



## Datenbanksysteme – von den Anfängen bis hin zu MySQL

*Sommerzeit. Urlaubszeit. Im Internet Flüge buchen, die passende Reiselektüre bestellen und auf Wikipedia Informationen zum Urlaubsort sammeln. Verschiedenste Aktivitäten im Netz, doch haben sie technisch gesehen eines gemeinsam: Sie alle basieren auf Datenbanksystemen<sup>1</sup>. Eines der beliebtesten ist MySQL und seine Geschichte reicht bis in die 1950er Jahre zurück.*

### Der Ursprung

Die wichtigste Grundlage zur Entwicklung von Datenbanken war Mitte der 50er Jahre die Erfindung der Festplatte. Das erste Festplattenlaufwerk hatte noch die Dimensionen eines Kleiderschranks, wog 500 kg und besaß bei dieser enormen Größe nur eine Speicherkapazität von 5 MB.

Aus heutiger Sicht sind solche Ausmaße für ein Speichermedium dieser Aufnahmefähigkeit nur noch schwer vorstellbar. Mit dem Einsatz immer neuerer Technologien verringerte sich die physische Größe von Festplatten bei steigender Speicherkapazität.

Wenige Jahre nach der Erfindung der Festplatte – Anfang der 60er Jahre – wurden erste Anwendungen zur Massendatenverarbeitung umgesetzt. Die Organisation der einzelnen Informationen dafür erfolgte auf der Festplatte in einfachen Dateien.

Datenredundanzen (Inhalte, die mehrfach vorhanden sind) und -inkonsistenzen (Widersprüchlichkeit zwischen Daten) waren dabei kaum zu vermeiden. Sämtliche Dateien waren auf einer Ebene gespeichert und die entsprechenden Mitarbeiter mussten den Überblick über Tausende von Dateien behalten, denn eine Baumstruktur mit Unterverzeichnissen – wie sie heute üblich – gab es zu dieser Zeit noch nicht.

Dieses Problem wurde durch die Einführung von Dateiverwaltungssystemen gelöst: Ein Meilenstein auch in der Entwicklung von Datenbanksystemen. Die einzelnen Dateien konnten nun in separaten Verzeichnissen gespeichert und verwaltet werden. Damit war eine Übersicht über Tausende von Dateien im selben Verzeichnis nicht mehr erforderlich. Systemdateien waren in anderen Verzeichnissen abgelegt als Dateien der einzelnen Benutzer.

---

<sup>1</sup> Ein Datenbanksystem besteht aus zwei Komponenten:

- Datenbank (strukturierter Datenbestand, der von einem Datenbankmanagementsystem verwaltet wird)
- Datenbankmanagementsystem, abgekürzt: DBMS (Software zum Verwalten von Datenbanken)



Einige Jahre später kamen die ersten Datenbanksysteme – basierend auf dem Modell der Dateisysteme – zum Einsatz. Nun war eine Datenhaltung frei von Redundanzen und Inkonsistenzen möglich. Niemand brauchte mehr befürchten, dass wegen eines Datenbankfehlers beispielsweise im Flugzeug ein Platz doppelt gebucht wurde oder dass ein Platz aus einem ähnlichen Grund nicht besetzt war.

Schon in frühen Stadien von Datenbanksystemen erfüllten sie Anforderungen wie

- Datenintegration,
- Konsistenzüberwachung,
- Operationen [suchen oder ändern],
- Zugriffskontrolle und
- Datensicherung.

Diese Vorgaben gehen auf den britischen Mathematiker und Datenbanktheoretiker Edgar Frank Codd [1923 – 2003] zurück. Er definierte Regeln für sichere Datenbanken sowie Grundlagen für sichere Datenhaltung und Abfragen.

## Verschiedene Konzepte

Die ersten Datenbanksysteme basierten auf einem hierarchischen Konzept. Das primäre Kennzeichen einer jeden Hierarchie ist die pyramidenförmige Rangordnung. Innerhalb der Datenbanken wird diese Beziehung [Verknüpfung] in einer Baumstruktur bzw. in sogenannten Parent-Child-Verhältnissen dargestellt. Diese Baumstruktur kann man sich genauso vorstellen wie die Verzeichnisstruktur auf dem heimischen Computer.

Innerhalb des hierarchischen Datenbankmodells gab es entscheidende Nachteile: Es war nicht möglich, mehrere dieser Baumstrukturen oder mehrere Ebenen desselben Baums zu verknüpfen. Diese Datenbanken waren sehr eng mit der Hardware verknüpft und jede Änderung, z. B. das Hinzufügen eines Datenfeldes, erforderte einen enorm hohen programmiertechnischen Aufwand.

Um diese Unzulänglichkeiten zu überwinden, begann die Suche nach neuen Datenbankkonzepten. Einer der führenden Köpfe war der bereits erwähnte Edgar Frank Codd<sup>2</sup>. Er entwickelte das relationale Modell, welches bis heute aktuell und marktbeherrschend ist.

---

<sup>2</sup> Das Sprichwort „Der Prophet gilt nichts im eigenen Land“ trifft in vollem Umfang auf Codd zu. Während Universitäten und andere Forschungseinrichtungen nach seinem Modell bereits eifrig experimentierten, wurden seine bahnbrechenden Ideen von seinem damaligen Arbeitgeber IBM nur äußerst mühsam angenommen.



Codds größte Leistung lag auf konzeptioneller Ebene: Seine Idee, die Datenbankorganisation und den Aufbau der Datenstruktur von der physikalischen Speicherung zu trennen, war vollkommen neu und machte Datenbanken wesentlich flexibler. Änderungen wie das vorher erwähnte Hinzufügen von Datenfeldern konnten nun mit minimalem Aufwand umgesetzt werden.

Relationale Datenbankmanagementsysteme (RDBMS) verwalten ihre Daten in Tabellen einer flachen Struktur. Diese Tabellen werden in der Fachwelt Relationen genannt, woher auch der Name dieses Modells stammt. Die Daten können verlustfrei und beliebig miteinander verknüpft werden, was eine enorme Flexibilität gewährleistet.

Der nächste wichtige Entwicklungsschritt innerhalb von Datenbanksystemen ist das objektorientierte Datenbankmanagementsystem bzw. die Objektdatenbank. Bei diesem Konzept werden die einzelnen Daten als Objekte mit bestimmten Eigenschaften abgespeichert. Dieses Datenbankkonzept wurde aus dem Modell der objektorientierten Programmiersprachen entwickelt. Auch hier weist der Name auf den gedanklichen Vorläufer hin.

Ein großer Vorteil dieses Modells ist die Möglichkeit, zusammenhängende Objekte auch zusammen abzuspeichern. Objektorientierten Datenbanken kennen demnach semantische Verknüpfungen. In der Kommunikation zwischen Datenbank und Anwendung sind bei diesem Konzept zahlreiche Zwischenschritte überflüssig, was mögliche Fehler auf ein Minimum reduziert.

Dieser Datenbanktyp ist bisher noch wenig verbreitet, was wiederum die Kommunikation mit diversen Anwendungen erschwert, denn viele Schnittstellen sind noch nicht auf Objektdatenbanken ausgerichtet. Aufgrund der komplexen Struktur der Datenbankobjekte und den noch fehlenden Schnittstellen zu den entsprechenden Anwendungen sind momentan noch diverse Zwischenschritte als Überbrückung notwendig. Die Folge sind Performance-Einbußen. Diese können zwar mit leistungsfähigen Systemen abgefangen werden; dies ist aber nur bis zu einem gewissen Grad möglich.

## **MySQL – flexibel, leistungsfähig und zuverlässig**

Schwedische Produkte, z. B. von IKEA, Volvo oder Saab, sind bekannt und beliebt. Ein weiteres Erzeugnis aus Schweden hat ebenfalls eine große Fangemeinde: Das Datenbanksystem MySQL. Warum entscheiden sich so viele professionelle Anwender für dieses System? Die Projekt-Webseite zu MySQL nennt folgende Gründe:

- Flexibilität,



- Leistungsfähigkeit und
- Zuverlässigkeit.

Aber was bedeutet das genau? MySQL gehört wie Oracle oder DB2 zu den relationalen Datenbanksystemen, die im Hinblick auf Skalierung bereits als sehr flexibel beschrieben wurden. MySQL kann leicht und komfortabel auf die Bedürfnisse des jeweiligen Projektes angepasst werden. Sollten sich während des Betriebs die Anforderungen ändern, lässt sich MySQL problemlos skalieren.

Dieses Datenbanksystem zeigt die Flexibilität auch in den unterstützten Betriebssystemen: Nach Angaben der Projektseite läuft das Datenbanksystem auf über 20 verschiedenen. Die Unterstützung des Betriebssystems ist wichtig, da es die Grenzen für die Datenbank durch verschiedene Parameter setzt, die beispielsweise den verwendeten Speicher, die möglichen, gleichzeitig offenen Ports und vieles mehr festlegen. Je größer diese Parameter dimensioniert werden können, z. B. bei einem sehr leistungsfähigen Server, umso mehr Daten kann MySQL verwalten.

Auch im High-End-Bereich macht MySQL in Sachen Leistungsfähigkeit und Zuverlässigkeit eine gute Figur. Eine wichtige Eigenschaft dabei ist die Ausfallsicherheit. Diese wird bei zahlreichen Großunternehmen durch Cluster<sup>3</sup> realisiert.

Ein Cluster hat hinsichtlich der Verwaltung besondere Anforderungen an die Software – in unserem Fall an das Datenbanksystem – und auch an die Administration. Vergleichbar dazu ist eine Person, die sich ganz anders organisiert, wenn sie den Tag allein frei gestalten kann [Stand Alone] oder sich mit beispielsweise der ganzen Familie abstimmen muss [Cluster].

MySQL kann sowohl auf einem einzigen Server als auch im Clusterverbund eingesetzt werden und zeigt sich dadurch als leistungsfähiger, zuverlässiger Partner.

## In guter Gesellschaft

Das Datenbanksystem MySQL tritt in der Praxis häufig in Kombination mit anderen Produkten auf. Ein solches Konglomerat ist unter dem Akronym LAMP bekannt. Es steht für

- Linux [Betriebssystem]
- Apache [Webserver]
- MySQL [Datenbanksystem]
- PHP [Programmiersprache]

---

<sup>3</sup> Ein Cluster ist ein Zusammenschluss von mehreren Rechnern, um beispielsweise die Ausfallsicherheit zu steigern (High Availability, HA-Cluster).



In der Praxis sieht das folgendermaßen aus: Ein Anwender gibt beispielsweise in seinem Browser eine URL [Uniform Resource Locator] ein. Diese Anfrage erreicht den Apache-Webserverdienst auf einem Linux-Rechner. Mittels PHP-Skripten werden die Informationen aus der MySQL-Datenbank extrahiert. Das Ergebnis – die zusammengestellte Webseite – sendet der Webserver an den anfragenden Browser zurück. Der Anwender kann nun die Inhalte der Webseite ansehen.

Das Konglomerat LAMP ist nicht auf exakt diese Produkte festgelegt, auch wenn sie am häufigsten in dieser Form vorkommt. Jede Komponente ist durch eine entsprechende austauschbar, z. B. kann statt PHP Python oder Perl eingesetzt werden.

Neben LAMP existieren unter anderem MAMP für Mac oder WAMP für Windows.

## Aus der Praxis

Wo wird MySQL – beispielsweise in der Kombination LAMP – eingesetzt?

Eine gute Bekannte eröffnet den Reigen der Beispiele: die freie Enzyklopädie Wikipedia. Die Wiki-Software, auf der Wikipedia basiert, heißt MediaWiki. Sie wurde von Magnus Manske speziell für Wikipedia entwickelt. Die erste Version kam im Jahr 2002 das erste Mal zum Einsatz und verwendet die Kombination LAMP.

An nächster Stelle: Das Hessische Statistische Landesamt verwendet MySQL für die Speicherung und Auswertung von Hochschuldaten im Rahmen der DHstat [Datenauswertung Hochschulstatistik]. Es werden zahlreiche Informationen zu Studierenden, Angestellten, Gasthörern und Prüfungsergebnissen gesammelt und verarbeitet. Die Ergebnisse dienen beispielsweise der Finanzmittelzuweisung des Landes an die einzelnen Einrichtungen.

Und schließlich: Die Baumarktkette OBI setzt ebenfalls MySQL ein. Seit der Einführung ihres neuen Warenwirtschaftssystems werden sämtliche Stamm- und Bewegungsdaten in einer MySQL-Datenbank verwaltet und auf die jeweiligen Kassensysteme verteilt [repliziert]. Zur Anwendung kommt hier das Master-Slave-Verfahren: Ein zentraler Server, auf dem die MySQL-Datenbank läuft, stellt den sogenannten Master dar. Dieser sendet in regelmäßigen Abständen die Veränderungen der Datenbankinformationen an die jeweiligen Kassensysteme [Slaves].

## Fazit

Die Flexibilität, Leistungsfähigkeit und Zuverlässigkeit von MySQL machen dieses Datenbanksystem zu einem geschätzten Produkt im professionellen wie auch im privaten Bereich. Neben den genannten Features gestalten sehr gute Administrationstools und



zahlreiche andere Hilfen den Umgang mit diesem Produkt komfortabel und anwenderfreundlich. Auch die ADACOR setzt das Datenbanksystem MySQL und die Kombination LAMP in vielen Projekten erfolgreich ein. Die proprietäre Zukunft von MySQL ist jedoch ungewiss. Aktuell gehört das Unternehmen der Firma Oracle.

### Relationale Datenbankmanagementsysteme (RDBMS) im Vergleich:

RDBMS	Lizenz	Vorteile	Nachteile
DB2	proprietär	weit verbreitet alle gängigen Serverplattformen werden unterstützt ist auch als eingebettete [embedded] Variante verfügbar	ohne Wartungsvertrag gibt es Einschränkungen, z. B. fehlende Replikation hoher Administrationsaufwand für Einsteiger ungeeignet
MySQL	proprietär und GPL	leichte Installation sehr gute Dokumentation geringer RAM-Verbrauch Clustering	begrenzter Featureumfang wird nicht von allen kommerziellen Programmen unterstützt
Oracle	proprietär	sehr viele Features Clustering sehr gute Dokumentation sehr sicher [kein Datenverlust auch bei einem Servercrash während einer Transaktion] wird von jeder kommerziellen Software unterstützt	hohe Lizenzkosten sehr hoher RAM-Verbrauch