Parte

7

NORMALIZAÇÃO

Nas sessões anteriores foi possível compreender como se dá a analise de requisitos de um negócio, a conseqüente formatação de um banco de dados usando a abordagem entidade-relacionamento e sua transformação para Modelo Lógico.

Ainda dentro do Modelo Lógico de dados é importante saber que a criação de um banco de dados não advém somente da análise de requisitos, podendo se formar a partir de uma coleção de dados armazenados de forma organizada, porém não informatizado. Assim sendo, é importante para o analista ou DBA saber aproveitar esta organização e construir modelo de dados baseado no que já existe.

Dentro deste novo conceito de modelagem dos dados é importante rever nas seções anteriores os conceitos de independência de dados e de consistência de dados. Neste último, destaca-se o fato da consistência de dados poder ser obtida de diversas formas, a saber:

- Pelo Sistema Gerenciador de Banco de Dados;
- Pelos aplicativos; e
- Pela própria construção do sistema.

Pelo próprio Sistema Gerenciador de Banco de Dados é oferecida consistência através de algumas regras de integridade que ele mesmo implementa, garantindo, por exemplo, validade e completeza das informações.

As regras mais importantes oferecidas pelo Sistema Gerenciador de Banco de Dados são:

- Integridade de domínio;
- Integridade de Chave;
- Integridade de Vazio;
- Integridade referencial.

SGBD → Regras de Integridade

Validade
Completeza

Pelos aplicativos, consideramos que nada foi implementado no Sistema Gerenciador de Banco de Dados. Assim sendo, é indispensável que sejam criadas rotinas para garantia de uma consistência mínima dos dados, como por exemplo, o respeito a domínios de chave estrangeira.

Pela própria construção do sistema, que é o que nos interessa neste momento. Por este método é necessário controlar a construção do sistema através da criação de tabelas segundo regras que garantam a manutenção de certas propriedades. Assim, as tabelas que atendem a um determinado conjunto de regras, diz-se estarem em uma determinada **forma normal**.

7.1 DEPENDÊNCIA FUNCIONAL

O conceito de dependência funcional é muito importante para o entendimento da normalização. Uma dependência funcional é uma restrição entre dois conjuntos de atributos de uma base de dados.

Se o valor de um atributo ou conjunto de atributos **A** permite descobrir o valor de outro atributo ou conjunto de atributos **B**, dizemos que **A** determina funcionalmente **B**, ou que **B** depende de **A**, e denotamos:

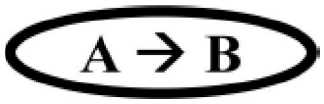

Figura 53 - Representação da Dependência Funcional

Exemplos:

RG → { Nome, CIC, Depto., RG_Supervisor, Salário }

Número_Projeto → { Nome_Projeto, Localização }

{ RG_Empregado, Número_Projeto } → **Horas**

Observe as tabelas abaixo:

Matrícula	Nome	Endereço	...	...	...	...
85001	Pedro	Av. D1, 25				
85001	Pedro	Av. D1, 25				
85001	Pedro	Av. D1, 25				
86005	Ana	Av. C-4, 35				

Dependência funcional de matrícula

...	...	...	Disciplina	Descrição	Instrutor	...
			CP302	Banco de Dados	Nestor	
			CP303	Comunicações	Ana	
			CP304	Eng. Software	Jorge	
			CP302	Banco de Dados	Nestor	

Dependência funcional de disciplina

...	...	...	...	...	Instrutor	Sala
					Nestor	102
					Ana	500
					Jorge	102
					Nestor	1024

Dependência funcional de instrutor

7.2 NORMALIZAÇÃO

O processo de normalização pode ser visto como o processo no qual são eliminados esquemas de relações (tabelas) não satisfatórias, decompondo-as, através da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam as propriedades desejadas.

O processo de normalização como foi proposto inicialmente por Codd. Esse processo conduz um esquema de relação através de uma bateria de testes para certificar se o mesmo encontra-se na 1ª, 2ª e 3ª Formas Normais. Essas três Formas Normais são baseadas nas dependências funcionais dos atributos da relação.

7.2.1 Objetivo da Normalização

- Evitar problemas provocados por falhas no projeto, bem como eliminar a "mistura de assuntos" e repetições desnecessárias de dados.

- Uma Regra de Ouro que devemos observar quando do Projeto de um Banco de Dados baseado no Modelo Relacional de dados é a de "não misturar assuntos em uma mesma tabela".

O Processo de Normalização aplica uma série de Regras sobre as Tabelas de um Banco de Dados, para verificar se estas estão corretamente projetadas. Embora existam cinco formas normais (ou regras de Normalização), na prática usamos um conjunto de três Formas Normais.

Normalmente após a aplicação das Regras de Normalização, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente existente.

Este processo causa a simplificação dos atributos de uma tabela, colaborando significativamente para a estabilidade do modelo de dados, reduzindo-se consideravelmente as necessidades de manutenção.

7.2.2 Vantagens da Normalização

- Otimiza o desempenho das atualizações;

- Agrupa dados de tal forma que numa Relação cada dado torna-se um dado dependente da Chave, de toda a Chave e nada mais que a Chave primária da Relação;

- Provê um meio formal de se estruturar a informação, de tal forma, que fica claro, que tipos de dados existem e que dependências funcionais são satisfeitas.

- Ajuda a dirimir dúvidas de Projeto, pois por vezes, com a finalidade de melhorar o desempenho das funções de acesso, o projetista do Banco de Dados pode tomar decisões que comprometam as funções de atualização. Ao seguir estritamente o processo de normalização isso seria evitado;

7.2 FORMAS NORMAIS

7.2.1 Primeira Forma Normal (1ª FN)

- Uma Tabela está na Primeira Forma Normal quando seus atributos não contêm grupos de repetição;

- Todos os atributos de uma tabela devem ser atômicos (indivisíveis), ou seja, não são permitidos atributos multivalorados, atributos compostos ou atributos multivalorados compostos.

- Tem por objetivo corrigir registros em que, para a chave primária, ocorrem "grupos de repetição", ou seja, atributos que possam assumir múltiplos valores.

Regra:

Se um depósito de dados apresentar grupos de repetição, desmembrá-lo, criando depósitos menores e sem grupo de repetição, ou separar tabelas aninhadas.

Exemplo:

CodPesq	Tipo	Desc	Matr	Nome	Cat	Cred	DtMatr	Duracao
MA001	Mat. Aplic	Informática	5645 5963 8942 1787 4893	Mario Carlos Manoel Joaquim Osmar	M1 M2 G3 D4 D3	20 30 17 60 54	12/03/00 15/02/00 25/04/00 07/01/00 25/03/00	10 12 16 30 28
INF003	Sist. Lin	Redes Neurais	4893 5645 8942	Osmar Mario Manoel	D3 M1 G3	54 20 17	25/03/00 12/03/00 25/04/00	28 10 16

Tabela aninhada

Pesquisa

CodPesq	Tipo	Desc
MA001	Mat. Aplic	Informática
INF003	Sist. Lin	Redes Neurais

Aluno

CodPesq	Matr	Nome	Cat	Cred	DtMatr	Duracao
MA001	5645	Mario	M1	20	12/03/00	10
MA001	5963	Carlos	M2	30	15/02/00	12
INF003	8942	Manoel	G3	17	25/04/00	16
INF003	1787	Joaquim	D4	60	07/01/00	30
INF003	4893	Osmar	D3	54	25/03/00	28

Figura 54 - 1ª Forma Normal

7.2.1 Segunda Forma Normal (2ª FN)

- Uma Tabela está na Segunda Forma Normal quando, além de encontrar-se na primeira forma normal, cada coluna não chave depende da chave primária completa.

- Tem por objetivo corrigir registros de chave composta que tenham atributos dependentes apenas de uma parte da chave.

Regra:

Se um depósito de dados apresentar atributos dependentes apenas de uma parte da chave composta, desmembrá-lo, criando depósitos menores em que os atributos sejam dependentes de toda a chave.

Obs.:

Se na 1ª Forma Normal a tabela possuir apenas uma coluna como chave primária, então ela já está na 2ª Forma Normal.

Exemplo:

Aluno

CodPesq	Matr	Nome	Cat	Cred	DtMatr	Duracao
MA001	5645	Mario	M1	20	12/03/00	10
MA001	5963	Carlos	M2	30	15/02/00	12
INF003	8942	Manoel	G3	17	25/04/00	16
INF003	1787	Joaquim	D4	60	07/01/00	30
INF003	4893	Osmar	D3	54	25/03/00	28

Dependência Parcial

Pesquisa

CodPesq	Tipo	Desc
MA001	Mat. Aplic	Informática
INF003	Sist. Lin	Redes Neurais

PesqAluno

CodPesq	Matr	DtMatr	Duracao
MA001	5645	12/03/00	10
MA001	5963	15/02/00	12
INF003	8942	25/04/00	16
INF003	1787	07/01/00	30
INF003	4893	25/03/00	28

Aluno

Matr	Nome	Cat	Cred
5645	Mario	M1	20
5963	Carlos	M2	30
8942	Manoel	G3	17
1787	Joaquim	D4	60
4893	Osmar	D3	54

A Segunda Forma normal está ligada a ocorrência de **Chaves Primárias compostas.**

Figura 55 - 2ª Forma Normal

7.2.3 Terceira Forma Normal (3ª FN)

- Encontra-se na segunda forma normal; e

- Na definição dos campos de uma entidade podem ocorrer casos em que um campo não seja dependente diretamente da chave primária ou de parte dela, mas sim dependente de outro campo da tabela, campo este que não seja a Chave Primária.

- Tem por objetivo corrigir registros que apresentem atributos dependentes de um atributo não chave.

- Uma coluna não chave primária depende funcionalmente de outra coluna ou combinação de colunas não chave primária.

- Eliminam-se campos calculados.

Regra:

Se um depósito de dados tiver atributo dependente de um atributo não chave, desmembrá-lo, criando depósitos menores em que os atributos só dependam da chave primária.

Exemplo:

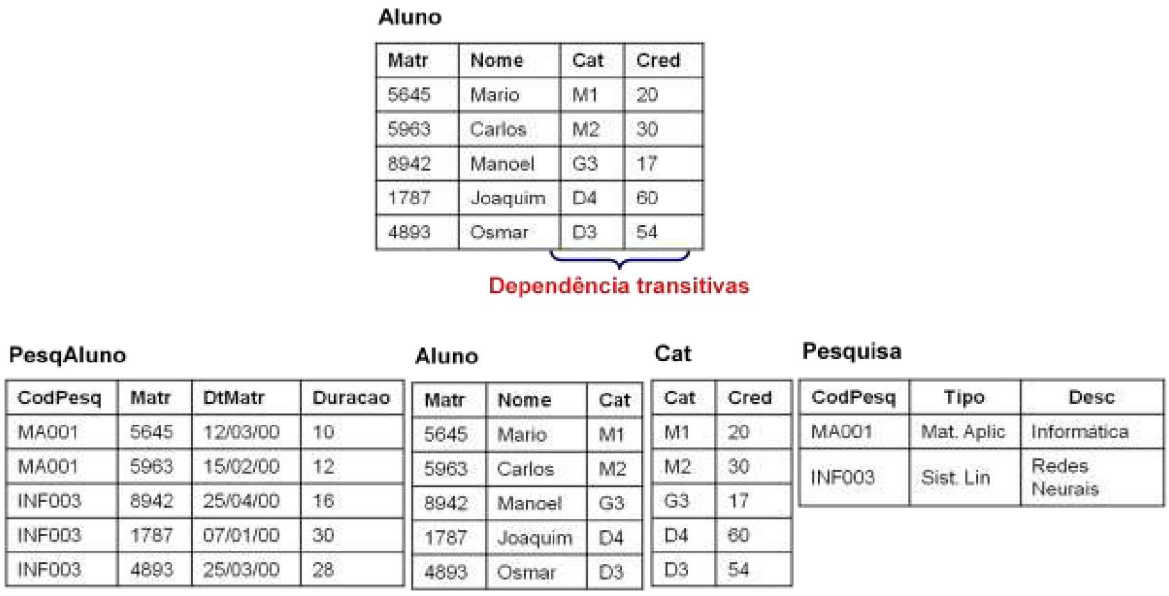

Figura 56 - 3ª Forma Normal

7.2.4 Demais Formas Normais

Para a maioria dos documentos e arquivos, a decomposição até a 3ª Forma Normal é suficiente para obter o esquema de um banco de dados correspondente ao documento. Na literatura encontram-se outras formas normais, como a forma normal de Boyce/Codd, a 4ª Forma Normal e a 5ª Forma Normal.

A **Quarta Forma Normal** é empregada se ainda existir alguma redundância na 3ª Forma Normal. Isso acontecerá quando um atributo não chave contiver valores múltiplos para uma mesma chave.

A **Quinta Forma Normal** é um caso muito raro de ocorrer. É como se tivéssemos na 4ª Forma Normal três campos em uma tabela e esses campos tivessem valores multivalorados.

A **Forma Normal Boyce/Codd** é um aperfeiçoamento da 3ª Forma Normal e serve para impor a condição de que uma tabela contenha dependência apenas em função de chave(s) candidata(s). Qualquer outra dependência deve ser eliminada e transferida para outra tabela onde o determinante seja chave candidata.