

Sistemas de Gerência de Banco de Dados (SGBD)

Estudamos um pouco dos bancos de dados, principalmente, os modelos hierárquicos, orientados a objetos e relacionais. Neste material, abordaremos detalhes dos **sistemas de gerência de banco de dados relacionais**. Você já deve ter acessado o site de alguns deles... Agora, veremos os conceitos e as características relacionados a esses sistemas. Boa jornada!

Vamos relembrar: o que são os bancos de dados? Os bancos de dados podem ser considerados como componentes essenciais na vida das pessoas que vivem na sociedade moderna. Muitas vezes, ao realizarmos atividades de nosso cotidiano, estamos utilizando banco de dados e não percebemos.

Como exemplo, podemos imaginar a situação de ir a uma instituição financeira para realizar um depósito de dinheiro, ou acessar um site de internet com o objetivo de comprar um produto on-line. Provavelmente, nessas atividades, uma pessoa ou algum programa de computador acessará um banco de dados.

Na área de jogos digitais, utilizaremos banco de dados para armazenar as informações referentes ao jogo. Por exemplo, podemos armazenar o progresso no jogo, as configurações e até mesmo as imagens que serão utilizadas. Rabin (2013) argumenta que uma arquitetura de jogo deve fazer a separação dos dados do código para que os não programadores possam modificar o jogo com mais facilidade. A maneira de fazer isso pode ser armazenando os dados em arquivos, em XML ou em bancos de dados.

Kozovits (2003) destaca como vantagens da utilização de banco de dados relacionais em aplicações de jogos multiplayer a eficiência de suas consultas, a estrutura de relação dos dados e a segurança das transações. Essas características todas são implementadas e mantidas pelo SGBD.

Administrador de banco de dados (DBA)

Este profissional é responsável por autorizar o acesso, coordenar e monitorar a utilização do banco, adquirir recursos de *hardware* e *software* quando necessário e avaliar o desempenho e os problemas de segurança do sistema.

Projetistas de banco de dados

São responsáveis por identificar os dados a serem armazenados, bem como identificar as estruturas apropriadas para representar e armazenar esses dados.

Analista de sistemas e programador de aplicações

São responsáveis por determinar as necessidades dos usuários finais e implementar as aplicações de banco de dados.

Produtores de ferramentas para o SGBD

Produzem ferramentas opcionais na forma de pacotes que podem ser utilizados no SGBD, como pacotes para monitoria de desempenho ou geração de dados de testes.

Projetistas do SGBD

São eles que projetam e implementam o SGBD.

Usuário final

Pode interagir com o banco de dados por meio de aplicações de banco de dados ou pelas linguagens do SGBD.

2/10

do SGBD. • Analista de sistemas e programador de aplicações: são responsáveis por determinar as necessidades dos usuários finais e implementar as aplicações de banco de dados. • Projetistas do SGBD: são eles que projetam e implementam o SGBD. • Produtores de ferramentas para o SGBD: produzem ferramentas opcionais na forma de pacotes que podem ser utilizados no SGBD, como pacotes para monitoria de desempenho ou geração de dados de testes.

Vantagens de utilizar um SGBD

Controle de redundância: as mesmas informações não são armazenadas várias vezes. Desta maneira, o espaço de armazenamento é reduzido, e a inconsistência de informações é evitada.

Restrição de acesso não autorizado: os usuários possuem senhas para acesso, e existem níveis de permissão. Dessa forma, um usuário sem experiência não terá a possibilidade de apagar informações importantes em um banco de dados.

Armazenamento persistente para objetos do programa: podem ser usados para oferecer armazenamento persistente para objetos e estruturas de dados de programas, por exemplo: objetos complexos escritos em linguagens de programação como C++ e Java.

Estruturas de armazenamento e técnicas de pesquisa para processamento eficiente de consultas: os SGBDs possuem estruturas de dados especializadas para aumentar a velocidade de consultas e atualizações; uma dessas estruturas são os chamados índices.

Múltiplas interfaces do usuário: são fornecidas várias formas de acesso aos dados armazenados. Para usuários leigos, existem as interfaces gráficas (GUIs – Graphical User Interfaces).

Restrições de integridade: deve ser oferecida a capacidade de restrições de integridade, como a especificação de tipos válidos de dados para um determinado item armazenado no banco.

Execução de ações utilizando regras: permite que determinadas ações sejam executadas automaticamente quando determinados eventos e condições ocorram. Esse é o caso dos gatilhos (triggers), que são regras ativadas por atualizações em tabelas nos bancos de dados.

Implicações adicionais do uso da abordagem de banco de dados: maior padronização, flexibilidade, menor tempo para desenvolvimento de aplicações, disponibilidade de informações atualizadas e economia.

Quando não usar um SGBD

Devido aos custos adicionais do uso de um SGBD:

- há um alto investimento inicial em hardware, software e treinamento;

- oferece generalidade para a definição e o processamento de dados;

- há um esforço adicional para oferecer funções de segurança, controle de concorrência, recuperação e integridade.

Algumas circunstâncias tornam desejável o uso de arquivos comuns para o armazenamento de informações, são elas:

- aplicações que utilizem banco de dados simples, bem definidas e que não necessitem de mudanças frequentes;

- requisitos rigorosos, que envolvam programas que devem ser executados em tempo real;

- sistemas embarcados com capacidade limitada de armazenamento;

- aplicações que não utilizam múltiplos usuários requisitando acesso às informações armazenadas no banco de dados.

Quando se trata de jogos MMOG, duas principais razões justificam a utilização de SGBD: o sistema de cobrança e a autenticação de usuários. Este tipo de aplicação, comum também no mundo dos negócios, consolidou a aplicação de SGBDs para essas finalidades, argumenta Kozovitz (2003).

Kozovitz (2003) ainda sugere que deve ser realizada uma avaliação da necessidade de utilizar SGBDs para armazenamento dos dados do jogo. É que é importante pensar em uma arquitetura para o jogo que permita a utilização de um SGBD de baixo custo.

Conceitos básicos para a utilização de um SGBD

Modelo de dados

É uma coleção de conceitos que podem ser utilizados para descrever a estrutura de um banco de dados (tipos de dados, relacionamentos e restrições) e também um conjunto de operações básicas para especificar recuperações e atualizações no banco de dados (HEUSER, 2009; ELMASRI, 2011).

Categorias de modelos de dados

Existem diversos tipos de modelos e diferentes formas de classificação. Abordaremos a classificação que leva em conta o tipo de conceitos que os modelos de dados utilizam para descrever a estrutura do banco de dados.

Seguindo essa classificação, os modelos de dados podem ser categorizados das formas que explicaremos a seguir.

Modelos de dados conceituais ou de alto nível	Modelos de dados físicos ou de baixo nível	Modelos de dados de implementação
Oferecem conceitos que são próximos ao modo como os usuários percebem os dados.	Fornecem uma visão mais detalhada do modo como os dados estão realmente armazenados no computador.	Fornecem ao usuário final conceitos que podem ser facilmente entendidos, não ficando distante do modo como os dados são organizados e armazenados no computador. Estes modelos são utilizados com frequência em SGBDs comerciais tradicionais.

• Modelos de dados conceituais ou de alto nível Oferecem conceitos que são próximos ao modo como os usuários percebem os dados. • Modelos de dados físicos ou de baixo nível Fornecem uma visão mais detalhada do modo como os dados estão realmente armazenados no computador. • Modelos de dados de implementação Fornecem ao usuário final conceitos que podem ser facilmente entendidos, não ficando distante do modo como os dados são organizados e armazenados no computador. Estes modelos são utilizados com frequência em SGBDs comerciais tradicionais.

Esquemas e instâncias

Esquema: é a descrição do banco de dados, especificada durante o projeto do banco de dados e não se espera que mude com frequência.

Instância: é o banco de dados em si. São os dados atuais armazenados no banco de dados em um momento particular. Também pode ser chamada de estado do banco de dados.

Arquitetura de três esquemas e interdependência de dados

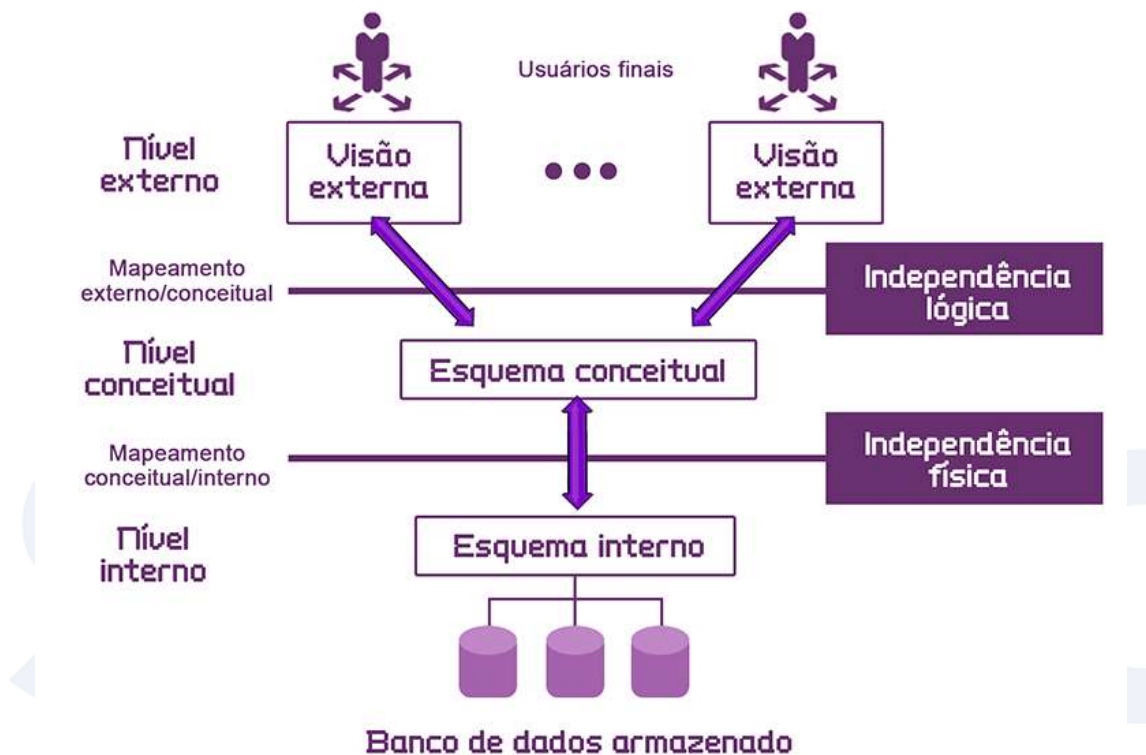
A arquitetura de três esquemas foi proposta pelo comitê ANSI/SPARC visando a facilitar o entendimento de algumas características do enfoque de banco de dados:

- ◆ isolamento de programas e dados;
- ◆ suporte de visões múltiplas de usuários;
- ◆ uso de um catálogo para armazenar a descrição (esquema) do banco de dados.

Arquitetura de três camadas

O objetivo da arquitetura de três esquemas é separar as aplicações do usuário do banco de dados físico. Nesta arquitetura, os esquemas podem ser definidos nos três níveis demonstrados na figura 1 e descritos a seguir.

- ◆ **Nível interno ou esquema interno:** usa um modelo de dados que mostra a estrutura de armazenamento físico do banco de dados, os detalhes dos dados guardados e os caminhos de acesso.
- ◆ **Nível conceitual ou esquema conceitual:** uma descrição global do banco de dados, descrevendo a estrutura do banco de dados como um todo, sem fornecer detalhes do modo como os dados estão fisicamente armazenados.
- ◆ **Nível externo ou esquema de visão:** descreve as visões do banco de dados para um grupo de usuários. Cada visão descreve a quais porções do banco de dados um grupo de usuários terá acesso.



A imagem está mostrando um esquema com as camadas dispostas de cima para baixo, iniciando com o nível externo, mostrando imagens de dois usuários, cada um com um retângulo embaixo, com o texto "visão externa" no centro. Entre os dois retângulos há reticências. Destes retângulos, partem setas relacionando-os com o conteúdo da camada abaixo. Entre a camada do nível externo e a do nível conceitual, há uma linha separando com um retângulo na ponta, contendo o texto: independência lógica. No nível conceitual, camada do meio, há um retângulo com o seguinte texto no centro: "esquema conceitual". No nível interno, há um retângulo com o texto "esquema interno" ligando a três cilindros representando o banco de dados. Entre o nível interno e o nível conceitual, há uma linha contendo um retângulo com o texto "Independência física".

Figura 1 – A arquitetura de três esquemas

Esquemas e instâncias

A independência de dados pode ser definida como a capacidade de se alterar um esquema em um nível do banco de dados sem ter que alterar um nível mais alto (Figura 1).

Existem dois tipos de independência de dados.

Independência lógica de dados: permite alterar apenas o nível conceitual, não havendo nenhuma alteração no nível externo ou nas aplicações do usuário.

Independência física de dados: permite alterar o nível interno sem ter que alterar o nível conceitual, nível externo ou as aplicações do usuário.

Linguagens do SGBD

Têm a finalidade de garantir a especificação do esquema de um banco de dados e permitir consultas a ele, bem como atualizações.

As linguagens do SGBD podem ser expressas das formas a seguir.

DDL – Data Definition Language (Linguagem de Definição de Dados)	DML – Data Manipulation Language (Linguagem de Manipulação de Dados)	DCL – Data Control Language (Linguagem de Controle de Dados)	VDL – Vision Definition Language (Linguagem de Definição de Visões)
É utilizada para especificar o esquema do banco de dados. As expressões na DDL são interpretadas gerando a especificação de um conjunto de tabelas que é armazenada no catálogo do banco de dados. Exemplos de comandos: CREATE e ALTER .	É utilizada para permitir a realização de consultas no banco de dados. Exemplos de comandos: SELECT e DELETE .	É utilizada para controlar o acesso aos dados no banco de dados. Exemplos de comandos: GRANT e REVOKE .	É utilizada para a definição de visões no banco de dados. Exemplo de comando: CREATE VIEW .

• DDL – Data Definition Language (Linguagem de Definição de Dados) É utilizada para especificar o esquema do banco de dados. As expressões na DDL são interpretadas gerando a especificação de um conjunto de tabelas que é armazenada no catálogo do banco de dados. Exemplos de comandos: **CREATE** e **ALTER**. • DML – Data Manipulation Language (Linguagem de Manipulação de Dados) É utilizada para permitir a realização de consultas no banco de dados. Exemplos de comandos: **SELECT** e **DELETE**. • DCL – Data Control Language (Linguagem de Controle de Dados) É utilizada para controlar o acesso aos dados no banco de dados. Exemplos de comandos: **GRANT** e **REVOKE**. • VDL – Vision Definition Language (Linguagem de Definição de Visões) É utilizada para a definição de visões no banco de dados. Exemplo de comando: **CREATE VIEW**.

Interfaces de SGBDs

As interfaces amigáveis ao usuário oferecidas por um SGBD podem incluir:

Interfaces baseadas em menu para clientes web ou de navegação	Interfaces baseadas em formulários	Interfaces gráficas com o usuário (GUI)	Interfaces para o DBA
Apresentam uma lista de menus que conduzem a formulação de uma solicitação.	Um formulário é apresentado para o usuário a fim de ser preenchido.	Exibe um esquema para o usuário na forma de diagramas. A consulta é construída ao se manipular o diagrama.	Interface onde o DBA pode configurar todos os detalhes técnicos definidos nos modelos, usando comandos ou opções do software.

• Interfaces baseadas em menu para clientes web ou de navegação Apresentam uma lista de menus que conduzem a formulação de uma solicitação. • Interfaces baseadas em formulários Um formulário é apresentado para o usuário a fim de ser preenchido. • Interfaces gráficas com o usuário (GUI) Exibe

um esquema para o usuário na forma de diagramas. A consulta é construída ao se manipular o diagrama. • Interfaces para o DBA Interface onde o DBA pode configurar todos os detalhes técnicos definidos nos modelos, usando comandos ou opções do software.

Utilitários de SGBDs

Os SGBDs possuem utilitários de banco de dados que ajudam o DBA a gerenciar o sistema. Esses utilitários possuem os seguintes tipos de funções básicas:

- ◆ Carga de dados
- ◆ Cópia de segurança (backup)
- ◆ Reorganização do armazenamento do banco de dados
- ◆ Monitoria do desempenho

Classificação

Como forma de classificar um SGBD, pode-se utilizar o modelo de dados no qual ele se baseia. O principal modelo de dados usado atualmente em muitos SGBDs comerciais é o modelo de dados relacional. Alguns exemplos de SGBDs relacionais são: Oracle, Microsoft SQL Server, MySQL, PostgreSQL, entre outros.

Outro critério adotado para classificação é o número de usuários suportado pelo sistema. Desta forma, os sistemas são classificados como monousuário (admite apenas um usuário de cada vez) ou multiusuários (admite múltiplos usuários simultaneamente).

O terceiro critério é o número de locais onde o banco de dados está distribuído. Como classificação básica para número de locais, utilizam-se os termos centralizado (se os dados estiverem armazenados em um único computador) e distribuído (podem ter o banco de dados real e o SGBD distribuídos por vários locais, conectados por uma rede de computadores).



Conclusão Vimos os conceitos e as características, bem como os papéis envolvidos no gerenciamento de um SGBD. Com esses conceitos, percebemos que trabalhar com um SGBD implica muitas tarefas. Da mesma forma que ele esconde detalhes com os quais não devemos nos preocupar, ao trabalhar com banco de dados, trabalhamos com diferentes modelos e esquemas para gerenciar os dados e organizá-los de modo eficiente e que proporcione flexibilidade ao desenvolvimento. Este material, somado aos outros desta unidade sobre este tema, dispõe o conjunto de informações necessárias para que você desenvolva a competência para utilizar bancos de dados no projeto de jogo. Continue visualizando os conteúdos a seguir.

Referências bibliográficas

- Elmasri, R., Navathe, S.B. **Sistemas de banco de dados**. São Paulo: Pearson Brasil, 2011.
- HEUSER, Carlos Alberto. **Projeto de Bancos de Dados**. Vol 4. Porto Alegre: Editora Sagra Luzzatto, 2009.
- Kozovitz Lauro, E.; Melo Rubens N., Feijó Bruno. **Um estudo do uso de SGBDs relacionais em arquiteturas de jogos multi-jogador**. Rio de Janeiro: PUC, 2003.
- Disponível em: . Acesso em: 19 nov. 2015.