

Relationale Datenbanken und Datenbank- Managementsysteme

- Josef Putz (2013) -

Datenbanken sind aus unserer schönen neuen Online-Welt nicht mehr wegzudenken. Sämtliche Online-Shops, jedes soziale Netzwerk, jede Suchmaschine, jeder Blog und auch Enzyklopädien wie Wikipedia bestehen aus nichts anderem, als einer Datenbank.

Datenbanken sind geordnete Sammlungen von gleichartigen Daten, zumeist in Tabellenform. Gleichartig heißt, dass die Daten innerhalb einer Tabelle ähnlicher Natur sind, z. B. eine Adresskartei, der Bestand einer Bibliothek, die Verbindungsdaten von Mobilfunkkunden etc.

„Relational“ bedeutet, dass zwischen mehreren Tabellen Beziehungen (Relationen) hergestellt werden können, um wirkliche Vorgänge edv-mäßig abzubilden. Beispiel: Eine Tabelle enthält die Titel der Bücher einer Bibliothek, eine weitere Tabelle enthält die Namen von Benutzerinnen dieser Bibliothek. Über eine dritte Tabelle kann dargestellt werden, wann welche Benutzerin welches Buch ausgeliehen hat.

Prinzipiell können solche Datenbanken auch mit einem Tabellenkalkulationsprogramm erstellt werden, allerdings ist es einigermaßen schwierig, die Relationen zwischen den Tabellen herzustellen. Aus diesem Grund benutzt man für den Umgang mit einer Datenbank ein sogenanntes **Datenbankmanagementsystem** (DBMS).

Erfinder der Relationalen Datenbanken ist der amerikanische Mathematiker **Edgar F. Codd** (1923-2003).

2. Mai 2003

In der vergangenen Woche starb der Computerpionier Edgar F. Codd kurz vor dem Erreichen des 80. Lebensjahres. Der bei IBM angestellte Codd legte Anfang der siebziger Jahre die Grundlage für den Betrieb relationaler Datenbanken.

Edgar F. Codd war das jüngste von sieben Kindern eines Schuhmachers aus Dorset. Ein Stipendium gestattete es ihm, an der Universität Oxford Mathematik und Chemie zu studieren. Die eigentliche Karriere von Codd begann nach dem Zweiten Weltkrieg. Codd siedelte schon 1949 in die USA über, wo er als Mathematiker bei IBM eine Anstellung fand. Mitte der sechziger Jahre wandte sich Codd in seiner Arbeit bei IBM den Datenbanken zu. Datenbanken waren damals mächtige Systeme, die mit einigem Aufwand programmiert wurden. Das Paradestück von IBM hiess SABRE und war ein Herzstück eines computerisierten Reservierungssystems, das IBM für American Airlines entwickelt hatte und 1965 in Betrieb nahm. Die Datenbank wurde in vier Jahren entwickelt und war speziell auf IBMs 7090 und seine Plattenstapel ausgelegt. Zeitweilig arbeiteten 200 Datenbankprogrammierer an dem Projekt, das als Zukunft der Computertechnik galt. Das Projekt verzögerte sich immer wieder, weil American Airlines etwa neue Datenfelder wollte. Dann musste die Datenbank mit einer halben Million Codezeilen umgeschrieben und angepasst werden. In dieser Situation veröffentlichte Edgar Codd zwei kleinere Aufsätze (1969, 1970), die den ganzen von seinem Arbeitgeber betriebenen Aufwand in jedem Punkte ad absurdum führten und die Welt der Datenbanken veränderten. Er definierte, was eine Datenbank ausmacht, was sichere Datenhaltung und -abfrage bedeutet. Vor allem aber koppelte Codd die logische Organisation einer Datenbank, den Aufbau der Datenstruktur von den physikalischen Speichermedien ab. «Seine Ideen waren sensationell und revolutionär», urteilt der Datenbankexperte Professor Klaus Dittrich vom Institut für Informatik der Universität Zürich, «er stand plötzlich quer in der Landschaft mit seinen Ideen von einem vernünftigen Datenbankdesign.» Codds Aufsätze zündeten nicht sofort. IBM verschraubte weiterhin seine Datenbanken mit der Hardware. Aber in den Universitäten und Forschungseinrichtungen begann man mit der Datenmodellierung nach Codd. An der Universität von Kalifornien entstand so das Datenbanksystem Ingres, aus dem sich später Sybase und der Microsoft SQL Server entwickelten. Bei IBM entstand das System R, aus dem sich IBM DBs, Tandems Non-Stop SQL und Oracle entwickelten. Für seine bahnbrechenden Ideen erhielt Edgar Codd etliche Auszeichnungen, von denen der Turing Award der ACM im Jahre 1981 wohl die grösste Ehrung gewesen sein dürfte. Das ganz grosse Geld kassierte er nicht. Bis zu seiner Pensionierung im Jahre 1984 stand Codd im Dienst einer IBM, die sich nur langsam mit relationalen Datenbanken anfreunden konnte (Detlef Borchers)

Verbreitet eingesetzte DBMS heißen z. B.: **MS Access, OpenOffice Base, phpmyadmin, Oracle, dBase**

Alle diese Systeme speichern die Daten in **Tabellen**, die Eingabe der Daten erfolgt über **Formulare**, um die Daten aus einer oder mehreren Tabellen auszuwerten, erstellt man eine **Abfrage** und um die Daten in übersichtlicher Form darzustellen, benutzt man **Berichte**.

Aus Sicherheitsgründen hat man niemals direkten Zugriff auf die Daten in der Datenbank sondern immer nur über das DBMS.

Folgende Begriffe sind im Zusammenhang mit Datenbanken wichtig:

1. Datenfeld

Ein Datenfeld bezeichnet die Art eines bestimmten Datums und ist normalerweise in Spalten einer Tabelle angeordnet, z. B. alle Vornamen, alle Hausnummern etc. einer Tabelle. Diese stehen in der Regel untereinander.

2. Datensatz

Ein Datensatz ist die Gesamtheit aller zu einem Vorgang gehörenden Daten innerhalb einer Tabelle und ist im Regelfall zeilenweise angeordnet, z.B. Kundennummer, Vorname, Nachname, Adresse etc. eines Kunden.

3. Primärschlüssel

Jede Tabelle benötigt einen Primärschlüssel, damit Beziehungen zu anderen Tabellen hergestellt werden können. Dieser ist in der Regel eine fortlaufende Nummer (kann vom DBMS automatisch vergeben werden). In der Praxis zeigt sich der Primärschlüssel z. B. als Kundennummer oder als Katalognummer – damit ist der betreffende Kunde, bzw. der betreffende Katalogartikel eindeutig bestimmt.

4. Datentyp

Je nachdem, welcher Art die Daten sind, die in einem Datenfeld stehen sollen, wird diesem ein Datentyp zugewiesen. Hier unterscheidet man zwischen **INT** für ganze Zahlen, **TEXT** für Texteingaben (auch Hausnummern!) und **DATE** für Datumsangaben. Dies hat mit der Speicherplatzverwaltung und damit direkt mit der Geschwindigkeit der Datenbank zu tun. Eine Suchmaschine könnte Informationen nicht in der gewohnten Schnelligkeit liefern, wenn unnütz Speicherplatz vergeudet wäre.

Jede Datenbank kann über die Datenbankabfragesprache **SQL** (Structured Query Language) bedient werden. Die grundlegenden Befehle (SELECT FROM, INSERT INTO, DELETE, UPDATE) funktionieren in allen DBMS gleich, dennoch gibt es bei spezielleren Befehlen Unterschiede, die aber in der Dokumentation des jeweiligen DBMS erläutert sind. Wir beschränken uns in der Einführung ohnehin auf die überall gängigen Befehle, mit denen bereits sinnvolles Arbeiten mit der Datenbank möglich ist.

Alle Datenbanken sind nach folgender Hierarchie strukturiert – dies muss man sich vor Augen halten, da man sonst leicht den Überblick verliert, auf welcher Ebene man sich nun eigentlich befindet.

Datenbank: Bibliothek									
Tabelle 1 Benutzer			Tabelle 2 Bücher			Tabelle 3 Ausleihvorgänge			
<u>Benutzer-ID</u>	Vorname	Nachname	<u>Bücher-ID</u>	Titel	Autor	<u>Vorgangs-ID</u>	<u>Benutzer-ID</u>	<u>Bücher-ID</u>	Datum
1	Hans	Huber	1	Momo	Michael Ende	1	2	1	12.05.12
2	Lena	Lachner	2	Nana	Emile Zola	2	2	3	12.05.12
3	Moni	Müller	3	Michel	Astrid Lindgren	3	3	3	13.07.12

Aufgabe: Stelle dar, wer wann welches Buch ausgeliehen hat.

Der erste Schritt beim Erstellen einer Datenbank ist, dass man ein Blatt Papier und einen Bleistift nimmt und sich a) die benötigten Tabellen und b) die einzelnen Datenfelder überlegt, erst dann ist es sinnvoll, den Computer starten.

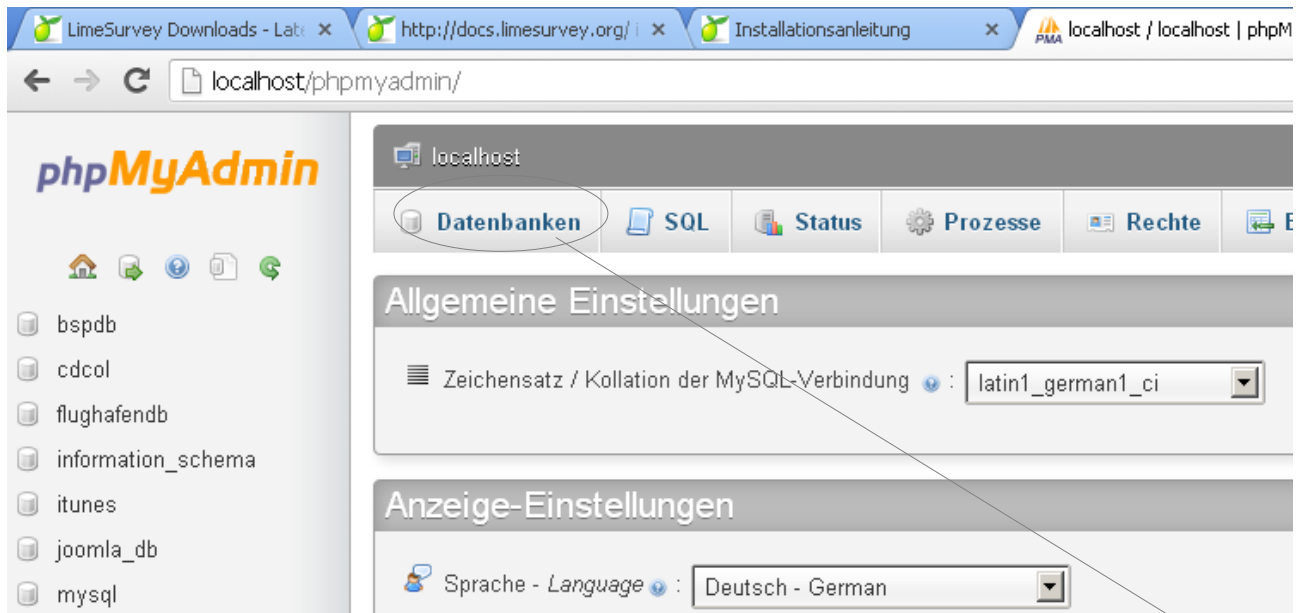
Wir wollen unsere ersten Schritte mit der Datenbank **MySQL** unternehmen. Diese bietet mehrere Vorteile:

1. Sie ist kostenlos und kann problemlos heruntergeladen und installiert werden
2. Sie ist weit verbreitet, deshalb gut dokumentiert und es gibt eine große Nutzergemeinde und entsprechend viele Erfahrungsberichte und Forumsbeiträge im Web.
3. Die meisten Webhostingangebote bieten MySQL-Datenbanken an.
4. MySQL kann sowohl im Web als auch lokal (auf dem Rechner, der vor Dir steht) genutzt werden
5. MySQL kann sehr leicht über PHP (eine Skriptsprache) in HTML-Code (Webseiten) eingebunden

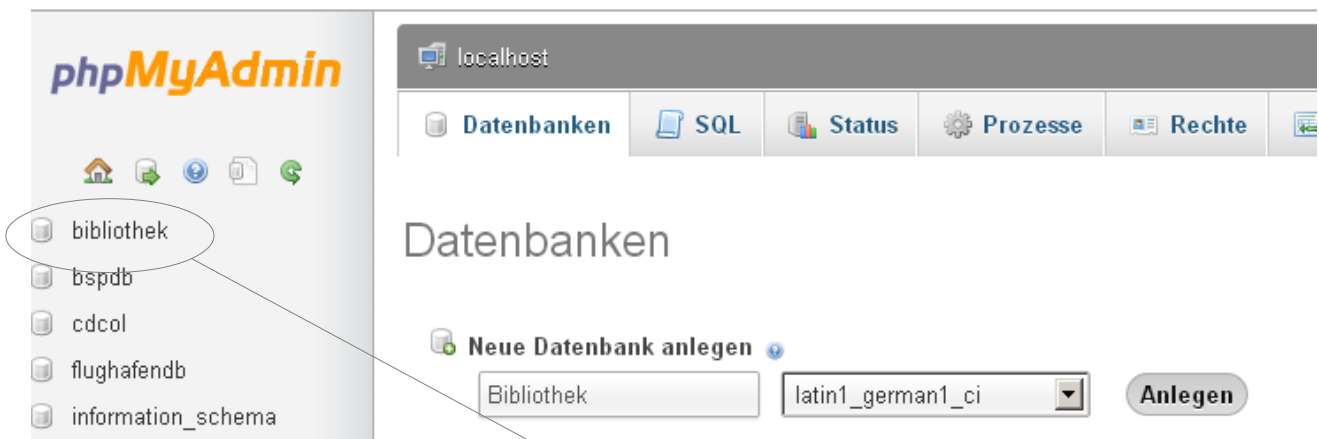
werden.

Das „My“ bezieht sich übrigens auf den Vornamen der Tochter eines der Entwickler dieser Datenbank, Michael Widenius, einem Norweger, hat also nichts mit dem englischen „my“ zu tun . . .

Datenbanken wie MS Access oder OpenOffice Base haben den Nachteil, nur lokal zu laufen und sind außerdem in der Bedienung äußerst sperrig. Nichtsdestotrotz können auch diese mit SQL-Befehlen gesteuert werden, haben aber auch eigene Bedienungskonzepte.



Dieses Fenster zeigt links die bereits angelegten Datenbanken. Mit Klick auf Datenbanken kann eine neue DATENBANK (Achtung: Nicht mit TABELLE verwechseln) angelegt werden. Innerhalb dieser Datenbank können wiederum neue Tabellen angelegt werden.



Wähle nun die neu angelegte Datenbank aus.



Nun kann eine neue Tabelle angelegt werden. Überlege Dir, wieviele Spalten diese haben soll. Es können jedoch später weitere Spalten hinzugefügt und auch entfernt werden.

Erzeuge Tabelle			
Tabellenname:			
Bücher			
Struktur			
Spalte	Bücher-ID	Titel	Autor
Typ	INT	TEXT	TEXT
Länge/Set1			
Standard2	Kein(e)	Kein(e)	Kein(e)
Kollation			
Attribute			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	---	---	---
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kommentare			

Hier siehst Du die neu angelegten Spalten „Bücher-ID“, „Titel“ und „Autor“ mit den Datentypen INT und TEXT. Für eine Tabelle „Ausleihvorgänge“ wäre noch eine Spalte Datum mit dem entsprechenden Datentyp anzulegen.

Aufgabe: Lege die Tabellen für die Benutzerinnen und die Ausleihvorgänge entsprechend dem Beispiel an.

Mit Klick auf „Einfügen“ kannst Du nun Deine Tabelle mit Werten befüllen. Später wirst Du auch Wege kennenlernen, dies online durch Besucher Deiner Webseite erledigen zu lassen. (Z. B. wenn Du einen Shop betreibst, und neue Kunden ihre Adresse hinterlassen wollen)

Nun wieder etwas trockene Datenbank-Theorie:

Um die Daten möglichst effizient zu speichern, werden diese **atomisiert**, d. h. in kleinstmögliche Einheiten zerlegt. Das bedeutet, dass z. B. der Name in Vorname und Nachname zerlegt und in jeweils einer eigenen Spalte gespeichert wird. Ebenso die Adresse. Sicher kennst Du Adressformulare, die sogar verlangen, Straße und Hausnummer getrennt einzugeben.

Ein weiterer wichtiger Punkt ist, dass **Redundanz** vermieden werden soll. Unter Redundanz versteht man das mehrfache Vorkommen der selben Daten. Z. B. wenn die Adresse einer Benutzerin in der Adresstabelle und zusätzlich in der Vorgangstabelle gespeichert wäre. Probleme entstehen, wenn solche Daten geändert werden müssen, da nicht gewährleistet ist, dass der Wert überall, wo er vorkommt, geändert wird.

Übrigens ist Redundanz an sich nichts Schlechtes: Jede Datensicherung ist redundant und wir würden uns schön bedanken, wenn unser Mobilfunkbetreiber keine Sicherung der Daten hätte und uns Monat für Monat sonst einen Betrag berechnen würde . . .

SQL-Befehle

Die „Structured Query Language (strukturierte Abfragesprache)“ enthält eine Handvoll Befehle, die ausreichen, um die nötigsten Tätigkeiten mit einer Datenbank zu erledigen.

Im Wesentlichen muss man sich folgendes merken:

- 1.) Insert Into
- 2.) Select
- 3.) Update
- 4.) Alter
- 5.) Delete / Truncate
- 6.) Drop

Die Erläuterung mit Beispielen zu den SQL-Befehlen wird in Kürze hier zu finden sein.