



# Interativa

## Administração de Banco de Dados

**Autores:** Profa. Gisele Lopes Batista Pinto  
Prof. Luiz Fernando de Lima Santos

**Colaboradores:** Prof. Roberto Macias  
Profa. Elisângela Mônaco de Moraes  
Prof. Emanuel Augusto Severino de Matos

### **Gisele Lopes Batista Pinto**

É licenciada em Matemática pelo Instituto de Matemática e Estatística da Universidade de São Paulo, com especialização em Automação e Informática Industrial pela Escola Politécnica da USP.

Depois de um instigante e divertido período lecionando Matemática no Ensino Médio e para cursos de Magistério em uma escola pública no Estado de São Paulo, voltou-se para a Tecnologia da Informação. Durante 11 anos, foi Administradora de Bancos de Dados (DBA) no Departamento de Informática da Reitoria da USP e há cinco anos integra o Grupo de Projetos Institucionais do mesmo Departamento, atuando nas áreas de Mídias Sociais e Inteligência Corporativa.

### **Luiz Fernando de Lima Santos**

É formado em Ciências da Computação pela Uninove e pós-graduado em Tecnologia da Informação pela UNIP. Atua há mais de 10 anos na área de Banco de Dados e *Business Intelligence*, com passagem por empresas como Unilever e Johnson & Johnson, tendo ainda experiência com os maiores bancos do mercado como Microsoft SQL Server e Oracle. É professor de *Banco de Dados* na UNIP nas modalidades presencial e a distância (EaD).

#### **Dados Internacionais de Catalogação na Publicação (CIP)**

P659a      Pinto, Gisele Lopes Batista  
Administração de banco de dados / Gisele Lopes Batista Pinto;  
Luiz Fernando Lima dos Santos. São Paulo: Editora Sol, 2020  
108 p., il.  
Nota: este volume está publicado nos Cadernos de Estudos e  
Pesquisas da UNIP, Série Didática, ISSN 1517-9230.  
1. Informática. 2. Banco de dados. 3. Administração. I. Santos  
dos, Luiz Fernando Lima. II. Título.  
CDU 004.65

U507.87 – 20

Prof. Dr. João Carlos Di Genio  
**Reitor**

Prof. Fábio Romeu de Carvalho  
**Vice-Reitor de Planejamento, Administração e Finanças**

Profa. Melânia Dalla Torre  
**Vice-Reitora de Unidades Universitárias**

Prof. Dr. Yugo Okida  
**Vice-Reitor de Pós-Graduação e Pesquisa**

Profa. Dra. Marília Ancona-Lopez  
**Vice-Reitora de Graduação**

### **Unip Interativa – EaD**

Profa. Elisabete Brihy  
Prof. Marcello Vannini  
Prof. Dr. Luiz Felipe Scabar  
Prof. Ivan Daliberto Frugoli

### **Material Didático – EaD**

Comissão editorial:

Dra. Angélica L. Carlini (UNIP)  
Dr. Ivan Dias da Motta (CESUMAR)  
Dra. Kátia Mosorov Alonso (UFMT)

Apoio:

Profa. Cláudia Regina Baptista – EaD  
Profa. Betisa Malaman – Comissão de Qualificação e Avaliação de Cursos

Projeto gráfico:

Prof. Alexandre Ponzetto

Revisão:

Andréia Andrade  
Michel Apt  
Sueli Brianezi Carvalho



# Sumário

## Administração de Banco de Dados

APRESENTAÇÃO .....	9
INTRODUÇÃO .....	9

### Unidade I

1 INTRODUÇÃO A BANCO DE DADOS .....	11
1.1 Histórico .....	12
1.1.1 Modelo hierárquico .....	14
1.1.2 Modelo em rede .....	15
1.2 Definição de banco de dados .....	16
1.3 Tipos de bancos de dados .....	17
1.3.1 Bancos de dados relacionais .....	17
1.3.2 Bancos de dados não relacionais .....	18
1.3.3 Bancos de dados orientados a objeto .....	18
1.4 Sistemas de Gerenciamento de Banco de Dados (SGBD) .....	19
1.5 Funções básicas de um SGBD .....	21
1.5.1 Método de acesso .....	21
1.5.2 Restrições de Integridade (RI) .....	21
1.5.3 Segurança .....	22
1.5.4 Controle de concorrência .....	22
1.5.5 Independência dos dados .....	22
1.6 Arquitetura básica de um SGBD .....	23
1.6.1 Nível físico .....	23
1.6.2 Nível conceitual .....	23
1.6.3 Nível de visão .....	23
1.7 Como os dados são armazenados .....	23
1.7.1 Diferença entre dado e informação .....	24
1.7.2 Dicionário de Dados (DD) .....	25
2 AGENTES DE INTERAÇÃO COM O SGBD .....	25
2.1 DBA – Data Base Administrator .....	25
2.1.1 AD – Administrador de dados .....	26
2.1.2 Desenvolvedor de banco de dados .....	26
2.1.3 Programador .....	26
2.2 Certificações .....	26
2.2.1 Certificação Microsoft .....	26
2.2.2 Certificação Oracle .....	27

## Unidade II

3 DIAGRAMA ENTIDADE-RELACIONAMENTO (DER) .....	33
3.1 Entidades do DER .....	33
3.2 Relacionamentos do DER.....	33
3.2.1 Relacionamentos binários.....	33
3.2.2 Relacionamentos reflexivos.....	34
3.2.3 Relacionamentos ternários.....	34
3.3 Atributos do DER .....	34
3.3.1 Atributos opcionais.....	35
3.3.2 Atributos compostos.....	35
3.3.3 Atributos multivalorados.....	35
3.4 Entidades fracas.....	35
3.5 Especialização .....	35
3.5.1 Especialização com ou sem exclusão mútua .....	35
3.5.2 Especialização com ou sem obrigatoriedade .....	36
3.6 Agregação.....	36
4 MODELO ENTIDADE RELACIONAMENTO (MER).....	36
4.1 Domínio.....	37
4.2 Atributo do MER.....	37
4.3 Tupla .....	38
4.4 Relação .....	38
4.5 Chave.....	38
4.5.1 Chave primária (PK) – <i>primary key</i> .....	38
4.5.2 Chave estrangeira (FK) – <i>foreign key</i> .....	39
4.5.3 Chave artificial ou delegada (SK) – <i>surrogate key</i> .....	39
4.6 Restrições de integridades básicas.....	40
4.6.1 Regra de integridade de entidade .....	40
4.6.2 Regra de integridade referencial .....	40
4.7 Tipos de relacionamento .....	43
4.8 Normalização.....	44
4.8.1 Primeira Forma Normal (1FN).....	44
4.8.2 Segunda Forma Normal (2FN).....	46
4.8.3 Terceira Forma Normal (3FN) .....	47

## Unidade III

5 PROJETO DE BANCO DE DADOS.....	53
5.1 Linguagens de banco de dados .....	53
5.2 Tipos de linguagem SQL .....	54
5.2.1 <i>Data Definition Language</i> (DDL).....	54
5.2.2 <i>Data Manipulation Language</i> (DML).....	55
5.2.3 <i>Data Control Language</i> (DCL) .....	56
5.2.4 <i>Data Transaction Language</i> (DTL).....	58
5.2.5 <i>Data Query Language</i> (DQL).....	59

5.3 Módulos de funcionamento do SGBD.....	64
5.4 Etapas do projeto de banco de dados.....	65
5.4.1 Levantamento de requisitos.....	66
5.4.2 Projeto conceitual.....	66
5.4.3 Projeto lógico.....	66
5.4.4 Projeto físico.....	67
5.4.5 Criação do banco de dados.....	67
5.4.6 Exemplo de projeto de banco de dados.....	67
6 TIPOS DE DADOS ( <i>DATA TYPES</i> ).....	69
6.1 Elementos de dados.....	70
6.1.1 Estrutura de dados.....	70
6.1.2 Unidades de medidas.....	72

## Unidade IV

7 ADMINISTRAÇÃO DE BANCO DE DADOS.....	78
7.1 <i>Backup</i> .....	79
7.1.1 Recuperação e concorrência.....	79
7.1.2 Pontos de sincronização.....	80
7.2 Meios de recuperação ( <i>restore</i> ).....	81
7.2.1 Falhas do sistema.....	81
7.2.2 Falhas dos meios físicos.....	82
7.3 Concorrência.....	82
7.4 Bloqueio ( <i>Lock</i> ).....	83
7.5 Arquitetura básica do ambiente de banco de dados.....	85
7.5.1 <i>Cluster</i> de alto desempenho.....	85
7.5.2 <i>Cluster</i> de alta disponibilidade.....	85
7.5.3 <i>Cluster</i> para balanceamento de carga.....	85
7.6 Replicação.....	86
7.6.1 Replicação síncrona ( <i>eager</i> ).....	86
7.6.2 Replicação assíncrona ( <i>lazy</i> ).....	86
7.6.3 Replicação unidirecional ( <i>master-slave</i> ).....	87
7.6.4 Replicação multidirecional ( <i>multi-master</i> ).....	87
7.6.5 Exemplo de funcionamento de um <i>cluster</i> com replicação.....	87
8 MODELO DIMENSIONAL.....	88
8.1 Fatos.....	89
8.2 Dimensão.....	90
8.2.1 Chave artificial ( <i>surrogate key</i> ).....	91
8.3 Medidas.....	91
8.4 Granularidade.....	91
8.5 Agregados.....	92
8.6 Passos do projeto de DW.....	93
8.7 <i>Data mart</i> (DM).....	95
8.8 Ferramentas de acesso aos dados.....	95





## APRESENTAÇÃO

O conceito de banco de dados é importante porque auxilia no desenvolvimento das aplicações para os sistemas de informação, comunicação e processos de automação. Por meio deste curso, você será capaz de avaliar as tecnologias e arquiteturas disponíveis no mercado e fazer a escolha mais adequada. Além disso, você poderá entender os assuntos referentes à modelagem de dados e identificar os aspectos que impactam no desempenho e na segurança dos dados da empresa.

Todo o controle sobre o armazenamento e a manipulação de dados (no que diz respeito ao acesso, à integridade física e lógica, à segurança, à redundância, à concorrência entre as diversas aplicações, à autorização para as diversas operações etc.), são de responsabilidade de um *software* denominado de Sistema Gerenciador de Banco de Dados, isto é, o SGBD. Esses *softwares* são capazes de gerenciar diversos bancos de dados em um único sistema.

## INTRODUÇÃO

O surgimento da tecnologia de banco de dados ocorreu no momento em que os especialistas no desenvolvimento de sistemas computacionais perceberam que, para a informatização de grandes organizações, várias questões relacionadas com o gerenciamento de dados necessitavam ser resolvidas de forma mais eficiente, por exemplo, o acesso aos dados de diferentes setores de uma mesma empresa, por aplicações diferentes, de forma compartilhada e consistente, evitando a falta de padronização e problemas de segurança dos dados. A manutenção de uma base de dados consistente, sem o auxílio das técnicas e ferramentas que envolvem os bancos de dados, torna-se um trabalho insano. A maior dificuldade em construir sistemas corporativos em uma organização ocorre quando os dados são armazenados de forma não integrada com um projeto global da empresa. Se cada setor mantiver uma base de dados local e independente, fica extremamente difícil a integração das aplicações e a construção de um sistema de apoio gerencial para a organização.

A reflexão que devemos fazer é a seguinte: a empresa pode não ter a certeza de qual é a verdade referente à informação, mas ela deve ser única em todos os setores da organização. Se o relatório do setor de pessoal mostrar que a empresa tem 500 funcionários, significa que todos os relatórios dos demais setores devem ter o mesmo resultado. E isso só é possível se a fonte dessa informação for única em todos os setores, ou seja, se estiver cadastrada no mesmo banco de dados da empresa e não mais em arquivos locais de cada setor. O banco de dados passa a ser o repositório único dos dados, com controle eficiente das operações realizadas sobre os dados armazenados, sem falar na segurança.

Portanto, inicialmente, apresentaremos o conceito e um breve histórico sobre o surgimento dos bancos de dados. Mostraremos os diversos tipos de bancos de dados e falaremos um pouco sobre a carreira do profissional da área. Depois, discutiremos os conceitos baseados na modelagem física e na lógica dos dados e as técnicas de desenvolvimento de um projeto de banco de dados e suas aplicações. Também apresentaremos a linguagem de programação usada para tratar os dados que estão e serão armazenados nas estruturas de banco de dados. Por fim, falaremos sobre a importância e as técnicas para salvar ou restaurar com segurança os dados armazenados.



# Unidade I

## 1 INTRODUÇÃO A BANCO DE DADOS

O desenvolvimento de um sistema de informação envolve a análise e o projeto de dois componentes: os dados e os processos.

O projeto de dados é considerado a parte estática do sistema, uma vez que diz respeito a um universo persistente de características que dificilmente sofre modificações após a sua definição.

O projeto de processos, por sua vez, é chamado de parte dinâmica, uma vez que as tarefas a serem realizadas sobre os dados podem variar conforme ocorre a evolução do sistema.

Considera-se projeto de um banco de dados a análise, o projeto e a implementação dos dados persistentes de uma aplicação, levando em conta a determinação da sua semântica (abstração dos dados de uma realidade) e, posteriormente, o modelo de dados e o Sistema Gerenciador de Banco de Dados (SGBD) a serem adotados.

O projeto de um banco de dados é composto das seguintes etapas:

### Descrição de requisito

Etapla em que são coletadas informações sobre os dados de interesse da aplicação, seu uso, ou seja, as operações de manipulação sobre elas e suas relações.

Para a realização dessa etapa, são necessárias tarefas como entrevistas com os futuros usuários do sistema, análise de documentações disponíveis, arquivos, relatórios etc. O resultado normalmente é uma descrição dos requisitos da aplicação mais detalhada e clara possível.

Caso nenhuma metodologia de engenharia de *software* esteja sendo empregada, essa descrição de requisitos é uma descrição informal, na forma de um texto, que contém as informações necessárias para o entendimento da aplicação.

### Projeto conceitual

Pode ser considerada a fase de análise dos dados ou requisitos capturados na etapa anterior.

Nessa etapa, é realizado o que chamamos de modelagem conceitual: são analisados os fatos de interesse, isto é, as entidades ou conjunto de ocorrências de dados e seus relacionamentos, juntamente com seus atributos e propriedades ou características, e construída uma notação gráfica para facilitar o entendimento dos dados e suas relações, tanto para os analistas quanto para os futuros usuários.

Essa etapa resulta em um modelo conceitual, em que a semântica da realidade deve estar correta. A ferramenta mais usada nessa etapa é o Diagrama Entidade-Relacionamento (E-R).

### Projeto lógico

Considerada a fase de projeto dos dados. Nessa etapa, o desenvolvimento do banco de dados começa a se voltar para o ambiente de implementação, uma vez que é feita a conversão do modelo conceitual para um modelo de dados de um banco de dados (modelo lógico). Esse modelo de dados pode ser relacional, orientado a objetos ou dimensional, por exemplo.

Essa etapa se baseia no uso de regras de mapeamento de um tipo de diagrama de acordo com o modelo de dados definido. Pode ser um diagrama E-R ou um diagrama de classes, entre outros. O resultado é uma estrutura lógica, como um conjunto de tabelas relacionadas.

### Projeto físico

Esta última etapa realiza a adequação do modelo lógico gerado na etapa anterior ao formato de representação de dados do Sistema Gerenciador de Banco de Dados escolhido para a implementação.

Para a realização dessa etapa, deve-se conhecer os elementos e o funcionamento do SGBD, para a criação da estrutura lógica definida no modelo. O resultado é a especificação do esquema da aplicação e da implementação das restrições de integridade.

## 1.1 Histórico

Podemos afirmar que a história dos bancos de dados se inicia na década de 1950, no momento em que surge a necessidade de armazenar os dados gerados nos computadores. Nessa época, os recursos disponíveis eram bem inferiores em relação aos existentes hoje em dia. Antigamente, os dados eram armazenados em fitas magnéticas ou em cartões perfurados, e eram lidos e gravados de forma sequencial.

Na década de 1960, apareceram os primeiros discos rígidos, e com isso os dados não precisaram mais ser gravados de forma sequencial. Surgiu, então, um dos primeiros modelos, conhecido como Modelo Hierárquico, que organizava dados em uma estrutura em árvore. Os SGBD mais conhecidos, nessa época, foram o IMS e o System 2000.

No final da década de 1960 e durante a década de 1970, surgiu o Modelo de Redes, que organizava os dados em uma estrutura formada por várias listas, que definia uma intrincada rede de ligações. Nessa época, os SGBD mais conhecidos foram o IDMS e o Total.

O Modelo Relacional é o modelo de dados dominante no mercado ainda hoje e surgiu na década de 1970. Ele foi proposto por Edgar Frank Codd e se tornou um marco em como pensar em banco de dados. Codd desconectou a estrutura lógica do banco de dados do método de armazenamento físico, organizando os dados em um conjunto de relações. Essas relações são denominadas de tabelas.

Em cima da teoria de Codd, foram criados dois protótipos de sistemas relacionais, que depois foram sendo aperfeiçoados com o tempo. O primeiro foi o Ingress, desenvolvido pela UCB, que serviu como base para os sistemas Ingres Corp., Sybase, MS-SQL Server, Britton-Lee, Wang PACE (que utilizava QUEL como linguagem de consulta). O outro, o System-R, foi desenvolvido pela IBM San Jose e serviu de base para o IBM SQL/DS, IBM DB2, Oracle, todos os bancos de dados da HP e o Tandem's Non-Stop SQL, que utilizava SEQUEL como linguagem de consulta. O termo Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR ou RDBMS em inglês) foi definido durante esse período.



Figura 1 – Dr. Peter Chen

O doutor Peter Chen propõe o modelo Entidade-Relacionamento (E-R) para projetos de banco de dados, dando uma nova e importante percepção dos conceitos de modelos de dados. Assim como as linguagens de alto nível, a modelagem ER possibilita ao projetista concentrar-se apenas na utilização dos dados, sem se preocupar com a estrutura lógica de tabelas.

A partir da década de 1980, com o advento da computação pessoal, novos modelos começaram a ser definidos, visando atender às necessidades de aplicações ditas não convencionais, ou seja, aplicações cujas entidades apresentam uma estrutura que não se adequa bem com a organização relacional de dados.

A Linguagem Estruturada de Consultas (SQL) se torna a linguagem padrão mundial para os Sistemas Gerenciadores de Banco de Dados. Nessa década, surge também o modelo cliente-servidor, que muda a forma como os sistemas seriam concebidos.

A internet chega aos lares ainda um pouco rudimentar se compararmos com a de hoje, mas funcional o suficiente para que as pessoas se comuniquem. E, desde então, a pesquisa científica na área procura evoluir no sentido de definir e encontrar modelos que representem da melhor maneira possível os dados

de uma realidade, ou seja, que os organizem de forma mais próxima à maneira como estes são vistos e manipulados pelas pessoas no mundo real.

A grande maioria dos bancos de dados conhecidos hoje comercialmente foi criada nessa época. Com a evolução, começaram a surgir conceitos que são utilizados até os dias de hoje, como:

- OLTP – *Online Transaction Process* (Processos de Transação em Tempo Real)
- OLAP – *Online Analytical Process* (Processos Analíticos em Tempo Real)
- *Open Source* – *software* livre, que são os que mudariam o conceito de posse de um *software*. Os sistemas *open source* se baseiam em quatro leis, ou quatro liberdades, que são:
  - **Liberdade nº 0:** a liberdade de executar o programa, para qualquer propósito.
  - **Liberdade nº 1:** a liberdade de estudar como o programa funciona e adaptá-lo para suas necessidades. Acesso ao código-fonte é um pré-requisito para essa liberdade.
  - **Liberdade nº 2:** a liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.
  - **Liberdade nº 3:** a liberdade de aperfeiçoar o programa e liberar seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para essa liberdade.

### 1.1.1 Modelo hierárquico

O modelo hierárquico foi definido com base na observação de que muitas entidades do mundo real são organizadas hierarquicamente. Nesse modelo, os dados são organizados em um conjunto de tipos de registros (entidades), interconectados por meio de ligações (relacionamentos).

Uma ligação representa uma relação entre exatamente dois tipos de registros: pai e filho. Assim, tanto o esquema quanto os dados (ocorrências) são visualizados por meio de uma estrutura em árvore, em que um ou vários tipos de registros filhos podem estar vinculados a um tipo de registro pai.

Um esquema no modelo hierárquico é chamado de diagrama de estrutura em árvore, conforme podemos ver na Figura 2.

O sentido de acesso é sempre unidirecional, ou seja, sempre do pai para o filho (parte sempre da raiz e percorre os níveis inferiores).

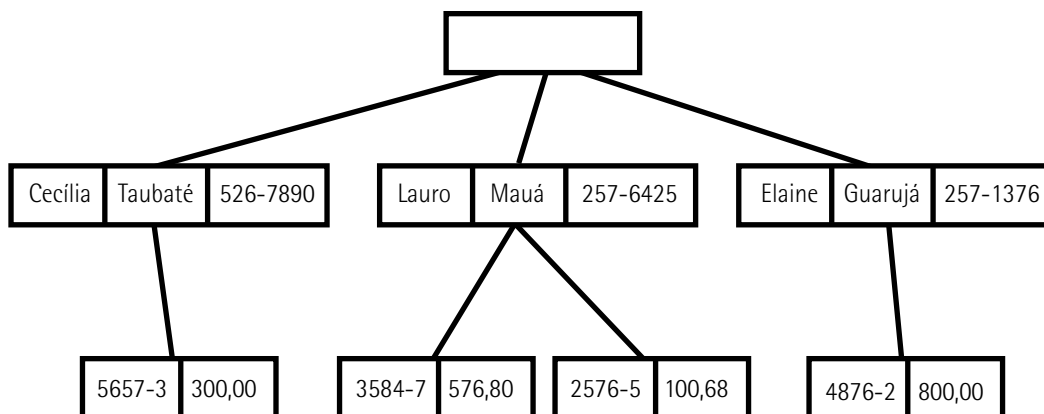


Figura 2 – Diagrama de estrutura em árvore



## Observação

O modelo hierárquico não suporta relacionamentos com cardinalidade N:N em decorrência da organização em árvore, que permite apenas cardinalidade 1:1 e 1:N, ou seja, um tipo de registro filho sempre está relacionado a um único tipo de registro pai. E um pai, por sua vez, pode se relacionar com vários filhos.

Os casos de cardinalidade N:N devem, então, ser modelados como uma relação 1:N, por meio da escolha de qual registro será tratado como pai. A indicação é escolher aquele registro que apresentar um maior valor médio de cardinalidade para ser o tipo de registro pai. A redundância de dados é inevitável, caso uma ocorrência de um tipo de registro filho se relacionar com mais de uma ocorrência de um tipo de registro pai. No modelo hierárquico, não há acesso direto a ocorrências de tipos de registros filho.

### 1.1.2 Modelo em rede

Esse modelo é muitas vezes denominado de Modelo DBTG-CODASYL (*Data Base Task Group* – subgrupo da *Conference on Data Systems and Language*), uma organização existente na década de 1970 responsável pela padronização de linguagens de programação de sistemas.

Similar ao modelo hierárquico, os dados no modelo de redes são organizados em tipos e ligações entre dois tipos de registros. Não existe restrição hierárquica, ou seja, quaisquer dois tipos de registros podem se relacionar. Assim, tanto o esquema quanto as ocorrências de dados são visualizados como um grafo direcionado. Um esquema no modelo de redes é chamado de diagrama de estrutura de dados, de acordo com a Figura 3.

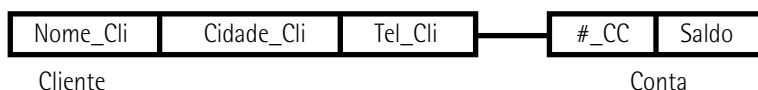


Figura 3 – Diagrama de estrutura de dado

Toda vez que existe um relacionamento com cardinalidade 1:1 ou 1:N, define-se um *set*, ou seja, o tipo de registro com cardinalidade fixa igual a 1 é chamado de *owner* e o outro, de *member*. Uma ocorrência de um *set* equivale a uma lista encadeada que parte do *owner* e encadeia todos os *members* relacionados a ele.

Quando existe um relacionamento com cardinalidade M:N, define-se o que se chama de *conector*: um tipo de registro adicional que estabelece uma ligação entre dois tipos de registro, ou seja, dois *sets* são definidos, transformando um relacionamento M:N em dois relacionamentos 1:N. Esse conector pode ou não ter atributos.



### Observação

No modelo em rede, cada relacionamento presente gera listas encadeadas e, com isso, as consultas que percorrem por muitos relacionamentos têm seu desempenho (*performance*) comprometido.

## 1.2 Definição de banco de dados

Um banco de dados pode ser definido como uma coleção de dados operacionais inter-relacionados. Esses dados são armazenados de forma independente dos programas que os utilizam, servindo assim a múltiplas aplicações de uma organização.

Além disso, o banco de dados deve ser o repositório único de armazenamento dos dados, pois, com isso, diminuimos a redundância e eliminamos redefinições de dados semelhantes de diversas fontes. O acesso ao banco de dados é realizado por meio de linguagens de alto nível para manipulação de dados. Em outras palavras, um banco de dados é um sistema de manutenção de registros. O seu objetivo principal é manter as informações para, quando solicitadas, serem disponibilizadas ao usuário.

Existem diversas plataformas de banco de dados, e a escolha depende do orçamento e de políticas de investimento em TI nas organizações. Houve um tempo em que apenas as plataformas proprietárias ofereciam produtos de qualidade e segurança, e somente empresas com elevado capital tinham condições de adquirir esses produtos. Mas, com o amadurecimento das plataformas *open source*, as pequenas empresas também podem desenvolver seu projeto de banco de dados de forma tão eficiente e segura quanto as empresas de grande porte.

Dentre as empresas fornecedoras de plataformas proprietárias de banco de dados, destacamos as seguintes:

- Oracle – [www.oracle.com](http://www.oracle.com)
- Microsoft – [www.microsoft.com](http://www.microsoft.com)
- Sybase (SAP) – [www.sybase.com](http://www.sybase.com)
- IBM – [www.ibm.com](http://www.ibm.com)



Apesar de oferecerem plataformas pagas, elas costumam disponibilizar uma versão gratuita do banco de dados. Mas essas versões gratuitas não são recomendadas para projetos em sistemas de produção, porque possuem limitações se comparadas com a versão paga. No *site* dos fornecedores, há as recomendações e restrições para o uso dessas versões.

Normalmente, elas são usadas na fase inicial do projeto, para ajudar na escolha da plataforma definitiva que será posteriormente adquirida. Funcionam como um *test drive*, em que podemos conhecer o produto sem custos.

As soluções chamadas *open source* mais conhecidas são:

- MySQL – [www.mysql.com](http://www.mysql.com)
- PostgreSQL – [www.postgresql.org](http://www.postgresql.org)
- Cassandra – [cassandra.apache.org](http://cassandra.apache.org)
- SQLite – [www.sqlite.org](http://www.sqlite.org)



## Saiba mais

Para saber mais e se manter atualizado(a) com os novos projetos e eventos sobre *software* livre, recomendamos os *sites* indicados a seguir.

<http://softwarelivre.org>

<http://www.softwarelivre.gov.br>

## 1.3 Tipos de bancos de dados

Existem diferentes tipos de bancos de dados, os mais conhecidos são:

- relacionais;
- não relacionais;
- orientados a objeto.

### 1.3.1 Bancos de dados relacionais

Bancos relacionais são os bancos derivados das primeiras pesquisas, realizadas em meados da década de 1970. São o tipo de banco mais utilizado no mundo. Os dados são separados em entidades, conforme cada assunto, e gravados como atributos dessas entidades. Permitem que essas entidades se relacionem entre si e proporcionam uma forma rápida e segura de armazenar e recuperar os dados.

Os mais conhecidos são:

- SQL Server Microsoft – [www.microsoft.com/sqlserver](http://www.microsoft.com/sqlserver)

- MySQL – [www.mysql.com](http://www.mysql.com)
- PostgreSQL – [www.postgresql.org](http://www.postgresql.org)
- Informix – [www-01.ibm.com/software/data/Informix](http://www-01.ibm.com/software/data/Informix)
- DB2 – [www-01.ibm.com/software/data/db2](http://www-01.ibm.com/software/data/db2)
- Sybase – [www.sybase.com](http://www.sybase.com)
- Oracle – [www.oracle.com](http://www.oracle.com)

### 1.3.2 Bancos de dados não relacionais

Um banco de dados é dito não relacional quando não suporta instruções e operações de junção na linguagem SQL. São muito utilizados em sistemas para a internet, por causa da velocidade e da escalabilidade maior em relação aos bancos relacionais. As primeiras pesquisas surgiram em 1998, mas o termo NoSQL (*not only sql*) como conhecemos hoje surgiu em 2009.

Os mais conhecidos são:

- Hadoop – [hadoop.apache.org](http://hadoop.apache.org)
- Cassandra – [cassandra.apache.org](http://cassandra.apache.org)
- Hypertable – [hypertable.org](http://hypertable.org)
- Amazon Simple DB – [aws.amazon.com/pt/simplifiedb](http://aws.amazon.com/pt/simplifiedb)
- CloudData – [www.cloudata.org](http://www.cloudata.org)



#### Saiba mais

Para mais informações sobre os bancos não relacionais, recomendamos o *site* indicado a seguir.

<http://nosql-database.org>

### 1.3.3 Bancos de dados orientados a objeto

Dizemos que um banco de dados é orientado a objetos quando cada informação é armazenada na forma de objetos, ou seja, utiliza a estrutura de dados chamada de orientação a objetos.

Os mais conhecidos são:

- db4objects – [www.db4o.com](http://www.db4o.com)

- GemStone – [www.gemstone.com](http://www.gemstone.com)
- Caché – [www.intersystems.com.br](http://www.intersystems.com.br)
- Zope – [zope.org](http://zope.org)

## 1.4 Sistemas de Gerenciamento de Banco de Dados (SGBD)

Um SGBD é um sistema cujo objetivo principal é gerenciar o acesso e a correta manutenção dos dados armazenados em um ou mais bancos de dados. O acesso aos dados é disponibilizado por meio de uma interface que permite a comunicação com a aplicação desenvolvida.

Em outras palavras, um SGBD é um *software* que serve de interface entre o usuário e o banco de dados em si.

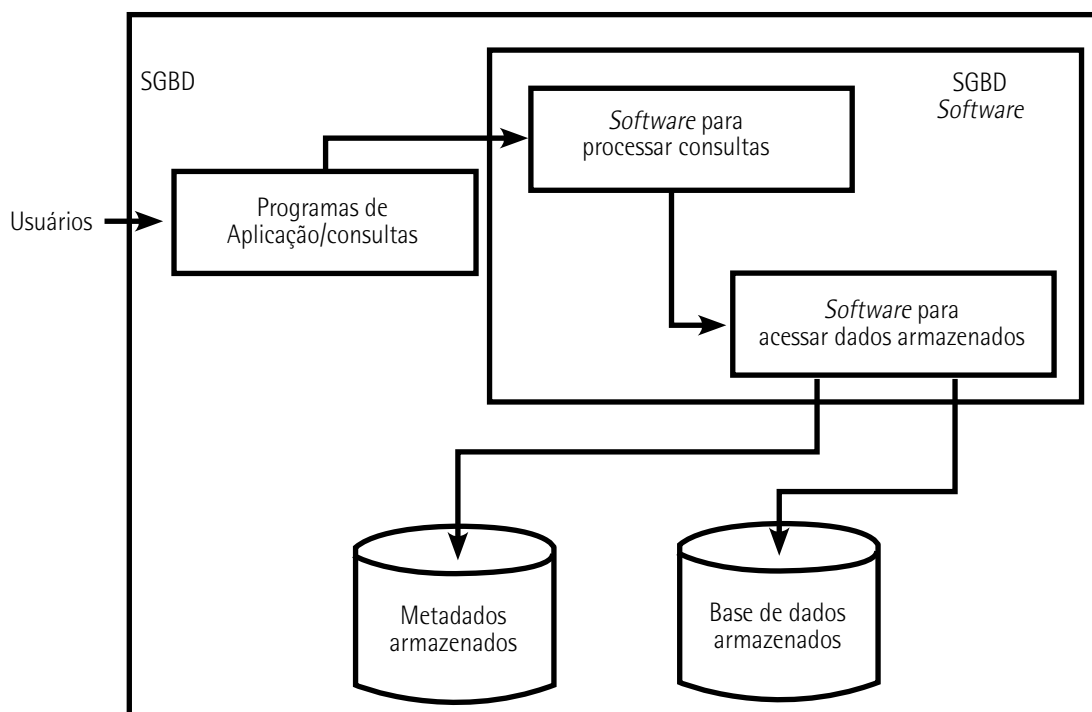


Figura 4 – Esquema de um SGBD

Os SGBD têm interfaces bem parecidas. Todos possuem o que chamamos de Object Explorer no lado esquerdo, onde é exibida a estrutura do banco de dados.

Eles utilizam um formato análogo ao do Windows Explorer, permitindo a navegação entre os objetos do banco de dados. Do lado direito, temos uma tela dividida em duas partes na vertical. Normalmente, a parte de cima é onde se digitam os comandos SQL, e a parte de baixo é onde se visualizam os resultados.

A seguir, alguns exemplos de interface utilizados pelo administrador de banco de dados para gerenciar os bancos de dados.

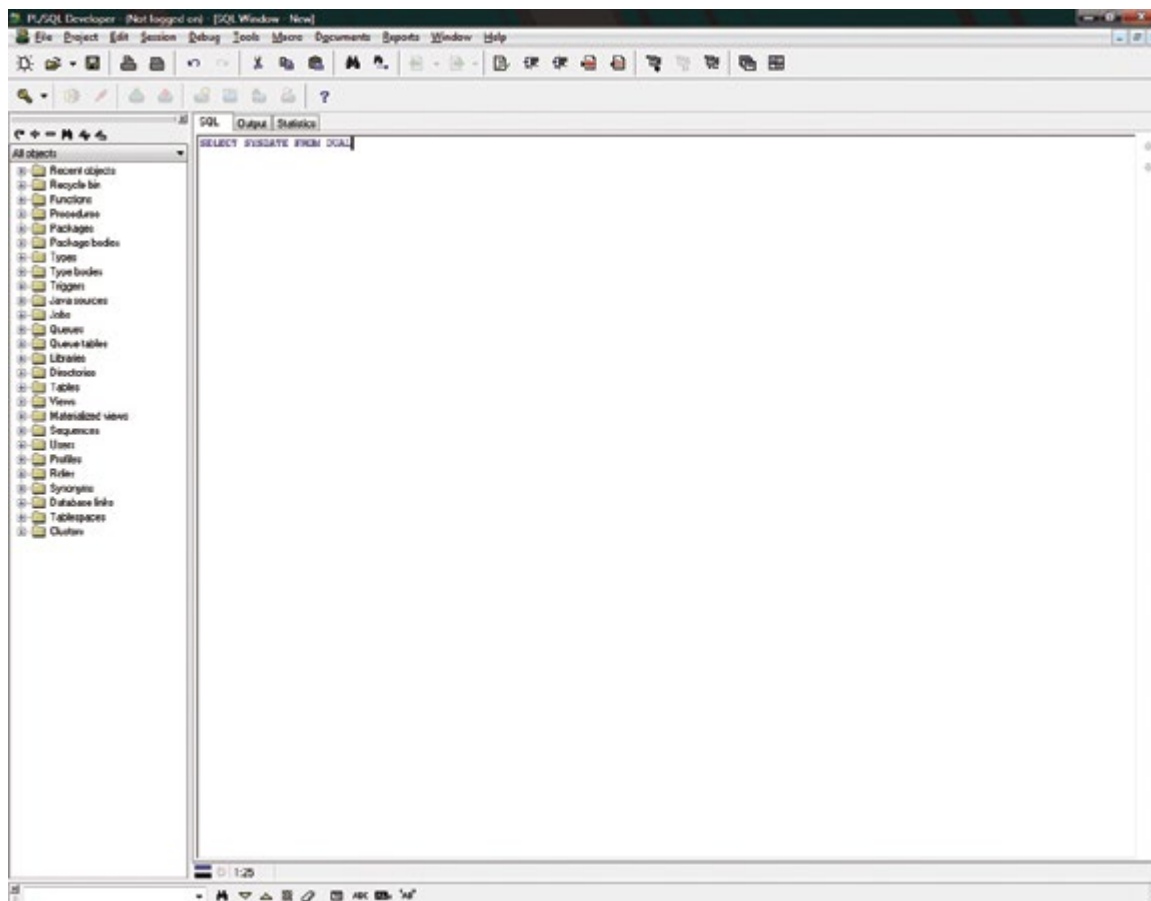


Figura 5 – SQL Server Management Studio

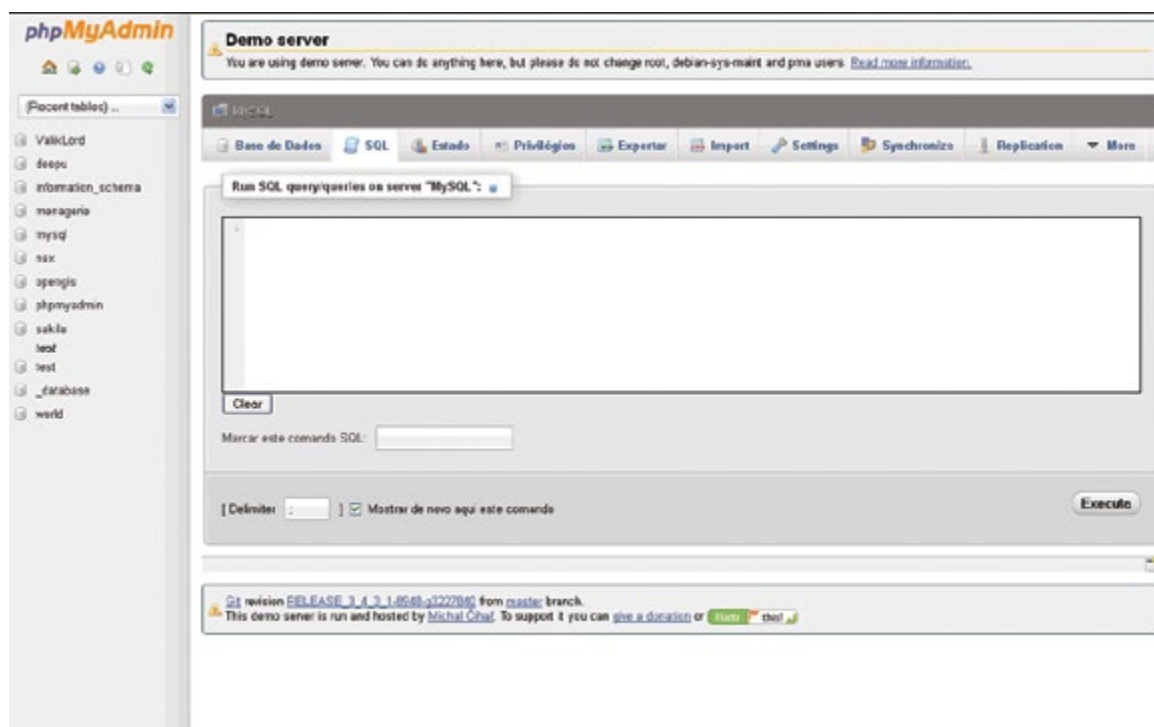


Figura 6 – PHPMyAdmin



## Observação

Você pode testar o Sistema Gerenciador do Banco de Dados MySQL, conhecido como phpMyAdmin, pelo *link*:

<http://demo.phpmyadmin.net/master-config>

## 1.5 Funções básicas de um SGBD

Com base na definição, as seguintes funções básicas são encontradas:

### 1.5.1 Método de acesso

Duas categorias de linguagem, conhecidas como DDL (*Data Definition Language*) e DML (*Data Manipulation Language*), devem ser suportadas.

A DDL permite a especificação do esquema da organização, ou seja, entidades com seus atributos e tipos de dados associados, os relacionamentos entre essas entidades e os índices de acesso associados aos atributos. Por esquema, entende-se uma organização lógica dos dados de uma realidade, adequados ao modelo de dados do SGBD.

A DML permite as operações de manipulação de dados, executadas pelas aplicações inclusão, alteração, exclusão e consulta. As consultas devem ser executadas de maneira eficaz, procurando minimizar o número de acessos ao meio de armazenamento secundário, assim como o volume de dados gerados nos resultados intermediários do processamento da consulta. Para tanto, a antecipação da aplicação de filtros e a busca apenas de dados relevantes para os relacionamentos, em uma certa etapa do processamento, são exemplos de estratégias que são consideradas.

### 1.5.2 Restrições de Integridade (RI)

Integridade está associada à ideia de dados corretos, consistentes no banco de dados. As restrições de integridade preocupam-se em manter dados sempre coerentes, verdadeiros com a realidade em questão.

Devem garantir os estados possíveis de serem assumidos pelos dados, por exemplo, alterações no valor do salário não podem diminuir o valor existente. A manutenção dos relacionamentos válidos entre os dados deve ser mantida, isto é, se existe uma regra que diz que uma turma de alunos, por exemplo, não pode estar vinculada a mais de uma disciplina, então deve existir uma restrição de integridade que garanta essa regra de negócio. Para tanto, o SGBD deve disponibilizar uma linguagem para especificação de restrições de integridade, chamada de DCL (*Data Constraint Language*), e por meio dessa linguagem é possível programar testes e ações.

### 1.5.3 Segurança

Esse controle evita a violação dos dados por agentes e/ou situações não previstas, ou seja, as falhas.

Políticas de autorização de acesso devem permitir que apenas agentes autorizados, sejam usuários ou aplicações, realizem certas operações sobre certos dados. Para tanto, faz-se necessário manter uma matriz de autorização, que especifica, para cada agente e cada dado, as operações autorizadas. Por dado, entende-se alguma porção do banco de dados, como um ou mais registros, um arquivo completo ou vários, alguns campos de um registro etc. O mecanismo de visões permite especificar a porção do banco de dados a que um agente tem direito de acesso.

A recuperação de falhas deve possibilitar o retorno do banco de dados a um estado consistente de seus dados após a ocorrência de uma falha. Para tanto, o SGBD deve manter arquivos históricos (*log*), que cadastram todas as alterações realizadas no banco de dados por transações. Entende-se por transação um conjunto de operações de manipulação de dados a que é submetido o banco de dados, sendo que todas essas operações devem ser efetivadas ou, na ocorrência de uma falha, nada deve ser efetivado, para preservar a consistência dos dados.

Os arquivos históricos (*log*) devem registrar, entre outras coisas, a identificação das transações, os arquivos manipulados, os registros atualizados, a operação feita e os valores atual e antigo. No caso de falhas de transação ou de sistema, o arquivo histórico (*log*) deve ser consultado e as ações realizadas por transações inacabadas devem ser desfeitas. Caso todas as modificações da transação estejam registradas, é possível refazê-la. Já para falhas de meio de armazenamento, que tornam inoperantes as unidades de disco onde se encontra o banco de dados, deve-se restaurar o banco de dados a partir do último *backup* realizado e consultar o *log* para refazer ou desfazer as transações cadastradas após esse *backup*.

### 1.5.4 Controle de concorrência

Esse controle evita conflitos de acesso simultâneo a um dado por mais de uma transação.

Se esse controle não existisse, os dados consultados por uma transação poderiam se tornar inválidos caso fossem atualizados por outra transação. Esse controle geralmente é feito pelo uso de estratégias de bloqueio (*lock*), que garantem que apenas uma transação manipule um dado, durante o espaço de tempo que necessitar, sem que ocorra interferência de outras transações.

### 1.5.5 Independência dos dados

Essa funcionalidade do SGBD é uma decorrência direta das vantagens trazidas pelo uso de um banco de dados. Independência de dados significa transparência de gerenciamento e armazenamento, assim como do esquema global da organização, para as aplicações.

O primeiro caso é chamado de independência física, ou seja, a aplicação não se preocupa com detalhes de localização física dos dados ou controles de integridade e segurança. A independência

lógica garante, pelo mecanismo de visões, que uma aplicação tenha condições de especificar a porção do banco de dados a que deseja ter acesso, não precisando ter conhecimento do esquema global.

### 1.6 Arquitetura básica de um SGBD

Um SGBD geralmente interage com diversas aplicações de uma organização, assim como com os meios de armazenamento de dados. No primeiro caso, a aplicação se vale de comandos DML embutidos no seu código. No segundo caso, geralmente existe uma interface com o sistema operacional do equipamento para leitura e gravação de dados.

Assim, um SGBD lida com diversos níveis de visão de um mesmo dado, de maneira a abstrair detalhes da organização dos dados.

#### 1.6.1 Nível físico

Nível mesmo abstrato, em que se sabem detalhes físicos da organização de um dado, ou seja, o esquema físico. É uma estruturação de baixo nível, em que é descrito como os dados estão armazenados em termos de *bytes*. O módulo de gerenciamento de arquivos lida com esse nível de visão dos dados, uma vez que localiza, lê e grava registros.

#### 1.6.2 Nível conceitual

Nível intermediário, que suporta uma descrição de mais alto nível em relação ao nível físico, em que a preocupação está em descrever quais dados são significativos para uma organização, isto é, têm semântica, são relevantes para o dia a dia da organização. Nesse caso, trabalha-se com os dados em nível de entidades e relacionamentos, tais quais estão definidos no esquema. É similar, por analogia, à descrição de um registro em uma linguagem de programação, ou seja, trabalha-se com nomes de arquivos e campos, e não com sequências de *bytes*. Os usuários que lidam nesse nível devem ter conhecimento completo de todo o esquema da organização, como o administrador do banco de dados ou o usuário especializado.

#### 1.6.3 Nível de visão

Nível mais alto de abstração, sendo particular de cada aplicação que manipula dados do banco de dados. Nesse caso, cada aplicação pode determinar o universo de dados a que deseja ter acesso. Esse universo de dados pode ser uma porção do esquema global da organização. Assim sendo, cada aplicação abstrai detalhes irrelevantes, desnecessários para seu processamento. Os usuários desse nível são os programas de aplicação.

### 1.7 Como os dados são armazenados

Os dados são armazenados em áreas chamadas páginas. O tamanho dessas páginas pode variar de banco para banco. Nelas, são armazenados os dados e os metadados (são os dados dos dados).

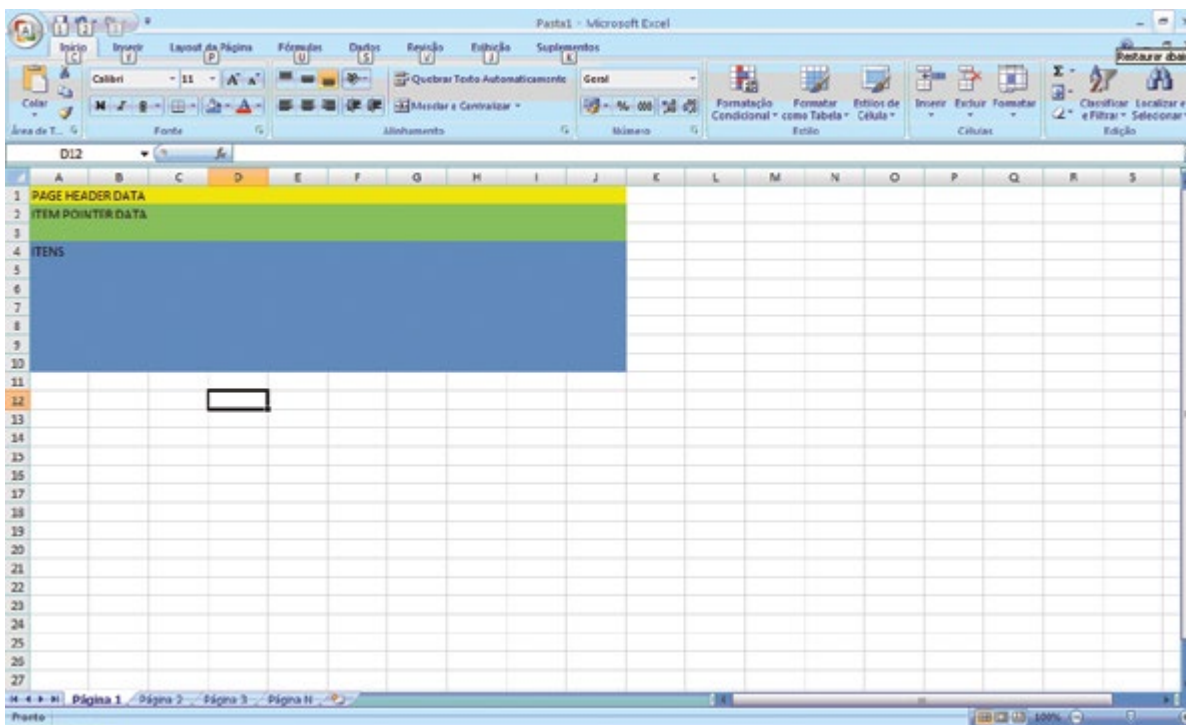


Figura 7 – Exemplo de uma página de dados

Os elementos de uma página são:

- PAGE HEADER DATA:** armazena as informações, como a última atualização dos dados, a posição do próximo dado a ser gravado etc.;
- ITEM POINTER DATA:** grava as informações sobre os índices dos dados;
- ITEMS:** são os dados e metadados, propriamente ditos.

## 1.7.1 Diferença entre dado e informação

Para simplificar, podemos dizer que tudo o que é armazenado pode ser considerado como dado. Por exemplo: um nome de uma pessoa, um nome de um produto, uma data, um local, um endereço, um número de telefone etc.

A informação surge quando agregamos dois ou mais dados e, a partir deles, inferimos uma conclusão. Por exemplo, vamos considerar os seguintes dados registrados:

- Estado = São Paulo
- Data = 28 de agosto de 1978
- Nome = Luiz Fernando



Nesse caso, temos os seguintes dados (que isoladamente não têm nenhum significado): temos um Estado, temos uma data e um nome de uma pessoa. Mas, dependendo do contexto em que esses dados aparecem, podemos chegar a algumas conclusões, como: a pessoa é paulista, se essa data se refere à data de nascimento, podemos afirmar que, em 2020, ela fez 42 anos e é do signo de virgem.

### 1.7.2 Dicionário de Dados (DD)

É o catálogo responsável pela manutenção dos metadados que dizem respeito à estrutura do esquema, à integridade, às configurações do SGBD para efeitos de controle, segurança e desempenho (*performance*), às estimativas de acesso e estimativas sobre os dados. É constantemente consultado durante a realização de várias das suas tarefas, como processamento de consultas, pré-compilação de comandos DML, verificação de integridade em operações de atualização etc.

O administrador do banco de dados tem acesso às suas informações por meio de ferramentas especiais.



#### Lembrete

Estrutura de esquema se refere às entidades (com atributos e tipos de dados associados) e relacionamentos.

## 2 AGENTES DE INTERAÇÃO COM O SGBD

Um SGBD deve se comunicar com vários agentes (usuários ou programas), com o objetivo de atender às necessidades de dados de diversas aplicações, permitir o desenvolvimento de aplicações que utilizam um banco de dados, assim como possibilitar que aspectos de *performance* possam ser otimizados, conforme a demanda de acesso a dados pelas aplicações.

Os sistemas gerenciadores de banco de dados são projetados com muitos recursos visuais e aplicações internas que servem como verdadeiros assistentes. Ainda assim, eles não se autoadministram, ou seja, não eliminam a necessidade dos profissionais especializados, apenas agilizam tarefas trabalhosas, como escrever *scripts*.

### 2.1 DBA – Data Base Administrator

Especialista no SGBD adotado pela organização, o DBA (Administrador de Banco de Dados) é o profissional que controla diversos aspectos funcionais, como definição e modificação do esquema, das autorizações de acesso e das regras de integridade. Além disso, tem o controle dos procedimentos de segurança, execução de *backups* e recuperação de falhas de meio de armazenamento, assim como monitoração de *performance* de acesso de dados e tomada de decisões para melhorias no SGBD.

**Lembrete:** o *backup* parece simples, mas, na verdade, precisa ser bem planejado. Surge a questão: fazer o *backup* completo, que armazena todos os registros do banco de dados, várias vezes ao dia,

prejudicando a *performance* do sistema, ou fazer *backups* incrementais, que só armazenam a lista dos registros que foram alterados e que complicam uma eventual restauração?

### 2.1.1 AD – Administrador de dados

É o profissional responsável pelos dados armazenados, especializado em Projeto de Banco de Dados. É de sua responsabilidade o desenvolvimento de relatórios gerenciais e a distribuição da informação dentro da empresa em todos os níveis. Interage com os usuários da aplicação a ser desenvolvida, com o objetivo de definir os requisitos de dados e especificar o esquema do banco de dados. Deve ser um especialista em desenvolvimento de sistemas.

### 2.1.2 Desenvolvedor de banco de dados

É o profissional responsável pelo desenvolvimento de programas de aplicação dentro do banco de dados. Esses programas interagem com o SGBD por meio de comandos de manipulação de dados (DML) embutidos no seu código. São desenvolvidos usando a linguagem padrão do banco de dados, como Oracle PL-SQL, Microsoft T-SQL, Transact SQL da Sybase, entre outros, e podem ser rotinas que são executadas sob outras aplicações ou mesmo em conjunto com elas.

### 2.1.3 Programador

É o profissional que desenvolve aplicações utilizando ferramentas compatíveis com o SGBD. Essas ferramentas podem ser compiladores de linguagens de programação tradicionais que permitem a integração de comandos da linguagem da DML no seu código, programas que usam alguma linguagem de programação (C#, PHP, VB.NET, DELPHI, JAVA etc.), programação de sistemas e aplicativos de manipulação de dados, geradores de interfaces gráficas, geradores de relatórios etc.

## 2.2 Certificações

Dentro do mercado de tecnologia como um todo, além do tempo de atuação na área e a experiência, outra forma de se comprovar o domínio de um profissional sobre uma determinada tecnologia é por meio de certificações.

Essas certificações são criadas e mantidas pelos fornecedores das soluções de tecnologia por meio de provas aplicadas e gerenciadas por parceiros globais. A seguir, veremos as certificações mais importantes dentro da área de banco de dados.

### 2.2.1 Certificação Microsoft

- MCTS – *Microsoft Certified Technology Specialist* (primeiro nível)
- MCITP – *Microsoft Certified IT Professional* (segundo nível)

MCM – *Microsoft Certified Master* (top)

- Divide-se em três caminhos:
  - Database Administrator
  - Database Developer
  - BI Developer



### Saiba mais

Para obter informações sobre as carreiras e certificações da Microsoft, recomendamos consultar o *site* indicado a seguir.

[www.microsoft.com/learning](http://www.microsoft.com/learning)

### 2.2.2 Certificação Oracle

- *Oracle Certified Associate* (primeiro nível)
- *Oracle Certified Professional* (segundo nível)
- *Oracle Certified Master* (terceiro nível)
  - Duas provas por nível
  - Necessidade de cursos presenciais



### Saiba mais

Para informações sobre as certificações nos produtos da Oracle, recomendamos consultar o *site* indicado a seguir.

[education.oracle.com](http://education.oracle.com)



### Observação

Apesar de as provas serem mantidas pelas empresas fornecedoras da tecnologia, existe uma agência externa que as aplica. Trata-se da Prometric. No *site* indicado a seguir você encontra, por exemplo, informações sobre certificações e preço de cada prova.

[www.prometric.com](http://www.prometric.com)



### Resumo

Nesta unidade, você aprendeu que um banco de dados é um conjunto de registros dispostos em uma estrutura regular, possibilitando sua reorganização e a produção de informação.

O surgimento da tecnologia de banco de dados ocorreu no momento em que os especialistas no desenvolvimento de sistemas computacionais perceberam que, para a informatização de grandes organizações, várias questões relacionadas com o gerenciamento de dados necessitavam ser resolvidas de forma mais eficiente.

Problemas como redundância, manutenção dos dados, atualizações, correções, falta de padronização, segurança, entre outros, com o uso dos bancos de dados são resolvidos. Os dados passam a ser armazenados em um único local, em vez de serem arquivados em diversos formatos e aplicações locais, sem nenhuma gerência e segurança.

Todo o cuidado com os dados, no que diz respeito ao acesso, à integridade, à segurança, à concorrência, à autorização de uso etc., é incumbência de um *software* de gerenciamento de banco de dados, totalmente projetado para isso. Além disso, você pode ter uma ideia sobre a carreira profissional nessa área e como obter certificação nas diferentes plataformas.

Na área de Tecnologia da Informação, existe um profissional que só é lembrado nos momentos críticos: o DBA. Sua função maior é assegurar que o banco de dados esteja funcionando e acessível todo o tempo que o sistema necessitar, com rapidez e confiabilidade. Ou seja, quando o sistema está "redondo", ninguém se lembra de que existe um DBA que trabalha arduamente para isso. Porém, se o banco de dados ficar *off-line* por alguns minutos, você receberá uma dezena de telefonemas de reclamações. Entre as funções administrativas que devem ser executadas em um banco de dados, estão *backup* periódico, verificação de consistência e otimização (*tuning*).

O profissional que atua como DBA precisa ter grande conhecimento da modelagem dos dados, para saber o que implicam as alterações. Essa é uma das razões pelas quais o perfil do DBA fica cada vez mais amplo. Sua participação na fase de modelagem de dados é muito importante, pois seus conhecimentos de *performance* e otimização em muito contribuem para a definição de dados. Outro conhecimento que pode agregar valor a um DBA é o de processos administrativos e de negócios, para que ele entenda o porquê do banco de dados que administra e qual a importância de cada tabela, o que pode fazer diferença em uma otimização.



## Exercícios

**Questão 1.** Ao se pesquisar sobre Bancos de Dados encontram-se diversas definições ou diversos conceitos que de alguma forma afirmam que os BDs são coleções de dados que se relacionam de forma a criar um sentido dentro de um determinado contexto (empresarial, científico e social). São considerados de vital importância para empresas, já que por meio dos Sistemas de Informação todos os dados fundamentais são armazenados, relacionados e acessados em todos os níveis da corporação. Com os Banco de Dados e os SIs os processos são executados, controlados e as tomadas de decisão se dão a partir dos resultados das informações recuperadas e organizadas nos BDs da empresa. Desde a década de 1960, eles se tornaram a principal peça dos sistemas de informação. Os Bancos de Dados são gerenciados ou operados pelos Sistemas Gerenciadores de Bancos de Dados (SGBD), que surgiram na década de 1970 e vêm evoluindo tanto na sua estrutura quanto no seu poderio de armazenamento. Os Bancos de Dados são utilizados em todos os tipos de aplicação, mas na atualidade suas principais aplicações se dão no controle de operações empresariais e no gerenciamento de informações científicas, tais como os Bancos de Dados Geográficos e os Banco de Dados Sociais.

Considerando os conceitos sobre Banco de Dados, examine as afirmações a seguir e indique a alternativa **incorreta**:

- A) O projeto de dados é considerado a parte estática de um sistema de informação e as funcionalidades são consideradas como a parte dinâmica do SI.
- B) Dentro de um ciclo de desenvolvimento de um SI, logo no início do ciclo, são determinados os requisitos do sistema ou aplicação; nessa etapa são coletadas informações sobre os dados de interesse da aplicação, fundamentais para um bom projeto de Banco de Dados.
- C) O Modelo Relacional proposto por Edgar Frank Codd se tornou um marco em como pensar em banco de dados, já que a estrutura lógica do banco de dados organiza os dados muito próximos do armazenamento físico dos discos magnéticos, trazendo grandes vantagens sobre os outros modelos com relação ao espaço de armazenamento e ao tempo de acesso.
- D) Pode-se afirmar que na atualidade não seria possível construir aplicações multiusuárias sem o uso de Banco de Dados e dos Sistemas Gerenciadores de Bancos de Dados (SGBDs).
- E) Como os Bancos de Dados, na atualidade, são utilizados para armazenar todos os tipos de informações, que vão de pessoas até empresariais, a segurança do BD se tornou crítica e exige do gerenciador do banco de dados mecanismos que garantam a integridade, a disponibilidade e a confidencialidade das informações armazenadas.

Resposta correta: alternativa C.

### Análise das alternativas

A modelagem de banco de dados relacional de Codd tornou-se um marco na tecnologia de BD devido à desconexão da estrutura lógica do banco de dados com os métodos de armazenamento físico vigente na década de 1970. Toda a modelagem do Banco de Dados está associada à teoria matemática das relações ou conjuntos, o que traz a organização dos dados de uma aplicação para mais próximo do pensamento humano. Os dados passam a ser vistos como conjuntos matemáticos que podem ser manipulados pela teoria das relações.

A) Alternativa correta.

Justificativa: os dados em um sistema de informação representam as propriedades de uma determinada entidade que devem ser persistidas durante o tempo e garantidas quanto à suas integridades. Devido a essa propriedade eles são considerados como a parte estática do sistema de informação. Os dados de um cliente de uma empresa em um sistema de compras tem que ser armazenados e garantidos ao longo de todas as suas operações realizadas nos SIs da empresa. Os dados ou características básicas desse cliente (nome, CPF, telefone, endereço, RG etc.) raramente são alterados, mas os dados de suas operações comerciais são constantemente modificados.

B) Alternativa correta.

Justificativa: os requisitos de um sistema determinam o escopo a ser desenvolvido, isto é, o conjunto de necessidades dos clientes ou usuários que utilizarão o sistema em suas tarefas do dia a dia. Por meio de reuniões, entrevistas, documentos existentes dos processos organizacionais são determinadas as funcionalidades, a usabilidade e os dados necessários para a execução das tarefas operacionais envolvidas no sistema.

C) Alternativa incorreta.

Justificativa: o Modelo Relacional foi proposto por Edgar Frank Codd e se tornou um marco em como pensar em banco de dados porque Codd desconectou a estrutura lógica do banco de dados do método de armazenamento físico, organizando os dados em um conjunto de relações denominadas de tabelas.

D) Alternativa correta.

Justificativa: aplicações multiusuárias são complexas, já que diversas operações podem ser realizadas e manipulam coleções de dados operacionais inter-relacionados. Esses dados devem ser corretamente armazenados, relacionados e garantidos quanto às suas integridades, exigindo um grande esforço de programação dos dispositivos de armazenamento. Dessa forma, o uso de estruturas de Banco de Dados e dos monitores de BDs (SGBDs) é fundamental no sucesso dos sistemas de informação.

E) Alternativa correta.

Justificativa: existem diversos mecanismos dos gerenciadores de banco de dados para suporte às aplicações na guarda e manuseio dos dados existentes nos Banco de Dados. Como diversas aplicações

acessam e manipulam os mesmos dados durante suas tarefas, se torna fundamental que os SGBSs garantam a integridade, a disponibilidade e a confidencialidade dos dados contidos em seus arquivos.

**Questão 2.** Como todo sistema de computador, um sistema de gerenciamento de banco de dados (SGBD) é um conjunto de componentes de *software* que permitem armazenar, modificar e extrair informações de um Banco de Dados. Atualmente existem muitos tipos diferentes de SGBDs, desde os considerados pequenos sistemas que funcionam em computadores pessoais, celulares, *tablets* até sistemas extremamente grandes que residem em grandes computadores das grandes organizações, em servidores (plataforma baixa) ou nas máquinas denominadas de *mainframes* (ou plataforma alta). Para a execução de todas as atividades envolvidas em um SGBD existem basicamente três componentes: a) a linguagem de definição de dados que especifica os conteúdos, a estrutura da base de dados e define os elementos de dados; b) a linguagem de manipulação de dados para obter e alterar os dados no banco de dados; c) um dicionário de dados onde estão as definições de elementos de dados e respectivas características que descreve os dados, quem os acede etc. Apesar de todos esses recursos, os SGBDs não se autoadministram e exigem profissionais especializados para sua administração e controle.

Considerando os conceitos sobre os SGBDs, examine as afirmações a seguir e indique a alternativa **incorreta**:

- A) Os SGBDs modernos são projetados e construídos com muitos recursos internos que praticamente eliminam a necessidade de uma organização possuir uma estrutura de pessoas especializadas para sua organização e administração.
- B) O *backup* dos bancos de dados de uma empresa precisa ser bem planejado já que diversos aspectos operacionais devem ser considerados nessa atividade: Quando efetuar um *backup* total ou como se fazer *backups* parciais sem perder a integridade dos dados? Que aplicações podem, ou não, ser executadas nos momentos de *backups* dos bancos de dados? Outras considerações também podem ser necessárias.
- C) O mercado preocupa-se muito com os profissionais que manipulam os Sistemas de Banco de Dados nas empresas e as certificações profissionais são uma forma de comprovação do conhecimento de pessoas em uma determinada tecnologia. Isso acontece com os profissionais denominados de DBAs.
- D) Diversos autores afirmam que a utilização de um SGBD traz enormes vantagens que podem ser resumidas em: controle da redundância, restrição a acessos não autorizados, permitir um armazenamento persistente para os objetos dos programas e estruturas de dados, permitir a inferência em ações seguindo regras, permitir múltiplas interfaces para os utilizadores, permitir a representação de relacionamentos complexos entre os dados, reforçar as restrições de integridade e a recuperação de dados em caso de pane.
- E) Um modelo de banco de dados, para ser considerado relacional, deve seguir as denominadas "Doze regras" de Codd, que foram criadas para impedir que os fornecedores de banco de dados não utilizassem somente o nome relacional, mas que seus produtos implementassem a teoria das relações matemáticas do modelo.

Resposta correta: alternativa A.

### Análise das alternativas

A) Afirmativa incorreta.

Justificativa: um SGBD da atualidade oferece uma gama de recursos aos seus utilizadores: armazenamento persistente, flexibilidade no uso dos dados, estruturas de dados que permitem o acesso eficiente a grandes quantidades, interface de programação por meio de uma linguagem de consulta extensa, gestão de transações com o acesso simultâneo a dados, o isolamento das transações, a atomicidade, admite a resiliência (a capacidade de se recuperar de muitos tipos de falhas ou erros). Todavia, apesar de todos esses recursos automáticos, os bancos de dados e os SGBDs não se autoadministram, apenas facilitam as tarefas que seriam trabalhosas aos programadores e administradores dos BDs da organização.

B) Afirmativa correta.

Justificativa: o planejamento de backups de banco de dados é uma tarefa complexa devido às consequências operacionais que um mau planejamento pode provocar. Essa atividade deve ser realizada por um especialista nos SGBDs da organização. E esse profissional é o Administrador de Banco de Dados (DBA), que controla os aspectos funcionais envolvidos com o uso dos BDs da empresa.

C) Afirmativa correta.

Justificativa: o mercado preocupa-se muito com os profissionais que manipulam os sistemas de banco de dados nas empresas, e as certificações profissionais representam uma forma de comprovação do conhecimento de pessoas em uma determinada tecnologia. Isso acontece com os profissionais denominados DBAs.

D) Afirmativa correta.

Justificativa: sem um método construtor, o objeto não poderia ser inicializado e não se teria a atribuição de valores às suas propriedades.

E) Afirmativa correta.

Justificativa: Edgard Codd criou as "Doze Regras", pois se preocupava com as adaptações que os fornecedores do mercado faziam em seus gerenciadores de bancos de dados tradicionais, os quais eram chamados de Relacionais. Como exemplo, vemos que as regras 0 e 1 já delimitam os conceitos que os gerenciadores relacionais devem seguir: Regra 0 – O sistema precisa ser qualificado como relacional, como um banco de dados, e como um sistema de gerenciamento; Regra 1 – todas as informações no banco de dados devem ser representadas de uma única forma, nomeadas por valores em posições de colunas (dentro de registros) no formato de tabelas.