

# HANSER



## Vorwort

Lutz Fröhlich

PostgreSQL 9

Praxisbuch für Administratoren und Entwickler

ISBN (Buch): 978-3-446-42239-1

ISBN (E-Book): 978-3-446-42932-1

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-42239-1>

sowie im Buchhandel.

## ■ 1.1 Einführung und Motivation

*PostgreSQL* ist aktuell die weltweit am stärksten wachsende Datenbank. Immer mehr Unternehmen sind davon überzeugt, dass Open-Source-Produkte in vielen Bereichen gegenüber kommerziellen Produkten klar im Vorteil liegen. Viele Open-Source-Produkte haben bewiesen, dass eine starke Entwickler-Community in der Lage ist, Produkte dynamischer und mit mehr Qualität zu entwickeln und weiter auszubauen. Das Thema Support ist ebenfalls kein Problem. Es gibt genügend IT-Dienstleister, die professionellen Support für Open-Source-Produkte anbieten.

Die zunehmende Popularität von PostgreSQL ist auch darauf zurückzuführen, dass die andere große Open-Source-Datenbank *MySQL* nach der Übernahme durch die Firma Oracle ums Überleben kämpft. Es ist ein zunehmender Umstieg von MySQL auf PostgreSQL zu verzeichnen.

Ein weiterer Vorteil, den PostgreSQL bietet, ist der sehr hohe Kompatibilitätsgrad zur Oracle-Datenbank. Dadurch ist es möglich, mit überschaubarem Migrationsaufwand von Oracle nach PostgreSQL zu migrieren. Darüber hinaus können Oracle- und PostgreSQL-Datenbanken problemlos im Mischbetrieb eingesetzt werden.

Nicht zuletzt wird die zunehmende Verbreitung von PostgreSQL aber auch durch dessen Robustheit in der Transaktionsverarbeitung, hohe Zuverlässigkeit sowie sehr gute Performance unterstützt. Die Version 9 sticht durch neue Features wie eingebauter Replikation, *Hot Standby* und *Streaming Replication*, zusätzlichen Performance-Features, Spalten-Triggern sowie vielen anderen sehr nützlichen Erweiterungen hervor.

Dennoch ist PostgreSQL eine schlanke und sehr gut zu verwaltende Datenbank, die sich auf ihr Kerngeschäft, die Verwaltung von Datenbeständen, konzentriert. Sie ist in den vergangenen Jahren erwachsen geworden und wird heute in vielen Bereichen kommerziell eingesetzt. PostgreSQL läuft auf vielen Plattformen, wie z.B. Linux, MacOS X, Solaris oder Windows. Obwohl es sich um ein Open-Source-Produkt handelt, kann kommerzieller Support zu einem vernünftigen Preis hinzugekauft werden. Einem professionellen Einsatz steht damit nichts im Wege.

Das Buch beschäftigt sich detailliert mit den Themen Administration von und Entwicklung auf PostgreSQL 9. Darüber hinaus finden Sie viele Informationen zum Thema „Migration

von Oracle oder MySQL nach PostgreSQL“ sowie zur Planung und Implementierung von PostgreSQL-Datenbanken in eine bestehende IT-Landschaft. Das schließt den Datenaustausch mit anderen Datenbanksystemen, insbesondere Oracle, mit ein.

Trotz einer hohen Kompatibilitätsrate zu Oracle gibt es für die Migration einiges zu beachten. Im Buch finden Sie viele praktische Erfahrungen, die bei verschiedenen Migrativen gesammelt werden konnten. Für die Replikation zwischen PostgreSQL und Oracle gibt es unterschiedliche Lösungen, die mehr oder weniger stabil laufen und unterschiedliche Kosten verursachen.

Das Buch ist als Schnelleinstieg für IT-Profis geschrieben und setzt grundlegende Kenntnisse von relationalen Datenbanksystemen voraus. Auf eine grundlegende Erläuterung von gängigen Begriffen wird deshalb bewusst verzichtet, auch um den Umfang des Buches überschaubar zu halten. Dennoch finden Sie viele Beispiele und Praxistipps, die auch Einsteigern die Möglichkeit bieten, sich in das Produkt einzuarbeiten.

## ■ 1.2 Die Geschichte von PostgreSQL

PostgreSQL geht zurück auf das POSTGRES-Projekt, das an der University of California at Berkeley in den 80er-Jahren angesiedelt war. Die erste vorzeigbare Version erschien im Jahre 1987 als *Postgres* Version 1. Als Reaktion auf die ersten Kritiken wurde das noch heute in PostgreSQL vorhandene Rule-System entwickelt. Version 3 erschien im Jahre 1991 mit einer Weiterentwicklung der Abfrageeinheit. 1993 beendete die University of California das Projekt mit der Version 4.2, um die rasant wachsenden Support-Anforderungen nicht mehr tragen zu müssen.

Nach Hinzufügen eines SQL-Abfrage-Interpreters im Jahre 1995 wurde die Software unter dem Begriff *Postgres95* ins Web gestellt, mit dem Quellcode des originalen Berkeley-Postgres. Das Produkt war zu dieser Zeit komplett in ANSI C geschrieben. Durch Verbesserung in den Bereichen Wartbarkeit und Performance lief es schließlich bis zu 50 % schneller als das originale Berkeley-Postgres.

Die Entscheidung, die Jahreszahl aus dem Produktnamen zu entfernen, fiel im Jahre 1996. Damit wurde Postgres95 zu PostgreSQL, und es begann die ständige Weiterentwicklung von PostgreSQL als Open-Source-Produkt. Obwohl Letzteres über viele Jahre ein Schattendasein im Licht der großen kommerziellen Datenbanken, aber auch der durch den Internet-Boom schnell verbreiteten Open-Source-Datenbank MySQL führte, erfolgte seine konsequente Weiterentwicklung durch die Community.

Heute präsentiert sich PostgreSQL als ausgereift und stabil und erfüllt (fast) alle Anforderungen an ein modernes relationales Datenbanksystem. Für viele überraschend: Die Performance ist gleich oder nur unwesentlich schlechter im Vergleich zu so manchem kommerziellen Produkt. Eine positive Überraschung sind auch die neuen Replikations-Mechanismen der Version 9, bis hin zur Hot Standby-Datenbank.

## ■ 1.3 Verwendete Version

Das Buch bezieht sich auf die während des Schreibens vorliegende aktuelle Version 9.1.0. Schauen Sie nach weiteren Veröffentlichungen, insbesondere für neuere Versionen, auf der Webseite des Verlages oder der Autoren-Webseite. Alles rund um die PostgreSQL-Community finden Sie auf der offiziellen Webseite <http://www.postgresql.org>.

## ■ 1.4 Konventionen

Begriffe in spitzen Klammern bezeichnen eine zu ersetzenende Variable (so ist zum Beispiel der Ausdruck <VERSION> durch die aktuelle Version 9.1.0 zu ersetzen).

Die meisten Darstellungen beziehen sich gleichermaßen auf UNIX- und Windows-Betriebssysteme. Die Darstellung der Umgebungsvariablen erfolgt im Wesentlichen im UNIX-Format, d.h. z.B. `$BIN` statt `%BIN%` für Windows. Sie können das Format einfach nach Windows übertragen. Das Gleiche gilt für das Trennzeichen der Pfade: `"/` unter Unix sowie `\` unter Windows.