

Histórico do PostgreSQL

Diego S. Cintra¹

*¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul
(UFMS)
Caixa Postal 549 – 79.070-900 – Campo Grande – MS – Brasil
diego_2337@hotmail.com*

Campo Grande, 26 de fevereiro de 2015

Resumo. Foi-me requisitado a pesquisa e transcrição (resultante da minha análise) sobre o histórico do SGBD PostgreSQL, um dos mais utilizados na atualidade. Hoje em dia, parece algo simples e prático receber um minimundo, construir um modelo relacional e com base nesse instanciar um banco de dados utilizando uma ferramenta SGBD; não só simples, mas essas ferramentas são indispensáveis na atualidade, dada a imensa quantidade de informação produzida por diferentes áreas da sociedade e diferentes sistemas comerciais. Entretanto, toda a complexidade de consultas SQL, procedimentos armazenados e extensibilidade de código via colaboração por Internet não existiam na época do surgimento do PostgreSQL, o que caracterizou para os mantenedores desse sistema um grande desafio ao longo dos anos. Portanto, neste documento, teremos uma visão geral sobre o cenário dos bancos de dados à ocasião do surgimento desse SGBD, suas principais versões e responsáveis, características e principais usuários. Essas informações visam ampliar a compreensão de como se deu o desenvolvimento dessa ferramenta, seus desafios, vantagens e desvantagens e a importância de uma ferramenta SGBD tão extensa capaz de manipular diversas consultas extremamente complexas.

1 Histórico da situação dos bancos de dados relacionais

Por volta da década de 70, a IBM obtinha resultados das pesquisas que havia feito para diminuir os gastos relativos ao armazenamento de informações em discos magnéticos: a linguagem SQL fora criada, e o *System R* havia sido finalizado e iria definir o comportamento de diversos sistemas relacionais posteriores. Em meio a essas criações, muitos artigos que descreviam o comportamento deste sistema relacional haviam sido liberados, o que chamou a atenção de dois cientistas da *University of California at Berkeley* (UCB), Michael Stonebraker e Eugene Wong. Conseguindo financiamento de departamentos de pesquisa militares, um time de desenvolvimento composto de discentes e docentes deu vida ao INGRES, que foi comercializado por um tempo por seus criadores (a partir de 2004, o código fonte do SGBD foi liberado, e esse passou a ser um software livre).

2 Alternativa a bancos de dados relacionais e o *Postgres*

Apesar do relativo sucesso alcançado com o INGRES, Stonebraker retornou à universidade em meados dos anos 80 e começou a trabalhar em outro projeto a fim de resolver problemas inerentes a bancos de dados relacionais (ou seja, o fato desses não conseguirem armazenar “tipos”, ou objetos, como comumente são definidos atualmente). Eis que, deste projeto, surge o Postgres, resultado de cerca de oito anos de trabalho de Stonebraker e seus alunos de graduação, que introduziu a definição de tipos (seu principal objetivo até então), bem como a definição de índices para esses tipos, procedimentos e capacidade de descrever relações. Nesses oito anos, em ordem cronológica, surgiram diversos artigos explicando o funcionamento do SGBD, bem como um protótipo, versões 1, 2 e 3 (cada uma tendo alterações no sistema de regras e a última tendo uma otimização na *engine* de consultas). O retorno do projeto foi imenso, com uma extensiva comunidade que demandava cada vez mais funcionalidades. Portanto, a partir de 1993 (e a versão 4), a UCB abandonou o projeto.

3 Continuação de desenvolvimento e suporte a SQL

Apesar do abandono do SGBD, esse possuía uma licença BSD (que permite com que o código-fonte seja aberto), portanto seu desenvolvimento pôde ser continuado. Jolly Chen e Andrew Yu, dois estudantes orientados por Michael, substituíram a linguagem POSTQUEL de consultas (utilizada no INGRES) por SQL, distribuindo esse SGBD pela recém introduzida Internet, norteando um novo caminho para a vida do software: um grupo de desenvolvedores fora do ambiente da UCB viu o potencial do Postgres, e dedicaram-se extensivamente a adicionar novas funcionalidades e mantê-lo gratuitamente, sendo constantemente melhorado de forma colaborativa via Internet. A partir do meado dos anos 90 em diante, com o lançamento do

Postgres 6.0, o nome dele foi alterado para PostgreSQL (como é conhecido na atualidade) e hoje conta com uma imensa documentação e um time de desenvolvimento e manutenção composto de diversos colaboradores (voluntários) pelo mundo.

4 Versões do PostgreSQL

Com as novas versões lançadas do PostgreSQL, algumas funcionalidades muito importantes foram adicionadas:

- Na versão 6.0, o grande destaque se deu ao MVCC (do inglês *MultiVersion Concurrency Control*, ou controle de concorrência multiversão), que garante acesso simultâneo a uma determinada informação sem introduzir inconsistência nessas informações, através da criação de uma cópia do dado original que é acessada pelos processos (ou leitores dos bancos de dados). Outras adições incluíram novos tipos de dados (como data e hora e tipos geométricos), importantes características SQL (como subselect, chaves primárias, *casting*, entre muitas outras) e um aumento na velocidade de processamento de consultas;
- Já na versão 7.0, houve a implementação de suporte a IPv6, outer joins (como parte da implementação da sintaxe SQL92), adição de novas medidas de segurança, o WAL (*Write Ahead Log*, que faz com que todas as mudanças aplicadas a um banco de dados sejam primeiramente escritas em um *log*, antes de serem aplicadas), entre muitas outras;
- A versão 8.0 introduziu o *Point-in-time Recovery* (permitindo retornar ao ponto de falha exato, ao invés de ter de utilizar um *backup* ou um servidor de replicação), suporte a nativamente funcionar como um servidor no Windows e melhorias na troca de buffers, aumentando a eficiência e velocidade de consultas e alterações em tuplas;
- A última versão, 9.0 (lançada em 2010), introduziu suporte ao Windows 64 bits, novas características de segurança, permissões de administração mais condensadas e simples e novas funções de *triggers*, como exclusão condicional.

5 Características e utilização na atualidade

Atualmente, o PostgreSQL é amplamente utilizado por pequenas e grandes companhias e projetos, como CISCO, Skype, Fujitsu, SourceForge, LAMP, Debian, e diversos departamentos governamentais estadunidenses. Visando sempre a extensibilidade e o cumprimento dos padrões ANSI/ ISO SQL, seu destaque em comparação a outros SGBDs se dá pelo suporte completo a transações confiáveis – respeitando as propriedades ACID –, a capacidade simples de poder estender procedimentos através de procedimentos armazenados e seu rígido esquema de segurança. Alguns números mostram as principais vantagens do SGBD em questão:

- Tamanho máximo de um campo em uma tupla: 1 GB;
- Tamanho máximo de uma tupla: 1.6 TB;
- Tamanho máximo de uma tabela: 32 TB;
- Tamanho máximo de um banco de dados: não há.

Apesar de tamanha praticidade e extensibilidade, o PostgreSQL não possui uma popularidade tão grande quanto outros SGBDs (como o MySQL), o que dificulta encontrar provedores ou hosts que oferecem administração de instâncias dele. Para questões de velocidade e bancos de dados minimalistas, esse SGBD também não é indicado, visto que o foco é a complexidade de procedimentos (que podem ser customizados, como mencionado previamente). Entretanto, esse é um dos mais poderosos e gratuitos SGBDs disponíveis na atualidade, e indispensável para aqueles que precisam de armazenamento de dados e recuperações complexas.

6 Fonte

<http://www.postgresql.org/>