

Dieser Artikel ist Teil des
Open Source Jahrbuchs 2006

Bernd Lutterbeck
Matthias Bärwolff
Robert A. Gehring (Hrsg.)

Open Source
Jahrbuch 2006

Zwischen Softwareentwicklung und Gesellschaftmodell

erhältlich unter www.opensourcejahrbuch.de.

Die komplette Ausgabe enthält viele weitere interessante Artikel. Lob und Kritik zu diesem Artikel sowie weitere Anregungen können Sie uns einfach und unkompliziert mitteilen per E-Mail oder auf www.opensourcejahrbuch.de/feedback/.

Freie Datenbanken im Unternehmenseinsatz: Analyse und Vergleich der wichtigsten Open-Source-Datenbanken

JUTTA HORSTMANN



(CC-Lizenz siehe Seite 499)

Immer mehr Serverdienste werden durch Open-Source-Software abgedeckt. Nach dem Betriebssystem Linux und dem Webserver Apache kamen schnell weitere Komponenten hinzu, wie Mailserver oder Verzeichnisdienste. Der Bereich der Datenbank-Server hingegen ist weiterhin in proprietärer Hand und dominiert von den großen Anbietern Oracle, Microsoft und IBM. Doch mittlerweile entsteht ein Trend hin zur Open-Source-Datenbank, der sich aus drei Quellen speist: den guten Erfahrungen mit MySQL im Rahmen des LAMP-Stacks, der Reifung und Konsolidierung von Open-Source-Datenbanken in Bezug auf Features, Benutzerfreundlichkeit und Support sowie nicht zuletzt dem Wunsch der Anwender nach Einsparungen im Softwarebereich. Dieser Artikel führt in die Welt der Open-Source-Datenbanken ein, stellt fünf Kandidaten vor (Firebird, Ingres, MaxDB, MySQL, PostgreSQL) und vergleicht deren Features mit Blick auf den Einsatz im Rahmen großer Unternehmensanwendungen.

Schlüsselwörter: Open-Source-Datenbanken · Feature-Übersicht · Migration
· Marktübersicht · Unternehmenseinsatz

1 Einführung

Das Jahr 2005 war ein aufregendes im Land der Open-Source-Datenbanken.¹ Neue Versionen von MySQL (5.0) und PostgreSQL (8.0, 8.1) warteten mit wesentlichen

¹ Im Folgenden wird aus Gründen der Lesbarkeit häufig der Ausdruck „Datenbank“ oder „Datenbanksystem“ verwendet, wenn „Datenbankmanagementsystem“ die korrekte Bezeichnung wäre. Ein Datenbanksystem ist der Oberbegriff für ein System bestehend aus einer Datenbank (der physische Ort der Datenspeicherung), einem Datenbankmanagementsystem (Software zum Aufbau, zur Kontrolle und

Funktionserweiterungen auf. Oracle gelang mit dem Kauf von Innobase Oy der Zugriff auf eine wichtige Komponente von MySQL: den InnoDB-Tabellentyp. SUN startete kommerziellen Support für PostgreSQL. Mit Bizgres steht seit diesem Jahr ein vollständiger Open-Source-Business-Intelligence-Stack zur Verfügung. Und schließlich wurde im November in Frankfurt das erste Mal eine internationale Konferenz veranstaltet, die sich alleine dem Thema Open-Source-Datenbanken widmete.² Grund genug also, sich auch in diesem Buch des Themas anzunehmen.

Wie die meisten Bereiche der Open-Source-Software-Welt bietet auch der Datenbanksektor eine bunte Vielfalt an Auswahlmöglichkeiten. Da gibt es Software mit langjähriger Open-Source-Vergangenheit wie MySQL und PostgreSQL und Neueinsteiger wie Firebird, Ingres und MaxDB.

Und neben den genannten „großen“ Datenbank-Servern, die den Vergleich mit bekannten Namen wie Oracle oder MS SQL nicht scheuen müssen, stehen viele weitere Systeme, die in speziellen Anwendungsbereichen vorzügliche Dienste leisten, indem sie es ProgrammiererInnen ermöglichen, Datenbanken direkt in ihre Anwendung einzubetten. Beispiele sind die C-Bibliotheken SQLite und Berkeley DB³ oder die Java-Datenbanken HSQLDB⁴ und Derby.⁵

Dieser Artikel wird die „Big 5“ (Firebird, Ingres, MaxDB, MySQL, PostgreSQL) mit Blick auf ihren Einsatz im Rahmen großer Unternehmensanwendungen vorstellen und vergleichen. Dabei soll es nicht darum gehen, eine „Rangliste“ zu produzieren. Der Ansatz ist vielmehr, dass für jede Aufgabe das passende Werkzeug verwendet werden sollte. Das heißt die Entscheidung für ein bestimmtes Datenbankmanagementsystem (DBMS) muss sich an den Anforderungen des jeweiligen Einsatzszenarios orientieren.

Im Falle einer Migration von einem vorhandenen proprietären Datenbanksystem zu einer Open-Source-Lösung wird die Entscheidung für ein Ziel-System von den Funktionen und Features abhängen, welche die vorhandene Datenbank zur Verfügung stellt. Das heißt eine detaillierte Analyse des vorhandenen Systems ist unumgänglich. Dieser Artikel wird nicht speziell auf Migrationen eingehen, da die Evaluation der vorhandenen Open-Source-Datenbanken sowohl im Rahmen einer Migration als auch bei der Neuentwicklung eines Systems eine Rolle spielt und die selben Fragen aufwirft. Eine detaillierte Analyse des Migrations-Workflows für Open-Source-Datenbanken findet sich in Horstmann (2005a).

Der Aufbau des Artikels orientiert sich an einem realen Szenario: Ein Unternehmen sucht ein Datenbanksystem, welches bestimmte Anforderungen erfüllt und möchte dabei auch Open-Source-Datenbanken berücksichtigen. Der Entscheidungsfindungsprozess wird dabei von den folgenden Fragen und Überlegungen geleitet sein:

zur Manipulation von Datenbanken) und einer Datenkommunikationsschnittstelle (Kommunikation zwischen Datenbank und Anwendungsprogrammen).

2 <http://www.opendbcon.net>

3 <http://www.sqlite.org>, <http://www.sleepycat.com>

4 <http://hsqldb.org>

5 <http://db.apache.org/derby>

1. Können wir uns Open-Source-Software im Datenbankbereich vorstellen?
Was sind Vor- und Nachteile?
2. Welche Open-Source-Datenbanken stehen überhaupt zur Verfügung?
3. Welche Anforderungen stellen wir an das zukünftige Datenbanksystem?
4. Evaluation des Datenbank-Angebots im Open-Source-Bereich
 - a) Sammeln und Strukturieren von Vergleichskriterien
 - b) Vergleich von interessanten Kandidaten
5. Auswahl derjenigen Open-Source-Datenbank, welche am besten zu unseren Anforderungen passt

Diese Schritte sollen im Folgenden nachvollzogen werden.

2 Lieber nicht das Orakel fragen: Was spricht für Open-Source-Datenbanken, was dagegen?

Vor der Entscheidung für eine Open-Source-Datenbank muss man die Vor- und Nachteile eines solchen Schrittes abwägen. Während einige Befürchtungen gegenüber Open-Source-Datenbanken im nächsten Abschnitt entkräftet werden können, lassen sich andere nicht wegdiskutieren. Sie stehen den Vorteilen von Open-Source-Datenbanken gegenüber, die im Weiteren auch analysiert werden.

2.1 Contra Open-Source-Datenbanken

Gerade wenn im Unternehmen bisher wenig Erfahrung mit Open-Source-Software gemacht wurde, stehen viele Entscheider diesem Softwareentwicklungsmodell skeptisch gegenüber. Man befürchtet informellen Support, mangelnde langfristige Planung, ein kompliziertes Wirrwarr unterschiedlicher Lizenz-Modelle, unklare Haftbarkeiten und erhöhten administrativen Aufwand. Speziell *Open-Source-Datenbanken* wird ein Mangel an Features im Vergleich zu den proprietären Pendanten wie Oracle oder MS SQL Server unterstellt. Und nicht zuletzt stellt sich die Frage nach der Verfügbarkeit von Applikationen, die in der Lage sind, mit Open-Source-Datenbanken zusammenzuarbeiten.

Einige dieser Befürchtungen sind leicht zu entkräften. So gibt es für alle fünf großen Open-Source-Datenbanken Anbieter kommerziellen Supports: Zum Beispiel MySQL AB für MySQL, Computer Associates und Ingres Corp. für Ingres, IB-Phoenix (international) und HK-Software (Deutschland) für Firebird, SAP AG und MySQL AB für MaxDB und eine große Auswahl internationaler Firmen (u. a. Fujitsu und SUN) für PostgreSQL (Support im deutschsprachigen Raum bieten u. a. credativ

und Cybertec). Hinzu kommt, dass informeller Support oft besser ist als sein Ruf. Wer die Community-Blogs und -Mailinglisten verfolgt, lernt dabei mehr als durch das Studium einiger Regalmeter Handbücher. Und die Entwickler der jeweiligen Software direkt per IRC, Mail, Foren oder Bug-Tracker ansprechen zu können, ist für viele AnwenderInnen hilfreicher als die Nutzung einer Telefon-Hotline.

In eine ähnliche Richtung gehen Fragen der Gewährleistung und Haftbarkeit. Bei den drei Datenbanken, die Firmen gehören (MySQL, Ingres, MaxDB), ist auf jeden Fall genauso viel (oder wenig) Haftung wie im Closed-Source-Software-Bereich zu erwarten. Hinzu kommt, dass durch offenen Code und offene Bugtracking-Systeme eine Transparenz hergestellt wird, die Zurechenbarkeit und schnelles Ausmerzen von Fehlern erlaubt.

Was die langfristige Planung anbelangt, so befinden sich auf den Webseiten fast aller Open-Source-Datenbanken ausführliche Roadmaps.⁶

Das Thema Lizenzierung ist schon etwas komplizierter.⁷ Die fünf großen Open-Source-Datenbanken lizenzieren alle auf unterschiedliche Weise: Firebird verwendet die IPL,⁸ Ingres die CATOSL,⁹ MySQL und MaxDB duale Lizenzierung¹⁰ (GPL und kommerziell) und PostgreSQL steht unter der BSD-Lizenz.¹¹

Dass der Einsatz von Open-Source-Software erhöhten administrativen Aufwand, eine steile Lernkurve und geringe Benutzerfreundlichkeit¹² mit sich bringt, ist ein Vorurteil, das sich im Datenbankbereich nicht bestätigt. Neben sehr ausgereiften Kommandozeilen-Werkzeugen bringen alle fünf Datenbanken selbstverständlich auch Tools mit grafischen Oberflächen mit. Hinzu kommt, dass sie auch meist einfacher zu administrieren sind als ihre proprietären Gegenparts – mit ein Grund für den enormen Erfolg MySQLs im Bereich der Web-Anwendungen.

Zwei Argumente gegen den Einsatz von Open-Source-Datenbanken wiegen jedoch schwer. Zum einen ist dies die Frage nach der Unterstützung von Open-Source-Datenbanken durch Software-Hersteller. Also: Kann ich das Datenbankdesign-Tool oder die *Business Intelligence Suite*, die ich bisher mit proprietären Datenbanken genutzt habe, weiter einsetzen? Die Antwort lautet immer noch oft: Nein, das geht leider nicht.

Und zweitens muss man den Umfang an unterstützten Features erwähnen. Obwohl alle fünf Open-Source-Datenbanken einen sehr großen Funktionsumfang mitbringen,

6 MySQL: <http://dev.mysql.com/doc/mysql/en/roadmap.html>,
PostgreSQL: <http://www.postgresql.org/developer/roadmap>,
Firebird: <http://www.firebirdsql.org>

7 Verschiedene Internet-Quellen (z. B. <http://www.opensource.org>, <http://openfacts.berlios.de/index.phpml?title=Open-Source-Lizenzen>) sowie der Artikel von Ursula Widmer und Konrad Bähler auf Seite 165 in diesem Buch bieten aber Unterstützung bei der Navigation durch den Lizenzen-Dschungel.

8 <http://info.borland.com/devsupport/interbase/opensource/IPL.html>

9 <http://ca.com/opensource/catosl>

10 <http://www.mysql.com/company/legal/licensing>

11 <http://www.opensource.org/licenses/bsd-license.php>

12 Zum Thema Open-Source-Usability siehe <http://www.openusability.org>.

gibt es doch immer wieder Anforderungen, bei denen einzelne oder sogar alle passen müssen. *Materialized Views* – eine Technik, die im Data-Warehouse-Bereich genutzt wird, um die Bereitstellung von Daten auf komplexe Abfragen zu beschleunigen – stellt zum Beispiel keines der Projekte zur Verfügung. Abschnitt 4.2 geht im Detail auf die vorhandenen und fehlenden Features der fünf Datenbanken ein.

2.2 Pro Open-Source-Datenbanken

Zwei Argumente stehen im Vordergrund, wenn es darum geht, den Einsatz von Open-Source-Datenbanken zu begründen. Um genau zu sein, sind es hier die selben Argumente, die für Open-Source-Software im Allgemeinen meist ins Feld geführt werden: die niedrigen Kosten sowie die Vorteile des Open-Source-Softwareentwicklungsmodells.

Kostenberechnungen sind kompliziert, weil man nicht nur die Lizenzkosten, sondern auch die mit dem Einsatz und der Administration der Software verbundenen Arbeitskosten mit berücksichtigen muss. Wheeler (2005) stellt eine Liste von Studien zusammen, die die TCO (Total Cost of Ownership, die gesamten mit Erwerb und Betrieb einer Software verbundenen Kosten) von Open-Source- und Closed-Source-Software vergleichen, wobei meist die Open-Source-Szenarien besser abschneiden.

MySQL (2003) zum Beispiel nimmt für sich in Anspruch, weniger als halb so viel Kosten zu verursachen wie der Einsatz einer Closed-Source-Datenbank:

- Lizenzkosten reduziert um über 90 %
- Ausfallzeiten des Systems reduziert um 60 %
- Hardware-Ausgaben reduziert um 70 %
- Kosten für Administration, Entwicklung und Support reduziert um fast 50 %

Aber neben niedrigeren Kosten hat Open-Source-Software einen weiteren ganz entscheidenden Vorteil, und der ist – ganz simpel – der offene Quellcode. Der Einsatz von Open-Source-Datenbanken in einem Unternehmen bringt die folgenden Vorteile mit sich:

- Teilhabe an einer weltweiten Community inklusive dem direkten Zugang zu den Datenbank-EntwicklerInnen, zu umfassender Dokumentation und Support.
- Unabhängigkeit von einzelnen Anbietern und Produkt-Lebenszyklen. So besteht z. B. die Möglichkeit, dass IBM ab 2009 das Datenbank-Produkt Informix aufgibt (Flannery 2005). Demgegenüber bleibt Open-Source-Code zugänglich und kann von jedem weiterentwickelt werden, der das Projekt weiterführen möchte (vgl. z. B. die Weiterentwicklung des Mozilla Browsers trotz Beendigung des Projekts durch die Mozilla Foundation¹³).

13 <http://www.mozilla.org/projects/seamonkey>

- Ein hohes Sicherheits-Niveau aufgrund von ständiger Überprüfung des Codes durch eine internationale Entwicklergemeinschaft und schnelle Fehlerbeseitigung. So wird beispielsweise PostgreSQL im „Database Hacker’s Handbook“ im Vergleich mit verschiedenen Closed-Source-Datenbanken als auf der Grundlage der Standardeinstellungen sicherste Datenbank bezeichnet („by default [. . .] possibly the most security-aware database available“, Litchfield et al. 2005, S. 5).
- Flexible Anpassungen und Erweiterungen der Software entsprechend der eigenen Anforderungen, indem fehlende Funktionalitäten selbst (oder von beauftragten EntwicklerInnen) hinzugefügt werden.

Dass außerdem Open-Source-Software-Entwicklung zu besserem Code führt, ist nicht nur ein Mythos, der auf dem berühmten Zitat Eric S. Raymonds (2000) basiert: „Given enough eyeballs, all bugs are shallow.“¹⁴ Kürzlich analysierte Coverity, ein bekannter Produzent von Software-Analyse-Programmen, verschiedene Open-Source-Projekte. Der Linux Kernel (2.6.9) enthielt weniger als zwei Mängel auf 10 000 Zeilen Code.¹⁵ MySQL hatte weniger als drei Fehler in 10 000 Zeilen (Lemos 2005), während in PostgreSQLs Code sogar nur 0,25 Bugs auf 10 000 Zeilen gefunden wurden.¹⁶ Software mit geschlossenem Quellcode enthält hingegen nach Studien der Carnegie-Mellon-University zwischen 10 bis 300 Fehler pro 10 000 Zeilen (O’Neill 2003; Lemos 2004; Delio 2004).¹⁷

2.3 Zusammenfassung

Für den Einsatz von Open-Source-Datenbanken sprechen niedrigere Kosten sowie die Vorteile des Zugangs zum offenen Code, zu einer internationalen Community sowie zu den Entwicklern selbst. Die meisten Befürchtungen gegenüber ihrer Verwendung, wie z. B. mangelnder kommerzieller Support oder erhöhter administrativer Aufwand, können ausgeräumt werden.

Offen bleibt die Frage, ob die fünf Open-Source-Datenbanken einen Funktionsumfang mitbringen, der sich mit den proprietären Marktführern messen kann und die Produkte damit reif für den Einsatz in Unternehmensanwendungen macht. Dieses Thema soll im Folgenden beleuchtet werden.

14 Wenn nur genug Leute auf den Code schauen, haben Bugs keine Chance mehr.

15 http://www.coverity.com/datasheets/linux_report.pdf

16 <http://archives.postgresql.org/pgsql-www/2005-07/msg00067.php>,
http://www.enterprisedb.com/news_events/press_releases/07_12_05.do

17 <http://www.cyberpartnership.org/SDLFULL.pdf>

3 Von Elefanten, Delphinen, müden Katzen und feurigen Vögeln: Ein Spaziergang durch den Datenbank-Zoo

Man kann die Datenbank-Welt in zwei Gruppen unterteilen, die sich nach ihrer Architektur unterscheiden. Auf der einen Seite stehen die klassischen Client-Server-Systeme wie z. B. PostgreSQL oder MaxDB, auf der anderen Seite die eingebetteten Datenbanken. (Hinzu kommen hybride Systeme wie Firebird, die beide Architekturen unterstützen.) Dieser Artikel konzentriert sich auf die fünf Datenbank-Server Firebird, Ingres, MaxDB, MySQL und PostgreSQL. Bevor diese Systeme jedoch im Detail vorgestellt und verglichen werden, soll ein kurzer Blick auf die eingebetteten Datenbanken geworfen werden, die auf Open-Source-Lösungen auch in diesem Bereich hinweisen.¹⁸

3.1 Die „Kleinen“

Eine eingebettete Datenbank wird von der Applikation nicht über Inter-Prozess-Kommunikation angesprochen (z. B. TCP/IP oder UNIX Sockets), sondern läuft im selben Prozessraum wie die Anwendung selbst. Dies ermöglicht erhebliche Performanz-Gewinne. Die Datenbank wird dabei als Bibliothek gelinkt oder ist direkt ein Teil des Anwendungs-Codes. Sie zeichnet sich durch geringe Größe und Speicherbedarf aus und erfordert meist keine Administration oder Konfiguration. Beliebte Open-Source-Datenbanken im Embedded-Bereich sind Berkeley DB, Derby, HS-QLDB, Mckoi, SQLite und One\$DB.¹⁹ Tabelle 1 bietet einen Überblick über einige Charakteristika dieser Systeme.

Nach den eingebetteten Datenbanken sollen nun im Folgenden die fünf großen Open-Source-Datenbankserver vorgestellt werden.²⁰

3.2 Firebird

Firebird basiert auf InterBase, dem Datenbankmanagementsystem der Firma Borland.²¹ Dadurch reichen seine Wurzeln zurück bis 1984. Im Jahr 2000 entschied sich Borland, den Quellcode von InterBase v6.0 freizugeben, da das Datenbanksystem über keinen nennenswerten Marktanteil mehr verfügte. Die Hoffnung bestand,

18 Die „tierische“ Überschrift dieses Abschnitts bezieht sich auf die Maskottchen und Namen einiger Open-Source-Datenbanken: Der Elefant zielt das PostgreSQL-, der Delphin das MySQL-Logo und der Feuervogel das von Firebird. Sleepycat vermarktet die Berkeley DB.

19 <http://www.sleepycat.com>, <http://db.apache.org/derby>, <http://hsqldb.org>, <http://www.mckoi.com>, <http://www.sqlite.org> und <http://www.daffodildb.com/one-dollar-db.html>

20 Neben verschiedenen im Text genannten, haben auch folgende Quellen wesentliche Informationen geliefert: Fallmann et al. (2005), Manhart (2004), Arvin (2005) sowie die Websites http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems und <http://www.geocities.com/mailsoftware42/db>.

21 Siehe <http://firebird.sourceforge.net> und <http://www.firebird-datenbank.de> sowie Farooqi (2004).

	Berkeley DB	Derby	HSQLDB	Mckoi	SQLite	One\$DB
Firma	Sleepycat	IBM/ Apache Software Foundation	Keine	Keine	Keine	Daffodil
Lizenz	dual (Sleepycat Public/ Commercial Licence)	Apache License	Hypersonic License „BSD-style“	GPL	Public Domain	LGPL
Sprache	C oder Java	Java	Java	Java	C	Java
Bekannte Anwen- dungen	Subversion, OpenLDAP, MySQL Storage Engine, Google Accounts	–	OpenOffice Base	–	Amarok (KDE), Aperture (Apple)	–
Relational	nein	ja	ja	ja	ja	ja

Table 1: Eingebettete Open-Source-Datenbanken

durch kommerziellen Support für ein durch „frisches Blut“ aus der Open-Source–Community wiederbelebtes Produkt höhere Einnahmen zu erzielen als bisher über Lizenzkosten. Diese Rechnung ging offensichtlich nicht auf, denn wenig später (in Version 7.1) schloss Borland den Code wieder.

In der Zwischenzeit hatte sich jedoch eine aktive Community um die Codebasis gebildet, welche die Software seitdem unter dem Namen Firebird eigenständig weiterentwickelt. Bisher ist Firebird in Westeuropa und den USA wesentlich weniger bekannt als zum Beispiel MySQL oder PostgreSQL. Nichtsdestotrotz ist es vor allem in Brasilien, Russland und der Ukraine weit verbreitet²² und verfügt über eine aktive Fangemeinde.²³ Ein großer Teil der Anwender rekrutiert sich aus vormaligen Inter-Base-Kunden, von denen einer der hierzulande bekanntesten Nutzer die Deutsche Presse Agentur (dpa) ist.²⁴

3.3 Ingres

Die Entwicklung an Ingres begann im Jahr 1974 als Forschungsprojekt der University of California in Berkeley. 1982 wurde Ingres unter dem Dach der Firma Ingres

22 <http://ibdeveloper.com/2005/10/27/firebird-global-survey-results>

23 Firebird wurde in der Newsforge-Abstimmung (2004) zur „Favorite Open Source Database“ gewählt, „Preferred Database“ in der Umfrage von O’Reilly (2004), knapp hinter MySQL in LinuxQuestions „Database Of The Year“-Umfrage (2004), ebenso in der Abstimmung von Evans Data Corp. (2005) über „Open Source Databases Currently Used“.

24 http://www.ibphoenix.com/main.nfs?a=ibphoenix&s=1131278527:192093&page=ibp_users_firebird

Corporation kommerzialisiert. Das Unternehmen wurde dann zwölf Jahre später von Computer Associates (CA) gekauft.²⁵ Der selben Logik wie Borland folgend, veröffentlichte CA im Jahre 2004 den Quellcode von Ingres, bot aber weiterhin kommerziellen und Community-Support an. Im Herbst 2005 gliederte CA Ingres aus und verkaufte das Datenbanksystem an die Investment-Firma Garnett & Helfrich Capital, die es nun wiederum unter dem Namen Ingres Corporation weiterhin als Open Source anbietet.²⁶

3.4 MaxDB

Die Wurzeln von MaxDB liegen in Deutschland, wo das System zunächst als Forschungsprojekt namens „Distributed Databases on Mini-Computers“ an der TU Berlin entwickelt wurde.²⁷ 1981 wurde es von Nixdorf unter dem Namen DDB/4 kommerzialisiert und 1993 von der Software AG übernommen, welche die Datenbank zunächst in „Entire SQL DB Server“, später in „Adabas D“ umbenannte. SAP übernahm die Software im Jahr 1997 und änderte erneut den Namen, diesmal in „SAP DB“. 2000 stellte SAP die Datenbank als Open Source unter die GPL. Drei Jahre später gingen SAP und MySQL AB eine Technologie-Partnerschaft ein, die MySQL sowohl den Vertrieb und die Weiterentwicklung als auch eine erneute Umbenennung des Datenbanksystems erlaubte.²⁸ MaxDB wird jedoch auch weiterhin nahezu ausschließlich von SAP entwickelt, was auch die SAP-Zertifizierung mit einschließt.

MySQL AB offeriert seine Produkte unter einer so genannten dualen Lizenzierung: Wird die Software im Rahmen einer Applikation genutzt, welche selbst unter der GPL steht, so ist das Produkt kostenfrei. Ist dies nicht der Fall – zum Beispiel dient die Datenbank als Backend einer kommerziellen Closed-Source-Applikation –, so bietet MySQL AB eine kommerzielle Lizenzierung (d. h. verbunden mit Lizenzkosten) an.

SAP nutzte das Datenbanksystem als Backend der haus eigenen Großanwendungen wie z. B. SAP R/3. Es ist daher weltweit in SAP-Systemen so bekannter Firmen wie Toyota, Intel, DaimlerChrysler, Bayer, Yamaha und der Deutschen Post zu finden.

3.5 MySQL

MySQL wurde im Jahr 1995 von der schwedischen Firma MySQL AB veröffentlicht.²⁹ Open-Source-Software von Anfang an, entwickelte sich schnell eine sehr große Fange-

25 <http://opensource.ca.com/projects/ingres>

26 Weiterführende Informationen über den Ausgliederungsprozess bieten Loftus (2005), Vaas (2005a) sowie die neue Webseite <http://www.ingres.com>.

27 <http://dev.mysql.com/doc/refman/5.0/en/maxdb.html>, <http://www.sapdb.org/history.htm>, <http://www.mysql.com/products/maxdb>

28 Es ist kein Zufall, dass die zwei Kinder von einem der MySQL-Gründer die Namen My und Max tragen.

29 <http://www.mysql.com>

meinde rund um das Datenbanksystem, welches eine der vier Säulen des so genannten LAMP-Stacks bildet (Linux, Apache, MySQL, PHP/Python/Perl).

Wie MaxDB wird auch MySQL unter dualer Lizenzierung angeboten. Bekannte Nutzer sind Associated Press, Yahoo, Slashdot, SourceForge, die NASA, Lycos und T-Systems.³⁰ In einer Reihe von Open-Source-Projekten arbeitet MySQL als Backend, z. B. Typo3 und Drupal (CMS), Nucleus und Wordpress (Blog), Wikipedia/MediaWiki, PHPProjekt und eGroupware (Projekt Management/Groupware). MySQL AB bietet kommerziellen und Community-Support.

3.6 PostgreSQL

Das 1986 auf Ingres folgende Datenbank-Forschungsprojekt der Universität Berkeley wurde Postgres getauft.³¹ Als das Projekt im Jahr 1994 auslief, wurde noch ein SQL-Interpreter zum Code hinzugefügt und die Software umbenannt in Postgres95. Das Ganze wurde dann im Internet unter der BSD-Lizenz als Open-Source-Software veröffentlicht. 1996 einigte man sich schließlich auf den Namen PostgreSQL.

PostgreSQL ist ein Objekt-relationales Datenbanksystem mit einer großen Nutzer- und Entwickler-Community. Verschiedene Unternehmen bieten kommerziellen Support an, z. B. Pervasive Software in den USA, Cybertec in Österreich, Creditiv in Deutschland, SRA in Japan und Fujitsu in Australien. Zuletzt kündigte SUN Microsystems an, PostgreSQL als Datenbank in ihr Portfolio zu integrieren und zu unterstützen.

Bekannte Nutzer von PostgreSQL sind Afilias (Domain-Namen-Registrar), BASF und Fujitsu. Die ZEIT nutzt PostgreSQL zur Speicherung der Metadaten ihres Content-Management-Servers.³² In verschiedenen Open-Source-Projekten kommt PostgreSQL als Backend zum Einsatz, z. B. in der Kollaborationsplattform GForge, ERP-/CRM-/Groupware-Systemen wie EGS, FISTERRA und FacturaLUX sowie in verschiedenen Projekten, die sich der GIS-Implementation von PostgreSQL bedienen.

Spezifisch für PostgreSQL sind die so genannten Distributionen: Unternehmen können selbstentwickelte Features zum Datenbank-Code hinzufügen und dann selbst verteilen oder verkaufen – die BSD-Lizenz erlaubt das Schließen des Codes. Beispiele sind Greenplum mit Bizgres und EnterpriseDB.³³

4 Ans Eingemachte: Der Vergleich von fünf Open-Source-Datenbanken

Ein reines Nebeneinanderstellen von Datenbank-Features greift zu kurz und ist beim nächsten Release schon wieder obsolet. Daher wird im Folgenden zunächst ein Kri-

30 <http://www.mysql.com/customers>

31 <http://www.postgresql.org>

32 <http://www.zeit.de/software/index?page=all>

33 <http://www.greenplum.com>, <http://www.enterprisedb.com>

terienschema vorgestellt. Ein Unternehmen, das vor der Entscheidung steht, eine Open-Source-Datenbank einzusetzen, kann anhand dessen auch zukünftige Versionen umfassend analysieren und vergleichen.

Im Anschluss daran werden die aktuellen Versionen der fünf Open-Source-Datenbanken auf der Grundlage der vorgestellten Kriterien verglichen. Ein kurzer Überblick über die Fähigkeiten der Datenbanken im Data-Warehouse-Bereich rundet das Bild ab.

4.1 Äpfel und Birnen: Nach welchen Kriterien wird verglichen?

Ob der Einsatz einer Open-Source-Datenbank möglich und sinnvoll ist, hängt von den Anforderungen ab, die an den Funktionsumfang des Systems gerichtet werden. Dieser Abschnitt bietet einen Überblick über die Features der fünf vorgestellten Open-Source-Datenbanken unter dem Gesichtspunkt, ob sie für den Einsatz in anspruchsvollen Unternehmensanwendungen geeignet sind. Diese Szenarien unterscheiden sich vom Endnutzer- oder Hobby-Einsatz durch Größe, Ressourcenverbrauch und Funktionsumfang und in ihren Ansprüchen hinsichtlich Mehrbenutzerfähigkeit, Sicherheit, Datenintegrität und Hochverfügbarkeit.

Ein Datenbanksystem sollte daher möglichst die unten aufgelisteten Features bieten, um in großen Unternehmensanwendungen eingesetzt werden zu können (Bloor 2004; Gilfillan 2005; Conrad 2004; Vaas 2005*b*; Richter 2004; Strandell 2003). Zur besseren Übersichtlichkeit werden die Funktionalitäten vier Kategorien zugeordnet:

1. Datenintegrität: ACID-Transaktionen, Savepoints, zweiphasiges Commit, Locking auf Zeilenebene oder Multiversion Concurrency Control (MVCC)
2. Datenorganisation: Komplexe Datenbank-Objekte wie Sichten, Schemata und temporäre Tabellen
3. Datenzugriff:
 - Umfangreiche SQL-Features: Vererbung, nutzerdefinierte Datentypen, Operatoren und Funktionen, Sub-Selects, komplexe Abfragen, fortschrittliche Indexierungs-Strategien und Unterstützung von verteilten Datenbankabfragen
 - Möglichkeiten der Integration von Geschäftslogik in die Datenbank: Triggers, Events, Sequenzen, Stored Procedures
 - Eine große Auswahl an Schnittstellen, Treibern und APIs
4. Betrieb:
 - Performanz und Skalierbarkeit: Multi-Threading, Multi-Prozessor-Unterstützung, Caching, Partitionierung, Tablespaces, Clustering, Parallelisierung von Abfragen

- (Hoch-)Verfügbarkeit: Replikation, umfangreiche unterbrechungsfreie Datensicherungs- und Wiederherstellungs-Mechanismen
- Sicherheit: Verschlüsselung, Gruppen- und Rollenkonzept, feingranulare Rechteverwaltung und Logging

Die Tabelle 2 bietet einen Vergleich der Features der fünf Open-Source-Datenbankmanagementsysteme.³⁴

Die Fähigkeiten der einzelnen Datenbanken werden im Folgenden kurz diskutiert. Dabei muss beachtet werden, dass es wesentlich einfacher ist, SQL-Features und Standard-Konformität zu vergleichen oder die verfügbaren Stored-Procedure-Sprachen aufzuzählen, als Features aus den Bereichen Hochverfügbarkeit, Sicherheit oder Skalierbarkeit zu vergleichen. Diese sind nicht einfach abzählbar und es gibt im Allgemeinen auch keinen Standard, dessen Einhaltung man überprüfen kann. Implementationen desselben Features können sich grundlegend unterscheiden und oft steckt der Teufel im Detail. Auf jeden Fall aber können die jeweiligen Funktionalitäten benannt und klassifiziert werden.³⁵

4.2 Open-Source-Datenbanken: Features

Firebird

Firebird bringt zwar einen soliden Grundstock an Funktionalitäten mit, jedoch nur wenige anspruchsvollere Features.³⁶ Es bietet immerhin zweiphasiges Commit (aber ohne Unterstützung des XA-Standards) und die Multi-Generational-Architecture (ein Locking-Mechanismus, welcher der Multi Versioning Concurrency Control (MVCC) entspricht).

Hingegen fehlen Firebird die folgenden Funktionen, die die meisten anderen Datenbanken bieten:

- Geographischer/räumlicher Datentyp
- Schemas
- Temporären Tabellen
- Volltext-Suche
- Indexierungsalgorithmen außer B-Tree
- Replikation
- Load Balancing

³⁴ Eine detailliertere Version dieser Feature-Matrix sowie Links zu Quellen, die zur Bestückung der Matrix dienen, finden sich unter <http://www.osdbmigration.org:8080/osdb/osdb-features>.

³⁵ Erklärungen der einzelnen Konzepte, Akronyme und Techniken sind im Glossar am Ende dieses Buches zu finden.

³⁶ <http://www.firebirdsql.org/index.php?op=doc>,
http://www.ibphoenix.com/main.nfs?a=ibphoenix&page=ibp_download_15

Freie Datenbanken im Unternehmenseinsatz

	Firebird	Ingres	MaxDB	MySQL	PostgreSQL
Version	1.5	3.0	7.6	5.0	8.1
<i>Datenintegrität</i>					
ACID-Transaktionen	Ja	Ja	Ja	Ja *	Ja
2-phases Commit	Ja	Ja	Nein	Ja *	Ja
Fremdschlüssel	Ja	Ja	Ja	Ja *	Ja
CHECK-Bedingung	Ja	Nein	Ja	Nein	Ja
Savepoints	Ja	Ja	Ja	Ja *	Ja
Locking	MVCC	zeilenweise	zeilenweise, MVCC ab 7.7	MVCC und zeilenweise *	MVCC und zeilenweise
<i>Datenbankobjekte</i>					
Schema	Nein	Ja	Ja	Ja	Ja
Temporäre Tabellen	Nein	Ja	Ja	Ja	Ja
Stored Procedures	Ja	Ja	Ja	Ja	Ja
Trigger	Before/After	After	After	Before/After	Before/After
View	Ja	Ja	Ja	Ja	Ja
Materialized View	Nein	Nein	Nein	Nein	Nein
Updatable View	Ja	Nein	Ja	Ja	Ja
Expression Index	Nein	Nein	Nein	Nein	Ja
Partial Index	Nein	Nein	Nein	Nein	Ja
Bitmap Index	Nein	Nein	Nein	Nein	Ja
Volltext-Index	Nein	Nein	Nein	Ja (MyISAM)	Ja (TSearch2)
<i>SQL, Datentypen</i>					
Standard	92, 99	92, 99	92	92, 99	92, 99, 03
Nutzerdef. Typen	Nein	Ja	Nein	Nein	Ja
Nutzerdef. Funktionen	Ja	Ja	Ja	Ja	Ja
GIS	Nein	Ja	Nein	Ja	Ja
Boolean	Nein	Nein	Ja	Nein	Ja
Sub-Select	Ja	Ja	Ja	Ja	Ja
Full Outer Join	Ja	Ja	Ja	Nein	Ja
<i>Betrieb</i>					
Multi-Threading	Ja (Super)	Ja (Bs)	Ja (Db)	Ja (Bs)	Nein
Multi-Processing	Ja (Classic)	Ja	Ja	Ja	Ja
Abfrage-Parallelisierung	Nein	Ja	Ja	Ja (Cluster)	Nein
Replikation	Ja	Ja	Ja	Ja	Ja (Slony)
Multimaster Repl.	Ja	Ja	Nein	Nein	Nein
Clustering	Nein	Ja	Ja	Ja	Nein
Load Balancing	Nein	Ja	Nein	Ja	Nein
Tablespaces	Nein	Nein	Nein	Ja	Ja
Partitionierung	Nein	Ja	Nein	Nein (v.5.1)	Ja (CE)
Point-In-Time Recovery	Nein	Ja	Ja	Ja *	Ja

(Bs = Betriebssystem, Db = Datenbank, CE = Constraint Exclusion, * = InnoDB)

Tabelle 2: Vergleich von Open-Source-Datenbanken: Feature Matrix

- Clustering
- Tabellen-Partitionierung
- SSL-Verschlüsselung der Kommunikation zwischen Client und Server

Parallele Verarbeitung wird nur eingeschränkt unterstützt. Je nachdem, welche Datenbankserver-Variante gewählt wird. Der „Super Server“ bietet Multi-Threading, aber keinen Multi-Prozessor-Support. Beim „Classic Server“ ist es genau umgekehrt.

Ingres

Als ältestes der fünf Open-Source-Datenbankmanagementsysteme verfügt Ingres selbstverständlich über ein großes Arsenal von Funktionalitäten (Computer Associates 2004*a,b,c,d*; Vanthienen 2005). Besonders herauszuheben sind dabei der volle XML-Support, die GIS-Funktionalität, nutzerdefinierte Datentypen, Multi-Master-Replikation und Tabellen-Partitionierung. Ingres unterstützt Multi-Threading genauso wie Multi-Processing und bietet außerdem Abfrage-Parallelisierung, d. h. die Bearbeitung eines SQL-Statements kann durch mehrere Prozessoren gleichzeitig erfolgen.

Nichtsdestotrotz fehlen so wichtige Features wie „BEFORE“-Trigger, Updatable Views, Volltext-Indexierung, MVCC (nur Locking auf Zeilen-Ebene), CHECK-Bedingungen, Tablespaces und SSL-Verschlüsselung.

MaxDB

Die Stärken von MaxDB liegen vor allem in der parallelen Verarbeitung: Neben Ingres ist es das einzige System, das sowohl Multi-Threading als auch Multi-Processing und außerdem die parallele Abfrage-Bearbeitung bietet.³⁷ Auch im Bereich von Backup und Recovery kann MaxDB punkten: Komplette Backups im laufenden Betrieb werden genauso unterstützt wie Point-in-Time Recovery und Savepoints. Daneben werden Replikation und Clustering unterstützt, jedoch fehlt die Möglichkeit, Lasten im Cluster zu verteilen: Load-Balancing funktioniert nur zwischen den Prozessoren einer Maschine. Die Liste der fehlenden Features wird noch länger: Man vermisst einen räumlichen Datentyp, „BEFORE“-Trigger (geplant für Version 7.7), sämtliche Indexierungs-Algorithmen außer B*-Tree, zweiphasiges Commit, MVCC (geplant für Version 7.7), nutzerdefinierte Datentypen, kaskadierendes UPDATE, Tablespaces und Tabellen-Partitionierung.

Der Wechsel von Version 7.5 auf 7.6 brachte einige interessante neue Features, zum Beispiel bietet MaxDB nun Replikation per „Synchronisation Manager“ sowie nutzerdefinierte Funktionen.³⁸

³⁷ <http://dev.mysql.com/doc/maxdb>

³⁸ <http://dev.mysql.com/doc/maxdb/features76.html>

MySQL

Mit der Version 5.0 hat sich MySQL zu einer Datenbank gemauert, die auch für komplexe Unternehmens-Anwendungen einsetzbar ist.³⁹ Die neue Version bietet folgende Neuerungen:⁴⁰

- Stored Procedures (nach SQL:2003 Standard)
- Triggers
- (Updatable) Views
- Serverseitige Cursors
- Unterstützung des XA-Standards und zweiphasigen Commits
- Nutzer-definierte Funktionen
- Abfrage der Datenbank-Metadaten über das INFORMATION_SCHEMA
- „Strikter“ SQL-Modus⁴¹

Spezifisch für MySQL ist, dass es elf verschiedene Tabellentypen zur Auswahl gibt: MyISAM, MERGE, MEMORY (HEAP), BDB (BerkeleyDB), EXAMPLE, FEDERATED, ARCHIVE, CSV, BLACKHOLE, InnoDB und MySQL Cluster (NDB).⁴² Eine Datenbank kann Tabellen verschiedenen Typs enthalten. Die Wahl eines Typs ist abhängig von dem Einsatzzweck der Tabelle. Jeder Tabellentyp hat bestimmte Charakteristika, zum Beispiel:

- Nur der InnoDB-Typ bietet Fremdschlüssel, MVCC, Savepoints, Tablespace und XA-konforme verteilte Transaktionen mit zweiphasigem Commit.⁴³
- Nur InnoDB und Berkeley DB bieten ACID-Transaktionen.
- Nur InnoDB und NDB bieten Locking auf Zeilenebene (BDB: Locking auf Seiten-Ebene).
- Nur MyISAM-Tabellen unterstützen Volltext- und R-Tree-Indexierung.
- Für Clustering und Load Balancing verwendet man NDB-Cluster-Tabellen. Diese bieten auch die parallele Bearbeitung von Abfragen, von mehreren Cluster-Knoten gleichzeitig.

39 <http://dev.mysql.com/doc>

40 Für eine vollständige Liste aller Neuerungen in Version 5.0 siehe z. B. <http://dev.mysql.com/doc/refman/5.0/en/mysql-5-0-nutshell.html>

41 Normalerweise akzeptiert MySQL ungültige Werte, z. B. 31.02.2005 in einem Datumsfeld, Text in einem Zahlenfeld, und gibt nur eine Warnung aus. „strict“ unterbindet das Einfügen falscher Daten und erzeugt eine Fehlermeldung.

42 <http://dev.mysql.com/doc/mysql/en/storage-engines.html>

43 Dabei ist zu bemerken, dass MySQL-Fremdschlüssel nicht aufschiebbar („deferrable“) sind.

Neben den genannten Features unterstützt MySQL Replikation (Single-Master, Multi-Slaves), Multi-Threading und Multi-Prozessor-Support.

Nur eingeschränkt zur Verfügung stehen Sequenzen (eingeschränkte Implementierung über AUTO_INCREMENT) und Constraints (MySQL bietet nur UNIQUE und NOT NULL). Partitionierung (geplant für Version 5.1) und Rollen- oder Gruppenkonzepte für das Rechteemanagement fehlen vollständig.

PostgreSQL

PostgreSQL zeichnet sich durch eine enorme Funktionsvielfalt aus.⁴⁴ Zu den anspruchsvolleren Features gehören MVCC, CHECK Constraints, nutzerdefinierte Funktionen, Datentypen und Operatoren sowie räumliche, boole'sche und Netzwerkadress-Datentypen (IPv4, IPv6, MAC). Als einzige der Open-Source-Datenbanken stellt PostgreSQL die spezialisierten Indexierungs-Algorithmen Expression Index, Bitmap Index und Partial Index zur Verfügung. Daneben gibt es natürlich auch R-Tree, Heap- und Volltext-Indexierung. PostgreSQL bietet anspruchsvolle Replikation mittels der Slony-Software. Zweiphasiges Commit für verteilte Transaktionen steht ebenfalls zur Verfügung, jedoch bisher ohne Unterstützung des XA-Standards.

Schwächen zeigt PostgreSQL außerdem beim Thema Parallelisierung. Zwar stellt es Multi-Prozessor-Unterstützung zur Verfügung, jedoch kein Multi-Threading. Ebenfalls muss man auf eingebaute Clustering- und Load-Balancing-Lösungen verzichten.⁴⁵

PostgreSQL erlaubt, Stored Procedures in fast jeder Programmiersprache zu schreiben. Neben der Standard-Stored-Procedure-Sprache PL/pgSQL stehen C, Java, PL/perl, PL/PHP, PL/python, PL/Ruby, PL/sh, PL/TCL und PL/R zur Verfügung.

Im letzten Jahr veröffentlichte die PostgreSQL-Community mit Version 8.0 und 8.1 zwei Updates, die sich vor allem auf die Einführung von Features für große Unternehmens-Anwendungen konzentrierten: Point-in-Time Recovery, Savepoints, Tablespace (Version 8.0), Bitmap Scan, zweiphasiges Commit, Rollenkonzept für die Rechteverwaltung und Tabellen-Partitionierung (Version 8.1). Außerdem läuft PostgreSQL seit 8.0 auch nativ auf Windows-Plattformen.

4.3 Open-Source-Datenbanken für das Data Warehouse

Die oben diskutierten Features sind vor allem in der klassischen Datenverarbeitung interessant, dem „Online Transactional Processing“ (OLTP). Mit immer besseren Möglichkeiten zur Datenerfassung und -speicherung steigt jedoch auch der Bedarf an Lösungen zur Analyse dieser Daten („Online Analytical Processing“, OLAP). Dies geschieht mittels so genannter „Business Intelligence“-Software, die auf Daten in

⁴⁴ <http://www.postgresql.org/about>, <http://www.postgresql.org/docs/current/interactive/index.html>

⁴⁵ Es gibt aber an PostgreSQL angegliederte Projekte, die die fehlenden Tools entwickeln und zur Verfügung stellen. Diese finden sich auf PgFoundry (<http://pgfoundry.org>) und GBorg (<http://gborg.postgresql.org>).

einem Data Warehouse zugreift. Dabei handelt es sich um eine Datenbank, deren Schema speziell für diesen Einsatz optimiert wird.

Eine ausführliche Auseinandersetzung mit dem Thema Open-Source-Business-Intelligence und Data Warehousing findet sich bei Horstmann (2005b). Hier sollen nur kurz ein paar Hinweise auf die Fähigkeiten der besprochenen fünf Open-Source-Datenbanken im Data-Warehouse-Bereich gegeben werden.

Es gibt keine bekannte Fallstudie für den Einsatz von MaxDB oder Firebird im Rahmen eines Data Warehouses. Dasselbe gilt für Ingres, jedoch wird diese Datenbank im Rahmen der DATAlegro Data Warehouse Appliance eingesetzt (Howard 2005).

MySQL hingegen ist in dem Markt präsent. So nutzt beispielsweise der O'Reilly-Verlag das System im Rahmen einer selbstentwickelten Data-Warehouse-Lösung.⁴⁶ Die Business-Intelligence-Anbieter *Business Objects* und *Hyperion* (Essbase) unterstützen den Einsatz von MySQL mit ihren Produkten.⁴⁷

Besonders interessant ist jedoch, dass im letzten Jahr komplette „Open Source Business Intelligence Software Stacks“ entstanden sind, also Projekte, die mehrere Open-Source-Anwendungen zusammenfassen, um komplette Business-Intelligence-Lösungen anzubieten. Eines dieser Projekte ist Bizgres. Es enthält die Bizgres-Datenbank (ein PostgreSQL, das um OLAP-Funktionalitäten erweitert wurde), das Bizgres Toolkit (KETL für den Extraction-Transformation-Load-Prozess, Mondrian/JPIpivot für multidimensionale Analyse, JasperReports für Reporting und OpenI als Framework) und Bizgres Clickstream, eine komplette Beispielapplikation zur Analyse von Log-Daten.⁴⁸

4.4 And the Winner is. . . oder: Für jede Aufgabe das richtige Werkzeug

Es gelingt nicht, im Rahmen eines allgemeinen Vergleichs der Datenbanken eine eindeutige „Siegerin“ zu benennen. Alle fünf vorgestellten Open-Source-Datenbanken bieten einen Funktionsumfang, der ihren Einsatz in großen Unternehmens-Anwendungen rechtfertigt. Dabei hat jedoch jedes System seine Stärken und Schwächen, und welche Datenbank die richtige für einen konkreten Anwendungsfall ist, hängt in erster Linie von den Anforderungen des jeweiligen Szenarios ab. Wie immer gilt auch hier: Für jede Aufgabe das richtige Werkzeug.

5 Zusammenfassung

Die Welt der Open-Source-Datenbankserver ist in Bewegung. MySQL und PostgreSQL haben in den letzten Versionen Funktionserweiterungen erhalten, welche den Einsatz in großen Unternehmens-Anwendungen ermöglichen. Damit verwischen sich

46 http://www.mysqluc.com/presentations/mysql05/magoulas_roger.pdf

47 http://businessobjects.com/news/press/press2005/20050418_mysql_part.asp,
http://dev.hyperion.com/products/hub/hub_721_matrix.pdf

48 http://pgfoundry.org/docman/view.php/1000107/106/oss_bi_opendbcon2005_exp.pdf

auch die Unterschiede zwischen diesen beiden Datenbanken: Lange hielten sich die Vorurteile, MySQL sei zwar schnell, aber mangle an Funktionalitäten, während PostgreSQL über anspruchsvolle Features verfüge, aber langsam sei. Nun hat MySQL mit der Version 5.0 im Bereich der anspruchsvollen Features entscheidend aufgeholt, während PostgreSQL schon lange wesentliche Performanz-Gewinne zu verzeichnen hat.

Zu diesen beiden „klassischen“ Open-Source-Datenbanken gesellten sich in den letzten Jahren drei weitere: Firebird, Ingres und MaxDB. Alle verfügen über eine lange Geschichte im kommerziellen Bereich und versuchen sich nun mit wechselndem Erfolg daran, in die Open-Source-Softwareentwicklung und -Vermarktung einzusteigen.

So ist es *Firebird* gelungen, sich als Open-Source-Projekt eigenständig weiterzuentwickeln, nachdem Borland den InterBase-Code nach einem kurzen Ausflug in die Open-Source-Welt wieder schloss. Die Community wächst, die Datenbank verfügt über eine loyale Nutzerbasis und die Weiterentwicklung des Codes wird vorangetrieben. Als Schwachstellen sind derzeit noch die geringe Dokumentation zu nennen sowie die unzureichenden Fähigkeiten des Systems in Bezug auf Hochverfügbarkeit und Skalierbarkeit. Aber das Projekt hat Schwung und eine klare Roadmap, so dass die kommenden Versionen diese Probleme auf jeden Fall angehen werden.

Ingres lässt kaum Wünsche offen, was Funktionen, Skalierbarkeit und Stabilität angeht. Leider wurde die Software lange Jahre vom Anbieter Computer Associates (CA) vernachlässigt, kaum weiterentwickelt und schlecht vermarktet. Der Verkauf an die Investoren von Garnett & Helfrich Capital sowie der Versuch, seit letztem Jahr eine Open-Source-Community um das System aufzubauen, könnten dies ändern und Ingres wieder zu einer ernst zu nehmenden Größe im Datenbank-Markt machen.⁴⁹

SAP DB ist zwar schon seit 2000 Open Source, zog aber dennoch wenig Aufmerksamkeit auf sich. Erst als MySQL 2003 begann, das System als *MaxDB* anzubieten und zu vermarkten, wuchs das öffentliche Interesse. Die Weiterentwicklung der Datenbank geschieht jedoch weiterhin ausschließlich innerhalb der SAP AG, die nur geringes Interesse für eine Anbindung an die Open-Source-Community zeigt. Da auch keine öffentliche Roadmap vorliegt, ist unklar, ob und wann die fehlenden Funktionalitäten angegangen werden.

MySQL verbindet eine aktive Community und eine riesige Nutzer-Basis mit der Unterstützung und dem Marketing eines kommerziellen Unternehmens. Während der bisherige Erfolg sich hauptsächlich aus dem Einsatz in Web-Anwendungen speiste, bringt Version 5 Funktionalitäten mit, die MySQL auch für anspruchsvollere Szenarien

⁴⁹ Speziell ist dabei die „Ingres Million Dollar Challenge“ zu erwähnen: Um Open-Source-Entwickler auf das freie Datenbanksystem aufmerksam zu machen, schrieb Ingres im Sommer 2004 einen Preis in Höhe von einer Million USD für die Programmierung von Software aus, welche die Migration von verschiedenen Closed-Source-Datenbanken sowie MySQL hin zu Ingres unterstützen soll: <http://www.3.ca.com/Press/PressRelease.aspx?CID=61619>, <http://www.3.ca.com/Press/PressRelease.aspx?CID=69484>.

qualifiziert. Unklar ist bisher, welche Auswirkungen sich aus dem Kauf von InnoBase Oy (und damit des InnoDB-Tabellentyps) durch Oracle ergeben.

Die Herkunft aus dem universitären Bereich ist *PostgreSQL* anzusehen. Das Datenbankmanagementsystem verfügt über einen enormen Funktionsumfang und legt großen Wert auf Standardkonformität. Der Entwicklungsprozess vollzieht sich ausschließlich im Rahmen der Open-Source-Community. Nichtsdestotrotz gibt es eine Reihe von Unternehmen, die professionellen kommerziellen Support für PostgreSQL anbieten. Auch hier fehlen einige Features im Bereich der Skalierbarkeit und Hochverfügbarkeit – die Roadmap weist diesen aber hohe Priorität zu. Daneben ist zu bemerken, dass PostgreSQL im Rahmen der Bizgres-Distribution die einzige Open-Source-Datenbank ist, die einen *Business Intelligence Stack* für den Einsatz im Data-Warehouse-Bereich anbietet.

Für die nächsten Jahre ist bei MySQL und PostgreSQL eine Konsolidierung auf hohem Niveau zu erwarten, während die Entwicklung der Open-Source-Newcomer Ingres, Firebird und MaxDB spannend bleibt. Die kommenden Themen im Datenbank-Bereich sind OLAP-Fähigkeiten, XML-Unterstützung und die performante Verarbeitung sehr großer Datenmengen (*Very Large Databases*). Die Zukunft der Open-Source-Datenbanken wird davon abhängen, inwiefern sie innovative Lösungen in diesen Bereichen anbieten können.

Literatur

- Arvin, T. (2005), 'Comparison of different SQL implementations'.
<http://troels.arvin.dk/db/rdbms/> [27. Jan 2006].
- Bloor, R. (2004), Ingres and Open Source: The Relaunching of an Enterprise RDBMS, Studie für Computer Associates, Baroudi Bloor, Arlington, MA, USA.
http://www3.ca.com/Files/IndustryAnalystReports/bb_ingres.pdf [22. Jan 2006].
- Computer Associates (2004a), 'Ingres r3 Data Sheet'.
<http://opensource.ca.com/projects/ingres/documents/product/ingresr3/A001811E.pdf> [22. Jan 2006].
- Computer Associates (2004b), *Ingres r3 Database Administrator Guide*, Computer Associates.
<http://opensource.ca.com/projects/ingres/documents/product/ingresr3/A001841E.pdf> [22. Jan 2006].
- Computer Associates (2004c), 'Ingres r3 Release Summary'.
<http://opensource.ca.com/projects/ingres/documents/product/ingresr3/A001811E.pdf> [22. Jan 2006].
- Computer Associates (2004d), *Ingres r3 SQL Reference Guide*, Computer Associates.
<http://opensource.ca.com/projects/ingres/documents/product/ingresr3/A001851E.pdf> [22. Jan 2006].
- Conrad, T. (2004), 'PostgreSQL vs. MySQL vs. Commercial Databases: It's All About What You Need', *DevX*. <http://www.devx.com/dbzone/Article/20743> [22. Jan 2006].

- Delio, M. (2004), 'Linux: Fewer Bugs Than Rivals', *Wired News*.
<http://www.wired.com/news/linux/0,1411,66022,00.html> [27. Jan 2006].
- Evans Data Corp. (2005), 'Open Source Databases Currently Used'.
http://www.evansdata.com/n2/surveys/database/2005_1/db_05_1_xmp1.shtml
[22. Jan 2006].
- Fallmann, D., Fallmann, H., Pramböck, A., Reiterer, H., Schumacher, M., Steinmaurer, T. und Wagner, R. (2005), Comparison of the Enterprise Functionalities of Open Source Database Management Systems, Forschungsarbeit, Fabalabs Software GmbH, Honauerst. 4, 4020 Linz, Österreich.
<http://www.fabalabs.org/research/papers/FabalabsResearchPaper-OSDBMS-Eval.pdf>
[27. Jan 2006].
- Farooqi, M. (2004), 'Introduction to the Firebird Database', *Linux Journal*.
<http://www.linuxjournal.com/article/7010> [27. Jan 2006].
- Flannery, R. (2005), 'The 6 Biggest Questions About the Future of Informix'.
http://www.iuug.org/news/articles/Flannery_Weed_16jul05.html [27. Jan 2006].
- Gilfillian, I. (2005), 'Open Source Databases: A brief look at the Berkeley DB, Derby, Firebird, Ingres, MySQL and PostgreSQL DBMS', *Database Journal*.
<http://www.databasejournal.com/features/mysql/article.php/3486596> [22. Jan 2006].
- Horstmann, J. (2005a), Migration to Open Source Databases, Diplomarbeit, Technische Universität Berlin, Berlin. http://www.osdbmigration.org/misc/migrating_OSDB_jh.pdf
[21. Jan 2006].
- Horstmann, J. (2005b), 'Open Source Data Warehouse'. Ressourcen zur Diplomarbeit *Migration to Open Source Databases*,
http://www.osdbmigration.org/index.php/Open_Source_Data_Warehouse [27. Jan 2006].
- Howard, P. (2005), 'Ingres and Open Source – a success story'.
http://www.channelregister.co.uk/2005/03/24/ingres_and_open_source/ [27. Jan 2006].
- Lemos, R. (2004), 'Linux 'better than proprietary software'', *CNET News.com*.
<http://news.zdnet.co.uk/software/linuxunix/0,39020390,39181043,00.htm> [27. Jan 2006].
- Lemos, R. (2005), 'MySQL gets gold star on bug test', *CNET News.com*.
<http://news.zdnet.co.uk/software/applications/0,39020384,39186956,00.htm>
[27. Jan 2006].
- LinuxQuestions (2004), 'Database Of The Year'. Online-Umfrage,
<http://www.linuxquestions.org/questions/showthread.php?s=&threadid=116360>
[22. Jan 2006].
- Litchfield, D., Anley, C., Heasman, J. und Grindlay, B. (2005), *The Database Hacker's Handbook*, Wiley, Indianapolis, USA.
- Loftus, J. (2005), 'Open Source Ingres Out-hyped and Underdeveloped into Obscurity'. http://searchopensource.techtarget.com/originalContent/0,289142,sid39_gci1144401,00.html
[27. Jan 2006].

- Manhart, K. (2004), 'Open-Source-Datenbanken für den professionellen Einsatz', *tecChannel*.
<http://www.tecchannel.de/entwicklung/grundlagen/402121/> [27. Jan 2006].
- MySQL (2003), A Guide to Lower Database TCO – How the Open Source Database MySQL Reduces Costs by as Much as 90 %, techn. Bericht, MySQL AB.
http://cynergyssoftware.com/collateral/mysql_tco_whitepaper.pdf [21. Jan 2006].
- Newsforge (2004), 'Favorite Open Source Database'. Online-Umfrage,
<http://www.newsforge.com/pollBooth.pl?qid=54&aid=-1> [22. Jan 2006].
- O'Neill, D. (2003), 'Determining Return on Investment Using Software', *CrossTalk - The Journal of Defense Software Engineering*.
<http://www.stsc.hill.af.mil/crosstalk/2003/03/ONeill.html> [27. Jan 2006].
- O'Reilly (2004), 'Preferred Database'. Online-Umfrage,
<http://www.oreillynet.com/pub/pq/46> [22. Jan 2006].
- Raymond, E. S. (2000), 'The Cathedral and the Bazaar'.
<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/> [27. Jan 2006].
- Richter, P. N. (2004), 'Firebird targets the enterprise database', *Newsforge*.
<http://www.newsforge.com/article.pl?sid=04/11/29/1917247> [22. Jan 2006].
- Strandell, T. (2003), Open Source Database Systems: Systems Study, Performance And Scalability, Master's thesis, University of Helsinki, Helsinki, Finnland.
http://www.cs.helsinki.fi/u/tpstrand/thesis/Toni_Strandell_Masters_Thesis.pdf
[22. Jan 2006].
- Vaas, L. (2005a), 'CA Pushes Ingres Out of the Nest', *eWeek*.
<http://www.eweek.com/article2/0,1759,1883069,00.asp> [27. Jan 2006].
- Vaas, L. (2005b), 'Two Open-Source Databases Add Enterprise Appeal', *eWeek*.
<http://www.eweek.com/article2/0,1759,1748970,00.asp> [22. Jan 2006].
- Vanthenien, M. (2005), 'Ingres r3'. Präsentationsfolien,
<http://www.ca.com/be/english/past-events/lunch-s4/050317-open-source-databases.pdf>
[27. Jan 2006].
- Wheeler, D. A. (2005), 'Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!'. http://www.dwheeler.com/oss_fs_why.html
[21. Jan 2006].