

3 Was Datenbanken leisten und was sie nicht leisten

Erster Stolperstein bei Projekten, die auf der Basis von Datenbanken aufsetzen, ist der Begriff der Datenbank selbst. Aufgrund unterschiedlicher Vorkenntnisse und Ansichten kann es für einen Projektleiter sehr schwer werden, ein solches Projekt erfolgreich zu organisieren und abzuschließen.

Die unterschiedlichen Ansichten, was bereits als Datenbank bezeichnet werden darf und was nicht, führt in vielen Fällen gleich zu Beginn von Online-Projekten zu kontroversen Lösungsansätzen. Ebenso undurchsichtig ist häufig eine fundierte Begründung, warum etwas besser ohne versus mit einer Datenbank realisiert werden sollte.

Sie finden in diesem Kapitel die grundsätzlichen Definitionen eines Datenbanksystems und Erklärungen über üblicherweise eingesetzte Techniken und Anforderungen.

3.1 Datenbanken im Intranet und Internet

Wenn während der Realisierung eines Online-Projekts die Frage nach der Verwaltung vorhandener Datenbestände auf den Tisch kommt, dann ist man meistens an Lösungen zu üblichen Fragen der Datenorganisation und des Handlings interessiert.

- ▶ Wie kann die **inhaltliche Aktualisierung** der Website vereinfacht und beschleunigt werden?
- ▶ Wie kann ein Besucher **individuell auf ihn zugeschnittene Inhalte** angeboten bekommen?
- ▶ Wie wird ein **Newslettersystem** realisiert?
- ▶ Wie kann eine Website schnell **in andere Sprachen übersetzt** werden?
- ▶ Wie kann ein **Suchsystem** über die eigene Web-Präsenz organisiert werden?

- ▶ Welche Techniken stehen zur Realisierung von **Online-Foren** etc. zur Verfügung?
- ▶ Wie können notwendige **Aktualisierungen automatisiert** werden, damit nicht immer ein Mitarbeiter an diese Arbeiten denken muss?
- ▶ Wie können Mitarbeiter im Intranet **persönliche Informationen** wie beispielsweise Termine und Kontakte etc. speichern?
- ▶ etc.

Es wird schnell offensichtlich, dass zur Lösung dieser Anforderungen normales HTML nicht mehr ausreicht und in den meisten Fällen clientseitige Skriptsprachen wie **JavaScript** oder **VBScript** nicht zufrieden stellend arbeiten. Zwar wären Suchfunktionen auch mit einer Sprache zu realisieren, die vom Browser direkt interpretiert wird, aber das große Problem liegt in der *nicht vorhandenen Kontrolle*, ob die Funktionen auch bei jedem Benutzer aktiviert sind und ausgeführt werden können.

Häufig deaktivieren Administratoren in den Browsern der Mitarbeiter die benötigten Funktionen, um bessere Kontrolle zu gewährleisten oder um eventuelle Sicherheitsrisiken von vorneherein auszuschließen. Der Mitarbeiter bemerkt das sehr oft nicht und wundert sich permanent, warum sehr viele Webseiten nicht richtig funktionieren.

Schalten Sie beispielsweise beim Internet Explorer das **Active-Scripting** aus, werden alle Funktionen, die mit JavaScript realisiert wurden, nicht mehr nutzbar sein – für eine professionelle Web-Präsenz eigentlich undenkbar. Hinzu kommt außerdem, dass mit clientseitigen Sprachen keine Dokumente abgespeichert werden können, also zum Beispiel Aktualisierungen von Webseiten überhaupt nicht zu realisieren sind.

Für alle oben beschriebenen Anforderungen kommt im Grunde nur der Einsatz so genannter **serverseitiger Skriptsprachen** wie Perl, PHP oder vergleichbare Techniken in Betracht. Diese Sprachen ermöglichen unter anderem das Dateihandling, die Reaktion auf Benutzereingaben oder auch dynamische Seitengenerierung und viele Dinge mehr.

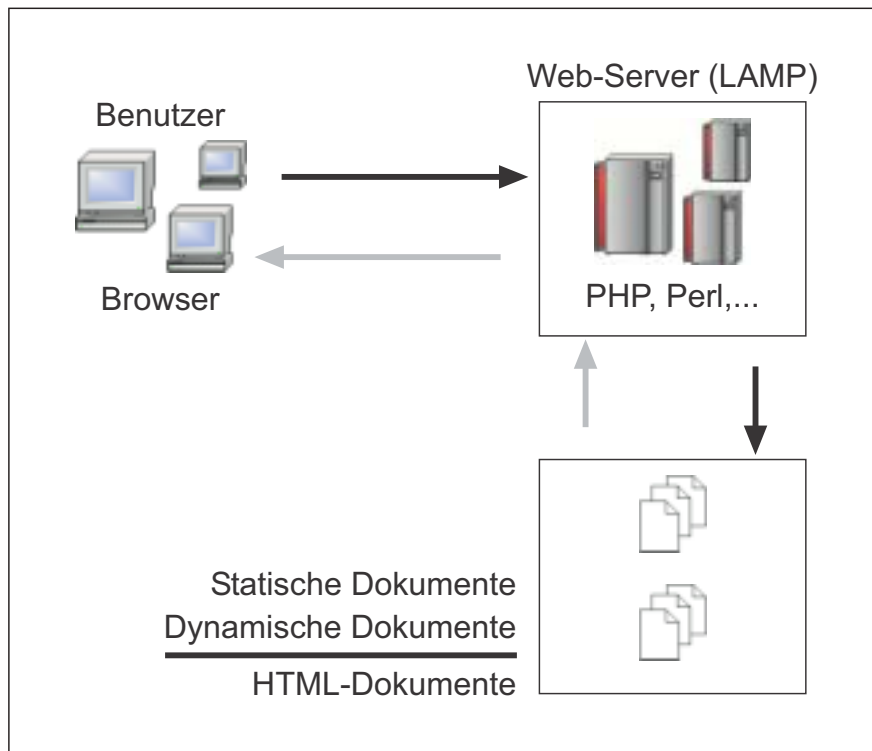


Abbildung 3.1: Statische und dynamische Seiten werden als HTML-Dokument zum Browser gesendet.

Wenn ein Benutzer in seinem Browser die Adresse einer Website eingibt, fordert er über einen Webserver Informationen an, die mittels des **HTTP-Protokolls** übertragen werden. Anhand der Dateierweiterung erkennt der Webserver, dass es sich unter Umständen nicht um eine statische Datei, sondern um ein Skript handelt, das zunächst ausgeführt werden muss. Dieses Skript kann den HTML-Inhalt eines Dokuments dynamisch erzeugen und der Webserver schickt diesen Inhalt zum anfragenden Browser zurück. Ob es sich bei der anschließend angezeigten Seite um eine statische oder dynamische Seite handelt, kann der Benutzer oft nicht erkennen.

Das Beispiel aus Abbildung 3.2 zeigt deutlich, wie in Abhängigkeit der vorhandenen Nachrichten eine Website dynamisch erzeugt wird. Prinzipiell schaut ein PHP-Skript in einer »Datenablage« nach, wie viele Nachrichten vorhanden sind und angezeigt werden müssen, und generiert anschließend die HTML-Ausgabe, die im Browser zu sehen sein wird.

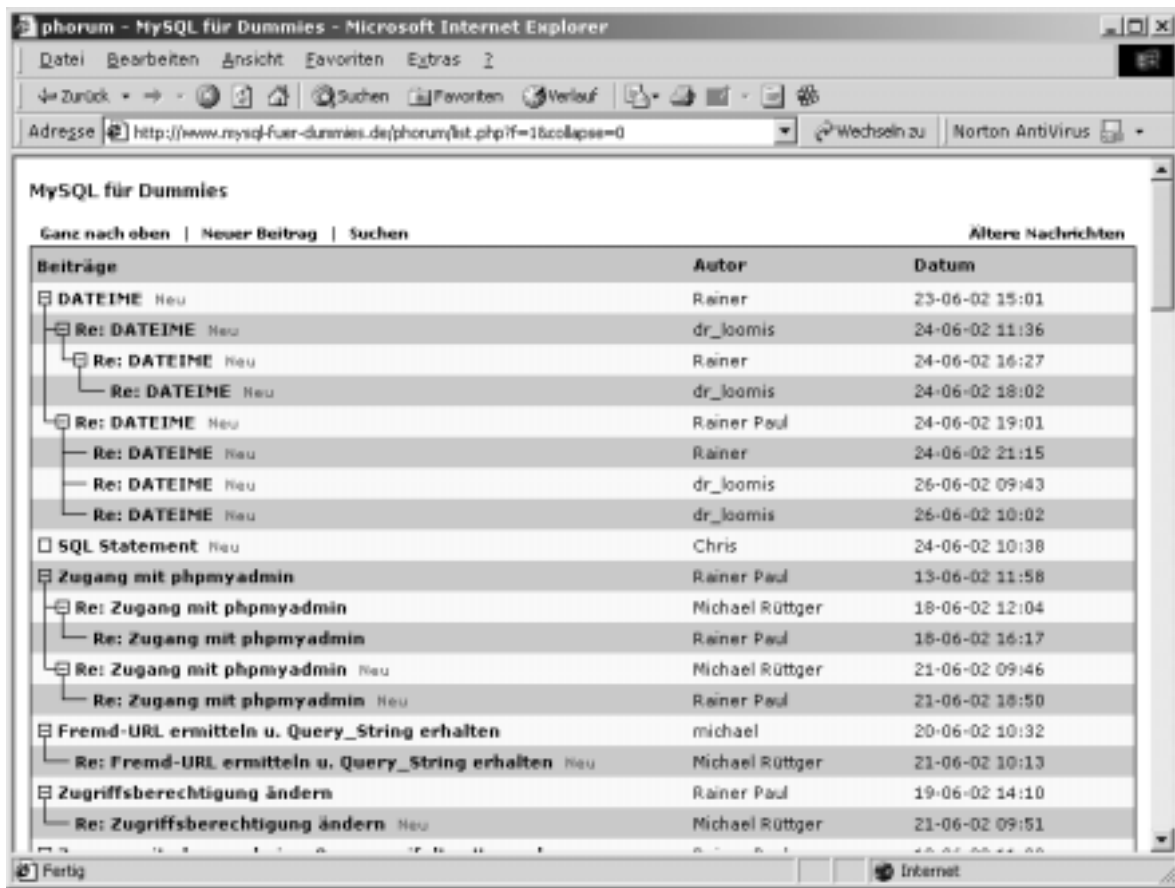


Abbildung 3.2: Dynamisch erzeugte Seite eines Forums

Jedem Beteiligten an einem Online-Projekt muss bei diesen programmtechnischen Umsetzungen ein grundsätzliches Prinzip völlig klar sein – die gespeicherten Daten werden in der Regel niemals so abgelegt, wie sie für den Menschen im Nachhinein Informationen liefern, sondern möglichst komprimiert und **ohne Redundanzen**.

Jetzt stellt sich noch die Frage, ob eine solche Umsetzung besser mittels einfacher Text-Dateien oder mit einem Datenbank-Management-System zu realisieren ist. Es ist ebenso falsch, für jede Lösung den Einsatz eines Datenbanksystems zu empfehlen, wie grundsätzlich auf eine solche zu verzichten. Damit überhaupt eine fundierte Entscheidung getroffen werden kann, benötigt man ein grundlegendes Verständnis über die folgenden Begriffe:

- Bilden Daten eine logisch zusammengehörende Information, wie zum Beispiel die Adresse einer Person, so werden diese Datenbestandteile in Form eines **Datensatzes** verwaltet und abgespeichert.

- ▶ Werden solche Datensätze auf einem Speichermedium strukturiert abgelegt, also gespeichert, geschieht das in Form von **Dateien**.
- ▶ Sind eine bestimmte Anzahl von Dateien ebenfalls wieder logisch einander zugeordnet, so wird diese Sammlung ganz allgemein als **Datenbank (DB)** bezeichnet.
- ▶ Existiert ein System, das eine oder mehrere Datenbanken verwaltet und organisiert, wird es als **Datenbanksystem (DBS)** bezeichnet.
- ▶ Im Gegensatz zu einfachen Textdateien ermöglicht ein Datenbanksystem dem späteren Benutzer den Zugriff auf die gespeicherten Daten, ohne dass er wissen muss, wie diese Daten intern strukturiert und in welchen Dateien diese verwaltet werden.
- ▶ Ein Datenbanksystem stellt eine Vielzahl von **Methoden** zur Verfügung, mit denen die einzelnen Datensätze manipuliert beziehungsweise bearbeitet werden können. Unter einer Manipulation wird zum einen das **Verändern, Erweitern** oder das **Löschen** von Datenbeständen verstanden und zum anderen das strukturierte Aufbereiten notwendiger Aussagen. So kann beispielsweise das Alter einer Person verschiedene Informationen liefern – je nachdem, was gerade benötigt wird, zeigt es uns die Tage auf, die diese Person bereits auf der Welt ist, oder es verrät uns, zu welchem Datum dieser Mensch das Licht der Welt erblickte – im Datensatz gespeichert wird bspw. nur der Zahlenwert.
- ▶ Ein Datenbanksystem sorgt durch bestimmte **Vorschriften** und **Regeln** dafür, dass Datensätze und Datenbanken nicht unfreiwillig zerstört oder gelöscht werden können, es verwaltet die Rechte verschiedener Benutzer und achtet im Idealfall sogar darauf, dass bereits einmal gespeicherte **Datensätze nicht doppelt abgelegt** werden.
- ▶ Das Datenbanksystem stellt meist mehrere **Schnittstellen** bereit, durch die andere Prozesse, Systeme oder Applikationen strukturiert und optimiert auf die gespeicherten Daten zugreifen können. Diese Zugriffe organisiert das **Datenbank-Management-System (DBMS)**.

Datenbank-System

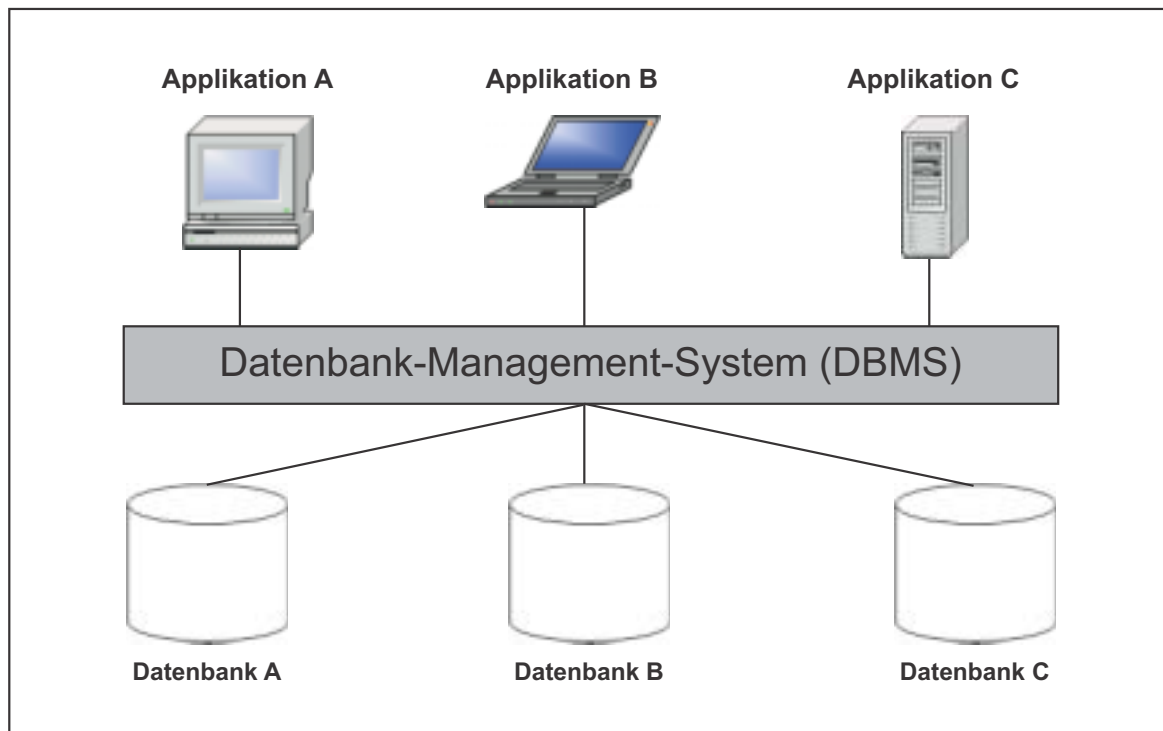


Abbildung 3.3: Zusammenfassung der Komponenten DBS, DBMS und DB

Ein Datenbanksystem besteht also aus einer Verwaltungssoftware, dem Datenbank-Management-System, das den Zugriff auf einzelne Datenbanken und damit auf die einzelnen Dateien organisiert. Mit der Abbildung 3.3 erhalten Sie zwar eine Übersicht über den Aufbau und die Funktionsweise eines Datenbanksystems, jedoch finden Sie keine Hinweise über die Art der Organisation. Diese wird später noch einmal in **hierarchische, netzwerkorientierte und relationale Datenbanken** eingeteilt.

Im Gegensatz zum Direktzugriff auf eine Textdatei sorgt das DBMS dafür, dass durchaus mehrere Personen *gleichzeitig* auf Datenbestände (Dateien) zugreifen können, ohne dass Konflikte entstehen. So können verschiedene Benutzer Daten aktualisieren, während andere Benutzergruppen nur Daten betrachten möchten. Einer möchte suchen, der andere sortieren etc. – das DBMS sorgt für einen reibungslosen Ablauf all dieser unterschiedlichen Anfragen.

Im Gegensatz zu einer Lösung, mit der direkt auf Dateien zugegriffen wird, existiert ohne ein DBS keine Organisation, die vor Konflikten schützt. Diese

Überprüfungen müssten selbst entwickelt und programmiert werden, was allerdings widersinnig wäre, da es dafür ja bereits Datenbanken gibt. Der Zugriff über ein DBMS kostet natürlich mehr Ressourcen und so kann es durchaus sein, dass sich der Verzicht auf ein solches System ebenfalls auszahlt.

Lassen Sie mich das an einem kleinen Beispiel verdeutlichen. Ein Unternehmen möchte selbst, mittels eines einfachen HTML-Formulars, die Seite »Aktuelles« ihrer Web-Präsenz pflegen. Ein PHP-Skript nimmt die Daten des Formulars auf, löscht die vorhandene Seite `aktuelles.html` und erzeugt mit den gerade eingegebenen Daten eine neue Seite mit gleichem Namen. Dieser Vorgang geht sehr schnell und erfordert ohne den Einsatz einer Datenbank auch wesentlich weniger Ressourcen. Allerdings besteht bei diesem Beispiel die Anforderung lediglich darin, auf einfache und strukturierte Weise eine HTML-Seite generieren zu können, so dass auch Mitarbeiter ohne Programmierkenntnisse Änderungen in einer Online-Präsenz vornehmen können. Kommt das Unternehmen aber auf die Idee, dass dem Benutzer der Website eine Art Archiv zur Verfügung stehen soll, in dem er auch noch nach Begriffen suchen kann, ist der Verzicht auf eine Datenbank wahrscheinlich eher hinderlich.

Datenbanken im Intranet oder Internet sind also ein Mittel, anfallende und wechselhafte Datenbestände ordentlich zu verwalten und zu bearbeiten. Um Speicherplatz zu sparen, werden üblicherweise nur Daten gespeichert, die zur Erzeugung späterer Information absolut notwendig sind; es wird also Datenredundanz vermieden. Um aus diesen Einzeldaten auf Anforderung wieder brauchbare Information für den Menschen zu erzeugen, stellt ein Datenbank-Management-System über strukturierte Abfragesprachen Methoden zur Verfügung, diese Informationen bei Bedarf zu generieren. Sobald auf ein DBMS verzichtet wird, müssen die Entwickler selbst für eine entsprechende Verwaltung sorgen, was meist mit mehr Aufwand verbunden und auch fehleranfälliger ist.

Sobald eine gewisse Menge an gespeichertem Datenmaterial überschritten wird, ist eine Organisation über normale Dateioperationen nicht mehr realisierbar. So verwenden Online-Übersetzer beispielsweise *über sechs Millionen Wörter* und Redewendungen, die in Sekundenschnelle von mehreren tausend Benutzern gleichzeitig abrufbar sein müssen. Der Buchhandel verwaltet Millionen von Büchern, die recherchiert und durchsucht werden können, und so

weiter. Ohne Datenbank-Einsatz sind solche Angebote über das Intranet und Internet nicht durchführbar.

3.2 Aufwand versus Benutzersicht

Sehr häufig wird von Lösungen mittels Online-Datenbanken Abstand genommen, da die zunächst erkennbaren Entwicklungskosten weit über die Beträge hinausgehen, die für die Entwicklungen statischer Websites benötigt werden. Stellt sich jedoch im Nachhinein heraus, dass die Seiten sehr häufig geändert und aktualisiert werden müssen, kann sich die zu Beginn hohe Investitionssumme schnell auszahlen. Agenturen oder Entwickler haben es jedoch nicht immer einfach, diesen Sachverhalt ihren Kunden deutlich zu machen.

3



Abbildung 3.4: Dynamisch generierte Webseite mit Hilfe von Online-Datenbanken

Nehmen Sie beispielsweise die Website einer bekannten Tageszeitung in meiner Region. Unter <http://www.rz-online.de> erhalten Sie rund um die Uhr immer aktuelle Informationen zu allen Dingen, die Menschen so bewegen.

Stellen Sie sich vor, Sie müssten diesem Verlag einen oder mehrere Entwürfe zur Realisierung einer Web-Präsenz vorlegen. Wenn Sie als Entwurf eine Website wie in Abbildung 3.4 abgeben, wird für den Auftraggeber überhaupt nicht ersichtlich, dass im Hintergrund eine Datenbank und serverseitige Skripten arbeiten. Er sieht vielmehr nur seine Anforderungen nach bestimmten Menüpunkten und Inhalten bestätigt, kann aber nicht erkennen, warum eine Datenbank notwendig ist.

Findet sich nun noch ein Konkurrent, der einen nahezu gleichen Entwurf abgeliefert, der ohne eine Datenbank arbeitet, sieht der Auftraggeber zunächst lediglich die zu erwartenden Kosten und wird ad hoc die statische Lösung bevorzugen.

Hier hilft nur ein gutes Pflichtenheft und eine verständliche Präsentation, warum eine Datenbank-Lösung in diesem speziellen Fall vorzuziehen ist. Gerade bei diesem Beispiel werden die anfänglichen Kosten sehr hoch sein, jedoch muss hier primär ein System angeboten werden, mit dem diese gewaltigen Mengen an täglich zu aktualisierenden Informationen sowohl personell als auch organisatorisch verwaltet werden können. Eine statische Lösung verschlingt natürlich nur einen Bruchteil der Entwicklungskosten, würde aber erstens nicht mehr zu handhaben und – wenn alle Aktualisierungen einzeln abgerechnet werden – auch nicht mehr zu finanzieren sein. Diesen Sachverhalt muss der Auftraggeber verstehen und dazu helfen in ganz entscheidendem Maße das Pflichtenheft und die Projektdokumentation.

Wenn bei Online-Projekten nicht blindlings für den Einsatz von Datenbanken entschieden wird, haben die folgenden Aussagen große Bedeutung:

- ▶ Datenbankgestützte Online-Lösungen bringen **höhere Entwicklungskosten** mit sich als die Entwicklung statischer Web-Präsenzen ohne genannte Techniken.
- ▶ Sobald erkennbar ist, dass die **Folgekosten**, durch beispielsweise Pflege und permanente Änderungen, höher als die Entwicklungskosten sein werden, muss an den Einsatz einer Webdatenbank gedacht werden.
- ▶ Letztendlich muss durch eine Gegenüberstellung der Kosten der **Nutzen** einer Datenbank deutlich gemacht werden.
- ▶ Als Faustregel gilt: Datenbank-Lösungen haben hohe Erstellungskosten, senken aber idealerweise anfallende Folgekosten, wenn sie durchdacht

und passend eingesetzt werden. Damit haben sie ihre Berechtigung bei allen Online-Projekten, egal in welcher Größenordnung.

3.3 Technische Voraussetzungen für Webdatenbanken

3

Um eine allgemein übliche Online-Präsenz zu verwirklichen, benötigen Sie grundsätzlich eine Internetadresse, die so genannte **Domain**, als auch einen Rechner mit permanenter Verbindung zum Internet, der Ihnen den benötigten Plattenplatz zur Verfügung stellt. Zusammenfassend bezeichnet man den Rechner als **Webserver**, den Plattenplatz als **Web-Space** und die gesamte Dienstleistung als **Web-Hosting**.

In den meisten Fällen wird für das Web-Hosting auf die Angebote externer Dienstleister, den **Internet-Service-Providern (ISPs)**, zurückgegriffen, da der Betrieb eines eigenen Webservers nur unter ganz bestimmten Umständen sinnvoll und finanzierbar ist. Ganz gleich, welche Variante bei der Konzeption empfohlen oder bevorzugt wird, wichtig ist, dass die Techniken vorhanden sind, die man zur Realisierung einer datenbankgestützten Online-Präsenz benötigt.

Je genauer das Lastenheft bereits im Vorfeld die Anforderungen und Wünsche beschreibt, desto einfacher ist die Auswahl der benötigten Funktionalität und damit die Anforderung an das Web-Hosting. Finden Sie in der Auftragsanforderung bereits erste Hinweise auf einen möglichen Datenbankeinsatz, ist die notwendige Technik schon nahezu vorgegeben.

- ▶ Sie benötigen definitiv eine **Webserver-Konfiguration**, die auch den Einsatz serverseitiger Skripten wie Perl, PHP oder ASP, JSP etc. zulässt. Diese Programmiersprachen werden unter anderem für die Anbindung an Datenbanksysteme eingesetzt und erlauben gleichzeitig, mit Hilfe der ermittelten Ergebnisse, die Erzeugung dynamischer Webseiten.
- ▶ Sie benötigen ein **Datenbanksystem**, das mit den zuvor beschriebenen Skriptsprachen zusammenarbeiten kann. In den meisten Fällen müssen Datenbanksysteme, die für den Einsatz im Internet gedacht sind, sehr hohe Verarbeitungszeiten und einfache Integration in die Web-Architektur mitbringen.
- ▶ Sie benötigen einfache und sichere Möglichkeiten, dieses Komplettsystem zu administrieren, was eine gewisse Anforderung an den Provider dar-

stellt, der diverse Techniken für die Wartung zur Verfügung stellen muss. Ebenso kommt nur der Einsatz von Software in Frage, die für den Betrieb in einem Netzwerk geschrieben wurde, somit mehrbenutzerfähig ist und eine **Fernwartung** erlaubt.

Welche Hersteller, Produkte oder Lizenzformen im Nachhinein zur Realisierung favorisiert werden, hängt zu einem Großteil von der Frage ab, ob die geplante Online-Präsenz auch mit einem bereits vorhandenen Unternehmensnetzwerk verbunden werden soll – also eine Intranet-/Internetlösung benötigt wird. Die Schwierigkeit liegt dabei nicht so sehr in der Auswahl geforderter Funktionen, sondern eher in der Kompatibilität der vorhandenen Architektur mit dem neuen System. Sollen bereits existierende Datenbankserver als Datenquelle für die Generierung von Webseiten dienen, so sind die notwendigen Sicherheitsvorkehrungen enorm und nicht zu unterschätzen. Häufig wendet man sich aus diesem Grund an die gleichen Hersteller oder Dienstleister, die auch die bereits vorhandene Infrastruktur gestellt haben. Die renommiertesten Unternehmen dieser Branche, wie IBM, SUN, Oracle oder auch Microsoft, bieten allenthalben komplette E-Commerce-Lösungen an, die nicht speziell neu entwickelt werden müssen, da sie auf vorhandene Techniken aufsetzen.

Zeigt Ihr Projekt in diese Richtung, ist dringend zu empfehlen, Fachkräfte und Spezialisten der jeweiligen Produkte zu konsultieren. Ob diese direkt in dem Projekt mitarbeiten oder nur Informationscharakter haben, hängt vom Umfang des Projekts ab. Solche Lösungen an dieser Stelle näher zu betrachten, kann nicht sehr sinnvoll sein, da sie stark herstellerspezifisch sind und sich schneller ändern, als ich dieses Buch schreiben kann.

Von allen dynamisch arbeitenden Online-Präsenzen sind jedoch die meisten mit allgemeinen und herstellerunabhängigen Produkten mehr als zufriedenstellend zu realisieren. Die allgemein weit verbreitete Ansicht, dass Skriptsprachen wie PHP oder Datenbank-Management-Systeme wie MySQL nicht dazu taugen würden, kommerzielle, hochkomplexe Anforderungen zu bewerkstelligen, zeigt nur die unreflektierte Weitergabe aufgeschnappter Informationen.

Provider-Checkliste			
Kriterium	Ja	Nein	Bemerkungen
Basiskriterien für geschäftlich genutzte Hosting-Angebote:			
Wird die Reservierung benötigter Domains vom ISP übernommen?	<input type="checkbox"/>	<input type="checkbox"/>	
Können auch internationale Domains wie .com, .net, .co.uk, .at, .info, .biz etc. reserviert werden?	<input type="checkbox"/>	<input type="checkbox"/>	
Steht genügend Web-Space zur Verfügung?	<input type="checkbox"/>	<input type="checkbox"/>	
Besteht genügend Freiraum bis zur Überschreitung des Datentransfervolumens?	<input type="checkbox"/>	<input type="checkbox"/>	
Wird eine Datentransfer-Statistik angeboten?	<input type="checkbox"/>	<input type="checkbox"/>	
Existiert eine Logfile-Auswertung?	<input type="checkbox"/>	<input type="checkbox"/>	
Stehen Standard-CGIs für bspw. die Formularauswertung zur Verfügung?	<input type="checkbox"/>	<input type="checkbox"/>	
Können individuelle Wünsche vom ISP umgesetzt werden oder sind Verträge und Tarife schnell wechselbar?	<input type="checkbox"/>	<input type="checkbox"/>	
Weiterführende Kriterien für die Nutzung von datenbankgestützten Websites:			
Können serverseitige Skriptsprachen wie PHP, Perl, Python oder SSI verwendet werden?	<input type="checkbox"/>	<input type="checkbox"/>	
Steht ein oder optional auch mehrere Datenbank-Management-Systeme wie bspw. MySQL zur Verfügung?	<input type="checkbox"/>	<input type="checkbox"/>	
Steht Ihnen eine eigene IP-Adresse zur Verfügung?	<input type="checkbox"/>	<input type="checkbox"/>	
Wird eine Firewall zur besseren Sicherung der Daten eingesetzt?	<input type="checkbox"/>	<input type="checkbox"/>	
Werden regelmäßige Backups der Datenbanken und des Web-Space durchgeführt?	<input type="checkbox"/>	<input type="checkbox"/>	
Besteht die Möglichkeit für einen SSH-Zugang zur besseren Pflege der Datenbanken?	<input type="checkbox"/>	<input type="checkbox"/>	
Können Cronjobs zur Automatisierung verschiedener Routinearbeiten eingerichtet werden?	<input type="checkbox"/>	<input type="checkbox"/>	

Provider-Checkliste			
Kriterium	Ja	Nein	Bemerkungen
Kann der Webserver nach individuellen Wünschen installiert werden (bspw. individuelle Wahl von PHP-Modulen)?	<input type="checkbox"/>	<input type="checkbox"/>	
Besteht ausreichende Support-Möglichkeit durch den ISP?	<input type="checkbox"/>	<input type="checkbox"/>	

Tabelle 3.1: Checkliste für die Wahl des richtigen Providers

In der Tabelle 3.1 finden Sie eine Sammlung von allgemeinen Kriterien, die zur Auswahl der richtigen Provider herangezogen werden können. Nichts spricht zunächst dagegen, die geplanten Anforderungen auf verschiedene Anbieter zu verteilen, jedoch sollte im Idealfall jedes Kriterium mit JA beantwortet werden können, damit Sie einen Partner haben, der Ihren geschäftlichen Ansprüchen gerecht werden wird.

Prinzipiell ist es nicht so wichtig, ob bereits bestehende Tarife die vorgeschlagenen Funktionen unterstützen, sondern eher, ob der oder die Provider schnell und unkompliziert auf individuelle Wünsche reagieren können. Ein ebenso wichtiges Merkmal ist die Performance des Gesamtpaketes. Bitten Sie Ihren potenziellen Partner um die **Nennung einiger Referenzkunden**, die bereits Datenbanklösungen auf der Basis der angebotenen Techniken realisiert haben, und überprüfen Sie die Lade- und Reaktionszeiten der genannten Online-Präsenzen. Diese Überprüfungen sollten Sie zu verschiedenen Zeiten des Tages durchführen, um so auch mögliche Stresszeiten mit sehr vielen Benutzern beurteilen zu können.

3.4 Datenbanksysteme und das Internet

Hat man sich für den Einsatz einer Online-Datenbank entschieden, beeinflusst diese Entscheidung in ganz gravierendem Maße den weiteren Projektlauf. Inhalte müssen in anderer Reihenfolge geplant werden und die Auswahl der zur Verfügung stehenden Techniken wird komplexer. So stellt sich logischerweise auch die Frage nach der einzusetzenden Datenbank, denn nicht jedes Datenbanksystem eignet sich gleich gut für den Einsatz im Internet.

Datenbank	Hersteller/Vertrieb	URL
Oracle/DB2	Oracle Corporation Deutschland	http://www.oracle.de/
DB2 und Informix	IBM Deutschland	http://www.ibm.de/
SQL-Server	Microsoft Deutschland	http://www.microsoft.de/
Sybase	Sybase Deutschland GmbH	http://www.sybase.de/
Adabas D	Software AG	http://www.adabas.de/
MySQL	MySQL AB	http://www.mysql.com/
PostgreSQL	PostgreSQL Inc.	http://www.pgsql.com/

Tabelle 3.2: Eine Auswahl gängiger Datenbanksysteme für das Intranet und Internet (grau = Open-Source-Produkte)

Damit für diese Frage eine korrekte Entscheidung getroffen werden kann, muss zunächst geklärt werden, welche Infrastruktur bereits genutzt wird. Befindet sich in einem Unternehmen beispielsweise ein SAP/R3-System im Einsatz, lohnt es sich zumindest, über eine Nutzung der schon vorhandenen Ressourcen nachzudenken. Auch ist die Frage nach den üblicherweise verwendeten Betriebssystemen an dieser Stelle genauso bedeutsam wie die Betrachtung, ob ein eigener Webserver eingesetzt oder auf die Leistungen eines externen ISPs zurückgegriffen wird. Ist Letzteres der Fall, müssen Sie einen Provider finden, der die von Ihnen favorisierte Technik überhaupt unterstützt. Die letzte entscheidende Frage betrifft die laufenden Kosten einer solchen Präsenz. Sie können davon ausgehen, dass bei der Nutzung eines SQL-Servers von Microsoft pro Monat wesentlich höhere Nutzungskosten anfallen als zum Beispiel beim Einsatz einer MySQL-Datenbank.

Dieser Sachverhalt darf aber niemals als Indiz für ein besseres Produkt bzw. eine bessere Datenbank angesehen werden. Selbstverständlich haben Produkte von Oracle, von Microsoft, von IBM ihre Anwendungsgebiete, aber ebenso finden Sie professionelle und perfekte Lösungen mit anderen Produkten. Niemals sollte als Entscheidungskriterium nur der Name oder eine bereits bestehende Verbindung zu einem Hersteller alleine entscheidend sein, sondern auch die Vor- und Nachteile der einzelnen Datenbanksysteme selbst bedacht werden – und Vor- und Nachteile finden Sie bei allen Systemen.

Eine der erfolgreichsten und am meisten eingesetzten Online-Datenbanken ist MySQL. Nahezu jeder Provider bietet diese Technik an, so dass es auch nicht schwer sein dürfte, den passenden Partner zu finden. Oracle hingegen ist erst vor kurzer Zeit nominiert worden, als einziges Datenbanksystem die meisten Sicherheitsrichtlinien umgesetzt zu haben, und der SQL-Server wird bei allen Providern eingesetzt, die ihre Webserver mit Microsoft-Technologie betreiben. Sie sehen, eine Entscheidung ist nicht so einfach, sondern benötigt eine genaue Sondierung der Anforderungen an das Projekt und ist zusätzlich abhängig von einer Vielzahl an Umfeldkriterien.

Ohne eine spezielle Datenbank zu bevorzugen, gelten für alle Webdatenbanken folgende Anforderungskriterien:

- ▶ Alle Anfragen und Funktionen, die aus dem Netz an die Datenbank gestellt werden, müssen schnell und ohne Zögern abgearbeitet werden. Die **Verarbeitungsgeschwindigkeit** ist von großer Bedeutung.
- ▶ Es müssen genügend **Schnittstellen** für das gewählte DBMS bestehen, damit Programmierer keine Schwierigkeiten haben, mit den zur Verfügung stehenden Skriptsprachen eine Verbindung zur Datenbank herzustellen und Operationen mit ihr ausführen zu können.
- ▶ Ein Online-System muss schnell und einfach **über das Internet administrierbar** sein.
- ▶ Online-Datenbanken müssen »unproblematisch« gegen unberechtigten Zugriff abzusichern sein. Zumindest muss überhaupt ein entsprechendes **Rechtesystem** existieren, auf das zurückgegriffen werden kann.
- ▶ Da durch das Internet von einer sehr hohen Anzahl Benutzer ausgegangen werden kann, die zusätzlich auch alle gleichzeitig auf eine Datenbank zugreifen könnten, wird ein System benötigt, das die entsprechenden **Spitzenlasten** unproblematisch abfangen kann. Zu diesem Punkt muss aber fairerweise ergänzt werden, dass Datenbanksysteme in ganz entscheidendem Maße von der **Hardware** abhängig sind, auf der sie betrieben werden.

Häufig diskutierte Fragen wie mögliche **Kapazitäten** oder speziell technische Funktionen sind an dieser Stelle meist noch zweitrangig, da alle genannten Systeme bereits eine sehr große Menge an Datenbeständen ohne Probleme verwalten können – und dazu zählen auch die beiden letzten DBMS wie MySQL oder PostgreSQL aus der Tabelle 3.2.

So dürfte die maximale Größenbeschränkung einer MySQL-Tabelle von 4 Gbyte auf üblichen Linux-Servern und, mit speziellen Tabellentypen sogar 8 Tbyte auf Linux-Alpha-Systemen, eindrucksvoll aufzeigen, welche gewaltigen Datenmengen mit diesem System zu verwalten sind.

Es ist von großer Bedeutung, dass ein Datenbank-Management-System möglichst einfach mit weiteren Software-Komponenten, die zum Betrieb eines Webservers benötigt werden, kombiniert werden kann. Das beste System ist *nutzlos*, wenn keine Skriptsprachen oder Module existieren, mit denen Zugriffe oder Operationen über den Applikations- und Webserver ausgeführt werden können. Je weiter verbreitet ein Online-Datenbanksystem ist, desto mehr Entwickler werden Sie finden, die gute Lösungen anbieten und umsetzen können, und umso höher ist die Wahrscheinlichkeit, dass es bereits eine Lösung – und damit Erfahrungswerte – gibt.

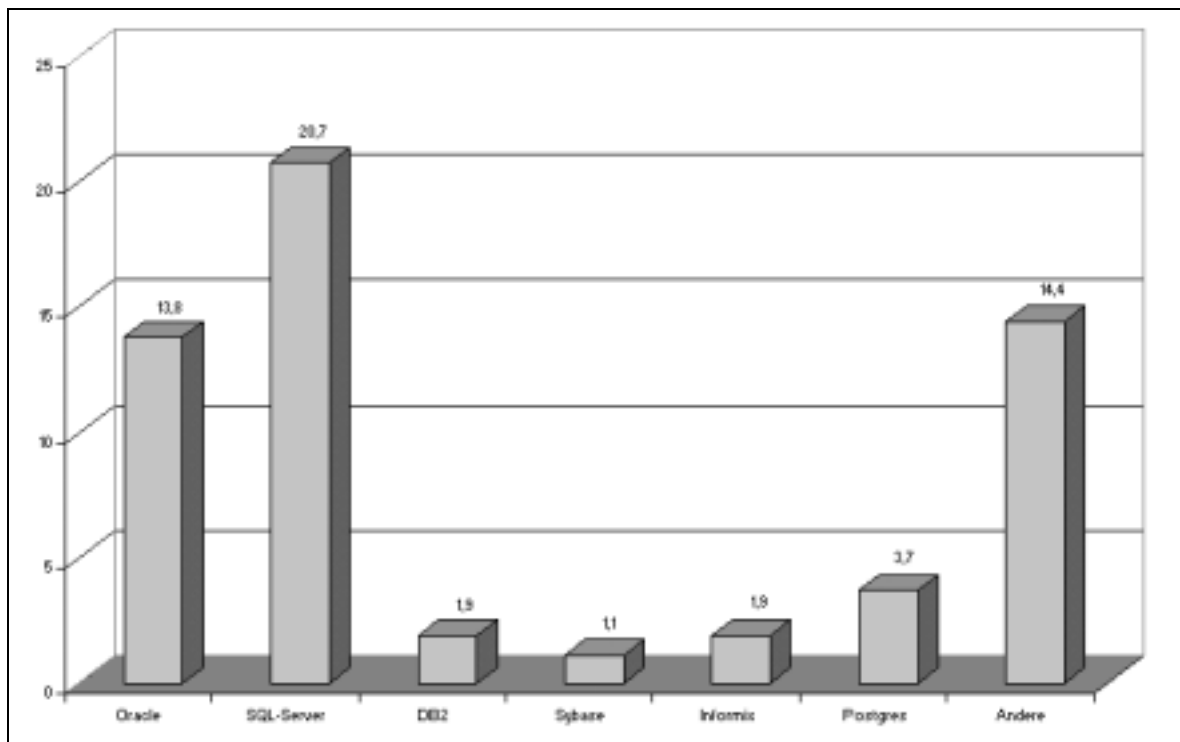


Abbildung 3.5: Von welchem Datenbanksystem wechseln die meisten Entwickler zu MySQL? Umfrage Juli 2002 mit etwa 1.100 Stimmabgaben.

Die Abbildung 3.5 zeigt eindrucksvoll, dass definitiv ein Trend zu **Open-Source-Produkten** wie zum Beispiel MySQL besteht. Für diese Umfrage gaben etwa 1.100 Entwickler ihre Stimme auf die Frage ab, von welchem Datenbanksystem sie zu MySQL gewechselt sind. Spricht man mit den Ent-

wicklern über ihre Gründe, stehen ganz eindeutig Lizenzkosten im Vordergrund, aber auch die Erkenntnis, dass die beiden populärsten Skriptsprachen wie Perl oder PHP sehr gut mit MySQL zusammenarbeiten können und eine hohe Performance zum Gelingen eines Datenbank-Projekts beiträgt. Außerdem sei es nicht so ein großes Problem, einen Provider zu finden, der eine **LAMP-Umgebung (Linux-Apache-MySQL-PHP)** als Web-Space anbietet, was auch eine Statistik (siehe Abbildung 3.6) von Netcraft, die von 1995 bis zum Juli 2000 geführt wurde, eindeutig belegt.



Abbildung 3.6: Einsatzstatistik von Netcraft über Webserver.
Quelle: <http://www.netcraft.com/>

Hier muss jedoch auch wieder darauf verwiesen werden, dass diese Statistik sicherlich nicht völlig korrekt ist, denn Netcraft untersuchte, ohne besondere Zielgruppe, zahlreiche Domains, ohne darauf zu achten, ob es sich bei dem Hosting-Verfahren um einen eigenen oder um einen gemieteten Webserver handelte. Dadurch wirkten sich eine Vielzahl von Domains auf die Auswertung aus, obwohl diese alle auf dem gleichen Webserver eines Providers gehostet wurden. Somit werden die tatsächlichen Nutzungswerte nicht so weit auseinander klaffen, wie das in der Abbildung 3.6 zu sehen ist.

Ob nun der Apache oder der Microsoft IIS die Nase vorne hat, soll auch keinen Entscheidungsgrund darstellen. Wichtig ist nur, dass man sich bei

Online-Projekten nicht einfach nach irgendwelchen Werbeaussagen richten sollte, sondern genau überprüfen muss, welche Technik für das aktuelle Projekt am geeignetsten erscheint – und bei dieser Auswahl haben die Open-Source-Lizenzen in Zukunft auf jeden Fall ein Wörtchen mitzureden.

3 3.5 Sicherheitsaspekte beim Datenbankbetrieb

Die Nutzung einer Online-Präsenz birgt grundsätzlich Risiken in sich, insbesondere, wenn es sich nicht nur um eine Präsentation im Internet handelt, sondern auch das firmeninterne Netz, also das Intranet mit dem Extranet kombiniert werden soll. Wenn Sie für ein solches Projekt auf den Einsatz eines eigenen Webservers nicht verzichten können oder wollen, müssen Sie natürlich selbst auf die Sicherheit der gespeicherten Daten achten. Wie bereits mehrfach angedeutet, möchte ich auf diesen Fall nicht genauer eingehen, da die einzusetzenden Techniken und Produkte einfach zu vielfältig sind, als dass ich sie hier zufrieden stellend vorstellen könnte.

Als Projektleiter sollten Sie jedoch auf jeden Fall darauf achten, dass sich in Ihrem Team zur Realisierung einer datenbankgestützten Online-Präsenz Fachleute zu Sicherheitsfragen befinden. Im Zweifel investieren Sie die Ausgaben und lassen einen externen, zertifizierten Sicherheitsberater bei der Konzeption mitwirken.

Eine andere Möglichkeit besteht darin, es ganz und gar zu vermeiden, eigene Server zu betreiben und auf die Leistungen eines Providers zurückzugreifen – aber Vorsicht, vertrauen Sie nicht blindlings auf die versprochene Sicherheit, denn auch Provider machen Fehler. Jedoch sind Sie zunächst aus dem Schneider, da die Provider meist alle notwendigen Mittel zur Verfügung stellen, die für ausgereiftes E-Business benötigt werden. Funktionen wie Kreditkarten-Transaktionen, Kreditkartenabrechnung, VPNs, Extranets, EDI und weitere können Sie entweder dazubuchen oder bei weiteren speziellen Anbietern ankaufen. So bietet zum Beispiel das Unternehmen WebTRADE.NET einen hervorragenden Service zu allen Fragen rund um das Thema Online-Transaktionen an. Unter <http://www.webtrade.net> können Sie sich ausführlich über Möglichkeiten des E-Payment informieren und sogar Demo-Zugänge in Anspruch nehmen, um die angebotenen Services vor der Nutzung im eigenen Projekt ausgiebig zu testen.

Der Verzicht auf den eigenen Webserver bringt in vielen Bereichen auch eine *erhebliche* Kostenersparnis mit sich, die Sie an anderer Stelle in Ihrem Projekt vielleicht besser gebrauchen können:

- ▶ Sie sparen die Telefongebühren für digitale Leitungen beziehungsweise können auf die monatlichen Mietkosten von Standleitungen verzichten.
- ▶ Sie können auf notwendige Schulungen und Weiterbildungen Ihrer Administratoren im Sicherheitsbereich (zumindest zum Teil) verzichten.
- ▶ Ihnen entfallen die Kosten für Firewall, Bridges, Router und so weiter.
- ▶ Verwenden Sie den Server eines Providers, benötigen Sie nicht unbedingt eigene Zertifizierungen, die zum Teil auch nicht ganz billig sind.
- ▶ etc.

Egal, für welche Variante Sie sich entscheiden, Sicherheit muss ein Thema sein, denn sobald Sie eine Datenbank einsetzen, verschärfen sich sämtliche Anforderungen noch einmal. Genauso wenig, wie ich näher auf die Konfigurationen des eigenen Servers eingehen möchte, verzichte ich auch auf allzu technische Hinweise bei der Erklärung zu Sicherheitsaspekten von Online-Datenbanken. Vielmehr möchte ich mich auf die alltäglichen Kleinigkeiten konzentrieren und Ihnen aufzeigen, mit welchen Gefahren Sie rechnen müssen und wie Sie in Ihrem Projekt etwas dagegen tun können. Speziellere Beispiele finden Sie später in diesem Buch, wenn es um die Umsetzung konkreter Projekte und die Erstellung von Lasten- und Pflichtenheften geht. Im Folgenden zunächst nur eine kleine Übersicht technischer Fallstricke, die unter allen Umständen vermieden werden müssen. Leider kann man bei sehr vielen Online-Projekten eine große Nachlässigkeit dieser Kriterien erkennen – die Risiken sind dann nicht abzuschätzen.

3.5.1 Technische Fallstricke

Durch die Notwendigkeit, Skripten zum Zugriff auf das gewählte Datenbanksystem einsetzen zu müssen, stellen diese Programme das erste aktive Sicherheitsproblem dar. Somit muss während der Planungs- und Entwicklungsphasen in der Dokumentation und auch in der praktischen Umsetzung darauf geachtet werden, dass alle möglichen Kriterien zur Absicherung eingesetzt und verwendet werden. Wenn Skripten auf Datenbanksysteme zugreifen sollen, benötigen sie so genannte Benutzerinformationen. Diese bestehen weit-

gehend aus einem **Benutzernamen** und einem **Passwort**. Diese Informationen müssen jederzeit, für jeden Zugriff auf die Datenbank, abrufbar sein. Einer der schlimmsten Fehler ist die Unterbringung dieser Kenndaten in demselben Skript, das auch den Zugriff auf die Datenbank herstellt.

A)



B)



Abbildung 3.7: Gefahr von Benutzerdaten in Klarschrift ...

Die Abbildung 3.7 zeigt in der Browserdarstellung A) die Ansicht, die Sie bei einer nicht korrekten Konfiguration eines Webservers bekommen können. Achten Sie bitte auf die unterstrichenen Zeilen in dem Source-Code. Für jemanden, der nicht selbst programmiert, scheinen diese Zeilen keinen Sinn zu ergeben, jedoch der Profi erkennt sofort, dass es sich hier um die **Host-Angabe**, den **Benutzernamen** und das **Passwort** für die Datenbank handelt. Die Darstellung B) zeigt das gleiche Dokument bei korrekter Interpretierung der Programmzeilen. Solch eine Verletzung von Sicherheitsrichtlinien ist nicht selten und ich glaube, es wird offensichtlich, was mit diesen Daten für ein Schaden angerichtet werden kann.

Dabei schafft hier eine wirkliche Kleinigkeit schon ein Maximum an Abhilfe:

- ▶ Benutzerdaten werden in ein *anderes* Skript ausgelagert, das das jeweils benutzende Programm lediglich einbindet.
- ▶ In dem zugreifenden Skript erscheinen *niemals* die Kenndaten in Klarschrift, sondern werden in Form von **Platzhalterbezeichnungen (Variablen oder Konstanten)** verwendet.
- ▶ Wenn die Möglichkeit besteht, verschlüsseln Sie Kennwörter (**zum Beispiel mittels MD5**), damit sie selbst bei Entdeckung wertlos bleiben.
- ▶ Diese ausgelagerten »Bibliotheken« werden zusätzlich in ein spezielles Verzeichnis kopiert und gegen unbefugten, externen Zugriff über das **HTTP-Protokoll** abgesichert.
- ▶ Eine spezielle Datei mit dem Namen robots.txt wird so modifiziert, dass dieses Verzeichnis für Such-Robots nicht erreichbar ist. Diese Methode ist zwar keine Garantie dafür, dass sich die Robots daran halten, aber da es sich lediglich um eine Zeile in einer Textdatei handelt, dürfte der Aufwand nicht ins Gewicht fallen.

Jedes Datenbanksystem besitzt eine Benutzerverwaltung und mindestens einen Benutzer, dem volle Administratorrechte eingeräumt sind. Solch ein Benutzer wird auch als **Root** und seine Rechte als **Root-Rechte** bezeichnet. Niemals sollte ein skriptengesteuerter Zugriff über das Netz mit einem Root-Benutzer durchgeführt werden, wenn keine ernsthafte Begründung dafür vorliegt. Sofern der Provider die Anlage neuer Benutzer gestattet, sollten Sie für die normale Nutzung einer Datenbank immer speziell angepasste Benutzer einrichten, die gerade nur die Rechte bekommen, die für die jeweiligen Aktionen notwendig sind.

In diesem Zusammenhang spielt auch noch einmal das genutzte **Betriebssystem** eine große Rolle. Jeder Datenbank-Administrator hat bei üblichen Datenbank-Management-Systemen völlig freien Dateizugriff. Erlauben Sie webbasierte Zugriffe mit diesen Root-Rechten und das Betriebssystem bietet keine ausreichenden Möglichkeiten, die Rechte so zu beschränken, dass nur von der Datenbank genutzte Dateien betroffen sind, erhält der Benutzer indirekt über die Datenbank vollen Zugriff auf das Dateisystem – was ohne Zweifel sehr fatale Auswirkungen haben kann. Noch größer wird die Gefahr bei so genannten **Telnet-Zugriffen**, durch die der Benutzer direkt auf die Betriebssystem-Oberfläche des Servers zugreifen kann.

Beispielsweise benötigt der Benutzer für den Browser aus Abbildung 3.7 B) nur das Recht, eine Datenabfrage (*SELECT*) durchzuführen. Die Rechte zum Anlegen oder Löschen neuer oder vorhandener Datenbanken oder gar zum Löschen von Datensätzen sind für dieses Skript nicht notwendig. Selbst wenn die Benutzerdaten bei solcher Vorgehensweise in falsche Hände geraten, kann der Angriff nicht viel Schaden anrichten.

- ▶ Sobald keine administrativen Aufgaben erledigt werden müssen, sollte niemals mit Root-Rechten auf die Datenbank zugegriffen werden.
- ▶ Steht ein direkter Zugriff auf den Datenbank-Server zur Verfügung, sollten sämtliche Datenbank-Befehle für Web-Benutzer deaktiviert werden, die Schaden an den Datensätzen und an den Rechten hervorrufen können.
- ▶ Legen Sie spezielle Benutzer mit passend eingeschränkten Rechten an, um die Sicherheit ein weiteres Mal zu erhöhen.

Wird die Datenbank und die Online-Präsenz ausnahmslos in einem Netz verwendet, in dem sämtliche Benutzer und Rechnerstationen bekannt sind, kann eine Datenbank so konfiguriert werden, dass nur Benutzer von bestimmten Rechnern für einen Zugriff auf die Daten zugelassen werden. Erfolgt somit ein Aufruf von nicht autorisierter Stelle, blockt das Datenbank-Management-System diese Anfrage ohne Wenn und Aber ab. In solchen speziellen Fällen kann sogar häufig der Port eines HTTP-Protokolls (**üblicherweise Port 3306**) gesperrt werden, wodurch keinerlei Zugriffe mehr von außerhalb des Intranets möglich sind.

Erfolgen Datenbankzugriffe personalisiert durch Mitarbeiter, so greift erneut die in diesem Buch angesprochene Richtlinie zum Umgang mit persönlichen Benutzerdaten beziehungsweise Passwörtern.

Achten Sie darauf, dass auch Log-Dateien, die üblicherweise bei der Verwendung von Serverprodukten zur Verfügung stehen, nicht in falsche Hände geraten und keine kritischen Informationen über Zugangsdaten enthalten.

Last but not least sollten Sie überall, wo die Anforderungen es zulassen und persönliche Daten eingegeben werden, mit einer geschützten Datenübermittlung (z.B. **SSL-Verbindung**) arbeiten und ein sauber programmiertes **Session-Management** verwenden. Sessions kontrollieren und organisieren unterschiedliche Zugriffe und unterscheiden mehrere gleichzeitige Anforderungen voneinander. Mit anderen Worten kann bei der Nutzung von Sessions kein Benutzer – sei es mutwillig oder unfreiwillig – dem anderen ins Gehege pfuschen. Sprachen wie Perl oder PHP unterstützen solche Methoden selbstverständlich und somit liegt es speziell an den Entwicklern, ob sie solche Funktionen einsetzen oder nicht.

Als Fazit bleibt zu sagen, dass eine vollständige Absicherung von Datenbanksystemen nur mit tief greifenden Kenntnissen über die Struktur der Systeme selbst, der zusätzlich eingesetzten Software-Produkte sowie der jeweiligen Betriebssysteme, auf denen die Datenbank-Server betrieben werden, möglich ist – wenn überhaupt. Trotz aller Möglichkeiten, die heute bereits existieren, kann aber nach Auffassung der meisten Fachleute niemals eine hundertprozentige Sicherheit gewährleistet werden. Somit bleibt lediglich, alles Mögliche zu tun, um es Angreifern nicht zu leicht zu machen, und – vor allem – die Auswertung von Logfiles während des Betriebs einer datenbankgestützten Online-Lösung zur Pflicht zu erklären. Somit gehören solche Kriterien in die Konzeption eines Datenbankprojekts und damit speziell in die Dokumentation. In jedem Lastenheft sollten notwendige Forderungen zur Sicherheit aufgeführt werden und spätestens im Pflichtenheft müssen sich Lösungen zu den genannten Forderungen wiederfinden.

Checkliste »Absicherung«			
Maßnahmen, Hinweise	Ja	Nein	Bemerkungen
Eigener Webserver			
Beschäftigen Sie einen Provider, der auch in der Sicherheitsadministration sehr gute Kenntnisse besitzt?	<input type="checkbox"/>	<input type="checkbox"/>	
Sorgen Sie für ausreichende Weiterbildung dieser Person(en)?	<input type="checkbox"/>	<input type="checkbox"/>	

Checkliste »Absicherung«			
Maßnahmen, Hinweise	Ja	Nein	Bemerkungen
Setzen Sie eine Firewall für Ihren Server und die Zugriffe ein?	<input type="checkbox"/>	<input type="checkbox"/>	
Beschränken Sie Root-Zugriffe derart, dass sie noch kontrollierbar bleiben?	<input type="checkbox"/>	<input type="checkbox"/>	
Sind alle externen Benutzer entsprechend in ihren Rechten beschränkt, damit keine Sicherheitslecks entstehen?	<input type="checkbox"/>	<input type="checkbox"/>	
Fremder Webserver			
Sorgt der Provider für ausreichenden Schutz bei administrativen Zugriffen?	<input type="checkbox"/>	<input type="checkbox"/>	
Sind die Sicherheitszertifikate des Providers auf dem neuesten Stand?	<input type="checkbox"/>	<input type="checkbox"/>	
Haben Sie die Möglichkeit, eigene Sicherheitszertifikate über den Provider zu buchen und zu installieren?	<input type="checkbox"/>	<input type="checkbox"/>	
Können Sie Ihre eingesetzten Systeme so absichern, dass Benutzer auch nur die für sie notwendigen Funktionen nutzen können – nicht mehr?	<input type="checkbox"/>	<input type="checkbox"/>	
Sorgen Sie dafür, dass in Skripten verwendete Zugangsdaten niemals in Klarschrift eingesetzt werden?	<input type="checkbox"/>	<input type="checkbox"/>	
Werden in einem Fehlerfall der Server keine Benutzerdaten sichtbar?	<input type="checkbox"/>	<input type="checkbox"/>	
...	

Tabelle 3.3: Checkliste potenzieller Gefahren bei Server-Betrieb