Eine Tour durch PostgreSQL-Contrib

Extend Your Database Server

Christoph Berg <christoph.berg@credativ.de>

credativ GmbH, Mönchengladbach

7. November 2013

CREATE EXTENSION

```
postgres=# CREATE EXTENSION <tab><tab>
```

fuzzystrmatch pg_trgm plsh cube autoinc chkpass lo btree_gin btree_gist test_parser plpython3u adminpack postgis_topology prefix tablefunc uuid-ossp plperl earthdistance unaccent plpythonu postgres_fdw hstore pgstattuple insert_username pgrowlocks tcn isn pg_stat_statements pg_buffercache xml2 postgis moddatetime plpgsql pgcrypto pltclu pltcl intarray refint citext pg_freespacemap tsearch2 ltree file_fdw seg sslinfo timetravel dict_int dblink plperlu pgmp plpython2u intagg pageinspect dict_xsyn

CREATE EXTENSION

- Neue Funktionen
- Datentypen
- Index-Methoden
- Procedural Languages (PL)
- Trigger
- Statistiken

credativ

CREATE EXTENSION

- PostgreSQL "contrib"
- Viele weitere im Internet (github, pgfoundry, pgxn, . . .)
- Debian/Ubuntu: apt.postgresql.org
- CREATE EXTENSION seit PostgreSQL 9.1
- Vorher \i /usr/share/postgresql/8.4/contrib/foobar.sql
- Vereinfachtes Handling, Backup

credativ

CREATE EXTENSION "uuid-ossp";

```
# select uuid_generate_v1();
           uuid_generate_v1
--------------------------------------
 6eb92d28-4634-11e3-ae7a-001f29960aed
```

CREATE EXTENSION pg_trgm;

- String wird in Trigramme zerlegt
  cat → " c", " ca", "cat", "at "
- Abstand zwischen Strings: <->

```
# select 'Oberhausen' <-> 'Essen';
 ?column?
 ----------
  0.866667
```

- CREATE INDEX ON test_trgm USING gist (t gist_trgm_ops);
  CREATE INDEX ON test_trgm USING gin (t gin_trgm_ops);
- "Meinten Sie . . . ?"

```
SELECT t, t <-> 'word' AS dist
  FROM suchbegriffe
    ORDER BY dist LIMIT 10;
```

CREATE EXTENSION plsh;

```
CREATE FUNCTION ls(text) RETURNS text AS '
#!/bin/sh
ls -l "$1"
' LANGUAGE plsh;

SELECT ls ('/etc/passwd');
                              ls
-----------------------------------------------------
 -rw-r--r-- 1 root root 2818 Feb  7  2013 /etc/passwd
```

CREATE EXTENSION cube;

- Multidimensional Cubes

```
# select '(5,7)'::cube <@ '[(3,3),(8,8)]' AS punkt_im_rech
 punkt_im_rechteck
-------------------
 t
```

- GiST-Support

CREATE EXTENSION earthdistance;

```
select earth_distance(ll_to_earth(51,6), ll_to_earth(52,10));
  earth_distance
------------------
 298660.368346433
```

CREATE EXTENSION postgis;

- Größte Extension, eigenes Open Source-Projekt

- Geo-Datentypen, OpenGIS, ISO SQL/MM
- "Welche Grundstücke liegen an dieser Straße?"
- Raster-Daten
- Routenplanung

credativ

CREATE EXTENSION pg_stat_statements;

- Liste der ausgeführten Queries
- Verbessert seit PostgreSQL 9.2
-
```
# SELECT query, calls, total_time, rows, 100.0 * shared_blks_hit /
                nullif(shared_blks_hit + shared_blks_read, 0) AS hit_percent
           FROM pg_stat_statements ORDER BY total_time DESC LIMIT 5;
-[ RECORD 1 ]---------------------------------------------------------
query       | UPDATE pgbench_branches SET bbalance = bbalance + ? WHERE bid =
calls       | 3000
total_time  | 9609.00100000002
rows        | 2836
hit_percent | 99.9778970000200936
-[ RECORD 2 ]---------------------------------------------------------
query       | UPDATE pgbench_tellers SET tbalance = tbalance + ? WHERE tid = ?
calls       | 3000
total_time  | 8015.156
rows        | 2990
hit_percent | 99.9731126579631345
-[ RECORD 3 ]---------------------------------------------------------
query       | copy pgbench_accounts from stdin
calls       | 1
total_time  | 310.624
rows        | 100000
hit_percent | 0.303951367781155015520
```

CREATE EXTENSION plpythonu;

- plpython2u, plpython3u
- "untrusted": CREATE FUNCTION nur für Superuser
- ```
  CREATE FUNCTION pymax (a integer, b integer)
    RETURNS integer
  AS $$
    if a > b:
      return a
    return b
  $$ LANGUAGE plpythonu;
  ```
  ```

CREATE EXTENSION adminpack;

- Support-Funktionen für pgAdmin3 ("Server-Instrumentierung")
- Files lesen/schreiben
- postgresql.conf, pg_hba.conf, Logfiles

CREATE EXTENSION prefix;

- Präfix-Matching mit Index-Support
- ```
  SELECT *
    FROM prefixes
   WHERE prefix @> '01805'
ORDER BY length(prefix) DESC
   LIMIT 1;
  ```
  ```

CREATE EXTENSION tablefunc;

- normal_rand, connectby
- crosstab

```
CREATE TABLE ct(id SERIAL, rowid TEXT, attribute TEXT, value TEXT);
INSERT INTO ct(rowid, attribute, value) VALUES('test1','att1','val1');
INSERT INTO ct(rowid, attribute, value) VALUES('test1','att2','val2');
INSERT INTO ct(rowid, attribute, value) VALUES('test1','att3','val3');
INSERT INTO ct(rowid, attribute, value) VALUES('test1','att4','val4');
INSERT INTO ct(rowid, attribute, value) VALUES('test2','att1','val5');
INSERT INTO ct(rowid, attribute, value) VALUES('test2','att2','val6');
INSERT INTO ct(rowid, attribute, value) VALUES('test2','att3','val7');
INSERT INTO ct(rowid, attribute, value) VALUES('test2','att4','val8');

SELECT *
FROM crosstab(
  'select rowid, attribute, value
   from ct
   order by 1,2')
AS ct(row_name text, a1 text, a2 text, a3 text, a4 text);
 row_name |  a1  |  a2  |  a3  |  a4
----------+------+------+------+------
 test1    | val1 | val2 | val3 | val4
 test2    | val5 | val6 | val7 | val8
```

CREATE EXTENSION plperl;

- Trusted, CREATE FUNCTION für alle erlaubt
- Keine Perl-Module erlaubt (use)
- ```
  create function match(text, text)
    returns boolean as $$
      ($a, $b) = @_;
      return $a =~ $b;
    $$ language plperl;
  ```
  ```

CREATE EXTENSION postgres_fdw;

```
create table t (t text);
insert into t values ('Hallo Welt');

create server loopback foreign data wrapper postgres_fdw
  options (port '5432');
create user mapping for current_user server loopback ;
create foreign table t2 (t text) server loopback
  options (table_name 't');

select * from t2;
     t
------------
 Hallo Welt
```

CREATE EXTENSION dblink;

- Zugriff auf andere PostgreSQL-Datenbanken
- Älteres Interface als postgres_fw
- ```
  CREATE VIEW myremote_pg_proc AS
    SELECT *
      FROM dblink('dbname=postgres', 'select proname, prosrc
      AS t1(proname name, prosrc text);

  SELECT * FROM myremote_pg_proc WHERE proname LIKE 'bytea%'
  ```


  ```

CREATE EXTENSION hstore;

- Perl-Style Hashes
- Semi-strukturierte Daten
- ```
  SELECT 'name => "Berg", vorname => "Christoph",
    firma => "credativ"'::hstore;
                                hstore
  ------------------------------------------------------------
   "name"=>"Berg", "firma"=>"credativ", "vorname"=>"Christop
  ```
- `SELECT h->'name';`
- Index-Support


  ```

CREATE EXTENSION plpgsql;

- PL/pgSQL: SQL + prozedurale Elemente
- Per Default installiert
- CREATE OR REPLACE FUNCTION get_all_foo() RETURNS SETOF foo AS

```
$BODY$
DECLARE
    r foo%rowtype;
BEGIN
    FOR r IN
        SELECT * FROM foo WHERE fooid > 0
    LOOP
        -- can do some processing here
        RETURN NEXT r; -- return current row of SELECT
    END LOOP;
    RETURN;
END
$BODY$
LANGUAGE plpgsql;

SELECT * FROM get_all_foo();
```

CREATE EXTENSION autoinc;

```
# CREATE SEQUENCE next_id;
# CREATE TABLE ids (id int4, idesc text);
# CREATE TRIGGER ids_nextid
    BEFORE INSERT OR UPDATE ON ids
    FOR EACH ROW
    EXECUTE PROCEDURE autoinc (id, next_id);
# INSERT INTO ids VALUES (0, 'first');
# INSERT INTO ids VALUES (null, 'second');
# INSERT INTO ids(idesc) VALUES ('third');
# SELECT * FROM ids ;
 id | idesc
----+--------
  1 | first
  2 | second
  3 | third
```

- Ähnlich "serial"

CREATE EXTENSION chkpass;

- Speichert crypt()-Passwörter
- Eingabe im Klartext
- Ausgabe verschlüsselt als Hash
- Vergleicht als Hash
- ```
  test=# create table test (p chkpass);
  test=# insert into test values ('hello');
  test=# select * from test where p = 'hello';
         p
  ---------------
   :fN6iG6x8Mv22M
  ```
  ```

CREATE EXTENSION fuzzystrmatch;

- Soundex: Ähnliche Namen, US-Zensus 1880

```
# CREATE TABLE s (nm text);
# INSERT INTO s VALUES ('john'), ('joan'), ('wobbly'), ('jack');
# SELECT * FROM s WHERE difference(s.nm, 'john') > 2;
  nm
------
 john
 joan
 jack
```

- Levenshtein: Abstand in Zahl der Änderungen am String

```
# SELECT levenshtein('GUMBO', 'GAMBOL');
 levenshtein
-------------
           2
```

- metaphone: Ähnlich Soundex

CREATE EXTENSION btree_gin;

- CREATE TABLE test (a int4);
- CREATE INDEX testidx ON test USING gin (a);
- Nützlich für mehrspaltige GIN-Indexe

CREATE EXTENSION btree_gist;

- Analog btree_gin
- Support für <-> (Abstand)
- Support für Exclusion Constraints mit <>

```
=> CREATE TABLE zoo (
  cage   INTEGER,
  animal TEXT,
  EXCLUDE USING gist (cage WITH =, animal WITH <>)
);
=> INSERT INTO zoo VALUES(123, 'zebra');
=> INSERT INTO zoo VALUES(123, 'zebra');
=> INSERT INTO zoo VALUES(123, 'lion');
ERROR:  conflicting key value violates exclusion
  constraint "zoo_cage_animal_excl"
DETAIL:  Key (cage, animal)=(123, lion) conflicts with
  existing key (cage, animal)=(123, zebra).
=> INSERT INTO zoo VALUES(124, 'lion');
```

credativ

CREATE EXTENSION pgstattuple;

- Liest ganze Tabelle oder Index
- `SELECT * FROM pgstattuple('pg_catalog.pg_proc');`

```
-[ RECORD 1 ]------+-------
table_len          | 458752
tuple_count        | 1470
tuple_len          | 438896
tuple_percent      | 95.67
dead_tuple_count   | 11
dead_tuple_len     | 3157
dead_tuple_percent | 0.69
free_space         | 8932
free_percent       | 1.95
```

- `SELECT * FROM pgstatindex('pg_cast_oid_index');`

CREATE EXTENSION unaccent;

- ```
  # SELECT unaccent('Hôtel');
   unaccent
  ----------
   Hotel
  ```

- Integration mit tsearch
  ```

CREATE EXTENSION lo;

- Mangement von BLOB-Referenzen über Trigger
- `CREATE TABLE image (title text, raster lo);`

```
CREATE TRIGGER t_raster BEFORE UPDATE OR DELETE ON image
    FOR EACH ROW EXECUTE PROCEDURE lo_manage(raster);
```

- Nützlich für JDBC- und ODBC-Anwendungen

credativ

CREATE EXTENSION insert_username;

```
CREATE TABLE username_test (
    name        text,
    username    text not null
);

CREATE TRIGGER insert_usernames
    BEFORE INSERT OR UPDATE ON username_test
    FOR EACH ROW
    EXECUTE PROCEDURE insert_username (username);

INSERT INTO username_test VALUES ('foobar');

SELECT * FROM username_test;
  name   | username
---------+----------
 foobar | cbe
```

CREATE EXTENSION isn;

```
# SELECT ean13('4220356483487');
      ean13
-----------------
 422-035648348-7
# SELECT isbn('978-0-393-04002-9');
     isbn
---------------
 0-393-04002-X
# SELECT isbn13('0901690546');
      isbn13
-------------------
 978-0-901690-54-8
# SELECT issn('1436-4522');
   issn
-----------
 1436-4522
```

CREATE EXTENSION citext;

- Case-insensitive text type
- Transparente Konvertierung groß/klein
- ```
  CREATE TABLE users (
      nick CITEXT PRIMARY KEY,
      pass TEXT   NOT NULL
  );

  INSERT INTO users VALUES ('larry', md5(random()::text));

  SELECT * FROM users WHERE nick = 'Larry';
  ```


  ```

CREATE EXTENSION pg_buffercache;

```
# SELECT c.relname, count(*) AS buffers
   FROM pg_buffercache b INNER JOIN pg_class c
   ON b.relfilenode = pg_relation_filenode(c.oid) AND
      b.reldatabase IN (0, (SELECT oid FROM pg_database
                            WHERE datname = current_database()))
   GROUP BY c.relname
   ORDER BY 2 DESC LIMIT 10;
             relname              | buffers
----------------------------------+---------
 ranges                           |     100
 prefixes                         |      91
 pg_depend                        |      84
 pg_proc                          |      80
 idx_prefix                       |      72
 pg_depend_reference_index        |      49
 pg_attribute                     |      49
 pg_depend_depender_index         |      37
 pg_proc_proname_args_nsp_index   |      35
 prefixes_pkey                    |      35
```

CREATE EXTENSION pgcrypto;

- Kryptographische Funktionen
- Hashes: md5, sha1, sha256, . . . (OpenSSL)
- hmac()
- crypt()
- OpenPGP
  - pgp_sym_encrypt(), pgp_sym_decrypt()
  - pgp_pub_encrypt(), pgp_pub_decrypt()
- gen_random_bytes()

CREATE EXTENSION pgmp;

- GNU Multi Precision library
- Große Integer, Primzahlen
- Rationale Zahlen (Brüche)

```
select '3/8'::mpq + '5/6'::mpq as fraction;
 fraction
----------
 29/24
```

CREATE EXTENSION tcn;

- Triggered Change Notification
- LISTEN/NOTIFY
- ```
  create trigger tcndata_tcn_trigger
     after insert or update or delete on tcndata
     for each row
     execute procedure triggered_change_notification();
  listen tcn;

  Asynchronous notification "tcn" with payload
     ""tcndata",I,"a"='2',"b"='2012-12-23'"
     received from server process with PID 22770.
  ```


  ```

CREATE EXTENSION pgrowlocks;

```
# select * from pgrowlocks('tbl');
 locked_row | locker | multi |  xids   |     modes       | pids
------------+--------+-------+---------+-----------------+------
 (0,1)      |  18227 | f     | {18227} | {"For Update"} | {0}
```

CREATE EXTENSION pg_freespacemap;

- Freier Platz in Tabellenblöcken
- ```
  # SELECT * FROM pg_freespace('foo');
   blkno | avail
  -------+-------
       0 |     0
       1 |     0
       2 |     0
       3 |    32
       4 |   704
       5 |   704
       6 |   704
       7 |  1216
  ```
- Selten notwendig
  ```

CREATE EXTENSION ltree;

This module implements a data type ltree for representing labels of data stored in a hierarchical tree-like structure. Extensive facilities for searching through label trees are provided.

```
ltreetest=> SELECT path FROM test WHERE path <@ 'Top.Science';
                  path
-------------------------------------
 Top.Science
 Top.Science.Astronomy
 Top.Science.Astronomy.Astrophysics
 Top.Science.Astronomy.Cosmology
```

CREATE EXTENSION pageinspect;

- Block-level debugging
- get_raw_page(), page_header(), heap_page_items()
- bt_metap(), bt_page_stats(), bt_page_items()
- fsm_page_contents

credativ

CREATE EXTENSION file_fdw;

The file_fdw module provides the foreign-data wrapper file_fdw, which can be used to access data files in the server's file system. Data files must be in a format that can be read by COPY FROM; see COPY for details. Access to such data files is currently read-only.

CREATE EXTENSION pltcl;

- pltcl, pltclu

credativ

CREATE EXTENSION seg;

This module implements a data type seg for representing line segments, or floating point intervals. seg can represent uncertainty in the interval endpoints, making it especially useful for representing laboratory measurements.

CREATE EXTENSION sslinfo;

The sslinfo module provides information about the SSL certificate that the current client provided when connecting to PostgreSQL. The module is useless (most functions will return NULL) if the current connection does not use SSL.

CREATE EXTENSION timetravel;

Long ago, PostgreSQL had a built-in time travel feature that kept the insert and delete times for each tuple. This can be emulated using these functions. To use these functions, you must add to a table two columns of abstime type to store the date when a tuple was inserted (start_date) and changed/deleted (stop_date):

```
CREATE TABLE mytab (
        ...             ...
        start_date      abstime,
        stop_date       abstime
        ...             ...
);
```

CREATE EXTENSION dict_int;

dict_int is an example of an add-on dictionary template for full-text search. The motivation for this example dictionary is to control the indexing of integers (signed and unsigned), allowing such numbers to be indexed while preventing excessive growth in the number of unique words, which greatly affects the performance of searching.

CREATE EXTENSION intarray;

The intarray module provides a number of useful functions and operators for manipulating null-free arrays of integers. There is also support for indexed searches using some of the operators.

CREATE EXTENSION intagg;

The intagg module provides an integer aggregator and an enumerator. intagg is now obsolete, because there are built-in functions that provide a superset of its capabilities. However, the module is still provided as a compatibility wrapper around the built-in functions.

CREATE EXTENSION dict_xsyn;

dict_xsyn (Extended Synonym Dictionary) is an example of an add-on dictionary template for full-text search. This dictionary type replaces words with groups of their synonyms, and so makes it possible to search for a word using any of its synonyms.

CREATE EXTENSION moddatetime;

moddatetime() is a trigger that stores the current time into a timestamp field. This can be useful for tracking the last modification time of a particular row within a table.

CREATE EXTENSION refint;

Functions for Implementing Referential Integrity

CREATE EXTENSION test_parser;

- test_parser is an example of a custom parser for full-text search. It doesn't do anything especially useful, but can serve as a starting point for developing your own parser.