Lars Bendzka

Integration von relationalen Datenbanken zu föderierten Systemen

Diplomarbeit

Diplom.de

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de/ abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlages. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 1997 Diplom.de
ISBN: 9783832406851

https://www.diplom.de/document/216613

Lars Bendzka

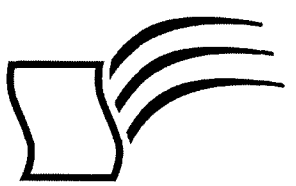
Integration von relationalen Datenbanken zu föderierten Systemen

Diplom.de

Lars Bendzka

Integration von relationalen Datenbanken zu föderierten Systemen

Diplomarbeit
an der Universität Hannover
September 1997 Abgabe



Diplomarbeiten Agentur
Dipl. Kfm. Dipl. Hdl. Björn Bedey
Dipl. Wi.-Ing. Martin Haschke
und Guido Meyer GbR

Hermannstal 119 k
22119 Hamburg

agentur@diplom.de
www.diplom.de

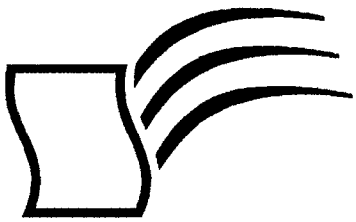
ID 685
Bendzka, Lars: Integration von relationalen Datenbanken zu föderierten Systemen /
Lars Bendzka - Hamburg: Diplomarbeiten Agentur, 1998
Zugl.: Hannover, Universität, Diplom, 1997

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden, und die Diplomarbeiten Agentur, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Dipl. Kfm. Dipl. Hdl. Björn Bedey, Dipl. Wi.-Ing. Martin Haschke & Guido Meyer GbR
Diplomarbeiten Agentur, http://www.diplom.de, Hamburg
Printed in Germany



Diplomarbeiten Agentur

Wissensquellen gewinnbringend nutzen

Qualität, Praxisrelevanz und Aktualität zeichnen unsere Studien aus. Wir bieten Ihnen im Auftrag unserer Autorinnen und Autoren Wirtschaftsstudien und wissenschaftliche Abschlussarbeiten – Dissertationen, Diplomarbeiten, Magisterarbeiten, Staatsexamensarbeiten und Studienarbeiten zum Kauf. Sie wurden an deutschen Universitäten, Fachhochschulen, Akademien oder vergleichbaren Institutionen der Europäischen Union geschrieben. Der Notendurchschnitt liegt bei 1,5.

Wettbewerbsvorteile verschaffen – Vergleichen Sie den Preis unserer Studien mit den Honoraren externer Berater. Um dieses Wissen selbst zusammenzutragen, müssten Sie viel Zeit und Geld aufbringen.

http://www.diplom.de bietet Ihnen unser vollständiges Lieferprogramm mit mehreren tausend Studien im Internet. Neben dem Online-Katalog und der Online-Suchmaschine für Ihre Recherche steht Ihnen auch eine Online-Bestellfunktion zur Verfügung. Inhaltliche Zusammenfassungen und Inhaltsverzeichnisse zu jeder Studie sind im Internet einsehbar.

Individueller Service – Gerne senden wir Ihnen auch unseren Papierkatalog zu. Bitte fordern Sie Ihr individuelles Exemplar bei uns an. Für Fragen, Anregungen und individuelle Anfragen stehen wir Ihnen gerne zur Verfügung. Wir freuen uns auf eine gute Zusammenarbeit

Ihr Team der *Diplomarbeiten* Agentur

Dipl. Kfm. Dipl. Hdl. Björn Bedey –
Dipl. Wi.-Ing. Martin Haschke –
und Guido Meyer GbR –

Hermannstal 119 k –
22119 Hamburg –

Fon: 040 / 655 99 20 –
Fax: 040 / 655 99 222 –

agentur@diplom.de –
www.diplom.de –

Zusammenfassung

Seit vielen Jahren werden Datenbanksysteme von fast allen größeren Unternehmen eingesetzt, um Informationen effizient zu speichern und zu verwalten. Heutzutage wird es dabei immer wichtiger, schon bestehende, historisch gewachsene und bisher dezentral verwaltete Datenbanksysteme zusammenzufassen; typischerweise, indem sie zu einem föderierten Datenbanksystem (FDBS) integriert werden. Ein FDBS besteht dabei aus autonomen und heterogenen Datenbanksystemen, auch Komponentendatenbanksysteme (KDBS) genannt, die über eine Föderierungsschicht gekoppelt werden. Da auf den einzelnen KDBS oft Anwendungen aufsetzen, die auch nach der Integration weiterhin ablauffähig bleiben sollen, muß bei der Föderierung insbesondere die lokale Autonomie der einzelnen KDBS gewährleistet bleiben.

Bei der Integration von Datenbanken können eine Vielzahl an Heterogenitäten, z.B. in Form von strukturellen oder semantischen Konflikten, auftreten, die zunächst identifiziert und dann gelöst werden müssen. In dieser Arbeit wurden dazu die bei der Integration auftretenden Konflikte klassifiziert und Lösungen in Form von Algorithmen zusammengestellt.

Basierend auf einer konzeptionellen Ausarbeitung zur Integration relationaler Datenbanken wurde in dieser Arbeit ein Programm entwickelt, mit dem es möglich ist, Relationen aus verschiedenen Datenbanken über die flexible Datenbankschnittstelle ODBC einzulesen und über eine graphische Oberfläche in ein föderiertes Schema zu integrieren. In dieser Arbeit werden dazu exemplarisch die DBMS Oracle und DB2 betrachtet. Weiterhin kann der Anwender mit Hilfe der Föderierungsschicht in einer dafür zur Verfügung gestellten Anfragesprache eine Anfrage an das föderierte Schema stellen. Die Ergebnisberechnung, sowie die Zerlegung und Optimierung der Anfrage ist dabei die Aufgabe des Programms, das gemeinsam mit einer Oracle-Datenbank die Föderierungsschicht bildet.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Erstellung dieser Diplomarbeit unterstützt haben. Bei Prof. Dr. Udo Lipeck bedanke ich mich für seine Vorschläge, Anregungen und konstruktive Kritik. Besonders bedanken möchte ich mich auch bei Dr. Michael Gertz für seine hervorragende Betreuung und sein Interesse an dieser Arbeit. Er stand sogar am Wochenende für Diskussionen zur Verfügung und gab mir Hilfestellung bei der Auswahl der Materialien. Weiterhin möchte ich allen denen danken, die mir durch Tips geholfen haben und die ich hier nicht namentlich erwähnt habe.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Dieses erste Kapitel soll als Einstieg in diese Arbeit die historische Entwicklung der Datenverwaltung von der Dateiverwaltung hin zu den föderierten Datenbanksystemen erläutern. Dabei werden Gründe für den Einsatz von Datenbanksystemen diskutiert und die heute notwendige „Zusammenführung" von Datenbanken, die sogenannte *Integration* oder auch *Föderierung* von Datenbanken, vorgestellt.

In Abschnitt 1.2 definieren wir das Ziel dieser Arbeit genauer und geben in Abschnitt 1.3 einen Überblick über die Gliederung des Textes.

1.1 Einführung

In den Anfängen der Datenverarbeitung waren die Anwendungen mangels Alternativen noch dateibasiert, und die Daten wurden sequentiell auf Magnetbändern gespeichert. Falls eine Anwendung erweitert wurde, mußten auch die zugehörigen Datensätze in den Dateien erweitert werden, was häufig zu Problemen führte.

Mit dem Aufkommen der ersten Datenbank-Managementsysteme (DBMS) in den 60er Jahren fingen eine ganze Reihe von Unternehmen an, ihre bestehenden Anwendungen zu modifizieren und neue Anwendungen auf der Grundlage dieser DBMS (der 1. Generation) zu realisieren. Da damals die Hardware noch sehr teuer war, versuchte man die Ressourcen möglichst effizient zu nutzen und baute Zentralrechner bzw. zentrale Rechenzentren auf, wobei jedoch die betrieblichen Abläufe in den Unternehmen geändert werden mußten.

Der Einsatz eines zentralen Datenbanksystems, das — wie der Name schon sagt — alle Daten einheitlich und zentral verwaltet, führte jedoch zu sehr mächtigen, unübersichtlichen und langsamen Systemen.

Die 80er und 90er Jahre brachten einen großen Fortschritt in der Entwicklung von Festspeichermedien und die Hardware wurde um ein Vielfaches preiswerter. Doch leider ließen sich die schon existierenden Datenbankanwendungen, die auf getrennten Systemen realisiert wurden, nicht so einfach zusammenführen. Dies würde erhebliche Eingriffe in die Anwendungsprogramme nach sich ziehen. Schon Mitte der 70er Jahre überlegte man sich deshalb Realisierungskonzepte für *verteilte Datenbank-Managementsysteme*

(vDBMS), also von DBMSen, deren Daten auf verschiedenen Rechnern gespeichert werden, die sich jedoch dem Anwender gegenüber wie ein zentrales Datenbanksystem verhalten.

Heutzutage stellt die Datenbanktechnologie in vielen Firmen und Institutionen einen wichtigen Aspekt für schnellen und rationellen Zugriff auf die Daten dar. Immer mehr Anwender benutzen Datenbanken bei ihrer täglichen Arbeit. Dabei zeichnen sich zwei Trends zur verteilten Verarbeitung ab. Der eine, sich inbesondere in Großunternehmen abzeichnende *Trend zur Dezentralisierung*, also zur Bildung von — mehr oder weniger — unabhängigen Teileinheiten mit eigener Gewinn- und Kostenverantwortung, löst bisher existierende zentrale Informationssysteme durch verteilte Informationssysteme ab. Der andere Trend geht zur *Integration* bzw. *Kooperation autonomer Informationssysteme*. Ein Beispiel dazu liefert wieder die Industrie: Um Kosten zu sparen, reduzieren viele Firmen ihre Fertigungstiefe (vgl. [Her95]). Sie stellen ihre Vorprodukte nicht mehr selbst her, sondern kaufen diese bei Vorlieferanten ein, die sich auf diese Produkte spezialisiert haben. Da die Firmen eng zusammenarbeiten und die Vorlieferanten meist „Just-In-Time" liefern, muß ein reibungsloser Datenaustausch gewährleistet sein. Hier sind die Datenbanken entsprechend zu „verbinden", so daß die Daten elektronisch zwischen den Betrieben übermittelt werden können.

Damit die Firmen bei einer Anfrage an ihr Datenbanksystem auch alle Datenbanken ihrer Vorlieferanten „erreichen", müssen alle bisher dezentral verwalteten Datenbanksysteme in *ein* Datenbanksystem zusammengeführt werden. Diesen Vorgang wollen wir im folgenden als **Integration** bezeichnen. In den einzelnen (Komponenten-) Datenbanken liegen sogenannte *Komponentenschemata* vor, die bei der Integration in *ein* Schema, das sogenannte **globale Schema**, zu integrieren sind. Auf dieses globale Schema können dann globale Anwendungen „aufsetzen".

Daß das Integrieren der Datenbanksysteme nicht ad-hoc geschehen kann, liegt auf der Hand, da in den Firmen oft Datenbanksysteme unterschiedlicher Hersteller vorliegen, die nicht einfach zusammengefaßt werden können. Weiterhin treten insbesondere folgende Probleme auf:

- In den Datenbanksystemen liegen unterschiedliche Datenmodelle vor, d.h. die eine Datenbank basiert z.B. auf dem relationalen Modell, eine andere hingegen auf dem objekt-orientierten Modell. Hier muß eine Möglichkeit gefunden werden, wie die *Heterogenität* bzgl. des verwendeten Datenmodells überwunden werden kann. Auch wenn die Datenmodelle identisch sind, können *semantische Heterogenitäten* auftreten (siehe weiter unten).

- Wir können davon ausgehen, daß auf die schon existierenden Datenbanksysteme Anwendungsprogramme zugreifen, die auch nach der Integration weiterhin ablauffähig bleiben müssen. Wenn die Datenbanken zusammengefaßt werden, darf dabei ihre Struktur nicht verändert werden — mit anderen Worten — die *lokale Autonomie* der Komponentendatenbanken muß gewährleistet bleiben.

- Die verschiedenen Datenbanken wurden von unterschiedlichen Programmierern erstellt. Diese modellierten zu den jeweiligen Anwendungen Datenstrukturen, also z.B. Relationenschemata und Integritätsbedingungen, nach ihren Vorstellun-

gen. Wenn später die Datenbanksysteme zusammengefaßt werden, „passen" diese Strukturen oft nicht zueinander. Die bei der Integration der Schemata auftretenden *Konflikte* wie z.B. Namens- oder Strukturkonflikte, müssen zunächst gefunden und dann beseitigt werden. Dabei ist zu beachten, daß die Semantik zwischen den Datenbankschemata oft nur den Anwendern bzw. Programmierern bekannt sind. Der Integrationsvorgang kann somit nicht automatisiert werden, sondern es sind immer wieder Benutzerinteraktionen nötig.

1.2 Ziel der Arbeit

In dieser Arbeit beschäftigen wir uns mit der Integration schon existierender, relationaler Datenbanken zu einem sogenannten **föderierten System**. Ein föderiertes Datenbanksystem ist ein Spezialfall eines verteilten Datenbanksystems und besteht aus *autonomen* und *heterogenen* Komponentendatenbanken, die logisch und evtl. physisch verteilt sind. Dabei sollen die Komponentendatenbanken bei weitgehender Erhaltung der lokalen Autonomie *föderiert* werden, also über eine Föderierungsschicht gekoppelt werden, so daß eine *einheitliche Sicht* auf unterschiedliche Strukturen möglich wird. Dabei wollen wir uns in dieser Arbeit darauf beschränken, daß in den Komponentendatenbanken das gleiche Datenmodell, hier das relationale Datenmodell, vorliegt. Wir betrachten somit homogene föderierte Systeme. Eine weitere Einschränkung machen wir, indem wir vorgeben, daß wir aus den Komponentendatenbanken nur Informationen lesen, nicht aber Informationen modifizieren wollen. Falls auch schreibender Zugriff erwünscht wird, müssen zusätzlich noch Transaktionskonzepte (z.B. [Scha96]) ausgearbeitet werden.

Wir betrachten in dieser Arbeit insbesondere das **relationale Datenmodell**, da sich der Großteil schon existierender Arbeiten zu den Thema „Integration von Datenbanken" fast ausschließlich mit anderen Datenmodellen, insbesondere mit dem ER-Modell, beschäftigen; für das *relationale Modell* liegen bisher jedoch kaum Arbeiten vor.

Einige der Arbeiten, die sich mit den anderen Datenmodellen beschäftigen, sind in Tabelle 1.1 zusammengestellt. Die anderen dort aufgeführten Datenmodelle wollen wir hier nicht weiter spezifizieren. Ein weiterer Überblick über Methoden zur Integration ist in [BLN86] zu finden.

In der Literatur werden die Begriffe **View-**, **Schema-** und **Datenbank-Integration** von verschiedenen Autoren unterschiedlich definiert. Dabei werden Methoden beschrieben, in der verschiedene Schemata von existierenden Datenbanken in ein globales, einheitliches Schema überführt werden oder bei denen mehrere Anwenderviews in ein integriertes Schema zusammengefaßt werden.

Wir haben in dieser Arbeit sowohl eine Schema-, als auch eine Datenbank-Integration ausgearbeitet und fassen die Begriffe typischerweise in dem Begriff **Integration** zusammen. Bei der Integration müssen die schon in vorherigen Abschnitt angesprochenen Probleme gelöst werden:

Literatur	Verwendetes Modell	Projekt oder Programm
[BC85]	Relationales Modell	
[BL84]	Entity-Relationship Modell	
[CV83]	Relationales Modell	
[DS84]	Entity-Relationship Modell	
[HTJ+95]	mehrere Modelle	Projekt $SIGMA_{FDB}$
[KDN89]	VODAK, Objekt-orientiertes Modell	Projekt KODIM
[NG82]	Navathe-Schkolnick Modell	
[NSE84]	Entity-Relationship Modell	
[Schm95]	ODMG-93, Objekt-orientiertes Modell	
[SGN93]	CANDIDE Modell	
[SM92]	Entity-Relationship Modell	Programm BERDI
[SP90]	Entity-Relationship Modell	
[SPD91]	Entity-Relationship Modell	

Tabelle 1.1: Andere Arbeiten zum Thema „Integration von Datenbanken"

- Die *lokale Autonomie* der Komponentendatenbanken muß gewährleistet bleiben, d.h. wir dürfen bei der Integration keine Strukturen in den Komponentendatenbanken verändern, da wir davon ausgehen, daß lokale Anwendungsprogramme vorhanden sind, die auch nach der Integration ablauffähig bleiben müssen.

- Die bei der Integration auftretenden *Konflikte* zwischen den zu integrierenden Relationen verschiedener Komponentendatenbanken müssen zunächst gefunden und dann beseitigt werden. Dabei müssen auch *semantische Heterogenitäten* geeignet überwunden werden.

  In dieser Arbeit wurde dazu eine Klassifizierung von Konflikten und deren Auflösungen zusammengestellt. Diese Zusammenstellung umfaßt die typischen Konflikte, die in der Literatur zum ER-Modell diskutiert werden, und zeigt zusätzliche Konflikte auf, bei denen z.B. Ausprägungen des Datenbankzustandes betrachtet werden.

- Wir müssen nach einer Möglichkeit suchen, die uns einen flexiblen Zugriff auf relationale Datenbanken unterschiedlicher Hersteller gestattet.

Da wir von homogenen Komponentendatenbanken ausgehen, brauchen wir die Heterogenität bzgl. des verwendeten Datenmodells nicht zu betrachten.

Im Rahmen dieser Arbeit wurde das Programm $SIRED_{FDBS}$ (**S**chema **I**ntegration of **RE**lational **D**atabases to **F**ederated **D**ata-**B**ase-**S**ystems) entwickelt, das dem Anwender ein flexibles Entwicklungswerkzeug zur Verfügung stellt, um ein einfaches föderiertes Datenbanksystem (FDBS) zu erstellen. Das Programm $SIRED_{FDBS}$ bildet später, gemeinsam mit einer Oracle-Datenbank, die Föderierungsschicht des FDBS. Der Anwender kann über eine einfach zu bedienende graphische Oberfläche verschiedene Relationen aus unterschiedlichen relationalen Komponentendatenbanken in ein *globales Schema*, das sogenannte **föderierte Schema**, integrieren. Dabei wird der Anwender