

Artigo do Professor: Andre Rodrigo Sanches

Disciplina: Fundamentos de Armazenamento e Manipulação de Dados

Introdução

Igualmente a muitas tecnologias na computação industrial, os fundamentos de bancos de dados relacionais surgiram na empresa IBM, nas décadas de 1960 e 1970, através de pesquisas de funções de automação de escritório. Foi durante um período da história na qual empresas descobriram que estava muito custoso empregar um número grande de pessoas para fazer trabalhos como armazenar e indexar (organizar) arquivos. Por este motivo, valia a pena os esforços e investimentos em pesquisar um meio mais barato e ter uma solução mecânica eficiente.

Muitas pesquisas foram conduzidas durante este período, cujos modelos hierárquicos, de rede e relacionais e outros modelos foram descobertos, bem como muita tecnologia utilizada hoje em dia.

Em 1970 um pesquisador da IBM - Ted Codd - publicou o primeiro artigo sobre bancos de dados relacionais. Este artigo tratava sobre o uso de cálculo e álgebra relacional para permitir que usuários não técnicos armazenassem e recuperassem grande quantidade de informações. Codd visionava um sistema onde o usuário seria capaz de acessar as informações através de comandos em inglês, onde as informações estariam armazenadas em tabelas.

Devido à natureza técnica deste artigo e a relativa complicação matemática, o significado e proposição do artigo não foram prontamente realizados. Entretanto ele levou a IBM a montar um grupo de pesquisa conhecido como *System R* (Sistema R).

O projeto do Sistema R era criar um sistema de banco de dados relacional o qual eventualmente se tornaria um produto. Os primeiros protótipos foram utilizados por muitas organizações, tais como MIT Sloan School of Management (uma escola renomada de negócios norte-americana). Novas versões foram testadas com empresas aviação para rastreamento do manufaturamento de estoque.

Eventualmente o Sistema R evoluiu para SQL/DS, o qual posteriormente tornou-se o DB2. A linguagem criada pelo grupo do Sistema R foi a *Structured Query Language* (SQL) - Linguagem de Consulta Estruturada). Esta linguagem tornou-se um padrão na indústria para bancos de dados relacionais e hoje em dia é um padrão

ISO (*International Organization for Standardization*). A ISO é a Organização Internacional de Padronização.

Os primeiros do mercado

Mesmo a IBM sendo a companhia que inventou o conceito original e o padrão SQL, eles não produziram o primeiro sistema comercial de banco de dados. O feito foi realizado pela Honeywell Information Systems Inc., cujo sistema foi lançado em junho de 1976. O sistema era baseado em muitos princípios do sistema que a IBM concebeu, mas foi modelado e implementado fora da IBM.

O primeiro sistema de banco de dados construído baseado nos padrões SQL começaram a aparecer no início dos anos 80 com a empresa Oracle através do Oracle 2 e depois com a IBM através do SQL/DS, servindo como sistema e repositório de informações de outras empresas.

Estes sistemas somente nasceram a partir da insistência de um jornal técnico em utilizar BNF para SQL e este jornal publicou tal artigo. BNF é o conjunto de sintaxes de linguagem de computador que explica exatamente como cada comando interage com os outros comandos e o que pode ou não ser realizado, como os comandos são formados em assim por diante. Por causa da publicação deste artigo, empresas puderam utilizá-lo para modelar seus próprios sistemas, os quais seriam 100% compatíveis com o sistema da IBM.

Evolução

O *software* de banco de dados relacionais foi sendo refinado durante a década de 80. Isso deveu-se ao **feedback** (retorno) que os usuários destes sistemas faziam, devido ao desenvolvimento de sistemas para novas indústrias e ao aumento do uso de computadores pessoais e sistemas distribuídos.

Desde sua chegada, os bancos de dados têm tido aumento nos dados de armazenamento, desde os 8 MB (*Megabytes*) até centenas de *Terabytes* de dados em listas de e-mail, informações sobre consumidores, sobre produtos, vídeos, informações geográficas, etc.. Com este aumento de volume de dados, os sistemas de bancos de dados em operação também sofreram aumento em seu tamanho.

Um dos projetos mais ambiciosos de banco de dados está ainda sob construção no CERN. A idéia é criar um banco de dados distribuído com

a capacidade de armazenamento de *Hexabytes* (1 *Hexabyte* = 1,000 *Petabytes* = $1 * 10^{18}$ *Bytes*) de dados e ficará pronto em 2005.

O padrão SQL passou da IBM para a ANSI (*American National Standards Institute*) - Instituto Nacional Americano para Padrões - e para a ISO, os quais formaram um grupo de trabalho para continuar o desenvolvimento. Este desenvolvimento ainda acontece com outras novas versões dos padrões definidos.

Orientação a Objetos

Em meados da década de 80 tornou-se óbvio que existiam várias áreas onde bancos de dados relacionais não eram aplicáveis, por causa dos tipos de dados envolvidos. Estas áreas incluíam medicina, multimídia e física de energia elevada, todas com necessidades de flexibilidade em como os dados seriam representados e acessados.

Este fato levou ao início de pesquisas em bancos de dados orientados a objetos, os quais os usuários poderiam definir seus próprios métodos de acesso aos dados e como estes seriam representados e acessados. Ao mesmo tempo, linguagens de programação orientadas a objetos (*Object Oriented Programming - POO*) tais como C++ começaram a surgir na indústria.

No início de 1990, temos a aparição do primeiro Sistema de Gerenciamento de Banco de Dados Orientado a Objetos, através da companhia Objectivity. Isso permitiu com que usuários criassem sistemas de banco de dados para armazenar resultados de pesquisas como o CERN (maior laboratório que trabalha com partículas físicas em pesquisas nucleares - europeu) e SLAC (Centro de Aceleração Nuclear - norte-americano), para mapeamento de rede de provedores de telecomunicações e para armazenar registros médicos de pacientes em hospitais, consultórios e laboratórios.

Introdução aos SGBD

Um sistema de gerenciamento de banco de dados SGBD consiste em uma coleção de dados inter-relacionados e em um conjunto de programas para acessá-los. Um conjunto de dados, normalmente referenciado como *banco de dados*, contém informações sobre uma empresa particular, por exemplo. O principal objetivo de um SGBD é

prover um ambiente que seja adequado e eficiente para recuperar e armazenar informações de banco de dados.

Os sistemas de banco de dados são projetados para gerenciar grandes grupos de informações. O gerenciamento de dados envolve a definição de estruturas para armazenamento de informação e o fornecimento de mecanismos para manipulá-las. Além disso, o sistema de banco de dados precisa fornecer segurança das informações armazenadas, caso o sistema dê problema, ou contra tentativas de acesso não-autorizado. Se os dados devem ser divididos entre diversos usuários, o sistema precisa evitar possíveis resultados anômalos.

A importância das informações na maioria das organizações e o conseqüente valor dos bancos de dados têm orientado o desenvolvimento de um grande corpo de conceitos e técnicas para o gerenciamento eficiente dos dados.

Cenário de problemas

Imaginemos parte de um banco que mantenha informações sobre todos os clientes e contas de poupança em sistemas de arquivos permanentes no banco. O sistema tem ainda uma quantidade de programas aplicativos que permitem ao usuário manipular os arquivos, incluindo:

- Um programa para debitar ou creditar em uma nova conta;
- Um programa para adicionar uma nova conta;
- Um programa para calcular o saldo de uma conta;
- Um programa para gerar posicionamentos mensais.

Esses programas aplicativos foram desenvolvidos por programadores em resposta às necessidades da organização bancária. Novos programas de aplicação são adicionados ao sistema à medida que as necessidades aparecem. Logo, mais arquivos e mais programas são acrescentados ao sistema.

O típico sistema de processamento de arquivos acima é suportado por um sistema operacional convencional. Registros permanentes são guardados em diversos arquivos, e em uma série de diferentes programas aplicativos é escrita para extrair e adicionar registros aos arquivos apropriados. Este esquema tem uma série de desvantagens:

- **Redundância e inconsistência de dados:** muitos programadores diferentes e programas implementados em linguagens diferentes podem gerar arquivos de formatos diferentes. Informações podem estar duplicadas em diversos

lugares. Gera inconsistência pois estas cópias podem estar com valores diferentes.

- **Dificuldade no acesso aos dados:** um diretor deseja a lista de todos os clientes que moram na cidade de CEP 34863. Ou ele extrai manualmente esta informação de uma lista de clientes ou pede a um programador escrever um programa que aplicativo. Suponha mais tarde que o mesmo diretor deseje uma lista com os clientes com mais de \$10000. Tal lista não existe e novamente o diretor tem as duas opções.
- **Isolamento dos dados:** como os dados estão espalhados, em arquivos separados e com formatos diferentes, é difícil escrever novos programas aplicativos para recuperar os dados adequados.
- **Anomalias de acesso concorrente:** para aperfeiçoar o desempenho geral do sistema e obter tempo de reposta mais rápido, deixamos que múltiplos clientes acessem e atualizem os dados simultaneamente. Isso gera dados inconsistentes. Exemplo: dois clientes sacarem dinheiro de uma mesma conta corrente.
- **Problemas de segurança:** nem todo usuário do sistema de BD deve ter acesso a todos os dados. Por exemplo: o RH pode ter acesso às informações cadastrais dos clientes, mas não aos valores de conta corrente. Se novos programas aplicativos forem adicionados, é difícil assegurar tais restrições de segurança.
- **Problemas de integridade:** os valores dos dados armazenados necessitam satisfazer certas restrições. Por exemplo, o saldo nunca estar abaixo de \$25. Estas restrições podem estar contidas nos programas aplicativos, mas quando novas restrições forem adicionadas, é difícil de alterar estes programas.

Por estes motivos faz-se necessária uma abordagem relacionada à aplicação de um SGBD, a fim de evitar os problemas descritos.