NORMALIZAÇÃO:

Resumo baseado no livro *Projeto de Banco de Dados*
Carlos Alberto Heuser

- Técnica que objetiva eliminar redundâncias de dados, através do reagrupamento de informações.
- Baseia-se no conceito de *forma normal*: regra que deve ser obedecida por uma tabela para que esta seja considerada bem projetada.

Considere o seguinte documento exemplo na forma de tabela não normalizada:

CodProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempoAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Silvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
			4112	João	A2	4	4/01/91	24
			6126	José	B1	9	1/11/92	12

A tabela acima pode ser descrita pelo seguinte esquema de tabela relacional:

Proj (codProj, tipo, descr, (codEmp, nome, cat, sal, dataIni, tempoAl))

Passagem à primeira Forma Normal (1FN):

Diz-se que uma tabela está na primeira forma normal, quando ela não contém tabelas aninhadas.

Para transformar um esquema de tabela não normalizada em um esquema na 1FN há duas alternativas:

1) Construir uma única tabela com redundância de dados:
   Cria-se uma tabela na qual os dados das linhas externas à tabela aninhada são repetidos para cada linha da tabela aninhada.
   ProjEmp (CodProj, tipo, descr, codEmp, nome, cat, sal, dataIni, tempoAl)
2) Construir uma tabela para cada tabela aninhada:
   Cria-se uma tabela referente à própria tabela que está sendo normalizada e uma tabela para cada tabela aninhada. A passagem à primeira forma normal por decomposição de tabelas é feita nos seguintes passos:
   1. É criada uma tabela na 1FN referente a tabela não normalizada e que contém apenas as colunas com valores atômicos, isto é, sem as tabelas aninhadas. A chave primária da tabela na 1FN é idêntica a chave da tabela não normalizada.
   2. Para cada tabela aninhada, é criada uma tabela na 1FN composta pelas seguintes colunas:
      - a chave primária de cada uma das tabelas na qual a tabela em questão está aninhada
      - as colunas da própria tabela aninhada

- são definidas as chaves primárias das tabelas na 1FN que correspondem a tabelas aninhadas.

No caso da tabela acima, o esquema resultante seria:
Proj (CodProj, tipo, descr)
ProjEmp (codProj, codEmp, nome, cat, sal, dataIni, tempoAl)

De forma geral, para determinar a chave primária de uma tabela na 1FN que corresponda uma tabela aninhada na forma não normalizada deve-se procededr como segue:

- tomar como parte da chave primária da tabela na 1FN a chave primária da tabela não normalizada
- verificar se esta chave primária é suficiente para identificar as linhas da tabela na 1FN

Caso seja suficiente, a chave primária da tabela na 1FN é a mesma que a da tabela aninhada na forma não normalizada

Caso contrário, deve-se determinar quais as demais colunas que são necessárias para identificas as linhas da tabela na 1FN, compondo assim a chave primária na 1FN.

Passagem à segunda Forma Normal (2FN):

Diz-se que uma tabela está na segunda forma normal, quando, além de estar na 1FN, não contém dependências parciais. Uma dependência parcial ocorre quandouma coluna depende apenas de parte da chave primária. (Em outras palavras, uma tabela está na segunda forma normal, quando, além de estar na 1FN, cada coluna não chave depende da chave primária completa).

Obviamente, uma tabela que está na 1FN e que possui apenas uma coluna como chave primária, já está na 2FN. O mesmo aplica-se para uma tabela que contenha apenas colunas chave primária.

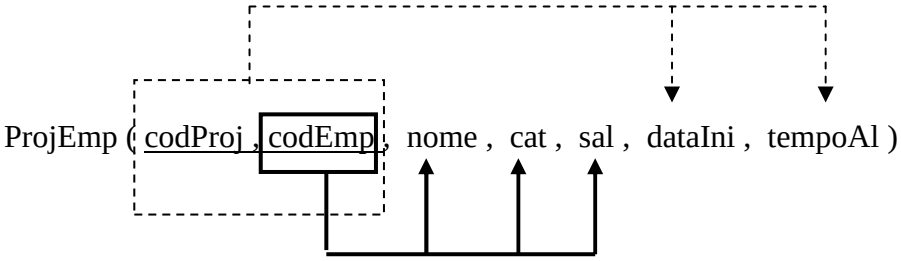
Em uma tabela relacional, diz-se que uma coluna C2 depende funcionalmente de uma coluna C1 (ou que a coluna C1 determina a coluna C2) quando, em todas as linhas da tabela, para cada valor de C1 aparece o mesmo valor de C2.

Ex:

...	Código	...	Salario	...
	E1		10	
	E3		10	
	E1		10	
	E2		5	
	E3		10	
	E2		5	
	E1		10	

A coluna *salario* depende funcionalmente da coluna *codigo*, pelo fato de cada valor de código estar associado sempre ao mesmo valor de salário. Exemplificando, o valor “E1” da coluna código identifica sempre o mesmo valor de salário (“10”).

Considere o exemplo que estamos trabalhando:



Os campos *dataIni* e *tampoAl* dependem da chave primária completa, mas os campos *nome*, *cat* e *sal* dependem apenas de *codEmp*.

A passagem à 2FN visa eliminar as dependências de parte da chave primária. O processo de passagem da 1FN para a 2FN é o seguinte:

1) Copiar para a 2FN cada tabela que tenha chave primária simples ou que não tenha colunas além da chave.
2) Para cada tabela com chave primária composta e com pelo menos uma coluna não chave
   - Criar na 2FN uma tabela com as chaves primárias da tabela na 1FN
   - Para cada coluna não chave:
     - "a coluna depende de toda a chave ou de apenas parte dela?"
       Caso a coluna dependa de toda a chave:
       - criar a coluna correspondente na tabela com a chave completa na 2FN

       Caso a coluna dependa apenas de parte da chave:
       - criar, caso ainda não existir, uma tabela na 2FN que tenha como chave primária a parte da chave que é determinante da coluna em questão
       - criar a coluna dependente dentro da tabela na 2FN

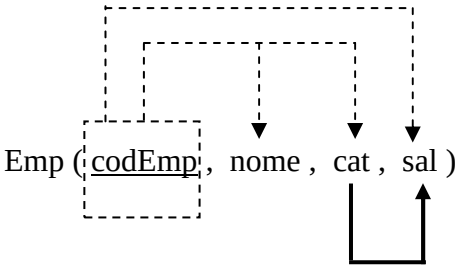
No caso das tabelas resultantes no passo anterior, o esquema na 2FN seria:

Proj (CodProj, tipo, descr)
ProjEmp (codProj, codEmp, dataIni, tempoAl)
Emp (codEmp, nome, cat, sal)

Passagem à terceira Forma Normal (3FN):

Uma tabela está na 3FN, quando além de estar na 2FN, não contém dependências transitivas. Uma dependência funcional transitiva ocorre quando uma coluna, além de depender da chave primária da tabela, depende de outra coluna ou conjunto de colunas da tabela. (Em outras palavras, uma coluna está na 3FN, quando além de estar na 2FN, toda coluna não chave depende diretamente da chave primária).

No exemplo anterior, considere que o salário de um empregado seja determinado pela sua categoria funcional. Neste caso, a informação de qual salário é pago a uma categoria funcional está representado redundantemente na tabela, tantas vezes quantos forem os empregados que possuirem a categoria funcional. A passagem à 3FN visa eliminar este tipo de redundância de dados.



O processo de passagem da 2FN à 3FN é o seguinte:

- copiar para o esquema na 3FN cada tabela que tenha menos que duas colunas não chave, pois neste caso não há como haver dependências transitivas
- para tabelas com duas ou mais colunas não chave:
  - criar uma tabela no esquema da 3FN com a chave primária da tabela em questão
  - para cada coluna não chave fazer a seguinte pergunta:
    - "a coluna depende de alguma outra coluna não chave? (dependência transitiva ou indireta)"

      Caso a coluna dependa apenas da chave
      - copiar a coluna para a tabela na 3FN

      Caso a coluna dependa de outra coluna
      - criar, caso ainda não exista, uma tabela no esquema na 3FN que tenha como chave primária a coluna da qual há a dependência indireta
      - copiar a coluna dependente para a tabela criada
      - a coluna determinante deve permanecer também na tabela original

No caso das tabelas resultantes no passo anterior, o esquema na 3FN seria:

Proj (CodProj, tipo, descr)
ProjEmp (codProj, codEmp, dataIni, tempoAl)
Emp (codEmp, nome, cat)
Cat (cat, sal)

Exercício:

1) No contexto de um sistema de controle acadêmico, considere a tabela abaixo:

Matricula (codAluno, codTurma, codDisciplina, nomeDisciplina, nomeAluno, codLocalNascimentoAluno, nomeLocalNascimentoAluno)

Verifique se a tabela obedece a segunda e a terceira formas normais. Caso não obedeça, faça as transformações necessárias.