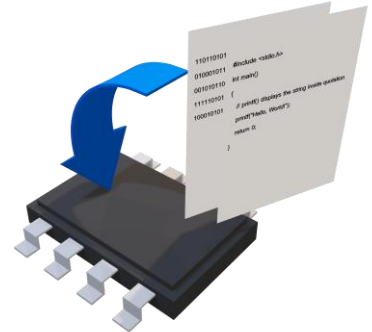




BANCO DE DADOS PARA ENGENHARIA

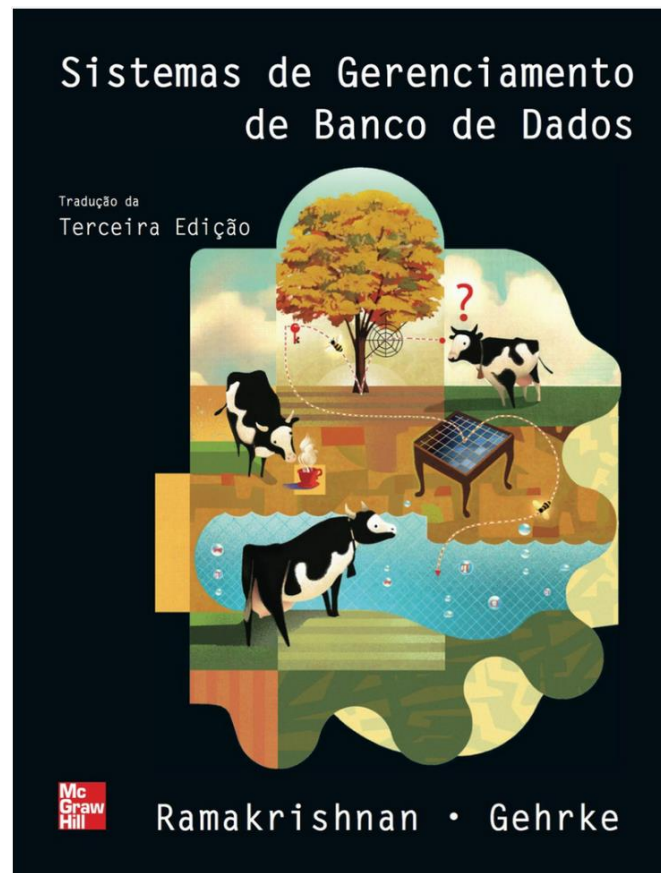
Prof. Armando L. Keller

alkeller@unisinos.br



Bibliografia

RAMAKRISHNAN, R. **Sistemas de gerenciamento de banco de dados.** Porto Alegre: AMGH, 2008

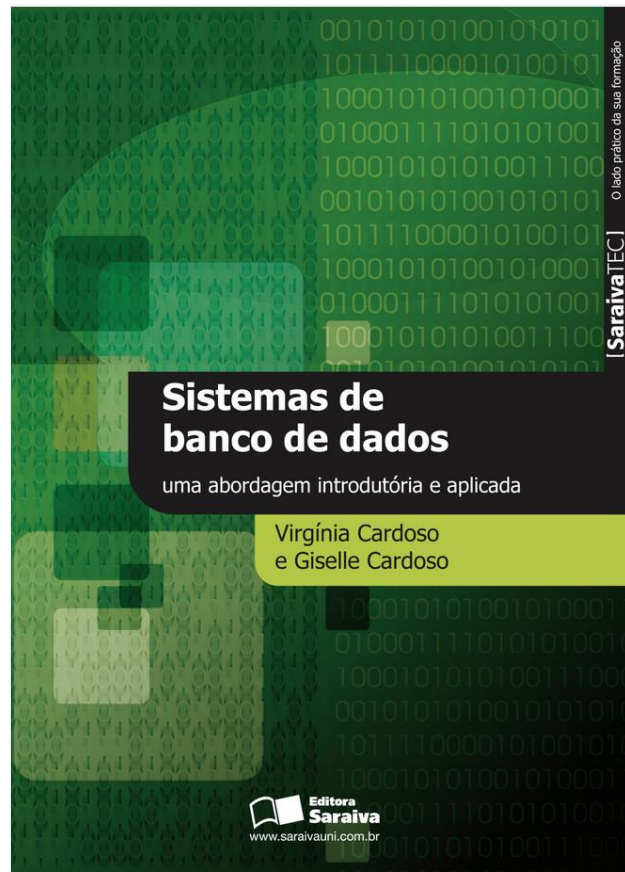


BANCO DE DADOS PARA ENGENHARIA

Prof. Armando Leopoldo Keller

Bibliografia

CARDOSO, V. M. **Sistemas de banco de dados**. São Paulo: Saraiva, 2008.



BANCO DE DADOS PARA ENGENHARIA

Prof. Armando Leopoldo Keller

Introdução

- O que são bancos de dados?

Bancos de dados são coleções de dados.

Estas coleções normalmente descrevem atividades de organizações relacionadas.

Introdução

Exemplo de informações que podem ser armazenadas em um banco de dados de uma universidade:

ENTIDADES

- Alunos
- Professores
- Cursos
- Turmas
- Salas

RELACIONAMENTOS entre entidades

- Matrícula dos alunos nos cursos
- Cursos ministrados por professor
- Uso de salas por cursos

Sistemas de Gerenciamento de Banco de Dados (SGBD)

Sistema de Gerenciamento de Banco de Dados (SGBD)

Os sistemas de gerenciamento de banco de dados (SGBD) são os softwares que auxiliam na utilização e manutenção de grandes conjuntos de dados.

Sem a utilização de um SGBD seria necessário armazenar os dados em arquivos e desenvolver softwares específicos para gerenciá-los.

Sistema de Gerenciamento de Banco de Dados (SGBD)

Exemplos de SGBDS:



Vantagens de um SGBD sobre arquivos de sistemas

- **Independência dos dados:** o SGBD fornece uma visão abstrata dos dados que oculta os detalhes de representação e armazenamento dos dados;
- **Acesso eficiente aos dados:** É obtido utilizando diversas técnicas para armazenar e recuperar dados.

Vantagens de um SGBD sobre arquivos de sistemas

- **Integridade e segurança dos dados:** Garantido que os dados sejam acessados sempre através do SGBD é possível forçar restrições de integridade como validações dos dados. É possível utilizar também controles de acesso, garantido que diferentes tipos de dados sejam acessíveis a diferentes classes de usuários;

Vantagens de um SGBD sobre arquivos de sistemas

- **Administração dos dados:** Os dados podem ser organizados de maneira que sejam minimizadas as redundâncias que seja feita uma recuperação mais eficiente dos dados;
- **Acesso concorrente e recuperação de falha:** Os SGBDs gerenciam o acesso concorrente aos dados, garantido a integridade dos mesmos, além de proteger os usuários dos efeitos de falhas de sistema;

Vantagens de um SGBD sobre arquivos de sistemas

- **Tempo reduzido de desenvolvimento de aplicativo:** Um SGBD bem projetado implementa funções que facilitam o acesso aos dados por aplicativos, tornando o acesso pelos aplicativos mais simples e rápido.

Descrevendo e armazenando dados em um SGBD

Para realizar o projeto do armazenamento de dados em um SGBD é necessário entender o modelo de dados que este utiliza.

Um dos modelos mais utilizados, e o primeiro que será visto neste curso, é o **modelo de dados relacional**.

Descrevendo e armazenando dados em um SGBD

Estes modelos de dados são compostos por construtores de alto nível que nem sempre são suportados pelo SGBD.

O modelo de dados semântico, por exemplo, é mais abstrato e facilita a descrição dos dados, normalmente utilizado para obter a descrição inicial dos dados. Após a modelagem este modelo é traduzido para ou modelo suportado pelo SGBD como o relacional.

Descrevendo e armazenando dados em um SGBD

Um dos modelos semânticos mais utilizados é o **modelo entidade-relacionamento (ER)**, que permite visualizar graficamente as entidades e os relacionamentos entre elas.

Modelo Relacional

Modelo Relacional

A base para a descrição de dados no modelo relacional são as **relações**, que podem ser consideradas como um conjunto de registros;

Esta descrição de dados em termo de modelos de dados é denominada de **esquema** (*schema*);

Modelo Relacional

Para o modelo relacional, o esquema de uma relação especifica o seu nome, o nome de cada **campo** (ou **coluna** ou **atributo**), além do tipo de cada campo.

Modelo Relacional

Por exemplo a informação sobre cada aluno em um banco de dados pode ser armazenada em uma relação com o seguinte esquema:

Alunos (**id-aluno**:string, **nome**:string, **login**:string, **idade**:integer, **média**: real)

Obs: Não é uma boa prática armazenar dados como idade, pois estes podem variar sem uma atualização do banco de dados. É mais eficiente armazenar a data de nascimento e fornecer a idade calculada a cada consulta.

Modelo Relacional

Neste esquema é possível verificar que cada registro na relação Alunos possui cinco campos, com os seus nomes e tipos identificados.

Estes dados podem ser organizados em forma de tabela, onde cada linha é um registro que descreve um aluno, e cada coluna é um campo ou atributo.

Modelo Relacional

Exemplo de instância da relação alunos.

<i>id-aluno</i>	<i>nome</i>	<i>login</i>	<i>idade</i>	<i>média</i>
53666	Jones	jones@cs	18	3,4
53688	Smith	smith@ee	18	3,2
53650	Smith	smith@math	19	3,8
53831	Madayan	madayan@music	11	1,8
53832	Guldu	guldu@music	12	2,0

Fonte: RAMAKRISHNAN, R

Modelo Relacional

Esta descrição de um conjunto de alunos ainda não está completa, para ela ser mais precisa é possível utilizar as **restrições de integridade**;

As restrições de integridade são basicamente as condições que os registros devem satisfazer. Como por exemplo todo o aluno deve possuir um valor único para o campo id-aluno, de modo que dois alunos não tenham o mesmo valor.

Outros modelos de dados

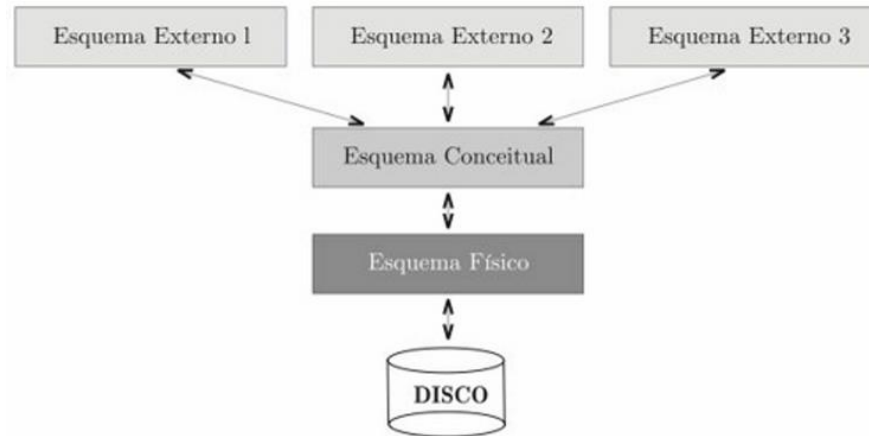
Além do modelo relacional existem outros modelos de dados como:

- Modelo hierárquico
- Modelo de rede
- Modelo orientado a objetos
- Modelo objeto-relacional

Níveis de abstração em um SGBD

Níveis de abstração em um SGBD

A descrição dos dados em um SGBD é separada em três níveis de abstração: Conceitual, físico e externo.



Fonte: RAMAKRISHNAN, R

Níveis de abstração em um SGBD

Para a definição dos esquemas externos e conceituais, é utilizada uma **linguagem de definição de dados** (*DDL – Data Definition Language*);

A DDL mais utilizada, é a **SQL** (*Structured Query Language*);

Alguns SGBDs suportam comandos SQL para descrever aspectos do esquema físico, mas estes não fazem parte da linguagem SQL padrão;

Esquema conceitual

O **esquema conceitual** também é conhecido como **esquema lógico** do SGBD;

Ele é responsável por descrever os dados armazenados em termos do modelo de dados do SGBD. No caso do SGBD relacional este esquema descreve todas as relações que estão armazenadas no banco de dados.

Esquema conceitual

O **esquema conceitual** também é conhecido como **esquema lógico** do SGBD;

Ele é responsável por descrever os dados armazenados em termos do modelo de dados do SGBD. No caso do SGBD relacional este esquema descreve todas as relações que estão armazenadas no banco de dados.

Esquema conceitual

No exemplo do banco de dados de uma universidade as relações contêm informações sobre as entidades, como aluno e professores, e sobre os relacionamentos, como a matrícula dos alunos nos cursos

Alunos(id-aluno: string, nome: string, login: string, idade: integer, média: real)

Professores(id-prof: string, nomep: string, sal: real)

Cursos(id-curso: string, nomec: string, créditos: integer)

Salas(num-sala: integer, endereço: string, capacidade: integer)

Matriculado(id-aluno: string, id-curso: string, nota: string)

Ministra(id-prof: string, id-curso: string)

Aula(id-curso: string, num-sala: integer, horário: string)

Esquema físico

O esquema físico especifica como as relações descritas no esquema conceitual realmente vão ser gravadas nos dispositivos de armazenamento;

Neste esquema são decididas quais organizações de arquivos que vão ser utilizadas para armazenar as relações e serão criadas estruturas de dados auxiliares (**índices**) para acelerar as operações de recuperação de dados.

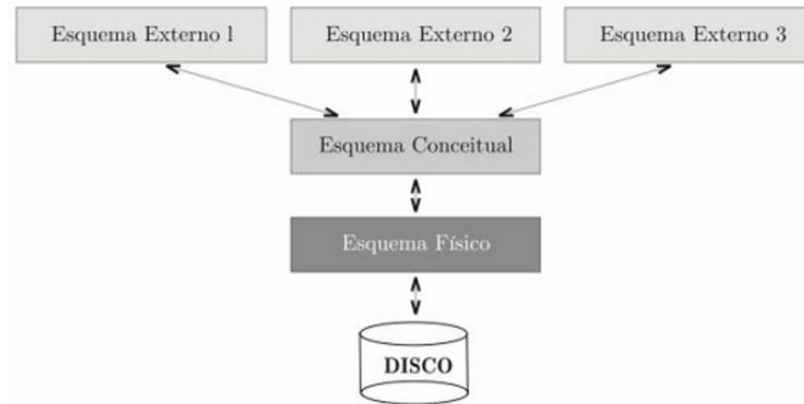
Esquema externo

Os esquemas externos, assim como os conceituais são representados em termos do modelo de dados do SGBD;

Eles permitem que o acesso aos dados seja customizado, e principalmente autorizado em níveis de usuários individuais ou grupos de usuários;

Esquema externo

Os bancos de dados possuem apenas um esquema conceitual, mas podem possuir inúmeros esquemas externos adaptados a cada grupo de usuários.



Esquema externo

Estes esquemas externos são uma coleção de uma ou mais **visões** e relações do esquema conceitual;

As visões em seu conceito são relações, no entanto os seus registros não são armazenados no SGBD, mas sim processados usando uma definição baseada nas relações armazenadas no SGBD.

Esquema externo

Considerando o banco de dados da universidade, um exemplo de visão seria a InfoCurso, onde seria permitido que o aluno consulte o professor ministrante do curso, assim como a quantidade de alunos matriculados, mesmo que estas informações não estejam armazenadas desta maneira.

InfoCurso (id-curso:string, nomep: string, matriculados: integer)

Esquema externo

Para o usuário, a consulta sobre os registros da visão é idêntica a uma relação. Embora os registros da visão não estejam armazenados, estes são processados sempre que for necessário.

A utilização de visões evita redundância no armazenamento de dados e a ocorrência de inconsistências.

Esquema externo

Um exemplo de inconsistência que poderia ser gerado se InfoCurso fosse uma relação ao invés de uma visão é se realizarmos a matrícula de um aluno na relação Matriculado sem alterar o valor do campo matriculados no registro do curso em InfoCurso.

Independência dos dados

A independência dos dados é uma das maiores vantagens da utilização de um SGBD, pois os programas são isolados das alterações no modo como o dado é estruturado e armazenado;

Esta independência é obtida através do uso dos níveis de abstração, principalmente do esquema conceitual e externo.

Independência dos dados

As relações contidas no esquema externo, relações de visões, são usualmente geradas por demanda baseadas nas relações correspondentes ao esquema conceitual;

Se o esquema conceitual for modificado é possível alterar a definição de uma relação de visão de modo que esta seja processada exatamente era antes da alteração, fornecendo as mesmas respostas do que antes. Esta propriedade é chamada de **independência lógica dos dados**.

Independência dos dados

O outro tipo de independência de dados é a **independência física dos dados**, que é obtida pela utilização dos diferentes níveis de abstração.

Como o esquema conceitual isola os usuários das alterações nos detalhes do armazenamento físico, é possível alterar o esquema físico sem que isto seja perceptível para o usuário, desde que o esquema conceitual permaneça o mesmo.

Consultas em um SGBD

Consultas em um SGBD

A facilidade de utilização do SGBD em relação ao armazenamento em arquivos dá-se principalmente pela possibilidade de realizar consultas (*query*) baseadas nos modelos de dados;

Estas consultas podem ser vistas praticamente como perguntas que seriam feitas ao SGBD;

Consultas em um SGBD

Utilizando o mesmo banco de dados da universidade, alguns exemplos de perguntas que poderiam ser feitas por um usuário são:

- Qual é o nome do aluno que possui o id-aluno 123456?
- Quantos alunos estão matriculados no curso 60208?
- Quantos alunos estão com a nota abaixo da média no curso 60208 ?

Consultas em um SGBD

Estas consultas podem ser realizadas através de uma **linguagem de consulta**.

Além das consultas, o SGBD permite aos usuários criar, modificar e excluir dados através da **linguagem de manipulação de dados** (*DML – Data Manipulation Language*), da qual a linguagem de consulta faz parte;

A linguagem SQL agrega funções de DDL e DML.

Gerenciamento de transações

Gerenciamento de transações

O acesso concorrente aos dados armazenados em um banco de dados é uma situação extremamente comum, como por exemplo em um sistema de reserva de assentos em um cinema, onde é possível reservar os assentos através dos terminais de autoatendimento, pela internet ou nos guichês de atendimento presencial. O acesso concorrente deve ser tratado com cuidado para evitar conflitos como a venda do mesmo assento para duas pessoas diferentes.

Gerenciamento de transações

Além do acesso concorrente o SGBD deve proteger os usuários dos efeitos de falhas do sistema, garantido que os dados sejam restaurados para um estados consistente quando o sistema for reiniciado após uma falha;

Utilizando o exemplo do sistema de reserva de assentos em um cinema como exemplo, se alguém solicitar a reserva de um determinado assento e o SGBD confirmar esta reserva, esta não pode ser perdida se o sistema falhar;

Gerenciamento de transações

Caso a falha ocorra entre a solicitação de reserva e a resposta do SGBD, as alterações parciais realizadas para a execução da reserva devem ser descartadas na reinicialização do sistema.

Gerenciamento de transações

As **transações** (*transaction*) podem ser definidas como uma execução qualquer de um programa de usuário em um SGBD;

Uma transação é a unidade básica de alteração reconhecida pelo SGBD, não sendo permitida a execução de transações parciais.

Execução concorrente de transações

O acesso concorrente ao dados deve ser planejado com muito cuidado, de modo que os usuários possam seguramente ignorar o fato de que existem diversos outros usuários utilizando os mesmos dados ao mesmo tempo;

Para garantir o funcionamento dos acessos concorrentes são utilizados **protocolos de bloqueio**.

Execução concorrente de transações

Existem dois tipos de protocolos de bloqueio em um SGBD:

- **Bloqueio compartilhado:** São bloqueios em um objeto mantidos por mais de uma transação ao mesmo tempo;
- **Bloqueio exclusivo:** Garante que nenhuma outra transação mantenha qualquer tipo de bloqueio em um determinado objeto.

Transações incompletas e falhas de sistema

Uma transação pode ser interrompida por diversos motivos, como por exemplo uma falha de sistema. Caso esta seja interrompida o SGBD deve assegurar que o estado do banco de dados retorne ao ponto anterior a transação assim que o sistema for reestabelecido, ou seja, retrocede as alterações parciais.

Transações incompletas e falhas de sistema

Um exemplo de reestabelecimento após falha pode ser visto em uma operação de transferência bancária onde um valor é debitado de uma conta A para ser creditado em uma conta B. Caso a falha ocorra após o débito do valor da conta A e antes do crédito na conta B, o valor que foi debitado deve ser restaurado ao valor original antes da transação.

Transações incompletas e falhas de sistema

Para controlar estas alterações o SGBD mantém um registro (*log*) de todas as ações de escrita no banco de dados;

É de suma importância que as ações sejam gravadas no log **antes** de serem aplicadas no banco de dados, esta propriedade é chamada de *Write-Ahead Log* (Gravação antecipada do Log) ou simplesmente *WAL*.

Transações incompletas e falhas de sistema

Para controlar estas alterações o SGBD mantém um registro (*log*) de todas as ações de escrita no banco de dados;

É de suma importância que as ações sejam gravadas no log **antes** de serem aplicadas no banco de dados, esta propriedade é chamada de *Write-Ahead Log* (Gravação antecipada do Log) ou simplesmente *WAL*.

OBRIGADO.

