

PostgreSQL 9.0 / Prolog

- Want to play along?
- PostgreSQL 9.0
 - `git clone git://git.postgresql.org/git/postgresql.git`
 - <http://www.postgresql.org/docs/current/static/anoncv.html>
 - `./configure; make; make install;`
- Pagila Sample Database
 - http://pgfoundry.org/frs/?group_id=1000150&release_id=998

No More Waiting

A Guide to PostgreSQL 9.0



OmniTI / Presentation / Robert Treat

Know More Waiting

A Guide to PostgreSQL 9.0



OmniTI / Presentation / Robert Treat

Know More Waiting

A Guide to Postgres 9.0



OmniTI / Presentation / Robert Treat

Postgres 9.0

- Performance
- Administration
- Development
- Procedures
- Replication



BUT FIRST!

Postgres 9.0 / Recap



Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)

Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)
- Commit Fests (July, September, November, January)

Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)
- Commit Fests (July, September, November, January)
- Alpha Release after each Commit Fest

Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)
- Commit Fests (July, September, November, January)
- Alpha Release after each Commit Fest
- Feature Freeze (February)

Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)
- Commit Fests (July, September, November, January)
- Alpha Release after each Commit Fest
- Feature Freeze (February)
- 9.0 ?!?

Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)
- Commit Fests (July, September, November, January)
- Alpha Release after each Commit Fest
- Feature Freeze (February)
- 9.0 ?!?
- Beta (April 29)

Postgres 9.0 / Recap

- 8.4 Released (2009-07-01)
- Commit Fests (July, September, November, January)
- Alpha Release after each Commit Fest
- Feature Freeze (February)
- 9.0 ?!?
- Beta (April 29)
- Release (May ? June ? July)

Postgres 9.0 / Stats



- New feature patches: 204

- New feature patches: 204
 - doesn't include direct commits

- New feature patches: 204
 - doesn't include direct commits
- Submitters: 84

- New feature patches: 204
 - doesn't include direct commits
- Submitters: 84
 - doesn't include committers

- New feature patches: 204
 - doesn't include direct commits
- Submitters: 84
 - doesn't include committers
- 1860 files changed

- New feature patches: 204
 - doesn't include direct commits
- Submitters: 84
 - doesn't include committers
- 1860 files changed
 - 150951 insertions

- New feature patches: 204
 - doesn't include direct commits
- Submitters: 84
 - doesn't include committers
- 1860 files changed
 - 150951 insertions
 - 82558 deletions

Postgres 9.0

- Performance
- Administration
- Development
- Procedures
- Replication

Postgres 9.0 / Perf / vacuum full



- VACUUM FULL now works like CLUSTER
 - The Old
 - move rows around, heavy scans, bloat indexes
 - syntax available with VACUUM FULL INPLACE
 - still used for system catalogs

- VACUUM FULL now works like CLUSTER
 - The Old
 - move rows around, heavy scans, bloat indexes
 - syntax available with VACUUM FULL INPLACE
 - still used for system catalogs
 - NEW
 - rewrite table and indexes
 - ~ 2% less efficient for tables, 90% more efficient for indexes
 - 1/3 the amount of time
 - continue to avoid it :-)

Postgres 9.0 / Perf / vacuum full



explain can now show buffers information

```
pagila=# explain (analyze, buffers) select * from actor;  
QUERY PLAN
```

```
-----  
Seq Scan on actor (cost=0.00..4.00 rows=200 width=25) (actual time=0.034..0.089 rows=200 loops=1)
```

Buffers: shared read=2

Total runtime: 0.149 ms

(3 rows)

```
pagila=# explain (analyze, buffers) select count(*) from actor;  
QUERY PLAN
```

```
-----  
Aggregate (cost=4.50..4.51 rows=1 width=0) (actual time=0.109..0.109 rows=1 loops=1)
```

Buffers: shared hit=2

-> Seq Scan on actor (cost=0.00..4.00 rows=200 width=0) (actual time=0.010..0.048 rows=200 loops=1)

Buffers: shared hit=2

Total runtime: 0.173 ms

(5 rows)

Postgres 9.0 / Perf / index not null

support index use for IS NOT NULL

PostgreSQL 8.4

```
pagila=# explain select * from address where address2 is not null;
              QUERY PLAN
-----
Seq Scan on address  (cost=0.00..20.03 rows=1 width=70)
  Filter: (address2 IS NOT NULL)
(2 rows)
```

PostgreSQL 9.0 Beta 1

```
pagila=# explain select * from address where address2 is not null;
              QUERY PLAN
-----
Index Scan using address_address2_idx on address  (cost=0.00..8.27 rows=1 width=70)
  Index Cond: (address2 IS NOT NULL)
(2 rows)
```

Postgres 9.0 / Perf / join removal

remove joins from execution plan where not needed

PostgreSQL 9.0 Alpha 4

```
pagila=# explain analyze select actor.last_name from actor left join film_actor using (actor_id) where  
actor.last_update > current_date;
```

QUERY PLAN

```
-----  
Nested Loop Left Join (cost=4.46..38.48 rows=27 width=35) (actual time=0.258..0.258 rows=0 loops=1)  
-> Seq Scan on actor (cost=0.00..5.50 rows=1 width=25) (actual time=0.257..0.257 rows=0 loops=1)  
    Filter: (last_update > ('now'::text)::date)  
-> Bitmap Heap Scan on film_actor (cost=4.46..32.64 rows=27 width=12) (never executed)  
    Recheck Cond: (actor.actor_id = film_actor.actor_id)  
    -> Bitmap Index Scan on film_actor_pkey (cost=0.00..4.45 rows=27 width=0) (never executed)  
        Index Cond: (actor.actor_id = film_actor.actor_id)  
Total runtime: 0.340 ms  
(8 rows)
```

this is not join removal!

Postgres 9.0 / Perf / join removal

PostgreSQL 8.4

```
pagila=# explain analyze select city from city left join country using (country_id) where city.last_update  
> current_date;
```

QUERY PLAN

```
-----  
Nested Loop Left Join (cost=0.00..17.95 rows=1 width=9) (actual time=28.022..28.022 rows=0 loops=1)  
  Join Filter: (city.country_id = country.country_id)  
    -> Seq Scan on city (cost=0.00..14.50 rows=1 width=11) (actual time=28.021..28.021 rows=0 loops=1)  
        Filter: (last_update > ('now'::text)::date)  
    -> Seq Scan on country (cost=0.00..2.09 rows=109 width=4) (never executed)  
Total runtime: 30.567 ms  
(6 rows)
```

PostgreSQL 9.0 Beta 1

```
pagila=# explain analyze select city from city left join country using (country_id) where city.last_update  
> current_date;
```

QUERY PLAN

```
-----  
Seq Scan on city (cost=0.00..14.50 rows=1 width=11) (actual time=0.748..0.748 rows=0 loops=1)  
  Filter: (last_update > ('now'::text)::date)  
Total runtime: 0.777 ms  
(3 rows)
```

Postgres 9.0 / Perf / tablespace costs



- Per Tablespace GUC Options
 - seq_page_cost
 - random_page_cost
- ALTER TABLESPACE ... SET / RESET

Postgres 9.0 / Perf / tablespace costs



Postgres 9.0 / Perf / Wait, there's more!



Postgres 9.0 / Perf / Wait, there's more!

- Increased GIN index creation for pre-ordered data
- ALTER TABLE “rewrite” commands can skip xlogging
- ALTER TABLE ... ALTER COLUMN ... SET DISTINCT
- Optimize `foo <> true => foo = false`, `foo <> false => foo = true`
- Improve `pg_restore` multi-workers logic for faster restores
- Windows 64bit Support

Postgres 9.0 / Perf / Wait, there's more!



Postgres 9.0

- Performance
- **Administration**
- Development
- Procedures
- Replication

initdb is dead (well, to me)

```
sa-x:postgres rob$ pgsq190/bin/pg_ctl -D data/ initdb -o "--locale=en_US.UTF-8"
```

The files belonging to this database system will be owned by user "rob".

This user must also own the server process.

The database cluster will be initialized with locale en_US.UTF-8.

The default database encoding has accordingly been set to UTF8.

The default text search configuration will be set to "english".

...

greatest feature in postgres 9?

```
sa-x:postgres rob$ pgsq|90b I/bin/psql -d pagila
FATAL: unrecognized configuration parameter "application_name"
psql (9.0beta I, server 8.3.5)
WARNING: psql version 9.0, server version 8.3.
    Some psql features might not work.
Type "help" for help.
```

```
pagila=# \d actor
```

```
Table "public.actor"
Column | Type | Modifiers
-----+-----+-----
actor_id | integer | not null default nextval('actor_actor_id_seq'::regclass)
first_name | character varying(45) | not null
last_name | character varying(45) | not null
last_update | timestamp without time zone | not null default now()
```

Indexes:

```
"actor_pkey" PRIMARY KEY, btree (actor_id)
```

```
"idx_actor_last_name" btree (last_name)
```

Referenced by:

```
TABLE "film_actor" CONSTRAINT "film_actor_actor_id_fkey" FOREIGN KEY (actor_id) REFERENCES actor(actor_id) ON UPDATE CASCADE ON DELETE RESTRICT
```

Triggers:

```
last_updated BEFORE UPDATE ON actor FOR EACH ROW EXECUTE PROCEDURE last_updated()
```


GRANT/REVOKE ON ALL object IN SCHEMA

```
pagila=# create role dylan;
```

```
CREATE ROLE
```

```
pagila=# grant select on all tables in schema public to dylan;
```

```
GRANT
```

```
pagila=# select oid::regclass, relacl from pg_class where relkind='r' and relnamespace='2200';
```

oid	relacl
-----+-----	
film_actor	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment_p2007_02	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment_p2007_03	{postgres=arwdDxt/postgres,dylan=r/postgres}
city	{postgres=arwdDxt/postgres,dylan=r/postgres}
actor	{postgres=arwdDxt/postgres,dylan=r/postgres}
category	{postgres=arwdDxt/postgres,dylan=r/postgres}
film	{postgres=arwdDxt/postgres,dylan=r/postgres}
address	{postgres=arwdDxt/postgres,dylan=r/postgres}
store	{postgres=arwdDxt/postgres,dylan=r/postgres}
staff	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment_p2007_04	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment_p2007_05	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment_p2007_06	{postgres=arwdDxt/postgres,dylan=r/postgres}
rental	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment_p2007_01	{postgres=arwdDxt/postgres,dylan=r/postgres}
country	{postgres=arwdDxt/postgres,dylan=r/postgres}
payment	{postgres=arwdDxt/postgres,dylan=r/postgres}
film_category	{postgres=arwdDxt/postgres,dylan=r/postgres}
language	{postgres=arwdDxt/postgres,dylan=r/postgres}
customer	{postgres=arwdDxt/postgres,dylan=r/postgres}
inventory	{postgres=arwdDxt/postgres,dylan=r/postgres}

ALTER DEFAULT PRIVILEGES

```
pagila=# alter default privileges grant select on tables to dylan;
```

```
ALTER DEFAULT PRIVILEGES
```

```
pagila=# create table payment_p2007_07 () inherits (payment);
```

```
CREATE TABLE
```

```
pagila=# \z payment_p2007_07
```

Schema	Name	Type	Access privileges	Column access
public	payment_p2007_07	table	postgres=arwdDxt/postgres+ dylan=r/postgres	

```
(1 row)
```

Postgres 9.0 / Admin / \d for children

```
pagila=# \d payment
```

Column	Type	Table "public.payment"	Modifiers
payment_id	integer		not null default nextval('payment_payment_id_seq'::regclass)
customer_id	smallint		not null
staff_id	smallint		not null
rental_id	integer		not null
amount	numeric(5,2)		not null
payment_date	timestamp without time zone		not null

Indexes:

```
"payment_pkey" PRIMARY KEY, btree (payment_id)
"idx_fk_customer_id" btree (customer_id)
"idx_fk_staff_id" btree (staff_id)
"payment_rental_id_idx" btree (rental_id)
```

Foreign-key constraints:

```
"payment_customer_id_fkey" FOREIGN KEY (customer_id) REFERENCES customer(customer_id) ON UPDATE CASCADE ON DELETE RESTRICT
"payment_rental_id_fkey" FOREIGN KEY (rental_id) REFERENCES rental(rental_id) ON UPDATE CASCADE ON DELETE SET NULL
"payment_staff_id_fkey" FOREIGN KEY (staff_id) REFERENCES staff(staff_id) ON UPDATE CASCADE ON DELETE RESTRICT
```

Rules:

```
<snip>
```

Number of child tables: 7 (Use \d+ to list them.)

Postgres 9.0 / Admin / \d for children

```
pagila=# \d+ payment
```

Column	Type	Modifiers	Storage
payment_id	integer	not null default nextval('payment_payment_id_seq'::regclass)	plain
customer_id	smallint	not null	plain
staff_id	smallint	not null	plain
rental_id	integer	not null	plain
amount	numeric(5,2)	not null	main
payment_date	timestamp without time zone	not null	plain

Indexes:

```
"payment_pkey" PRIMARY KEY, btree (payment_id)
"idx_fk_customer_id" btree (customer_id)
"idx_fk_staff_id" btree (staff_id)
"payment_rental_id_idx" btree (rental_id)
```

Foreign-key constraints:

```
"payment_customer_id_fkey" FOREIGN KEY (customer_id) REFERENCES customer(customer_id) ON UPDATE CASCADE ON DELETE RESTRICT
"payment_rental_id_fkey" FOREIGN KEY (rental_id) REFERENCES rental(rental_id) ON UPDATE CASCADE ON DELETE SET NULL
"payment_staff_id_fkey" FOREIGN KEY (staff_id) REFERENCES staff(staff_id) ON UPDATE CASCADE ON DELETE RESTRICT
```

Rules:

```
<snip>
```

**Child tables: payment_p2007_01,
payment_p2007_02,
payment_p2007_03,
payment_p2007_04,
payment_p2007_05,
payment_p2007_06,
payment_p2007_07**

Has OIDs: no

Postgres 9.0 / Admin / Wait, there's more!



Postgres 9.0 / Admin / Wait, there's more!

- show value when unique constraints are violated
- radius authentication
- add SQLSTATE as option to log_line_prefix
- allow collection of statistics on sequences
- GUC can now be configured based on user/database combination
- log changes to postgresql.conf during reload
- reindexing shared system catalogs is now crash safe
- add ability to reset statistics on individual tables
- support samehost/samenet options in pg_hba.conf
- added string_agg function (group_concat)
- improved tabular display for psql rows with null data
- add index methods to \di

Postgres 9.0 / Admin / Wait, there's more!



Postgres 9.0

- Performance
- Administration
- **Development**
- Procedures
- Replication


```
CREATE TRIGGER notify_waiting_list
AFTER UPDATE ON rental FOR EACH ROW
WHEN NEW.return_date <> OLD.return_date
THEN EXECUTE procedure send_notices();
```

- improves control for execution of triggers
- any boolean expression
- shows up in \d output
- some after triggers will gain significant speed up

Postgres 9.0 / Dev / conditional triggers

```
CREATE TRIGGER film_fulltext_trigger
BEFORE INSERT OR UPDATE OF title, description
ON FILM FOR EACH ROW EXECUTE PROCEDURE
tsvector_update_trigger('fulltext', 'pg_catalog.english', 'title', 'description')
```

- improves control for execution of triggers
- shows up in \d output
- simplify code
- trigger only fires based on column declaration in update clause

This is not the same thing as constraint exclusion!

“Yes, we suck at naming things”

- Magnus Hagander

Postgres 9.0 / Dev / exclusion constraints



- Generic constraint infrastructure
- UNIQUE constraints are like =
 - any indexable commutative operator
- BETWEEN-rows constraint
 - each row can conflict with any other row in the table

Postgres 9.0 / Dev / exclusion constraints



Postgres 9.0 / Dev / exclusion constraints



- Examples:
 - non-overlapping time intervals
 - calendars, scheduling, reservations
 - non-overlapping geometric shapes
 - zoning

Postgres 9.0 / Dev / exclusion constraints



Jeff Davis, “Not Just Unique”, Thu @ 1:30PM

- Examples:
 - non-overlapping time intervals
 - calendars, scheduling, reservations
 - non-overlapping geometric shapes
 - zoning

Jeff Davis, “Not Just Unique”, Thu @ 1:30PM

Jeff Davis, “Not Just Unique”, Thu @ 1:30PM

Postgres 9.0 / Dev / window functions

```
pagila=# pagila=# select date_trunc('week',payment_date), sum(avg(sum(amount))
pagila-# over (order by date_trunc('week', payment_date) rows between 1 preceding and 1 following)
pagila-# from payment_p2007_04 group by date_trunc('week',payment_date) order by 1;
    date_trunc    |  sum  |          avg
-----+-----+-----
2007-04-02 00:00:00 | 6562.62 | 7237.230000000000000000
2007-04-09 00:00:00 | 7911.84 | 7611.856666666666666667
2007-04-23 00:00:00 | 8361.11 | 7332.280000000000000000
2007-04-30 00:00:00 | 5723.89 | 7042.500000000000000000
(4 rows)
```

- enhancements to the frame clause
- now able to compute moving averages (easily)
- several options, check the docs!

Postgres 9.0 / Dev / Wait, there's more!



Postgres 9.0 / Dev / Wait, there's more!

- deferrable unique constraints
- new bytea hex format output
- hstore... more efficient, kv storage
- show value when unique constraints are violated
- added string_agg function (group_concat)
- support SQL Standard LIMIT/OFFSET
- drop if exists columns
- drop if exists constraints
- ordered aggregates
- rewrite listen/notify (payloads)

Postgres 9.0 / Dev / Wait, there's more!



Postgres 9.0

- Performance
- Administration
- Development
- Procedures
- Replication

plpgsql is now created by default

Postgres 9.0 / Proc / just DO it!

```
pagila=# DO $$
pagila$# DECLARE
pagila$#   v_part record;
pagila$#   v_sql text;
pagila$# BEGIN
pagila$#   set search_path = 'public';
pagila$#   for v_part in select tablename from pg_tables where tablename ~ 'payment' loop
pagila$#     v_sql := 'create index '||v_part.tablename||'_rental_id_idx ON '||v_part.tablename||'(rental_id)';
pagila$#     raise notice '%',v_sql;
pagila$#     execute v_sql;
pagila$#   end loop;
pagila$# END
pagila$# $$ language plpgsql;
NOTICE: create index payment_p2007_02_rental_id_idx ON payment_p2007_02(rental_id)
NOTICE: create index payment_p2007_03_rental_id_idx ON payment_p2007_03(rental_id)
NOTICE: create index payment_p2007_07_rental_id_idx ON payment_p2007_07(rental_id)
NOTICE: create index payment_p2007_04_rental_id_idx ON payment_p2007_04(rental_id)
NOTICE: create index payment_p2007_05_rental_id_idx ON payment_p2007_05(rental_id)
NOTICE: create index payment_p2007_06_rental_id_idx ON payment_p2007_06(rental_id)
NOTICE: create index payment_p2007_01_rental_id_idx ON payment_p2007_01(rental_id)
NOTICE: create index payment_rental_id_idx ON payment(rental_id)
DO
pagila=#
```

Postgres 9.0 / Proc / just DO it!

```
pagila=# \di payment_p2007_0*rental*
```

List of relations

Schema	Name	Type	Owner	Table
public	payment_p2007_01_rental_id_idx	index	postgres	payment_p2007_01
public	payment_p2007_02_rental_id_idx	index	postgres	payment_p2007_02
public	payment_p2007_03_rental_id_idx	index	postgres	payment_p2007_03
public	payment_p2007_04_rental_id_idx	index	postgres	payment_p2007_04
public	payment_p2007_05_rental_id_idx	index	postgres	payment_p2007_05
public	payment_p2007_06_rental_id_idx	index	postgres	payment_p2007_06
public	payment_p2007_07_rental_id_idx	index	postgres	payment_p2007_07

(7 rows)

Postgres 9.0 / Proc / Wait, there's more!



Postgres 9.0 / Proc / Wait, there's more!

- can now use expressions in OPEN cursor FOR EXECUTE
- improve variable recognition within plpgsql
- allow assignment of values to IN parameters
- Allow MOVE FORWARD n/ ALL, MOVE BACKWARD n/ ALL for plpgsql cursors
- add utility functions for plperl (quote_literal and friends)
- use strict now works in plperl
- change Perl warnings to elog(WARNING)
- Add Python 3 support
- support for C++ stored procs
- named parameters

Postgres 9.0 / Proc / Wait, there's more!



Postgres 9.0

- Performance
- Administration
- Development
- Procedures
- **Replication**

Postgres 9.0 / repl / hot standby

- allows for (read only) queries against a database during crash recovery
 - aka pitr replication / xlog shipping
- GUC Settings
 - primary
 - wal_level = 'hot standby'
 - vacuum_defer_cleanup_age = 0
 - secondary
 - hot_standby = on
 - max_standby_delay = 30s

Postgres 9.0 / repl / streaming replication

- Process to connect to the primary from a secondary server and “stream” changes to slaves
- Config primary for archive logging, then
 - `max_wal_senders`
- Config secondary for streaming replication
 - in `recovery.conf`
 - `standby_mode = on`
 - `primary_conninfo = ...`
 - in `postgresql.conf`
 - `wal_level = archive`
 - `hot_standby = on`

Postgres 9.0 / repl / Wait, there's more!



Postgres 9.0 / repl / Wait, there's more!

- actually, there isn't any more :-)

Postgres 9.0 / repl / Wait, there's more!



Heikki, "Built In-Replication", Thu @ 10:00AM

Postgres 9.0

- Performance
- Administration
- Procedures
- Tools
- Replication
- **Bonus Round!!**

Postgres 9.0 / Upgrades

- dump / restore recommend
- “pg_upgrade” (the tool formerly known as pg_migrator)
 - 8.3, 8.4 -> 9.0 (not dev releases)
- replication
 - ?

Postgres 9.0 / Upgrades

- Beware of incompatibilities
 - plpgsql variable conflicts
 - read the release notes!

Postgres 9.0 / Testing

- Download
 - 9.0 Beta 1 release
 - git, <http://github.com/gregs1104/peg>
- Test
 - import existing schemas
 - import existing data
 - run application / code tests
 - test new features
 - test pg_upgrade

Postgres 9.0 / More Info

- <http://www.depesz.com/>
- <http://wiki.postgresql.org/>
- <http://www.xzilla.net>
- PGCon, Ottawa, Canada, May 19th - 22nd (that's now!)

Postgres 9.0 / Shout Out

- Hubert Lubaczewski, aka “Depesz”
- Magnus Hagander
- Andreas Scherbaum
- Chris Biggs
- <http://www.omniti.com/is/hiring>

Thank you for listening.



OmniTI / Presentation