

PostgreSQL im praktischen Einsatz

2. Brandenburger Linux Infotag 2005

Stefan Schumacher

<stefan@net-tex.de>, PGP Key <0xB3FBAE33>

<http://www.net-tex.de/>

\$Header: /home/daten/cvs/postgresql/folien.tex,v 1.11 2005/04/25 08:58:57 stefan Exp \$

Gliederung

- Vorstellung PostgreSQL
- Einführung Objektrelationale DBMS
- PostgreSQL Features
- Zugriffsmethoden und Frontends

PostgreSQL

- ORDBMS unter BSD Lizenz
- Forschungsprojekt in Berkeley
- weitgehend SQL92/SQL99 konform

ORDBMS

- Schnitt aus Objektorientierten und Relationalen DBMS
- verwendet semantische Datenmodelle
- Basistypen können benutzerdefiniert erweitert werden
- Objektreferenzen, Typenvererbung, Subtabellen
- Funktionen, Rules, Trigger

Features

- Mengenoperationen UNION, INTERSECT, EXCEPT
- Subselects, SELECT geschachtelt in WHERE
- Joins - Alle Typen aus SQL99
- Fremdschlüssel - Referentielle Integrität

- Views - Einschränkung der Benutzersicht (gespeichertes SELECT)
- Rules - modifizieren Abfragen je Tabelle und Ereignis
- Trigger - Funktion bei *DML – Aktion*
- stored procedures - Funktionen im DB-Server

Rules & Trigger

- **Rules:** eigentlich: *query rewrite rule system*
- modifiziert Query entsprechend Regeln

```
CREATE VIEW myview AS SELECT * FROM mytab;    ≡  
CREATE TABLE myview (↗ mytab); CREATE RULE "_RETURN" AS ON SELECT  
TO myview DO INSTEAD SELECT * FROM mytab;
```

- **Trigger:** *Funktion*, die bei *DML – Aktion* aufgerufen wird
- FOR EACH ROW \rightsquigarrow row level
- FOR EACH STATEMENT \rightsquigarrow statement level

Funktionen

- stored procedures, Erweiterungen im Server
- interne Funktionen (sin(), dsqrt())
- in SQL geschriebene Funktionen
- in einer prozeduralen Sprache geschriebene Funktionen
- in C geschriebene Funktionen

Funktionen

Bsp. in PL/PGSQL:

```
CREATE FUNCTION hp2kw (FLOAT)
  RETURN FLOAT
  AS 'SELECT (\$1 * 0.736)'
  LANGUAGE 'sql'
CREATE;
```

```
SELECT ps, hp2kw(ps) FROM lkws;
```

Transaktionen

- gekapselter Block von SQL-Befehlen

Atomicity atomare Operationen entweder ganz oder gar nicht

Consistency vor und nach Transaktion konsistenter Zustand

Isolation konkurrierende Transaktion müssen isoliert werden

Durability nur konsistente Datenbankzustände

- *jede* Anfrage wird als Transaktion durchgeführt

Transaktionsfehler

- *dirty read*
Daten, die noch geändert werden, wurden gelesen
- *nonrepeatable read*
wiederholtes Lesen von Daten mit unterschiedlichen Werten
- *phantom read*
Ergebnismenge variiert

Transaktionsisolierung

- Implementiert: *Read Committed* & *Serializable*
- *Read Committed* verhindert *dirty read*
- *Serializable* verhindert alle Fehler
- emuliert (*keine* strikte mathematische) Serialisierung
- tötet konkurrierende DML-Transaktion \rightsquigarrow Wiederholung nötig
- SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
- explizites Locking möglich

MVCC

- Multi Version Concurrency Control
- Benutzer arbeitet auf geforktem Snapshot des Datenbestandes
- jede Spalte hat:
 - creation transaction ID
 - expiration transaction ID
- geben Status des Datenbestands wieder (veraltet/aktuell)
- UPDATE \rightsquigarrow neue Spalte mit veränderten IDs

Perl DBD/DBI

- DBI: API, unabhängig von der Datenbank
- DBD: Datenbank-Treiber, PostgreSQL-spezifisch stellt Zugriffsmethoden auf die DB zur Verfügung
- Apache: mod_perl lässt Perl ausführen
erlaubt keine persistenten Verbindungen
Apache::DBI für Apache oder
mod_proxy mit Round Robin

PHP & JDBC

- **PHP**
- Unterstützung für Pg enthalten
- ermöglicht Webbasierte Anwendungen mit Apache
- <http://www.pgsql.info/php/pgsql.php>
- **JDBC Treiber**
- Java 1.1 Core API
- <http://jdbc.postgresql.org/>

ODBC

- Verbindungsstandard in der MS Welt, auch unter Mac und Unix
- Einbindung in Windows NT/2000/XP als Datenquelle
- Anwendungsprogramme (MS-Office, OpenOffice.org, MS Access) können auf die Datenquelle zugreifen.
- Pg Treiber unter Windows installieren und an Pg-Server binden
- <http://gborg.postgresql.org/project/psqlodbc/>
- Perl DBD-ODBC existiert

pgcluster

- synchrones Multimaster Replikationssystem
- geeignet für Replikation oder Load Balancing
- HA durch abtrennen fehlerhafter Knoten
- <http://pgcluster.projects.postgresql.org/index.html>

pgcluster Load Balancer

