PostgreSQL: present and near future
1/Less obvious than it should
1.1/SQL sucks! Let's use something new!

- SQL is much more than a language to access your database: SQL allows you to forget about which is the most efficient way to obtain the data you need in a large dataset.
1.2/Transactions don't look the same everywhere

- We usually compare PostgreSQL to Oracle as if that was the opponent to beat, but try to CREATE TABLE inside a transaction in Oracle...
1.3/And much, much more

- Locking
- Indexes
- Storage optimized for lots of datatypes
- ...

...
1.4/...including an encredible (and profitable) community

- EnterpriseDB – Robert Haas – Helped Amazon build RDS
- 2ndQuadrant – Simon Riggs – Developers around the world working on key core features
- Citus – Andres Freund – CitusDB: clustered PostgreSQL
- PostgreSQL Professional – Oleg Bartunov – Full Text Search
- Nippon Telegraph & Telephone...
- Tom Lane
- ...

1.4/...including an encredible (and profitable) community
2/What's new in 9.6
2.1/Scalability

- A lot of work has gone in recent versions to make PostgreSQL highly scalable (vertically) by optimizing locked codeblocks.
- Multiple standby servers
- postgres_fdw
2.1.1/Multiple standby servers – replication
2.2/Parallel queries

- Previously only one processor/core could be used per query. Now it can be used in **sequential scans**, **joins** and **aggregates**.
2.3/SQL completeness

- CUBE, ROLLUP, GROUPING SETS
- WITH & WITH RECURSIVE
- MATERIALIZED VIEWS
- Full text search
- Row-level security
- JSONB
2.4/Durability

- Checksums
- WAL
- synchronous_commit = on/off
2.4.1/Durability - Architecture

- Client
- Postgres Shared memory
- Filesystem Cache
- RAM
- Structured data
- Durable storage
- WAL: Write Ahead Log
3/What's the community working on for v10?
3.1/Scalability

- Even more scalability work (large number of cores and clients required for testing)
- Quorum commit
- Logical replication – pglogical in core
- Pushing more stuff into postgres_fdw
3.1.1/Quorum commit
3.2/SQL completeness

- AUTONOMOUS TRANSACTIONS
3.3/Parallel queries

- CREATE INDEX – sorting
- Bitmap scans
- Index scans
- Asynchronous execution
3.4/Durability

- Amcheck – tool to check the correctness of data
- pg_heal
3.5/Performance

- Sorting improvements
- Constant evaluation improvements
- Fetching several tuples “at once”
4/What should Tryton be using but isn't and what can we learn?
4.1/Texts and searching

- Unaccent
- Full text search
- Trigram indexes for LIKE searches
4.2/Better indexes

- How do we create indexes which are not re-done on update?
4.3/To learn: tools to know what's going on

- PostgreSQL has lots of tools to try to understand what's going on.
http://www.NaN-tic.com

Albert Cervera i Areny
albert@nan-tic.com
@albertnan
linkedin.com/in/albertca
nàN-tic
Perquè vols canviar, i canvis.