

# Unidade 1

## CONCEITOS BÁSICOS DE GERENCIAMENTO DE BANCO DE DADOS

1.1	Introdução	2
1.2	Motivações para Sistemas de Banco de Dados	4
1.3	Vantagens da Abordagem de Banco de Dados	7
1.4	Principais componentes de um Sistema de Banco de Dados	8
	Bibliografia	12

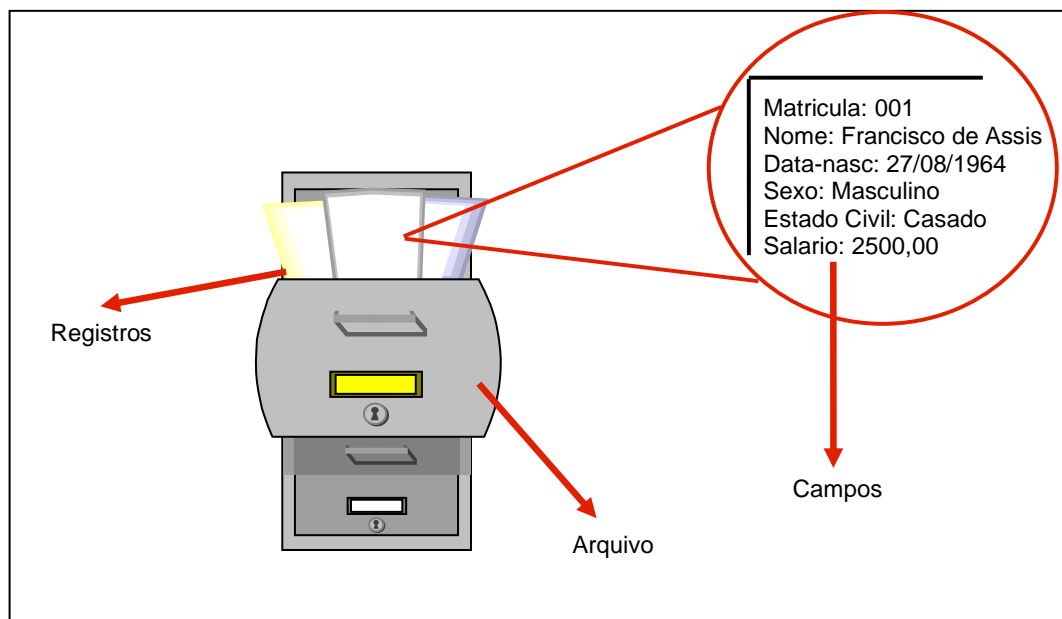
Nesse capítulo você aprenderá os principais conceitos relacionados as origens do Sistemas Gerenciadores de Banco de Dados (SGBD) a partir do conhecimento dos problemas que existiam no ambiente de computação pré-SGBD. Verá as vantagens que um ambiente de processamento de dados poderá ter ao adotar uma abordagem de Banco de Dados. Conhecerá os principais componentes para um ambiente de Banco de Dados, destacando especialmente o papel dos usuários que passam a atuar nesse ambiente. E por fim, conhecerá como a presença dos SGBD permitiu a separação entre os dados e os programas aplicativos, propiciando assim a Independência de Dados.

## CAPITULO 1 – CONCEITOS BÁSICOS DE BANCO DE DADOS

### 1.1 INTRODUÇÃO

#### Conceitos Básicos

Num passado não muito distante, muitas empresas armazenavam os dados referentes aos clientes, funcionários, fornecedores, produtos, compras, vendas, etc em documentos ou fichas que ficavam guardadas em armários denominados fichários. Estes armários normalmente possuíam varias gavetas, local onde efetivamente as fichas eram arquivadas. Tais armários recebiam o nome de **arquivo** e cada ficha armazenada dentro dele era denominada **registro** ou **cadastro** do objeto/entidade equivalente. Olhando o formato de cada ficha, perceberíamos um elenco de espaços a serem preenchidos, e cada um desses espaços era denominado **campo**. Observem que, os termos destacados em negrito acima são os mesmos termos utilizados na computação. Mais do que isso, os termos tanto na computação como nos métodos manuais de armazenamento, possuem o mesmo sentido.



**FIGURA1.1** Armazenamento de dados não computadorizado

Porém, devido às particularidades do armazenamento computadorizado, alguns novos elementos teóricos foram incorporados aos termos tradicionais, e deram origem a um elenco de denominações que não só facilitam a comunicação entre as pessoas envolvidas, mas também, auxiliam na padronização e melhoria da qualidade desses dados. Por esta razão, e para termos um linguajar comum durante a leitura desse material, destacamos abaixo um conjunto de termos utilizados na computação, relativos ao armazenamento de dados.

**Campo:** Um campo é representado por um conjunto de bytes. Dentro de um campo é possível armazenarmos uma unidade básica de informação. Tal informação pode ser identificada através de um nome e formatada tanto com relação a quantidade de bytes necessários (tamanho) quanto com relação ao tipo de conteúdo a ser armazenado (formato). (FEDELI, 2003, p.49,50)

Ex.: Matricula, Nome, Data\_Nasc, Sexo, Estado Civil, Salário, etc.

**Registro:** Registro é um conjunto de unidades de informação ou campo logicamente relacionadas (FARRER, et. all. 1999, p.148). Na formatação de um registro, cada campo pode ter um tipo de dado diferente, ou seja, no exemplo abaixo, o campo matrícula pode ser do tipo numérico ao passo que o campo nome pode ser do tipo caracter. (ASCENCIO, CAMPOS, 2002 p.215). No exemplo abaixo os elementos Matrícula, Nome, Data\_nasc, Sexo, Estado Civil e Salário correspondem ao cabeçalho dos dados (a estrutura ou esquema do registro), enquanto os valores ocupados em cada espaço abaixo dos elementos citados acima (001, Francisco de Assis, 27/08/1964, Masculino, Casado, 2500.00) corresponde as Instâncias ou Ocorrências de cada registro, que por sua vez, irão representar um determinado individuo (objeto) dentro do conjunto. Cada uma dessas instâncias representa um registro dentro do arquivo. O esquema de um registro possui informações relativas ao nome de cada campo, o tipo de dado que o campo pode receber e a quantidade de bytes (deslocamento) de cada campo.

Ex.:

Matrícula	Nome	Data_nasc	Sexo	Estado civil	Salário
001	Francisco de Assis	27/08/1964	Masc	Casado	2500,00

**Arquivo:** Um arquivo equivale a um conjunto de registros de um mesmo tipo. Isso quer dizer que somente elementos do mundo real que atendam aos requisitos definidos no esquema do registro (Layout) poderão ter seus valores armazenados nos campos que formam tal registro.

Ex.: Arquivo de Empregados. Coleção de todos os registros de todos os empregados.

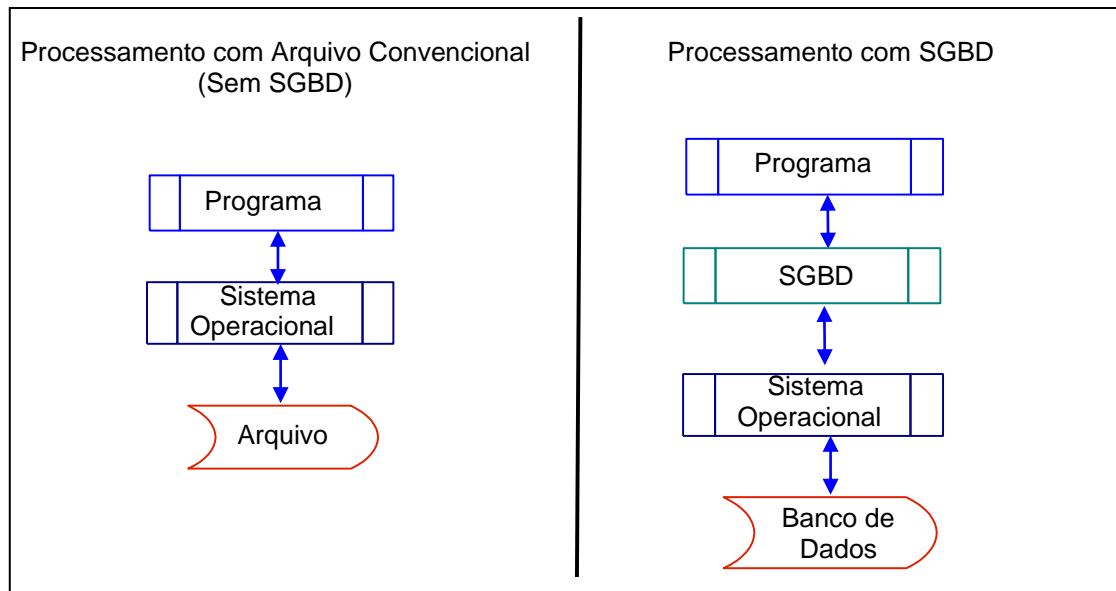
Matrícula	Nome	Data_nasc	Sexo	Estado civil	Salário
001	Francisco de Assis	27/08/1964	Masc	Casado	2500,00
002	Ana Paula Araújo	14/05/1973	Fem	Solteira	1300,00
003	Marcos Antonio Seixas	25/02/1960	Masc	Solteiro	4500,00
004	Maria de Nazaré Fernandes	06/12/1978	Fem	Solteira	950,00

**Banco de Dados:** Um banco de dados é um conjunto de arquivos relacionados entre si. Porém não são arquivos aleatórios gerados ao acaso, mas sim, um conjunto de arquivos que representam algum aspecto, algum ambiente do mundo real, também chamado de minimundo. Deve ser uma coleção lógica com algum significado inerente, projetado, construído e populado com dados que atendam a um objetivo. Além disso, deve ter um conjunto de usuários que dele façam uso tanto para consultas quanto para alimentar seus dados e que utilizam alguma aplicação desenvolvida para manuseá-lo (ELMASRI/NAVATHE, 2005, p.1).

Ex.: Tomemos como exemplo um Sistema de Controle de Pessoal. Um Banco de Dados envolveria possivelmente um conjunto de arquivos tais como: Empregados, Dependentes, Pagamentos, Benefícios, etc.

**Sistema de Gerência de Banco de Dados:** Corresponde a um conjunto de programas para administrar o Banco de Dados. A partir da adoção de um SGBD, os Sistemas Aplicativos<sup>1</sup> passam a não mais ter acesso aos dados do Banco de Dados diretamente. Todas as chamadas passam através do SGBD, ou seja, sempre que um programa de aplicação necessitar de ter acesso a algum arquivo, controlado pelo SGBD, terá necessariamente que se comunicar com este para aí sim, obter acesso a tal dado. Como poderão observar na figura 1.2 sem a utilização de um SGBD os programas aplicativos interceptam os dados, tendo somente o Sistema Operacional como intermediário. Já com o uso de SGBD, há uma nova camada de Software, que isola a intervenção das aplicações sobre os dados.

<sup>1</sup> Sistemas desenvolvidos internamente nas corporações ou comprados de fornecedores externos que são utilizados para o desempenho de diversas atividades dentro de uma organização. Ex.: Sistemas de Contas a Pagar, Contabilidade, etc.



**FIGURA1.2** Comparativo entre processamento sem SGBD e com SGBD

**Sistema de Banco de Dados:** Segundo DATE (2003, p.6) trata-se de um sistema computadorizado que armazena informação e as disponibiliza para que seus usuários busquem e atualizem estas informações. Estas informações podem ser qualquer coisa que seja útil para o desempenho das atividades de um indivíduo ou organização.

**Dados e Informação:** Você leitor, não estranhe se usamos os termos **dado** e **informação** de maneira similar. Na verdade, alguns autores preferem distinguir **dado**, como o conteúdo armazenado dentro de um banco de dados, enquanto **informação** equivale ao significado que este dado pode ter para um usuário específico. No entanto, usar um ou outro termo não comprometerá o entendimento geral dos objetivos da disciplina.

## 1.2 MOTIVAÇÕES PARA SISTEMAS DE BANCO DE DADOS

Antes de nos aprofundarmos nas questões associadas ao SGBD, vale a pena resgatarmos algumas dificuldades que encontrávamos no período pré-SGBD, ou seja, no período em que o armazenamento dos dados se dava em arquivos convencionais (SILBERSCHATZ; KORTH; SUDARSHAN; 1999, p.2 p.3):

**Redundância e Inconsistência de dados:** A **redundância**, ou seja, a existência de um mesmo dado em arquivos distintos era causada porque os aplicativos eram desenvolvidos por programadores diferentes durante um longo período de tempo, sem muitas vezes haver uma comunicação entre estes programadores. Além de causar a duplicação dos dados em bases de dados diferentes, aumentava a probabilidade de que estes dados tivessem sido criados com formatos diferentes. Já a **inconsistência** é fruto da redundância, pois, uma vez que um dado poderia estar sendo armazenado em locais distintos, tornava-se mais complexo garantir que as atualizações feitas sobre este dado num determinado local, estariam também sendo feitas em todos os demais locais onde o dado está armazenado.

Ex.: Arquivo de Empregados (Matricula, Nome, Cargo, Salário, Data de Nascimento).  
Arquivos de Pagamentos dos Empregados (Matricula, nome, salário, proventos, descontos)

Nosso exemplo, embora hipotético, retrata uma situação comum nos tempos passados. Como podem observar, temos os campos, nome e salário repetidos em dois arquivos distintos. Alguns de vocês, que já nasceram na geração Banco de Dados Relacional e suas fantásticas ligações entre tabelas devem imaginar impossível uma situação dessas. Mas podem acreditar que isso existiu e não cometeria erro muito grande

em afirmar que se fizemos uma varredura em algumas organizações do Estado, encontraremos ainda empresas que possuem algum sistema rodando nessas condições.

**Dificuldade no acesso aos dados:** Este problema era observado devido à rígida ligação entre dados e programas. Significava que, o leiaute de um arquivo (sua estrutura, os campos com seus formatos e tamanhos) estava definido dentro dos programas que acessavam estes dados. Sempre que você precisasse ler um determinado registro para capturar um campo específico, era necessário capturar todos demais campos que compunham este registro, para depois analisar e trabalhar com o campo específico. Se precisássemos criar novos programas (geralmente se construíam muitos programas de geração de relatórios) era necessário um consumo altíssimo de tempo de escrita do novo programa, especialmente devido aos enormes leiautes dos arquivos. Algumas vezes, garantir a corretude na transcrição desse layout em vários programas envolvia um bom tempo de conferência. Se o mecanismo de armazenamento fosse a fita magnética, o acesso ficava mais complicado ainda devido às limitações de leitura e gravação desse dispositivo (falaremos melhor da fita abaixo, no item h). Eu pensei em colocar um leiaute de programa COBOL como exemplo para vocês, mas imaginei que alguns teriam brotoejas de aversão a esta magnífica linguagem.

**Problemas de Integridade:** Sabemos que os valores dos dados armazenados precisam satisfazer certos tipos de *restrições de consistência*. Assim, não deve ser permitido armazenar conteúdo alfabético dentro de um campo que só aceite caracteres numéricos. As datas também devem seguir certas regras de formação, reconhecendo os valores colocados para Dia, Mês e Ano. Valores Monetários não podem, ou pelo menos, não deveriam aceitar letras ou caracteres especiais. Somam-se a isso, regras sobre o conteúdo do dado armazenado, tal como, data de nascimento de uma pessoa não pode ser maior que a data atual do momento em que está sendo feita a inclusão do dado. Bom, na era pré-SGBD, todas as regras para garantir a integridade sobre os dados estavam escritas dentro dos programas que compunham os Sistemas Aplicativos. Caso alguém se esquecesse de escrevê-las e o programa fosse executado, recebendo uma massa de dados para ser inserido num arquivo, o dado seria inserido sem grandes dificuldades e muitas vezes, completamente corrompido, pois o Sistema Operacional por si só não teria condições de julgar o que estava inconsistente no dado.

**Problema de Atomicidade:** A atomicidade trata da execução completa de um conjunto de ações. Imagine que tenhamos um sistema de transferência de valores entre contas. Imagine que após a retirada do valor da conta devedora ocorre um problema no sistema, antes que este valor seja lançado na conta credora. Observe que este exemplo mostra uma situação na qual, o processamento somente poderia ter sido concluído se débito e crédito ocorressem por completo, ou senão, nenhum deles deveria ser concluído. Infelizmente, esta atividade também estava sob o controle dos programas da Aplicação. Em geral, pela ausência completa de algum mecanismo de Controle de Transação<sup>2</sup>, a garantia de que as modificações sugeridas no exemplo citado acima seriam executadas corretamente deveriam ser previstas e controladas pelos programas de Aplicação.

**Anomalias de acesso concorrente:** Vocês devem se lembrar da caracterização da fita magnética, feita em alguma disciplina estudada anteriormente. Então devem lembrar que a fita trabalhava com processamento monousuário, ou seja, somente um programa podia acessar uma determinada fita por vez. Isso era com certeza, um transtorno. Acesso concorrente é exatamente isso: A possibilidade de duas ou mais programas terem acesso ao mesmo conjunto de dados, nem que seja somente para leitura. Mas este problema não era somente causado pelas fitas magnéticas. Mesmo posteriormente, quando se utilizavam discos o problema ainda persistia, porém, com alguns agravantes. De certa forma, podemos pensar que a fita era limitante, porém, justamente pelo fato de não permitir dois programas acessarem ao mesmo tempo, garantia também que as ações de um programa não causariam danos aos dados sendo acessados ao mesmo tempo por outro programa. Já com os discos, isso ficou mais complicado, pois estes dispositivos permitem o acesso compartilhado dos dados. Assim, aquela proteção que a fita nos proporcionava, se fragilizou nos discos. Com os discos, mais que um programa poderia trabalhar de maneira concorrente sobre o mesmo conjunto de dados. Pior do que isso, pela falta de algum mecanismo de Controle de Concorrência, mais do que um

---

<sup>2</sup> Mecanismo implementado pelos SGBDs que permite o agrupamento, dentro de um único evento de atualização, de modificações que possuam alguma relação lógica.

programa poderia atualizar o mesmo arquivo causando inclusive sobreposições de dados. Bom, podemos perceber que esta situação não poderia se perpetuar.

**Problemas de Segurança:** Nem todo o usuário do sistema deve ter acesso a todos os dados. Sabe-se que num sistema de Recursos Humanos, onde temos os dados cadastrais dos empregados de uma organização, seria bastante desagradável deixar aberto ao acesso de qualquer pessoa, algumas informações pertinentes ao empregado, tais como: resultado de entrevistas, rendimentos, faltas, punições e premiações. Assim, o ideal é que se estabeleçam critérios para que cada pessoa (usuário) só acesse aquilo que lhe é pertinente e que esteja sobre sua responsabilidade, ou seja, do seu poder. Se permitirmos que novos programas sejam adicionados ao sistema de maneira arbitrária, é difícil assegurar tais restrições de segurança. Nessa época, a garantia de proteção sobre os dados era feita por um conjunto de pessoas, que imbuídas da sua responsabilidade, impediam que pessoas não autorizadas acessassem dados indevidamente. Não estamos aqui para desconfiar de ninguém, mas sabe-se que algumas vezes, até mesmo por um descuido ou uma distração desses “vigilantes”, esta segurança poderia ficar comprometida.

**Problema para Compartilhamento dos Dados (Notas da Autora):** Houve uma época, não muito distante desta (em torno da década de 80), na qual as fitas eram o principal meio de armazenamento. Os discos, já existiam e eram essenciais ao processamento dos dados, porém, eram caros demais. Assim, todo (ou pelo menos uma grande parte) o volume dos dados que a empresa possuía era armazenado em fitas magnéticas. Lógico, vocês podem querer me perguntar, mas antes, como era? Bom, sempre há um “antes”, que antecede a um “antes”. Houve sim uma época em que nem as fitas existiam, mas isso é pré-história demais para essa nossa disciplina. Voltando as fitas, reconheço que elas tiveram um papel essencial na evolução do processamento de dados. Mas elas possuíam alguns inconvenientes. O método de acesso e gravação das fitas é seqüencial. Dessa forma, vocês já podem estar imaginando o transtorno que era acessar um registro específico. Se não pensou, não se assuste, mas se você quisesse ler o registro número 1000 teria que fazer uma leitura seqüencial na fita, perguntando registro a registro quem era o registro de número 1000. Isso era muito custoso e algumas vezes, por azar, ele poderia ser o último da lista. A gravação tinha outros inconvenientes. Sempre que você necessitasse gravar ou alterar um registro, era obrigado a transferir todos os dados que estavam na fita a ser alterada para uma nova fita e no ponto correto inserir um novo registro ou alterar um registro já existente. As fitas são um dispositivo **monousuário**, ou seja, somente um usuário poderá fazer uso de uma fita por vez. Além disso, as fitas não ficavam disponíveis para uso imediato. Sempre que alguém necessitasse ler os dados em uma fita específica, era obrigado a procurar o volume desejado, e colocá-lo na unidade de leitura para então ter acesso aos dados armazenados. Imaginem o transtorno que isso causava, já que os CPDs possuíam uma infinidade de volumes guardados em inúmeras estantes, separadas por prateleiras dentro de uma enorme sala.

Com o barateamento dos discos magnéticos, um volume significativo dos dados que ficavam armazenados nas fitas passou a ser armazenado nesses discos. Certamente isso foi um avanço significativo no acesso aos dados. Agora, os arquivos estavam totalmente disponíveis para acesso de quem necessitasse. No armazenamento em disco, o método de acesso também se aperfeiçoou e passamos a ter acesso **randômico**. O acesso **randômico** permite que você encontre um registro específico sem a necessidade de fazer uma leitura seqüencial no arquivo. A gravação também foi totalmente reformulada, permitindo a gravação de um novo registro sem a necessidade de se regravar todo o arquivo. Mas nem tudo foi melhoria. Uma das vantagens proporcionadas pelo disco converteu-se também num transtorno. O fato dos arquivos ficarem armazenados em um dispositivo sempre disponível, sem a necessidade da intervenção humana controlando a execução de programas sobre estes arquivos, permitia que pessoas não autorizadas acessassem tais arquivos de maneira indevida. Nas fitas, pela própria limitação física, que era a intervenção manual de um profissional do CPD para disponibilizá-la, funcionava também como um mecanismo de segurança. Com o armazenamento em disco, os dados passaram a estar disponíveis para o processamento **multiusuário**, o que, se mal controlado, poderá causar perda de integridade causada pela falta de *Controle de Concorrência*<sup>3</sup>.

Para ajudar a gerenciar tudo isso, existia um profissional na informática denominado **Scheduller**. Esta pessoa tinha por responsabilidade, entre diversas outras coisas, preparar a seqüência de processamentos que seriam feitos no CPD. O Schedule funcionava meio que como um *Gerenciador de Tarefas*, pois ele

---

<sup>3</sup> Mecanismo implementado pela maioria dos SGBDs que tem por objetivo controlar as ações de diversos usuários sobre os dados.



conhecia as pessoas, bem como suas responsabilidades e os limites de autorização. Ele sabia qual arquivo estava armazenado em cada fita. Sabia qual versão estava mais atualizada. Conhecia os direitos que as pessoas tinham sobre os dados. De certa forma, parte da segurança e da integridade dos dados estava nas mãos desse profissional.

A maioria dos problemas citados acima foram sanados ou pelo menos minimizados, graças a criação de programas de computador que agregaram para si, boa parte da responsabilidade que antes estava, ou nas mãos dos *scheduller* ou sob a responsabilidade dos programadores. Estes programas foram agrupados num produto denominado SGBD (Sistema Gerenciador de Banco de Dados).

### 1.3 VANTAGENS DA ABORDAGEM DE BANCO DE DADOS

Possivelmente podemos encontrar alguns desenvolvedores que por diversos motivos optam por não utilizar SGBD no desenvolvimento e uso de suas aplicações. No entanto, essa não é a regra já que a presença desse software favorece a um conjunto de benefícios conforme descritos abaixo (DATE, 2003, p.16).

#### **Compartilhamento dos Dados:**

Não somente as aplicações existentes podem compartilhar os dados no banco de dados, mas também, novas aplicações podem ser desenvolvidas e fazerem uso desses mesmos dados.

#### **Redução da Redundância:**

Adotando o SGBD para apoiar a administração das estruturas de dados, torna mais difícil a dispersão de dados em arquivos distintos e principalmente com formatos incompatíveis. Isso também é beneficiado com a adoção de práticas de Modelagem de Dados, orientado por profissionais especializados, como o AD e o DBA. Isso não significa que toda a redundância possa ou deva necessariamente ser eliminada. Em geral, sugere-se que toda redundância deva ser controlada. Algumas vezes, há motivos técnicos plausíveis para manter várias cópias distintas dos mesmos dados.

#### **Redução da Inconsistência:**

Certamente que, a redução da redundância amenizará se não conseguir eliminar, a inconsistência causada pela não atualização dos dados dispersos. Porém, conforme citado no tópico anterior, algumas redundâncias são propositais, e nesse caso, deve-se tomar cuidado especial para que as atualizações sejam sincronizadas, evitando a incompatibilidade futura dos dados.

#### **Permite suporte a transação:**

Uma transação é uma unidade lógica de trabalho que envolve várias operações no banco de dados, e em particular, operações de modificação. O objetivo maior do controle de transação é garantir a integridade sobre os dados. A qualidade de uma transação pode ser medida pelas propriedades ACID:

- **Atomicidade** → Define que todas as operações contempladas dentro de uma transação ou são totalmente feitas ou nenhuma operação é realizada.
- **Consistência** → Se antes do início da transação os dados eram consistentes, estes deverão também estar consistentes ao final da transação.
- **Isolamento** → As operações realizadas dentro de uma transação não devem interferir nas operações de outra transação, ainda que estejam executando em paralelo sobre o mesmo conjunto de dados.
- **Durabilidade** → Uma vez confirmada à transação, as modificações geradas pela mesma devem se tornar permanente.

**Manutenção da Integridade:**

O controle centralizado do banco de dados pode ajudar a minimizar os problemas relativos à qualidade dos dados e especialmente a integridade. Certamente que a redução da redundância é um dos fatores que colabora para a efetiva melhoria dessa qualidade. Vale lembrar que a integridade de dados é ainda mais importante em um ambiente de banco de dados do que em um ambiente de arquivos convencionais, justamente pelo compartilhamento que estes dados sofrem. Se estiverem incorretos, as consequências serão mais abrangentes.

**Reforço na segurança:**

O DBA, através das orientações recebidas do AD pode assegurar controles mais restritos de acesso aos dados. Mecanismos como o uso de *LOGINS* e *PASSWORDs* privadas podem garantir acessos distintos aos diversos usuários de um banco de dados. Estes mecanismos em conjunto com facilidades disponibilizadas pelos SGBD podem controlar níveis de acessos específicos (leitura, exclusão, inclusão) a cada item de informação no banco de dados. Percebe-se aqui, que pela própria centralização e compartilhamento dos dados, os níveis de segurança devam ser mais rigorosos do que nos métodos convencionais, já que a corrupção de uma base de dados afeta um elenco muito maior de usuários.

**Equilíbrio sobre requisitos contraditórios:**

Já que os requisitos agora são tratados no nível de organização e não mais no nível de usuários específicos, o DBA em conjunto com o AD tem condições de oferecer um serviço global que reflita as necessidades da organização como um todo e não em partes isoladas. Uma visão corporativa para alavancar, sob o ponto de vista dos dados, estruturas de dados que sirvam para diversos setores da empresa e que por esta razão, mereçam tratamento diferenciado.

**Padronizações:**

A centralização pode impulsionar a padronização das representações dos dados. Isso afeta desde a fase de modelagem conceitual, até na formatação das estruturas físicas do banco de dados. A padronização é muito importante especialmente desejável como auxílio ao *intercâmbio de dados*, ou a migração de dados entre sistemas. Além disso, a padronização da nomenclatura e documentação dos dados facilita a comunicação entre as diversas pessoas envolvidas com estes dados.

Além das vantagens citadas acima, dedicamos especial atenção ao item Independência de Dados, que se configura não só como uma vantagem para a adoção do SGBD, mas como um objetivo a ser alcançado por estes.

## **1.4 PRINCIPAIS COMPONENTES DE UM SISTEMA DE BANCO DE DADOS**

Um Sistema de Banco de Dados é composto de (DATE, 2003, p.7,8,9):

**Dados**

Os **Dados** são a parte mais importante do Banco de Dados. Eles podem ser utilizados em ambiente MONOUSUÁRIO e MULTIUSUÁRIO. No ambiente MONOUSUÁRIO somente um usuário tem acesso ao Banco de Dados por vez, ao passo que no ambiente MULTIUSUÁRIO vários usuários podem ter acesso ao Banco de Dados ao mesmo tempo em ambientes MULTIUSUÁRIO, o número de usuários simultâneos não é ilimitado. O máximo de usuários em um mesmo momento, utilizando o Banco de Dados (Conexões) pode ser determinado pelo número de licenças adquiridas do fornecedor do SGBD. No entanto, o ideal é que o usuário que esteja trabalhando em um ambiente compartilhado tenha a impressão que está interagindo sozinho com o Banco de Dados.



Os dados do Banco de Dados estarão ao mesmo tempo, INTEGRADOS e COMPARTILHADOS. São INTEGRADOS, pois o usuário trabalha com os dados como se fosse um único arquivo, ainda que os dados estejam espalhados em vários arquivos. Por sua vez, são também COMPARTILHADOS, uma vez que um mesmo dado pode ser utilizado por mais que um usuário ao mesmo tempo (DATE, 2003, p.7)

Observe, no entanto, que a integração dos dados vai além dos poderes de um software. O que quero dizer é que, o uso de um ambiente de Banco de Dados, com todas as inovações tecnológicas advindas do mesmo poderá não ser tão eficiente se não houver uma mudança de postura dos profissionais que elaboram e estruturam estes dados. Se continuarmos a criar arquivos arbitrariamente sem nos preocuparmos com dados já existentes e que podem ser compartilhados. Se não tivermos zelo na formatação dos tipos e tamanhos associados aos dados, dificilmente um SGBD terá a inteligência suficiente para “julgar” se o dado está correto quanto a regras de negócio ou necessidades do usuário.

Na unidade 4 estudaremos a Modelagem de Dados, na qual, teremos a oportunidade de visualizarmos o compartilhamento dos dados entre as estruturas.

Outra característica da Integração e Compartilhamento dos dados, é que usuários distintos estarão interessados em porções distintas dos dados armazenados e não necessariamente do Banco de Dados todo. Assim, cada usuário terá visões específicas sobre os dados armazenados, parecendo até, que são arquivos distintos.

## Hardware

Os componentes de Hardware consistem em:

- a) Discos Magnéticos, usados para guardar os dados armazenados;
- b) Dispositivos de Entrada/Saída
- c) Controladores de Dispositivos
- d) Canais de Entrada/Saída
- e) Processadores de Hardware
- f) Memória Principal

A análise do Hardware necessário para Banco de Dados, foge do escopo deste estudo.

## Software

Entre os dados fisicamente armazenados e o usuário que interagem com estes, existe uma camada de Software que trabalha como um interprete ou vigilante desses dados. Toda e qualquer tentativa do usuário de chegar até os dados armazenados, será interceptada por este vigilante, que após analisar a requisição, decidirá se permite ou não o acesso. Esta camada, denominada SGBD (Sistema de Gerenciamento de Banco de Dados) é um conjunto de programas que agem sobre aos arquivos armazenados. Atualmente, existem vários tipos de SGBD, fornecidos por diversos fabricantes, dentre eles citamos: ORACLE, SYBASE, SQL SERVER, DB2, ADABAS, MySQL, INFORMIX<sup>4</sup>, etc. O SGBD é o componente mais importante de todo o ambiente “Banco de Dados”. Quando dizemos que estamos adquirindo um Banco de Dados, na verdade, estamos adquirindo um SGBD (um desses listados acima), ou seja, um conjunto de programas, que juntamente com a estrutura de armazenamento permanente (Discos Magnéticos) permitirá ao adquirente (cliente/empresa) armazenar os seus dados e gerenciá-los.

## Usuários

Em geral, qualquer pessoa que utilize um Banco de Dados pode ser denominada usuário, claro. Mas, para efeito de separação das tarefas que cada um desempenha, bem como do poder de acesso que cada uma

---

<sup>4</sup> Os Bancos de Dados citados acima fazem parte da geração RDB (Relacional Database).

dessas pessoas tem sobre os dados, diferenciamos, basicamente 3 tipos de usuários. Partindo do que tem mais para o que tem menos poder, definimos:

a) **AD (Administrador de Dados) e DBA (Administrador de Banco de Dados):** Sabemos da relevância dos dados para uma empresa. Dessa maneira, deve haver dentro da empresa, alguma pessoa que seja responsável pela qualidade desses dados. Esta pessoa é denominada AD (Administrador de Dados). Podemos dizer que os principais conhecimentos requeridos de um AD estão ligados mais ao funcionamento da organização, do que a questões técnicas de um ou outro SGBD. É claro que algum conhecimento sobre as limitações e potencialidade de um SGBD facilitará o desempenho de suas atividades, mas a ausência desses conhecimentos poderá ser suprida pelo DBA. As responsabilidades de um AD podem ser resumidas em (DATE, 2003, p.9,16):

- i. Decidir que dados devem ser armazenados no banco de dados;
- ii. Estabelecer normas para tratar e manter estes dados. Uma das normas pode se referir ao controle dos acessos sobre esses dados, ou seja, normas de segurança.

O DBA é um profissional de TI (Tecnologia da Informação). Tem por finalidade criar o Banco de Dados e implementar os controles técnicos necessários de maneira a por em prática as normas e decisões definidas pelo AD (DATE, 2003, p.16). Tem o maior nível de poder, para interagir com o SGBD. Trata da Administração do Ambiente, de maneira a dar suporte ao desenvolvimento dos Sistemas Aplicativos feitos pelos **Analistas de Sistemas e Programadores**, bem como, favorecendo a utilização e continuidade dos sistemas utilizados pelos **Usuários Finais**. É o responsável pelo controle geral do sistema em um nível técnico.

Dentre essas funções citamos (DATE, 2003, p.36):

- **Definir o Esquema Conceitual:** Certamente que a responsabilidade sobre as entidades de negócio que deverão ser armazenadas é do AD. Porém, depois dessa elaboração conceitual, é o DBA que terá a responsabilidade de materializar esta elaboração, através de instruções DDL<sup>5</sup>, dando forma assim ao Esquema Conceitual correspondente ao Modelo Conceitual elaborado pelo AD. Maiores informações sobre Modelagem de Dados, tanto conceitual como lógica, poderá ser obtida na unidade 4. Alguns autores denominam a fase de elaboração conceitual feita pelo AD como **Modelagem Conceitual de Dados**, enquanto a etapa de tradução desse modelo para o banco de dados de **Modelagem Lógica de Dados**.
- **Definir o Esquema Interno:** Esse processo é denominado **Projeto Físico de Dados** preocupa-se como serão representados os dados no banco de dados fisicamente. Depois de elaborado o modelo lógico, a preocupação volta-se para as estruturas de dados do produto (no caso, do SGBD) que suportarão os dados projetados pelo AD. Nessa fase, o DBA também deve estar atendo para os mapeamentos conceitual/interno associado.
- **Contato com os usuários:** A fim de garantir a disponibilidade dos dados de que estes necessitam. Certamente o DBA deve apoiar a construção ou construir as estruturas externas que fornecerão acesso aos dados armazenados.
- **Definir restrições de segurança e integridade:** Pode ser considerada parte do esquema conceitual. A DDL conceitual incluirá os recursos para a especificação de tais controles. Citaremos brevemente alguns exemplos de integridade, que certamente ficarão mais bem esclarecidas ao longo desse material: Integridade da Entidade, Integridade Referencial, etc. Quanto à segurança, podemos controlá-la através do SGBD usando restrições impostas sobre o acesso e manipulação dos dados nas tabelas. Isso pode ser obtido através do comando DDL GRANT<sup>6</sup>.
- **Definir normas de descarga e recarga:** É de suma importância, na eventualidade de danos à parte do banco de dados causados por erro humano ou por falha de hardware, fazer retornar os dados envolvidos com um mínimo de demora e com as menores consequências ao restante do sistema. O

<sup>5</sup> DDL (Data Definition Language). Linguagem de Definição de Dados utilizada para construção das estruturas de dados dentro de um SGBD. É uma das subdivisões da linguagem SQL.

<sup>6</sup> GRANT é um comando da linguagem SQL, utilizado para atribuir direito de acesso aos dados para usuários específicos.

DBA deve definir e implementar uma estratégia de recuperação apropriada envolvendo, por exemplo, o descarregamento periódico do banco de dados na memória auxiliar de armazenamento (backup) e procedimentos para recarregá-lo, quando necessário.

- **Monitorar o desempenho e responder a requisitos de mudanças:** O DBA deve organizar o sistema de tal maneira que obtenha “o melhor desempenho para a empresa”; e efetuar os ajustes adequados quanto às necessidades de modificações.

b) **Analista de Sistemas e Programadores:** Interagem com o Banco de Dados, elaborando Sistemas Aplicativos, que serão utilizados pelos **Usuários Finais**. Para tanto, os Analistas e Programadores utilizam-se de ferramentas de desenvolvimento disponíveis no mercado (Linguagem de Programação, CASE, etc), que juntamente com o SGBD possibilitarão a elaboração de Sistemas Aplicativos. Este tipo de usuário deve possuir conhecimento de SQL<sup>7</sup>. Em ambiente de menor porte, o Analista de Sistema poderá agregar também as funções de DBA.

c) **Usuários Finais:** Este usuário interage com o SGBD através dos Sistemas Aplicativos elaborados pelos Analistas e Programadores. Normalmente, o conhecimento sobre o Banco de Dados que os Usuários Finais possuem, está diretamente relacionado às Interfaces (Telas) que lhe são fornecidas para manipulação (Leituras e Gravações) dos dados armazenados. Alguns usuários podem ainda interagir com o SGBD através de outras interfaces fornecidas pelos próprios fabricantes dos SGBDs. Estas interfaces podem ser definidas como **interfaces acionadas por menus ou formulários** e **interfaces acionadas por comando**. No primeiro caso, os usuários operam as requisições ao banco de dados através do preenchimento de itens de menus ou caixas de formulários. No segundo caso, o usuário emite comandos como SELECT e INSERT interativamente ao SGBD. De qualquer forma, a utilização de uma dessas duas interfaces exige certos conhecimentos de TI (Tecnologia da Informação) desses usuários o que algumas vezes dificulta o seu uso.

---

<sup>7</sup> SQL (*Structured Query Language*), que é a linguagem utilizada para comunicação com o Banco de Dados Relacional.

## BIBLIOGRAFIA

ASCENCIO, A. F. G., CAMPOS, E. A. V. de **Fundamentos da Programação de Computadores**. São Paulo: Prentice Hall, 2002.

DATE, C.J. **Introdução a Sistemas de Banco de Dados**. 8ed. Rio de Janeiro: Elsevier, 2003.

ELMASRI, R., NAVATHE, S. B. **Sistemas de Banco de Dados**. 4ed. São Paulo: Addison Wesley, 2005.

FARRER, H. At All. **Algoritmos Estruturados**. Rio de Janeiro: LTC, 1999.

FEDELI, R. D., POLLONI, E. G., PERES, F. E. **Introdução a Ciência da Computação**. São Paulo: Pioneira Thomson Learning, 2003.

KORTH, H. F. & SILBERSCHATZ, A. **Sistema de Bancos de Dados**. 3ed. São Paulo: Makron Books, 1999.

SETZER, W. W. **Banco de Dados**. 3ed. São Paulo: Edgard Blucher, 1999.