



Nesta aula você estudará os **bancos de dados** como uma parte importante dos sistemas de informação e os princípios para que se construa um bom banco de dados.

Será visto também os sistemas de gerenciamento de banco de dados e sua importância para a manutenção dos dados, bem como a forma de consulta aos dados.

Por fim, será estudado o conceito de ***data warehouse*** e ***data mining***, como operações importantes que utilizam bancos de dados para a descoberta de padrões e outras atividades após a utilização do sistema de informação.

Banco de Dados

Como visto no modelo dinâmico de um sistema de informação, é essencial que um sistema acesse um banco de dados.

Um banco de dados é um grupo lógico de arquivos relacionados entre si, armazenando dados e associações entre eles, para evitar uma variedade de problemas associados a um ambiente tradicional de arquivos (TURBAN *et al*, 2005).

Mais uma definição nesta mesma linha é um **conjunto integrado de elementos de dados relacionados logicamente** (O'BRIEN, 2002).

Outro conceito reflete uma similaridade com a pirâmide do conhecimento. **O aspecto hierárquico envolvido na organização de um banco de dados, indo desde o bit, que se agrupa em bytes, e que, por sua vez, constituem os campos. Os campos constituem um registro, e vários registros compõem finalmente um arquivo; o conjunto destes arquivos forma o banco de dados** (LAUDON e LAUDON, 2010).

Pode-se afirmar que **o Banco de Dados (BD) procura ter em sua representação uma "imagem" de uma situação do mundo real, constituído de objetos, relações entre estes objetos e eventos.**

A partir desta imagem, o BD então tem condições de fornecer informações, evidenciando situações que podem ter importância para um processo de tomada de decisão. **Os dados podem ter representações diversas para uma mesma situação** (MEDEIROS, 2007).

É necessário que um BD tenha uma representação eficiente, possibilitando acesso a informações corretas em tempo hábil. Certos princípios devem ser levados em conta para se obter um BD eficiente:

1. **Redundância:** O armazenamento dos dados de determinada empresa, ao longo de suas atividades, pode tender à redundância, ou seja, setores que dependem de informações comuns podem fazer a guarda dos mesmos dados simultaneamente.

A falta de cuidado na análise do sistema de informações pode acarretar em redundância, incorrendo em custos de armazenamento.

2. **Inconsistência:** Os dados armazenados referentes a determinada situação que pode sofrer alterações ao longo do tempo necessita de atualização.

Dados desatualizados podem gerar inconsistência de representação. A redundância pode também acarretar inconsistência, pois dados armazenados em locais diferentes podem sofrer alterações diferenciadas ao longo do tempo.

A inconsistência pode gerar tomada de decisões defasadas ou errôneas.

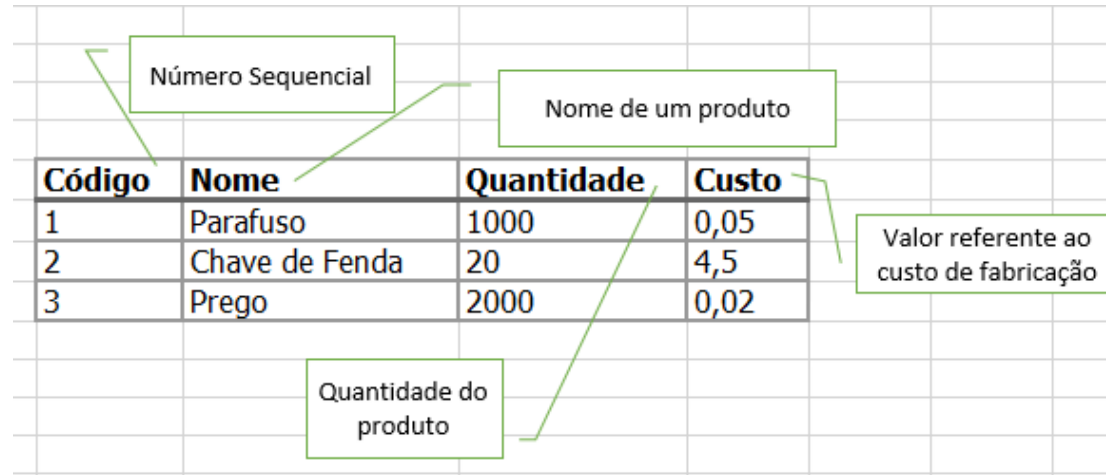
3. **Integração:** Os dados existentes em um BD geralmente são compartilhados por várias pessoas ou setores em uma empresa.

A necessidade de se haver integração é o de estabelecer procedimentos para o acesso em vários níveis e a atualização dos dados, de forma a manter a “imagem” do mundo real única e evitando ruídos na comunicação entre setores.

Armazenamento e Consulta a um Banco de Dados

A melhor maneira de se visualizar um registro de uma tabela em um banco de dados é através de uma lista ou relação.

Numa lista, os dados referentes a certo contexto serão colocados em colunas, um abaixo do outro, e as colunas dos dados de diferentes naturezas são colocadas uma ao lado da outra. Como exemplo, a lista a seguir consta de quatro colunas.



The diagram shows a table with four columns: 'Código', 'Nome', 'Quantidade', and 'Custo'. It contains three rows of data. Annotations with green boxes and arrows point to specific parts of the table: 'Número Sequencial' points to the 'Código' header; 'Nome de um produto' points to the 'Nome' header; 'Quantidade do produto' points to the 'Quantidade' header; and 'Valor referente ao custo de fabricação' points to the 'Custo' header.

Código	Nome	Quantidade	Custo
1	Parafuso	1000	0,05
2	Chave de Fenda	20	4,5
3	Prego	2000	0,02

Vemos que os dados referentes a cada um têm naturezas diferentes. E na primeira linha, estão os nomes das colunas.

O registro é identificado como uma linha da lista, com conteúdo em cada tipo de dado representado pela coluna. Temos então três registros nesta lista.

A lista é denominada de tabela, ela possui um nome ou identificador próprio. Um banco de dados de certo sistema pode ser constituído de várias tabelas.

A tabela então, possui uma estrutura padronizada, composta por atributos ou campos: **CÓDIGO**, **NOME**, **QUANTIDADE** e **CUSTO**. Este conjunto ainda é referenciado como o esquema da tabela. Cada valor em um registro referente a um atributo ou campo denomina-se item de um registro.

O armazenamento físico dos dados visto anteriormente, em termos de *bytes*, é ilustrado a seguir. Desde que consideremos para nomes os campos com 15 *bytes*, campos numéricos inteiros com 4 *bytes* e campos de números de moeda com 5 *bytes*, a seguinte representação física (apenas dos dados) pode ser considerada:

0	0	0	1	P	a	r	a	f	u	s	o	0	0	3	E8					
0	0	0	0	5	0	0	0	2	C	h	a	v	e	d	e	F	e	n	d	a
0	0	0	20	0	0	0	4	32	0	0	0	3	P	r	e	g	o			
0	0	7	D0	0	0	0	0	2												

Deve-se notar que os dados estão dispostos como numa fila de bytes. O tamanho total em bytes do registro que estamos considerando é de 28 (dois campos numéricos inteiros de 4 bytes, um campo *string* de 15 bytes e um campo moeda de 5 bytes).

É importante verificar que os valores inteiros são convertidos na representação hexadecimal para depois serem armazenados. Com 4 bytes, e cada byte guardando um número na faixa de 0 a 255, existe a possibilidade de guardar valores inteiros positivos numa faixa de 0 a 4.294.967.296 (considerando números cardinais, ou não negativos). Assim, no primeiro registro o número 1.000 (em hexadecimal 3E8), é armazenado em dois bytes e com os outros dois contendo zeros: 00 00 03 E8.

A seguir temos a representação da fila de bytes convertidos em caracteres ASCII hexadecimais:

0	0	0	1	50	61	72	61	66	75	73	6F	20	20	20	20	20	20	0	0	3	E8	
0	0	0	0	5	0	0	0	2	43	68	61	76	65	20	64	65	20	46	65	6E	64	61
20	0	0	0	14	0	0	0	4	32	0	0	0	3	50	72	65	67	6F	20	20	20	20
20	20	20	20	20	20	0	0	7	D0	0	0	0	0	2								

Assim, cada caractere é convertido em seu correspondente hexadecimal ASCII quando consideramos *strings* (Cadeia de caracteres). Em geral, os campos *string* em BD correspondem a itens com tamanho máximo de 255 caracteres, podendo ser variável em outros casos.

No exemplo anterior foi considerado ainda que manipulamos uma estrutura de registro de tamanho fixo, ou seja, todo e qualquer registro gravado no arquivo sempre terá um tamanho de 28 *bytes*. Se quisermos identificar onde começa o segundo registro, nos deslocamos até a posição 29 do arquivo e os próximos 28 *bytes* se referem a ele.

Ou então, de forma genérica, se quisermos saber onde começa o registro n , basta fazermos $np + 1$, onde p é igual ao tamanho do registro.

Na maioria dos sistemas de BD atuais, trabalha-se também com o conceito de registro de tamanho variável, sendo que os espaços em branco (caractere ASCII 20), são suprimidos permitindo um aproveitamento maior do dispositivo de armazenamento.

O conceito de registro está presente nas linguagens de programação, representando uma estrutura de dados heterogênea (ou seja, composta de vários tipos). Se quiséssemos expressar em linguagem Pascal, por exemplo, a definição da estrutura acima colocada, poderíamos ter na cláusula *type* no início de uma *unit*, a definição de um *record*, seguido de uma declaração de variável na cláusula *var*:

```
type
PRODUTO = record
  Codigo: integer;
  Nome: string;
  Quantidade: integer;
  Custo: real;
end;
var
  PROD: PRODUTO;
```

Podemos dizer ainda que o código anterior trabalha com o modelo lógico de registro. Caso queiramos ler um arquivo com esta estrutura, precisaríamos de um procedimento especial, montado de forma a ler o conteúdo do arquivo. Uma proposta para isso seria:

```
Procedure Ler;  
  Var F: File;  
Begin  
  Assign(F, 'PRODUTO');  
  Reset(F);  
  While Not Eof(F) Do  
  Begin  
    ReadLn(F, Prod);  
    Write(Prod.Codigo);  
    Write(Prod.Nome);  
    Write(Prod.Quantidade);  
    Write(Prod.Custo);  
    WriteLn;  
  End;  
  Close(F);  
End;
```

O que se pode entender do algoritmo colocado acima é que cada registro será lido dentro do *loop while* e mostrado na tela. Ou seja, é necessário especificar todo o procedimento preciso para se listar o conteúdo do arquivo, ou seus registros.

As metodologias utilizadas atualmente em *softwares* de BD permitem que tal esforço seja muito bem economizado em termos de programação. Desde que fornecidas as interfaces adequadas, o processo inteiro de uma listagem ou consulta (no inglês, *query*) pode ser obtido com apenas uma declaração SQL simples.

Podemos assim enviar o seguinte comando, para uma interface de um SGBD (desde que a estrutura da tabela, o modelo lógico, esteja criada):

```
SELECT * FROM PRODUTO;
```

Após a execução já obtemos o resultado da consulta formatado:

Código	Nome	Quantidade	Custo
1	Parafuso	1000	0,05
2	Chave de Fenda	20	4,5
3	Prego	2000	0,02

O comando SELECT indica que está sendo feita uma consulta, o asterisco indica que devem ser listados todos os atributos ou campos, a cláusula FROM seguida do nome da tabela PRODUTO indicam a origem dos dados. Vê-se que não é necessário fornecer nenhuma informação a respeito do tipo dos dados que queremos, nem do tamanho de cada um, nem mesmo interessando como os dados são guardados.

Sistemas de Gerenciamento de Banco de Dados

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é o responsável por todas as tarefas pertinentes ao armazenamento, recuperação, segurança e gerenciamento dos dados. Na atualidade podemos verificar a existência de vários SGBDs no mercado.

Um sistema de informação é desenvolvido de forma conjunta com um SGBD. Enquanto que um sistema de informação está encarregado do processamento da informação em si, o SGBD tem a tarefa do gerenciamento da armazenagem da informação.

A partir de um modelo de dados requisitado (tal modelo de dados pode ser obtido através do uso de ferramentas CASE - *Computer-Aided Software Engineering* – Engenharia de Software Auxiliada por Computador) para o suporte a um sistema de informação, qualquer SGBD deverá ser capaz de permitir a implementação deste modelo em um conjunto de estruturas, bem como permitir modificações e consultas

eficientes aos dados armazenados.

Os SGBDs fornecem uma interface de conexão ao banco de dados, a qual permite a comunicação do sistema de informação com o mesmo. Um usuário requisita os processos de um sistema de informação, e este interage com o SGBD emitindo solicitações de consultas ou modificações, e o SGBD retorna os dados necessários.

Além da interação com o usuário, durante a atividade de desenvolvimento de um sistema de informação, que pode ser feito numa ferramenta RAD (*Rapid Application Development*; por exemplo, Eclipse (Programação Java) e Visual Studio (Microsoft)), a interface permite o acesso aos dados em tempo real, otimizando bastante o processo de desenvolvimento.

Com o advento da internet, os sistemas de informação romperam a barreira das redes locais e internas às empresas para disponibilizarem informações de forma global na *web*. Qualquer cliente de uma empresa pode acessar a página da mesma e comprar produtos remotamente, em qualquer parte do mundo.

Os SGBDs foram então adaptados para contemplarem esta possibilidade de conexão de bancos de dados com sistemas na *web*. A operação destes sistemas, seja em redes intranet, extranet ou internet, ficou facilitada por parte dos usuários e permitindo grande economia para as companhias em função da diminuição da redundância.

Empresas com filiais ao redor do mundo podem compartilhar dados a partir de um único local físico, onde se encontram os servidores.

Também o advento da internet influiu no desenvolvimento de SGBDs mais seguros e confiáveis, visto que as páginas colocadas na *web* têm visibilidade irrestrita, ou seja, podem ser acessadas por qualquer pessoa (e assim deve ser, pois uma empresa procura maximizar a aquisição de mais clientes em suas operações).

Data Warehouse e Data Mining

A aquisição de novos sistemas por parte de uma empresa, com novas tecnologias em relação ao que ela já possui, faz com que, na maior parte das vezes, ela compartilhe diferentes sistemas de informação em um mesmo período de tempo.

Costuma-se denominar os sistemas antigos e existentes na empresa como *sistemas legados*.

Para que os diferentes sistemas de informação compartilhem uma mesma base de dados, é necessário conjugar os diferentes SGBDs (ou mesmo pastas de tabelas em arquivos) em um único local.

Em certas situações, é possível inclusive considerar a redundância em certos conjuntos de dados, que poderão ser tratados e filtrados mais tarde. Tal tratamento pode ser a uniformização de dados, como por exemplo, gravar todos os nomes de clientes em caixa-alta, sempre separar o código postal com um

hífen, etc. dependendo da definição da forma de armazenagem dos dados. Diz também que os dados em um *data warehouse* poderão estar não normalizados.

Assim, o conceito de *data warehouse* atende a esta expectativa, no sentido de se agrupar uma grande quantidade de dados localizados em diferentes fontes, fisicamente distantes inclusive, em um único repositório. Inmon coloca que os *data warehouses* são um **conjunto de dados granulares integrados, armazenado e gerenciando os dados em um certo período de tempo, que podem ser resumidos ou agregados para a criação de novas formas de dados.**

A partir da formação de um *data warehouse*, que disponibiliza por sua vez uma grande quantidade de dados, fica possível por parte da empresa a busca de certas informações referentes a *padrões* nos dados que se repitam num certo período de tempo.

Por exemplo, pode ser constatado num sistema de CRM (Sigla de *Customer Relationship Management*, um sistema com o objetivo de gerir relacionamentos da empresa com seus clientes) um padrão de comportamento de certo grupo de clientes, numa faixa etária bem definida, que comprem determinado produto em um período específico de tempo.

Tal informação poderá ser útil para que a empresa defina políticas de *marketing* direcionadas para esse grupo de clientes, o clássico que se refere à alta correlação de compra de fraldas e cervejas em supermercados, próximo ao final de semana, é um ótimo exemplo de identificação de padrões.

É importante ressaltar que tal padrão é invisível em primeira mão, não podendo ser captado simplesmente a partir das funções específicas do sistema de informação, mas apenas após seu agrupamento em um repositório (via *data warehouse*) e abarcando por sua vez um grande período de tempo. Isto permite então a identificação sistemática de padrões subjacentes aos dados.

Tal processo consiste então nos procedimentos da mineração de dados ou *data mining*. O'Brien assinala que "no *data mining*, os dados de um *data warehouse* são processados para identificar fatores e tendências-chave nos padrões das atividades de negócios".

A mineração de dados constitui-se então de diferentes técnicas que podem ser aplicadas a um conjunto de dados para a extração de padrões. A mineração de dados pode possibilitar às empresas a identificação de boas oportunidades de negócios, a partir dos dados armazenados em seus SGBDs e outras fontes complementares de dados.

Relembrando

Nesta aula foram estudados os bancos de dados como uma parte importante dos sistemas de informação, bem como seus princípios para a construção de um bom banco de dados para as necessárias

transformações de dados em informação. Viu-se ainda a importância dos sistemas de gerenciamento de banco de dados para a manutenção e a forma de consulta aos dados.

Tais conceitos são vitais para se conhecer o *data warehouse* e o *data mining*, como operações importantes que se utilizam bancos de dados para a descoberta de padrões e outras atividades após a utilização do sistema de informação.

Referências

INMON, W. H. *et al.* **Data Warehousing – Como transformar informações em oportunidades de negócios**. Berkeley, 2001, p. 10-11.

LAUDON, K; LAUDON, J. **Sistemas de Informação Gerenciais**, 9ª edição. São Paulo: Pearson Prentice Hall, 2010.

MEDEIROS, L. F. **Banco de Dados** – Princípios e Prática. Curitiba: IBPEX, 2007.

O ´BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da internet**. Tradução de Cid Knipel Moreira. São Paulo: Saraiva, 2002.

TURBAN, E.; RAINER JUNIOR, R. K.; POTTER, R. E. **Administração de tecnologia da informação**. Rio de Janeiro: Campus, 2005.