

# Normalização

# Normalização

- A normalização é um método que se emprega para aumentar a qualidade do projeto de banco de dados. É também uma base teórica para a definição das propriedades das relações.
- Através do processo de normalização pode-se, gradativamente, substituir um conjunto de entidades e relacionamentos por um outro, mais eficiente em relação às anomalias de atualizações (inclusão, alteração e exclusão) as quais podem causar certos problemas, tais como:
  - Informações redundantes no banco de dados;
  - Dependências parciais em relação a chave concatenada;
  - Redundâncias de dados desnecessárias;
  - Perdas acidentais de informação;
  - Dependências transitivas entre atributos não chave.

# Normalização

- Para resolver os problemas citados foram propostas formas normais que devem ser aplicadas a um modelo de dados com o objetivo de garantir um modelo de dados livre dos problemas citados acima, estável e eficiente.
- A Normalização é necessária porque podem ocorrer erros de modelagem, sendo que a normalização se apresenta como uma técnica para garantir a integridade de um modelo de dados.

# Normalização

Número Pedido	Nome do Cliente	CNPJ	Insc. Estadual	Código Produto	Qtde	Descrição	Valor Unitário	Unidade	Valor Total
3445	TCA	11111	111	45	20	Álcool	5,00	L	100,00
3446	TCA	11111	111	130	2	Tecido	20,00	M	40,00
3446	TCA	11111	111	35	30	Farinha	1,00	KG	30,00
9756	HTZ	456456	222	35	20	Farinha	1,00	KG	20,00
9756	HTZ	456456	222	90	60	Cola	3,00	L	180,00
2610	LOPES	995566	333	78	50	Cimento	30,00	KG	1500,00
2610	LOPES	995566	333	45	50	Álcool	5,00	L	250,00

# Normalização

- Caso esta entidade fosse implementada como uma tabela em um banco de dados, as seguintes anomalias iriam aparecer:
  - **Anomalia de inclusão:** ao ser incluído um novo cliente, o mesmo tem que estar relacionado a uma venda obrigatoriamente; Ao ser cadastrado uma nova venda, o mesmo cliente deverá ser cadastrado novamente;
  - **Anomalia de exclusão:** ao ser excluído um cliente, os dados referentes as suas compras serão perdidos
  - **Anomalia de alteração:** ao ser alterado por exemplo, o preço unitário de um determinado produto, será preciso atualizar todos os pedidos já cadastrados que tenham aquele determinado produto alterado, e alterar o valor do mesmo produto;

# Normalização

- **Formas Normais**
- As formas normais descrevem uma classificação de relações
  - O trabalho inicial de Codd identificou a primeira (1FN), a segunda (2FN) e a terceira (3FN) formas normais.
  - Posteriormente, outros pesquisadores acrescentaram a forma normal Boyce-Codd (FNBC), a quarta (4FN) e a quinta (5FN) formas normais.

- **Primeira Forma Normal (1FN)**

- Diz-se que uma tabela está na primeira forma normal quando ela não contém tabelas aninhadas.
- A primeira forma normal assegura que não existam repetições de valores nos atributos nem grupos repetidos de atributos das entidades de um modelo de dados.
- Em uma determinada realidade, às vezes encontramos algumas informações que se repetem, retratando ocorrências de um mesmo fato dentro de uma única linha e vinculada a sua chave primária.

- **Primeira Forma Normal (1FN)**

- O objetivo da primeira forma normal é eliminar o aninhamento de tabelas para que cada tabela tenha informações de um único assunto. Não podemos ter mais de um assunto em uma tabela.
- Ao observarmos a entidade PEDIDO, apresentada anteriormente, visualizamos que um certo grupo de atributos (produtos solicitados) se repete (número de ocorrências não definidas) ao longo do processo de entrada de dados na entidade.
- A 1FN diz que a tabela não deve conter mais de um assunto, nem grupos repetitivos de atributos ( atributos multivalorados ).

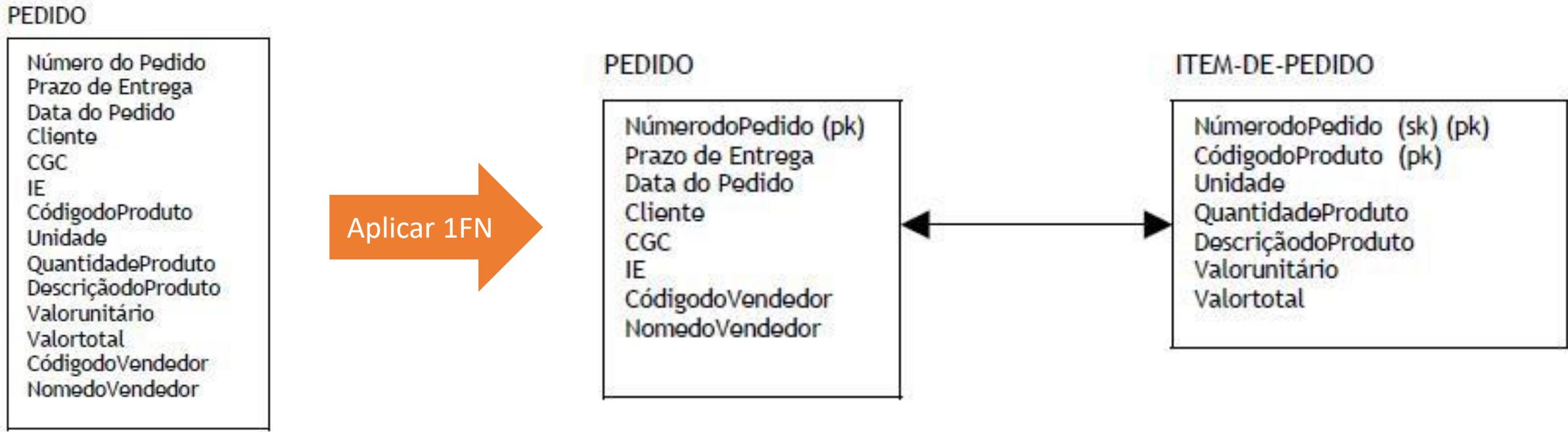
# Normalização

- **Primeira Forma Normal (1FN)**

- Para se obter uma tabela na primeira forma normal ( 1FN ), é necessário decompor a tabela não normalizada em tantas tabelas quanto for o número de conjuntos de atributos repetitivos.
  - Deve ser criada uma tabela para a tabela não-normalizada e uma tabela para cada tabela aninhada na tabela não-normalizada.
  - Nas novas entidades criadas, a chave primária é a concatenação da chave primária original mais os atributos do grupo repetitivo visualizados como chave primária deste grupo.

# Normalização

- Primeira Forma Normal (1FN)



Ao aplicarmos a 1FN sobre a tabela **PEDIDO**, obtemos mais uma tabela chamada **ITEM-DE-PEDIDO**, que herdará os atributos repetitivos e destacados da tabela **PEDIDO**.

Um **PEDIDO** possui no mínimo 1 e no máximo N ocorrências em **ITEM-DE-PEDIDOS** e um **ITEM-DE-PEDIDOS** pertence a 1 e somente 1 **PEDIDO**, logo o relacionamento **POSSUI** é do tipo 1:M (OU 1:N).

# Normalização

- **Dependência Funcional**

- Dada uma tabela qualquer, dizemos que uma coluna ou conjunto de colunas A é dependente funcional de um outra coluna B, se a cada valor de B existir nas linhas da tabela um único valor de A. Em outras palavras, A depende funcionalmente de B.

Exemplo:

- Na tabela PEDIDO, a coluna PRAZO-DE-ENTREGA depende funcionalmente da coluna NÚMERO-DO-PEDIDO. Em outras palavras, o NÚMERO-DO-PEDIDO determina o PRAZO-DE-ENTREGA.

# Normalização

- **Dependência Funcional Total**

- Na ocorrência de uma chave primária concatenada, dizemos que um atributo ou conjunto de atributos depende de forma completa ou total desta chave primaria concatenada quando para cada valor da chave está associado um valor para cada atributo.
- Como exemplo de dependência funcional total temos na entidade ITEM-DO-PEDIDO, o atributo QUANTIDADE-DO-PRODUTO depende de forma total da chave primária concatenada ( NÚMERO-DO-PEDIDO + CÓDIGO-DO-PRODUTO ).
- A dependência Total só ocorre quando a chave primária for concatenada, composta por várias colunas e quando existe um ou mais colunas que dependem da chave primária.

# Normalização

- **Dependência Funcional Parcial**

- Quando uma coluna ou conjunto de colunas A depende de outra coluna B que faz parte da chave primária concatenada dizemos que há uma dependência funcional parcial A em relação a B. Para cada valor da coluna B existe um valor associado para a coluna A.
- A dependência funcional parcial só existe quando a tabela possui chave primária composto por mais de uma coluna.
- Como exemplo de dependência funcional Parcial temos na entidade ITEM-DO-PEDIDO, as colunas NOME\_PRODUTO, PREÇO\_UNITARIO que dependem de forma parcial da chave primária concatenada ( NÚMERO-DO-PEDIDO + CODIGO\_PRODUTO ) através da coluna CODIGO-PRODUTO.

# Normalização

- **Dependência Funcional Transitiva**

- Quando uma coluna ou conjunto de colunas A depende de outra coluna B, que não pertence à chave primária, dizemos que A é dependente transitivo de B
- Como exemplo de dependência funcional transitiva temos que na tabela PEDIDO, as colunas NOME DO CLIENTE, ENDERECO, CIDADE, UF, CGC e INSCRICAO ESTADUAL são dependentes da coluna CGC que não pertence à chave primária da tabela PEDIDO.
- Ainda na tabela PEDIDO, o atributo NOME DO VENDEDOR é dependente transitivo do atributo CODIGO DO VENDEDOR que também não faz parte da chave primária da tabela PEDIDO.

# Normalização

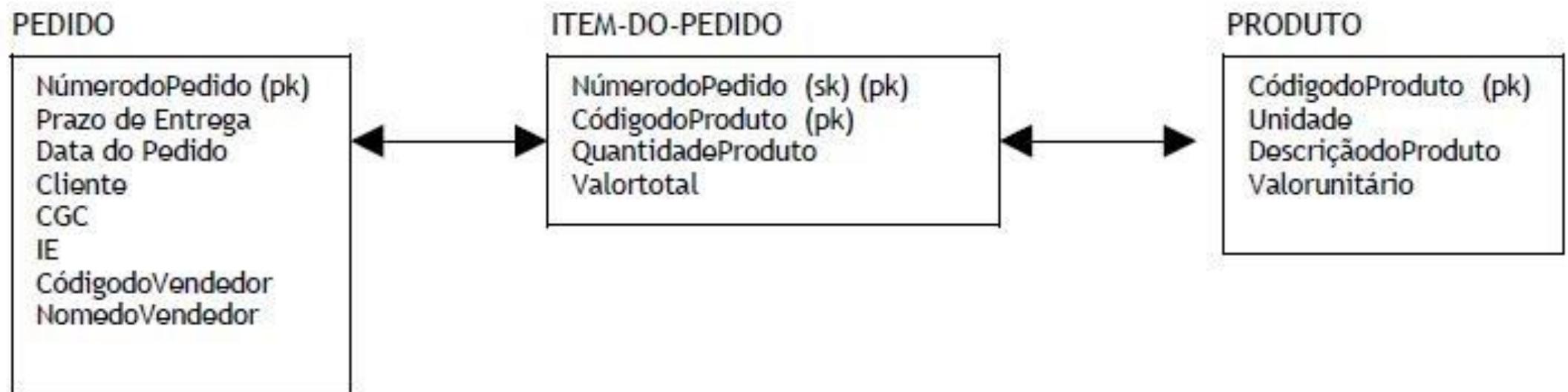
- **Segunda Forma Normal (2FN)**

- A segunda forma normal assegura que não exista dependência funcional parcial no modelo de dados.
- Para aplicarmos a segunda forma formal em uma tabela devemos observar se a tabela possui chave primária composta e verificar se existe alguma coluna com dependência parcial em relação a alguma coluna da chave primária concatenada.
- A aplicação da segunda forma normal ( 2FN ) sobre a tabela em observação gera novas tabelas, que herdarão a chave parcial e todos os atributos que dependem parcialmente da chave primária, ou seja, uma entidade para estar na 2FN não pode ter atributos com dependência funcional parcial em relação à chave primária.

# Normalização

## • Segunda Forma Normal (2FN)

- A entidade ITEM-DO-PEDIDO apresenta uma chave primária concatenada e por observação, notamos que os atributos UNIDADE, DESCRICAO-DO-PRODUTO e VALOR-UNITÁRIO dependem de forma parcial do atributo CODIGO-DOPRODUTO, que faz parte da chave primária.
- Quando aplicamos a 2FN sobre ITEM-DO-PEDIDO, será criada a entidade PRODUTO que herdará os atributos UNIDADE, DESCRICAO-DO-PRODUTO e VALOR-UNITARIO e terá como chave primária CODIGO-DO-PRODUTO.



- Se for preciso cadastrar um novo produto, sem que o mesmo não possua nenhum item de pedido, como poderia ser feito ?

# Normalização

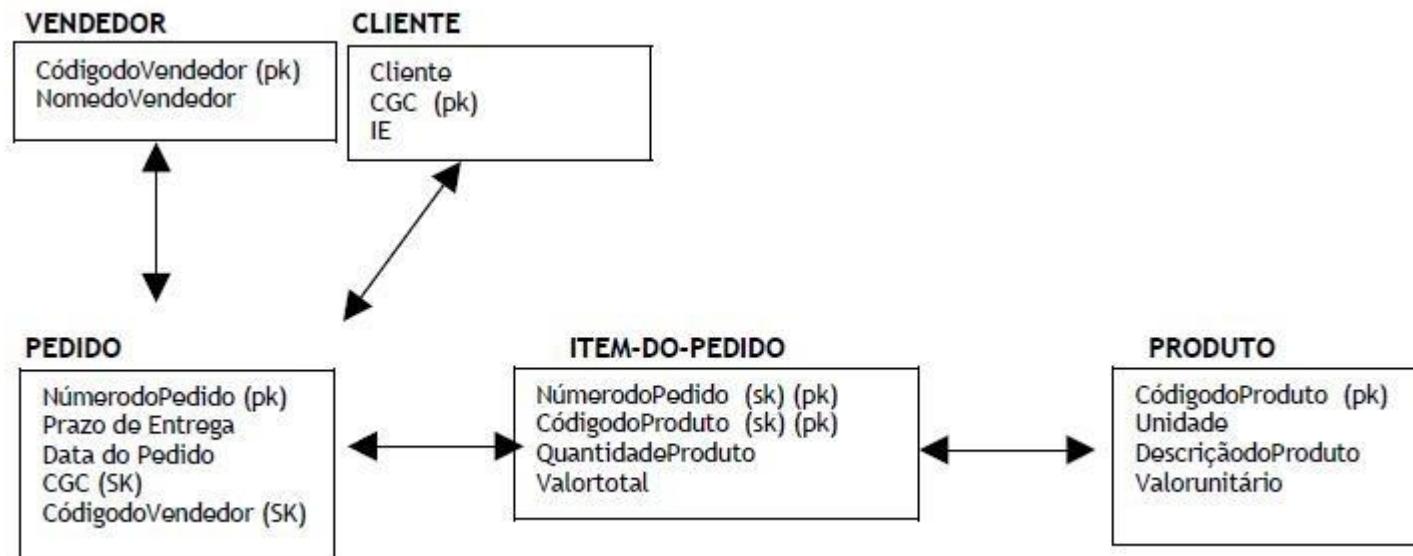
- **Terceira Forma Normal (3FN)**

- A terceira forma normal assegura que nenhuma tabela do modelo relacional possui atributos com dependência transitiva.
- Uma tabela está na 3FN se nenhuma coluna possui dependência transitiva em relação a outra coluna que não participe da chave primária. Ao retirarmos a dependência funcional transitiva, devemos criar uma nova entidade que contenha as colunas que dependem transitivamente de outra coluna que não faz parte da chave primária. A chave primária desta nova tabela é composta pela coluna que causou esta dependência.
- Além de não conter colunas com dependência funcional transitiva, as tabelas na 3FN não devem conter colunas derivadas que sejam o resultado de algum cálculo sobre outra coluna, o que de certa forma pode ser entendido como uma dependência funcional transitiva.
- Toda coluna derivada a partir de outra coluna deve ser retirada do modelo dados. O mesmo é válido para colunas que representam cálculos realizados com base em outras colunas da tabela.

# Normalização

- **Terceira Forma Normal (3FN)**

- Na tabela PEDIDO, a coluna VALOR-TOTAL é calculada a partir do produto entre as colunas PREÇO-UNITÁRIO e QUANTIDADE.
- Desta forma, VALOR-TOTAL depende tanto de PREÇO-UNITÁRIO quanto de QUANTIDADE o que constitui uma dependência funcional transitiva. Para eliminar essa anomalia devemos remover esta coluna da tabela.
- na tabela PEDIDO, podemos observar que a coluna NOME-DO-VENDEDOR depende transitivamente da coluna CODIGO-DO-VENDEDOR que não pertence a chave primária. Para eliminarmos esta anomalia devemos criar a tabela VENDEDOR com seus colunas. Da mesma forma e pelo mesmo motivo, também criaremos a tabela CLIENTE com suas colunas, cada tabela com seu assunto específico.



# Normalização

## • Forma Normal de Boyce/Codd(FNBC)

- Foi proposta como uma forma mais simples da 3FN, porém, é considerada como uma forma mais restrita que esta. As definições da 2Fn e 3FN, propostas por Codd, não cobriam certos casos.
- O Prof. Raymond Boyce apontou situações não cobertas em 1974. Os casos não cobertos pelas formas normais de Codd somente ocorrem quando 3 condições são satisfeitas:
  - A entidade possui várias chaves candidatas
  - As chaves candidatas são concatenadas
  - As chaves candidatas compartilham pelo menos um atributo em comum

# Normalização

- **Forma Normal de Boyce/Codd(FNBC)**

- A forma normal Boyce/Codd foi desenvolvida com o objetivo de resolver situações que as três formas normais não resolviam, entre elas o fato de haver várias chaves na entidade, formadas por mais de um atributo (chaves compostas) e sendo que entre elas existe em comum pelo menos um atributo.
- Para estar na FNBC, uma entidade precisa possuir somente atributos que são chaves candidatas.

# Normalização

- **Forma Normal de Boyce/Codd(FNBC)**
  - A diferença está na dependência através de uma chave candidata:
    - Se um atributo depender funcionalmente de um outro atributo que não faça parte da chave, mas que por sua vez dependa da chave (dependência transitiva da chave via outro atributo), então não estará na FNBC.
    - Uma relação na FNBC sempre estará na 3FN, entretanto, uma relação na 3FN pode não estar na FNBC.

# Normalização

- **Forma Normal de Boyce/Codd(FNBC)**

- Na verdade a FNBC é um aperfeiçoamento da 3FN que não resolia certas anomalias presentes na informação contida em uma entidade.
- O problema foi observado porque a 2FN e a 3FN só tratavam dos casos de dependência parcial e transitiva de atributos fora de qualquer chave, porém quando o atributo faz parte de uma chave ( primária ou candidata ) de uma entidade ele não é tratado pelas 2FN e 3FN.
- Uma entidade está na FNBC se, e somente se, todos os determinantes forem chaves candidatas. Notem que esta definição é em termos de chaves candidatas e não sobre chaves primárias.

# Normalização

- **Forma Normal de Boyce/Codd(FNBC)**
- Exemplo:
  - Considere a entidade FILHO que possui os atributos nome, endereço, data nascimento, nome escola, sala, nome professor.
  - Vamos considerar que um professor possa estar associado a mais de uma escola e uma sala, o que normalmente acontece na vida real.
  - Veja que para que se possa identificar o professor precisamos de uma chave concatenada pois somente o nome do professor não é suficiente para garantir a identificação do professor.
  - Assim temos como chaves candidatas NOME ESCOLA + SALA ou NOME ESCOLA + NOME PROFESSOR.

# Normalização

- **Forma Normal de Boyce/Codd(FNBC)**
- Exemplo:
  - Veja que esta entidade atende às três condições para a FNBC:
    - As chaves candidatas para a entidade FILHO são: NOME FILHO + ENDERECO ou NOME FILHO + SALA ou NOME FILHO + NOME PROFESSOR
    - As três chaves apresentam mais de um atributo, são portanto concatenadas
    - As três chaves compartilham um mesmo atributo NOME FILHO

# Normalização

- **Forma Normal de Boyce/Codd(FNBC)**

- Neste exemplo, ao aplicar a FNBC, a entidade FILHO deve ser dividida em duas entidades, uma que contem todos os atributos que designam um professor a uma sala de uma escola e uma outra entidade que contem os atributos que descrevem o FILHO.
- Assim temos duas entidades

FILHO = { nome, endereço, data nascimento, nome escola, sala}

SALA = { nome escola, sala, nome professor }

- Temos duas entidades agora, uma para descrever o filho e outra para descrever o professor que trabalha em uma sala de aula de uma escola.

# Normalização

- **Quinta Forma Normal**

- A quinta forma normal trata do conceito de dependência de junção, quando a noção de normalização é aplicada à decomposição e aplicada na reconstrução ( engenharia reversa ) devido a uma junção.
- A 5FN trata de casos particulares que ocorrem com pouca frequência na modelagem de dados e que são os relacionamentos múltiplos ( ternários, quaternários, ... , n-nários ).
- Ela fala que uma entidade está na sua 5FN quando o conteúdo desta entidade não puder ser reconstruído a partir de outras entidades menores, extraídas desta entidade.
- Ao se fazer a decomposição de uma entidade em outras entidades perde-se o conteúdo, a informação da entidade original, pois as entidades geradas pela decomposição não conseguem representar a informação original.

# Normalização

## • Exercícios

- Aplicar as Formas Normais cabíveis, nas questões abaixo. Você deve transformar os esquemas abaixo em conjuntos de esquemas que estejam na 2NF, 3NF e, justificar sua normalização de acordo com suas dependências funcionais.
  - Empregado (Número Empregado, Nome do Empregado, Número do Departamento, Nome do Departamento, Número do Gerente, Nome do Gerente, Número do Projeto, Nome do Projeto, Dia de Início do Projeto, Número de horas trabalhadas no projeto).
  - Ordem\_Compra (cd\_ordem\_compra, dt\_emissão, cd\_fornecedor, nm\_fornecedor, endereço\_fornecedor, cd\_material (n vezes), descrição\_material (n vezes), qt\_comprada (n vezes), vl\_unitário (n vezes), vl\_total\_item (n vezes), vl\_total\_ordem).

# Normalização

## • Exercícios

- Aplicar as Formas Normais cabíveis, nas questões abaixo. Você deve transformar os esquemas abaixo em conjuntos de esquemas que estejam na 2NF, 3NF e, justificar sua normalização de acordo com suas dependências funcionais.
  - Tabela de Notas Fiscais (Num\_NF, Série, Data emissão, Código Cliente, Nome cliente, Endereço cliente, CNPJ cliente, Código Mercadoria, Descrição Mercadoria, Quantidade vendida, Preço de venda, Total da venda da Mercadoria e Total Geral da Nota). Cada nota pode ter mais do que uma mercadoria.
  - Inscrição (Código do Aluno, Nome do Aluno, Telefone para contato, Ano de Admissão, Código da Disciplina, Nome da Disciplina, Nome do Curso, Data da Matrícula).
  - Paciente (num\_paciente, nome\_paciente, num\_quarto, descrição\_quarto, num\_cômodos\_quarto, {cód\_médico, nome\_médico, fone\_médico}).