

Banco de Dados: Introdução

Profa. Carmem Hara

Informações

- Web page:
<http://www.inf.ufpr.br/~carmem/ci218>
- e-mail: carmemhara@ufpr.br

Bibliografia

Livro Texto:

- ***Sistema de Gerenciamento de Banco de Dados***
Raghu Ramakrishnan, Johannes Gehrke
Terceira Edição, Ed. McGraw-Hill, 2007
- ***SQL for Web Nerds***, de Philip Greenspun,
<http://philip.greenspun.com/sql/>

Outros Livros

- *Sistemas de Banco de Dados*

Ramez Elmasri, Shamkant B. Navathe

Ed. Pearson Universidades; 1ª edição, 2019

- *Sistema de Banco de Dados*

Abraham Silberchatz, Henry F. Korth, S. Sudarshan

Ed. GEN LTC; 7ª edição, 2020

- *Foundations of Databases*

Serge Abiteboul, Richard Hull, Victor Vianu

Ed. Pearson; 1ª edição, 1994

Roteiro do Curso

- O modelo relacional
- Algebra relacional e SQL
- Projeto de BD
- Normalização
- Otimização de Consultas
- Controle de Concorrência e Recuperação

O que é um Sistema Gerenciador de Banco de Dados Relacional?

Sistema Gerenciador de Banco de Dados = SGBD

SGBD Relacional = SGBDR

- Uma coleção de arquivos para armazenar dados
- Um programa escrito por outras pessoas para acessar e atualizar estes arquivos

Onde os SGBDRs são utilizados?

- Backend de aplicações de banco de dados tradicionais e não tradicionais
- Backend para grandes Websites
- Backend para serviços Web

Exemplo de uma Aplicação de Banco de Dados Tradicional

Suponha que estamos construindo um sistema para armazenar informações sobre:

- alunos
- cursos
- professores
- quem cursa o que, quem ensina o que

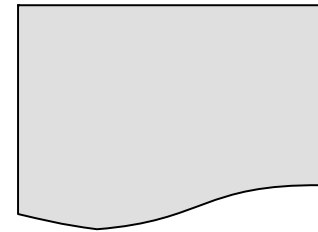
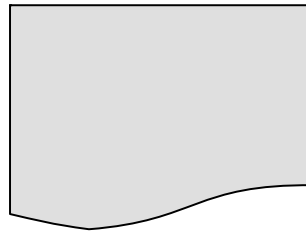
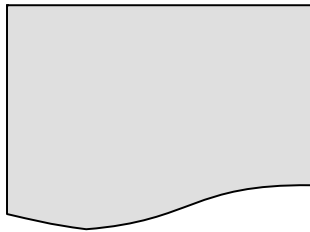
Podemos fazer isso sem um SGBD?

Claro que sim! Comece armazenando os dados em arquivos:

alunos.txt

cursos.txt

professores.txt



Agora escreva um programa que implemente as tarefas específicas

Implementando sem um SGBD...

- Matricule “Maria” em “BD”:

Escreva um programa que execute o seguinte:

Leia ‘alunos.txt’

Leia ‘cursos.txt’

Procure&atualize o registro “Maria”


Procure&atualize o registro “BD”

Grave “alunos.txt”

Grave “cursos.txt”

Problemas sem um SGBD...

- Falha no sistema:



```
Leia 'alunos.txt'  
Leia 'cursos.txt'  
Procure&atualize o registro "Maria"  
Procure&atualize o registro "BD"  
Grave "alunos.txt"  
Grave "cursos.txt"
```

FALHA !

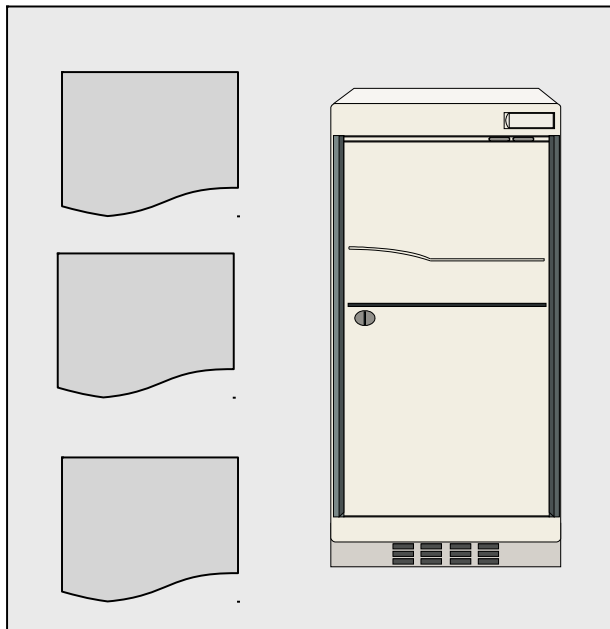
- Qual o problema ?
- Grande quantidade de dados (p. ex. 500GB)
 - Qual o problema ?
- Acesso simultâneo por diversos usuários:
 - precisamos de bloqueio: similar ao conceito em SO, porém agora os dados estão em disco.

Com um SGBD

Arquivos

Servidor de BD

Aplicações



conexão
(ODBC, JDBC)



Funcionalidade de um SGBD

O programador vê SQL, que possui dois componentes:

- Data Definition Language - DDL
- Data Manipulation Language - DML
 - linguagem de consulta

Mas o SGBD ainda possui:

- otimizador de consultas
- processador de consultas
- gerenciador de armazenamento
- gerenciador de transações (concorrência, recuperação)

Funcionalidade de um SGBD

- Arquitetura Cliente-servidor
- É um programa escrito por um grupo de pessoas
 - inicialmente ficamos impressionados com o seu desempenho
 - Depois descobrimos que ele é extremamente lento
 - Em geral as tarefas podem ser executadas mais rápido sem utilizar um SGBD
 - Mas SGBDs são *genéricos e convenientes*

Como um Programador vê um SGBD

- Começa com uma DDL para criar tabelas:

```
CREATE TABLE Aluno (  
    Nome CHAR(30)  
    CPF CHAR(9) PRIMARY KEY NOT NULL,  
    Categoria CHAR(20)  
)
```

- Utiliza uma DML para povoar as tabelas:

```
INSERT INTO Aluno  
VALUES('Maria', '123456789', 'graduacao')  
.....
```

Como um Programador vê um SGBD

- Tabelas:

Aluno:

CPF	Nome	Categoria
123-45-6789	Maria	graduacao
234-56-7890	Joao	especial.

Matricula:

CPF	NCurso
123-45-6789	C444
123-45-6789	C444
234-56-7890	C142
	...

Cursos:

NCurso	Nome	Semestre
C444	Banco de Dados	2sem/2005
C541	Sist. Operacionais	1sem/2004

- Implementado como arquivos, mas por “debaixo dos panos” pode ser bastante complexo

“*independencia de dados*” = separa a visão *lógica* da implementação *física*

Transações

- Matricule “Maria” em “Banco de Dados”:

```
BEGIN TRANSACTION;

INSERT INTO Matricula
  SELECT Aluno.CPF, Cursos.NCurso
  FROM Aluno, Cursos
  WHERE Aluno.nome = 'Maria' and
         Curso.nome = 'Banco de Dados'

-- Mais atualizações aqui ....

IF tudo-foi-realizado-com-sucesso
  THEN COMMIT;
ELSE ROLLBACK
```

Se houver falha no sistema, a transação aborta

Transações

- Uma *transação* = sequência de sentenças que ou são executadas como um todo ou falham como um todo
- Transações tem as propriedades ACID:
 - A = atomicidade
 - C = consistência
 - I = independência
 - D = durabilidade

Consultas

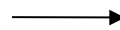
- Encontre todos os cursos de “Maria”

```
SELECT C.nome
FROM Aluno A, Matricula M, Cursos C
WHERE A.nome="Maria" and
      A.CPF = M.CPF and
      M.NCurso = C.NCurso
```

- O que acontece dentro do SGBD?
 - O processador de consulta procura a melhor maneira de responder a consulta.

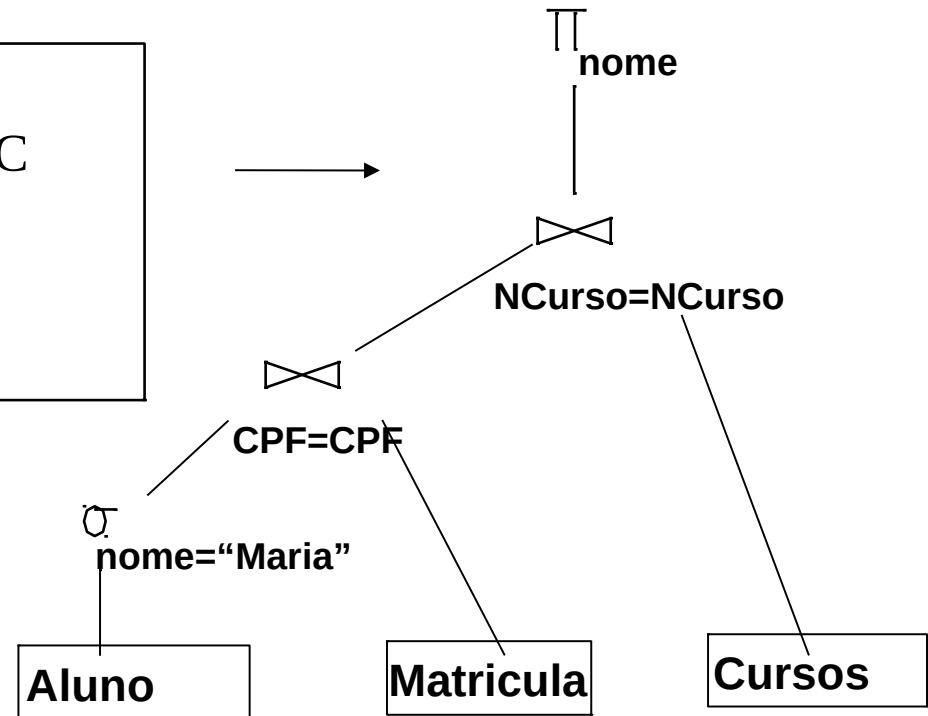
Consultas, dentro do SGBD

Consulta SQL Declarativa



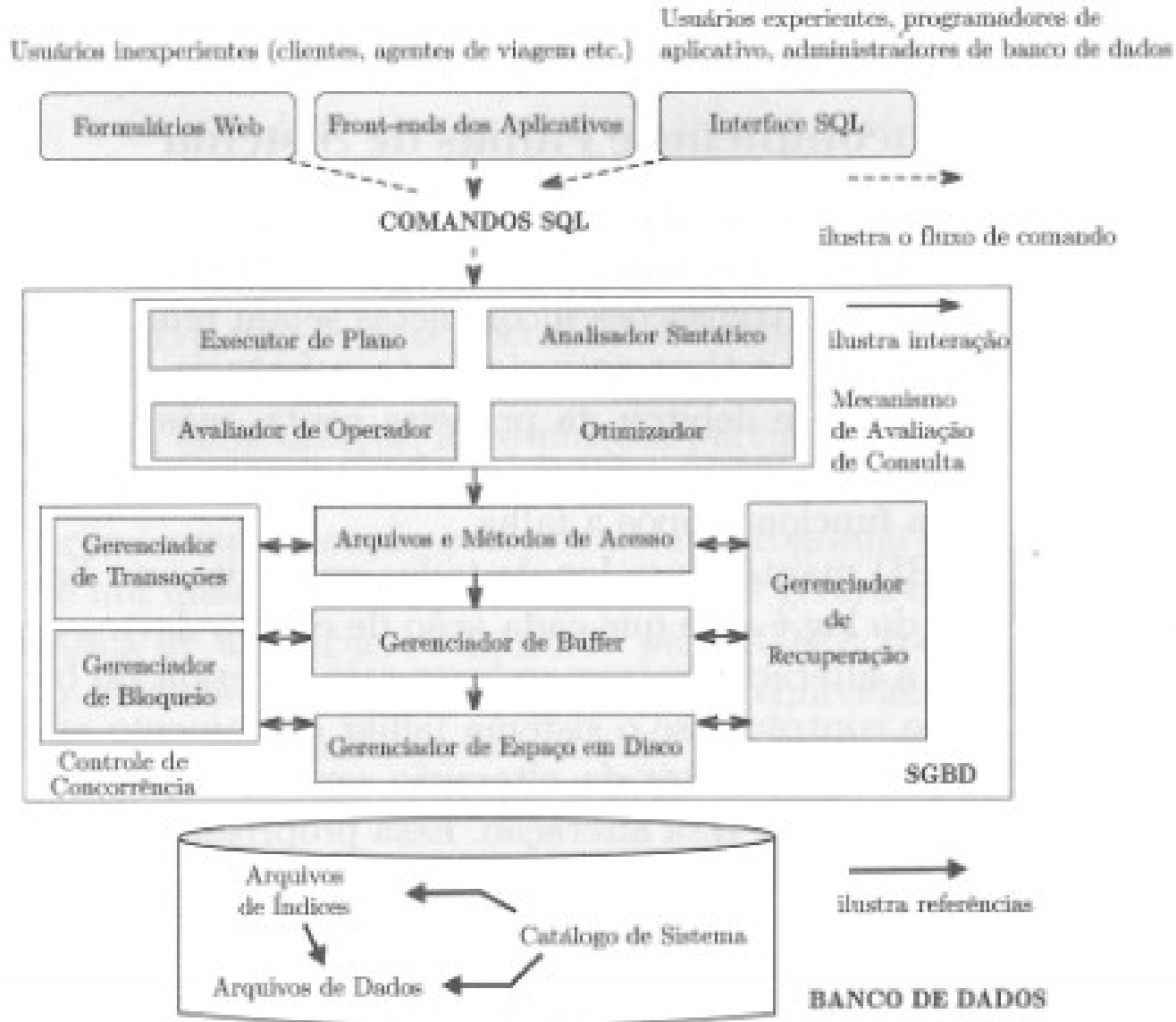
Plano de consulta imperativa

```
SELECT C.nome
FROM Aluno A, Matricula M, Cursos C
WHERE A.nome="Maria" and
      A.CPF = M.CPF and
      M.NCurso = C.NCurso
```



O otimizador de consultas escolhe o melhor plano de consulta

Arquitetura de um SGBD



Novas Tendências em Bancos de Dados

- Main memory database systems
- XML
- Integração de dados
- Stream data management
- Modelos NoSQL: chave-valor, documentos, grafos (RDF)

Estrutura

- Avaliação
 - 1 lista de exercicio
 - 2 provas

Modelo Relacional

Roteiro

- Modelo Relacional
- Algebra Relacional

Leitura:

- Capítulo 3, “Sistemas Gerenciamento de Banco de Dados”
- <http://philip.greenspun.com/sql/>

História

- 1970 – modelo introduzido por E. F. Codd
- Meados da década de 70: protótipos
 - Ingres (UC Berkeley, 73-77)
 - System R (IBM, 74-78)
- 1979: primeiro produto SQL (Oracle)
- década de 80: predominância relacional absoluta de SGBDs

SGBDs relacionais

- Comerciais:
 - Oracle
 - IBM (with DB2)
 - Microsoft (SQL Server)
 - Sybase
 - Access
- Software Livre (Unix) :
 - Postgres
 - Mysql

Modelo Relacional

- *Modelo Relacional*: baseado no modelo de relação

Esquema da Relação:

- Aluno (**id**: integer, **nome**: string[30], **cod_curso**: string[5])

nome da relação

nome do atributo

domínio do atributo

Relação ou Instância de um Esquema de Relação:

- conjunto de n-tuplas $r = \{t_1, \dots, t_m\}$, onde cada tupla t_i é uma lista ordenada de n valores $t = \langle v_1, \dots, v_n \rangle$

Aluno = $\{ \langle 123, \text{“Joao”}, \text{“CC”} \rangle, \langle 124, \text{“Maria”}, \text{“EE”} \rangle \}$

Exemplo de relação

id	nome	cod_curso
123	Joao	CC
124	Maria	EE

← Esquema da Tabela
ou relação

← Tupla ou Linha

↑
Atributo ou Coluna

- $r(R)$ é um subconjunto de $(\text{dom}(A1) \times \text{dom}(A2) \times \dots \times \text{dom}(An))$
Exemplo: $\text{tabAluno}(\text{Aluno})$ é um subconjunto de
 $\text{string}[8] \times \text{string}[30] \times \text{integer}$

Problemas

- Como definir as relações necessárias para armazenar os dados de uma aplicação?

—————→ **Projeto de banco de dados**

- Como consultar as relações resultantes do projeto?

—————→ **Linguagem de consulta: SQL**