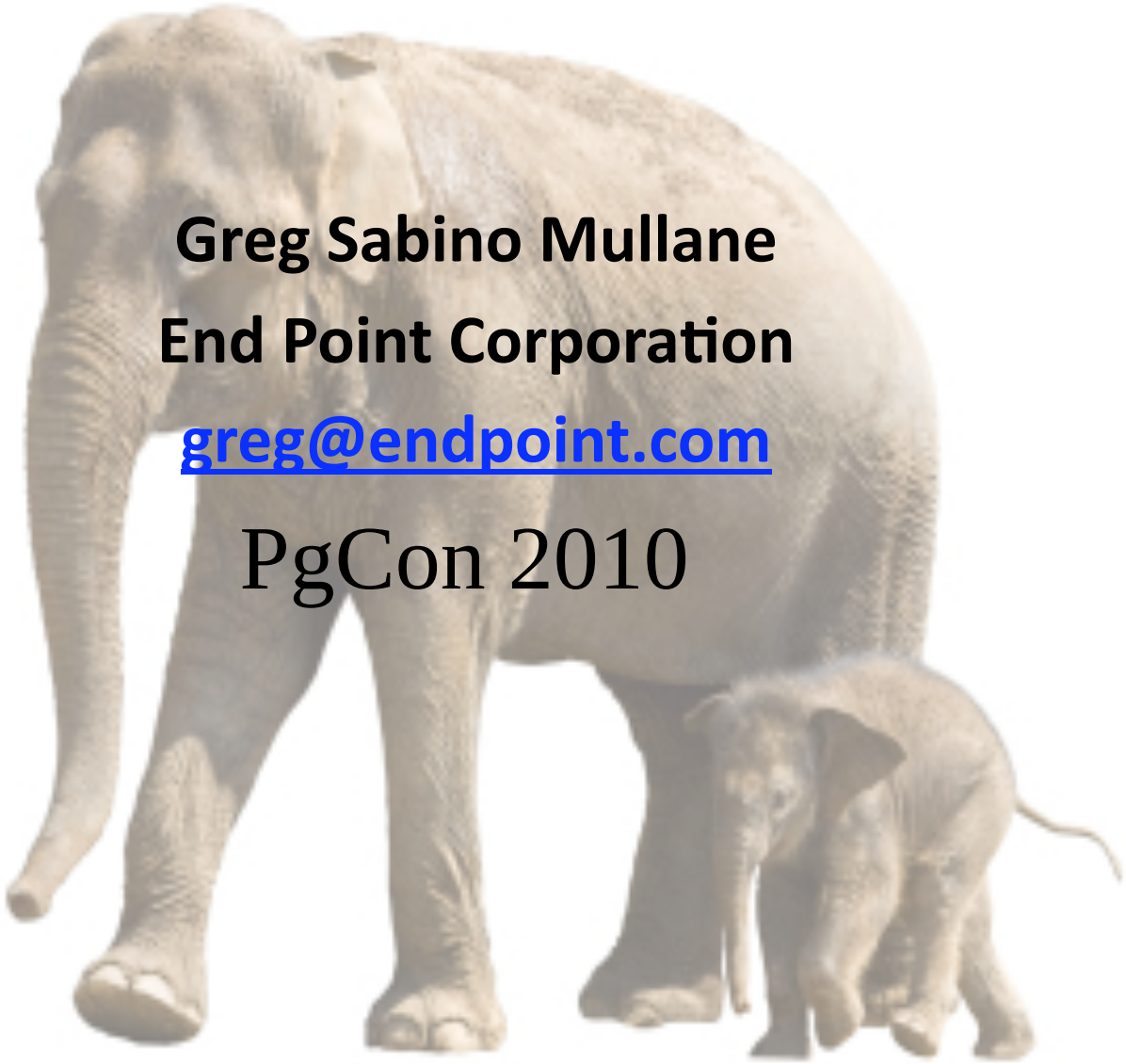
Postgres (for non-Postgres people)



Greg Sabino Mullane

End Point Corporation

greg@endpoint.com

PgCon 2010

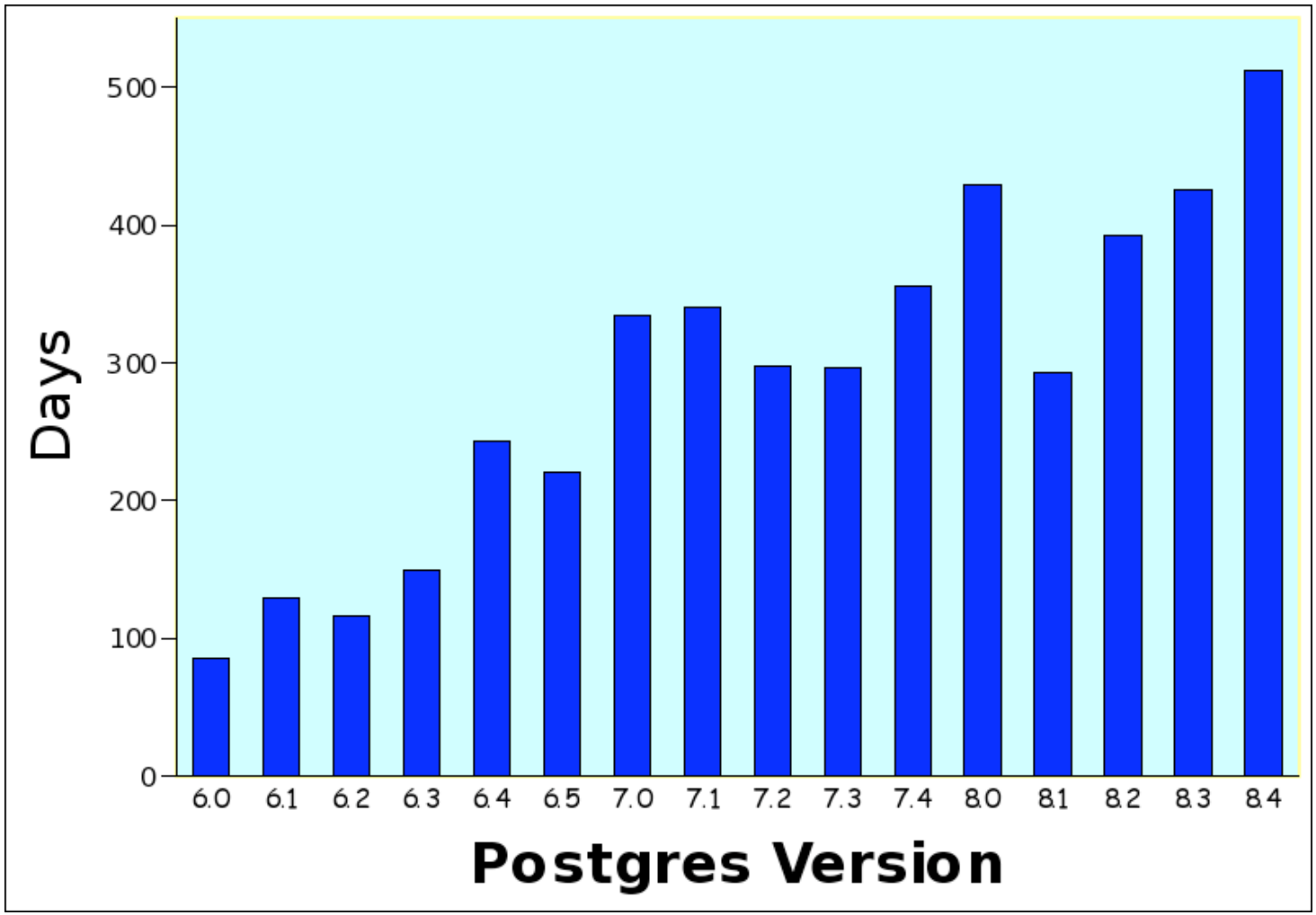
SELECT SUM(info) FROM talk;

- Conversion advice
- Postgres gotchas
- Postgres limitations and features
- Organization and community

The Part of Whys

- Why MySQL?
- Why Postgres?

- Postgres? PostgreSQL? Postgre? PostgresSQL?



Migrating Your App

- Schema
- Data
- Application
- Support

Migration : Schema

- Mysqldump?
- --compatible=postgresql --no-data
- Redesign
- Conversion tools

Migration : SQL Spec

- Who cares?
- Postgres vs. the spec
- MySQL vs. the spec
- Oracle vs. the spec

Migration : Schema

- Tables
- Engines and plugins
- Customization

Migration : Schema : Data Types : Numbers

- **INTEGER** (smallint, bigint, serial)
- **NUMERIC** (double precision, money)
- **REAL**

Migration : Schema : Data Types : Text

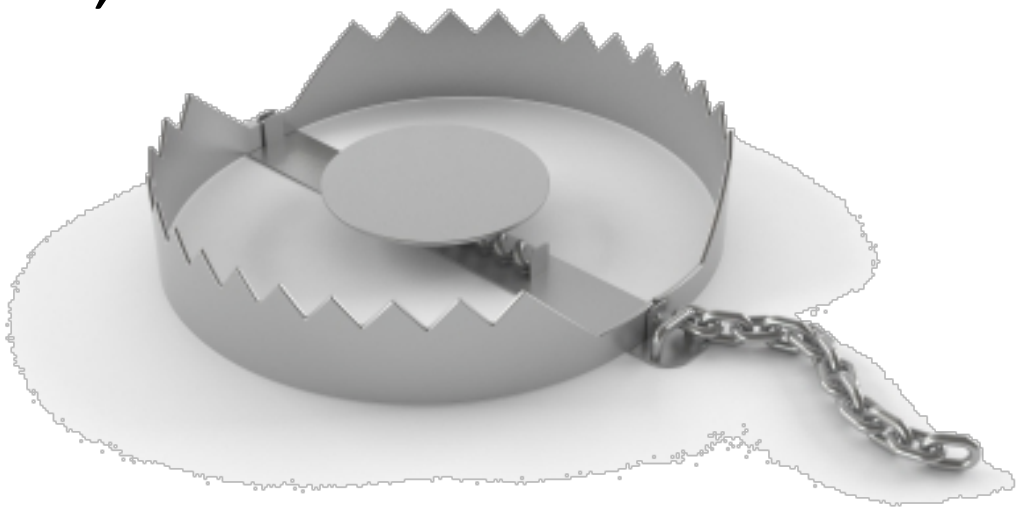
- **TEXT**
- **VARCHAR(n)**
- CHAR(n)

Migration : Schema : Data Types : Dates

- **DATE**
- **TIMESTAMPTZ**
- TIME
- INTERVAL

Migration : Schema : Data Types : Boolean

- **BOOL**
- TRUE, ‘t’, ‘y’, ‘yes’, ‘on’, 1
- FALSE, ‘f’, ‘n’, ‘no’, ‘off’, 0



Migration : Schema : Data Types : Binary

- **BYTEA**
- Internal or external?

Migration : Schema : Data Types : Others

- **Geometric** (line, path, box, circle, polygon)
- **Arrays**
- **ENUM**
- **CIDR, INET, MACADDR**
- **UUID**
- **XML**

Migration : Schema : Sequence

- Auto-increment
- **SEQUENCE**
- **SERIAL**
- INTEGER NOT NULL DEFAULT nextval(‘foo’)

Migration : Schema : DEFAULT

- **cdate TIMESTAMPTZ NOT NULL DEFAULT now()**
- **(almost) anything at all**
- No magic

Migration : Schema : FOREIGN KEYS

- **Fully supported**
- **ON DELETE CASCADE**
- **ON DELETE RESTRICT**
- **ON DELETE SET NULL**
- **ON DELETE SET DEFAULT**
- **DEFERRED INITIALLY DEFERRED**

Migration : Schema : Indexes

- **CREATE INDEX foo ON mytab(mycol);**
- **CREATE INDEX foo2 ON mytab(mycol) WHERE size = ‘grande’;**
- **CREATE INDEX foo3 ON mytab(LOWER(blurb));**
- **CREATE INDEX foo4 ON mytab(LOWER(blurb)) WHERE size = ‘grande’;**

Migration : Schema : Indexes

- CREATE INDEX **CONCURENTLY** foo ON mytab(mycol);
- No rebuilding of tables
- Reverse scan, bitmap

Migration : Schema : Triggers

- CREATE TRIGGER mytrig BEFORE INSERT OR DELETE OR UPDATE ON mytab FOR EACH ROW EXECUTE PROCEDURE somefunc();

Migration : Schema : Functions

- Very custom
- Pl/PGSQL == Pl/SQL

Migration : Data

- No autocommit, fsync off, no autovacuum
- COPY vs. INSERT
- Turn off indexes and triggers (FK)
- ANALYZE

ANALYZE

- Cost-based planner
- Autovacuum
- Initial load
- Default_statistics_target
  - 10
  - 100
  - 1000
  - More?

VACUUM

- MVCC – Oracle vs. Pg
- Rollback segment
- Concurrency and locking
- Table bloat
- VACUUM FULL, CLUSTER, REINDEX

Migration : Application

- aka SQL
- Clients (PHP exception)
- SQL modes
  - Traditional
  - Postgresql
- “Any client can change its own session sql_mode value at any time.”

Migration : SQL : NULL

- NULL is NULL

Migration : SQL : SELECT

- Query planner
- JOINS
- Subselects

Migration : SQL : GROUP BY

- SELECT a,b,c FROM mytab GROUP BY a,b
- Standard or not?

DISTINCT ON

Migration : SQL : Text match

- CHAR / VARCHAR / TEXT case sensitive
- SELECT count(*) FROM mytab WHERE mycol = ‘Fred’;

- …WHERE LOWER(mycol) = ‘fred’;

Migration : SQL : full text search

- Tsearch2: contrib vs. core
- Powerful, complex
- Transactional!

Migration : SQL : DELETE

- One table only
- No LIMIT, no ORDER BY
- No QUICK, LOW PRIORITY, IGNORE
- DELETE FROM ONLY mytab WHERE…
- DELETE FROM mytab USING otab WHERE mycol = otab.id AND otab.size = 'foo';
- DELETE FROM mytab WHERE mycol IN (SELECT id FROM otab WHERE size = ‘foo’);

Migration : SQL : UPDATE

- Only one table
- No ORDER BY, LIMIT, etc.
- UPDATE mytab SET mycol=123 FROM otab WHERE mycol = otab.id AND otab.size = ‘foo’;

Migration : SQL : INSERT

- Only one table
- No REPLACE or ON DUPLICATE KEY UPDATE
- UPSERT via plpgsql or app logic
- DEFAULT
- INSERT INTO mytab DEFAULT VALUES

Migration : SQL : ||

- Ugh
- CONCAT
- Wrapper functions
- NULL an empty string

Migration : SQL: GRANT

- ..and REVOKE
- Almost any object
- Immediate effect

Migration : SQL : locking

- Use sparingly
- Share, exclusive
- Advisory
- ALTER TABLE, REINDEX, VACUUM FULL

Migration : SQL : Aliases

- AS optional but recommended
- SELECT SUM(mycol) AS mysum
- SELECT * FROM pg_class c, mytab t…

Migration : SQL : Advanced

- WITH
- Windowing
- Recursive functions

Migration : Support

- Routine stuff: autovacuum
- Monitoring (check_postgres)
- Backups (MVCC based)

Postgres : Quirks : lowercase

- Implicit case folding
- CREATE TABLE abc (A int);
- CREATE TABLE ABC (a int);
- CREATE TABLE “abc” (“a” int);
- CREATE TABLE “ABC” (“A” int);
- SELECT * FROM “ABC” WHERE “A”=123;

Postgres : Quirks : Casting

- 8.3 removed some implicit casts
- SELECT 1 = ‘1’::text;
- ERROR: operator does not exist: integer=text
- 1. Fix the app
- 2. Add the casts back in

Postgres : Quirks : COUNT(*)

- MyISAM vs INNODB
- Use an index
- Triggers
- SELECT reltuples FROM pg_class WHERE relname = ‘foo’;

Postgres : Quirks : Schemas

- Database versus schemas versus cluster
- Changes are cheap
- No name-based schemas
- The ‘public’ schema
- contrib/dblink

Postgres : Quirks : vacuum

- Newer the better
- autovacuum

Postgres : Quirks : i18n

- SQL_ASCII vs UTF-8
- Per database only

Postgres : Quirks : Hints

- No hints per se
- Smart planner
- SET enable_* EXPLAIN
- Few other knobs – per cluster

Postgres : Quirks : Rules

- Last resort.

Postgres : Drawbacks

- Replication
- In-place upgrade (~~pg_migrator,~~ Bucardo)
- Tuned for a toaster
- Name
- Publicity and marketing

pg_upgrade

Postgres : Strengths

- Custom everything
- Transactional DDL
- PostGIS
- Query planner
- Support
- Authentication: pg_hba.conf
  - PAM, ident, Kerberos, LDAP, …

Postgres : Organization

- Who?
- Transparent meritocracy
- “core”
- Committers
- Mailing list, IRC, wiki
- Company roles
- Spread the risk
- Cannot be bought (assets or people)

Postgres : Community

- Pgfoundry, github, bucardo.org, …
- PostGIS
- Mailing list? Bug report? User?
- Interaction!
- Wiki, advocacy, sysadmin, docs, packagers
- Clients, tools, replication systems
- Volunteers (IRC)

Postgres : Infrastructure

- Servers, companies
- Build farm
- Wiki

Postgres : Development

- Individual developers vs. companies
- CVS vs. git git wins!

Postgres : Version

- Roughly every year
- 8.3 8.3.10
- Major version, minor version, revision

Postgres : Patches

- Bug report, mailing list post, TODO list
- Run up the flagpole (Tom Lane test)
  - Stable?
  - Useful?
  - Spec compliant?
  - Side effects?
  - Best approach to problem?
- Diff format, formatting, docs
- Added to patch queue, commitfest
- Patch reviewer

Postgres : Patches

- High standards
- \timing
- \dfS

Postgres : Licensing

- BSD (or MIT) PostgreSQL license
- Free as in speech
- Free as in beer


SELECT questions FROM audience;