

## Vorwort

Datenbanken und Datenbanktechnologie spielen heutzutage in praktisch allen Bereichen, in denen Computer eingesetzt werden, eine tragende Rolle. Die Nachteile konventioneller Dateisysteme, die zunehmenden Anforderungen an die Datenhaltung und an die Auswertung von Daten sowie neue Anwendungsgebiete sorgten und sorgen für eine rasante Entwicklung und Verbesserung der Konzepte und Techniken von Datenhaltungssystemen. Die Forderung nach verallgemeinerten und standardisierten Funktionen zur Datendefinition und Datenmanipulation sowie zur Integritätsüberwachung und Zugriffskontrolle bewirkten in einem stufenweisen Prozeß die Entwicklung von einfachen Dateisystemen zu allgemeinen *Datenbanksystemen*. Ein Datenbanksystem übernimmt alle Aufgaben der effizienten Datenhaltung und -verwaltung in einem Anwendungssystem. Es zeichnet sich vor allem durch einen hohen Grad an Datenunabhängigkeit, durch ein logisches Datenmodell und durch die Bereitstellung einer Anfragesprache aus. Weiterhin unterstützt es den Mehrbenutzerbetrieb und bietet ein Transaktionskonzept sowie Maßnahmen zur Datensicherung an.

In diesem Kurs werden wir der Frage nachgehen, wie Datenbanksysteme, die bekanntermaßen große Softwaresysteme darstellen, realisiert werden, d.h., welchen Implementierungsanforderungen sie unterliegen, wie die Architektur solcher Systeme aussieht und welche allgemeinen Konzepte für die Implementierung von Datenbanksystemen existieren. Hierbei sind verschiedene Sichtweisen auf Datenbanksysteme möglich, die spezielle Konzepte und Techniken erfordern können. Datenbanksysteme können klassifiziert werden nach dem zugrundeliegenden Datenmodell, nach Anwendungsgebieten und nach Einsatzformen.

Im Laufe der Zeit sind verschiedene Datenbankmodelle entworfen worden, die zur Entwicklung der entsprechenden Datenbanksysteme geführt haben. Diese Modelle beschreiben im wesentlichen verschiedene Sichtweisen auf Datenbankobjekte und den auf ihnen definierten Operationen. Wichtige Beispiele sind das hierarchische Modell, das Netz-

werkmodell, das relationale Modell, das objekt-orientierte Modell und das objekt-relationale Modell.

Aus Anwendungssicht haben sich Datenbanksysteme mittlerweile in den administrativen und betriebswirtschaftlichen Anwendungsgebieten etabliert. „Klassische“ Anwendungsbereiche sind zum Beispiel Personalwesen, Produktionsplanung und -steuerung, Waren- und Materialwirtschaft, Rechtswesen, Medizin, Bankwesen, Buchungswesen, usw. Diesen sogenannten „Standard“-Datenbankanwendungen ist gemein, daß die von ihnen verarbeiteten Daten – im wesentlichen handelt es sich um alphanumerische Daten – eine einfache Struktur aufweisen. Andererseits haben sich in den letzten Jahren in zunehmendem Maße eine ganze Reihe neuer, „nicht-klassischer“ Computeranwendungen herauskristallisiert, die aus der Sicht herkömmlicher Datenbanksysteme als „Nicht-Standard“-Datenbankanwendungen bezeichnet werden. Beispiele sind CAD-Datenbanksysteme für den rechnergestützten Entwurf, geographische Informationssysteme und Geo-Datenbanksysteme für geowissenschaftliche Anwendungen wie Kartographie und Anwendungen zur räumlichen Analyse, Expertendatenbanksysteme, deduktive Datenbanksysteme oder Wissensbankverwaltungssysteme für wissensbasierte Anwendungen wie Expertensystemanwendungen und entscheidungsunterstützende Anwendungen sowie Multimedia-Datenbanksysteme für Anwendungen, die Text, Bild, Ton und Sprache integrieren. Alle diese Nicht-Standard-Anwendungen zeichnen sich durch das Auftreten enormer Datenmengen, durch eine große Anwendungskomplexität und durch komplex strukturierte und große Objekte aus. So werden beispielsweise in CAD-Datenbanksystemen, Geo-Datenbanksystemen und Multimedia-Datenbanksystemen geometrische Objekte wie Punkte, Polylinien, Polygone und Polyeder verwaltet, die eine komplexe Struktur aufweisen und besondere Speicherungs-, Zugriffs- und Anfragemechanismen erfordern.

Unabhängig vom zugrundeliegenden Datenmodell und vom betrachteten Anwendungsgebiet haben sich ausgehend von monolithischen Systemen (wir verstehen hierunter zentralisierte Systeme auf Ein-Prozessor-Rechnern) unterschiedliche neue Einsatzformen von Datenbanksystemen entwickelt. Diese beruhen im wesentlichen auf Entwicklungen im Bereich der Hardware (Workstation-Server-Konfigurationen, Mehrprozessorsysteme, homogene/heterogene Rechnernetze, erweiterte Speicherarchitekturen mit sehr großen und teilweise nicht-flüchtigen Hauptspeichern) und auf der Zunahme verteilter Systeme (insbesondere von Client-Server-Systemen). Entsprechende Datenbanksysteme umfassen verteilte Datenbanksysteme, Client-Server-Datenbanksysteme und parallele Datenbanksysteme.

Die Mannigfaltigkeit der Entwicklungsrichtungen weist auf das weite Spektrum der Anforderungen in bezug auf die Architektur und die Implementierung von Datenbanksystemen hin und zwingt aus Platzgründen zu einer Auswahl des behandelten Lernstoffs. In

diesem Kurs werden wir uns auf Implementierungskonzepte und -techniken konzentrieren, die (im wesentlichen) allen Arten von Datenbanksystemen gemeinsam sind. An einigen Stellen erfolgt eine Einschränkung auf ein bestimmtes Datenmodell. So werden wir uns beispielsweise bei der Behandlung der Anfrageverarbeitung und -optimierung auf das relationale Modell beschränken. Aus Anwendungssicht stehen Standard-Datenbanksysteme im Vordergrund. Allerdings werden wir auch Aspekte diskutieren, die sich speziell und ausschließlich auf Nicht-Standard-Datenbanksysteme beziehen. So besprechen wir bei der Betrachtung von Indexstrukturen auch Indexstrukturen für geometrische Objekte, wie sie in Geo- und CAD-Datenbanksystemen vorkommen. Was die verschiedenen Einsatzformen von Datenbanksystemen anbetrifft, so beschränken wir uns im gesamten Kurs auf monolithische Systeme. Auf Einschränkungen und Spezialisierungen der einzelnen Themenbereiche wird an den entsprechenden Stellen hingewiesen.

Globales Lernziel dieses Kurses ist es, ein grundlegendes Verständnis für die Architektur und die Implementierung von Datenbanksystemen zu entwickeln. Hierbei wird auch Wert auf die Einführung der englischen Fachbegriffe gelegt, die jeweils in runden Klammern angegeben werden. Dies soll den Studenten beim Lesen englischsprachlicher Fachliteratur unterstützen. Manchmal wird auch eine Übernahme eines englischen Fachbegriffs erfolgen, wenn eine deutsche Übersetzung oder ein gleichwertiger deutscher Begriff zu ungenau erscheint.

Der Kurs ist in sieben Kapitel unterteilt. Jedes Kapitel enthält am Ende Literaturhinweise, die dem Leser eine Vertiefung des Stoffs ermöglichen. Kapitel 1 beschäftigt sich mit der Architektur von Datenbanksystemen und den Anforderungen an ihre Implementierung. Die traditionelle Datenorganisation in Dateisystemen mit ihren gravierenden Nachteilen wird den Datenbanksystemen mit ihren Vorteilen gegenübergestellt. Ferner wird ein abstraktes 3-Ebenen-Modell vorgestellt, das heute weitgehend als Grundlage für den Aufbau von Datenbanksystemen gilt. Schließlich wird aus der Sicht des Software-Engineering eine hierarchische und modulare Systemarchitektur für ein Datenbanksystem beschrieben.

Kapitel 2 behandelt das Speichersystem eines Datenbanksystems. Die wesentlichen Aufgaben eines Speichersystems untergliedern sich in die beiden Bereiche der Externspeicherverwaltung und der Systempufferverwaltung. Wesentliches Ziel der Externspeicherverwaltung ist es, die Daten der Datenbank auf einem externen Speichermedium persistent und möglichst speicherplatzeffizient zu speichern und gleichzeitig einen möglichst laufzeiteffizienten Zugriff auf diese Daten zu ermöglichen. Um die Daten einer Datenbank verarbeiten zu können, müssen sie vom Externspeicher in einen bestimmten Bereich des Hauptspeichers, dem Systempuffer, geladen und bei Änderung später wieder vom

Hauptspeicher auf den Externspeicher zurückgeschrieben werden. Um die hierbei auftretenden Probleme kümmert sich die Systempufferverwaltung.

Kapitel 3 führt Indexstrukturen als spezialisierte, externe Datenstrukturen zum effizienten Zugriff auf die Daten einer Datenbank ein, die eine sonst erforderliche sequentielle Suche vermeiden. Nach der Erläuterung des Begriffs und der Aufgaben einer Indexstruktur werden verschiedene Klassifikationen für Indexstrukturen vorgestellt. Anschließend werden aus der Fülle der vorhandenen Indexstrukturen wichtige Vertreter betrachtet. Behandelt werden zum einen Indexstrukturen für alphanumerische Daten wie baumbasierte und hashbasierte Indexstrukturen und zum anderen geometrische Indexstrukturen, die weniger bekannt sind, aber in wesentlichem Maße Bedeutung für die Unterstützung von räumlichen oder geometrischen Datenbanksystemen besitzen. Nach einer kurzen Charakterisierung geometrischer Objekte, Operationen und Anfragetypen sowie der Aufgaben und Eigenschaften geometrischer Indexstrukturen werden als wichtige Vertreter eindimensionale Einbettungen, externe Strukturen für Punktmengen (z.B. das Grid-File) und externe Strukturen für Rechteckmengen (z.B. die R-Baum-Familie) beschrieben.

Kapitel 4 befaßt sich mit dem externen Sortieren großer externer Datenbestände auf Sekundärspeicher anhand eines Sortierschlüssels. Diese Sortierverfahren zeichnen sich im Gegensatz zu internen Verfahren durch einen sequentiellen Datenzugriff aus. Ziel ist es, beim Sortieren die Anzahl externer Seitenzugriffe zu minimieren. Sortieren ist eine wichtige Funktion eines Datenbanksystems und wird z.B. zur Beantwortung von Benutzeranfragen in einer gewünschten Sortierreihenfolge, zur Eliminierung von Duplikaten in einer Menge von Datensätzen, zur Unterstützung der Implementierung bestimmter relationaler Algebraoperationen wie einer bestimmten Join-Variante oder zur Erzeugung von Partitionen durch Zerlegung einer Datensatzmenge in disjunkte Gruppen benötigt. Als wichtigste Vertreter werden verschiedene Mergesort-Varianten betrachtet.

Kapitel 5 hat Transaktionen und Concurrency Control zum Thema. Eine Datenbank stellt eine Informationsbasis dar, auf die von vielen verschiedenen Anwendungsprogrammen und interaktiven Benutzern unabhängig, ohne Wissen voneinander und insbesondere nebenläufig zugegriffen wird. Es ist die Aufgabe des Concurrency Control, den nebenläufigen Zugriff auf geteilte Daten zu steuern und die beteiligten Prozesse zu synchronisieren, um inkonsistente Datenbankzustände zu verhindern. Synchronisation im Datenbankbereich bedeutet Serialisierung konkurrierender Zugriffe auf gemeinsam benutzte Datenbankobjekte. Das grundlegende Konzept für Concurrency Control (und Recovery) ist die Transaktion. Sie bezeichnet eine Folge von zusammengehörenden Operationen, die eine logische Arbeitseinheit bilden und entweder vollständig oder gar nicht ausgeführt werden. Einen breiten Raum nimmt die Darstellung von Sperrverfahren ein, die wohl als das wichtigste Synchronisationsmittel für Transaktionen in Datenbanksystemen anzusehen sind.

Kapitel 6 befaßt sich mit denjenigen Funktionen eines Datenbanksystems, die die Wiederherstellung eines korrekten Datenbankzustands nach dem Auftreten eines Datenbankfehlers ermöglichen und die Unteilbarkeit und Dauerhaftigkeit von Transaktionen sicherstellen. Unteilbarkeit wird erreicht, indem die Operationen aller nicht beendeten Transaktionen abgebrochen und rückgängig gemacht werden. Dauerhaftigkeit wird bewirkt, indem sichergestellt wird, daß alle Operationen von beendeten Transaktionen Fehler überleben. Die Wiederherstellung des letzten konsistenten Datenbankzustands wird auch Recovery genannt. Verschiedene Recovery-Techniken werden erläutert und Algorithmen hierzu im Rahmen der in Datenbanksystemen weit verbreiteten Log-basierten Recovery vorgestellt.

Kapitel 7 behandelt die Verarbeitung von Anfragen. Die Möglichkeit, mit Hilfe einer Anfragesprache Anfragen an eine Datenbank zu stellen, sei es durch ein Anwendungsprogramm oder aber ad hoc durch den Endbenutzer am Computer, und unmittelbar darauf eine Antwort zu erhalten, gehört mit zu den herausragenden Eigenschaften eines Datenbanksystems. Ziel der Anfrageverarbeitung ist es, eine gegebene Anfrage unter Ausnutzung logischer Gesetzmäßigkeiten und externer physischer Speicherungsstrukturen möglichst effizient auszuführen. Der Übersetzung einer Anfrage (lexikalische, syntaktische und semantische Analyse) folgt deren Optimierung und Ausführung. Die Optimierung einer Anfrage ist erforderlich, um aus der Menge der möglichen Ausführungsalternativen für eine Anfrage möglichst eine mit minimalen Kosten auszuwählen. Nach einer Anfrageumformung auf der Basis von Äquivalenzregeln (algebraische Optimierung) werden verschiedene Auswertungspläne erzeugt, die mit Hilfe der Kostenschätzung bewertet werden und zum letztendlich „optimalen“ Auswertungsplan führen. Eine effiziente Anfrageauswertung hängt natürlich insbesondere von der physischen Speicherung der Daten ab. Vorhandene Indexe können die Anfrageverarbeitung beträchtlich beschleunigen. Am Beispiel relationaler Operationen werden verschiedene Implementierungs- und Auswertungsmöglichkeiten aufgezeigt.

Der Kurs wendet sich an Studenten, die bereits Erfahrungen mit Datenbanksystemen haben und mehr über die Interna dieser Systeme erfahren möchten. Hilfreich sind allgemeine Kenntnisse über Datenbanksysteme, die man zum Beispiel über das Studium des Kurses 01665 „Datenbanksysteme“ erlangen kann. Studenten, die sich für verschiedene Einsatzformen von Datenbanksystemen interessieren, sei der Kurs 01666 „Datenbanken in Rechnernetzen“ empfohlen. Hier werden Fragestellungen verteilter Datenbanksysteme und Client-Server-Systeme behandelt.

Markus Schneider