

Banco de Dados - Ciclo 1 - Webaula 9



WEBAULA 9

≡ Arquitetura de um sistema gerenciador de banco de dados relacional (SGBDR).

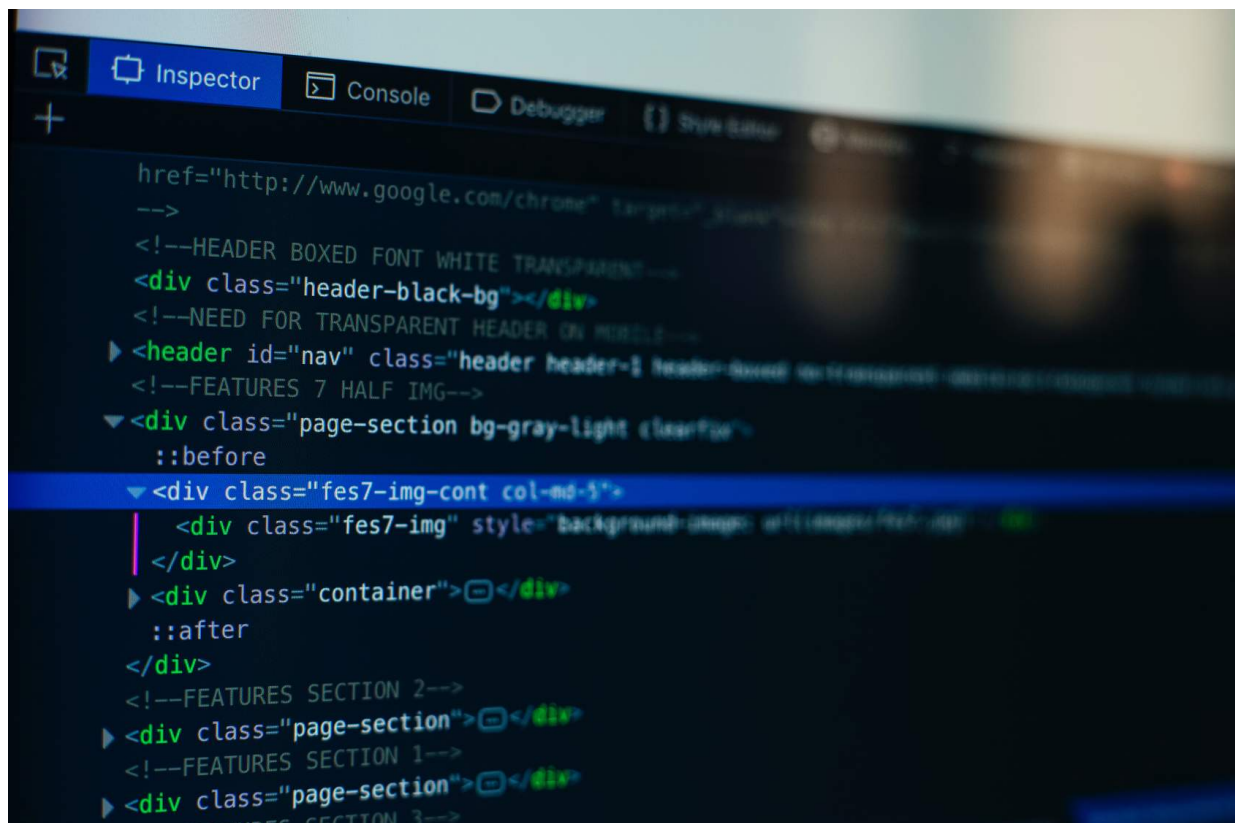
≡ Referências

QUESTION BANKS

Arquitetura de um sistema gerenciador de banco de dados relacional (SGBDR).

Na seção anterior foi possível conhecer os modelos de bancos de dados, ou seja, quais as principais possibilidades de organização de estruturas para o armazenamento dos dados. Atualmente é possível encontrar aplicações nos mais variados segmentos, cada qual com suas especificidades, e com os mais variados modelos de bancos de dados apresentados.

No entanto, o modelo relacional é atualmente o mais utilizado, principalmente em sistemas de informação que se propõem a manter dados do cotidiano das empresas. Agora vamos abordar com maior ênfase o modelo relacional, principalmente sua evolução para os sistemas gerenciadores de bancos de dados relacionais (SGBDRs).



Mas, antes, faremos uma analogia entre o armazenamento baseado em arquivos e os dados mantidos em SGBDRs. Esta etapa é fundamental e vai ajudá-lo a compreender os diferenciais de um SGBDR e sua importância no contexto do desenvolvimento de sistemas, em especial os corporativos.

Processamento de arquivos versus SGBDR

Segundo Elmasri e Navathe (2012), quando o modelo relacional foi apresentado, na década de 70, tinha-se clareza quanto a como os dados seriam organizados e armazenados: através de tabelas e relações entre as tabelas com valores próprios (chaves). Porém, ainda segundo o autor, não havia uma solução para controlar essas relações e muitos outros aspectos que iriam surgir ao longo do tempo. Eis que durante alguns anos o modelo foi utilizado através de arquivos gerenciados pelo sistema operacional, caracterizando o que se denominou processamento de arquivos.

Algumas características marcaram o processamento tradicional de arquivos (ELMASRI; NAVATHE, 2012):

1

Definição dos dados como parte do código da aplicação:

As estruturas utilizadas pelo sistema eram definidas e mantidas pela aplicação, ou seja, as propriedades sobre as estruturas, como nomes de arquivos, atributos, tipos de dados e mecanismos de acesso, entre outras, faziam parte da codificação da aplicação, estabelecendo certa dependência.

2

Ausência de múltiplas visões sobre os dados:

Os dados armazenados não dispunham de funcionalidades para que os usuários pudessem visualizá-los sob diferentes visões. Imagine a apuração de um relatório de vendas agrupado por vendedores, por exemplo. Para tanto, seria necessário o processamento significativo dos arquivos envolvidos com total controle da aplicação.

3

Controle do compartilhamento dos dados deficitário:

Mesmo naquela época era comum uma aplicação ser acessada por vários usuários simultaneamente. Por isso era necessário controlar o acesso aos dados para evitar inconsistências. Este controle era feito pela aplicação em conjunto com as funcionalidades disponíveis no sistema operacional – era, portanto, suscetível a erros de codificação.

4

Dados armazenados sem mecanismos de controle de segurança:

Os arquivos eram gerenciados pela aplicação, assim como o acesso aos registros e as operações sobre eles. Dessa forma, o controle da segurança sobre os dados era estabelecido na codificação da aplicação, o que também está propenso a erros. Além disso, não havia mecanismos de recuperação (em caso de falhas) e de criptografia dos dados armazenados. Boa parte dessa responsabilidade também era atribuída ao sistema operacional.

Durante muitos anos, empresas de desenvolvimento de softwares utilizaram as soluções baseadas no processamento tradicional de arquivos. Atualmente há muitos sistemas que manipulam grandes volumes de dados que ainda utilizam essa solução. Esses sistemas são denominados sistemas legados, e dificilmente deverão migrar para novas formas de estrutura de armazenamento, pois o custo da migração é significativo.

Com o passar dos anos, empresas que lançaram soluções para o armazenamento de dados utilizando o modelo relacional identificaram a necessidade de incorporar funcionalidades à proposta inicial de Edgar Frank Codd. Assim surgiu o sistema gerenciador de banco de dados relacional (SGBDR).



Dica

Utilizamos o termo “sistema legado” para caracterizar um sistema antigo que permanece em operação. Em geral, são sistemas que utilizam tecnologias obsoletas, ou seja, antigos não mais usuais.

Segundo Date (2004), Edgar Frank Codd, precursor do modelo relacional, definiu um conjunto de regras básicas que caracterizam a abordagem, denominadas "as 13 regras de Codd". Antes de abordar o conceito de SGBDR, é importante conhecer estas regras. Ainda segundo Date (2004), na década de 80, Codd, como era conhecido, publicou um artigo onde definia as 13 regras para que um SGBD fosse considerado relacional. Elas estão descritas a seguir:



"1.ª – Regra fundamental - Um SGBD relacional deve gerir os seus dados usando apenas suas capacidades relacionais, ou seja, tabelas e relações entre tabelas."



"2.ª – Regra da informação - Toda informação deve ser representada de uma única forma: como dados em uma tabela, ou seja, a tabela é a unidade básica de armazenamento."



"3.ª – Regra da garantia de acesso - Todo dado (valor atômico) pode ser acessado logicamente (e unicamente) através do nome da tabela, do valor da chave primária da linha, e do nome da coluna."



"4.ª – Tratamento sistemático de valores nulos - Os valores nulos (diferentes do zero, da string vazia, da string de caracteres em branco e de outros valores não nulos) existem para representar dados não existentes de forma sistemática, independentemente do tipo de dado."



"5.ª – Catálogo on-line baseado no modelo relacional - A descrição do banco de dados é representada no nível lógico como dados ordinários, isto é, em tabelas. Isso permite que usuários autorizados manipulem o metadados da mesma forma como é manipulado o conjunto de dados, em termos de consulta."

6

"6.ª – Regra da sublinguagem compreensiva - Um sistema relacional pode suportar várias linguagens e formas de uso, mas deve possuir ao menos uma linguagem com sintaxe bem definida e expressa por cadeia de caracteres, com habilidade de apoiar a definição de dados, a definição de visões, a manipulação de dados, as restrições de integridade, a autorização e a fronteira de transações."

7

"7.ª – Regra da atualização de visões - Toda visão que for teoricamente atualizável será também atualizável pelo sistema."

8

"8.ª – Inserção, atualização e eliminação de alto nível - A capacidade de manipular a relação base ou as relações derivadas como um operador único não se aplica apenas à recuperação de dados, mas também à inserção, alteração e eliminação de dados.

"

9

"9.ª – Independência física dos dados - Programas de aplicação ou atividades de terminal permanecem logicamente inalteradas, quaisquer que sejam as modificações na representação de armazenagem ou nos métodos de acesso internos."

10



“10.ª – Independência lógica dos dados - Programas de aplicação ou atividades de terminal permanecem logicamente inalteradas quaisquer que sejam as mudanças de informação que permitam, teoricamente, a não alteração das tabelas base.”



“11.ª – Independência de integridade - As relações de integridade específicas de um banco de dados relacional devem ser definidas em uma sublinguagem de dados e armazenadas no catálogo, e não em programas.”



“12.ª – Independência de distribuição - A linguagem de manipulação de dados deve possibilitar que as aplicações permaneçam inalteradas, estejam os dados centralizados ou distribuídos fisicamente.”



“13.ª – Regra da não subversão - Se o sistema relacional possui uma linguagem de baixo nível (um registro por vez), não deve ser possível subverter ou ignorar as regras de integridade e restrições definidas no alto nível (muitos registros por vez).”

Encerramento

Portanto, segundo Date (2004), o Sistema Gerenciador de Banco de Dados é o software que manipula todos os acessos ao banco de dados relacional.

O SGBDR – “R” de relacional – é um software complexo que envolve um conjunto de processos que garante, além do acesso, a correta manutenção sobre os dados. Todas as deficiências identificadas no modelo de processamento tradicional de arquivos são solucionadas pelo SGBDR. Além disso, inúmeras outras funcionalidades são implementadas e disponibilizadas, como veremos na próxima aula.

CONTINUE

Referências

Referências

BEIGHLEY, Lynn. Use a cabeça SQL. Rio de Janeiro: Alta Books, 2008.

DATE, C. J. Introdução a sistemas de bancos de dados. Rio de Janeiro: Elsevier, Campus, 2004.

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6. ed. São Paulo: Pearson Education do Brasil; Addison Wesley, 2012.

GARCIA-MOLINA, H. Implementação de sistemas de bancos de dados. Rio de Janeiro: Campus, 2001.

KHOSHAFIAN, Setrag. Banco de dados orientado a objeto. Rio de Janeiro: Infobook, 1994.

LISBOA, F. Zend Framework: desenvolvendo em PHP 5 orientado a objetos com MVC. São Paulo: Novatec, 2008.

MULLER, Robert J. Projeto de banco de dados: usando UML para modelagem de dados. São Paulo: Berkeley, 2002.

RAMAKRISHNAN, R.; GEHRKE, J. Sistemas de gerenciamento de banco de dados. São Paulo: McGraw-Hill, 2008.

SILBERSCHATZ, A. Sistema de banco de dados. Rio de Janeiro: Elsevier, 2012.

SILVA, L. Banco de dados para web: do planejamento à implementação. São Paulo: Érica, 2001.