

Ciencias da Computação



Tipos de Banco de Dados

Integrante(s): 14007766 – *Lucas Roberto Moreira*

Professor: André Olímpio

Disciplina: *Fundamentos de Banco de Dados*

São José dos Campos – 2017

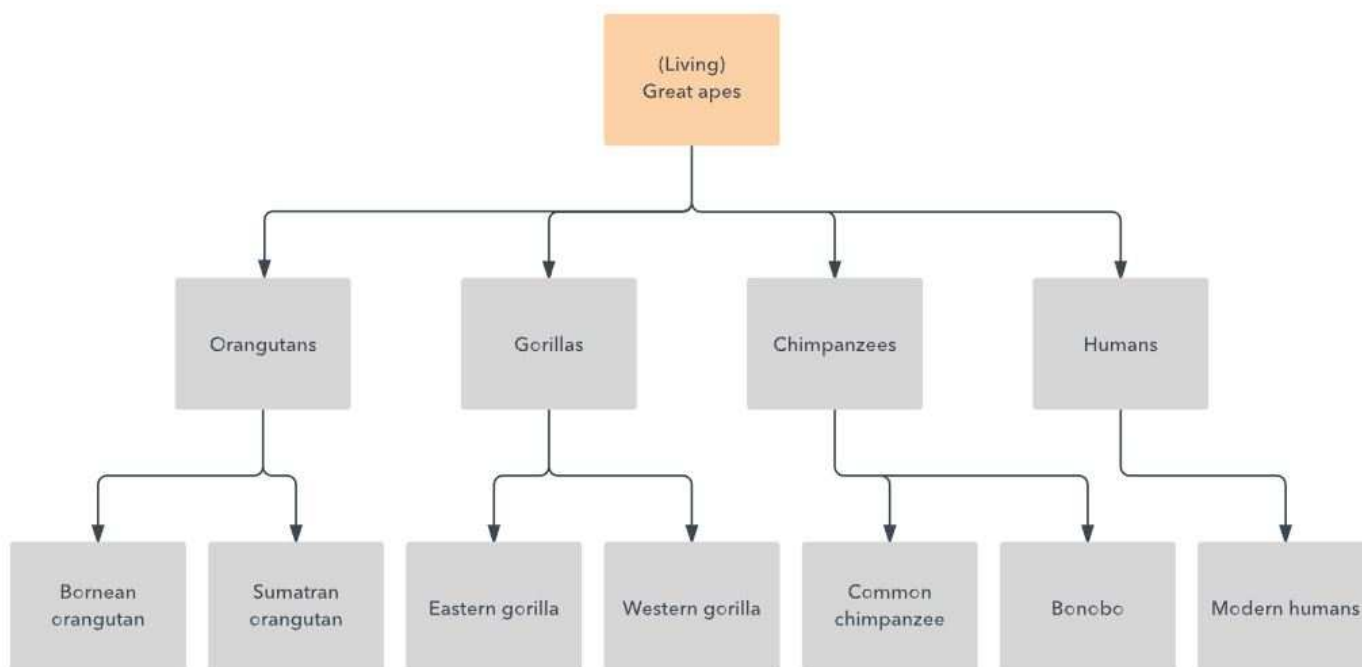
Hierárquicos

Um banco de dados hierárquico consiste em uma coleção de registros que são conectados uns aos outros por meio de ligações. Um registro é uma junção de campos, cada um contendo apenas um valor de dados. Uma ligação é uma associação entre exatamente dois registros. O modelo hierárquico é, portanto, similar ao modelo de rede, no sentido de que dados e relacionamentos entre dados são também representados por registros e ligações, respectivamente.

Um diagrama com estrutura de árvore é um esquema para um banco de dados hierárquico. Tal diagrama consiste em dois componentes básicos: retângulos, que correspondem a tipos de registro, e linhas, que correspondem a ligações.

O esquema de banco de dados é, portanto, representado como uma coleção de diagramas com estrutura de árvore. Para cada diagrama, existe uma única instância de uma árvore do banco de dados. A raiz dessa árvore é um nó auxiliar. Os filhos desse nó são instâncias de fato do tipo registro apropriado. Cada instância pode, por sua vez ter diversas instâncias de vários tipos de registro, como especificados no diagrama com estrutura de árvore correspondente.

No caso de relacionamentos muitos-para-muitos, a duplicação de registros tem dois inconvenientes principais: atualizações podem levar a inconsistência de dados e o desperdício de espaço é inevitável. A solução é registro virtual, tal registro não contém valores de dados, ele contém um ponteiro lógico para um registro físico particular. Quando um registro é duplicado em diversas árvores de banco de dados, uma única cópia daquele registro é mantida em uma das árvores e todas as outras ocorrências do mesmo são substituídas por um registro virtual contendo um ponteiro para aquele registro físico. A linguagem de manipulação de dados para essa nova configuração leva ao mesmo caso em que a duplicação de registro é permitida. Assim, um usuário não precisa preocupar-se com essas mudanças.



<https://goo.gl/mzoY7v> - Acessado em 31/08/2017, às 13:43 horas.

<https://goo.gl/qwwAC4> - Acessado em 31/08/2017, às 13:56 horas.

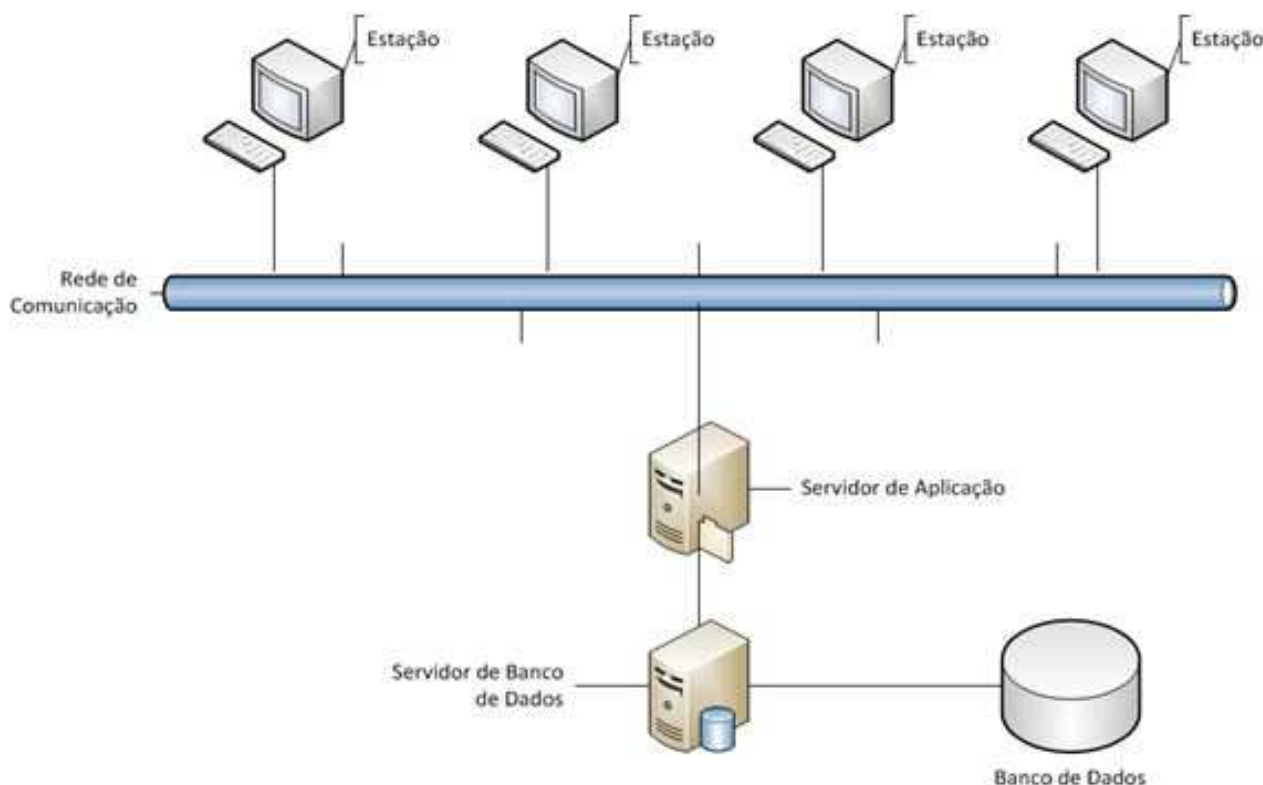
Distribuído

O sistema de Gerenciamento de Banco de Dados Distribuídos (SGBDD) controla o armazenamento e processamento de dados relacionados logicamente por meio de sistemas computacionais interconectados através de uma rede, em que tanto os dados como as funções de processamento são distribuídos entre os diversos locais.

Existem dois tipos de banco de dados distribuídos, os homogêneos e os heterogêneos. Os homogêneos são compostos pelos mesmo bancos de dados, já os heterogêneos são aqueles que são compostos por mais de um tipo de banco de dados.

Em um banco de dados distribuídos os arquivos podem estar replicados ou fragmentados, esses dois tipos podem ser encontrados ao longo de seus nós. Quando os dados se encontram replicados, existe uma cópia de cada um dos dados em cada nó, tornando as bases iguais (Ex: tabela de produtos de uma grande loja). Já na fragmentação, os dados se encontram divididos ao longo do sistema, ou seja, a cada nó existe uma base de dados diferente se olharmos de uma forma local, mas se analisarmos de uma forma global os dados são vistos de uma forma única, pois cada nó possui um catálogo que contém cada informação dos dados dos bancos adjacentes.

A replicação dos dados pode se dar de maneira síncrona ou assíncrona. No caso de replicação síncrona, cada transação é dada como concluída quando todos os nós confirmam que a transação local foi bem-sucedida. Na replicação assíncrona, o nó principal executa a transação enviando confirmação ao solicitante e então encaminha a transação aos demais nós.



<https://goo.gl/LwkrSh> - Acessado em 31/08/2017 às 14:14 horas.

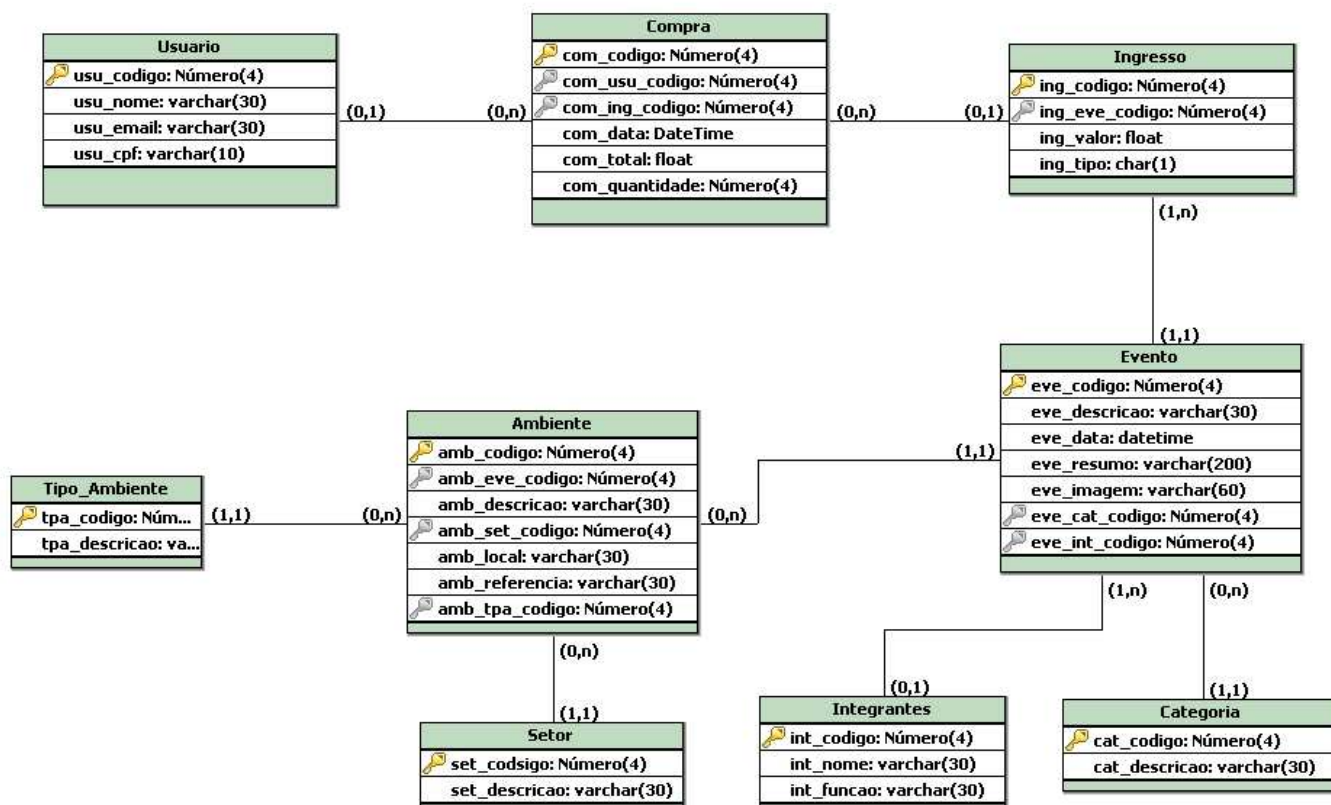
<https://goo.gl/Jqb8tV> - Acessado em 31/08/2017 às 14:18 horas.

Relacional

Um banco de dados relacional armazena dados em tabelas. Tabelas são organizadas em colunas, e cada coluna armazena um tipo de dados (inteiro, números reais, strings de caracteres, data, etc.). Os dados de uma simples “instância” de uma tabela são armazenados como uma linha. Por exemplo, a tabela Cliente teria colunas como numeroCliente, primeiroNome e sobrenome, e uma linha na tabela teria algo como {123, “Arilo”, “Dias”}.

Tabelas tipicamente possuem chaves, uma ou mais colunas que unicamente identificam uma linha na tabela. No caso da tabela Cliente a chave seria a coluna numeroCliente. Para melhorar o tempo de acesso aos dados de uma tabela, são definidos índices. Um índice provê uma forma rápida para buscar dados em uma ou mais colunas em uma tabela

O uso mais comum de SGBDRs é para implementar funcionalidades simples do tipo CRUD (do inglês Create, Read, Update e Delete – que significa as operações de Inserção, Leitura, Atualização e Exclusão de dados). Por exemplo, uma aplicação pode criar uma nova compra e inseri-la no banco de dados. Ela pode ler uma compra, trabalhar com seus dados e então atualizar o banco de dados com a nova informação. Ela pode ainda optar por excluir uma compra existente, talvez porque o cliente a cancelou. A grande maioria das interações com um banco de dados provavelmente implementará as funcionalidades básicas de CRUD.



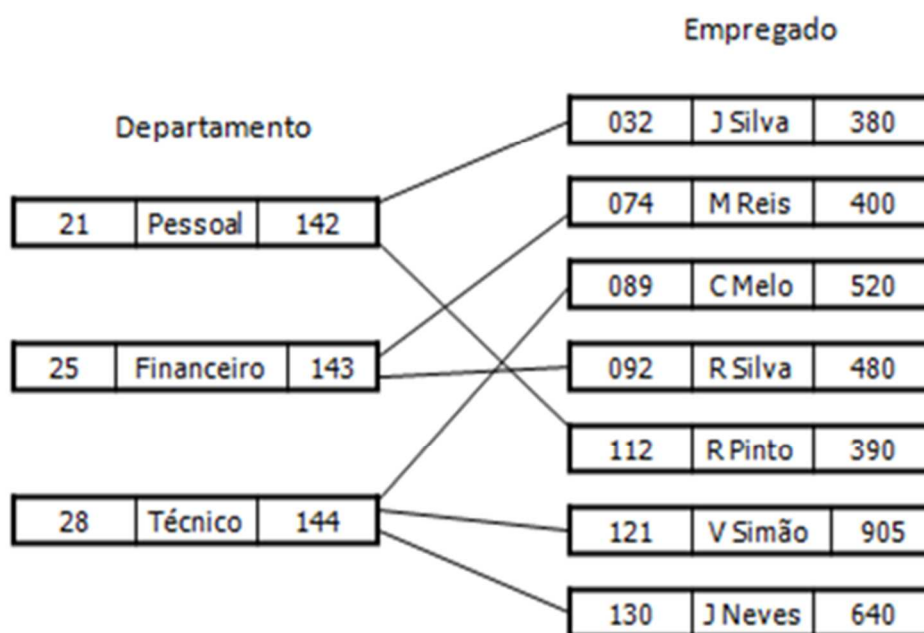
<https://goo.gl/dT5xET> Acessado 31/08/2017 às 16:02 horas.

<https://goo.gl/bRQijx> Acessado 31/08/2017 às 16:10 horas.

De Redes

O modelo em redes surgiu como uma extensão ao modelo hierárquico, eliminando o conceito de hierarquia e permitindo que um mesmo registro estivesse envolvido em várias associações. No modelo em rede, os registros são organizados em grafos onde aparece um único tipo de associação (set) que define uma relação 1:N entre 2 tipos de registros: proprietário e membro. Desta maneira, dados dois relacionamentos 1:N entre os registros A e D e entre os registros C e D é possível construir um relacionamento M:N entre A e D. O gerenciador Data Base Task Group (DBTG) da CODASYL (Committee on Data Systems and Languages) estabeleceu uma norma para este modelo de banco de dados, com linguagem própria para definição e manipulação de dados. Os dados tinham uma forma limitada de independência física. A única garantia era que o sistema deveria recuperar os dados para as aplicações como se eles estivessem armazenados na maneira indicada nos esquemas. Os geradores de relatórios da CODASYL também definiram sintaxes para dois aspectos chaves dos sistemas gerenciadores de dados: concorrência e segurança. O mecanismo de segurança fornecia uma facilidade na qual parte do banco de dados (ou área) pudesse ser bloqueada para prevenir acessos simultâneos, quando necessário. A sintaxe da segurança permitia que uma senha fosse associada a cada objeto descrito no esquema. Ao contrário do Modelo Hierárquico, em que qualquer acesso aos dados passa pela raiz, o modelo em rede possibilita acesso a qualquer nó da rede sem passar pela raiz.

Exemplo de um BD de Rede



<https://goo.gl/kJPFya> Acessado 31/08/2017 às 16:23 horas.

<https://goo.gl/zrXDyq> Acessado 31/08/2017 às 16:25 horas.

Orientado a objetos

Serve para proporcionar uma alternativa à utilização de frameworks objeto-relacionais, unificando os modelos de classes e de dados de uma aplicação Java através da persistência dos próprios objetos manipulados.

Esse tipo de banco de dados permite uma modelagem que são ditadas como mais próximas do mundo real, ainda que isto não seja possível de fato, em tese reduzindo a manutenção. Embora seja possível em outros tipos de DBs, a capacidade de criação de novos tipos de dados é melhor, o que obviamente permite criar estruturas de dados mais avançadas com melhores abstrações, mais flexíveis e teoricamente mais confiáveis, permitindo hierarquia.

Funciona melhor com linguagens orientada a objeto, evitando a tal da impedance mismatch. A navegação pelos dados é feita de forma mais natural e expressiva na maioria dos casos, Alguns padrões de uso pode aumentar a performance (não usa JOIN)

Algumas desvantagens também ocorrem nesse tipo de BD, como Falta de padronização. Cada fornecedor usa uma forma diferente, determinada uma modelagem diferente. Há várias correntes que propõe formas diferentes de uso deste modelo.

Falta de fundamentação matemática. Falta uma forma melhor de expressar consultas complexa de forma como as pessoas estão acostumadas fazer no modelo relacional.

Os produtos, apesar de longa existência, ainda não são maduros, as pessoas não o adotam porque não há adoção geral e por causa disto, não há investimentos suficientes para melhorias.

Para adotar o modelo fielmente há perda de performance em vários cenários. Para evitar isto, há uma quebra de modelo, vários tipos de acesso precisam ser feitos de forma indireta através de objetos intermediários desnecessários naquela consulta. Também pode haver complicações na concorrência automática por trazer dificuldade e assim exige mais do desenvolvedor.

Pode ser difícil obter performance por esconder problemas reais no modelo adotado, Falta também mecanismo já bem estabelecidos no modelo relacional para acesso aos dados de forma segura

Nesse tipo de BD pode haver problemas também em ferramentas, documentação, experiência, profissionais qualificados.

Mais esse tipo de BD é utilizado em projetos onde se considere a adoção de BDOOs como engine de persistência de dados, em contraposição às ferramentas e produtos tradicionais que seguem o paradigma relacional. Em aplicações onde se objetive uma maior naturalidade entre o modelo de classes e o modelo de dados, sem necessidade de mapeamento entre eles.

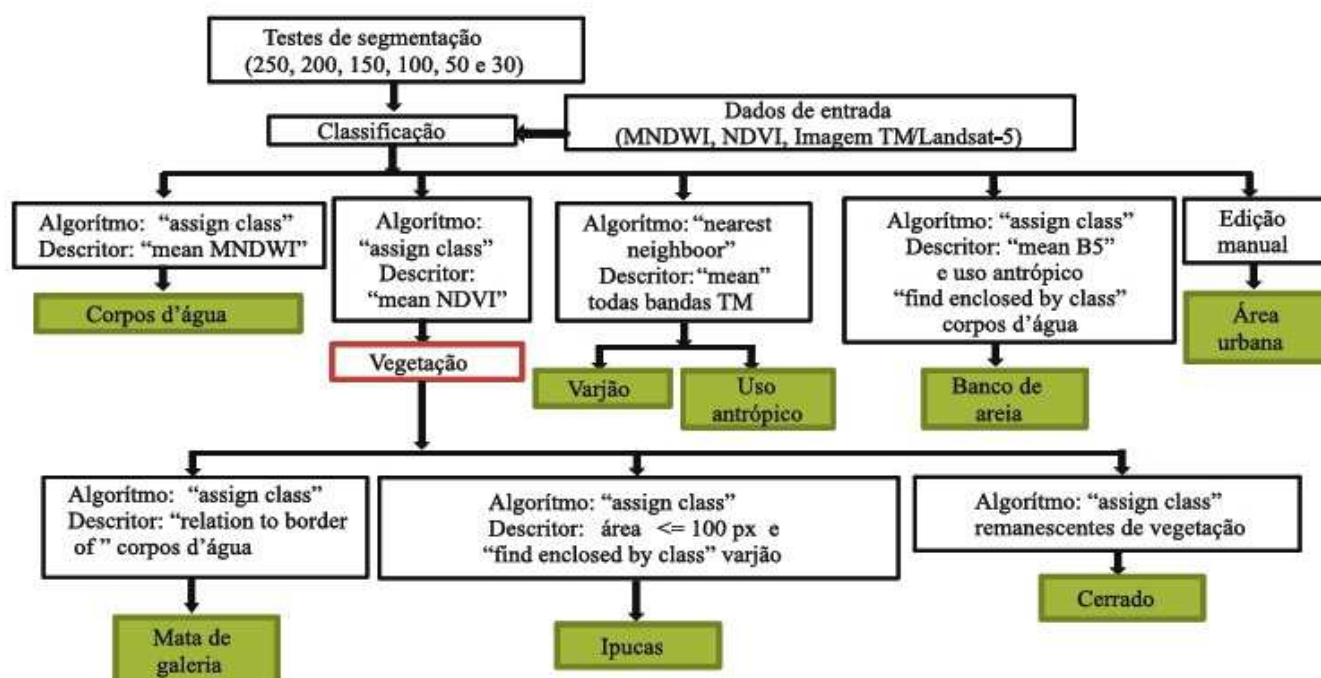
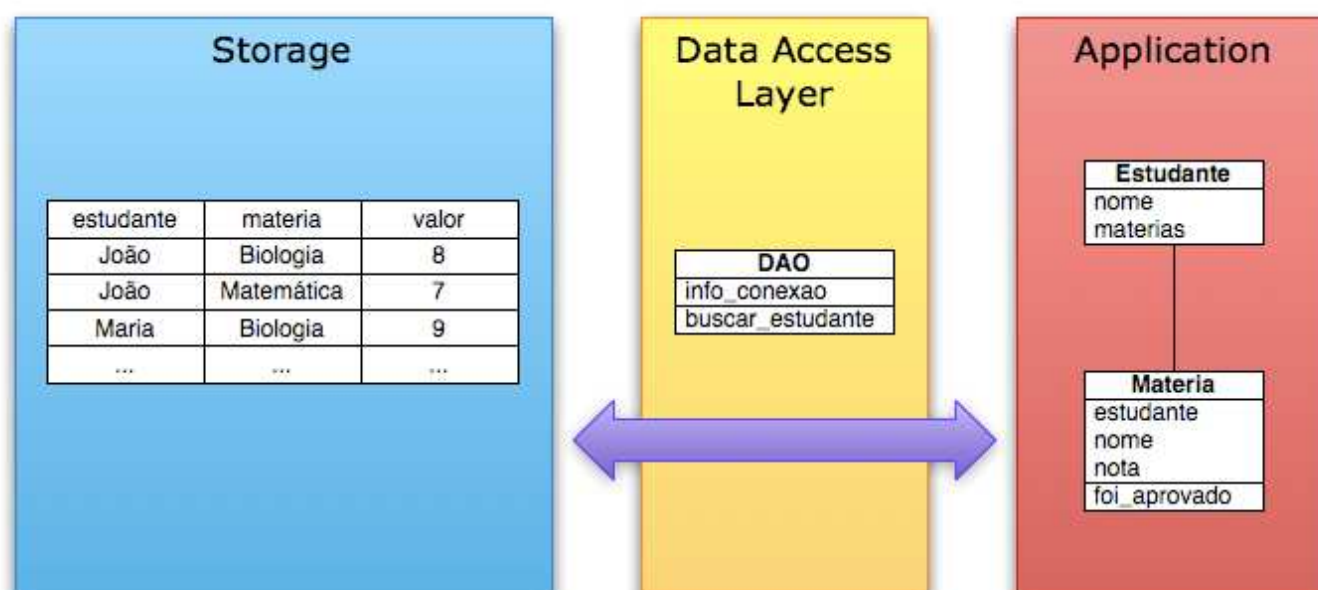


Figura 1. Fluxograma da sequência dos algoritmos e dos descritores utilizados no processo de classificação.



<https://goo.gl/zaGd6M> Acessado 02/19/2017 às 16:59 horas

<https://goo.gl/djJxt2> Acessado 02/09/2017 às 17:03 horas.

<https://goo.gl/SFYmyr> Acessado 02/09/2017 às 17:05 horas.

<https://goo.gl/iK3vLf> Acessado 02/09/2017 às 17:11 horas.

Objeto-relacional

No início da década de 90, surgiram vários SGBD-OOs (sistemas gerenciadores de bancos de dados orientados a objetos), onde o centro da atenção deixou de ser a relação com a tabela e passou a ser com a classe.

Em 1991, um grupo de fabricantes de SGBDs e empresas que trabalham com padrões criaram o grupo ODMG (Object Database Management Group), responsável por padronizar as funcionalidades dos bancos de dados orientados a objetos. Esse grupo definiu não só o modelo de dados OO, mas também os padrões ODL (Object Definition Language) e OQL (Object Query Language).

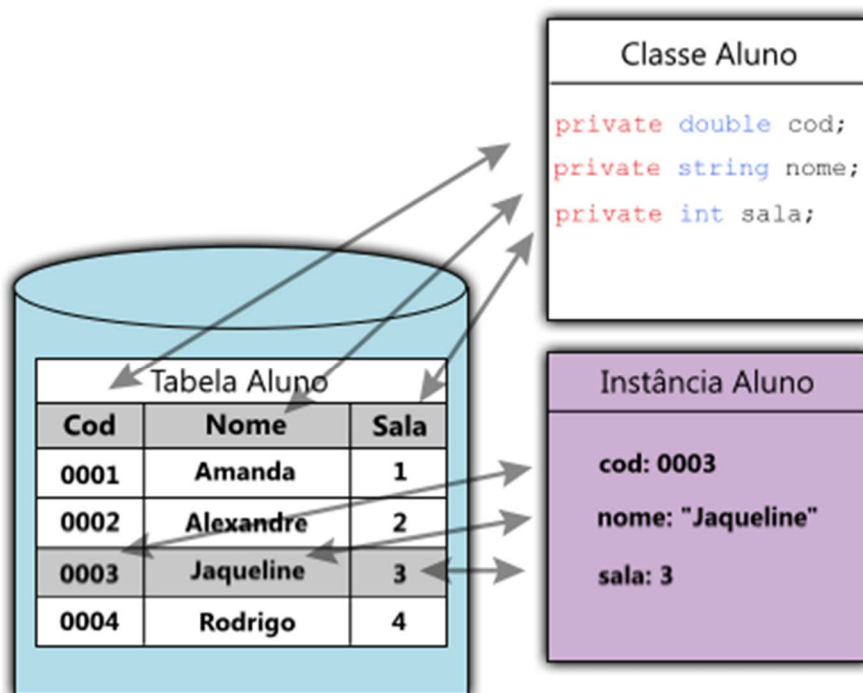
Um SGBD-OO é um sistema com as características de SGBDs, acrescidos da manipulação de objetos contemplando:

1. Definição de objetos complexos, com estrutura aninhada, como conjuntos ou listas de objetos, além dos tipos de dados primitivos;
2. Implementação de encapsulamento, determinando que todo acesso aos objetos seja por meio da aplicação de um procedimento;
3. Identidade de objeto, fazendo com que os objetos sejam distinguidos por um identificador único, mesmo que os valores dos atributos sejam os mesmos.

Atualmente existem poucos bancos orientados a objetos. Podemos citar: Gemstone, ObjectStore, Versant, Jasmine, Poet, Objectivity e O2.

Os SGBD-ORs (sistemas gerenciadores de bancos de dados objetos-relacionais) surgiram como uma reação dos principais fabricantes de SGBD-R aos SGBD-OOs. Nos bancos de dados objeto-relacionais, o banco relacional tem uma parte transformada, além de receber a adição de novos recursos que permitam implementações orientadas a objetos.

Um SGBD-OR é um SGBD que suporta SQL-3. Leia no Box 1 sobre a evolução do SQL. Alguns bancos objeto-relacionais atuais: Oracle, PostgreSQL, Informix, DB2, Cachê e SQLServer.



<https://goo.gl/ertkX6> Acessado 02/09/2017 às 17:13 horas.

<https://goo.gl/XtgTzN> Acessado 02/09/2017 às 17:15 horas.

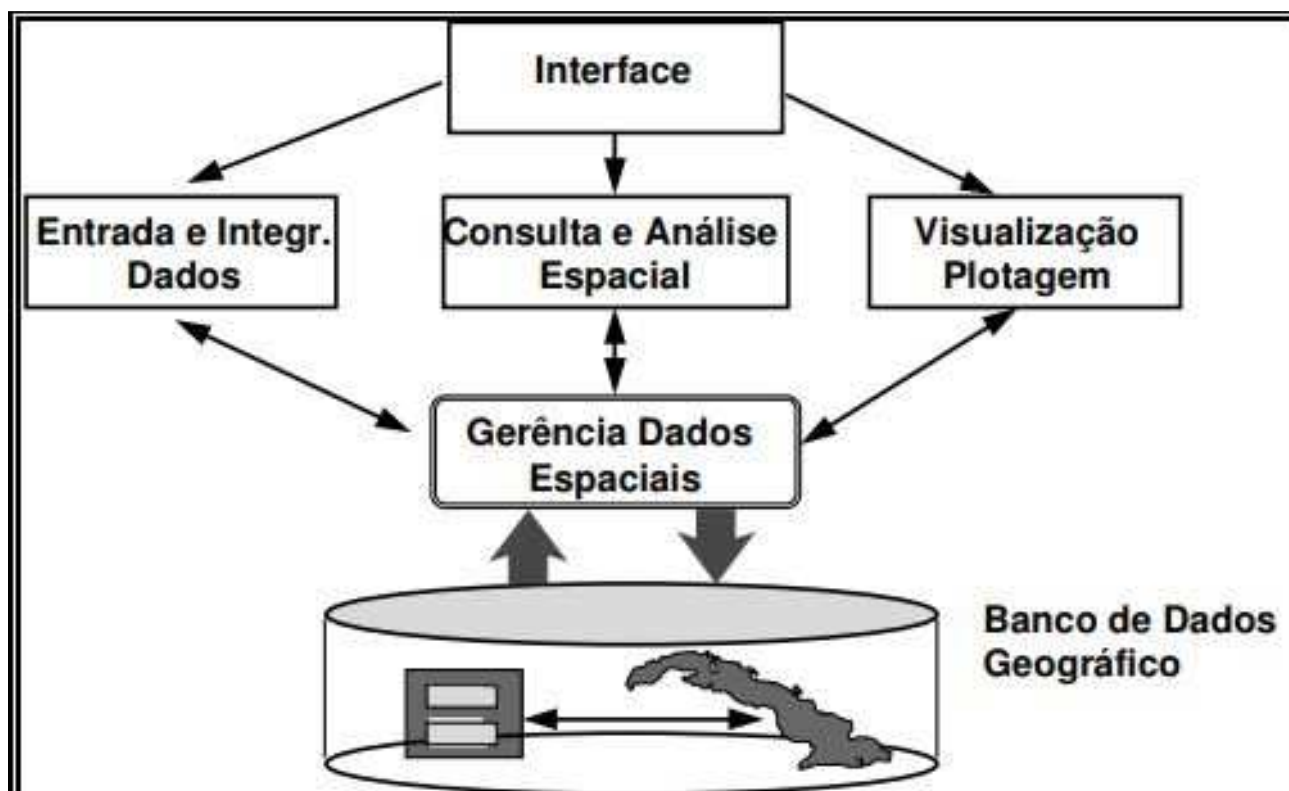
Geográfico

O BDG, também chamado de Banco de Dados Espacial (BDE), é semelhante ao relacional, com a grande e importante diferença de suportar feições geométricas em suas tabelas

Este tipo de base com geometria oferece a possibilidade de análise e consultas espaciais. É possível calcular nestes casos, por exemplo, áreas, distâncias e centroides, além de realizar a geração de buffers e outras operações entre as geometrias.

O termo sistemas de informação geográfica (SIG) é aplicado para sistemas que realizam o tratamento computacional de dados geográficos. A principal diferença de um SIG para um sistema de informação convencional é sua capacidade de armazenar tanto os atributos descritivos como as geometrias dos diferentes tipos de dados geográficos. Assim, para cada lote num cadastro urbano, um SIG guarda, além de informação descritiva como proprietário e valor do IPTU, a informação geométrica com as coordenadas dos limites do lote. A partir destes conceitos, é possível indicar as principais características de SIGs:

1. Inserir e integrar, numa única base de dados, informações espaciais provenientes de meio físico-biótico, de dados censitários, de cadastros urbano e rural, e outras fontes de dados como imagens de satélite, e GPS.
2. Oferecer mecanismos para combinar as várias informações, através de algoritmos de manipulação e análise, bem como para consultar, recuperar e visualizar o conteúdo da base de dados geográficos.



<https://goo.gl/Z64rJn> Acessado 02/09/2017 às 17:21 horas.

<https://goo.gl/47ziSc> Acessado 02/09/2017 às 17:32 horas.

NoSQL

O termo NoSQL foi primeiramente utilizado em 1998 como o nome de um banco de dados não relacional de código aberto. Seu autor, Carlos Strozzi, alega que o movimento NoSQL é completamente distinto do modelo relacional e portanto deveria ser mais apropriadamente chamado “NoREL” ou algo que produzisse o mesmo efeito.

Com a crescente popularização da internet, diversos novos dados foram surgindo e tratá-los foi se tornando gradualmente mais complexo e sua manutenção cada vez mais cara.

Em 2006, o artigo: BigTable A distributed Storage System for Stryctyred Data, publicado pelo google em 2006, traz novamente à tona o conceito NoSQL.

No início de 2009, o termo NoSQL é reintroduzido por um funcionário do Rackspace, Eric Evans, quando Johan Oskarson da Last.fm queria organizar um evento para discutir banco de dados open source distribuídos.

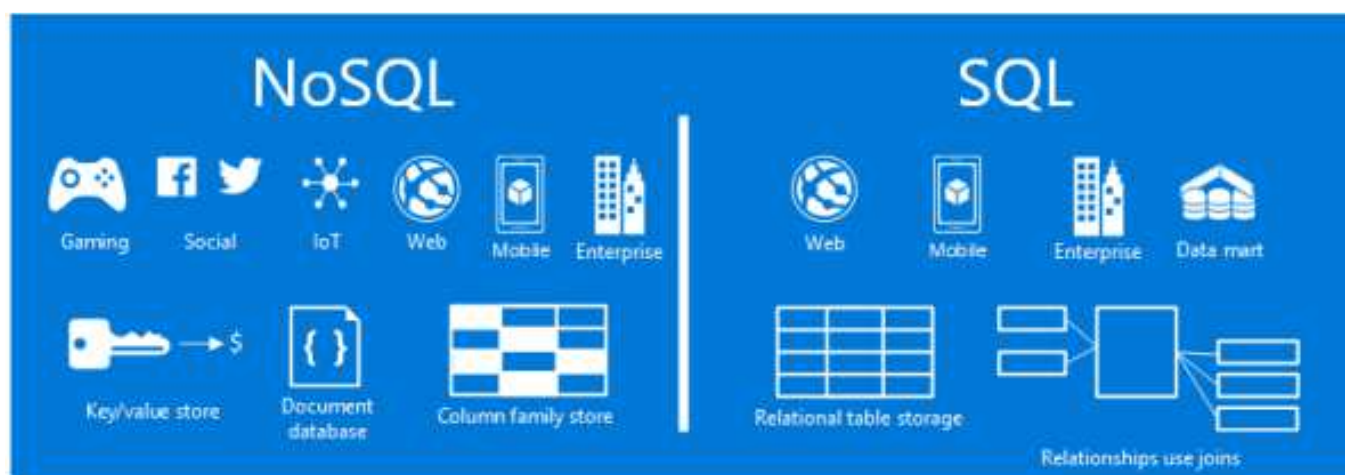
O nome era uma tentativa de descrever o surgimento de um número crescente de banco de dados não relacionais e fazia uma referência ao esquema de atribuição de nomes dos bancos de dados relacionais mais populares do mercado como MySQL, MSSQL, PostgreSQL, etc.

A partir de então os bancos de dados não relacionais passaram a ser conhecidos como NoSQL, e com crescente popularização das redes sociais, a geração de conteúdo por dispositivos moveis bem como o número cada vez maior de pessoas e dispositivos conectados, faz com que o trabalho de armazenamento de dados com o objetivo de utilizá-lo em ferramentas analíticas, comece a esbarrar nas questões de escalabilidade e custos de manutenção desses dados.

Banco de dados relacionais escalam, mas quanto maior o tamanho, mais custoso se torna essa escalabilidade, seja pelo custo de novas máquinas, seja pelo aumento de especialistas nos bancos de dados utilizados.

Já os não relacionais, permitem uma escalabilidade mais barata e menos trabalhosa, pois não exigem máquinas extremamente poderosas e sua facilidade de manutenção permite que um número menor de profissionais seja necessário.

Assim, os bancos de dados NoSQL, vão ficando mais populares entre as grandes empresas pois reúnem as características de poder trabalhar com dados semiestruturados ou crus vindos de diversas origens (arquivos de log, websites, arquivos multimídia, etc..).



<https://goo.gl/g2HQwc> Acessado 03/09/2017 às 01:48 horas.

<https://goo.gl/J8TBkJ> Acessado 03/09/2017 às 01:52 horas.