

# 1. Conceitos de Banco de Dados

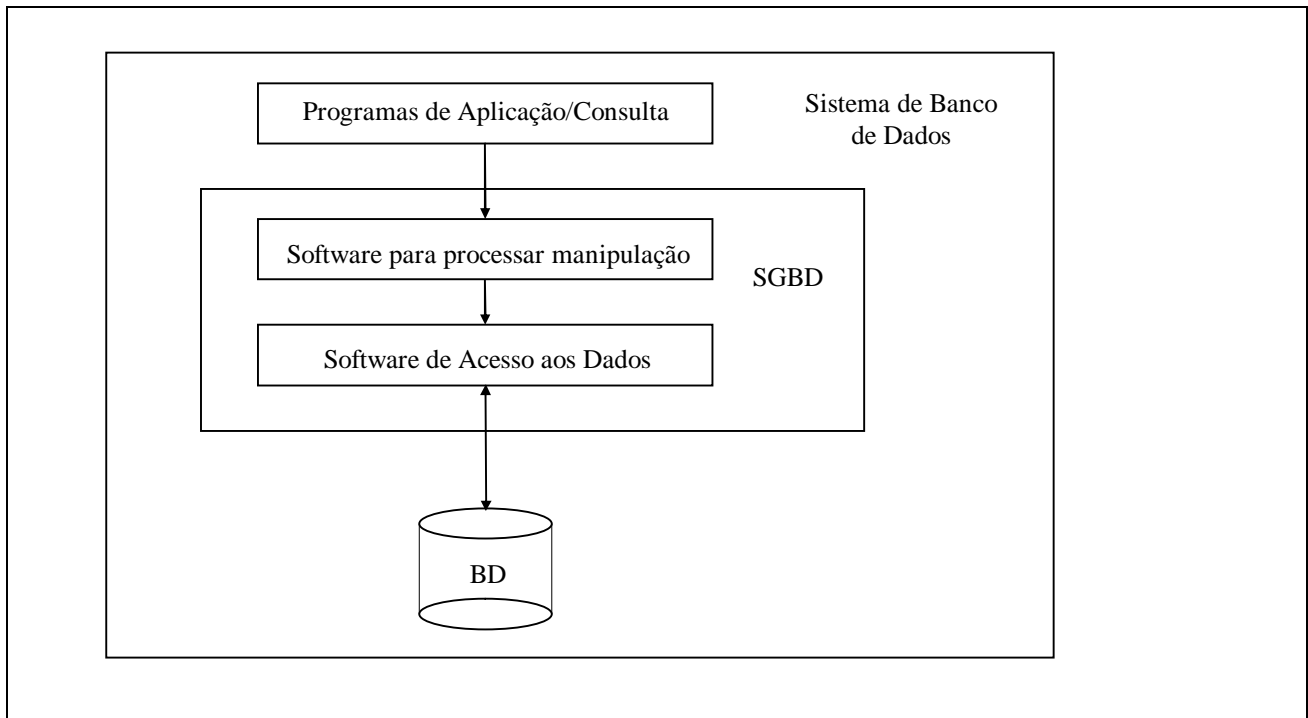
A tecnologia aplicada aos métodos de armazenamento de informações vem crescendo e gerando um impacto cada vez maior no uso de computadores, em qualquer área em que os mesmos podem ser aplicados. Para tanto, ao se aventurar nesta área de Banco de Dados, faz-se necessário, num primeiro momento, conhecer algumas definições basilares que permeiam esta temática.

## 1.1. Definições

- i) **Dado** – é um “fato conhecido”, que pode ser armazenado e que possui um significado. Por exemplo: 01/01/2008 é a representação de uma data.
- ii) **Informação** – é o dado, que foi processado por um determinado sistema, e que tem algum valor para tomadas de decisões.
- iii) **Banco de dados (BD) ou Database (DB)** – “(...) pode ser definido como um conjunto de dados devidamente relacionados” (MACHADO, 2004:20). Desta forma, segundo o mesmo autor, um banco de dados possui as seguintes propriedades:
  - É uma coleção lógica coerente de dados com um significado inerente; uma disposição desordenada dos dados não pode ser referenciada como um banco de dados.
  - Ele é projetado, construído e populado com valores de dados para um propósito específico; um banco de dados possui um conjunto predefinido de usuários e aplicações.
  - Ele representa algum aspecto do mundo real, o qual é chamado de minimundo<sup>1</sup>; qualquer alteração efetuada no minimundo é automaticamente refletida no banco de dados.
- iv) **SGBD** (Sistema de Gerenciamento de Banco de Dados) – consiste em uma coleção de dados inter-relacionados e um conjunto de programas para acessá-los (KORTH e SILBERSCHATZ, 1999). O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados.
- v) **Sistema de Banco de Dados** – corresponde ao conjunto formado por um banco de dados mais as aplicações (programas de aplicação/consulta + SGBD) que manipulam o mesmo, conforme pode ser visualizado na Figura 1.1.

---

<sup>1</sup> Minimundo - “Porção da realidade, captada pelo analista, a qual a gestão de negócios de um a organização tem interesse em observar, controlar e administrar. A complexidade existente no momento de analisar um minimundo pode levar o analista a dividi-lo em partes menores, às quais damos o nome de visão do processo de negócio” (MACHADO, 2004:20).



**Figura 1.1** - Um ambiente de Sistema de Banco de Dados

**Fonte:** Adaptado de FCT (1997:3)

## 1.2. Usuários

Quando se pensa no desenvolvimento de uma aplicação, utilizando-se de banco de dados, necessariamente existe certo número de pessoas envolvidas, desde a análise, projeto, implantação até manutenção. Todavia, vale um parêntese – nem sempre uma organização dispõe de todos estes profissionais, e assim, algumas pessoas passam a desempenhar mais de uma função.

### 1.2.1. Administrador de Banco de Dados (DBA)

O DBA é a pessoa (ou grupo de pessoas) responsável pelo controle do sistema de banco de dados, isto é, realiza tarefas como a autorização de acesso ao banco de dados, a coordenação e o monitoramento de seu uso. Para Date (1990:38-40), as responsabilidades do DBA incluem as seguintes:

- decidir que informações manter no banco de dados;
- decidir a estrutura de armazenamento e a estratégia de acesso;
- servir de elo de ligação com o usuários;
- definir os controles de segurança e integridade;
- definir a estratégia de recuperação do banco de dados;
- monitorar o desempenho e atender as necessidades de modificações;
- rotinas de carga (para criar uma versão inicial do banco de dados a partir de um ou mais arquivos);
- rotinas de despejo na memória e recuperação;
- rotinas de reorganização dos dados no banco de dados, em vista de diversas razões de desempenho;
- rotinas estatísticas, visando, por exemplo, o monitoramento do desempenho do banco de dados; e
- rotinas de análise das estatísticas.

### 1.2.2. Projetista de Banco de Dados

O Projetista de Banco de Dados é responsável pela identificação dos dados que devem ser armazenados no banco de dados, escolhendo a estrutura correta para representar e armazenar dados. Muitas vezes, os projetistas de banco de dados atuam como “staff” do DBA, assumindo outras responsabilidades após a construção do banco de dados. É função do projetista também avaliar as necessidades de cada grupo de usuários para definir as visões que serão necessárias, integrando-as, fazendo com que o banco de dados seja capaz de atender a todas as necessidades dos usuários.

### 1.2.3. Usuários Finais

Existem basicamente três categorias de usuários finais que são os usuários finais do banco de dados, fazendo consultas, atualizações e gerando documentos:

- usuários casuais: acessam o banco de dados casualmente, mas que podem necessitar de diferentes informações a cada acesso; utilizam sofisticadas linguagens de consulta para especificar suas necessidades;
- usuários novatos ou paramétricos: utilizam porções pré-definidas do banco de dados, utilizando consultas preestabelecidas que já foram exaustivamente testadas;
- usuários sofisticados: são usuários que estão familiarizados com o SGBD e realizam consultas complexas.

### 1.2.4. Analistas de Sistemas e Programadores de Aplicações

Os analistas determinam os requisitos dos usuários finais e desenvolvem especificações para transações que atendam estes requisitos, e os programadores implementam estas especificações como programas, testando, depurando, documentando e dando manutenção no mesmo. É importante que, tanto analistas quanto programadores, estejam a par dos recursos oferecidos pelo SGBD.

## 1.3. Vantagens e desvantagens do uso de um SGBD

### 1.3.1. Controle de Redundância

No processamento tradicional de arquivos, cada grupo de usuários deve manter seu próprio conjunto de arquivos e dados. Desta forma, acaba ocorrendo redundâncias que prejudicam o sistema com problemas como:

- toda vez que for necessário atualizar um arquivo de um grupo, então todos os grupos devem ser atualizados para manter a integridade dos dados no ambiente como um todo;
- a redundância desnecessária de dados levam ao armazenamento excessivo de informações, ocupando espaço que poderia estar sendo utilizado com outras informações.

### 1.3.2. Compartilhamento de Dados

Um SGBD multi-usuário deve permitir que múltiplos usuários acessem o banco de dados ao mesmo tempo. Este fator é essencial para que múltiplas aplicações integradas possam acessar o banco.

O SGBD multi-usuário deve manter o controle de concorrência para assegurar que o resultado de atualizações sejam corretos. Um banco de dados multi-usuários deve fornecer recursos para a construção de múltiplas visões.

### 1.3.3. Restrição a Acesso não Autorizado

Um SGBD deve fornecer um subsistema de autorização e segurança, o qual é utilizado pelo DBA para criar “contas” e especificar as restrições destas contas; o controle de restrições se aplica tanto ao acesso aos dados quanto ao uso de softwares inerentes ao SGBD.

### 1.3.4. Representação de Relacionamentos Complexos entre Dados

Um banco de dados pode incluir uma variedade de dados que estão interrelacionados de várias formas. Um SGBD deve fornecer recursos para se representar uma grande variedade de relacionamentos entre os dados, bem como, recuperar e atualizar os dados de maneira prática e eficiente.

### 1.3.5. Tolerância a Falhas

Um SGBD deve fornecer recursos para recuperação de falhas tanto de software quanto de hardware.

### 1.3.6. Quando não Utilizar um SGBD

Em algumas situações, o uso de um SGBD pode representar uma carga desnecessária aos custos quando comparado à abordagem processamento tradicional de arquivos como por exemplo:

- alto investimento inicial na compra de software e hardware adicionais;
- generalidade que um SGBD fornece na definição e processamento de dados;
- sobrecarga na provisão de controle de segurança, controle de concorrência, recuperação e integração de funções.

Problemas adicionais podem surgir caso os projetistas de banco de dados ou os administradores de banco de dados não elaborem os projetos corretamente ou se as aplicações não são implementadas de forma apropriada. Se o DBA não administrar o banco de dados de forma apropriada, tanto a segurança quanto a integridade dos sistemas podem ser comprometidas. A sobrecarga causada pelo uso de um SGBD e a má administração justificam a utilização da abordagem processamento tradicional de arquivos em casos como:

- o banco de dados e as aplicações são simples, bem definidas e não se espera mudanças no projeto;
- a necessidade de processamento em tempo real de certas aplicações, que são terrivelmente prejudicadas pela sobrecarga causada pelo uso de um SGBD;
- não haverá múltiplo acesso ao banco de dados.

## 1.4. Arquitetura/Visões do Banco de Dados

### 1.4.1. Abstração de Dados

Um dos grandes objetivos de um Sistema de Banco de Dados é prover aos usuários uma visão abstrata dos dados, ou seja, o sistema omite detalhes de como os dados são armazenados e mantidos no banco de dados.

Existem três níveis de abstração de dados que compõem a arquitetura de um banco de dados: nível físico (ou interno), nível lógico (ou conceitual) e nível de visão (ou externo).

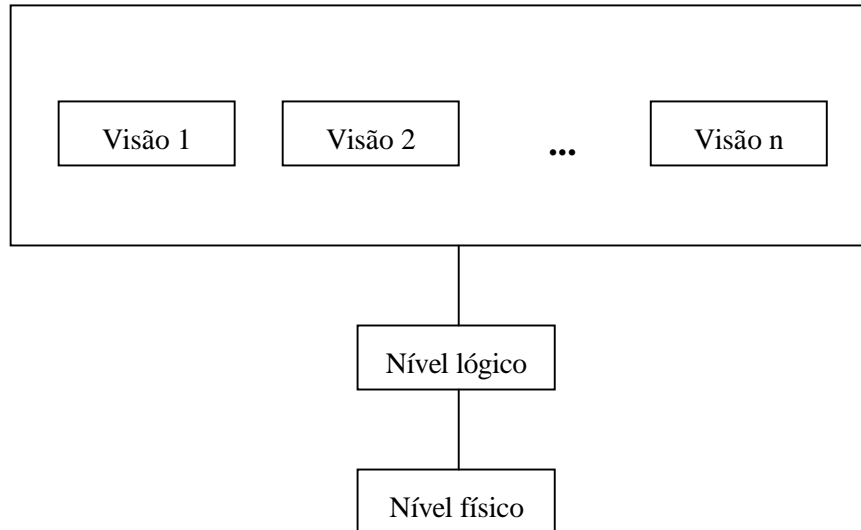
### 1.4.2. Arquitetura do Banco de Dados

Como descrito no item anterior, a arquitetura de um banco de dados divide-se em três níveis gerais:

- i) **Nível físico ou interno** – descreve *como* os dados estão realmente armazenados no banco de dados, compreendendo o nível mais baixo de abstração;

- ii) **Nível lógico ou conceitual** – descreve *quais* dados estão armazenados no banco de dados e quais os inter-relacionamentos entre eles, compreendendo o nível intermediários de abstração; é utilizado pelos DBAs para a definição de quais informações devem ser mantidas no banco de dados;
- iii) **Nível de visão ou externo** – descreve parte do banco de dados, pois está direcionado aos usuários finais do banco de dados, que normalmente precisam conhecer apenas parte das informações armazenadas no banco de dados.

A inter-relação entre esses três níveis de abstração está ilustrada na Figura 1.2.



**Figura 1.2** – Os três níveis de abstração de dados

Fonte: Silberschatz *et al.* (1999:5)

Para facilitar a compreensão entre esses três níveis de abstração, será tomado como exemplo uma analogia com o conceito de tipos de dados em linguagens de programação.

Nas linguagens semelhantes ao Pascal, um registro pode ser declarado da seguinte forma:

```

Type cliente = record
    nome_cliente: string;
    seguro_social: string;
    rua_cliente: string;
    cidade_cliente: string;
end;
    
```

**Nível físico:** um registro de cliente pode ser descrito como um bloco consecutivo de memória (ex.: bytes);

**Nível conceitual:** o cliente é um registro, que possui quatro campos *nome\_cliente*, *seguro\_social*, *rua\_cliente* e *cidade\_cliente* – todos do tipo *string*;

**Nível visões:** cada usuário da empresa tem a visão dos dados que lhe interessa; por exemplo, um usuário do setor de RH pode ter acesso ao *nome\_cliente* e *seguro\_social*.

## 1.5. Instâncias e Esquemas do Banco de Dados

**Instância:** é uma coleção de informações armazenadas no banco de dados em um determinado momento; assim, como o banco de dados muda a todo o momento em que é realizada uma operação de inserção, exclusão ou alteração, a instância do banco de dados também é alterada.

**Esquema:** é o projeto geral do banco de dados, ou seja, corresponde à estrutura do banco de dados (tabelas, campos, tipos de dados, entre outros); por isso não muda com frequência.

Segundo Silberschatz *et al.* (1999), os sistemas de banco de dados apresentam diversos esquemas, subdivididos de acordo com os níveis de abstração já discutidos: o esquema físico (nível mais baixo), o esquema lógico (nível intermediário) e os subesquemas (no nível mais alto).

## 1.6. Independência de Dados

É a habilidade de modificar a definição de um esquema de um nível, sem afetar a definição de um esquema de nível superior. Existem dois níveis de independência de dados:

- **Independência de dados lógica:** é a capacidade de alterar o esquema conceitual sem ter que alterar o esquema externo ou as aplicações do usuário; um exemplo de uma alteração no nível lógico é a mudança de um tamanho de um campo – necessidade de aumentar o número de casas decimais em um campo float;
- **Independência de dados física:** é a capacidade de alterar o esquema interno sem ter que alterar o esquema conceitual, o esquema externo ou as aplicações do usuário; um exemplo de uma alteração no nível físico é alguma modificação para aprimorar o desempenho do banco de dados.

## 1.7. Linguagens de acesso a um SGBD

Existe uma linguagem específica para definir o esquema conceitual e físico de um banco de dados relacional. A linguagem utilizada é a **Linguagem de Consulta Estruturada** (*Structured Query Language - SQL*), que apesar de existir um padrão definido, varia de acordo com o SGBD que se esteja trabalhando.

A SQL pode ser dividida em 3 grupos:

- **Linguagem de Definição de Dados (Data Definition Language - DDL):** linguagem responsável pela definição dos objetos de armazenamento de dados no SGBD; como exemplo pode-se citar os comandos para definição de tabelas (e seus respectivos campos) que irão compor o banco de dados – *CREATE TABLE* nome\_tabela, *DROP TABLE* nome\_tabela.
- **Linguagem de Manipulação de Dados (Data Manipulation Language - DML):** linguagem responsável pelo comandos de manipulação dos dados em um SGBD; como exemplo pode-se mencionar os comandos de consulta, inclusão, exclusão e atualização de informações no banco de dados – *SELECT*, *INSERT*, *DELETE*, *UPDATE*.
- **Linguagem de Controle dos Dados (Data Control Language - DCL):** linguagem responsável por controlar os aspectos de acesso de usuários ao SGBD; como exemplo pode-se mencionar os comandos *GRANT* (que autoriza ao usuário executar determinadas operações sobre um objeto no SGBD) e *REVOKE* (que remove a autorização do usuário executar determinadas operações sobre um objeto no SGBD).

Obs.: Cada uma destas linguagens será estudada em maiores detalhes, no capítulo SQL.

Os aplicativos são escritos em uma linguagem de programação, que pode ser: DELPHI, VISUAL BASIC, JAVA, C, PASCAL, entre outras. A SQL é utilizada de forma embutida nestas linguagens. A linguagem SQL engloba numa única linguagem todos os recursos necessários para a manipulação de bases de dados relacionais.

## Referências Bibliográficas

- DATE, C. J. (1990). **Introdução a sistemas de banco de dados**. Tradução (da 4ª edição original) de Contexto Traduções. – Rio de Janeiro: Campus, 1990.
- FERRER, Inara. (2004). **Apostila de Banco de Dados**. CEFET/MT – Centro Federal de Educação Tecnológica de Mato Grosso, 2004.
- F.C.T. (1997). **Apostila de Banco de Dados**. Unimar-Universidade de Marília, 1997.
- KORTH, Henry F.; SILBERSCHATZ, Abraham. (1999). **Sistema de Banco de Dados**. 3 ed. Makron Books, 1999.
- MACHADO, Felipe Nery Rodrigues (2004). **Banco de Dados: Projeto e Implementação**. – São Paulo: Érica, 2004