit Treiber funktioniert nicht!",
36         connectionTest(
37             "hit.db2.Db2Driver",
38             "jdbc:db2://jedi.informatik.uni-leipzig.de:446;rdname=DBA1"));
39
40     *****
41     jdbc:db2:DBA1 zu jdbc:db2://yoda.informatik.uni-leipzig.de/DBA1 ändern
42     *****
43     assertTrue(
44         "DB2Connect funktioniert nicht",
45         connectionTest("COM.ibm.db2.jdbc.net.DB2Driver", "jdbc:db2://yoda.informatik.uni-leipzig.de/DBA1
46     *****
47     Ende jdbc:db2:DBA1 zu jdbc:db2://yoda.informatik.uni-leipzig.de/DBA1 ändern
48     *****
49 }
50

```

Abbildung 16 - Geänderte Codestellen von TestInstall.java

In Zeile 18 und 19 müssen die Variablen user und password an die eigenen Benutzerdaten angepasst werden. Um eine Verbindung aufzubauen muss ein „DB2Driver“ eingebunden werden. Da ursprünglich eine 2-Tier Umgebung verwendet wurde, muss man den Treiber und den Ort an dem DB2Connect installiert ist in Zeile 45 ändern. Der richtige DB2Driver ist COM.ibm.db2.jdbc.net.DB2Driver, der Pfad muss in jdbc:db2://yoda.informatik.uni-leipzig.de/DBA1 geändert werden. (Achtung: Hierbei kommt es auf Groß- und Kleinschreibung an).

Nachdem der Quellcode in eclipse [ECL] ausgeführt wurde ergab der JunitTest (Abbildung 17) keinerlei Fehler. Also muss DB2Connect korrekt installiert sein.

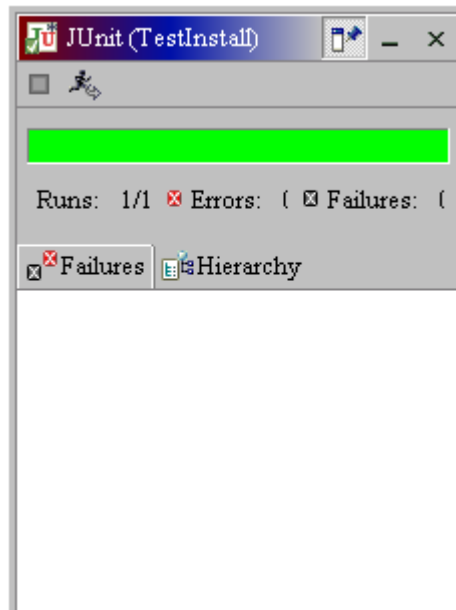


Abbildung 17 - Ergebnis des JunitTests

5.1 JDBCTestPrg.java

Das JDBCTestPrg.java greift auf die DBA1 DB2 Datenbank zu und liest alle Tabellennamen aus. Auch hier waren nur wenige Änderungen vorzunehmen.

```
28//.....
29//Login und Pwd definieren
30//.....
31 static String user = "prak20";
32 static String password = "prak20";
33//.....
34// Ende Login und Pwd definieren
35//.....
36 String jdbcUrl = "jdbc:db2:DBA1";
37 public static void main(String argv[]) {
38     try {
39         System.out.println(
40             "Esp Programm zum Zugriff auf eine DB2 Datenbank mit JDBC 1.0\von Stefan Munnz");
41// System.out.println("Bitte wählen Sie, welche Treiber/Betriebssystem Variante Sie benutzen möchten\n");
42// System.out.println("1 = OS/390 (native IBM JDBC Treiber)\n");
43// System.out.println("2 = Windows, HitJDBC 1.0 Treiber, der Hit Treiber muss auf Ihrem System installiert sein\n");
44// System.out.println("3 = Windows, DB2Connect, DB2Connect muss installiert sein und eine Verbindung mit Namen DBA1 zur DB
45 BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
46 String sDriver = stdin.readLine(); // Eingabe lesen
47 Connection con;
48 // Treiber und URL an die Auswahl anpassen:
49 if (sDriver.equals("1")) {
50     Class.forName("ibm.sql.DB2driver");
51     // Create the connection
52     con = DriverManager.getConnection("jdbc:db2os390:DBA1", user, password);
53 } else if (sDriver.equals("2")) {
54     Class.forName("hit.db2.Db2Driver");
55     // Create the connection
56     con = DriverManager.getConnection(
57         "jdbc:db2://jedi.informatik.uni-leipzig.de:496:rdbname=DBA1",
58         user,
59         password);
60 } else if (sDriver.equals("3")) {
61     Class.forName("COM.ibm.db2.jdbc.net.DB2Driver");
62     // Create the connection
63//.....
64// URL anpassen
65//.....
66     con = DriverManager.getConnection("jdbc:db2://yoda.informatik.uni-leipzig.de/DBA1", user, password);
67// Ende URL anpassen
68//.....
69//.....
```

Abbildung 18 - Änderungen in JDBCTestPrg.java (Login und URL)

In Zeile 31 und 32 müssen wiederum die Variablen user und password an die eigenen Benutzerdaten angepasst werden. In Zeile 66 ist noch eine Anpassung der URL vonnöten, damit der DB2Connect Server gefunden wird. Die URL muss auch wieder jdbc:db2://yoda.informatik.uni-leipzig.de/DBA1 lauten.

Da mittlerweile sehr viele Tabellen auf der Datenbank liegen, wurde nur der Inhalt der Tabelle GUESTBOOK in Abbildung 19 ausgegeben. Es ist zu beachten, dass man nur Werte von den Tabellen ausgeben kann für die man auch Benutzerrechte hat.

```

Console [sterminated> C:\jdk1.4.0_03\bin\javaw.exe (05.08.03 14:03)]
Bsp Programm zum Zugriff auf eine DB2 Datenbank mit JDBC 1.0
von Stefan Munnz
3 = Windows, DB2Connect, DB2Connect muss installiert sein und eine Verbindung mit Namen DBA1 zur DB definiert sein

3
**** JDBC Entry within class sample01.
**** JDBC Connection.
**** JDBC Statement Created
**** JDBC Result Set Created

spruth@informatik.uni-tuebingen.de

**** JDBC Result Set output completed
**** JDBC Statement Closed
**** JDBC Disconnect.
**** JDBC Exit from class sample01 - no Errors.

```

Abbildung 19 - Ausgabe von JDBCTestPrg.java

Der Ablauf der Programms wird in Abbildung 20 erläutert.

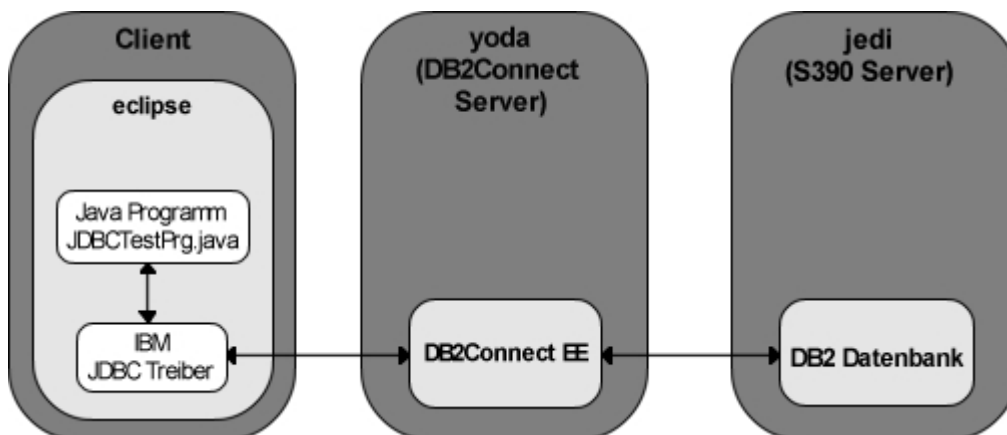


Abbildung 20 - Ablauf des JDBCTestPrg.java in Zusammenhang mit dem Middle-Tier und dem Backend

Das Java Programm greift über den in IBM JDBC Treiber auf den DB2Connect Server zu und übermittelt ihm die SQL-Query `"SELECT * FROM GUESTBOOK"`. Darauf hin wird von yoda aus eine Verbindung zu jedi über den Port 446 erstellt. Die Anfrage kommt zunächst in die Eingabewarteschlange. Jedi wertet dann die Anfrage aus und übermittelt das Ergebnis, in unserem Fall nur „spruth@informatik.uni-tuebingen.de“, an yoda zurück. Dieser liefert das Ergebnis über den Port 446 an den anfragenden Client zurück.

6 Literaturverzeichnis

[CSA] IBM: *Websphere Studio Workbench Homepage*

<http://www.unisysworld.com/monthly/2001/05/clientserver.shtml>

[SPR] Paul Herrmann, Udo Keschull und Wilhelm G. Spruth:

Einführung in z/OS und OS/390 Web-Services und Internet-Anwendungen für Mainframes

[DB2] IBM: *Which Edition of DB2 Connect is Right for You?*

<http://www7b.software.ibm.com/wsdd/http.downloads/1/%3C%3Ek8urhrnflbry9pxzxdub2zu/dmdd/db2/0303zikopoulos1/db2connect.pdf>

[IBM] IBM: *DB2Connect Features and benefits*

<http://www-3.ibm.com/software/data/db2/db2connect/features.html>

[PED] IBM: *DB2Connect Personal Edition*

<http://www-3.ibm.com/software/data/db2/db2connect/db2connect-pe.html>

[EED] IBM: *DB2Connect Enterprise Edition*

<http://www-3.ibm.com/software/data/db2/db2connect/db2connect-ee.html>

[UED] IBM: *DB2Connect Unlimited Edition*

<http://www-3.ibm.com/software/data/db2/db2connect/db2connect-ue.html>

[ASE] IBM: *DB2Connect Application Server Edition*

<http://www-3.ibm.com/software/data/db2/db2connect/db2connect-ase.html>

[TUT] Degner : *Tutorial 8 Installation und Konfiguration von DB2Connect*

<http://jedi.informatik.uni-leipzig.de/degner/tutor8.pdf>

[MUN] Stefan Munz: *Java Programmierung mit JDBC Zugriff auf eine OS/390 DB2 Datenbank*

<http://www-ti.informatik.uni-tuebingen.de/~csp/diplom/munz/>

[ECL] eclipse: Homepage

<http://www.eclipse.org>