ENTENDIMENTO E ANÁLISE TEÓRICO SOBRE BANCO DE DADOS RELACIONAIS E NÃO ESTRUTURADOS

Adriano Cunha de Santana (81) 99877-3504

|  |  |
| --- | --- |
| Pedra Bruta Decorativa - Sodalita em Promoção | Ofertas na Americanas | PODCAST: A Sabedoria da Natureza - Nova Acrópole  **Um Pouco de Filosofia...**  A informação está em toda parte, em todas as coisas, explicita ou implicitamente integrada a cada elemento com o qual interagimos. Está na natureza e no funcionamento de tudo, a informação é a matéria prima por trás da matéria. Está no DNA, no átomo, na molécula, na proteína, nas ligações iônicas, no macro e no microcosmo. Tudo possui uma quantidade de informação esperando por ser extraída, codificada, apropriadamente.  Como algo bruto e natural, precisa ser observado, pensado, compreendido para que se torne algo reutilizável, prático, útil para o crescimento humano e social.  Quando nos apropriamos da informação passamos a chamá-la de dados. E é neste estado que a informação ganha seu papel em nossas vidas, como dado que representa algo com o qual podemos interagir, reproduzir, recuperar, salvar e segundo nossas necessidades: editar e reeditar!  Tudo o que é abstraído ganha forma quando é documentado, e algo está documentando quando é armazenado em nosso “banco de dados”, seja nossa memória, seja nossos SSD’s. e é nesse contexto que o “banco de dados” se torna primordial para toda e qualquer tecnologias. |
| O que é um banco de dados? O que toda empresa deveria saber - MadeFY  Imagem de: O que é SQL e para que ele serve? | Banco de dados conceito de Nuvem de palavras - Foto de stock de Analisar royalty-free  **Dados, Técnicas, Ferramentas...**  A cada ano a quantidade de dados que guardam informação processada, sobre tudo o que se pode imaginar, cresce exponencialmente.  Para regular esse universo, várias ferramentas tem sido desenvolvidas, evoluindo a intereação e a integração entre os dados em si e entre estes e nós. Sendo assim, podemos iniciar breve análise das tecnologias voltadas para gestão da informação através de recursos de TI.  **SQL** *significa “Structured Query Language”, ou “Linguagem de Consulta Estruturada”, em português[[1]](#footnote-1).* Para interagir de uma forma universal e homogenea foi preciso desenvolver uma linguagem especifica, estruturada para acessar, manipular e resgatar os dados de maneira diversificada e eficiente. |
| Exemplos de bancos de dados SQL mais populares | **A Palavra Chave é ESTRUTURADO**  SQL-based database explained  A principal característica de alguns bancos de dados cuja base de acesso e manipulação é o SQL são seus aspectos estruturais, com base matematica e de forte relação entre suas partes.  Fundamentado na teoria dos conjuntos, matrizes e outros recursos matemáticos, os dados armazenados sob esta estrutura estão “mapeados” por relações fortes e fracas, multiplas ou únicas e com uma diversidades de apresentação que atende a maioria das necessidades de interrelação com os dados.  **O que é um banco de dados relacional?**  *Por bancos de dados relacionais normalmente se entendem bancos de dados que armazenam dados em um formato tabular estruturado (usando linhas e colunas)[[2]](#footnote-2)*. É através dessas relações que as tabelas formam uma base comum, representante de um contexto e de um minimundo representativo. |
|  | **NoSQL – Nem só de estrutura viverás...**  *Bancos de dados NoSQL são criados para modelos de dados específicos e têm esquemas flexíveis para a criação de aplicativos modernos. Os bancos de dados NoSQL são amplamente reconhecidos por sua facilidade de desenvolvimento, funcionalidade e performance em escala. Esta página inclui recursos para ajudar você a compreender melhor os bancos de dados NoSQL e a começar a usá-lo[[3]](#footnote-3)*.  Ou seja, alguns tipos de informações precisam gerar os dados de uma maneira tão flexível que se torna imperativo não haver uma relação rígida, estruturalmente fixa e predefina. É neste subtipo que os bancos *NoSQL* encontram seu lugar ao sol.  E para dar conta do recado, tres categorias se apresentam:   1. **Chave-valor:** particionáveis, escalabilidade horizontal e alcance raramente entregue por outros tipos de bancos. Aplicados em desenvolvimento de jogos, publicidade IoT. O Amazon DynamonDB é um exemplo deste tipo de banco. 2. **Documento:** intuitivo, orientado a objetos/documentos JSON, muito eficiente, flexível e prático. Permite grande homogeneidade entre código e dados, já que os dados são armazenados em modelos altamente integrado com a ferramenta de desenvolvimento. Semiestruturado, escalável segundo as necessidades dos sistemas. Exemplo deste tipo Mongo DB. 3. **Gráficos:** quando o aplicativo e os dados precisam de alto acoplamento a melhor indicação é este tipo de banco. Podemos visualizar este cenário típico em redes sociais, mecanismo de merchandises, análise de perfis fraudulentos e grafos de conhecimento científico. Os mais populares são Neo4j e Giraph. 4. **Em Memória:** volátil, confiável, rápido e seguro. Tempos de resposta extremamente rápido. Jogos e redes sociais como o Tinder são grandes usuários deste tipo de armazenamento de dados em canche. Como exemplo temos o MemoryDB que armazena em memória um minimundo específico que permitem oferecer baixa latência, alta taxa de transferência servindo como fornecedor de dados primários para vários microserviços.  PRÓS E CONTRAS |
| A tabela abaixo extraida do site aws.amazon.com/pt/nosql sintetiza bem o que é preciso considerar ao planejar implantar uma ou outra solução:   |  |  |  | | --- | --- | --- | |  | **Bancos de dados relacionais** | **Bancos de dados NoSQL** | | Cargas de trabalho ideais | Bancos de dados relacionais são projetados para aplicativos transacionais e fortemente consistentes de processamento de transações online (OLTP) e são bons para processamento analítico online (OLAP). | Os bancos de dados do NoSQL são projetados para vários padrões de acesso aos dados que incluem aplicativos de baixa latência. Os bancos de dados de pesquisa NoSQL são projetados para análise de dados semiestruturados. | | Modelo de dados | O modelo relacional normaliza dados em tabelas, compostas por linhas e colunas. Um esquema define estritamente tabelas, colunas, índices, relações entre tabelas e outros elementos do banco de dados. O banco de dados impõe a integridade referencial nos relacionamentos entre as tabelas. | Os bancos de dados NoSQL fornecem uma variedade de modelos de dados, como chave-valor, documento e gráfico, que são otimizados para performance e escala. | | Propriedades ACID | Bancos de dados relacionais fornecem propriedades de atomicidade, consistência, isolamento e durabilidade (ACID):   * A atomicidade exige uma transação para executar completamente ou não é executada de forma alguma. * A consistência exige que, quando uma transação é confirmada, os dados devem estar em conformidade com o esquema do banco de dados. * O isolamento exige que as transações simultâneas sejam executadas separadamente umas das outras. * A resiliência exige a capacidade de se recuperar de uma falha do sistema ou falta de energia inesperada para o último estado conhecido. | Os bancos de dados NoSQL geralmente fazem compensações relaxando algumas das propriedades ACID dos bancos de dados relacionais para um modelo de dados mais flexível que pode ser escalado horizontalmente. Isso torna os bancos de dados NoSQL uma excelente opção para casos de uso de baixa latência e alta taxa de transferência que precisam ser escalados horizontalmente além das limitações de uma única instância. | | Performance | A performance normalmente depende do subsistema do disco. A otimização de consultas, índices e estrutura de tabela é necessária para alcançar máxima performance. | A performance geralmente é uma função do tamanho do cluster do hardware subjacente, da latência de rede e do aplicativo que faz a chamada. | | Escala | Os bancos de dados relacionais geralmente escalam verticalmente o tamanho ao aumentar os recursos de computação do hardware, ou escalam horizontalmente o tamanho ao adicionar réplicas para cargas de trabalho somente leitura. | Os bancos de dados NoSQL normalmente são particionáveis porque os padrões de acesso podem escalar horizontalmente o tamanho usando arquitetura distribuída para aumentar a taxa de transferência que fornece performance consistente em escala quase ilimitada. | | APIs | As solicitações para armazenar e recuperar dados são comunicadas usando consultas compatíveis com uma Structured Query Language (SQL – Linguagem de consultas estruturadas). Essas consultas são analisadas e executadas pelo banco de dados relacional. | APIs baseadas em objetos permitem que desenvolvedores de aplicativos armazenem e restaurem facilmente estruturas de dados. As chaves de partição permitem que os aplicativos procurem pares de chave-valor, conjuntos de colunas ou documentos semiestruturados que contenham objetos e atributos de aplicativos serializados. | | |

CONCLUSÃO

Por fim, percebemos que a escolha e definição de qual tipo de banco escolher deve levar em conta aspecto muito específico de finalidade. Olhar a parte e o todo da solução esperada para cada problema de modo a considerar, inclusive, a combinação de tipos distintos de armazenamento.

As vantagens e desvantagem de cada um deles deve ser planificado ao custo e benefício, de modo que o primeiro não supere esse último e a segurança e desempenho desejado sejam entregues com a maior precisão possível.

1. Fonte: https://www.tecmundo.com.br/software/146482-sql-que-ele-serve.htm [↑](#footnote-ref-1)
2. https://blog.devart.com/sql-vs-nosql.html [↑](#footnote-ref-2)
3. <https://aws.amazon.com/pt/nosql/> [↑](#footnote-ref-3)