

1. Introdução à banco de dados

1.1 Histórico

No decorrer da década de 1950 e início da década de 1960, foram desenvolvidas fitas magnéticas para o armazenamento dos dados. Entretanto, os dados armazenados nas fitas eram lidos apenas na forma sequencial. No final da década de 1960 e início da década de 1970, com o surgimento das unidades de armazenamento denominadas “discos rígidos”, a posição de armazenamento dos dados era indiferente, livrando-se assim da sequencialidade. Dessa forma, iniciou-se os primeiros projetos de criação de bancos de dados, pois havia a necessidade de criar e manipular bases de dados por meio de estruturas que fossem armazenadas no disco rígido. Surgiram então os modelos de bancos de dados em rede e hierárquicos (SILBERCHATZ *et al.*, 2006).

O modelo de Banco de dados hierárquico é descrito por um diagrama de estrutura de árvore, existiam dois componentes básicos: as caixas correspondentes aos tipos de registros e as linhas que representam as ligações entre os tipos de registro. O modelo em rede é composto de uma estrutura mais completa, possui as propriedades básicas de registros, conjuntos e ocorrências, e utiliza a linguagem de definição de bancos de dados e linguagem de manipulação de dados, além de permitir uma evolução mais eficiente do modelo hierárquico.

Os primeiros conceitos de bancos de dados relacionais surgiram nas décadas de 1960 e 1970 na empresa IBM (*Internationall Business Machines*) através de pesquisas para a automação de escritórios, pois as empresas descobriram que tinham um alto custo com pessoas que trabalhavam em funções de armazenamento e organização de arquivos.

Em 1970, o pesquisador da IBM Ted Codd publicou o artigo intitulado “*Relational Model of Data for Large Shared Data Banks*” na revista *Association for Computing Machinery* (ACM) falando sobre bancos de dados relacionais. Seu trabalho, baseado na álgebra e cálculo relacional, teve por objetivo um sistema onde o usuário podia ser capaz de acessar informações, armazenadas na base de dados, através de comandos em inglês.

Devido ao seu fraco desempenho em relação aos modelos em rede e hierárquico, o modelo relacional não era usado na prática. Na década de 1980, a IBM através do desenvolvimento de um projeto denominado “*System R*”, proporcionou a construção de um banco de dados relacional eficiente. O protótipo funcional do sistema *System R* levou a construção do primeiro produto de banco de dados relacional da IBM, o SQL/DS. A partir desse momento, surgiram novos produtos comerciais de banco de dados relacionais, como por exemplo: IBM DB2, Oracle, Ingres e REC Rdb (SILBERCHATZ *et al.*, 2006).

No início da década de 1980, surgem os primeiros esforços das organizações ANSI (*America National Standards Institute*) e ISO (*International Standards Organization*) para a padronização dos métodos de acesso aos bancos de dados relacionais. Através desses esforços criou-se então a linguagem SQL (*Structured Query Language*) utilizada atualmente como a linguagem padrão para bancos de dados relacionais e base para outros tipos de modelos pós-relacionais.

1.2 Conceitos

Na forma conceitual, um sistema de gerenciamento de banco de dados (SGBD)

é composto por uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. A coleção de dados, é denominada “banco de dados”, e contém as informações relevantes ao domínio da aplicação (SILBERCHATZ *et al.*, 2006). O conjunto de programas do SGBD é responsável por gerenciar o banco de dados, fornecendo suporte aos métodos de acesso, ou seja, formas de armazenar e recuperar informações de maneira conveniente e eficiente.

Segundo Elmasri & Navathe (2010), um SGBD é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O conjunto de programas, é portanto, um sistema de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de dados entre vários usuários e aplicações. Já o banco de dados implica em especificar os tipos de dados, as estruturas de as restrições dos dados que estarão armazenados.

Entretanto, Date (2003) cita que um sistema de banco de dados é apenas um sistema computadorizado de manutenção de registros, onde o banco de dados pode ser considerado equivalente a um armário eletrônico com funções do tipo:

- * Acrescentar novos arquivos;
- * Remover arquivos;
- * Alterar arquivos;
- * Inserir dados em arquivos existentes;
- * Buscar dados em arquivos existentes;
- * Excluir dados de arquivos existentes;
- * Alterar dados de arquivos existentes.

1.3 O uso e as vantagens

Uma maneira de exemplificar o uso de banco de dados, imagina-se que se deseja realizar uma simulação computadorizada que avalia o impacto sobre o rendimento da cultura do trigo e a intensidade da doença denominada giberela sob diferentes cenários de clima (associados a mudanças climáticas).

Para esse experimento têm-se 7 cenários de clima com dados (temperatura máxima, temperatura mínima, temperatura média, precipitação, radiação solar, umidade relativa e velocidade do vento) diários em um período de 100 anos. Cada um desses conjuntos de dados estão associados a 28 estações de uma determinada região. Serão utilizadas 3 tipos de cultivares (ciclo longo, ciclo curto, ciclo intermediário). Serão utilizadas 3 épocas de sementeiras e será rodado o modelo de simulação com e sem doença.

Tendo como base o experimento descrito, pode-se imaginar a quantidade de dados que deverão ser manipulados. Nesse caso, torna-se praticamente impossível trabalhar com essa enorme quantidade de dados usando métodos tradicionais. Nesse sentido, pode-se utilizar sistemas de gerência de banco de dados, pois eles foram projetados para gerenciar grandes blocos de informação. Esse gerenciamento envolve definir estruturas para armazenamento e fornecer mecanismos para manipulação dos dados (SILBERCHATZ *et al.*, 2006).

Algumas vantagens de se usar a abordagem de banco de dados são bem claras. Por exemplo (DATE, 2003): (i) densidade: a não necessidade de arquivos de papel volumosos; (ii) velocidade: o computador processa os dados com muito maior rapidez

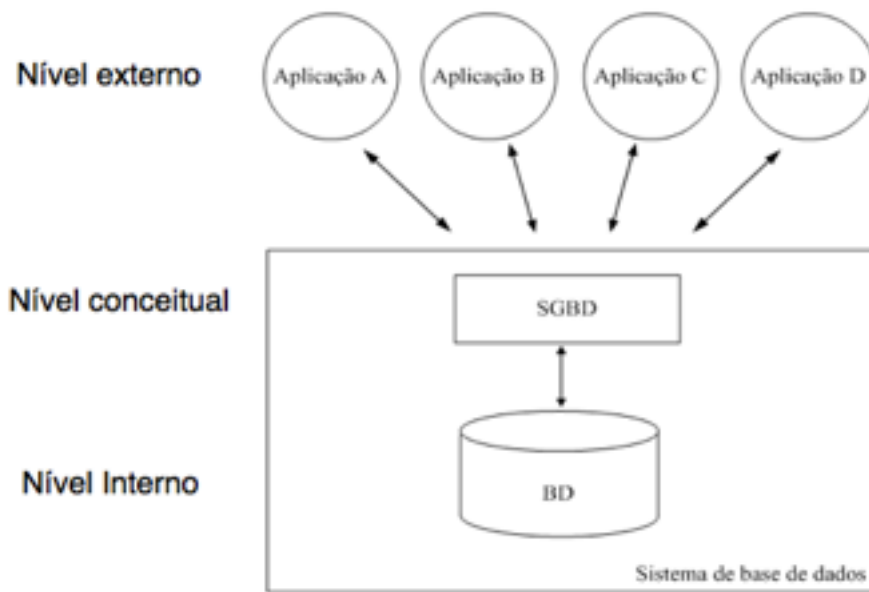
que um ser humano; (iii) menos trabalho monótono: elimina o tédio de manter e manipular dados manualmente; (iv) atualidade: disponibilidade de informações precisas e atualizadas; (v) proteção: segurança contra perda e acesso ilegal.

Na abordagem usando banco de dados, um único repositório de dados é definido uma só vez e ele pode ser acessado por diversos usuários. Nesse sentido, a abordagem de banco de dados possui algumas vantagens em relação as abordagens tradicionais, que são citadas abaixo: (ELMASRI & NAVATHE, 2010; DATE, 2003).

- * Natureza auto-descritiva: o banco de dados possui uma definição do banco de dados - metadados - armazenado em um catálogo do SGBD;
- * Isolamento entre os programas, dados e abstração de dados: têm-se uma estrutura específica para tratar os dados, independente da aplicação que os utiliza;
- * Suporte para múltiplas visões dos dados: Vários usuários podem acessar o mesmo banco de dados, entretanto, cada usuário pode visualizar somente as partes de dados de seu interesse;
- * Compartilhamento de dados e o processamento multi-usuário: Diversos usuários podem trabalhar com os dados ao mesmo tempo, no entanto, o SGBD implementa uma técnica de controle de concorrência para organizar as requisições aos dados de forma que eles sejam consistentes;
- * Redundância controlada: evitar que mesmos dados possam estar repetidos de formas diferentes;
- * Integridade mantida: assegurar que os dados armazenados no banco de dados estejam corretos, ou seja, representem de forma adequado o domínio de aplicação modelado;
- * Suporte a transações: uma transação é uma unidade lógica de trabalho do banco de dados. Ela é composta por várias execuções operações de atualização.
- * Segurança: pode-se definir níveis de segurança aos dados com base nas restrições necessárias. Também deve fornecer a possibilidade de realização de cópias de segurança (*backups*) e recuperação de cópias de segurança (*restores*).

1.4 Arquitetura

A arquitetura básica de SGBD é composta basicamente por três níveis (Figura 1). O primeiro nível é denominado nível interno e é responsável pela gerência de como os dados estão armazenados no meio físico do SGBD.



Figura

1. Arquitetura do SGBD.

O segundo nível é denominado conceitual. Esse nível é o responsável por fornecer uma visão única da aplicação. No terceiro nível, chamado de nível externo ou nível do usuário, cada aplicação possui a sua visão particular dos dados da aplicação.

2. Modelos de bancos de Dados

Uma das principais características do uso de banco de dados é permitir a abstração dos dados, ocultando detalhes do armazenamento dos dados, que são desnecessários para a maioria dos usuários de banco de dados. Segundo Elmasri & Navathe (2010), um modelo de dados fornece um conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados. O modelo de dados fornece recursos que permitem que os níveis de abstração do banco de dados possam ser executados.

Para Silberchatz *et al.* (2006) um modelo de dados tem por finalidade apoiar a estrutura do banco de dados, pois ele fornece ferramentas conceituais para descrever os dados, relações entre os dados, semântica dos dados e restrições de consistência.

Os modelos de dados podem ser divididos em níveis de abstração. Os modelos de alto nível ou conceituais, possuem conceitos que descrevem os dados da forma como os usuários os percebem. Os modelos de baixo nível ou físicos descrevem os conceitos de como os dados são implementados, armazenados e manipulados no computador. A seguir são descritos alguns modelos de dados.

2.1 Modelo relacional

Conforme citado anteriormente, o modelo relacional foi introduzido por Ted Codd em 1970. O modelo usa o conceito de uma relação matemática como seu bloco de construção básica e tem a sua base teórica na teoria dos conjuntos e na lógica de predicados de primeira ordem.

O modelo relacional representa o banco de dados como um coleção de relações (tabelas). Onde cada relação é representada por uma tabela de valores, onde cada

linha da tabela representa uma coleção de valores de dados relacionados. Esse modelo possui um conjunto de conceitos associados, que são: relações, atributos (campos) e tuplas (registros).

O conceito de relação, ou tabela, é semelhante ao conceito de conjunto da matemática, ou seja, uma tabela é constituída por uma série de elementos (atributos) que possuem características comuns. Um exemplo é mostrado na Figura 2, na tabela denominada “station”. Na mesma figura, pode-se observar o conceito de atributo ou campo. Cada atributo indica uma determinada característica associada a tabela. No caso da Figura 2, tem-se os atributos: id, description, city, nickname, google, organization, code_organiz. O domínio é caracterizado como sendo os valores possíveis do atributo. O conceito tupla, ou registro, indica uma linha da relação (um conjunto de pares atributo-valor), conforme mostra a Figura 2.

station (relação)

	id [PK] integer	description character varying(50)	city integer	nickname character var	google character(1)	organization integer	code_organiz character var
	367	LONDRINA	882	LOND		5	2351003
	368	BEBEDOURO (PETROLINA PE)	4357	BEBE		8	
T u p l a s	369	Uniao da Vitoria	1695	UNIA		11	26145103
	370	monitorada;n				11	cod_area
	371	Uniao da Vitoria	1695			11	26145103
	372	Lapa	1490			11	25474946
	373	Guaratuba	1444			11	25324831
	374	Guarapuava	1442			11	25215130
	375	Sao Miguel do Iguacu	1660			11	25115408
	376	Assis Chatoubreand	1331			11	24235332
	377	Curitiba	1400			11	25264916
	378	Santa Helena	1630			11	24505420
	379	Telemaco Borba	1680			11	24205037

Figura 2. Conceitos do modelo relacional.

2.2 Modelo orientado à objetos

O termo sistema de banco de dados orientado a objetos é usado para se referir aos sistemas de bancos de dados que aceitam um sistema de tipo orientado a objeto, e permitem acesso direto aos seus dados por uma linguagem de programação orientada a objeto usando um sistema de tipo nativo da linguagem (SILBERCHATZ *et. al.*, 2006).

Alguns conceitos são inerentes aos bancos de dados orientados a objetos. Dentre esses conceitos pode-se citar: tipos de dados complexos, tipos estruturados, herança, identificadores de objetos (OID) e tipos array. Os padrões para os métodos de acesso aos bancos de dados orientados a objeto são definidos pelo grupo ODMG (*Object Database Management Group*). As versões SQL:1999 e SQL:2003 da linguagem SQL, padrão para bancos de dados relacionais, já possuem recursos que permitem a manipulação de conceitos de orientação a objetos.

2.3 Modelo objeto-relacional

O modelo objeto-relacional surgiram como uma forma de estender os SGBDs relacionais com características presentes em SGBDs orientados a objeto (ELMASRI & NAVATHE, 2010). Alguns SGBDs comerciais como Oracle, Informix, PostgreSQL já implementam características objeto-relacional.

A linguagem SQL (SQL:1999) traz recursos para manipulação das características objeto-relacional. Dentre essas características pode-se citar (ELMASRI & NAVATHE, 2010):

- * construtores de tipos: permitem especificar objetos complexos. Dentre esses objetos inclui-se o tipo `row` e `array`. Outros construtores como `set`, `list` e `bag` ainda não fazem parte das especificações da linguagem SQL:1999;
- * Especificação de identidade de objetos: inclui um mecanismo de referência (*reference type*)
- * Mecanismos de herança: são disponibilizados comandos que permitem trabalhar com heranças. A herança é caracterizada quando um objeto possui características específicas e também herda característica de outro objeto genérico.
- * Encapsulamento de operações: é possível realizar encapsulamento de operações através de tipos definidos pelo usuário, os quais podem incluir operações como parte de suas declarações.

2.4 Modelo Geográfico

Os sistemas de informação geográfica (*Geographic Information System - GIS*) são usados para coletar, modelar, armazenar, a analisar informações que descrevem as propriedades físicas do mundo geográfico. O escopo abrange dois tipos de dados: (i) dados espaciais: que são derivados de mapas, imagens digitais, fronteiras administrativas e políticas, estradas, redes de transporte, dados físicos, como rios, características de solo, regiões climáticas e elevações da terra; (ii) dados não-espaciais: engloba dados sócio-econômicos (como contagem de censo), dados econômicos e informações de vendas ou de marketing (ELMASRI & NAVATHE, 2010).

Segundo Elmasri & Navathe (2010), as aplicações GIS podem se dividir em 3 categorias: (1) cartográfica (irrigação, análise de rendimento de culturas, avaliação de terra, gerenciamento e planejamento de instalações, estudos paisagísticos, análise de padrões de tráfego); (2) aplicações de modelagem digital de terrenos (estudos dos recursos da terra, avaliação de engenharia civil e militar, levantamento de solos, estudos de poluição do ar e da água, controle de enchentes e gerenciamento de recursos hídricos); (3) aplicações de objetos geográficos (sistemas de navegação de veículos, análise geográfica do mercado, distribuição e consumo de serviços públicos, análise econômica de produtos e serviços).

Os dados GIS podem ser representados em dois formatos: raster e vetorial. Os dados do formato raster são caracterizados como um arranjo de pontos, no qual, cada ponto representa um valor de um atributo para uma localização do mundo real. Os dados no formato vetorial representam objetos geométricos, como por exemplo, linhas, pontos e polígonos (SILBERCHATZ *et al.*, 2006; ELMASRI & NAVATHE, 2010).

Sobre os dados GIS são possíveis executar várias operações, dentre elas pode-se citar: interpolação, interpretação, análise de proximidade, processamento de imagens raster, análise de redes, união e intersecção.

2.5 XML

A tecnologia XML (*eXtensible Markup Language*) possui uma meta-linguagem

de marcação que contém métodos de acesso que possibilitam a manipulação dos dados. Um documento XML é organizado de forma hierárquica, conforme mostra a Figura 3, carrega a estrutura dos dados (*tags*) e os seus conteúdos (dados). Ele é composto por elementos simples (um exemplo: `description` linha 5) ou elementos compostos (exemplo: `station` linha 4 e 12), atributos (exemplo: `id` linha 4), instruções de processamento (exemplo: linha 1) e comentários (exemplo: linha 2) (W3C, 2012).

```
1.<xml version="1.0"/>
2.<!--exemplo de documento XML -->
3.<data>
4.    <station id="367">
5.        <description>LONDRINA</description>
6.        <city>882</city>
7.        <nickname>LOND</nickname>
8.        <google/>
9.        <organization>5</organization>
10.       <code_organiz>2351003</code_organiz>
11.    </station>
12.    <station id="368">
13.        <description>BEBEDOURO</description>
14.        <city>4357</city>
15.        <nickname>BEBE</nickname>
16.        <google/>
17.        <organization>8</organization>
18.        <code_organiz/>
19.    </station>
20.</data>
```

Figura 3. Documento XML

A tecnologia XML é composta por diversos recursos de manipulação dos documentos XML. Alguns desses recursos se assemelham aos existentes nos SGBDs relacionais. Os recursos da tecnologia XML e suas funções são citados a seguir:

- * DTD (*Document Type Definition*): primeira especificação para a definição de estrutura em documentos XML;
- * XML Schema: especificação mais completa para a definição de estrutura de documentos XML. Possui mais recursos que a DTD;
- * XPath: linguagem de consulta para documentos XML baseada em expressões de caminho;
- * XQuery: linguagem de consulta para documentos XML. Semelhante ao SQL. Possui mais recursos que o XPath;
- * XSL (*eXtensible Stylesheet Language*): Linguagem de transformação de documentos XML;
- * DOM (*Document Object Model*) e SAX (*Simple API for XML*): APIs de manipulação dos documentos XML. Conjunto de classes e métodos.

2.6 Modelo Temporal

Um banco de dados modela o estado de algum aspecto do mundo real. Normalmente os bancos de dados modelam apenas um estado, o estado atual do mundo real. Dessa forma, eles não modelam estados passados, a não ser que se faça uma modelagem específica para isso. Por padrão, quando o estado do mundo real

muda, o banco de dados é atualizado, e as informações sobre o estado antigo são perdidas.

Entretanto, em algumas situações, é necessário armazenar e recuperar informações sobre estados passados dos dados. Nesse contexto, os bancos de dados que armazenam informações sobre estados do mundo real com o passar do tempo são chamados de bancos de dados temporais (SILBERCHATZ *et al.*, 2006).

Quando trata-se da questão do tempo, pode-se observar que o tempo válido para um fato é o conjunto de intervalos de tempo durante o qual o fato é verdadeiro no mundo real. O tempo da transação é intervalo de tempo durante o qual o fato é atual no banco de dados, o qual é gerado automaticamente pelo sistema de banco de dados com base nos dados fornecidos. Uma relação temporal é caracterizada por gerenciar algum tipo de tempo, válido ou de transação. Já uma relação bitemporal gerencia os dois tipos: tempo válido e tempo de transação (SILBERCHATZ *et al.*, 2006).

A linguagem SQL define tipos de dados para se trabalhar com a questão temporal. Os tipos são: **date**, **time** e **timestamp**. Também possui um tipo de dados denominado **interval**, o qual permite armazenar um período de tempo. Várias propostas de extensão tem sido feitas para implementar a questão temporal no SQL, mas até a versão SQL:2003, não existe qualquer suporte especial além de tipos de dados e operações relacionadas ao tempo - projeção, junção e seleção (SILBERCHATZ *et al.*, 2006).

REFERÊNCIAS

DATE C. J. *Introdução a Sistemas de Banco de Dados*. 8ª Edição. Ed. Campus/Elsevier. 2003.

HEUSER, C. A. *Projeto de Banco de Dados*. 6ª Edição. Porto Alegre. Ed. Bookman. 2009.

ELMASRI R; NAVATHE S. *Sistemas de Banco de Dados*. 4ª Edição. Ed. Pearson Addison Wesley. 2005.

SILBERCHATZ A; KORTH H.; SUDARSHAN S. *Sistema de Banco de Dados*. 5ª Edição. Editora Elsevier. 2006.