Sonderdruck aus Nr. 20
vom 19. Mai 2006

www.computerwoche.de

COMPUTERWOCHE

NACHRICHTEN ◆ ANALYSEN ◆ TRENDS

IDG

# Was Datenbanken leisten sollten

*Datenbanken galten lange Zeit als „commodity". Doch die gewachsenen Anforderungen an Datenintegration, der Umgang mit diversen Datenstrukturen und Verfügbarkeit verlangen den Herstellern einen Kraftakt ab.*

VON CHRISTIAN ANTOGNINI UND URS MEIER*

Um die aktuelle Entwicklung im Bereich der Informationssysteme zu verstehen, ist es sinnvoll, einen kurzen Blick auf deren Werdegang zu werfen. In den Anfängen der Computertechnik ging es im Wesentlichen um Number Crunching, das Zeitalter der Berechnungen sozusagen. In den späten 60er Jahren kam dann der erste Paradigmenwechsel: Dank einem neuen Typ Software, den Datenbank-Management-Systemen, konnte man beliebige Informationen speichern und wieder abrufen. Das Zeitalter der Information war angebrochen.

### Aus Information wird Wissen

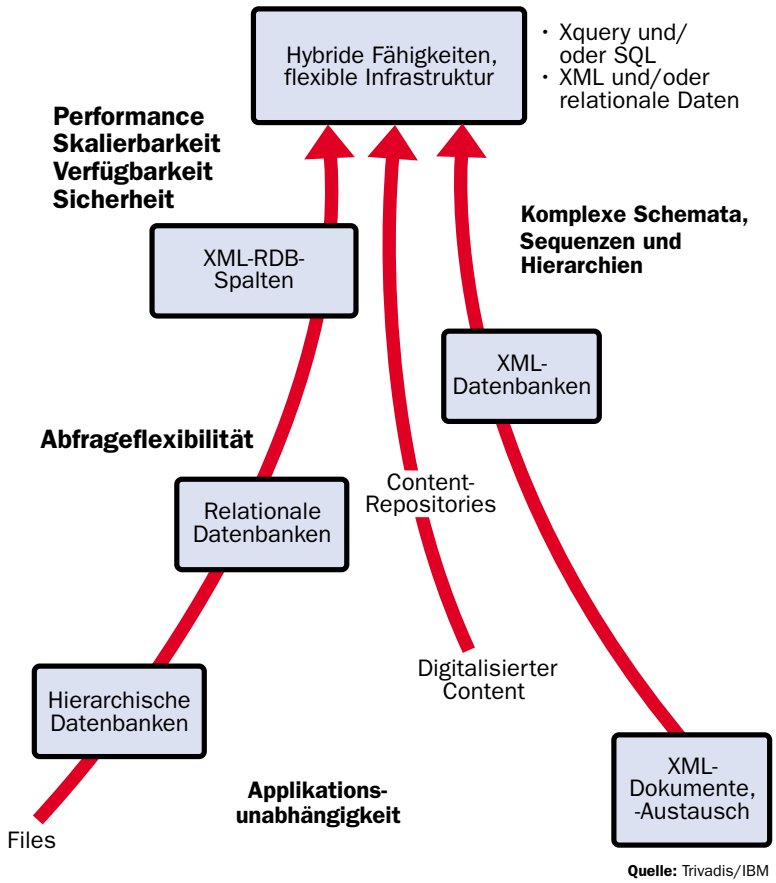
In den 90er Jahren wurde klar, dass ein weiterer Paradigmenwechsel erfolgen musste, um den ganzen Nutzen aus den gespeicherten Daten zu ziehen. Man stellte Beziehungen zwischen existierenden Datenfragmenten her, um nicht nur bestehende Daten abzufragen, sondern neues Wissen herzuleiten: das Zeitalter des Wissens also. Heute speichern wir Datenmengen, die um ein Vielfaches umfangreicher sind als noch vor zehn Jahren. Um daraus aber einen echten Nutzen zu ziehen, müssen wir Techniken etwa zum Data Mining entwickeln und erweitern, Daten verdichten und überschaubar präsentieren können.

### Datenintegration

Um neues Wissen aus bestehender Information herzuleiten, ist es normalerweise notwendig, mehr als eine Datenbank heranzuziehen. Denn eine typische Datenbank ist normalerweise auf nur einen speziellen Teil des Geschäfts beschränkt. Man muss also in der Lage sein, diese Teile zu integrieren. Grundsätzlich sind hier drei Wege möglich:

Der erste besteht darin, die Daten physisch von den Quell-Datenbanken in eine zentrale Datenbank zu kopieren, die auf eine Wissensgewinnung spezialisiert ist. Gemeinhin wird eine solche Datenbank als Data Warehouse bezeichnet, der Daten-

## Für jede Datenstruktur




Quelle: Trivadis/IBM

**Ein Hauptmerkmal von DB2 Viper ist die vollständige Einbindung von nativem XML in die Datenbank.**

# Die Highlights der Großen

## IBM DB2 Viper

- Native XML-Datenbank;
- Speicherung größerer Datenmengen bei gleichzeitiger Erweiterung des Self-Managements;
- Zweigleisigkeit bei den Entwicklungsumgebungen: Open Source und Microsoft-Welt;
- Datenintegration in die Websphere-Produktlinie ausgelagert.

## Microsoft SQL Server

- Optionen zur Hochverfügbarkeit von Daten und Diensten;
- starkes Gewicht auf Self-Tuning und Self-Healing;
- Offline-Fähigkeiten und Anpassung an unterschiedlichste Speicherbedürfnisse durch den Entity Data Bus;
- ganzheitliche Betrachtung einer Anwendung vom Client bis zum Datenbank-Server.

## MySQL

- Offene Architektur für Datenimport/-export;
- umfangreicher Multi-Language-Support;
- Geo-Datenhaltung und Textsuche;
- Cluster-Technik, die eine Daten- und Diensteverfügbarkeit erhöht und Skalierbarkeitsanforderungen erfüllt.

## Oracle Database

- Datenintegration mit transportablen Tablespaces, Datenpumpe und Datenreplikation per Streams;
- Storage Engines und Abfrageunterstützung für XML, Text, Geodaten, Video, Bilder, Ton sowie Toolkit zur Implementierung kundenspezifischer Storage Engines;
- Daten- und Dienstehochverfügbarkeit integriert oder als Option, Skalierbarkeit bis in den Terabyte-Bereich.

transfer erfolgt im Rahmen eines so genannten ETL-Vorgangs (Extrahieren, Transformieren, Laden). Hauptvorteil ist hier die Datenkonsistenz im Zielsystem, die während des Ladevorgangs gesichert werden kann, sowie der schnelle Datenzugriff bei der Wissensherleitung aus den integrierten Daten, unabhängig vom Quellsystem. Negativ anzumerken sind dabei die Duplizierung von Daten und die Abhängigkeit von einem periodischen Ladevorgang, der den Realtime-Zugriff auf die Basisdaten unmöglicht macht.

Der zweite Ansatz beinhaltet den bedarfsorientierten Zugriff auf die Daten der Quellsysteme. Anders gesagt: Die Datenintegration erfolgt über eine Softwareschicht, also nicht durch physisches Kopieren. Vorteilhaft hierbei ist, dass die Daten jeweils nur einmal existieren und Synchronisationsprobleme deshalb wegfallen. Nachteilig wirkt sich jedoch aus, dass die Abhängigkeiten zwischen den Systemen zunehmen, der Datenzugriff langsamer wird und allgemein die Datenkonsistenz über mehrere Systeme hinweg schwer einzuhalten ist.

Datenreplikation als dritte Möglichkeit ist der Kompromiss der beiden Strategien und vereint deren Stärken. Die Daten sind logisch nur einmal vorhanden, für den schnellen Zugriff aber trotzdem physisch verteilt. Das Problem hier: Replikation ist entsprechend komplexer aufzusetzen und zu betreiben.

### Hybride Datenhaltung

Mit dem Übergang in das Wissenszeitalter haben sich auch die Datenbankinhalte geändert. Wurden früher nur Zahlen und allenfalls in der Größe beschränkte Zeichenketten gespeichert, so sind diese Zeiten definitiv vorbei. Aktuelle Systeme müssen in der Lage sein, strukturierte (Zahlen, Zeichenketten und Records) wie auch in ihrer Speicherform unstrukturierte Daten (Textdokumente, Bilder, Tondokumente, geografische Informationen) zu speichern und nach bestimmten Inhalten zu durchsuchen. Es ist wichtig zu erwähnen, dass auch in spezialisierte Speichersysteme wie zum Beispiel Dokumentenablagen die darüber liegende Abfragesprache „hineinschauen“ können muss und nicht nur einen großen Binärstring an die Applikation weitergeben darf, um dieser einen wesentlichen Teil der Arbeit aufzubürden. Da ein Hersteller niemals alle Speicherbedürfnisse voraussehen kann, sollte die Möglichkeit gegeben sein, eigene Speichersubsysteme zu bauen und in die Abfragesprache zu integrieren.

Drei weitere Forderungen an moderne Datenbanken ergeben sich aus der Globalisierung der Märkte sowie der zunehmenden Anzahl der Business-to-Customer-Dienste (B2C), die oftmals auch über mobile Geräte nutzbar sein müssen. In Sachen Verfügbarkeit heißt es hier eindeutig: keine Downtime!

Verfügbarkeit meint aber nicht nur, dass das System erreichbar ist, sondern dass es den Dienst auch mit einer akzeptablen Performanz zur Verfügung stellt. Da der Benutzer eine langsam antwortende Website einfach wegklickt, sind nicht performante Systeme gleichzusetzen mit nicht erreichbaren Systemen.

Bei ständig wachsender Anzahl der Nutzer verdoppelt sich die Menge der gespeicherten Daten praktisch von Jahr zu Jahr. Terabyte-Datenbanken sind heute keine Seltenheit mehr, in ein paar Jahren werden sie die Regel sein. In diesem Umfeld ist die Skalierbarkeit ein absolutes Muss. Dies gilt natürlich nicht nur für die Datenbank, sondern für die gesamte Applikation und die Infrastruktur.

### Die Herausforderungen

Die drei Hauptanforderungen an die Datenbanksysteme der Zukunft lauten also Datenintegration, hybride Datenhaltung sowie Verfügbarkeit bei hoher Performanz und Skalierbarkeit. Wie die namhaften Datenbankhersteller diesen Ansprüchen begegnen, lesen Sie in den nachfolgenden Beiträgen. *(ue)* ◆

*__CHRISTIAN ANTOGNINI__ und __URS MEIER__ sind leitende Consultants und Partner bei der Trivadis AG in Zürich. Die Autoren der Beiträge auf den nachfolgenden Seiten sind ebenfalls Trivadis-Consultants und auf die jeweiligen Produkte spezialisiert.

# Microsofts Pläne für SQL Server

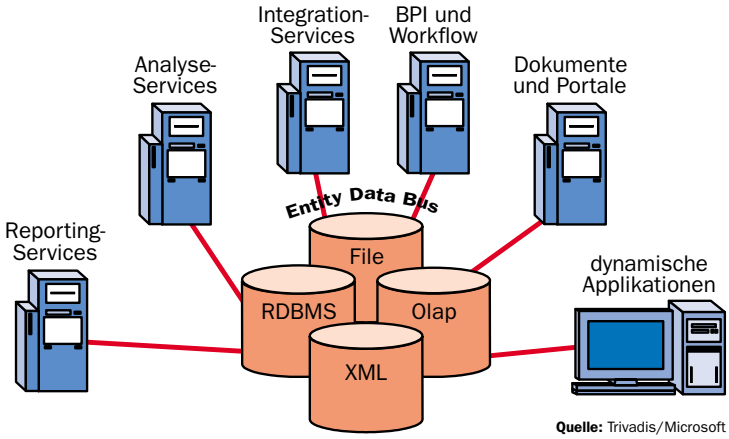
Microsofts Vision im Bereich Daten lässt sich mit dem Slogan „Your Data, Any Place, Any Time“ zusammenfassen. Dieses Ziel korreliert in vielen Bereichen mit den vorher beschriebenen allgemeinen Trends. Besonderen Handlungsbedarf sieht der Hersteller in Bezug auf höchste Verfügbarkeit, automatischen Betrieb, Integration von unterschiedlichsten Datenformaten und Datenservices. Die im Folgenden aufgeführten neuen Funktionen sollten in den nächsten 24 bis 36 Monaten zur Verfügung stehen.

### Defizite beim Self-Management

SQL Server 2005 bietet bereits heute viele Möglichkeiten, um große Datenmengen effizient zu verwalten und die Verfügbarkeit der Datenbank sicherzustellen. Dazu gehören File-Groups, Partitions, Database-Mirroring, Failover-Cluster, Database-Snapshots und Replikation. Damit die Verfügbarkeit der Datenservices noch weiter erhöht werden kann, ist es notwendig, dass Systeme frühzeitig Fehler erkennen und darauf selbständig reagieren können. Deshalb arbeitet Microsoft mit Nachdruck an SQL-Server-Features wie Self-Tuning, Self-Organizing und Self-Maintaining. Ein weiteres Ziel dabei ist die Verbesserung der Total Cost of Ownership (TCO).

Die Integration von unterschiedlichen Datenformaten und -services überlässt Microsoft nicht nur der Datenbank, sondern bemüht dafür eine ganze Datenplattform, wie auch ADO.NET als Bindeglied zwischen dem Daten- und Applikations-Layer. Mit der neuen Datenzugriffstechnik LINQ verschwimmen die Grenzen der unterschiedlichen Datenrepräsentationen. Dadurch kann mit einem einheitlichen Programmiermodell auf die unterschiedlichen Datenstrukturen (Relational, Objekte, XML etc.) zugegriffen werden.

## Nicht nur Datenbank




**Quelle:** Trivadis/Microsoft

**Zur Integration unterschiedlicher Datenformate bemüht Microsoft über den Entity Data Bus die gesamte Plattform.**

### Dynamische Applikationen

Der so entstehende „Entity Data Bus“ bildet die Basis für die kommenden „Dynamic Applications“. Diese zeichnen sich durch ihre hohe Anpassungsfähigkeit an sich verändernde Geschäftsbedürfnisse aus. Microsoft geht davon aus, dass diese Applikationen nicht immer mit dem Netzwerk verbunden sind, und bietet daher Möglichkeiten für die lokale Datenhaltung. Neben SQL Server Express Edition wird auch SQL Server Everywhere auf den Markt kommen. Die Everywhere-Version braucht noch weniger lokale Ressourcen als die restlichen SQL-Server-Varianten, ist aber dank ADO.NET und LINQ aus Applikationssicht identisch zu programmieren und bietet Replikations- und Synchronisations-Möglichkeiten. Sie ist vor allem dazu gedacht, dass auf dem Client diejenigen Daten gehalten werden können, die einen Offline-Betrieb ermöglichen.

Die Business-Intelligence-Fähigkeiten von SQL Server wird Microsoft weiter ausbauen. Dabei hat man den ganzen Stack von der relationalen Datenbank über die Analysis Services bis hin zur Client-Applikation (Excel, Entwicklungswerkzeuge) im Auge.

**Meinrad Weiss**

---

# MySQL hat aufgeholt

Die schwedische Firma MySQL misst der Datenintegation eine große Bedeutung zu. So kann man Tabellen in herstellerunabhängige Dateien exportieren, was den Ladeprozess in andere Systeme vereinfacht. Der umgekehrte Weg ist natürlich auch möglich, ebenso können Daten direkt aus Komma-separierten Dateien (CSV-Dateien etwa von Excel erzeugt) in die Datenbank gelesen werden. Bleibt man in der MySQL-Welt, so können aufgrund der dateiorientierten Speicherform auf einfache Art und Weise Tabellen von einem Datenbanksystem in das andere kopiert werden. Will man erweiterte Funktionalität – wie im BI-Umfeld benötigt – nutzen, so stehen die Produkte des Open-Source-Projekts Pentaho zur Verfügung.

### Hilfreiche Features

Bei der Speicherung komplexer Datenstrukturen ist MySQL nicht so weit fortgeschritten wie andere relationale Datenbank-Management-Systeme (RDBMS). Auf der anderen Seite erlaubt es MySQL, das Codierungsschema einer Zeichenkette und den Vergleichsmodus in praktisch beliebigen Granularitätsstufen auf Server-, Datenbank-, Tabellen- oder Attributsebene zu spezifizieren. Ein Feature, das in multilingualen Applikationen extrem hilfreich ist. MySQL unterstützt neben der Speicherung der klassischen skalaren Datentypen auch die von Large Binary Objects (LOBs), Volltextsuche und Spatial Datatypes für geografische Informationssysteme. Im Hochverfügbarkeitsbereich stehen mehrere Optionen zur Verfügung. Replikation erlaubt das Nachführen von Standby-Systemen mittels der Übertragung von SQL-Befehlen – ein Konzept, das nur geringen Verkehr zwischen den Datenbanken verursacht. MySQL Cluster hingegen ist eine echte Active-Active-Lösung, wobei mehrere Knoten die Daten für die Clients dupliziert halten. Im Gegensatz zum Shared-Disk-Ansatz ist MySQL Cluster eine Shared-Nothing-Architektur, die in der von MySQL gewählten Form für Daten- und für Dienstehochverfügbarkeit sorgt. Zudem kann das Cluster für Skalierungsanforderungen eingesetzt werden.

Der Skalierungsansatz von MySQL basiert also auf dem Scale-out-Prinzip – viele kleine Server statt weniger großer SMP-Maschinen. Aktuell besteht in Version 5.0 aber noch das Problem, dass beim Hinzufügen neuer Knoten die vorhandenen Daten neu verteilt werden müssen. Damit die einzelnen Knoten auch bei großen Datenmengen genügend Hauptspeicher haben, sind 64-Bit-Systeme faktisch unverzichtbar.

Als Open-Source-Produkt ist bei MySQL der volle Funktionsumfang umsonst. Man kann auf Kostenbasis professionelle Supportverträge abschließen, wobei die Qualität auf Topniveau liegt – eine positive Erfahrung, die traditionelle RDBMS-Hersteller nicht immer verschaffen können.

**Yann Neuhaus**

# Oracle 10g auf hohem Niveau

Oracle konnte im aktuellen Release 10g, Release 2, an eine lange Tradition der Datenintegration anschließen und baut seine Sichtweise des Grid Computing konsequent aus. Daten können physisch integriert werden, sei es durch Kopieren der Datenbankdateien auf OS-Ebene und nachfolgendes „Einhängen“ in das Ziel-Datenbanksystem, sei es durch parallelisiertes Laden aus einem speziellen Speicherformat mit der Data Pump (auch zwischen Datenbanken möglich) oder aber durch Laden aus herstellerneutralen Textdateien, welche aus der Datenbank wie Tabellen angesprochen werden können („External Tabellen“). Der verteilte Ansatz wird durch den SQL-Zugriff via Datenbank-Links unterstützt und durch spezielle Gateways auch auf Datenbanksysteme anderer Hersteller. Der Kompromiss der beiden Ansätze, die Datenreplikation, existiert in einer bewährten Form seit langem, wird aktuell aber ergänzt und mittelfristig abgelöst durch das Streams-Konzept.

Die hybride Datenhaltung ist bei Oracle seit Jahren Standard. Sie wird einerseits durch spezielle Engines und Abfragesprachen-Erweiterungen unterstützt, andererseits werden neue Bedürfnisse wie die Ablage von XML-Dokumenten mit Hilfe der objektrelationalen Erweiterungen implementiert.

### Für Desaster gewappnet

Oracle hat sein Flaggschiff für Hochverfügbarkeit, Real Application Cluster, zu einem Skalierungswerkzeug weiterentwickelt. Daneben wurde im Wissen, dass ein hochverfügbarer Dienst auch hochverfügbare Daten benötigt, die Unterstützung von Standby-Datenbanken mit der integrierten Data-Guard-Option so weit verbessert, dass praktisch alle denkbaren Desaster-Szenarien abgefangen werden können. Multiprozessor-Systeme mit Dutzenden CPUs sind für Oracle und dessen parallelisierbare Queries kein Hindernis mehr. Stimmen Hardware und Applikation, skaliert die Datenbank in den TB-Bereich. **Martin Wunderli**

# DB2 Viper wird hybrid

Für das dezentrale Umfeld (Linux, Unix und Windows) hat IBM unter dem Codenamen „Viper“ eine neue Version von DB2 angekündigt, die in der zweiten Hälfte dieses Jahres verfügbar sein wird. Ein Hauptmerkmal von „Viper“ ist die vollständige Einbindung von nativem XML in die Datenbank. So wird es möglich, Daten dem neu geschaffenen Datentyp XML zuzuweisen und zusammen mit relationalen Daten in derselben Datenbank abzulegen. Dies geschieht, ohne dass die XML-Dokumente zuvor zerlegt oder in Large Objects (LOBs) umgewandelt werden müssen.

Um die hierarchischen und relationalen Strukturen zu bearbeiten, kann mittels SQL, XQuery oder neuer SQL/XML-Funktionen auf beide Strukturen zugegriffen werden. Durch den neuen Datentyp lässt sich auch jedes beliebige XML-Tag innerhalb einer XML-Struktur indexieren, was sich wiederum sehr positiv bei Suchvorgängen auswirkt.

Die stetig wachsenden Datenmengen führten dazu, dass zwei weitere Features implementiert wurden, die in der DB2-Version für z/OS schon lange zum Funktionsumfang gehören. Mit dem Range Partitioning ist erreicht worden, dass die Skalierungsgrenzen etwa bezüglich maximaler Tabellengrößen um ein Vielfaches nach oben verschoben wurden. Über die Komprimierung von Daten erhofft man sich eine bessere Ausnutzung des Speicherplatzes und eine erhebliche Verbesserung der I/O-Performance.

### Automatische Verwaltung

Der bereits in der Vorgängerversion zu erkennende Trend zur Automatisierung hält an. Das Konzept der automatischen Ressourcenverwaltung ist erweitert und verbessert worden. Neu ist, dass es nun einen „Self Tuning Memory Manager“ gibt, welcher zur Laufzeit den Arbeitsspeicherbedarf der jeweiligen Auslastung des DB-Servers anpasst.

Bei den Plattformen ist zu sehen, dass sich in nächster Zeit DB2 auf dem Mainframe weiter eisern halten wird. Im dezentralen Bereich kann es zu einer Verlagerung von Unix auf Linux kommen, und in Bezug auf die OS-Architektur geht es ganz klar in Richtung 64 Bit. Für Windows-Anwender ist interessant, dass nun mehrere DB2-Instanzen und/oder unterschiedliche Fixpack-Versionen parallel laufen können. Auch eine Fixpack-Deinstallation sollte wie unter Unix möglich sein.

Zur Unterstützung der Anwendungsentwicklung ist die Eclipse-basierende „DB2 Developer Workbench“ integriert. Sie ersetzt das bisherige „DB2 Development Center“. Für alle Editionen gilt: Sie haben dieselben APIs, arbeiten mit demselben SQL und unterliegen demselben Deployment. Nebst den diversen Open-Source-Projekten setzt IBM aber auch weiter stark auf den Support der Microsoft-Welt zum Beispiel für Visual Studio 2005. Parallel dazu wird es auch im Bereich der Datenintegration einige Neuerungen geben. Diese fließen allerdings in IBMs Websphere Information Integrator ein. **Stefan Buess**