
MODELO LÓGICO DE DADOS

MODELO LÓGICO DE DADOS

- ❑ Segunda etapa do projeto do Banco de Dados para um SI, cujo objetivo é obter uma descrição de como implementar o BD, dependente do Hardware e do Software escolhidos.

 - ❑ São considerados nesta etapa os seguintes elementos:
 - Qual o tipo do SGBD escolhido?
 - Quais os requisitos de desempenho?
 - Quais os requisitos de segurança?
 - Como os dados serão manipulados?
 - Como otimizar as consultas?
-

MODELO LÓGICO DE DADOS

- ❑ Existem vários tipos de SGBD's no mercado que implementam diferentes modelos, tais como o Modelo Relacional e o Modelo Orientado a Objetos.
 - ❑ Nesta etapa do curso daremos ênfase ao Modelo Relacional.
-

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

- ❑ Proposto por Edgar F. Codd nos anos 70 dando origem aos SGBDR's
 - ❑ Princípio: as linhas de uma tabela representam os fatos descritos nas entidades
 - ❑ Uma linha de uma tabela é chamada de *tupla*
 - ❑ O tipo de dados que descreve cada coluna é chamado de *domínio*
 - ❑ Durante a especificação do domínio, deve-se especificar o tipo, o tamanho e o domínio dos atributos, sempre de acordo com o SGBD escolhido.
-

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

Tabela DEPARTAMENTO		
Nome	@Número	RG Gerente
Contabilidade	1	101010
Engenharia	2	303030
Marketing	3	202020

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

- ❑ O ideal é a criação de um Dicionário de Dados, que deve ser mantido assim como a Modelagem Conceitual.

Tabela FUNCIONÁRIO			
Coluna	Tipo	Tamanho	Domínio
RG	Int	8	0 a 999999999
Nome	Varchar	30	a-z, A-Z
Salário	Decimal	7,2	>= 200,00 e <99999,99
Data Nascimento	DateTime	10	> 01/01/1900 e < 01/01/2200

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

- ❑ Tipos de dados mais utilizados no SGBD SQL SERVER: Char(n), Varchar(n), DateTime, Decimal(n,m), Real, Int, Text
 - ❑ Toda tabela deve possuir uma **chave primária** (*Primary Key – PK*), que deve ser mínima
 - ❑ As **Associações/Relacionamentos** entre duas entidades do Modelo Conceitual são implementadas no Modelo Relacional com o uso de **chaves estrangeiras** (*Foreign Key – FK*)
-

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

Tabela DEPARTAMENTO		
Nome	@Número	RG Gerente
Contabilidade	1	101010
Engenharia	2	303030
Marketing	3	202020

FK do relacionamento
"Gerencia"

Do auto relacionamento
"Supervisiona"

Do relacionamento
"Lotação"

Tabela EMPREGADO				
Nome	@RG	Depto	RG Supervisor	Salário
José	101010	1	NULL	2.000,00
Maria	202020	2	101010	800,00
Pedro	303030	2	101010	1.500,00
João	404040	3	202020	350,00

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

- ❑ As FKs não precisam ter o mesmo nome das chaves primárias, embora seja recomendável
 - ❑ As FKs precisam ser do mesmo tipo das PKs que referenciam
 - ❑ A abordagem relacional gera o que se chama de **regras de integridade do Modelo Relacional:**
 - 1. Integridade de domínio dos atributos**
Garantido pelo SGBD com base nos tipos de dados e regras (rules) definidas
-

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

2. Integridade de chave

A PK não pode conter um valor nulo (NULL)

3. Integridade Referencial

Os valores possíveis das FKs devem pertencer ao conjunto das PKs referenciadas e existentes atualmente no BD ou possivelmente serem NULL. Não pode existir na FK um valor que não exista na tabela na qual ela é PK

- A Integridade referencial pode criar problemas durante a manipulação dos dados (atualização e remoção)
Ex: Como remover o Departamento de Código 1?
Ex: Como atualizar o Código do Departamento de 2 para 4?
-

MODELO LÓGICO DE DADOS

Modelo Relacional de BD

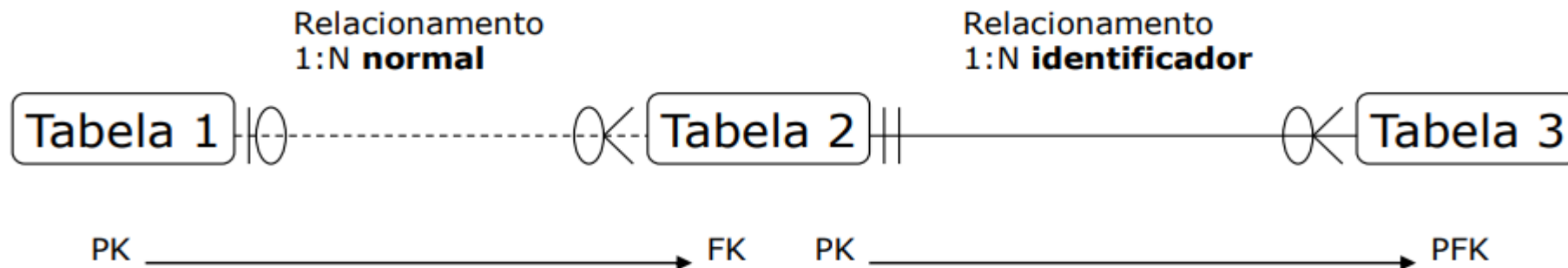
- Soluções para problemas de atualização da PK e remoção de linhas referenciadas:
 1. **Bloqueia**: não atualiza/remove enquanto as linhas relacionadas existirem (RESTRICT);
 2. **Propaga**: Remove/atualiza as linhas relacionadas (CASCADE DELETE/UPDATE)
 3. **Set Null**: Coloca NULL nas linhas da FK e remove/altera a linha da PK
 4. **Set Default**: Coloca o valor Default definido para a FK e remove/altera a linha PK

 - Os SGBDRs implementam essas soluções, porém é o analista, com base na realidade que está modelando, quem deve definir qual é a melhor solução e documentá-la no Dicionário de Dados.
-

MODELO LÓGICO DE DADOS

Regras de Transformação

- ❑ Durante a aplicação das regras, deve-se levar em consideração o desempenho, a manutenção e a perda de espaço de armazenamento, nesta ordem
- ❑ Para representarmos o Diagrama Físico Relacional, somente necessitamos dos seguintes símbolos:

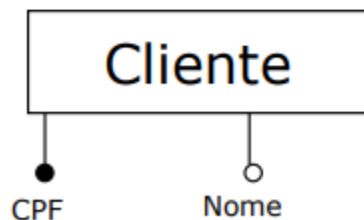


Modelo Físico dos Dados

Regras de Transformação

Regra 1:

- ❑ Cada entidade forte vira uma tabela, cada atributo simples vira uma coluna com um tipo de dado e domínio a ser definido. É necessário também identificar a chave primária(PK) e representá-la com a palavra PK logo após o nome do atributo



Cliente			
CPF	int		PK
Nome	varchar(50)		

MODELO LÓGICO DE DADOS

Regras de Transformação

Regra 2:

- Toda associação/relacionamento N:N vira uma tabela contendo dois relacionamentos identificadores. As chaves primárias das tabelas relacionadas viram PK e FK na nova tabela gerada. Os atributos próprios do relacionamento, se existirem, ficam na tabela gerada. As FKs não podem aceitar valores NULL, e são representadas com a palavra FK logo depois do nome do atributo
-

MODELO LÓGICO DE DADOS

Regras de Transformação

Regra 2:

- Exercícios: aplique a regra 2 aos seguintes casos:
 1. Pedido (NroPedido, ValorTotal, Data)
Produto (CódBarras, Nome, PreçoVenda)
Venda[NroPedido, CódBarras](Quantidade, PreçoPago)
 2. Médico (CRM, Nome, Endereço)
Paciente (Código, Nome, Endereço)
Consulta[CRM, Código](Data, ValorPago, Diagnóstico)
 3. Filme (CódFilme, Nome)
Ator(CódAtor, Nome)
Elenco[CódFilme, CódAtor]
-

MODELO LÓGICO DE DADOS

Regras de Transformação

Regra 3:

- ❑ Em Associações/Relacionamentos 1:N, acrescente o atributo chave do lado 1 na tabela do lado N como FK contendo um relacionamento 1:N normal. Os atributos dos relacionamentos serão acrescentados na tabela do lado N.

 - ❑ Exercício: aplique a regra 3 ao seguinte caso:
 - 1. Departamento (CódDepto, Nome)
 - Empregado (CPF, Nome)
 - Lotação*[CódDepto, CPF](DataInicio)
-

MODELO LÓGICO DE DADOS

Regras de Transformação

Regra 4:

- ❑ Em Associações/Relacionamentos 1:1, adicione a PK de uma das tabelas como FK na outra tabela. Caso existam, os atributos do relacionamento serão colocados na tabela que possuir a FK. Dica: tente colocar a FK na tabela que possua menos linhas

 - ❑ Exercício: aplique a regra 4 ao seguinte caso:
 1. Professor (CódProf, Nome, Endereco)
Curso (CódCurso, Descricao)
Coordena[CódProf, CódCurso](Gratificacao)

 2. Estado (Sigla, Nome)
Governador (CodGov, Nome)
Governa[Sigla, CodGov](DataInicio)
-

MODELO LÓGICO DE DADOS

Regra 5:

- ❑ Em Associações/Relacionamentos com entidade fraca, a mesma vira uma nova tabela. Adicione a PK da tabela do lado 1 como PFK na tabela do lado N. Escolha um ou mais atributos da tabela do lado N para compor a PK. Represente o relacionamento como 1:N identificador. Os atributos do relacionamento, caso existam, serão acrescentados na tabela do lado N. Inclua a opção "ON DELETE CASCADE" para o novo relacionamento

 - ❑ Exercício: aplique a regra 5 ao seguinte caso:
 - 1. Funcionário (CódFunc, Nome, Endereco)
 - Dependente (CódDep, Nome, Idade)
 - Depende*[CódFunc, CódDep]
-

MODELO LÓGICO DE DADOS

Regras de Transformação

Regra 6:

- ❑ Em Auto Relacionamentos, aplique as regras para relacionamentos 1:N, N:N e 1:1
 - ❑ Exercícios: aplique a regra 6 aos seguintes casos:
 1. Empregado (CódEmp, Nome)
Supervisiona[CódEmp, CódEmp]
 2. Disciplina (CodDisciplina, Nome, CargaHoraria)
Pré-requisito[CodDisciplina, CodDisciplina]
-

MODELO LÓGICO DE DADOS

Regras de Transformação

Regra 7:

- ❑ Em Heranças/Especializações gere uma tabela para a entidade pai e para cada entidade filha. As PKs das entidades "pais" serão PFKs nas entidades filhas
 - ❑ Exercício: aplique a regra 7 aos seguintes casos:
 1. Pessoa, PessoaFísica e PessoaJurídica
 2. Veículo, Automóvel e Moto
-

