



REPLICATION 9 MADE EASY

www.PostgreSQL.org

PostgreSQL

- Čistě komunitní projekt
- 15 let vývoje, 600 tis. Řádek kódu
- Vývoj zajišťuje cca 30 nadšenců (dnes již většinou na fulltime), + desítky dalších přispěvatelů a příznivců 230 (8.4), 275 (9.0)
- PostgreSQL se používá v finančnictví, telco, vědeckých institucích, NASA, v medicíně, v strojírenství, dopravě

Historie

- 6.2 – oprava vážných chyb
- 7.4 – eliminace výkonnostních problémů (2003)
- 8.0 – podpora MS Windows (2004)
- 8.4 – integrace enterprise funkcí (2009)
- 9.0 – integrace podpory replikace (2010)
- 9.1 – synchronní replikace, ... (2011)

Metodiky vývoje

- Jednoroční cyklus, zmražení seznamu patchů, zpracování přijatých patchů, testování, minimální formalizace procesu, důraz na kvalitu kódu, kvalitu dokumentace, kvalitu komentářů v CVS
- Ve verzích 8.3, 8.4 nastává nezvladatelný nárůst fronty přijatých patchů – požadavek na změnu procesu
- Éra commitfestů – kratší vývojové cykly, každý patch je recenzován – úspora času commiterů

Nástroje

- Klasické vývojové nástroje gcc, gdb, diff, patch
- CVS, dnes již Git
- Mailing lists + zpřístupněné archivy
- Wiki – mediawiki
- DocBook
- Jednoduché PHP evidence
- Testovací farma

PostgreSQL 8.4

- Nativní funkce Unnest a Array_agg
- EXECUTE USING a CASE v PL/pgSQL
- Rozšíření příkazu RAISE, RETURN QUERY
- Variadické funkce a defaultní hodnoty parametrů
- Locales per databáze
- Analytické funkce, CTE
- Visibility maps a VACUUM, semijoin, antijoin
- Paralelní import dumpu

Variadické funkce

```
CREATE OR REPLACE FUNCTION
  myleast(VARIADIC anyarray)
RETURNS anyelement AS $$
SELECT min(v)
  FROM unnest($1) g(v)
$$ LANGUAGE sql;
```


PostgreSQL 9.0

- Podpora asynchronní replikace
- Zjednodušená migrace
- Hromadné nastavování práv
- Anonymní bloky a pojmenované parametry
- Agregáční funkce s klauzulí ORDER BY

Anonymní bloky

```
DO $$  
BEGIN  
  FOR i IN 1..10  
  LOOP  
    RAISE NOTICE 'Hello %', i;  
  END LOOP;  
END;  
$$
```

Agregační funkce s klauzulí ORDER BY

```
SELECT string_agg(prijmeni, ','  
                  ORDER BY prijmeni)  
FROM zamestnanci  
GROUP BY oddeleni;
```

Pojmenovane implicitní parametry funkce

```
CREATE FUNCTION foo(a int = 3, b int = 0)  
RETURNS ...
```

```
SELECT foo();
```

```
SELECT foo(10,20);
```

```
SELECT foo(10, b := 30);
```

```
SELECT foo(10);
```

```
SELECT foo(b := -1, a := 20);
```

PostgreSQL 9.1

- Režim synchronní replikace
- SQL/MED
- COLLATION na úrovni sloupců
- Prává úroveň izolace SERIALIZABLE
- Datový typ JSON
- Příkaz MERGE
- Integrace s SE-Linuxem
- Funkce Median, Left, Right, Concat, Format
- Globální dočasné tabulky

Podpora COLLATES

```
test=# select * from jmena  
      order by jmeno collate "cs_CZ";
```

```
  jmeno  
-----  
Crha  
Doucha  
Gregor  
Hrozný  
Chromečka  
(5 rows)
```

Podpora COLLATES

```
test=# select * from jmena where jmeno >= 'Hrozný'  
                                     collate "cs_CZ";
```

```
 jmeno  
-----  
Hrozný  
Chromečka  
(2 rows)
```

```
test=# select * from jmena where jmeno >= 'Hrozný' collate "C";
```

```
 jmeno  
-----  
Hrozný  
(1 row)
```

Triggery INSTEAD OF

```
CREATE VIEW v AS SELECT * FROM t1, t2 WHERE a = b;
```

```
CREATE OR REPLACE FUNCTION trg_handler()  
RETURNS trigger AS $$  
BEGIN  
    INSERT INTO t1 VALUES(new.a);  
    INSERT INTO t2 VALUES(new.b);  
RETURN new;  
END;
```

```
CREATE TRIGGER foo_trg  
    INSTEAD OF INSERT ON v  
FOR EACH ROW EXECUTE PROCEDURE trg_handler();
```