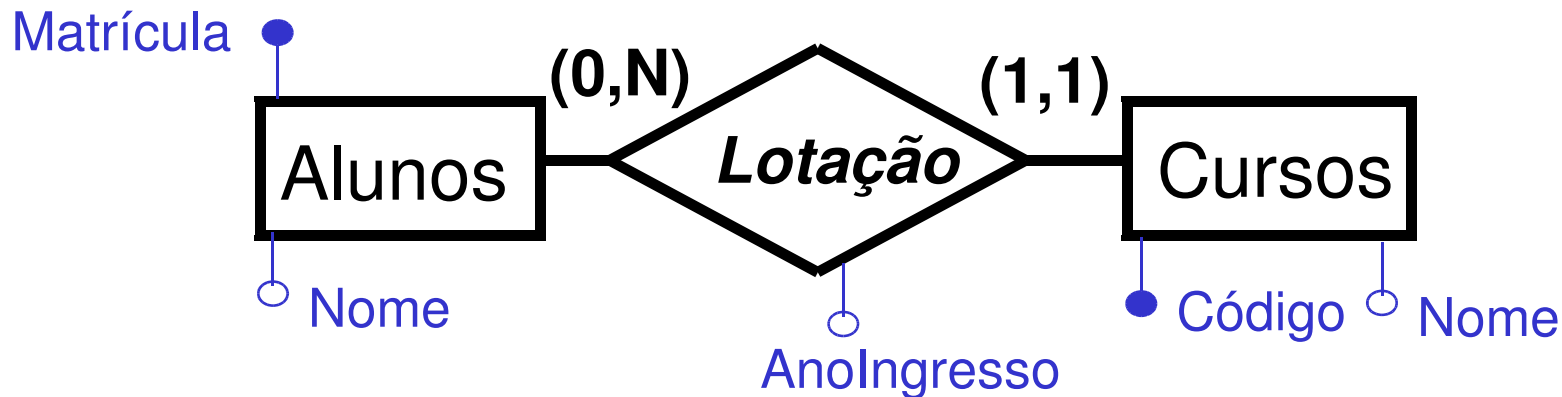


Modelo de Dados

- Modelo para organização dos dados de um BD
 - define um conjunto de conceitos para a representação de dados
 - exemplos: entidade, tabela, atributo, ...
 - existem modelos para diferentes níveis de abstração de representação de dados
 - modelos conceituais
 - modelos lógicos
 - modelos físicos
 - organização dos arquivos de dados em disco (organização seqüencial, uso de índices *hashing* ou *B-trees*, ...)
 - não são manipulados por usuários ou aplicações que acessam o BD
 - » decisões de implementação de cada SGBD

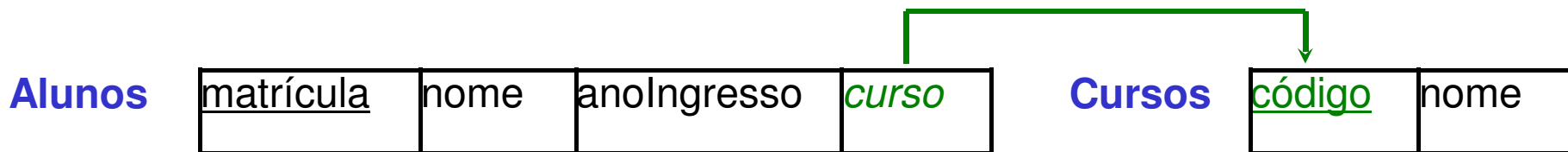
Modelos Conceituais

- Representação com alto nível de abstração
 - modela de forma mais natural os fatos do mundo real, suas propriedades e seus relacionamentos
 - independente de BD
 - preocupação com a semântica da aplicação
 - exemplo: modelo entidade-relacionamento



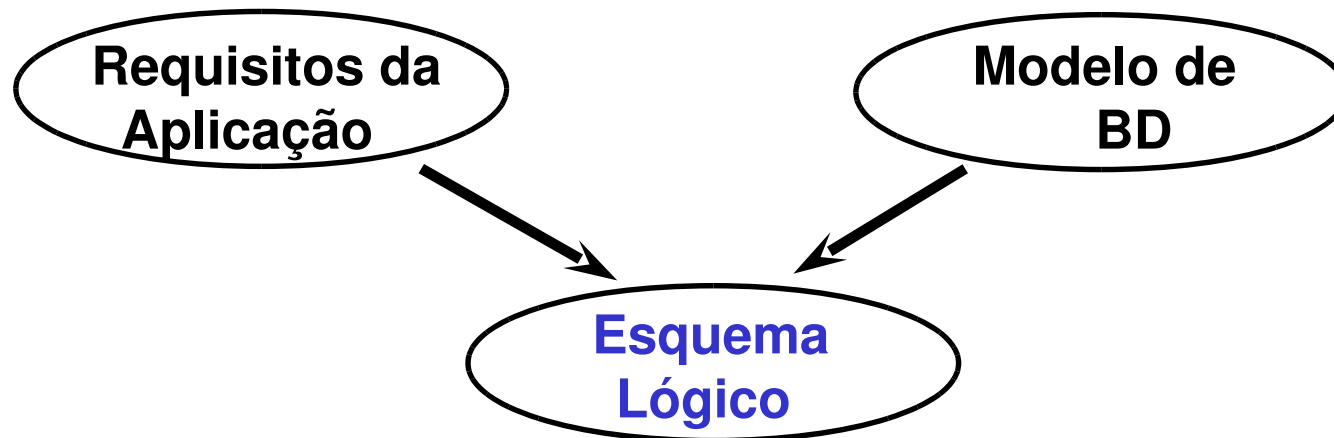
Modelos Lógicos

- Representam os dados em alguma estrutura (lógica) de armazenamento de dados
 - também chamados de modelos de BD
 - dependente de BD
 - exemplos
 - modelo relacional (tabelas)
 - modelos hierárquico e XML (árvore)
 - modelo orientado a objetos (classes - objetos complexos)



Modelos de BD (Lógicos)

- Suporte a métodos de acesso
 - especificação dos conceitos do modelo (DDL)
 - dados, seus domínios, relacionamentos e restrições
 - manipulação de conceitos modelados (DML)
- Esquema (lógico) de BD
 - resultado da especificação dos dados de um domínio de aplicação em um modelo de BD



Modelos de BD

- 1^a geração: Modelos pré-relacionais
 - modelos hierárquico e de rede
- 2^a geração: Modelo relacional
- 3^a geração: Modelos pós-relacionais
 - modelos orientado a objetos, objeto-relacional, temporal, geográfico, ...

Modelos Pré-Relacionais

- Modelos com várias limitações
 - não representam adequadamente relacionamentos do mundo real
 - exemplo: hierarquias (1-1 ou 1-N)
 - problemas de performance
 - exemplo: varredura em estruturas em grafo (rede)
 - inexistência de uma linguagem de consulta declarativa
 - consultas exigem programação pela aplicação
 - manipulam um registro por vez
 - baixa performance de acesso

Modelos Pós-Relacionais

- Novos modelos de dados para atender requisitos de algumas categorias de aplicações
 - BDOO
 - dados com representação complexa
 - BDT
 - suporte a representação da história dos dados
 - ...
- Exemplos de áreas de aplicação
 - engenharia, arquitetura, cadastro urbano, ...

Modelo Relacional

- Definido em 1970 (E. Codd – IBM/Califórnia)
- Modelo com uma sólida base formal
 - teoria dos conjuntos
- Modelo simples
 - estruturas tabulares
 - poucos conceitos
- Linguagens declarativas para a manipulação de dados
 - álgebra relacional e cálculo relacional (formais)
 - SQL (comercial)

Modelo Relacional - Características

- Organização dos dados
 - conceitos do modelo
 - atributo, relação, chave, ...
- Integridade
 - restrições básicas para dados e relacionamentos
- Manipulação
 - linguagens formais e SQL

Modelo Relacional - Organização

- O modelo apresenta cinco conceitos
 - *domínio*
 - *atributo*
 - *tupla*
 - *relação*
 - *chave*

Domínio

- Conjunto de valores permitidos para um dado
- Exemplos
 - *inteiro*, *string* (domínios básicos)
 - *data*, *hora* (domínios compostos)
 - $[0, 120]$, (*'M'*, *'F'*) (domínios definidos)
- Para um domínio existem operações válidas
 - *inteiro* (somar, dividir, i_1 maior que i_2 , ...)
 - *data* (extrair dia, extrair mês, d_1 anterior a d_2 , ...)
- Definição de domínios de dados
 - DDL (indicação de tipos de dados [+RIs])

Atributo

- Um **item de dado** do BD
- Possui um nome e um domínio
- Exemplos
 - nome: *string*
 - idade: [0,120]

Tupla

- Um conjunto de pares (*atributo, valor*)
 - define uma ocorrência de um fato do mundo real ou de um relacionamento entre fatos
- Valor de um atributo
 - definido no momento da criação de uma tupla
 - deve ser
 - compatível com o domínio OU *NULL* (valor inexistente ou indeterminado)
 - atômico (indivisível: não-estruturado e monovalorado)
- Exemplo
 - tupla de aluno: {(nome, 'João'), (idade, 34), (matrícula, 03167034), ...}

Relação

- Composto por um *cabeçalho* e um *corpo*
- Cabeçalho
 - número fixo de atributos (*grau da relação*)
 - atributos não-ambíguos
- Corpo
 - número variável de tuplas (*cardinalidade da relação*)

Relação

- Definição formal
 - subconjunto do produto cartesiano de todos os domínios $D_1 \times D_2 \times \dots \times D_n$ de atributos
- Logo, relação é um conjunto
 - na prática, uma relação em um BD é chamada **tabela**
 - uma tabela admite uma **coleção** de tuplas

Chave

- Conjunto de um ou mais atributos de uma relação
- Tipos de chaves
 - chave primária (pk)
 - atributo(s) cujo (conjunto de) valor(es) **identifica(m) unicamente** uma tupla em uma relação
 - notação para a pk de uma relação R : $pk(R)$
 - conceitos associados
 - chaves candidatas e chaves alternativas
 - exemplos
 - alunos: **matrícula**
 - cidades: **(nome, estado)**

Chave

- Tipos de chaves
 - chave estrangeira (*fk*)
 - atributo(s) de uma relação R_1 que estabelece(m) uma **equivalência de valor** com a chave primária de uma relação R_2 ($fk(R_1) \rightarrow pk(R_2)$)
 - notação para uma *fk* de uma relação R : $fk(R)$
 - **domínio($fk(R_1)$) = domínio($pk(R_2)$)**
 - R_1 e R_2 podem ser a mesma relação
 - exemplos
 - alunos: **curso** (referência a um código de curso)
 - cursos: **código**

Modelo Relacional - Integridade

- Consistências básicas
 - identificação para os dados
 - garantia de acesso a todos os dados sem ambigüidade
 - Regra de Integridade de Entidade (**RIE**)
 - garantia de relacionamentos válidos
 - Regra de Integridade Referencial (**RIR**)

Regra de Integridade de Entidade

- Dada uma tupla t_k de uma relação R
 - condição 1: $\neg \exists a_i \in pk(t_k) \mid a_i \text{ is NULL}$
 - condição 2: $\neg \exists t_j \in R \mid t_j \neq t_k \wedge pk(t_j) = pk_i(t_k)$

Regra de Integridade Referencial

- Dadas duas relações R_1 e R_2 , uma chave estrangeira $fk_i(R_1) \rightarrow pk(R_2)$ e uma tupla $t_k \in R_1$, então
 - se $fk_i(t_k)$ is NULL $\Rightarrow fk_i(R_1) \not\subset pk(R_2)$
 - se $fk_i(t_k)$ is not NULL $\Rightarrow \exists t_j \in R_2 \mid pk(t_j) = fk_i(t_k)$

Implicações da RIE

- Dada uma tupla $t_k \in R$
 - se ocorre **inclusão (I)** ou **atualização (A)** de t_k
 - se $\{\exists a_i \in pk(t_k) \mid a_i \text{ is NULL}\} \vee$
 $\{\exists t_j \in R \mid t_k \neq t_j \wedge pk(t_k) = pk(t_j)\}$
então IMPEDIMENTO
senão EFETIVAÇÃO;
 - se ocorre **exclusão (E)** de t_k
 - nenhuma consistência a ser feita

Implicações da RIR

- Dadas duas relações (R_1 e R_2) e uma chave estrangeira $fk_i(R_1) \rightarrow pk(R_2)$, três ações podem ser tomadas

– IMPEDIMENTO

- a operação de atualização não é efetivada

– CASCATA

- se ocorre **E** de uma tupla $t_j \in R_2$, então $\forall t_k \in R_1 \mid fk_i(t_k) = pk(t_j) \Rightarrow E(t_k)$
- se ocorre **A** da $pk(t_j)$ de uma tupla $t_j \in R_2$, então $\forall fk_i(t_k) \in R_1 \mid$ se $fk_i(t_k) = \text{OLD}(pk(t_j)) \Rightarrow fk_i(t_k) \leftarrow \text{NEW}(pk(t_j))$

– ANULAÇÃO

- se ocorre **E** ou **A** de uma tupla $t_j \in R_2$, então $\forall t_k \in R_1 \mid$
 $fk_i(t_k) \rightarrow pk(t_j) \Rightarrow fk_i(t_k) \text{ is NULL}$

Implicações da RIR

- Situação 1: operações sobre R_1 (onde está fk_i)
 - I ou A de uma tupla t_k
 - se $fk_i(t_k)$ is NULL então
 - se $fk_i(R_1) \subset pk(R_1)$
 - então IMPEDIMENTO
 - senão EFETIVAÇÃO
 - senão
 - se $\neg \exists t_j \in R_2 / fk_i(t_k) \rightarrow pk(t_j)$ então
 - se $fk_i(R_1) \subset pk(R_1)$
 - então IMPEDIMENTO
 - senão IMPEDIMENTO ou ANULAÇÃO
 - senão EFETIVAÇÃO
 - E de uma tupla t_k
 - nenhuma consistência a ser feita

Implicações da RIR

- Situação 2: operações sobre R_2 (onde está pk)
 - E de uma tupla t_j ou A da $pk(t_j)$
 - $\forall t_k \in R_1 \mid fk_i(t_k) \rightarrow pk(t_j)$ então
 - se $fk_i(R_1) \not\subseteq pk(R_1)$
 - então IMPEDIMENTO ou
 - CASCATA ou
 - ANULAÇÃO
 - senão IMPEDIMENTO ou CASCATA
 - I de uma tupla t_j
 - nenhuma consistência a ser feita