

O Modelo Relacional

História

- 1970 - modelo introduzido por E. F. Codd
- Meados da década de 70: protótipos
 - INGRES (UC Berkeley, 73 – 77)
 - System R (IBM Research at San Jose, 74 – 78)
- 1979: primeiro produto SQL (Oracle)
- Meados da década de 80: predominância relacional absoluta no mercado de SGBDs.
- Hoje: IBM, Oracle, Sybase, Informix, Microsoft,...

Razões do Sucesso

- Sistemas pré-relacionais (hierárquicos e de rede):
 - Complexos
 - Consultas navegacionais (requeriam programação)
- Sistemas relacionais:
 - Muito mais simples
 - Consultas declarativas (*ad hoc*)
- Tecnologia para otimização de consultas
 - Viabilizou (tornou eficientes) SGBDs relacionais

Conceitos Básicos

- Noção fundamental: a relação, formalização matemática da idéia informal de tabela.
- BD relacional: conjunto de relações.
- Relação: conjunto de tuplas (linhas da tabela).
 - Não ordenada, sem linhas repetidas (é conjunto).
- Cada tupla é uma lista de valores de atributos (campos ou colunas da tabela).

Conceitos Básicos (cont.)

- Esquema de uma relação: especifica o nome da relação (tabela), mais o nome e o tipo de cada atributo (coluna).
 - Ex.: *Alunos(num_aluno: string, nome: string, login: string, idade: integer, media: real)*
 - Grau: número de colunas.
- Instância de uma relação: o conteúdo da tabela.
 - Cardinalidade: número de tuplas (linhas).

Conceitos Básicos (cont.)

- Esquema do banco de dados: conjunto de esquemas das relações mantidas no BD.
- Instância do banco de dados: conjunto das instâncias das relações mantidas no BD.

Exemplo de Instância de Relação

num_aluno	nome	login	idade	media
55666	João	joao@ime	20	8,5
55777	Maria	maria@med	21	8,2
55888	João	joao@poli	22	8,8

- Cardinalidade = 3, grau = 5.
- Sem linhas repetidas (conjunto de tuplas).
- Pode haver uma instância de relação com duas colunas idênticas?

Criando Relações em SQL

```
CREATE TABLE Alunos  
(num_aluno CHAR(8),  
 nome CHAR(30),  
 login CHAR(30),  
 idade INTEGER,  
 media REAL)
```

- Cria a relação Alunos. Note que o tipo (domínio) de cada atributo é especificado.
 - Esse tipo será checado pelo SGDB sempre que houver inserção ou modificação de tuplas.

```
CREATE TABLE Matriculas  
(num_aluno CHAR(8),  
 num_curso CHAR(6),  
 nota REAL)
```

- Outro exemplo: relação Matriculas (de alunos em cursos).

Adicionando e Removendo Tuplas

- Inserção de uma tupla:

```
INSERT INTO Alunos(num_aluno, nome,  
                  login, idade, media)  
VALUES (55777, 'Maria', 'maria@med', 21, 8.2)
```

- Remoção de todas as tuplas que satisfazem alguma condição (no caso, nome = 'João'):

```
DELETE  
FROM Alunos A  
WHERE A.nome = 'João'
```



Restrições de Integridade (RIs)

- Condições que devem ser satisfeitas por *qualquer* instância do BD.
 - São especificadas quando o esquema é definido e checadas sempre que uma relação é modificada.
 - Objetivo: fidelidade na correspondência entre os dados armazenados e seu significado no mundo real.
 - Exemplo: restrições de domínio (ou tipo).
- Uma instância de relação é válida se satisfaz todas as RIs especificadas.
 - SGBDs devem rejeitar instâncias inválidas.



Restrições de Chave

- Um conjunto de atributos S é uma superchave (*superkey*) de uma relação R se nenhuma instância válida de R pode conter duas tuplas distintas com os mesmos valores em todos os atributos de S.
- Um conjunto de atributos K é uma chave (*key*) de uma relação R se:
 1. K for uma superchave de R, e
 2. nenhum subconjunto próprio de K for superchave de R.
- O termo chave candidata é usado em vez de chave.
 - Se houver mais de uma chave para uma relação, uma delas é escolhida para ser a chave primária (*primary key*).
 - Exemplo: num_aluno é uma chave para Alunos.



Chaves Primárias e Chaves Candidatas em SQL

- Podem haver várias chaves candidatas (especificadas com UNIQUE), mas só uma chave é designada como primária.

```
CREATE TABLE Alunos
(num_aluno CHAR(8),
 nome      CHAR(30),
 login     CHAR(30),
 idade     INTEGER,
 media     REAL,
 PRIMARY KEY (num_aluno),
 UNIQUE (login) )
```



Declarações de Chaves São Restrições de Integridade!

- “Para um dado aluno em um dado curso há uma única nota.”

```
CREATE TABLE Matriculas
(num_aluno CHAR(8),
 num_curso CHAR(6),
 nota REAL,
 PRIMARY KEY (num_aluno,
 num_curso) )
```
- “Alunos podem estar matriculados num só curso e receber uma só nota pelo curso. Além disso, dois alunos não podem ter a mesma nota no mesmo curso.”

```
CREATE TABLE Matriculas
(num_aluno CHAR(8),
 num_curso CHAR(6),
 nota REAL,
 PRIMARY KEY (num_aluno),
 UNIQUE (num_curso, nota) )
```

Restrições de Integridade Referencial

- Chave estrangeira (*foreign key*): conjunto de atributos de uma relação que referencia uma tupla em outra relação. (Precisa corresponder a uma chave primária da segunda relação.) É uma espécie de “ponteiro lógico”.
 - Sinônimo: atributos transpostos ou atributos de ligação.
- Exemplo: num_aluno é uma chave estrangeira referenciando a relação Alunos:
Matriculas(num_aluno: string, num_curso: string, nota: real)

Restrições de Integridade Referencial

- Integridade referencial significa que toda tupla referenciada por uma chave estrangeira precisa existir. Em outras palavras: o SGBD garante que as referências são sempre válidas.
 - Você conhece um ambiente sem integridade referencial?
 - O Web! Quem ainda não tentou seguir um *link* inválido?
- Declarações de chave estrangeira são restrições de integridade referencial.

Chaves Estrangeiras em SQL

- “Somente alunos que aparecem na relação Alunos podem se matricular em cursos.”

```
CREATE TABLE Matriculas
(num_aluno CHAR(8), num_curso CHAR(6), nota REAL,
PRIMARY KEY (num_aluno, num_curso),
FOREIGN KEY (num_aluno) REFERENCES Alunos)
```

Matriculas			Alunos				
num_aluno	num_curso	nota	num_aluno	nome	login	idade	media
555666	M-007	7.5	55666	João	joao@ime	20	8,5
555666	C-202	8.5	55777	Maria	maria@med	21	8,2
555777	A-303	9.0	55888	João	joao@poli	22	8,8
555888	C-303	8,8					

Garantindo a Integridade Referencial

Considere as relações Alunos e Matriculas.

- num_aluno em Matricula é uma chave estrangeira referenciando Alunos.
- O que o SGBD deve fazer se uma tupla com um aluno não existente é inserida em Matriculas? (Rejeitar a inserção!)
- E se uma tupla for removida de Alunos? Várias alternativas:
 - Remover também as tuplas de Matriculas que a referenciam.
 - Rejeitar a remoção de uma tupla referida por outras.
 - Colocar um valor *default* no num_aluno das tuplas que a referenciam.
 - Colocar *null* no num_aluno das tuplas que a referenciam.

De Onde Vem as RIs?

- Restrições de integridade são baseadas na semântica da empresa ou negócio descrito pelas relações no BD.
- Podemos checar uma instância de BD e ver se uma RI é violada ou não. Mas examinando uma instância NUNCA podemos concluir uma RI deve ser imposta ou não.
 - Uma RI diz respeito a *todas as possíveis* instâncias!
 - Por exemplo, examinando a instância de Alunos vemos que o nome não é chave. Mas o fato de num_aluno ser chave é um dado da realidade modelada.
- Restrições de chave e de integridade referencial são as mais comuns, mas RIs mais gerais são também suportadas.

Linguagens de Consulta Relacionais

- Uma grande vantagem do modelo relacional: suporte para consultas simples e poderosas.
- Consultas (*queries*) são declarativas: você especifica o que deseja, e não como o sistema deve processar sua consulta.
 - Isto funciona graças à semântica precisa das consultas relacionais.
 - O otimizador de consultas (query optimizer) pode rearranjar a consulta e mesmo assim garantir que o resultado é o desejado.

A Linguagem de Consulta SQL

- Linguagem de consulta relacional mais usada. A padronização atual é o SQL-92.
- Para obter os dados de todos os alunos com 21 anos ou mais, podemos escrever:

```
SELECT *  
FROM Alunos A  
WHERE A.idade >= 21
```

O resultado é a tabela:

num_aluno	nome	login	idade	media
55777	Maria	maria@med	21	8,2
55888	João	joao@poli	22	8,8

- Para obter só os nomes e logins, troque a 1ª linha:
SELECT A.nome, A.login

Consultando Múltiplas Relações

- Qual o resultado da seguinte consulta?

```
SELECT A.nome, M.num_curso
FROM Alunos A, Matriculas M
WHERE A.num_aluno = M.num_aluno
      AND M.nota > 8,7
```

Dada esta instância de Matriculas:

num_aluno	num_curso	nota
555666	M-007	7.5
555666	C-202	8.5
555777	A-303	9.0
555888	C-303	8.8

O resultado é:

A.nome	M.num_curso
Maria	A-303
João	C-303

Semântica de uma Consulta

- Um método de avaliação *conceitual* para a consulta anterior:
 - Trate a cláusula FROM: compute o produto cartesiano de Alunos e Matriculas.
 - Trate a cláusula WHERE: verifique as condições especificadas e descarte as tuplas que não as satisfazem.
 - Trate a cláusula SELECT: remova colunas não desejadas.
- Lembre-se, isto é *conceitual*. A avaliação real será muito mais eficiente, mas produzirá os mesmos resultados.

Produto Cartesiano das Instâncias de Alunos e Matrículas

A.num_aluno	A.nome	A.login	A.idade	A.media	M.num_aluno	M.num_curso	M.nota
555666	João	joão@ime	20	8.5	555666	M-007	7.5
555666	João	joão@ime	20	8.5	555666	C-202	8.5
555666	João	joão@ime	20	8.5	555777	A-303	9.0
555666	João	joão@ime	20	8.5	555888	C-303	8.8
555777	Maria	maria@med	21	8.2	555666	M-007	7.5
555777	Maria	maria@med	21	8.2	555666	C-202	8.5
555777	Maria	maria@med	21	8.2	555777	A-303	9.0
555777	Maria	maria@med	21	8.2	555888	C-303	8.8
555888	João	joao@poli	22	8.8	555666	M-007	7.5
555888	João	joao@poli	22	8.8	555666	C-202	8.5
555888	João	joao@poli	22	8.8	555777	A-303	9.0
555888	João	joao@poli	22	8.8	555888	C-303	8.8

O Modelo Relacional: Resumo

- Representação tabular dos dados.
- Restrições de integridade podem ser especificadas de acordo com a semântica da aplicação. O SGBD impede violações.
 - Dois tipos importantes de RIs: chaves e chaves estrangeiras.
 - Além disso, *sempre* temos restrições de domínio.
- Poderosa linguagem declarativa para consultas (SQL).