

Unidade 3a: NORMALIZAÇÃO

NORMALIZAÇÃO

Alguns problemas sérios podem surgir a partir da criação das relações, mesmo quando todas as regras de conversão são utilizadas convenientemente;

Estes problemas são denominados ANOMALIAS DE ATUALIZAÇÃO e foram identificados por Codd para justificar a necessidade do processo de NORMALIZAÇÃO;

Exemplo: **Relação EMPREGADO_DEPTO**

CPF	Nome	Endereço	Num_Depto	Nome_Dept
098765432	Maria	Rua Y, 234	2	Informática
111111111	Carlos	Rua Z, 567	3	Financeiro
222222222	José	Rua K, 678	2	Informática
123456789	João	Rua X, 123	5	Pessoal

ANOMALIA DE INSERÇÃO:

Para inserir um novo empregado na relação anterior devemos incluir todos os atributos do departamento para o qual o empregado trabalha ou, se não os conhecemos, inserimos nulos. Todas estas informações devem ser compatíveis com a tabela DEPARTAMENTO.

É difícil inserir um novo departamento que contenha empregados que já participem da relação EMPREGADO_DEPTO

ANOMALIA DE DELEÇÃO:

Se apagarmos uma tupla que representa o ultimo empregado que trabalhava em um determinado departamento, teremos toda a informação sobre o departamento perdida.

ANOMALIA DE ALTERAÇÃO:

Se trocarmos o valor de um dos atributos de um departamento em particular, digamos o nome do departamento 2 devemos atualizar todas as tuplas de empregados deste departamento

DEPENDÊNCIA FUNCIONAL

Determina uma restrição entre dois conjuntos de atributos de um BD Relacional.

Denotado pelo símbolo \rightarrow , onde $X \rightarrow Y$ significa que o atributo X determina funcionalmente o atributo Y ou, ainda, o atributo Y é dependente funcional do atributo X.

Ex. 1: $\text{MATRICULA} \rightarrow \text{NOME_FUNCIONARIO}$

–MATRICULA determina funcionalmente NOME_FUNCIONARIO;

–NOME_FUNCIONARIO é dependente funcional de MATRICULA

–Informalmente : Conhecido o valor de uma MATRICULA eu consigo determinar qual o NOME_FUNCIONARIO (único) com o qual esta MATRICULA está relacionada.

Ex. 2:

MATRICULA, NUMERO_PROJETO \rightarrow NUM_HORAS_TRABALHADAS

- MATRICULA E NUMERO_PROJETO determinam funcionalmente NUM_HORAS_TRABALHADAS;
- NUM_HORAS_TRABALHADAS é dependente funcional de MATRICULA E NUMERO_PROJETO.

– Informalmente :

Conhecido o valor de uma MATRICULA E NUMERO_PROJETO, eu consigo determinar qual o NUM_HORAS_TRABALHADAS com o qual esta MATRICULA está relacionada.

FORMAS NORMAIS

Uma relação é dita estar numa determinada forma normal se satisfaz uma série de condições definidas para cada forma normal.

- **1ª Forma Normal (1FN)** - Uma relação está na 1FN se e somente se todos os domínios de atributo contêm somente valores escalares. Ou seja, todos os atributos são atômicos. Em cada posição de linha e coluna em uma relação só pode haver exatamente um valor, nunca uma coleção de valores.
- **2ª Forma Normal (2FN)** - Uma relação está na 2FN se e somente se, estiver na 1NF e todos os seus atributos que não sejam chave forem dependentes funcionais de **TODA** a chave primária.
- **3ª Forma Normal (3FN)** - Uma relação está na 3FN, se e somente se, estiver na 2FN e os atributos não chave não forem dependentes transitivos da chave primária

EXEMPLO: seja a tabela abaixo, não normalizada:

CodProj	Nome	Descr	CodEmp	NomeEmp	Categoria	Salário	Dt_I	Dt_F
1	A	XXX	E1	CARLOS	EA	100	10/10/01	10/11/01
			E2	MILTON	EA	100	10/10/01	10/11/01
			E3	JORGE	EB	50	20/10/01	10/11/01
2	B	YYY	E1	CARLOS	EA	100	10/05/01	10/09/01

Relação não normalizada:

Projeto (CodProj, Nome, Descr, (CodEmp, NomeEmp, Categoria, Salário, Dt_I, Dt_F))

Primeira Forma Normal (1FN)

É a transformação do esquema de tabela não normalizada em um esquema relacional na primeira forma normal (1FN). Uma tabela encontra-se na 1FN quando não contém tabelas aninhadas, ou seja, todos os atributos são atômicos (um só valor).

Passo 1: Criar a tabela na 1FN de forma a conter apenas colunas com valores atômicos. A chave da tabela na 1FN é idêntica a chave da tabela não normalizada.

Passo 2: Para cada tabela aninhada é criada uma tabela na 1FN.

Passo 3: Definir as chaves primárias das tabelas na 1FN correspondentes às tabelas aninhadas.

Projeto (CodProj, Nome, Descr)

ProjEmp (CodEmp, CodProj, NomeEmp, Categoria, Salário, Dt_I, Dt_F)

Segunda Forma Normal (2FN)

Uma tabela está na 2FN quando está na 1FN e cada coluna não chave depende da chave primária completa. Uma tabela que não está na 2FN possui dependências funcionais parciais, ou seja, contém colunas não chave que dependem de uma parte da chave primária.

Em uma tabela relacional, uma coluna C2 depende funcionalmente de uma coluna C1 quando, em todos os valores da tabelas, para cada valor de C1 aparece o mesmo valor de C2.

Categoria	Salário
1	20
2	30
1	20
2	30
3	40

Os valores da coluna SALÁRIO dependem funcionalmente dos valores da coluna CATEGORIA.

A coluna CATEGORIA determina a coluna SALARIO.

CATEGORIA → SALARIO

Relações na 1FN:

Projeto (CodProj, Nome, Descr)

Neste caso, os atributos Nome e Descr dependem somente da chave (CodProj). Portanto, esta relação já se encontra na 2FN.

ProjEmp (CodEmp, CodProj, NomeEmp, Categoria, Salário, Dt_I, Dt_F)

Neste caso, os atributos NomeEmp, Categoria e Salário dependem somente da chave CodEmp. Os atributos Dt_I e Dt_F dependem da chave primária completa CodEmp + CodProj (para determinar a data em que um *empregado* iniciou e finalizou um *projeto*).

Para passar à 2FN (eliminar dependências de parte da chave) é necessário dividir a tabela **ProjEmp** em duas tabelas:

ProjEmp (CodEmp, CodProj, Dt_I, Dt_F)

Empregado (CodEmp, NomeEmp, Categoria, Salário)

Passo 1: Copiar para a 2FN cada tabela que tenha chave primária simples ou que não tenha colunas além da chave.

Passo 2: Para cada tabela com chave primária composta e com, pelo menos, uma coluna não chave:

A) Criar na 2FN uma tabela com as chaves primárias da tabela na 1FN.

B) Para cada coluna não chave, verificar se depende de toda a chave primária ou não. Se depender, criar a coluna correspondente na tabela com a chave completa na 2FN. Se não depender, criar, caso ainda não exista, uma tabela na 2FN que tenha como chave primária a parte da chave que é determinante da coluna em questão juntamente com a coluna.

Terceira Forma Normal (3FN)

Uma tabela está na 3FN quando está na 2FN e toda coluna não chave depende diretamente da chave primária. Desta forma não há dependências funcionais transitivas ou indiretas. Tais dependências ocorrem quando uma coluna não chave depende funcionalmente de outra coluna ou combinação de colunas não chaves.

Relações na 2FN:

Projeto (CodProj, Nome, Descr)

ProjEmp (CodEmp, CodProj, Dt_I, Dt_F)

Empregado (CodEmp, NomeEmp, Categoria, Salário)

Na tabela Empregado existe uma dependência transitiva: Salário depende de Categoria que é um campo não chave. Sendo assim:

Empregado (CodEmp, NomeEmp, Categoria, Salário)



Dependência Transitiva

Passo 1: Copiar para a 3FN cada tabela que tenha menos que duas colunas não chave, pois neste caso não ocorrerá casos de dependências transitivas.

Passo 2: Para cada tabela com duas ou mais colunas não chave:

- Criar na 3FN uma tabela com a chave primária da tabela em questão. Para cada coluna não chave, verificar se há dependência de alguma outra coluna não chave. Se não existir, copiar a coluna para a tabela na 3FN. Se existir, criar, caso ainda não exista, uma tabela na 3FN que tenha como chave primária a coluna que origina dependências, juntamente com as colunas que dependem desta. A coluna determinante deve também permanecer na tabela original.

Relações na 3FN:

Projeto (CodProj, Nome, Descr)

ProjEmp (CodEmp, CodProj, Dt_I, Dt_F)

Empregado (CodEmp, NomeEmp, Categoria)

Categoria (Categoria, Salário)

EXERCÍCIO: Dada a tabela abaixo, identifique se está normalizada. Caso contrário, obtenha a 1FN, 2FN e 3FN.

Matrícula	Nome	Cod Cargo	NomeCargo	CodProj	DataFim Projeto	Horas
120	João	1	Programador	01,08	17/07/95, 12/01/96	37, 12
121	Hélio	1	Programador	01,08, 12	17/07/95, 12/01/96, 21/03/96	45, 21,107
270	Gabriel	2	Analista	08, 12	12/01/96, 21/03/96	10, 38