# **Apostila de Banco de Dados e SQL**

|  |  |
| --- | --- |
| Autoria | Prof. Daniella Vieira |
| Versão | 1.0 |

Sumário

[**Apostila de Banco de Dados e SQL** 1](#_Toc50025193)

[Introdução 2](#_Toc50025194)

[Gerenciamento de dados nas organizações 2](#_Toc50025195)

[Banco de Dados 3](#_Toc50025196)

[Conceitos importantes 4](#_Toc50025197)

[SGBD - Sistema de Gerenciamento de Banco de Dados 5](#_Toc50025198)

[Características Gerais de um SGBD 6](#_Toc50025199)

[Arquitetura de um SGBD 9](#_Toc50025200)

[Modelos de Dados 10](#_Toc50025201)

[Componentes de um Banco de Dados 11](#_Toc50025202)

[Tipos de SGBD 12](#_Toc50025203)

[Esquema de Organização dos Dados 12](#_Toc50025204)

[Tipos de Banco de Dados 14](#_Toc50025205)

[Banco de Dados Hierárquicos 14](#_Toc50025206)

[Banco de Dados em Redes 14](#_Toc50025207)

[Banco de Dados Orientados ao Objeto 16](#_Toc50025208)

[Banco de Dados Relacional 16](#_Toc50025209)

# Introdução

No censo Americano em 1880, de 50 milhões de pessoas, foram coletados dados, tais como: idade, sexo, ocupação, educação, número de pessoas deficientes.Os registros foram organizados em livros escritos a mão.

Hermann Hollerith (1860-1929) - funcionário do United States Census Bureau - inventou, em 1880, uma máquina para realizar as operações de recenseamento da população. A máquina fazia a leitura de cartões de papel perfurados em código BCD (Binary Coded Decimal) e efetuava contagens da informação referente à perfuração respectiva.

A máquina que Hollerith usou para tabular o censo daquele ano nos Estados Unidos levou não mais que dois anos e meio. A máquina tinha um leitor de cartões, um contador, um classificador e uma tabulação criada pelo mesmo.

Em 1896, Hollerith cria a Tabulating Machine Company, com a que pretendia comercializar a sua máquina. A fusão desta empresa com outras três (International Time Recording Company, a Computing Scale Corporation, e a Bundy Manufacturing Company), deu lugar, em 1924, à International Business Machines Corporation – a IBM.

E depois disso vieram tecnologias a exemplo de:

* Fitas magnéticas: é uma mídia de armazenamento não-volátil que consiste em uma fita plástica coberta de material magnetizável. A primeira aplicação da fita magnética foi o registro de áudio, razão pela qual ela foi inventada por Fritz Pfleumer em 1928, na Alemanha.
* Discos magnéticos, disco rígido ou disco duro, popularmente chamado também de HD (derivação de HDD do inglês hard disk drive), "memória de massa" ou ainda de "memória secundária" é a parte do computador onde são armazenados os dados. O primeiro disco rígido foi construído pela IBM em 1956.
* Disco óptico é chato, circular, usualmente feito de camadas de policarbonato, acrílico e alumínio. Em termos de funcionamento os discos ópticos diferem dos discos magnéticos por utilizarem as propriedades da luz ao invés das propriedades eletromagnéticas.

## Gerenciamento de dados nas organizações

O gerenciamento da informação é o conjunto de ações referentes aos processos de identificação, armazenamento, gerenciamento, busca e distribuição de informações.

|  |
| --- |
| Consulte o site que apresenta o relatório “Global Digital Growth”. Disponível em: <https://datareportal.com/reports/digital-2020-global-digital-overview> |

# Banco de Dados

Todos nós sabemos existirem gigantescas bases de dados gerenciando nossas vidas. De fato sabemos que nossa conta bancária faz parte de uma coleção imensa de contas bancárias de nosso banco. Nosso Título Eleitoral ou nosso Cadastro de Pessoa Física, certamente estão armazenados em Bancos de Dados colossais. Sabemos também que quando sacamos dinheiro no Caixa Eletrônico de nosso banco, nosso saldo e as movimentações existentes em nossa conta bancária já estão à nossa disposição.

Nestas situações sabemos que existe uma necessidade em se realizar o armazenamento de uma série de informações que não se encontram efetivamente isoladas umas das outras, ou seja, existe uma ampla gama de dados que se referem a relacionamentos existentes entre as informações a serem manipuladas.

Estes Bancos de Dados, além de manterem todo este volume de dados organizado, também devem permitir atualizações, inclusões e exclusões do volume de dados, sem nunca perder a consistência. E não podemos esquecer que na maioria das vezes estaremos lidando com acessos concorrentes a várias tabelas de nosso

banco de dados, algumas vezes com mais de um acesso ao mesmo registro de uma mesma tabela!

O fato de montarmos uma Mala Direta em um micro PC-XT com um drive já faz de nós um autor de um Banco de Dados?

Claro que não! Um Banco de Dados é antes de mais nada uma coleção logicamente coerente de dados com determinada significação intrínseca. Em outras palavras um arquivo contendo uma série de dados de um cliente, um arquivo com dados aleatoriamente gerados e dois arquivos padrão dbf (dBase) que tem uma relação definida entre ambos, não pode ser considerada uma Base de Dados Real.

Um Banco de Dados contém os dados dispostos numa ordem pré-determinada em função de um projeto de sistema, sempre para um propósito muito bem definido.

Um Banco de Dados representará sempre aspectos do Mundo Real. Assim sendo uma Base de Dados (ou Banco de Dados, ou ainda BD) é uma fonte de onde poderemos extrair uma vasta gama de informações derivadas, que possui um nível de interação com eventos como o Mundo Real que representa. A forma mais comum de interação Usuário e Banco de Dados, dá-se através de sistemas específicos que por sua vez acessam o volume de informações geralmente através da linguagem SQL.

Os Administradores de Banco de Dados (DBA) são responsáveis pelo controle ao acesso aos dados e pela coordenação da utilização do BD. Já os projetistas de Banco de Dados (DBP) são analistas que identificam os dados a serem armazenados em um Banco de Dados e pela forma como estes serão representados.

Os Analistas e Programadores de Desenvolvimento, criam sistemas que acessam os dados da forma necessária ao Usuário Final, que é aquele que interage diretamente com o Banco de Dados.

## Conceitos importantes

1. **Dado**

Fato do mundo real que está registrado e possui um significado implícito no contexto de um domínio de aplicação.

Exemplo: Idade.

1. **Informação**

Fato útil que pode ser extraído direta ou indiretamente a partir dos dados.

Exemplos: Faixa etária da sala - faixa menor e maior de idade de todos os alunos da sala.

1. **Banco de dados (BD)**

Coleção de dados inter-relacionados e persistentes que representa um subconjunto dos fatos presentes em um domínio de aplicação (universo de discurso).

Exemplo: Banco de dados da biblioteca.

1. **Área de banco de dados**

Área que pesquisa tecnologias para dar suporte eficiente ao gerenciamento de dados de sistemas de informação.

1. **Tabela Lógica**

União de linhas e colunas para armazenamento de dados dos sistemas.

Exemplo: Tabela de clientes

1. **Campo**

Equivalente a coluna. É a unidade básica de uma tabela. Possui um nome, tipo e tamanho.

Exemplo: Nome, e-mail, endereço

1. **Registro**

Equivalente a linha. É uma coleção de itens de dados.

Exemplo: Um registro sobre o cliente

Um registro pode ser dividido em vários campos: um campo pode ser um nome, um número ou uma combinação de caracteres.

1. **Esquema de Banco de Dados**

É a descrição do BD.

1. **Instância de BD**

São os dados armazenados no BD em um determinado instante de tempo.

# SGBD - Sistema de Gerenciamento de Banco de Dados

Um SGBD - Sistema de Gerenciamento de Banco de Dados é uma coleção de programas que permitem ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades.

Um conceito que deverá ficar bastante claro inicialmente é o que envolve a separação clara entre os Gerenciadores de Base de Dados dos Gerenciadores de Arquivo.

Sistemas baseados em "Banco de Dados" baseados em Btrieve e dBase (Fox e Clipper), podem no máximo simular as características típicas de um ambiente de Banco de Dados. As linguagens Delphi (utiliza opcionalmente o padrão dBase) e o VB (que utiliza o Access), recomendam a utilização de Banco de Dados reais, porém utilizam àqueles "Banco de Dados" que possuem algumas características de Bancos de Dados, mas possuem características típicas de Gerenciadores de Arquivo.

Vamos definir algumas regras básicas e claras para um sistema de manipulação de dados ser considerado um SGBD. Fica implícito que se ao menos uma das características abaixo não estiver presente no nosso "candidato" a SGBD, este poderá ser um GA (Gerenciador de Arquivo) de altíssima qualidade, "quase" um

SGBD, mas não um SGBD.

**Regra 1: Autocontenção -** Um SGBD não contém apenas os dados em si, mas armazena completamente toda a descrição dos dados, seus relacionamentos e formas de acesso. Normalmente esta regra é chamada de Meta-Base de Dados. Em um GA, em algum momento ao menos, os programas aplicativos declaram estruturas (algo que ocorre tipicamente em C, COBOL e BASIC), ou geram os relacionamentos entre os arquivos (típicos do ambiente xBase). Por exemplo, quando você é obrigado a definir a forma do registro em seu programa, você não está lidando com um SGBD.

**Regra 2: Independência dos Dados** - Quando as aplicações estiverem realmente imunes a mudanças na estrutura de armazenamento ou na estratégia de acesso aos dados, podemos dizer que esta regra foi atingida. Portanto, nenhuma definição dos dados deverá estar contida nos programas da aplicação. Quando você resolve criar uma nova forma de acesso, um novo índice, *se precisar alterar o código de seu aplicativo*, você não está lidando com um SGBD.

**Regra 3: Abstração dos Dados** - Em um SGBD real é fornecida ao usuário somente uma representação conceitual dos dados, o que não inclui maiores detalhes sobre sua forma de armazenamento real. O chamado Modelo de Dados é um tipo de abstração utilizada para fornecer esta representação conceitual. Neste modelo, um esquema das tabelas, seus relacionamentos e suas chaves de acesso são exibidos ao usuário, porém nada é afirmado sobre a criação dos índices, ou como serão mantidos, ou qual a relação existente entre as tabelas que deverá ser mantida íntegra. Assim se você desejar inserir um pedido em um cliente inexistente e esta entrada não for automaticamente rejeitada, você não está lidando com um SGBD.

**Regra 4: Visões**- Um SGBD deve permitir que cada usuário visualize os dados de forma diferente daquela existente previamente no Banco de Dados. Uma visão consiste de um subconjunto de dados do Banco de Dados, necessariamente derivados dos existentes no Banco de Dados, porém estes não deverão estar

explicitamente armazenados. Portanto, toda vez que você é obrigado a replicar uma estrutura, para fins de acesso de forma diferenciada por outros aplicativos, você não está lidando com um SGBD.

**Regra 5: Transações**- Um SGBD deve gerenciar completamente a integridade referencial definida em seu esquema, sem precisar em tempo algum, do auxílio do programa aplicativo. Desta forma exige-se que o banco de dados tenha ao menos uma instrução que permita a gravação de uma série modificações simultâneas e uma instrução capaz de cancelar uma série modificações. Por exemplo, imaginemos que estejamos cadastrando um pedido para um cliente, que este deseje reservar 5 itens de nosso estoque, que estão disponíveis e, portanto, são reservados, porém existe um bloqueio financeiro (duplicatas em atraso) que impede a venda. A transação deverá ser desfeita com apenas uma instrução ao Banco de Dados, sem quaisquer modificações suplementares nos dados. Caso você se obrigue a corrigir as reservas, através de acessos complementares, você não está lidando com um SGBD.

**Regra 6: Acesso Automático**- Em um GA uma situação típica é o chamado Dead-Lock, o abraço mortal. Esta situação indesejável pode ocorrer toda vez que um usuário travou um registro em uma tabela e seu próximo passo será travar um registro em uma tabela relacionada à primeira, porém se este registro estiver previamente travado por outro usuário, o primeiro usuário ficará paralisado, pois, estará esperando o segundo usuário liberar o registro em uso, para que então possa travá-lo e prosseguir sua tarefa. Se por hipótese o segundo usuário necessitar travar o registro travado pelo primeiro usuário (!), afirmamos que ocorreu um abraço mortal, pois cada usuário travou um registro e precisa travar um outro, justamente o registro anteriormente travado pelo outro! Imaginemos um caso onde o responsável pelos pedidos acabou de travar o Registro Item de Pedido, e, necessita travar um registro no Cadastro de Produtos, para indicar uma nova reserva. Se concomitantemente estiver sendo realizada uma tarefa de atualização de pendências na Tabela de Itens, e para tanto, previamente este segundo usuário travou a Tabela de Produtos, temos a ocorrência do abraço mortal. Se a responsabilidade de evitar esta ocorrência for responsabilidade da aplicação, você não está lidando com um SGBD.

**Conclusão:** Um SGBD deve obedecer INTEGRALMENTE às seis regras acima. Em caso contrário estaremos diante de um GA ou de um "quase" SGBD.

Características Gerais de um SGBD

Os SGBD têm sete características operacionais elementares sempre observadas, que passaremos a listar:

**Característica 1: Controle de Redundâncias-** A redundância consiste no armazenamento de uma mesma informação em locais diferentes, provocando inconsistências. Em um Banco de Dados as informações só se encontram armazenadas em um único local, não existindo duplicação descontrolada dos dados. Quando existem replicações dos dados, estas são decorrentes do processo de armazenagem típica do ambiente Cliente-Servidor, totalmente sob controle do Banco de Dados.

**Característica 2: Compartilhamento dos Dados** - O SGBD deve incluir software de controle de concorrência ao acesso dos dados, garantindo em qualquer tipo de situação a escrita/leitura de dados sem erros.

**Característica 3: Controle de Acesso** - O SGDB deve dispor de recursos que possibilitem selecionar a autoridade de cada usuário. Assim um usuário poderá realizar qualquer tipo de acesso, outros poderão ler alguns dados e atualizar outros e outros ainda poderão somente acessar um conjunto restrito de dados para escrita e leitura.

**Característica 4: Interfaceamento** - Um Banco de Dados deverá disponibilizar formas de acesso gráfico, em linguagem natural, em SQL ou ainda via menus de acesso, não sendo uma "caixa-preta" somente sendo passível de ser acessada por aplicações.

**Característica 5: Esquematização** - Um Banco de Dados deverá fornecer mecanismos que possibilitem a compreensão do relacionamento existentes entre as tabelas e de sua eventual manutenção.

**Característica 6: Controle de Integridade** - Um Banco de Dados deverá impedir que aplicações ou acessos pelas interfaces possam comprometer a integridade dos dados.

**Característica 7: Backups** - O SGBD deverá apresentar facilidade para recuperar falhas de hardware e software, através da existência de arquivos de pré-imagem ou de outros recursos automáticos, exigindo minimamente a intervenção de pessoal técnico.

Existe a possibilidade de encontramos Bancos de Dados que não satisfaçam completamente todas as características acima, o que não a inválida como Banco de Dados. Na prática podemos encontrar situações onde a primeira característica não seja importante, pois podemos ter o Banco de Dados baseado totalmente em um único servidor, e as redundâncias podem ser aceitas em algumas situações sob controle da aplicação (algo não muito recomendado, mas passível de aceitação, em situações onde a existência do nome do cliente em um arquivo contendo duplicatas emitidas, possibilita o acesso a apenas uma tabela sem relacionamentos, e sabe-se de antemão que uma duplicata depois de emitida, não pode ter seu cliente alterado).

A segunda característica (Compartilhamento dos Dados) pode ser desconsiderada principalmente em ambiente de desenvolvimento, ou ainda em aplicações remotas.

O Controle de Acesso pode ser descartado em pequenas empresas, sendo que o aplicativo em questão, mais o software de rede, podem facilmente se incumbir desta característica, no caso de pequenas empresas, com reduzido número de pessoas na área operacional.

O Interfaceamento e a Esquematização, são características sempre disponíveis, o que varia neste caso é qualidade destes componentes, que vai desde o sofrível até o estado da arte. É muito conveniente que esta característica seja muito boa em um Banco de Dados, onde estiverem em atuação mais de um Administrador de Banco de Dados e tivermos um número relativamente alto de sistemas desenvolvidos ou em desenvolvimento neste ambiente.

De fato, quanto maior o número de pessoas envolvidas no desenvolvimento de aplicações e gerenciamento do Banco de Dados, mais importante tornam-se estas duas características, pois cada novo sistema desenvolvido precisará sempre estar adequado ao Banco de Dados da Empresa e aderente aos padrões de acesso utilizados nos sistemas concorrentes.

As interfaces ISQL e WinSQL devem deixar muito claro ao estudante como uma interface pobre (no caso a existente no ISQL) perde muito, quando comparada a uma interface mais recursiva. A esquematização existente no Banco de Dados é muito melhor do que aquela mantida em alguma pasta, em algum arquivo do CPD, que sempre está “um pouquinho” desatualizada.

O Controle de Integridade, é outra característica sempre presente nos Bancos de Dados, mas existem diferenças quando da implementação desta característica. Assim, é comum encontrarmos Bancos de Dados que suportam determinado acesso, enquanto outros não dispõe de recurso equivalente.

O Backup em tempo de execução, é outra característica sempre disponível, porém temos aplicações que invariavelmente são comprometidas por falhas de hardware, e outras, que o mesmo tipo de falha não causa perda alguma de dados ou de integridade. Novamente, cada Banco de Dados tem esta característica melhor ou pior implementada, cabendo ao Administrador de Banco de Dados escolher aquele que lhe oferecer mais segurança.

Devemos ressaltar ainda, que podemos ter um Banco de Dados Modelo A, que respeite integralmente as regras básicas e disponha de todas as características apresentadas, enquanto um Modelo B que apesar de respeitar as regras básicas, não suporte uma ou outra característica desejável, mas tenha um desempenho excelente, enquanto o Modelo A seja apenas razoável no quesito desempenho, nos levará seguramente a escolher o Modelo B como sendo o ganhador para nossa instalação!

Isto ocorre pois, na prática, todo usuário deseja um tempo de resposta muito pequeno. O chamado “prazo de entrega” muito comum em Bancos de Dados operando nos limites de sua capacidade, ou nos casos onde o hardware está muito desatualizado, é fonte de inúmeros problemas para o pessoal de informática. Neste caso é melhor abrirmos mão de uma Interface Amigável, de um gerenciamento Automático de Backups ou ainda de outras características que não julgarmos fundamentais, para nos livrarmos do problema típico de ambiente extremamente comprometido, por má performance do Banco de Dados.

A escolha do Banco de Dados da empresa, portanto é uma decisão muito delicada, na medida em que está irá acarretar troca de aplicativos e troca de hardware. Os investimentos diretamente aplicados no Banco de Dados, costumam ser infinitamente menores do que aqueles a serem aplicados na empresa, visando sua perfeita adequação ao novo SGBD. Esta decisão, sempre que possível, deve ser tomada por especialistas em Banco de Dados, com profundos conhecimentos de Análise de Sistemas, de Banco de Dados e de Software de Gerenciamento de Bases de Dados, de forma a evitar que a empresa escolha um Banco de Dados inadequado aos seus propósitos, e que pouco tempo depois, seja obrigada a perder todos investimento realizado em Software e Hardware.

## Arquitetura de um SGBD

Podemos dizer que o Banco de Dados tem um Nível Interno, onde é descrita a estrutura de armazenamento físico dos dados, um Nível Intermediário, onde temos a descrição lógica dos dados e um Nível Externo onde são descritas as visões para grupos de usuários.

Não podemos deixar de lembrar ainda que o Banco de Dados garante a Independência Lógica e Física dos Dados, portanto podemos alterar o esquema conceitual dos dados, sem alterar as visões dos usuários ou mesmo alterar o esquema interno, sem, contudo, alterar seu esquema conceitual.

# Modelos de Dados

O Modelo de Dados é basicamente um conjunto de conceitos utilizados para descrever um Banco de Dados. Não existe uma única forma de representação deste modelo, porém qualquer forma que permita a correta compreensão das estruturas de dados compreendidas no Banco de Dados, pode ser considerada adequada. Vamos descrever sucintamente este modelo, pois estes serão objetos de outras disciplinas:

**Modelo Orientado ao Registro:** São modelos que representam esquematicamente as estruturas das tabelas de forma bastante próxima a existente fisicamente. Basicamente são apresentados os registros de cada tabela (inclusive seus campos) e seus relacionamentos elementares. O Modelo Relacional, o Modelo de Rede e o Hierárquico são exemplos deste tipo de representação.

**Modelo Semântico:** São modelos onde existe uma representação explícita das entidades e relacionamentos. O Modelo Entidade-Relacionamento e o Funcional, são exemplos deste tipo de abordagem.

**Modelo Orientado ao Objeto:** São modelos que procuram representar as informações através dos conceitos típicos da Programação Orientada ao Objeto, utilizando o conceito de Classes que irão conter os objetos. Citamos os Modelos O2 e o de Representação de Objetos como exemplos típicos desta abordagem.

O conceito de instância, sempre muito presente, poderia ser definido como sendo o conjunto de dados que definem claramente um Banco de Dados em determinado instante. Devemos entender então o Banco de Dados como sendo não apenas um conjunto de dados digitados, mas também todo o esquema e regras armazenada e

controladas pelo SGBD.

Em outras palavras, podemos dizer que os SGBD, vieram para eliminar todo o trabalho que anteriormente um programador de aplicação realizava controlando o acesso, integridade e redundância dos dados.

# Componentes de um Banco de Dados

Um Banco de Dados é composto pelas seguintes partes:

**Gerenciador de Acesso ao Disco:** O SGBD utiliza o Sistema Operacional para acessar os dados armazenados em disco, controlando o acesso concorrente às tabelas do Banco de Dados. O Gerenciador controla todas as pesquisas queries) solicitadas pelos usuários no modo interativo, os acessos do compilador DML, os acessos feitos pelo Processador do Banco de Dados ao Dicionário de Dados e também aos próprios dados.

O **Compilador DDL** (Data Definition Language) processa as definições do esquema do Banco de Dados, acessando quando necessário o Dicionário de Dados do Banco de Dados.

O **Dicionário de Dados** contém o esquema do Banco de Dados, suas tabelas, índices, forma de acesso e relacionamentos existentes.

O **Processador do Banco de Dados** manipula requisições à própria Base de Dados em tempo de execução. É o responsável pelas atualizações e integridade da Base de Dados.

O **Processador de Pesquisas** (queries) dos usuários, analisa as solicitações, e se estas forem consistentes, aciona o Processador do Banco de Dados para acesso efetivo aos dados.

As aplicações fazem seus acessos ao pré-compilador DML da linguagem hospedeira, que os envia ao **Compilador DML** (Data Manipulation Language) onde são gerados os códigos de acesso ao Banco de Dados.

# Tipos de SGBD

Podemos citar como tipos principais os Bancos de Dados Relacionais, os Hierárquicos, os de Rede, os Semânticos, os Orientados a Objetos e os Universais.

Os Bancos de Dados alvo de nosso estudo serão os Relacionais, sendo que os demais tipos serão apenas citados superficialmente, por não serem parte integrante de nosso curso.

## Esquema de Organização dos Dados

Em Linguagem C os estudantes tomaram (ou irão tomar contato) com os ponteiros de registro, que aqui representaremos como sendo números de acesso ao registro. Visando diferenciar o número do registro físico do número do registro apontado pelo ponteiro, usaremos o símbolo (#) para indicar o número do registro físico, enquanto o símbolo (\*) será utilizado pelo para indicar o endereço indicado pelo ponteiro, a semelhança da representação usual dos programadores da Linguagem C.

Vamos supor o arquivo abaixo ordenado alfabeticamente (físico):

#1- Amarelo \*2

#2- Azul \*3

#3- Branco \*4

#4- Preto \*5

#5- Verde \*6

#6- Vermelho --

Supondo desejarmos incluir a cor Laranja, seríamos obrigados a re-escrever todo o arquivo de modo a Laranja ocupar o registro 4. Vamos antes de fazer uma nova ordenação, analisar a solução abaixo:

#1- Amarelo \*2

#2- Azul \*3

#3- Branco \*7

#4- Preto \*5

#5- Verde \*6

#6- Vermelho --

#7- Laranja \*4

Observe que o registro #3 (Branco) passou a apontar para o registro \*7, que contém o novo dado (Laranja). O novo dado passa a apontar para o registro previamente apontado pelo registro que agora o aponta. Parece, e é confuso, mas se você analisar o esquema abaixo perceberá que apesar do palavreado confuso, facilmente qualquer um de nós percebe a maneira adequada de inserir novos registros.

Algo --> Apontado

Algo --> Novo --> Apontado

#1 Algo --> \*2

#2 Apontado não aponta (é o último física e logicamentex)

#1 Algo --> \*3

#2 Apontado não aponta (é o último logicamente)

#3 Novo --> \*2 (é o último fisicamente)

A chamada perda de ponteiros, fenômeno dos mais temidos pelos profissionais de sistema, nada mais é que a perda de referência lógica entre registros de uma tabela.

Existem diversas técnicas de acesso como as chamadas Btree+ (Arvore Binária Balanceada), Hashing, Sequencial Ordenado, Hashing Dinâmico, Hashing Extensível e Hashing Linear, próprios para um curso específico de Banco de Dados, que não chegaremos a analisar em nosso curso.

Sabemos que em linguagem C foi (ou será) apresentada a técnica de balanceamento de estruturas, que mostrou (ou mostrará) como um Banco de Dados é arranjado internamente.

**Exercício:** Represente esquematicamente o arranjo alfabético dos seguintes times de futebol: Fluminense, Flamengo, Vasco, Botafogo, Madureira, América e Olária. Suponha ainda que o Bangu queira participara do campeonato, como ficaria a nova ordem da tabela?

## Modelos de banco de dados

## Modelo de Arquivo Simples

Um banco de dados de arquivo simples é um banco de dados projetado com base em uma única tabela. Em geral, os bancos de dados de arquivo simples estão em formato de texto simples, no qual cada linha contém apenas um registro. Os campos do registro são separados por delimitadores, como tabulações e vírgulas.

Os Livros, assim como os Autores, estão armazenados nesta tabela única, resultando na repetição de valores de dados.

|  |  |  |
| --- | --- | --- |
| ID Autor | Autor | Livro |
| AD0001 | Oscar Wilde | A Vision |
| AD0002 | Leo Tolstoy | War and Peace |
| AD0023 | Oliver Goldsmith | Citizen of the World |
| AD0023 | Oliver Goldsmith | The DesertedVillage |

## Banco de Dados Hierárquicos

Em um modelo de banco de dados hierárquico, os dados são organizados em uma estrutura semelhante a uma árvore. Os dados são armazenados como registros conectados entre si por meio de vínculos. Um registro é uma coleção de campos. Um registro no modelo de banco de dados hierárquico corresponde a uma linha no modelo de banco de dados relacional.

Seguem o estilo de um organograma empresarial (Diretoria-Divisão-Seção-Setor) ou de biblioteca (Exata-Matemática-Algebra Linear-Vetores). Este modelo é capaz de representar este tipo de organização de forma direta, mas apresenta inconvenientes quando esta situação não aparece claramente com relações de hierarquia.

O Exemplo a seguir (Folha de Pagamento) deve servir para esclarecer melhor o estilo deste modelo:

Membros > Aluno

Membros > Professor > Tempo integral

Membros > Professor > Tempo parcial

## Banco de Dados em Redes

Neste modelos os dados são dispostos em registros, previamente classificados em classes que descrevem a estrutura de determinado tipo de registro. Os registros são descritos em relações de conjuntos onde são estabelecidas as ligações lógicas entre eles.

O esquema abaixo representa este tipo de Ligação

Fábrica

#1 Nome Local ... Apontada Aponta\_Início Aponta\_Final

Injeção

#7 Nome Máquina ... Apontada (\*1) Aponta\_I(\*15) Aponta\_F(\*18)

#15 Paulo 28 (Idade) ... (\*7) (\*17)

#18 João 25 ... (\*17) (\*7)

Um confusão habitualmente verificada, diz respeito a confusão que existe entre o conceito do Modelo de Redes e o existente na matemática. No modelo de Redes temos sempre um elemento distitivo, o registro base e a partir dele são dispostos os demais registros. Temos sempre *tipos de conjunto*, que dispõe de três elementos, a saber: nome, tipo de registro pai e tipo de registro filho. Supondo um Registro contido no Arquivo de Disciplinas ministradas na Íbero, este seria um registro pai, na medida em que conteria a referência aos seus registros filhos (os alunos cursando aquela disciplina).

As restrições impostas pelo Modelo de Redes podem ser descritas como de ordem de Entrada e de Existência. Em relação as restrições de entrada citamos a obrigatoriedade de cada novo registro estar conectado (ou apontado, como preferem os programadores C) ao conjunto indicado. Em relação a restrições de Existência podemos dizer que um componente de um tipo de registro pode existir de forma independente de outros desde que esteja conectado a algum outro registro fazendo parte de algum conjunto, ou sendo base de um novo conjunto. A identificação de um conjunto pode ser verificada através do esquema de ligação entre o registro pai e o registro filho, assim sendo, cada instância de conjunto apresenta um elemento de distinção, o tal registro pai, e os registros filhos devidamente ordenados, e portanto passíveis de serem acessados pelos seus elementos.

Exemplo: Disciplina Tópicos Avançados e seus Alunos

Registro de Disciplinas

Informática

Álvaro

Maurício

Cláudio

Registro de Alunos

O exemplo anterior representa uma instância de conjunto, no caso Disciplinas (Tópicos Avançados) e seus alunos (no caso Álvaro, Amorim e Cláudio).

## Banco de Dados Orientados ao Objeto

Representam os dados como coleções que obedecem a propriedades. São modelos geralmente conceituais dispondo de pouquíssimas aplicações reais. Neste Modelo não seria interessante a existência de uma tabela de funcionários e dentro dela alguma referência para cada registro, de forma a podermos saber onde (em que departamento) o funcionário está alocado. Um conjunto de regras disponibilizaria em separado os funcionários da fábrica, que, no entanto, estariam agrupados aos demais, para o sistema de folha de pagamento.

## Banco de Dados Relacional

O Modelo de Dados relacional representa os dados contidos em um Banco de Dados através de relações. Estas relações contêm informações sobre as entidades representadas e seus relacionamentos. O Modelo Relacional, é

claramente baseado no conceito de matrizes, onde as chamadas linhas (das matrizes) seriam os registros e as colunas (das matrizes) seriam os campos. Os nomes das tabelas e dos campos são de fundamental importância para nossa compreensão entre o que estamos armazenando, onde estamos armazenando e qual a relação existente entre os dados armazenados.

Cada linha de nossa relação será chamada de *TUPLA* e cada coluna de nossa relação será chamada de *ATRIBUTO*. O conjunto de valores passíveis de serem assumidos por um atributo, será intitulado de *DOMÍNIO*.

Estes tópicos serão estudados cuidadosamente na disciplina Análise de Sistemas, que se incumbirá de apresentar cuidadosamente regras e normas para elaboração destes modelos.

Em nosso curso, voltado à construção prática dos Bancos de Dados, e não de sua construção teóricas, apenas citaremos os aspectos básicos da construção teórica, de forma a facilitar ao estudante o relacionamento que existe entre Análise de Sistemas e Banco de Dados.

O domínio consiste em um grupo de valores atômicos a partir dos quais um ou mais atributos retiram seus valores reais. Assim sendo Rio de Janeiro, Paraná e Pará são estados válidos para o Brasil, enquanto Corrientes não é um estado válido (pertence a Argentina e não ao Brasil).

O esquema de uma relação, nada mais são que os campos (colunas) existentes em uma tabela. Já a instância da relação consiste no conjunto de valores que cada atributo assume em um determinado instante. Portanto, os dados armazenados no Banco de Dados, são formados pelas instâncias das relações.

As relações não podem ser duplicadas (não podem existir dois estados do Pará, no conjunto de estados brasileiros, por exemplo), a ordem de entrada de dados no Banco de Dados não deverá ter qualquer importância para as relações, no que concerne ao seu tratamento. Os atributos deverão ser atômicos, isto é, não são passiveis de novas divisões.

Chamaremos de Chave Primária ao Atributo que definir um registro, dentre uma coleção de registros. Chave Secundária (Terceária, etc), serão chaves que possibilitarão pesquisas ou ordenações alternativas, ou seja, diferentes da ordem criada a partir da chave primária ou da ordenação natural (física) da tabela. Chamaremos de Chave Composta, aquela chave que contém mais de um atributo (Por exemplo um cadastro ordenado alfabeticamente por Estado, Cidade e Nome do Cliente, necessitaria de uma chave composta que contivesse

estes três atributos). Chamaremos de Chave Estrangeira, aquela chave que permitir a ligação lógica entre uma tabela (onde ela se encontra) com outra na qual ele é chave primária.

**Exemplo**:

Cidade Estado

\* CidCodi \* EstCodi

CidNome EstNome

EstCodi (E)

CidCodi e EstCodi, são chaves primárias respectivamente das tabelas Cidade e Estado, enquanto EstCodi é chave estrangeira na tabela de cidades. É precisamente por este campo (atributo, ou coluna), que será estabelecida a relação entre as tabelas Cidade-->Estado.

# 