

# Normalização e seus conceitos

Profa. Ana Paula Wauke

# Chave

- Superchave – conjunto de atributos que identificam um registro dentro de um conjunto;
- Problema para ser chave:
  - Pode haver atributos de outras entidades;
- Chaves candidatas: poderiam ser identificadoras de registro; envolve conjuntos de atributos que poderiam assumir tais papéis identificadores;
- Chave primária: chave candidata escolhida.
- Chaves alternativas: candidatas que não são primária;
- Chave estrangeira: faz ligação com chave primária de outra tabela;

# Restrições para Relações

- Cardinalidade
- Dependência de existência;
  
- Relações: tabelas.

# Fases do Projeto E-R

- Especificação das necessidades do usuário;
- Projeto conceitual - visão da empresa;
- Especificação das necessidades funcionais;
- Projeto físico;
  
- Na criação das tabelas:
  - Importante: normalização;

# Normalização

- 6 mais conhecidas:
  - 1FN (1a. Forma normal) ;
  - 2FN (2a. Forma normal)
  - 3FN (3a. Forma normal)
  - FNBC (Forma normal de Boyce e Codd)
  - 4FN (4a. Forma normal)
  - 5FN (5a. Forma normal)
- Uma forma normal engloba as anteriores, ou seja, para estar na 2a. tem que estar na 1a. e assim sucessivamente

# Normalização

- Dependência funcional:
  - Um atributo X identifica atributo Y, ou seja, Y depende de X;
  - $X \twoheadrightarrow Y$ ; (lê-se X determina Y ou Y depende de X);
  - Ex.: cidade  $\twoheadrightarrow$  estado; estado  $\twoheadrightarrow$  país;
  - Dependência funcional trivial: elemento é parte do determinante. Ex.: {banco, agencia}  $\twoheadrightarrow$  banco;
  - Dependência funcional não trivial: determinante identifica atributo qualquer (esse é o caso de nosso interesse). Ex.: {banco,agencia}  $\twoheadrightarrow$  cidade;

# Normalização

- Dependência funcional:
  - Outro exemplo:
    - Entidade: EMPREGADO
    - MATRÍCULA ---> NOME\_EMPREGADO, ENDERECO\_EMPREGADO, SALÁRIO
  - Dependencia funcional composta:
    - Ex.: Entidade: EMPREGADO
    - EMPRESA, MATRICULA --> NOME\_EMPREGADO, ENDEREÇO\_EMPREGADO, SALÁRIO

# Normalização

- Transitividade:
  - X determina Y e Y determina Z;
  - Ex.: cidade  $\Rightarrow$  estado e estado  $\Rightarrow$  país; então cidade  $\Rightarrow$  país;
  - Outro exemplo:
    - Entidade: DEPARTAMENTO
    - CODIGO\_DEPARTAMENTO  $\rightarrow$  NOME\_DEPARTAMENTO, SIGLA\_DEPARTAMENTO, MATRICULA\_GERENTE, NOME\_GERENTE
    - Neste caso o atributo NOME\_GERENTE depende do atributo MATRICULA\_GERENTE. O atributo MATRICULA\_GERENTE depende da chave primária CODIGO\_DEPARTAMENTO.



# Normalização (1FN)

- 1FN (1ª. Forma Normal)
  - Uma entidade está na 1FN se ela não contém grupo de atributos repetitivos (multivalorados);
  - exemplo de PEDIDO DE VENDA:
  - Um pedido de vendas possui os dados: Número do pedido, data pedido, código do cliente, nome do cliente, endereço do cliente, a lista de: (codigo de produto, nome do produto, quantidade pedida, preço do produto, total do produto) e total geral do pedido.

# Normalização (1FN)

- Na estrutura não normalizada teríamos uma tabela ou entidade chamada pedido com os atributos acima:

Cod Pedido	Data Pedido	Cod Cliente	Nome Cliente	End. Cliente	Cod. Produto	Nome Prod.	Qtde	Preco	Total Produto	Total Pedido
1	11/04/2000	2	Ana	Av K, 1	32	CD	3	5	15	35
1	11/04/2000	2	Ana	Av K, 1	45	DVD	1	8	8	35
1	11/04/2000	2	Ana	Av K, 1	18	Modem	4	3	12	35
2	20/04/2000	1	Fernando	Rua Z, 3	18	Modem	1	3	3	9
2	20/04/2000	1	Fernando	Rua Z, 3	32	CD	1	6	6	9
3	15/03/2000	2	Ana	Av K, 1	100	Pentium	2	50	100	106
3	15/03/2000	2	Ana	Av K, 1	101	Papel	2	1	2	106
3	15/03/2000	2	Ana	Av K, 1	560	Camisa	2	2	4	106
4	18/04/2000	4	Roberto	Av X, 32	18	Modem	1	3	3	3

Supõem-se que o atributo derivado total do pedido não será gravado no banco uma vez que pode ser gerado via processamento (programa).

# Normalização (1FN)

- Analisando as operações básicas sobre a tabela descobrimos as anomalias desta representação não normalizada:
  - **Inserção** não se pode entrar com um novo cliente até que o cliente efetue um pedido de venda. A mesma observação vale para produto.
  - **Remoção** se removermos um pedido, destruiremos informações sobre produto e cliente.
  - **Atualização** Se for alteração a descrição de um produto ou nome de cliente, deve-se pesquisar seqüencialmente toda a tabela para efetuar a alteração.
- Há problemas de armazenamento de informações redundantes, como dados de cliente e produto, entre outros problemas.

# Normalização (1FN)

- A solução para estes problemas é dividir esta relação em duas outras, que denominaremos PEDIDO e PRODUTO-PEDIDO.
- Ficando na 1FN assim:
- Entidade **PEDIDO**: NUMERO\_PEDIDO --> DATA\_PEDIDO, CODIGO\_CLIENTE, NOME\_CLIENTE, ENDERECO\_CLIENTE, TOTAL\_PEDIDO.
- Entidade **PRODUTO\_PEDIDO**: NUMERO\_PEDIDO, COD\_PRODUTO --> NOME\_PRODUTO, QTDE\_PEDIDA, PRECO\_PRODUTO

# Normalização (1FN)

- Aplicações da 1FN:
  - Atributos compostos serem decompostos e armazenados em simples;
  - Para cada atributo multivalorado cria-se uma tabela: (chave primária passa como chave estrangeira , atributo multivalorado) - ambos são chaves primárias → chave composta;
  - Para cada grupo repetido, cria-se tabela nova com grupos repetidos + chave primária da inicial → comporá chave primária.

# Normalização (1FN)

- 1FN (1ª. Forma Normal)
  - Uma entidade está na 1FN se ela não contém grupo de atributos repetitivos (multivalorados);
  - exemplo de PEDIDO DE VENDA:
  - Um pedido de vendas possui os dados: Número do pedido, data pedido, código do cliente, nome do cliente, endereço do cliente, a lista de: (codigo de produto, nome do produto, quantidade pedida, preço do produto, total do produto) e total geral do pedido.

# Normalização (2FN)

- Uma relação está na 2FN se e somente se ela estiver na 1FN e todos os atributos não chave forem funcionalmente dependentes da chave primária.
- Para 2FN: **Remover os atributos não funcionalmente dependentes de toda uma chave primária.**
- **Exemplo:**
- A entidade PEDIDO não apresenta este problema, portanto já está na 2FN.
- A entidade PRODUTO\_PEDIDO já apresenta problemas e precisa remover os atributos.
  - Entidade **PRODUTO\_PEDIDO**: NUMERO\_PEDIDO, COD\_PRODUTO --> NOME\_PRODUTO, QTDE\_PEDIDA, PRECO\_PRODUTO
- Fica:
  - Entidade **PRODUTO\_PEDIDO**: NUMERO\_PEDIDO, COD\_PRODUTO --> QTDE\_PEDIDA
  - Entidade **PRODUTO**: COD\_PRODUTO --> NOME\_PRODUTO, PRECO\_PRODUTO

# Normalização (2FN)

- A solução completa ficará na 2FN assim:
  - Entidade **PEDIDO**: NUMERO\_PEDIDO --> DATA\_PEDIDO, CODIGO\_CLIENTE, NOME\_CLIENTE, ENDERECO\_CLIENTE, TOTAL\_PEDIDO.
  - Entidade **PRODUTO\_PEDIDO**: NUMERO\_PEDIDO, COD\_PRODUTO --> QTDE\_PEDIDA, PRECO\_NEGOCIADO\_PRODUTO
  - Entidade **PRODUTO**: COD\_PRODUTO --> NOME\_PRODUTO, PRECO\_PRODUTO
- **Necessidade levantada neste ponto e que não tem nada em haver com o processo:**
- Cabe aqui uma observação quanto ao novo atributo PRECO\_NEGOCIADO\_PRODUTO. Este atributo surgiu da condição de negócio em que existe um preço de referência para cada produto (PRECO\_PRODUTO) e o preço efetivamente negociado. Caso fosse desconsiderado esta hipótese poderíamos ter perdas de informação histórica com o tempo, uma vez que se alterar o preço de um produto, ele “automaticamente” alteraria o valor do pedido em outras épocas.



# Normalização (3FN)

- Uma entidade está na 3FN se ela está na 2FN e não possui dependências transitivas.
- Uma entidade que está na 2FN pode ter um atributo que não é uma chave mas que por si identifica outros atributos. Refere-se a isto como uma dependência transitiva
- Analisando o exemplo do pedido de venda, constata-se que a entidade PEDIDO possui dependência transitiva.

# Normalização (3FN)

- Portanto a entidade PEDIDO que estava assim:
- Entidade **PEDIDO**: NUMERO\_PEDIDO --> DATA\_PEDIDO, CODIGO\_CLIENTE, NOME\_CLIENTE, ENDERECO\_CLIENTE, TOTAL\_PEDIDO.
- fica normalizada na 3FN assim:
- Entidade **PEDIDO**: NUMERO\_PEDIDO --> DATA\_PEDIDO, CODIGO\_CLIENTE, TOTAL\_PEDIDO.
- Entidade **CLIENTE**: CODIGO\_CLIENTE --> NOME\_CLIENTE, ENDERECO\_CLIENTE
- *Note que o atributo transitivo era o codigo\_cliente que não fazia parte da chave primária da entidade PEDIDO na 2FN, e este mesmo atributo determina a existência de nome e endereço cliente, que não são diretamente dependentes do número do pedido.*

# Normalização (3FN)

- **Todo o exemplo estará na 3FN da seguinte forma:**
- Entidade **PEDIDO**: NUMERO\_PEDIDO --> DATA\_PEDIDO, CODIGO\_CLIENTE, TOTAL\_PEDIDO.
- Entidade **CLIENTE**: CODIGO\_CLIENTE --> NOME\_CLIENTE, ENDERECO\_CLIENTE
- Entidade **PRODUTO\_PEDIDO**: NUMERO\_PEDIDO, COD\_PRODUTO --> QTDE\_PEDIDA, PRECO\_NEGOCIADO\_PRODUTO
- Entidade **PRODUTO**: COD\_PRODUTO --> NOME\_PRODUTO, PRECO\_PRODUTO
- **Neste modelo já normalizado podemos:**
  - **Inserção** Inserir a qualquer momento um novo cliente ou produto.
  - **Eliminação** Se eliminar uma linha de um pedido qualquer não estaremos perdendo informações
  - **Atualização** Se muda qualquer nome de cliente ou descrição de produto, é suficiente acessar o objeto desejado e modificá-lo.