

# Will Postgres Live Forever?

BRUCE MOMJIAN



This presentation explains the long life of open source software, and the life cycle differences between proprietary and open source software. *Title concept from Renee Deger*

<https://momjian.us/presentations>

*Creative Commons Attribution License*

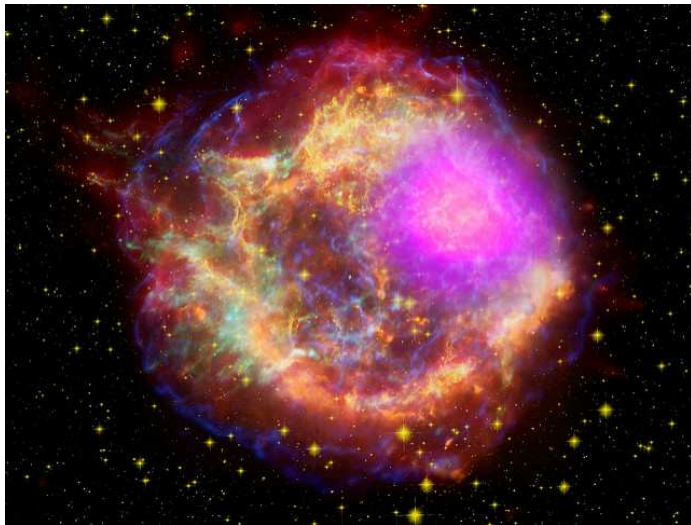


*Last updated: November, 2020*

# Outline

1. Forever
2. Software life cycle
3. Open source adoption
4. Postgres innovation
5. Community structure
6. Conclusion

# 1. Forever



<https://www.flickr.com/photos/gsfcr/>

# Forever Is a Long Time

- Age of the Universe: 13.7 billion years
- Age of the Earth: 4.5 billion years
- Age of civilization: 6,000 years
- Civilized era vs. Earth years: 0.00001%
- Digital era vs. Earth years:  $\sim 0\%$

# Brief Digital History

1804: Jacquard loom

1945: ENIAC

1970: E. F. Codd Relational Theory

1974: System R

1977: Ingres

1986: University-based Postgres

1994: Postgres95

1996: Internet-based Postgres

## 2. Software Life Cycle



<https://www.flickr.com/photos/tarynmarie>

# Proprietary Software Life Cycle

1. Innovation
2. Market growth
3. Market saturation
4. *Maximize profit, minimize costs (development, support)*
5. Maintenance mode (no new features, no innovation)
6. End-of-life

# Open Source Software Life Cycle

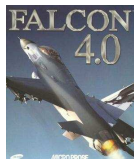
1. Parity with proprietary software, low cost
2. Market growth
3. *Continue innovation or decline*
4. Source code is always available to continue



# Illustrative Example of Open Source Growth

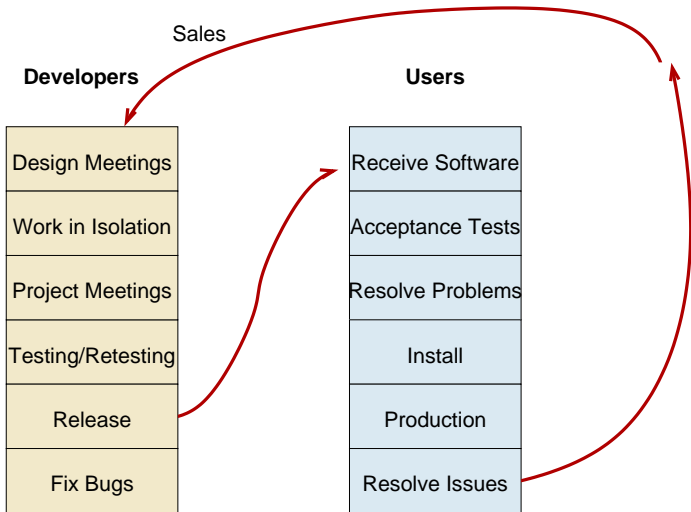
One of the longest-developed computer games:

- 1984: Spectrum HoloByte begins Falcon development
- 1998: MicroProse releases Falcon 4.0
- 1999: MicroProse ends development
- 2000: leak of source code
- 2003: Benchmark Sims (BMS) releases community modifications
- 2005: Lead Pursuit releases Allied Force, which includes BMS mods
- 2015: GOG.com republishes Falcon 4.0
- 2015: BMS releases version 4.33, plus later minor releases

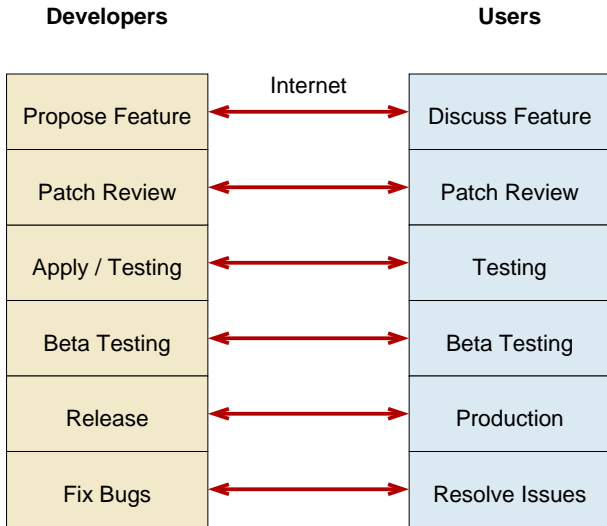


[https://en.wikipedia.org/wiki/Falcon\\_4.0](https://en.wikipedia.org/wiki/Falcon_4.0)

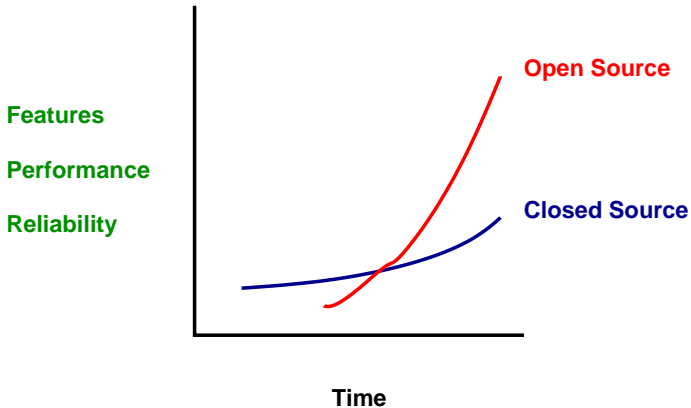
# Proprietary Development Flow



# Open Source Development Flow



# Rise of Open Source



Linux attained feature parity with:

- HP-UX
- AIX
- Solaris

and then went on to innovate beyond them.

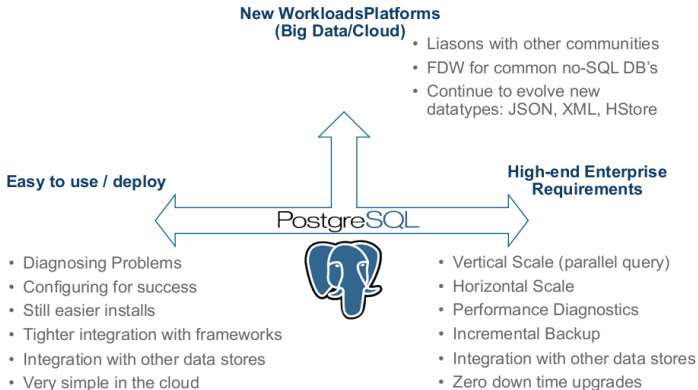
# Postgres

Postgres nearing feature parity with:

1. Oracle
2. DB2
3. MS-SQL
4. Sybase
5. Informix
6. Ingres Corp.

and then going on to innovate beyond them.

# Many Focuses



# When Does Software Die?

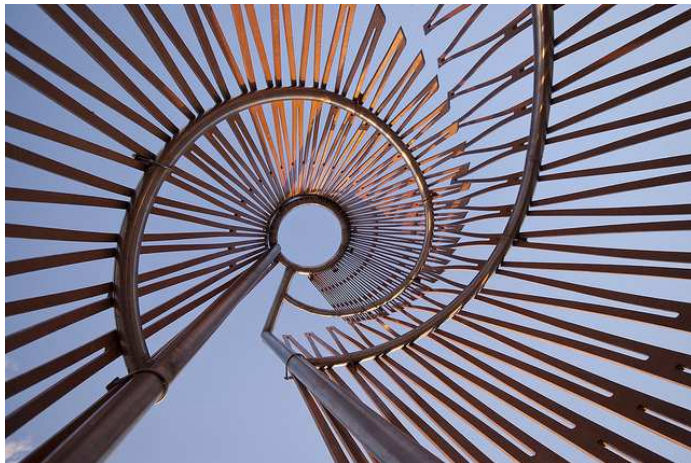
- Proprietary software dies when the owner of the source code can no longer profit from it.
- It declines long before death due to profit maximization.
- Open source cannot die in the same way.
- Open source remains active while it serves a purpose.
- It can always be resurrected if useful.
- Postgres was given new life in 1996.



# Ideas Don't Die

1. Ideas don't die, as long as they are shared.
2. Ideas are shared, as long as they are useful.
3. Postgres will live, as long as it is useful.

### 3. Open Source Adoption



<https://www.flickr.com/photos/99438314@N02/>

# Open Source Survey, 2016

When the first survey launched 10 years ago, hardly anyone would have predicted that open source use would be ubiquitous worldwide just a decade later, but for many good reasons that's what happened. Its value in reducing development costs, in freeing internal developers to work on higher-order tasks, and in accelerating time to market is undeniable. Simply put, open source is the way applications are developed today.

*Lou Shipley  
President And CEO  
Black Duck Software*

# Advantages of Open Source

1. Competitive features, innovation
2. Freedom from vendor lock-in
3. Quality of solutions
4. Ability to customize and fix
5. Cost
6. Speed application development
7. Reduce development costs
8. Interoperability
9. Breadth of solutions

# Open Source Today

Open source today is unequivocally the engine of innovation; whether that's powering technology like operating systems, cloud, big data or IoT, or powering a new generation of open source companies delivering compelling solutions to the market.

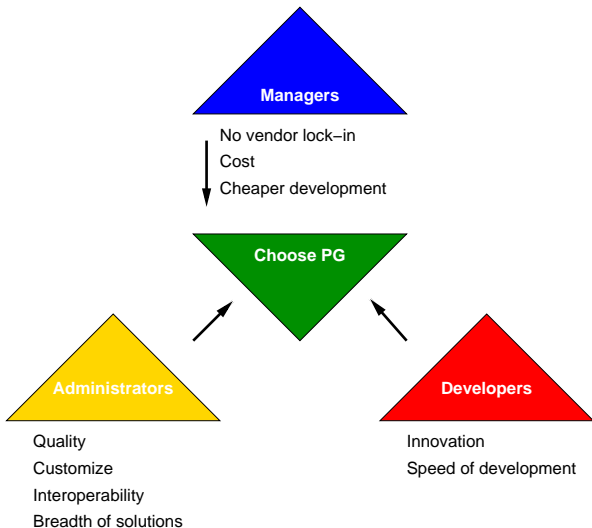
Paul Santinelli  
General Partner  
North Bridge

# Open Source Usage, 2016

1. Operating Systems
2. Database
3. Development tools

Database didn't appear in the top three the previous year's survey (2015).

# Advantages of Open Source for Database Decision Makers



## 4. Postgres Innovation

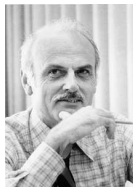


[https://www.flickr.com/photos/tomas\\_vondra/](https://www.flickr.com/photos/tomas_vondra/)



# Relational Innovation

- E. F. Codd introduces relational theory
- Row, column, table
- Constraints
- Normalization, joins
- Replaces key/value data storage systems
- Pre-Postgres



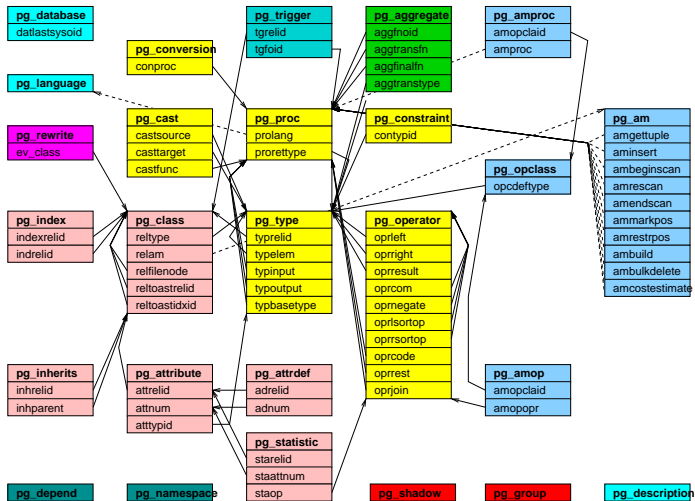
# University Postgres Innovation

- Michael Stonebraker creates university Postgres
- Allows extendability via system table contents:
  - Data types
  - Indexing methods
  - Server-side languages



[https://en.wikipedia.org/wiki/Michael\\_Stonebraker](https://en.wikipedia.org/wiki/Michael_Stonebraker)

# Postgres Extensibility



# Postgres Extension Data Type

```
CREATE EXTENSION isn;
```

```
\dT
```

List of data types

Schema	Name	Description
public	ean13	International European Article Number (EAN13)
public	isbn	International Standard Book Number (ISBN)
public	isbn13	International Standard Book Number 13 (ISBN13)
public	ismn	International Standard Music Number (ISMN)
public	ismn13	International Standard Music Number 13 (ISMN13)
public	issn	International Standard Serial Number (ISSN)
public	issn13	International Standard Serial Number 13 (ISSN13)
public	upc	Universal Product Code (UPC)

<https://momjian.us/main/writings/pgsql/central.pdf>

# Postgres Server-Side Languages

- PL/Java
- PL/Perl
- PL/pgSQL (like PL/SQL)
- PL/PHP
- PL/Python
- PL/R (like SPSS)
- PL/Ruby
- PL/Scheme
- PL/sh
- PL/Tcl
- PL/v8 (JavaScript)
- SPI (C)

# Postgres Index Types

- BRIN
- BTree
- Hash
- GIN (generalized inverted index)
- GiST (generalized search tree)
- SP-GiST (space-partitioned GiST)

# Postgres Innovation: Full Text Search

- Supports full text search capabilities in a relational database
- Whole-word, word prefix, *and*, *or*, and *not* searches
- Stemming for 21 languages
- *Pg\_trgm* extension allows search of letter combinations and similarity
- Specialized indexing, operators, and functions
- Full transaction semantics

# Postgres Innovation: Full Text Search

```
EXPLAIN SELECT line
FROM fortune
WHERE to_tsvector('english', line) @@ to_tsquery('pandas');
                                QUERY PLAN
```

```
-----...
Bitmap Heap Scan on fortune  (cost=12.41..94.25 rows=21 width=36)
  Recheck Cond: (to_tsvector('english'::regconfig, line) @@ to_ts...
->  Bitmap Index Scan on fortune_idx_ts  (cost=0.00..12.40 rows...
      Index Cond: (to_tsvector('english'::regconfig, line) @@ t...
```



# Postgres Innovation: NoSQL

- Supports NoSQL capabilities in a relational database
- Mix structured and unstructured data in the same row and query; the best of both worlds
- Specialized indexing, operators, and functions
- Full transaction semantics

# Postgres Innovation: NoSQL

```
EXPLAIN SELECT data->>'last_name'  
FROM friend2  
WHERE data::jsonb @> '{"first_name" : "Jane"}'  
ORDER BY 1;                                QUERY PLAN
```

```
-----  
Sort (cost=24.03..24.04 rows=1 width=139)  
  Sort Key: ((data ->> 'last_name'::text))  
    -> Bitmap Heap Scan on friend2 (cost=20.01..24.02 rows=1 ...  
      Recheck Cond: (data @> '{"first_name": "Jane"}'::jsonb)  
        -> Bitmap Index Scan on friend2_idx (cost=0.00..20.01 .....  
          Index Cond: (data @> '{"first_name": "Jane"}'::js...
```

# Postgres Innovation: Range Types

- Combines start and stop times into a single field
- Allows sophisticated indexing and comparisons
- Allows automatic range overlap prevention

# Postgres Innovation: Range Types

```
EXPLAIN SELECT *  
FROM car_rental  
WHERE time_span @> '2007-08-01 00:00:00'::timestampz;
```

QUERY PLAN

```
-----  
Index Scan using car_rental_idx on car_rental (cost=0.15..8.17..  
  Index Cond: (time_span @> '2007-08-01 00:00:00-04'::timestamp..
```

# Postgres Innovation: Geometric Types

- Handle multi-dimensional data
  - Points
  - Lines
  - Circles
  - Polygons
- Multi-dimensional indexing and operators
- Allows efficient nearest neighbor searches
- Avoids using a separate geometric data store

# Postgres Innovation: Geometric Types

```
EXPLAIN SELECT *  
FROM dart  
ORDER BY location <-> '(50, 50)::point  
LIMIT 2;
```

QUERY PLAN

```
-----  
Limit (cost=0.14..0.33 rows=2 width=20)  
-> Index Scan using dart_idx on dart (cost=0.14..92.14..  
    Order By: (location <-> '(50,50)::point)
```

# Postgres Innovation: GIS

- PostGIS is a full-featured Geographical Information System (GIS)
- Implemented as a extension
- Independent development team and community



<https://postgis.net/>

# Postgres Innovation: GIS

```
SELECT ST_Area(the_geom)/10000 AS hectares  
FROM bc_municipality  
WHERE name = 'PRINCE GEORGE';
```

hectares

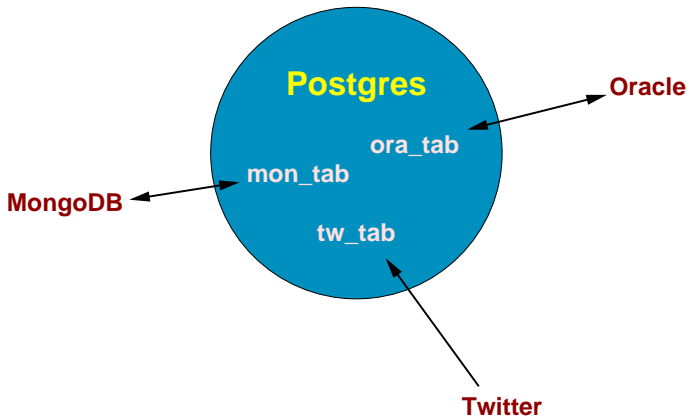
-----  
32657.9103824927



# Postgres Innovation: Foreign Data Wrappers

- 100+ interfaces to foreign data
- Read/write
- Sophisticated push down of joins, sorts, and aggregates

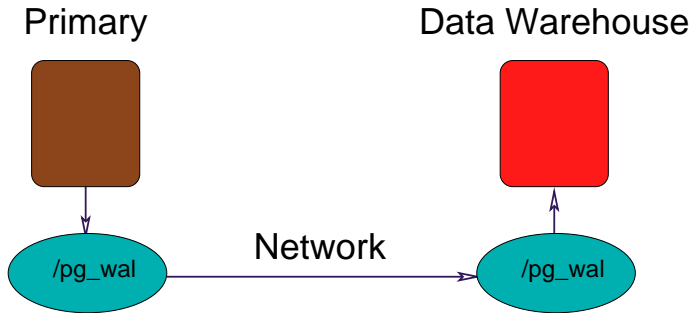
# Postgres Innovation: Foreign Data Wrappers



# Postgres Innovation: Data Analytics

- Aggregates
- Optimizer
- Server-side languages, e.g., PL/R
- Window functions
- Bitmap heap scans
- Tablespaces
- Data partitioning
- Materialized views
- Common table expressions (CTE)
- BRIN indexes
- GROUPING SETS, ROLLUP, CUBE
- Just-in-time compilation (JIT)
- Parallelism
- Sharding (in progress)

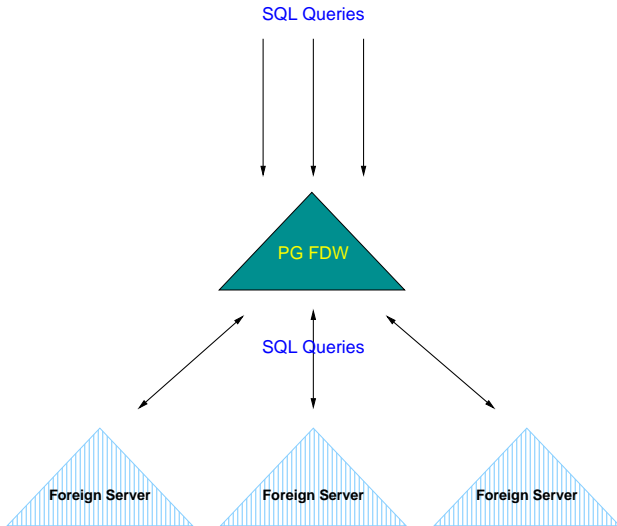
# Postgres Innovation: Data Analytics



# Postgres Innovation: Sharding

- Allows multi-host databases
- Uses existing functionality
  - Partitioning
  - Parallelism
  - Foreign data wrappers
  - Logical replication
- Needs new functionality
  - Global transaction manager
  - Global snapshot manager

# Postgres Innovation: Sharding



## 5. Community Structure



[https://www.flickr.com/photos/tomas\\_vondra/](https://www.flickr.com/photos/tomas_vondra/)

# Community Structure

- BSD license guarantees software will be available forever, including for proprietary use.
- Development and leadership is diversified geographically, culturally, and is multi-company.



# Still Going Strong

- 32 years of development
- 22 years of annual major releases
- ~180 features per major release
- Quarterly minor releases
- Most-loved relational database
  - <https://insights.stackoverflow.com/survey/2018/#technology-most-loved-dreaded-and-wanted-databases>

# PgLife

## PgLife Postgres Community Life

### Users

**General** Re: Pg\_upgrade problem.  
**Other** Re: INCLUDING COMPRESSION  
**Announce** Announcing the release of Apache AGE 0.4.0

### Developers

**Hackers** Re: Converting contrib SQL functions to new style  
**Commit** Doc: Document known problem with Windows collation versions.  
**Versions** **Stable:** 13.2+, 12.6+, 11.11+, 10.16+, 9.6.21+ | **Development:** 14 devel

### External

**Blogs** Jimmy Angelakos: Changing your huge table's data types in production [FOSDEM 2021]  
**News** PostgreSQL Weekly News - April 11, 2021  
**Media** Rockset Enables Real-Time Analytics for MySQL and PostgreSQL - GlobeNewswire  
**Events** PostgresLondon 2021

### IRC (also Slack)

#### **OverCoder:**

**average:** OverCoder: am I reading this right in your image, a programmer is excusing himself about code quality because "i wrote it on a plane"  
<https://irc.overcoder.dev/uploads/Bc3e6f9688e81f8d/image.png> ?

**average:** OverCoder: that's such a braggadocio ..

**average:** i've never seen that before

**doctormon:** It's a faint.

**doctormon:** Sorry, feint ;-)

**average:** aah.. i see

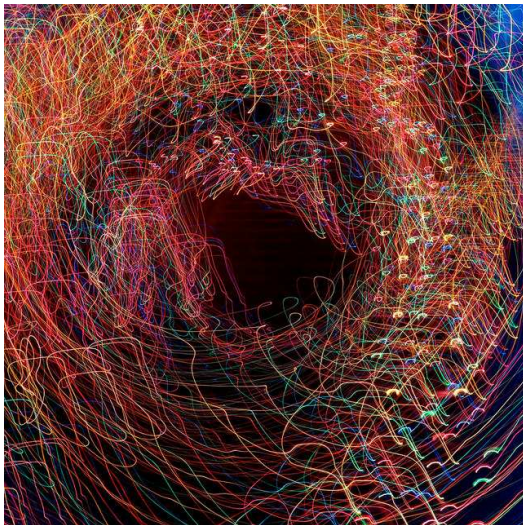
**crunchy\_david:** davidfetter: M as in agento-May

London 02:54 Berlin 03:54 Moscow 04:54 Mumbai 07:24 Beijing 09:54 Tokyo 10:54 Los Angeles 18:54 New York 21:54 São Paulo 22:54  
*Content updates automatically | About | Submit Feedback*

<http://pglife.momjian.us>

<https://momjian.us/main/writings/pgsql/sharding.pdf>

## 6. Conclusion



<https://momjian.us/presentations>

<https://www.flickr.com/photos/pagedooley/>