

Ao final deste assunto, você será capaz de:

- Conhecer os conceitos básicos de sistemas de banco de dados;
- Identificar se um sistema de armazenamento de dados é um sistema de banco de dados ou um sistema tradicional de arquivos;
- Identificar os profissionais que trabalham na área de banco de dados e suas principais funções.

O primeiro uso do computador se deu dentro o meio acadêmico, mas o seu desenvolvimento começou a acontecer apenas com a utilização dessa máquina para fins bélicos. A partir do momento em que se despertou para a utilização do computador como uma máquina que poderia ser utilizada para automatizar procedimentos comerciais (rotinas administrativas das empresas), buscou-se representar a execução dessas rotinas através de programas que pudessem calcular ou responder pelos procedimentos em questão, registrando as informações contidas nos documentos transitados nas diversas áreas da empresa, bem como gerando novos documentos com informações tratadas a partir daqueles dados iniciais. Os dados que esses programas utilizavam só existiam em função dos mesmos, quer dizer, cada programa possuía seus dados, que podiam ser os mesmos de outros programas.

1.1.1. Gerenciamento de Dados

Inicialmente, as linguagens de programação eram as responsáveis, com seus pequenos recursos, pelo gerenciamento e armazenamento dos dados, bem como pelo seu processamento. O surgimento dos métodos de acesso veio, dentro desta fase, revolucionar o desenvolvimento, pois estes apresentavam uma maneira mais eficiente de recuperar os dados armazenados. De maneira geral, estes métodos eram fornecidos pelos fabricantes de hardware com seus respectivos sistemas operacionais. Esta "facilidade" proporcionou uma padronização na recuperação de dados, no tocante a sua programação, pois a aplicação já não dependia mais da perícia do programador para preencher seus requisitos.

Para melhor compreender o gerenciamento de dados, é necessário conhecer alguns termos, relacionados a seguir:

- Arquivo: Um arquivo é um conjunto de informações sobre determinado assunto ou negócio. Poderíamos dizer também que arquivo é uma coleção de registros de uma mesma natureza ou tipo. Ora, esses arquivos precisam estar organizados com conteúdos de relevância e classificados de forma racional para que possa constituir de fato um conjunto de informações, caso contrário não passará de um lote de lixo.
- Registro: Um item de um arquivo que represente o tipo de conteúdo do arquivo. Por exemplo, se considerarmos um arquivo de funcionários de uma empresa, as informações de um funcionário constituirão o registro do funcionário.
- Item de dado ou campo: Cada informação que consta desse registro. Cada uma dessas informações constitui um item de dado ou campo do registro. Por exemplo, os funcionários de uma empresa podem compor o seu arquivo de funcionários. Cada funcionário teria, então, um registro dentro desse arquivo, de modo que cada registro teria as informações particulares desse funcionário. Logo, cada uma das informações sobre um funcionário poderia ser representada por um campo, como nome, endereço, telefone, etc.

1.1.2. Sistema Tradicional de Arquivos

São aqueles em que os dados do sistema estão armazenados fisicamente separados uns dos outros. O acesso é feito pelos programas de aplicação, associando o nome externo dos arquivos e definindo todo o registro independente da utilização dos campos.

Esse tipo de estrutura pode ser representado pela figura a seguir:

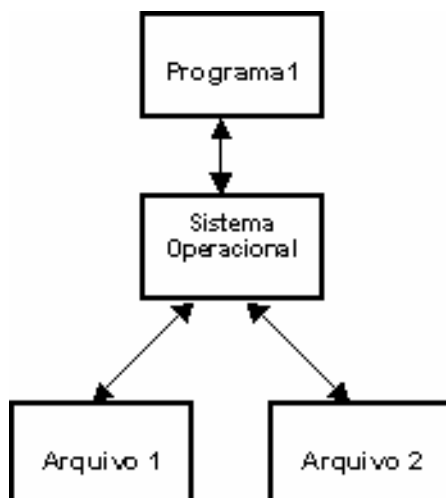


Fig. 1.1 1 - Sistema Tradicional de Arquivos.

Ou seja, nesse tipo de estrutura a informação sobre o objetivo dos arquivos e para que servem os campos que o compõe está definida, normalmente, no programa que os criou e, por conseguinte, apenas o programador que fez esse programa sabe o que contém esses arquivos. A documentação sobre esse programa normalmente não existe ou reside esquecida numa gaveta, quando não está completamente desatualizada.

A falta de documentação pública, ou seja, os arquivos que existem no computador estão definidos nos programas que os criaram, faz com que a informação sobre os arquivos que residem naquele computador esteja dispersa entre as várias pessoas que fizeram os programas para criação dos arquivos. Isso gera um problema, pois, quando a informação está concentrada demais em uma única pessoa, esta ganha poder e quando a mesma não está disponível, a informação sobre os arquivos não existe. Por outro lado, o fato de termos os vários arquivos criados por vários programas, portanto, por vários programadores, implica em dispersão excessiva da informação.

Tais atitudes levaram a criação de disfunções como:

- a) O usuário considerava-se o DONO do sistema. Cada setor era o proprietário dos dados e informações guardadas em seus arquivos.

Consequência: grande burocracia e demora na obtenção de informações.

- b) Sendo a empresa um conjunto de subsistemas dependentes entre si, as aplicações instaladas em cada departamento, sob a ótica imediatista, geravam problemas de integração entre dados e procedimentos.

Consequência: criação de sistemas de interface, tanto em nível de programas quanto em nível de procedimentos.

- c) A repetição de dados, de forma descontrolada, criava a inconsistência de informações entre os sistemas e, portanto, entre usuários de um mesmo tipo.

Consequência: desconfiança quanto à validade dos sistemas e entre os próprios usuários.

Resumindo, os principais problemas com o Sistema de Arquivos são:

- **Dados fortemente acoplados a programas;**
- Informação sobre arquivos dispersa;
- Informação sobre arquivos desatualizada ou inexistente;
- Grande tendência à redundância de dados¹;
- Possibilidade de inconsistência de dados.

1.1.3. Sistemas de Banco de Dados

A idéia mais comum, para pessoas leigas em processamento de dados e técnicos diretamente não envolvidos com o assunto, é que o sistema de banco de dados reúne e guarda todos os dados da organização em algum tipo de armazenamento de massa, sendo operado por um conjunto de programas, responsáveis pela manutenção dos dados.

Vejamos, agora, uma definição mais completa para o tema: "Banco de Dados é um conjunto de dados estruturado, definido conforme o Modelo de Dado adotado, acrescido das regras de relação de dependência existente entre eles" (DATE, C.J.,

¹ Redundância de Dados: a falta de controle sobre o conteúdo e objetivo dos arquivos faz com que sejam criados arquivos similares ou com o mesmo teor de conteúdo de arquivos que já existam, implicando, portanto em redundância ou repetição de dados entre os vários arquivos.

Introdução aos Sistemas de Banco de Dados). Esta conceituação é boa, pois induz a uma definição dos dados com base em regras de estruturação (Normalização, por exemplo) e relacionamento.

Segundo C. J. Date, um dos mais conhecidos autores de livros sobre a teoria dos Sistemas Gerenciadores de Bancos de Dados, "um banco de dados é um depósito de dados armazenados, sendo geralmente integrado e compartilhado".

- Por integrado, queremos dizer que o banco de dados reúne vários arquivos num único, com o objetivo de eliminar a redundância de arquivos.
- Por compartilhado, entendemos como sendo a capacidade de compartilhar os dados armazenados dentre diversos usuários mesmo que concorrentemente, ou seja, ao mesmo tempo.

Uma outra conceituação poderia ser: banco de dados consiste de um conjunto de arquivos relacionados entre si. Existem alguns pesquisadores que consideram Banco de Dados como sendo um conjunto de bases de dados. Neste estudo, no entanto, vamos considerar os dois termos equivalentes.

Mas, por que Banco de Dados? Porque, com o uso de banco de dados, a administração da informação fica centralizada, não cabendo mais à aplicação, individualmente, possuir tabelas, discos e fitas. Conseqüentemente, deverá existir na empresa uma pessoa identificável que se responsabilizará pela administração dos dados armazenados no banco de dados. Esta pessoa é o Administrador de Banco de Dados, também chamado de DBA. Mais adiante discutiremos a figura do DBA.

1.1.4. Comparação com a Abordagem Tradicional de Arquivos

O sistema de banco de dados surgiu com o objetivo de resolver os problemas decorrentes do uso do sistema tradicional de arquivos.

A seguir apresentamos um quadro comparativo entre os dois sistemas:

<i>Sistema tradicional de arquivos</i>	<i>Sistema de banco de dados</i>
<p>Dados fortemente acoplados a programas</p> <p>Informação sobre arquivos dispersa</p> <p>Informação sobre arquivos desatualizada ou inexistente</p>	<p><u>Independência dos Dados:</u> representa a forma física de armazenamento dos dados no Banco de Dados e a recuperação das informações pelos programas de aplicação. Esta recuperação deverá ser totalmente independente da maneira com que os dados estão fisicamente armazenados. Quando um programa retira ou inclui dados no SGBD compacta-os para que haja um menor consumo de espaço no disco. Este conhecimento do formato de armazenamento do campo é totalmente transparente para o usuário. A independência dos dados permite os seguintes recursos:</p> <ul style="list-style-type: none"> • Os programas de aplicação definem apenas os campos que serão utilizados independentemente da estrutura interna dos arquivos; • Quando há inclusão de novos campos no arquivo, será feita manutenção apenas nos programas que utilizam esses campos, não sendo necessário modificar os demais programas. <p>Obs.: Nos sistemas tradicionais este tipo de operação requer a alteração no layout (definição de arquivos dentro dos programas) de todos os programas do sistema que utilizam o arquivo.</p>
Grande tendência à redundância de dados	<p><u>Redução ou Eliminação de Redundâncias:</u> possibilita a eliminação de dados privativos de</p>

<i>Sistema tradicional de arquivos</i>	<i>Sistema de banco de dados</i>
	cada sistema. Os dados, que eventualmente são comuns a mais de um sistema, são compartilhados por eles, permitindo o acesso a uma única informação sendo consultada por vários sistemas.
Possibilidade de inconsistência de dados	<u><i>Eliminação de Inconsistências</i></u> : através do armazenamento da informação em um único local com acesso descentralizado e, sendo compartilhada com vários sistemas, os usuários estarão utilizando uma informação confiável. A inconsistência ocorre quando um mesmo campo tem valores diferentes em sistemas diferentes. Exemplo, o estado civil de uma pessoa é solteiro em um sistema e casado em outro. Isto ocorre porque esta pessoa atualizou o campo em um sistema e não o atualizou em outro. Quando o dado é armazenado em um único local e compartilhado pelos sistemas, este problema não ocorre ou, pelo menos, é mais difícil ocorrer.

Podemos considerar mais algumas vantagens desse controle centralizado:

- *Compartilhamento dos Dados*: permite a utilização simultânea (concorrente) e segura de um dado, por mais de uma aplicação ou usuário, independente da operação que esteja sendo realizada. Deve ser observado apenas o processo de atualização concorrente, para não gerar erros de processamento (atualizar simultaneamente o mesmo campo do mesmo registro). Os aplicativos são por natureza multiusuários.

- *Padronização dos Dados*: permite que os campos armazenados na base de dados sejam padronizados segundo um determinado formato de armazenamento (padronização de tabela, conteúdo de campos, etc.) e ao nome de variáveis seguindo critérios padrões pré-estabelecidos pela empresa. Ex. Para o campo "Sexo" somente será permitido armazenamento dos conteúdos "M" ou "F".
- *Restrições de Segurança*: define para cada usuário o nível de acesso a ele concedido (leitura, leitura e gravação ou sem acesso) ao arquivo e/ou campo. Este recurso impede que pessoas não autorizadas utilizem ou atualizem um determinado arquivo ou campo.
- *Manutenção de Integridade*: exige que o conteúdo dos dados armazenadas no Banco de Dados possua valores coerentes ao objetivo do campo, não permitindo que valores absurdos sejam cadastrados. Exemplo: Um funcionário que faça no mês 500 horas extras, ou um aluno que tenha nascido no ano de 1860.
- *Evitar Necessidades Conflitantes*: representa a capacidade que o administrador de Banco de Dados deve ter para solucionar "prioridades sempre altas" de todos os sistemas, tendo ele que avaliar a real necessidade de cada sistema para a empresa para priorizar a sua implantação.

1.1.5. Sistema Gerenciador de Banco de Dados (SGBD)

Conceitos básicos

“Um SGBD é uma coleção de programas que permite ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades”.

Corresponde a um conjunto de funções-programas utilizadas para administrar o Banco de Dados. Assim, os programas de uma aplicação não acessam diretamente os dados armazenados em uma Base. Qualquer solicitação para recuperação de dados passa pelo SGBD.

Podemos representar essa estrutura de dados pela figura a seguir:

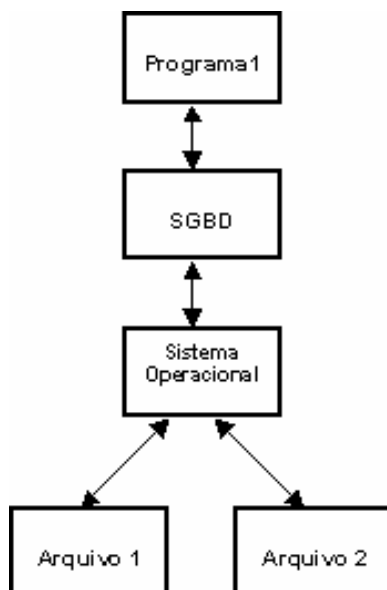


Fig. 1.1 2 - Sistema Gerenciador de Banco de Dados.

Ao observar a figura, verificamos o aparecimento de um elemento estranho entre o programa e o sistema operacional, denominado **SGBD**² (**Sistema Gerenciador de Banco de Dados**). O SGBD é um programa produzido comercialmente por um fabricante de *software*, o qual passa a fazer a gerência dos bancos de dados, ou seja, os dados contidos nos arquivos e a informação sobre o objetivo do banco, arquivos e campos, agora, passam a ser do SGBD. Portanto, podemos tecer as seguintes considerações:

- Como esse programa é de uso geral e não mais específico de uma aplicação, as informações sobre o banco de dados estarão centralizadas no SGBD e, portanto, disponíveis para quem quiser consultar. Como os arquivos não estão mais definidos nos programas, podemos dizer que estes são independentes de programas.
- Como as definições dos bancos de dados estão todas a serviço do

²A denominação DBMS aparece em algumas publicações sobre bancos de dados. Esta nada mais representa que as iniciais em inglês de Sistema Gerenciador de Banco de Dados, ou seja, **Data Base Management System**.

gerenciador de banco de dados, então a possibilidade de redundância de dados diminui substancialmente.

Veja dois exemplos a seguir que ajudam a entender a diferença entre o acesso a dados via o sistema de arquivos e o acesso via o SGBD:

- Quando um cliente entra numa loja de departamentos, ele vai até as diversas seções e escolhe os seus produtos e, só quando tem os produtos escolhidos, os leva ao caixa para pagamento. Este exemplo pode ser usado para demonstrar o funcionamento do acesso ao dado via o sistema de arquivos, ou seja, o cliente é o responsável pela pesquisa e separação do item desejado.
- Quando um cliente entra numa loja de venda de autopeças, dirige-se ao balcão e apresenta ao vendedor uma lista de itens desejados. O balconista fica responsável pela pesquisa e entrega dos itens ao cliente. Este exemplo pode ser usado para demonstrar o funcionamento do acesso ao dado via o SGBD.

Características operacionais do SGBD ⁽⁴⁾

O SGBD tem 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 existente 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 o invalida 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. 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 os aplicativos 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.

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 dispõem 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 abrimos mão de uma Interface Amigável, de um Gerencialmente Automático de Backups ou ainda de outras características que não julgamos fundamentais, para nos livrarmos do problema típico de ambiente extremamente comprometido, por mau desempenho do Banco de Dados.

A escolha do Banco de Dados da empresa, portanto é uma decisão muito delicada, na medida em que está poderá 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 os investimentos realizados em *Software* e *Hardware*.

Componentes Funcionais de um SGBD

Já foi observado que o SGBD não é apenas um programa, mas um conjunto de programas, cada um com uma importante função, de modo que esse conjunto possa gerenciar os bancos de dados sob responsabilidade do SGBD. Portanto, os componentes do SGBD são os seguintes:

- Gerenciador de Arquivos: *gerencia a alocação de espaço e armazenamento em disco e estruturas de dados.*
- Gerenciador do Banco de Dados: *proporciona interface entre os dados de*

baixo nível e os programas de aplicação e consultas.

- Processador de Consultas: traduz comandos de uma linguagem de consulta em instruções de baixo nível para que o Gerenciador do Banco de Dados entenda. Tenta aperfeiçoar os pedidos de consulta dos usuários.
- Pré-Compilador DML: compila comandos DML em rotinas da linguagem do host. Precisa interagir com o processador de consultas para gerar código apropriado.
- Compilador DDL: converte comandos DDL em um conjunto de tabelas contendo metadados, que são armazenados no DD.
- Arquivo de Dados: armazenam os dados propriamente ditos.
- Dicionário de Dados: armazena informações sobre a estrutura do banco de dados.
- Índices: proporcionam acesso rápido aos itens de dados com valores específicos.

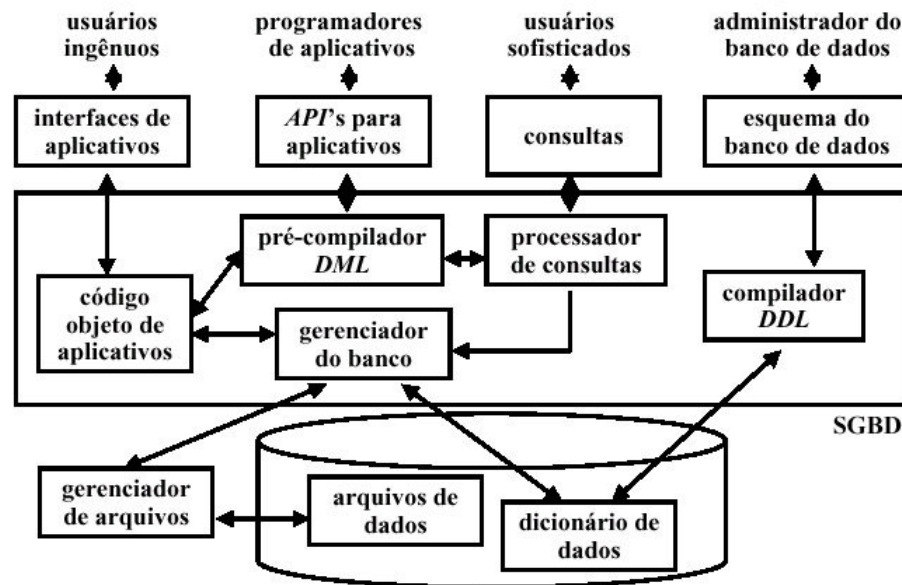


Fig. 1.1 3 - Componentes funcionais do SGBD ⁽³⁾.

³ Korth, H.F. e Silberschatz, A.; Sistemas de Bancos de Dados, Makron Books, 2a. edição revisada, 1994.

1.1.6. Comparação entre Sistema Gerenciador de Banco de Dado (SGBD) e Gerenciador de Arquivo (GA) ⁽⁴⁾

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

Sistemas baseados em "Banco de Dados" e embasados 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.

Regras básicas

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: Auto-Contenção - 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

⁴ SURIAN, Prof. Jorge & NICOCHELLI, Prof. Luiz. Apostila de Banco de Dados e SQL, 1997.

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 for necessário *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 ⁽⁵⁾ - Um SGBD deve gerenciar completamente a

⁵ As propriedades ACID das Transações: Apropriadamente implementadas, as transações normalmente atendem ao “teste ACID”, onde:

- **A** significa **atomicidade**, a execução das transações é total ou nenhuma;

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 de modificações simultâneas e uma instrução capaz de cancelar uma série de modificações. Por exemplo, imaginemos que estejamos cadastrando um pedido para um cliente, que este deseje reservar cinco 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,

-
- **C** significa **consistência**. Isto é, todos os bancos de dados têm restrição de consistência ou informação sobre os relacionamentos entre os elementos de dados. Espera-se que as transações preservem a consistência do banco de dados.
 - **I** significa **isolamento**, o fato de que cada transação deve parecer que está sendo executada como se nenhuma outra transação estivesse sendo executada;
 - **D** significa **durabilidade**, a condição que determina que se uma transação nunca deve ser perdida uma vez que tenha sido completada.
-

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 da aplicação, você não está lidando com um SGBD.

Conclusão

Um SGBD deve obedecer INTEGRALMENTE as seis regras acima. Em caso contrário estaremos diante de um GA ou de um "quase" SGBD. Claro, que os sistemas gerenciadores de banco de dados foram evoluindo até conseguirem alcançar todas essas regras e os primeiros produzidos falharam em uma ou outra regra. Portanto, essa posição rígida não é aceita por todos os autores.

Considerações Finais

Atualmente, existe uma tendência de mercado em se dizer que qualquer problema será resolvido, caso a empresa adquira um Banco de Dados. Naturalmente, em um ambiente com acesso constante ao Banco de Dados (acesso concorrente, obviamente), onde a segurança seja de vital importância e que o desempenho da aplicação escrita estiver comprometendo a empresa, considerando-se logicamente uma aplicação bem escrita, sem dúvida a aquisição de um Banco de Dados *poderá ser* o primeiro passo na solução do problema.

Analogamente ao que ocorreu com o aparecimento das primeiras linguagens de programação voltadas ao Windows, onde estas foram apresentadas como capazes de alavancar os negócios da empresa, e no geral causaram mais frustração do que solução, a aquisição do Banco de Dados, pode gerar o mesmo tipo de problema.

É fundamental que a empresa candidata a utilizar um Banco de Dados se normalize totalmente, pois soluções "quebra-galho", típicas do ambiente que dispõe de um Gerenciador de Arquivo, tendem a ser impossíveis em um ambiente estruturado sobre o Banco de Dados. Portanto, sob pena de se realizar um grande investimento, e não se colher fruto algum, é muito conveniente, que a empresa *antes* de adquirir um Banco de Dados, passe por um processo de adaptação, preferencialmente contando com pessoal especializado, geralmente consultores, que *não* tenham qualquer ligação com fabricantes de Bancos de Dados.

1.1.7. Esquemas e Instâncias

Estes são enfoques que representam duas formas de se ver um banco de dados. A seguir apresentamos um conceito sobre cada um desses enfoques:

- **Esquema**: Em qualquer modelo de dados utilizado, é importante distinguir a “descrição” do banco de dados do “banco de dados” por si próprio. A descrição de um banco de dados é chamada de “esquema de um banco de dados” e é especificada durante o projeto do banco de dados. Geralmente, poucas mudanças ocorrem no esquema do banco de dados. É o “projeto geral” (estrutura) do banco de dados. Tem as seguintes características:
 - Não muda com frequência;
 - Há um esquema para cada nível de abstração e um subesquema para cada visão de usuário.
- **Instância**: Os dados armazenados em um banco de dados em um determinado instante do tempo formam um conjunto chamado de “instância do banco de dados”. A instância altera toda vez que uma alteração no banco de dados é feita.

O SGBD é responsável por garantir que toda instância do banco de dados satisfaça ao esquema do banco de dados, respeitando sua estrutura e

suas restrições.

O conceito de instância 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 armazenadas e controladas pelo SGBD.

1.1.8. Os profissionais de Banco de Dados e suas principais funções ⁽⁶⁾

O sistema gerenciador de banco de dados é uma ferramenta indispensável para a gestão dos bancos de dados utilizados por qualquer instituição. Entretanto, um software, qualquer que seja, não tem conhecimento ou consciência para determinar que tipo de uso ou que competência deve ser dada aos usuários de um banco de dados. É necessário que exista uma pessoa ou um grupo de pessoas que se responsabilizem pelo controle dos bancos de dados. Estes são os Administradores de Banco de Dados.

Existem dois grupos de administradores de dados, os quais são apresentados a seguir:

- Administrador de Dados (Data Administrator - DA)
 - Desenvolve e administra centralizadamente estratégias, procedimentos, práticas e planos capazes de disponibilizar os dados corporativos necessários, quando necessários, com integridade, privacidade, documentação e compartilhamento.
 - Participa dos levantamentos de dados, e regras de negócio da empresa. Elabora e/ou acompanha a confecção de modelos.
 - Participa da compatibilização do planejamento de sistemas com os

⁶ BITTENCOURT, Rogério Gonçalves. Apostila de BANCO DE DADOS-I – 5º. Semestre, Ciência da Computação, FACEX – Faculdade de Ciências Exatas, UNIVALI – Universidade do Vale do Itajaí, 1999.

modelos lógicos.

- Participa de pesquisa de softwares de apoio, relacionados à área de AD, assim como SGBD.
- É responsável pela definição e atualização do esquema do banco de dados.

- Administrador do SGBD (Data Base Administrator - DBA)

- Definição da estrutura de armazenamento e a estratégia (ou método) de acesso:
 - O DBA tem que decidir também como os dados deverão ser representados no banco de dados, e precisa especificar esta representação escrevendo a definição da estrutura de dados (usando a linguagem de definição interna de dados). Além disso, têm também que ser especificados os mapeamentos associados entre a definição da estrutura de dados e o esquema conceitual. Na prática, provavelmente a DDL interna ou a DDL conceitual incluirão os meios para especificação desse mapeamento, mas as diferentes definições devem estar claramente separadas. Da mesma forma que com o esquema conceitual, o esquema interno e o mapeamento correspondente existirão tanto no formato fonte como no formato objeto.
- Concessão de autorização para acesso a dados;
- Definição de controles de integridade;
- Definição de estratégias para cópia de segurança e recuperação:
 - A partir do momento em que a empresa começa efetivamente a se basear em bancos de dados, ela se torna dependente de forma crítica da operação bem sucedida desse sistema. Caso ocorra avaria a alguma porção do banco de dados - causada por erro humano, digamos, ou por falha no *hardware* ou no sistema operacional de suporte -, é

essencial que se possam reparar os dados envolvidos com um mínimo de atraso, bem como causar o menor efeito possível ao restante do sistema. (Por exemplo, não deve ser afetada de nenhuma forma a disponibilidade dos dados que *não* sofreram avaria.) O DBA tem que definir e implementar uma estratégia apropriada de recuperação, envolvendo, por exemplo, a descarga periódica do banco de dados para uma fita de *back-up* e procedimentos para recarregar porções relevantes do banco de dados a partir dessa fita.

- Monitoramento do desempenho:
 - O DBA é responsável tanto pela organização do sistema como por obter o desempenho que seja “o melhor para a empresa”, e por fazer os ajustes necessários quando há mudanças de requisitos. Quaisquer mudanças nos detalhes de armazenamento e acesso têm que ser seguidas por mudanças correspondentes na definição do mapeamento do armazenamento, de tal forma que o esquema conceitual possa permanecer constante.
 - É claro que o DBA irá precisar de diversos programas utilitários para ajudá-lo em suas tarefas. Esses utilitários devem ser uma parte essencial de um sistema prático de banco de dados. Abaixo encontramos alguns exemplos dos tipos de utilitários necessários:
 - Rotinas de carga (para criar a versão inicial do banco de dados).
 - Rotinas de reorganização (para, por exemplo, rearrumar o banco de dados e reutilizar espaço ocupado por dados obsoletos).
 - Rotinas de controle de uso (para anotar cada operação feita no banco de dados, juntamente com a informação sobre o usuário que a realizou e os

registros dos estados anterior e posterior).

- Rotinas de recuperação (para restaurar o banco de dado a um estado anterior depois de uma falha de *hardware* ou de programação).
 - Rotinas de análise estatística (para ajudá-lo na monitoração do desempenho).
- Execução de rotinas de desempenho;
 - Modificação da organização física.

ATIVIDADE

Responda as questões a seguir:

1. Conceitue Arquivo.
2. Cite dois problemas decorrentes do uso do Tradicional Sistema de Arquivos.
3. Conceitue Sistema de Banco de Dados.
4. Cite uma característica do sistema de banco de dados que o tornou o seu uso mais interessante que o Tradicional Sistema de Arquivos.
5. Conceitue Sistema Gerenciador de Banco de Dados.
6. Qual a diferença entre esquema e instância de um banco de dados?
7. Cite uma função do DBA.
8. Qual perfil profissional mais se aproxima do Analista Desenvolvedor de Sistemas, o DA ou o DBA? Justifique.