

# PostgreSQL: Loved, Dreaded or Wanted?

Fábio Telles Rodriguez

# Fábio Telles Rodriguez

- DBA Oracle e PostgreSQL +15 anos
- Colaborador da Comunidade Brasileira de PostgreSQL
- Blog: [savepoint.blog.br](http://savepoint.blog.br)
- Twitter: @telles
- Telegram: @fabio\_telles





# Como as pessoas escolhem suas ferramentas de trabalho?

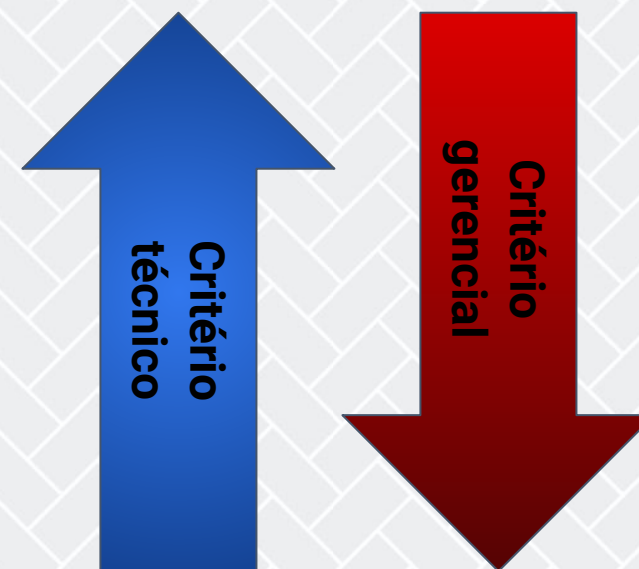
E onde os bancos de dados entram nisso?





# Como escolhemos ferramentas?

- Melhor resolve o seu problema
- Tem o melhor custo / benefício
- Maior familiaridade
- Está na moda
- O que todo mundo usa
- Representa menor risco para o gestor



# Menor risco para o gestor

Figure 1. Magic Quadrant for Operational Database Management Systems



Source: Gartner (October 2018)





# Menor risco para o gestor

- Gestores não gostam de assumir riscos  
“ninguém nunca foi demitido por usar ...”
- Gestores enxergam “market share”
- Avalia empresas e não softwares



# Todo mundo usa



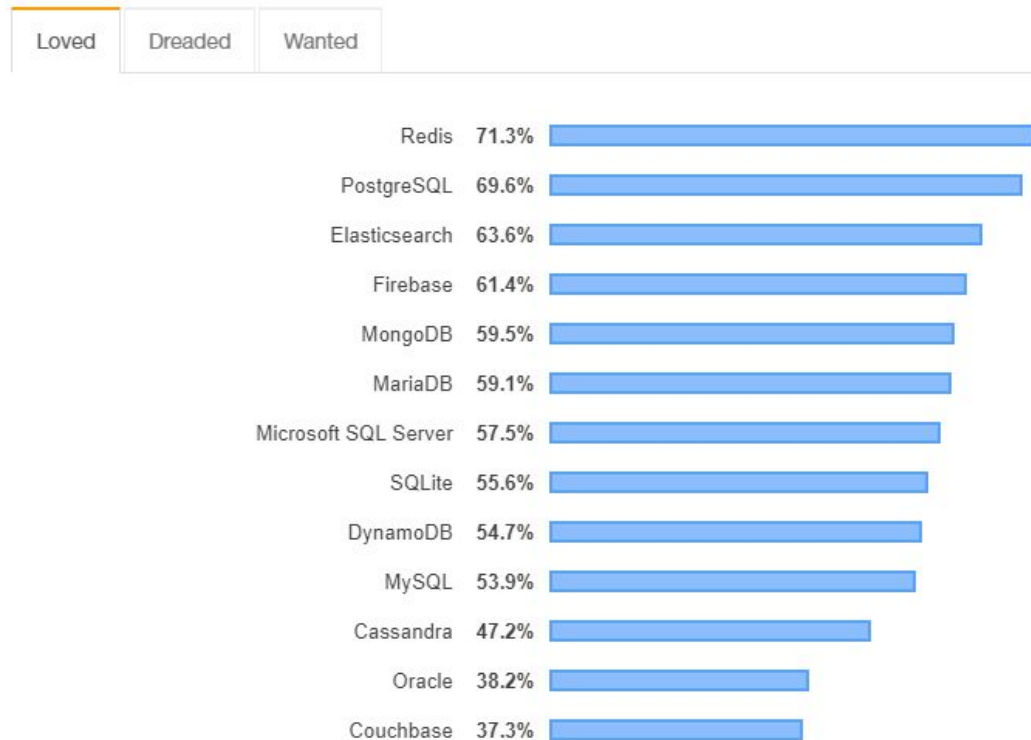


Todo mundo usa



# Está na moda

## Most Loved, Dreaded, and Wanted Databases



# Está na moda

- O banco de dados é uma peça crítica do sistema
- Não se confia seus dados a plataformas que não foram amplamente testadas
- DBAs são em geral conservadores





# Maior familiaridade

## DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



347 systems in ranking, May 2019

Rank			DBMS	Database Model	Score		
May 2019	Apr 2019	May 2018			May 2019	Apr 2019	May 2018
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1285.55	+5.61	-4.87
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1218.96	+3.82	-4.38
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	1072.19	+12.23	-13.66
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	478.89	+0.17	+77.99
5.	5.	5.	MongoDB +	Document	408.07	+6.10	+65.96
6.	6.	6.	IBM Db2 +	Relational, Multi-model ⓘ	174.44	-1.61	-11.17
7.	↑ 8.	↑ 9.	Elasticsearch +	Search engine, Multi-model ⓘ	148.62	+2.62	+18.18
8.	↓ 7.	↓ 7.	Redis +	Key-value, Multi-model ⓘ	148.40	+2.03	+13.06
9.	9.	↓ 8.	Microsoft Access	Relational	143.78	-0.87	+10.67
10.	↑ 11.	10.	Cassandra +	Wide column	125.72	+2.11	+7.89



# Melhor custo/benefício

- **Benchmarks** são caros e complexos
- SGDBs **proprietários** não permitem publicar testes sem sua prévia autorização
- **TPC** usa **transações/\$** para diferentes cenários como OLTP, BI e WEB
- SGDBs livres nunca lançaram testes oficiais no TPC
- O TPC foi praticamente **abandonado** nos últimos anos



# Melhor custo/benefício

TPC-E Advanced Sorting Results

Sponsor	System	Performance (TpsE)	Price/TpsE	System Availability	Date Submitted	DB Software Name
	Lenovo ThinkSystem SR650	7,012	90.99 USD	4/17/2019	3/29/2019	Microsoft SQL Server 2017 Enterprise Edition
	Lenovo ThinkSystem SR650	6,779	92.49 USD	9/10/2018	9/5/2018	Microsoft SQL Server 2017 Enterprise Edition
	FUJITSU Server PRIMERGY RX2540 M4	6,606	92.85 USD	3/31/2018	3/31/2018	Microsoft SQL Server 2017 Enterprise Edition
	Lenovo ThinkSystem SR950	11,357	98.83 USD	11/6/2017	11/1/2017	Microsoft SQL Server 2017 Enterprise Edition
	Lenovo ThinkSystem SR650	6,598	93.48 USD	10/19/2017	6/27/2017	Microsoft SQL Server 2017 Enterprise Edition
	FUJITSU Server PRIMERGY RX4770 M3	8,796	116.62 USD	7/31/2016	7/12/2016	Microsoft SQL Server 2016 Enterprise Edition
	Lenovo System x3850 X6	9,068	139.85 USD	7/31/2016	5/31/2016	Microsoft SQL Server 2016 Enterprise Edition





# Melhor resolve o seu problema

OLTP

BI



# Melhor resolve o seu problema

OLTP

BI



# Melhor resolve o seu problema

OLTP



BI

teradata.





# Então estamos todos errados?



# Como escolhemos ferramentas?

## Critério inicial

## Critério sugerido

Melhor resolve seu problema	Atende aos requisitos funcionais
Maior familiaridade	A curva de aprendizado é aceitável para a sua equipe
Está na moda	Aderente aos novos padrões de mercado
O que todo mundo usa	Possui um ecossistema consistente
Representa o menor risco para o gestor	Já foi adotado com sucesso em cenários semelhantes



“Eu uso a ferramenta que  
paga as minhas contas”

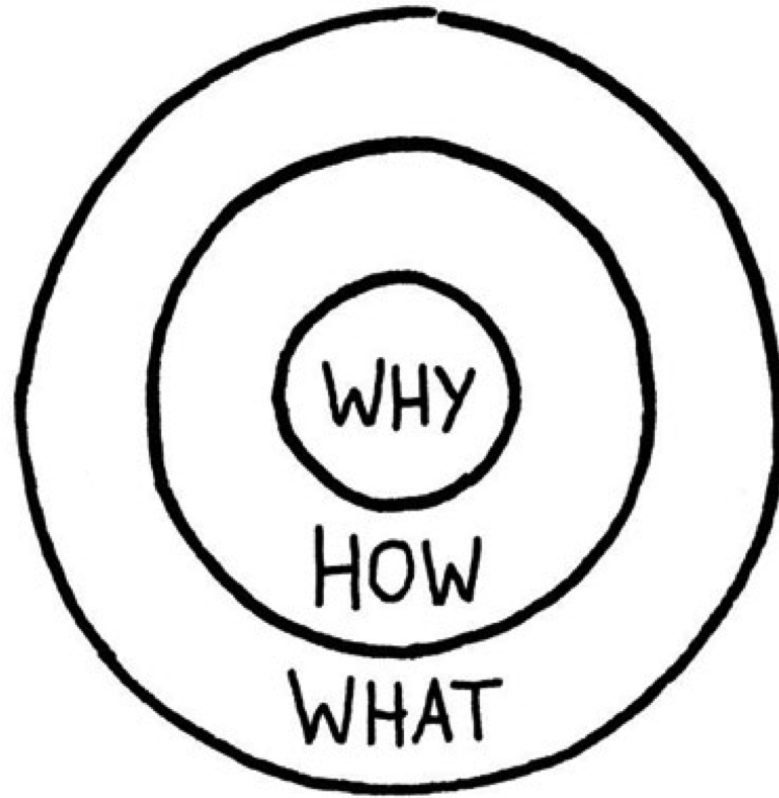




# Database Golden Circle

O que move as pessoas?

# Golden Circle



Simon Sinek Ted Talk: <https://www.youtube.com/watch?v=ayaO26BmkPk>



# What

O que faz um gerenciador de banco de dados?

Operações com dados:

- Guardar
- Disponibilizar
- Organizar
- Manipular





# How

- Modelo de desenvolvimento
- Ciclo de lançamento de funcionalidades
- Modelo de negócios
- Relação com os clientes
- Ecossistema ao redor
- Capacidade de resolver problemas



# Why

O que motiva o seu surgimento?

- Visão histórica
- Motivação inicial
- Capacidade de transformar motivação em modelo de negócios





# Visão histórica

## Oracle:

- Início em 1977
- Clone do SystemR, futuro DB2
- A IBM esconde as especificações do SystemR, mas ainda assim o Oracle assume a liderança do mercado junto com o Ingres no início dos anos 80

## SQL Server:

- Início em 1989
- Versão lançada pela Microsoft do Sybase rodando em Windows
- O Sybase era baseado no Ingres

## MySQL:

- Início em 1994
- Clone do mSQL criado para armazenar dados de dispositivos de rede
- Fez muito sucesso em aplicações simples com o LAMP, motor da WEB 2.0



# E muita coisa mudou de lá para cá...

SHUTDOWN ABORT the last Oracle database running Amazon Fulfillment!

Traduzir Tweet

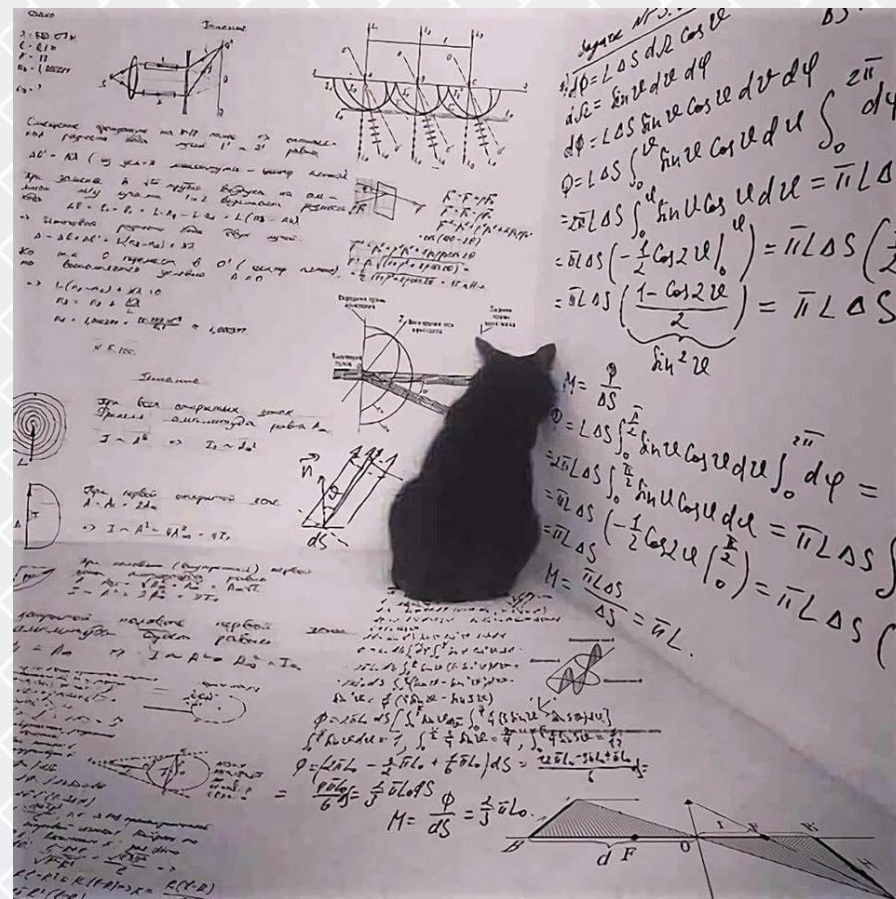


11:00 - 29 de mar de 2019





# Cenário atual



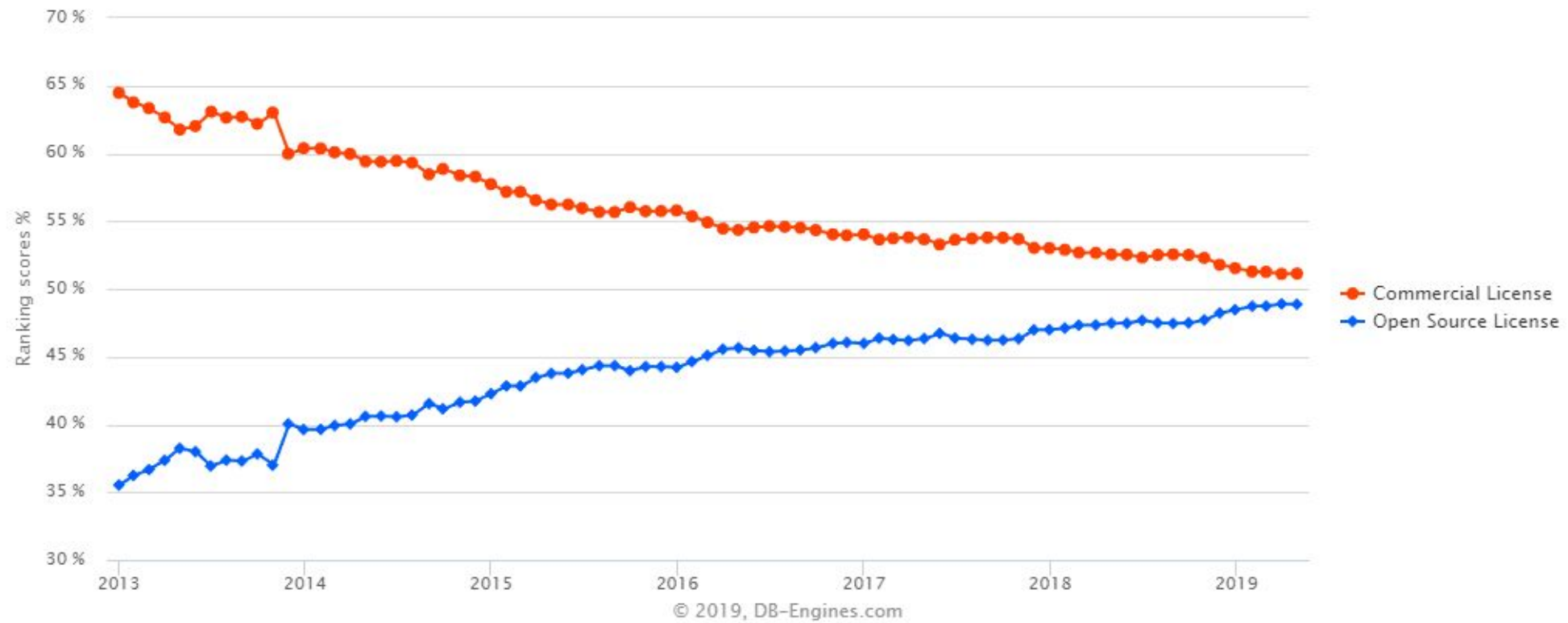
# Cenário atual

- Cloud
- DevOps
- Crescimento de infra baseada em Software Livre
- Microsserviços
- Persistência poliglota



# Cenário atual

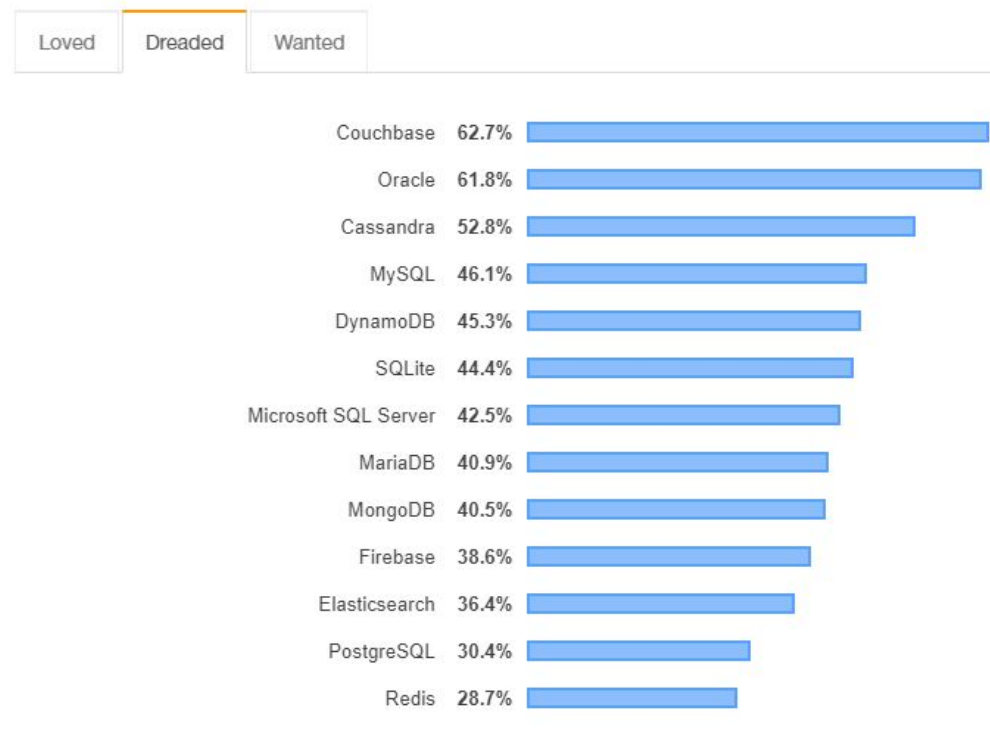
## Popularity trend





# Cenário atual

## Most Loved, Dreaded, and Wanted Databases



# Cenário atual



# Postgres Golden Circle

Why people love Postgres?



# Why Postgres?

1986:

- Resposta às limitações o **Ingres**
- **Reescrita** a partir do zero
- Incorporar elementos de **orientação a objeto**
- Foco na **extensibilidade**
- Cria o “rule system” precursor das hoje conhecidas “**constraints**”
- Mantém a linguagem **QUEL** criada no Ingres
- Mantém a **licença BSD** utilizada em Berkeley





# Postgres → PostgreSQL

1992:

- Michael Stonebraker sai de Berkeley e funda a **Illustra**

1994:

- O projeto é encerrado na universidade de **Berkeley**

1995:

- Dois **estudantes** de graduação de Berkeley **retomam** o projeto;
- Adota a linguagem SQL;

1996:

- Deixa definitivamente a universidade de Berkeley e passa a ser mantido pela **comunidade**





| Postgres inner circle (why)

Why Postgres  
is so cool?



# Postgres inner circle

- Do the **right thing**
- Write a **clean** and **beautiful** code
- Focus on **reliability** and **maintainability**
- **Open** and **permissive** licence





# Postgres inner circle



| Postgres middle circle (how)

How Postgres  
is so cool?





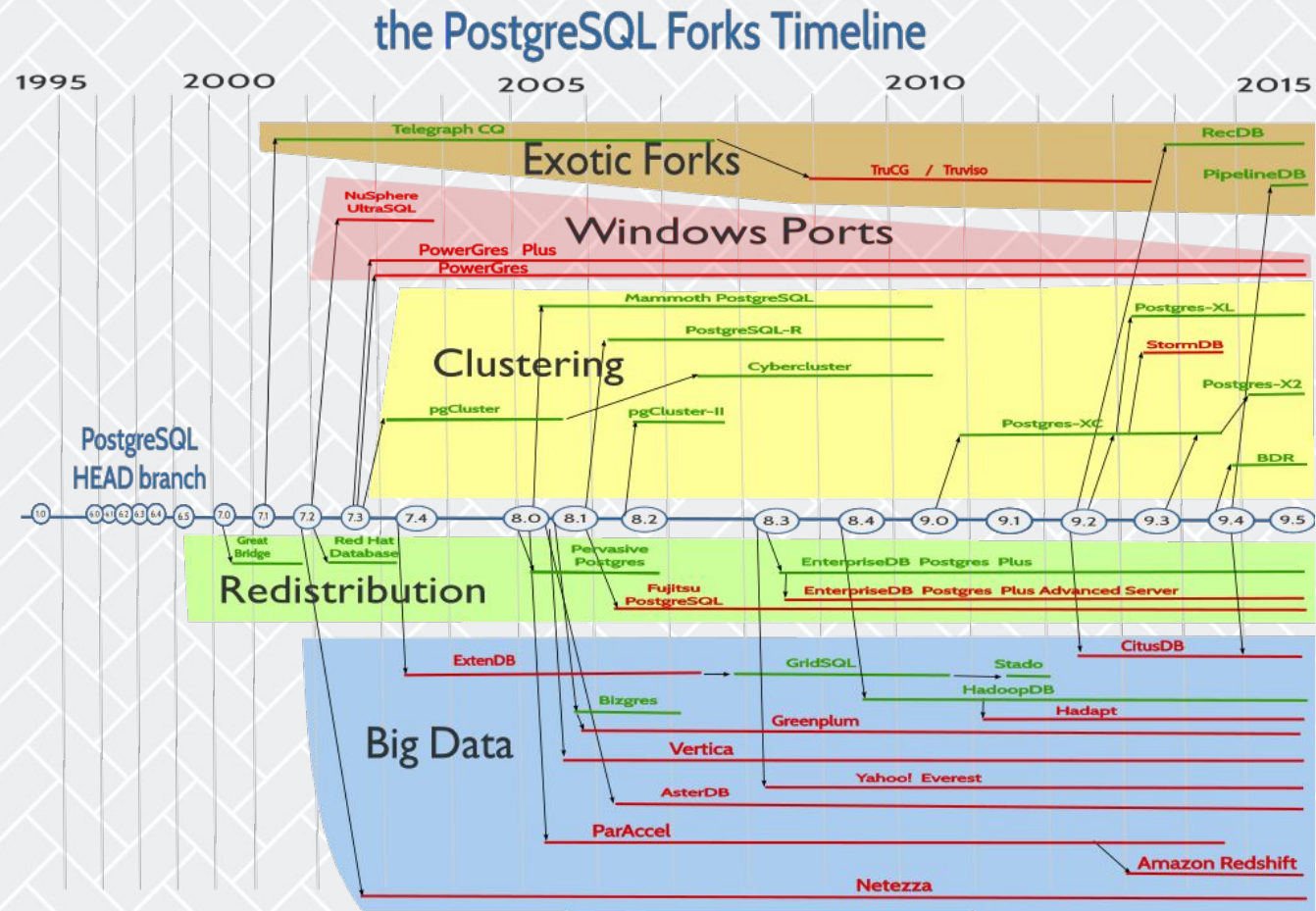
# Desenvolvimento aberto

- Todas as listas de discussão são abertas
- Documentação do processo de desenvolvimento em [https://wiki.postgresql.org/wiki/Developer\\_and\\_Contributor\\_Resources](https://wiki.postgresql.org/wiki/Developer_and_Contributor_Resources)
- Ferramentas abertas: <https://doxygen.postgresql.org>
- Commit fests: <https://commitfest.postgresql.org>





# Postgres middle circle (how)



# Forks

*[https://wiki.postgresql.org/wiki/PostgreSQL\\_derived\\_databases](https://wiki.postgresql.org/wiki/PostgreSQL_derived_databases)*

- AgensGraph (orientado a grafos)
- Aurora (AWS)
- BDR (replicação multimaster)
- CitusDB (BI)
- EnterpriseDB (compatibilidade com Oracle)
- HadoopDB (cluster shared nothing)
- PipelineDB (streaming)
- PostgresX2 (cluster)
- Pivotal (BI)
- Redshift (BI)
- ToroDB (document model)





# Forks

*[https://wiki.postgresql.org/wiki/PostgreSQL\\_derived\\_databases](https://wiki.postgresql.org/wiki/PostgreSQL_derived_databases)*

- AgensGraph (orientado a grafos)
- Aurora (AWS)
- BDR (replicação multimaster)
- CitusDB (BI)
- EnterpriseDB (compatibilidade com Oracle)
- HadoopDB (cluster shared nothing)
- PipelineDB (streaming)
- PostgresX2 (cluster)
- Pivotal (BI)
- Redshift (BI)
- ToroDB (document model)



# Desenvolvimento aberto

Empresas que patrocinam o PostgreSQL:





# Desenvolvimento aberto

## Pessoas que contribuíram no desenvolvimento da última versão:

Abhijit Menon-Sen / Adam Bielanski / Adam Brightwell / Adam Brusselback / Aditya Toshniwal / Adrián Escoms / Adrien Nayrat / Akos Vandra / Aleksander / Alekseev / Aleksandr Parfenov / Alexander Korotkov / Alexander Kukushkin / Alexander Kuzmenkov / Alexander Lakhin / Alexandre Garcia / Alexey Bashtanov / Alexey Chernyshov / Alexey Kryuchkov / Alik Khilazhev / Álvaro Herrera / Amit Kapila / Amit Khandekar / Amit Langote / Amul Sul / Anastasia Lubennikova / Andreas Joseph Krogh / Andreas Karlsson / Andreas Seltenreich / André Hänsel / Andrei Gorita / Andres Freund / Andrew Dunstan / Andrew Fletcher / Andrew Gierth / Andrew Grossman / Andrew Krasichkov / Andrey Borodin / Andrey Lizenko / Andy Abelisto / Anthony Bykov / Antoine Scemama / Anton Dignös / Antonin Houska / Arseniy Sharoglazov / Arseniy Sher / Arthur Zakirov / Ashutosh Bapat / Ashutosh Sharma / Ashwin Agrawal / Asim Praveen / Atsushi Torikoshi / Badrul Chowdhury / Balazs Szilfai / Basil Bourque / Beena Emerson / Ben Chobot / Benjamin Coutu / Bernd Helmle / Blaz Merela / Brad DeJong / Brent Dearth / Brian Cloutier / Bruce Momjian / Catalin Iacob / Chad Trabant / Chapman Flack / Christian Duta / Christian / Ullrich / Christoph Berg / Christoph Dreis / Christophe Courtois / Christopher Jones / Claudio Freire / Clayton Salem / Craig Ringer / Dagfinn Ilmari Mannsåker / Dan Vianello / Dan Watson / Dang Minh Huong / Daniel Gustafsson / Daniel Vérité / Daniel Westermann / Daniel Wood / Darafei Praliaskouski / Dave Cramer / Dave Page / David Binderman / David Carlier / David Fetter / David G. Johnston / David Gould / David Hinkle / David Pereiro Lagares / David Rader / David Rowley / David Steele / Davy Machado / Dean Rasheed / Dian Fay / Dilip Kumar / Dmitriy Sarafannikov / Dmitry Dolgov / Dmitry Ivanov / Dmitry Shalashov / Don Seiler / Doug Doole / Doug Rady / Edmund Horner / Eiji Seki / Elvis Pranskevichus / Emre Hasegeli / Erik Rijkers / Erwin Brandstetter / Etsuro Fujita / Euler Taveira / Everaldo Canuto / Fabien Coelho / Fabrício de Royes Mello / Feike Steenbergen / Frits Jalvingh / Fujii Masao / Gao Zengqi / Gianni Ciolli / Greg Stark / Gunnlaugur Thor Briem / Guo Xiang Tan / Hadi Moshayedi / Hailong Li / Haribabu Kommi / Heath Lord / Heikki Linnakangas / Hugo Mercier / Igor Korot / Igor Neyman / Ildar Musin / Ildus Kurbangaliev / Ioseph Kim / Jacob Champion / Jaime Casanova / Jakob Egger / Jean-Pierre Pelletier / Jeevan Chalke / Jeevan Ladhe / Jeff Davis / Jeff Janes / Jeremy Evans / Jeremy Finzel / Jeremy Schneider / Jesper Pedersen / Jim Nasby / Jimmy Yih / Jing Wang / Jobin Augustine / Joe Conway / John Gorman / John Naylor / Jon Nelson / Jon Wolski / Jonathan Allen / Jonathan S. Katz / Julien Rouhaud / Jürgen Pütz / Justin Pryzby / KaiGai Kohei / Kaïting Chen / Karl Lehenbauer / Keith Fiske / Kevin Bloch / Kha Nguyen / Kim Rose Carlsen / Konstantin Knizhnik / Kuntal Ghosh / Kyle Samson / Kyotaro Horiguchi / Lætitia Avrot / Lars Kanis / Laurenz Albe / Leonardo Cecchi / Liudmila Mantrova / Lixian Zou / Lloyd Albin / Luca Ferrari / Lucas Fairchild / Lukas Eder / Lukas Fittl / Magnus Hagander / Mai Peng / Maksim Milyutin / Maksym Boguk / Mansur Galiev / Marc Dilger / Marco Nenciarini / Marina Polyakova / Mario de Frutos Dieguez / Mark Cave-Ayland / Mark Dilger / Mark Wood / Marko Tiikkaja / Markus Winand / Martín Marqués / Masahiko Sawada / Matheus Oliveira / Matthew Stickney / Metin Doslu / Michael Banck / Michael Meskes / Michael Paquier / Michail Nikolaev / Mike Blackwell / Minh-Quan Tran / Mithun Cy / Morgan Owens / Nathan Bossart / Nathan Wagner / Neil Conway / Nick Barnes / Nicolas Thauvin / Nikhil Sontakke / Nikita Glukhov / Nikolay Shaplov / Noah Misch / Noriyoshi Shinoda / Oleg Bartunov / Oleg Samoilov / Oliver Ford / Pan Bian / Pascal Legrand / Patrick Hemmer / Patrick Kreecker / Paul Bonaud / Paul Guo / Paul Ramsey / Pavan Deolasee / Pavan Maddamsetti / Pavel Golub / Pavel Stehule / Peter Eisentraut / Peter Geoghegan / Petr Jelínek / Petru-Florin Mihancea / Phil Florent / Philippe Beaudoin / Pierre Ducroquet / Piotr Stefaniak / Prabhat Sahu / Pu Qun / QL Zhuo / Rafia Sabih / Rahila Syed / Rainer Orth / Rajkumar Raghuwanshi / Raúl Marín Rodríguez / Regina Obe / Richard Yen / Robert Haas / Robins Tharakan / Rod Taylor / Rushabh Lathia / Ryan Murphy / Sahap Ascii / Samuel Horwitz / Scott Ure / Sean Johnston / Shao Bret / Shay Rojansky / Shubham Barai / Simon Riggs / Simone Gotti / Sivasubramanian Ramasubramanian / Stas Kelvich / Stefan Kaltenbrunner / Stephen Froehlich / Stephen Frost / Steve Singer / Steven Winfield / Sven Kunze / Taiki Kondo / Takayuki Tsunakawa / Takeshi Ideriha / Tatsuo Ishii / Tatsuro Yamada / Teodor Sigaev / Thom Brown / Thomas Kellerer / Thomas Munro / Thomas Reiss / Tobias Bussmann / Todd A. Cook / Tom Kazimiers / Tom Lane / Tomas Vondra / Tomonari Katsumata / Torsten Grust / Tushar Ahuja / Vaishnavi Prabakaran / Vasundhar Boddapati / Victor Drobny / Victor Wagner / Victor Yegorov / Vik Fearing / Vinayak Pokale / Vincent Lachenal / Vitaliy Garnashevich / Vitaly Burovoy / Vladimir Baranoff / Xin Zhang / Yi Wen Wong / Yorick Peterse / Yugo Nagata / Yuqi Gu / Yura Sokolov / Yves Goergen / Zhou Digoal



# One more thing...



| Postgres outer circle (what)

So what in  
Postgres  
is so cool?





# Postgres outer circle (what)

- Confiabilidade  
(Rock solid database)
- Extensibilidade
- Funcionalidades excepcionais
- Comunidade efervescente

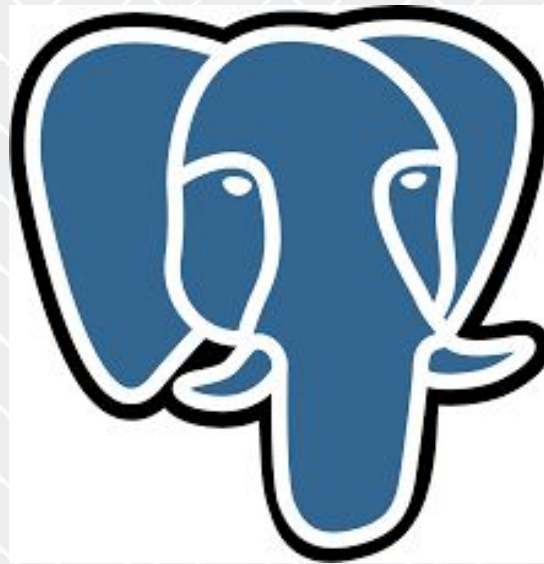




# Confiabilidade



INGRES:  
É Lento mas chega lá



POSTGRES:  
Os elefantes nunca  
esquecem



# Extensibilidade

- Linguagens procedurais: SQL, PL/pgSQL, PL/Perl, PL/Python, PL/TCL, PL/sh, PL/R, PL/Java, PL/Lua, PL/V8 (javascript), etc
- Extensões: contrib, catalogo, pgxn, github, etc
- Foreign Data Wrapper (FDW): SGDBs, web services, etc
- Storage Engines plugáveis





# Funcionalidades excepcionais

- Conformidade com o padrão SQL
- MVCC / Parallel Query / Standby / Logical Replication / Partitioning / Sharding
- Índices: B-Tree, Hash, Gist, GIN, BRIN, etc
- Tipos de dados: enum, geometric, network, Text Search, UUID, XML, JSON, JSONB, arrays, composite types, range types, domain types



THE KISS PRINCIPLE

**KEEP  
IT  
SIMPLE,  
STUPID**





# KISS

- Documentação simples e objetiva
- Sintaxe elegante, consistente e previsível
- Baixa curva de aprendizado



# E tem mais...

- Fácil de instalar (<5min):
  - Repositórios oficiais .rpm e .deb para a maioria das distribuições Linux para todas as versões suportadas do PostgreSQL
- Fácil de configurar:
  - postgresql.conf e pg\_hba.conf
- Fácil de manter e expandir:
  - Monitoramento, backup, criação de réplicas, etc
- Fácil de automatizar
  - Containers, kubernetes, etc
- Ampla cadeia de serviços agregados:
  - Suporte 24/7, DBaaS, treinamentos, etc
- Conformidade com o padrão SQL



# E não menos importante: a comunidade!

- Listas por email: <https://www.postgresql.org/list>
- IRC: <https://www.postgresql.org/community/irc>
- Slack: <https://postgres-slack.herokuapp.com>
- Telegram: <https://t.me/postgresqlbr>
- PGConf.Brasil: <https://www.pgconf.com.br>



# Venha fazer parte da manada!

## PGConf Brasil 2019

- 1, 2 e 3 de agosto
- São Paulo - SP, Hotel Century Flat
- Inscrições abertas
- Grade já disponível
- [www.pgconf.com.br](http://www.pgconf.com.br)





# Dúvidas?





[contato@timbira.com.br](mailto:contato@timbira.com.br)

**TIMBIRA.COM.BR**



/Timbira



/TimbiraBrasil