

Banco de Dados

**Normalização: 1FN, 2FN e 3FN**

Prof. Eldane Vieira

# Introdução

- A normalização é um processo matemático formal, fundamentado na teoria dos conjuntos, uma vez que o modelo relacional também segue a mesma estrutura.
- A normalização indicará o nível de qualidade de uma tabela.

# Introdução

- A normalização é usada para decompor ou dividir tabelas que não estejam bem projetadas em tabelas menores e eficientes, e assim ter projetos com alta qualidade.
- No processo de normalização, as regras são chamadas de formas normais e têm como meta impedir redundâncias na organização das tabelas e evitar resultados indevidos na atualização de uma tabela.

# Objetivo da normalização

- Analisar os esquemas de tabela para que alcance determinadas propriedades desejáveis, por exemplo:
  - Minimização de redundâncias;
  - Redução de irregularidade de inserção, exclusão e atualização com a tabela populada.

# Objetivo da normalização

- Quando um banco de dados é definido com poucas tabelas ou mesmo uma única tabela, isso pode resultar em dados redundantes, pesquisas demoradas, atualizações longas e exclusões que eliminam dados úteis.

# Objetivo da normalização

- Como exemplo, na imagem a seguir, os dados de produtos, vendedores, clientes e pedidos estão todos armazenados em uma única tabela chamada Vendas.

Nome Produto	Pedido Número	Nome Cliente	Endereço Cliente	Limite de Crédito	Data	Nome Vendedor
Limpadora a Vácuo	1458	Davi Bachmamm	Rio de Janeiro	US\$ 5,000	05/05/00	Carlos Book
Computador	2730	Helena Daudt	Vancouver	US\$ 2,000	05/06/00	João Hans
Refrigerador	2461	José Stolaruck	Chicago	US\$ 2,500	07/03/00	Silvio Pherguns
Televisão	456	Pedro Albuquerque	São Paulo	US\$ 4,500	09/05/00	Frederico Raposo
Rádio	1986	Carlos Antonelli	Porto Alegre	US\$ 2,000	18/09/00	Rui Ments
CD Player	1815	Davi Bachmamm	Rio de Janeiro	US\$ 5,000	18/04/00	Silvio Pherguns
Limpadora a vácuo	1963	C.V. Ravishandar	Bombaim	US\$ 7,000	03/01/00	Carlos Book
Limpadora a vácuo	1855	Carlos Antonelli	Porto Alegre	US\$ 2,000	13/05/00	João Hans
Refrigerador	1943	Davi Bachmamm	Rio de Janeiro	US\$ 5,000	19/06/00	Silvio Pherguns
CD Player	2315	Davi Bachmamm	Rio de Janeiro	US\$ 5,000	15/07/00	João Hans

*Uma consulta como: Qual cliente comprou computador? Envolve dados que não precisariam ser consultados.*

*A atualização do endereço de Davi Bachmann, por exemplo, envolveria várias linhas.*

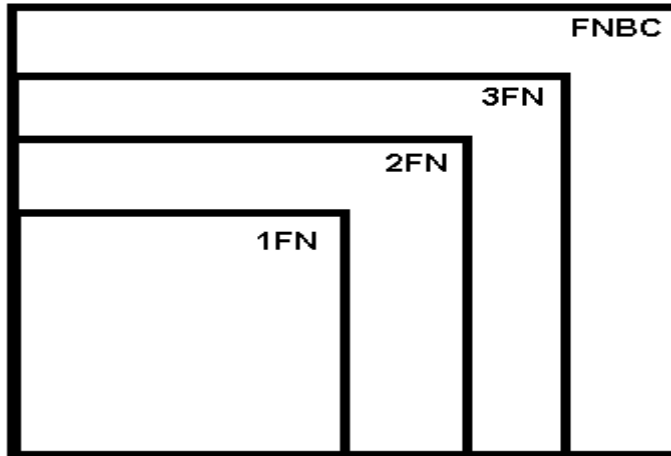
*A exclusão da compra do Helena Daudt, também implicará na deleção dos dados do cliente, ou seja, perderá o cadastro.*

# Níveis de normalização

- Para entender a normalização, é necessário saber que são impostas restrições à tabela em análise.
- Para que a tabela passe para um nível superior, ela precisa atender aos níveis inferiores primeiro.
  - Uma forma normal inferior é pré-requisito para o nível posterior.
  - Quando a tabela atende à última forma normal, está implícito que ela atenderá também às formas normais anteriores.

# Níveis de normalização

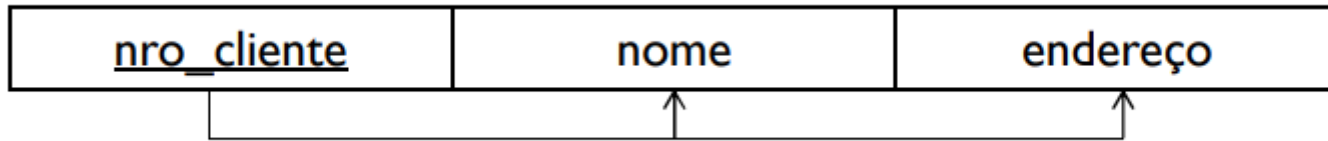
- Os principais níveis de normalização são os seguintes:
  - Primeira forma normal (1FN)
  - Segunda forma normal (2FN)
  - Terceira forma normal (3FN)
  - Forma normal Boyce Codd (FNBC)





# Dependências funcionais

- Dependências funcionais são restrições entre atributos, usadas para garantir a consistência da base de dados e para a avaliação da qualidade dos esquemas da relação.



- Pelo apresentado na imagem a chave nro\_cliente é uma dependência funcional para obter o nome e o endereço. Podendo ser representado da seguinte forma:
  - $\text{nro\_cliente} \rightarrow \text{nome}$
  - $\text{nro\_cliente} \rightarrow \text{endereço}$

# Primeira forma normal - 1FN

- É uma forma, mais simples e abrangente exigindo apenas que todos os atributos tenham domínios atômicos, isto é, valores indivisíveis.
- No exemplo a seguir tem-se no esquema da tabela Alunos o atributo multivalorado endereços, ou seja, não possui valores atômicos.
  - Alunos (matrícula, nome, {endereços})

# Primeira forma normal - 1FN

- Para normalizar a tabela é preciso colocar valores indivisíveis.
- A seguir são apresentadas duas soluções:
  - 1ª solução(chave primária composta, matrícula e endereço):
    - Alunos (matrícula, nome, endereço)
  - 2ª solução(Duas tabelas, uma com chave primária matrícula e a outra com chave primária composta, matrícula e endereço):
    - Alunos (matrícula, nome)
    - Alunos\_Endereços(matriculaFK, endereço)
      - matrículaFK é chave estrangeira que referencia o campo matrícula em Alunos.

# Segunda forma normal - 2FN

- A tabela está na 2FN, se ela estiver na 1FN, e se cada campo que não componha a chave primária depender inteiramente da chave primária, ou seja, de todos os atributos que a compõem.

# Segunda forma normal - 2FN

- A dependência funcional total ocorre quando temos uma chave primária composta e o atributo em questão depende de toda a chave não apenas de uma parte dela.
- Dizemos que um atributo depende funcionalmente de outro se para cada valor do atributo  $A$  sempre houver um valor único no atributo  $B$  ( $A \rightarrow B$ ).
  - $A$  determina funcionalmente  $B$  ou  $B$  é funcionalmente dependente de  $A$ .
  - Para todo valor de  $A$  existe um único valor de  $B$ .

# Segunda forma normal - 2FN

- Na 2FN, os atributos que não pertencem a chave primária devem depender inteiramente da chave, ou seja, ter uma dependência funcional total da chave primária.

## Segunda forma normal - 2FN

- Considere o seguinte esquema da tabela AlunosCursos para analisar se a mesma se encontra na 2FN.
  - AlunosCursos (matrícula, idCurso, PontuacaoObtida, localCurso)

# Segunda forma normal - 2FN

- Analisando o atributo PontuacaoObtida observa-se que ele depende diretamente da matrícula e do idCurso.
  - (matrícula, idCurso) → PontuacaoObtida



## Segunda forma normal - 2FN

- Outro atributo da tabela AlunosCursos a ser analisado é o localCurso, que está diretamente ligado ao idCurso.
- Seria o atributo localCurso dependente de matrícula?
  - Observa-se que não há nenhuma ligação direta entre localCurso e matrícula.
  - Dessa forma, a relação não atende à 2FN.

# Segunda forma normal - 2FN

- Para deixar a tabela na 2FN, é preciso remover da tabela a coluna que não depende totalmente da chave primária.
  - AlunosCursos (matrícula, idCursoFK, PontuacaoObtida)
    - idCursoFK é chave estrangeira que referencia o campo idCurso da tabela LocalCursos.
  - LocalCursos (idCurso, localCurso),

# Terceira forma normal - 3FN

- O primeiro passo é atender à 1FN e 2FN.
- A 3FN exige que cada atributo não pertencente à chave primária dependa somente da chave primária, ou seja, que os atributos não chave sejam independentes entre eles. Não pode ocorrer dependência transitiva.
  - Exemplo de dependência transitiva:
    - Sendo X a chave primária, se  $(X \rightarrow Z)$  e  $(Z \rightarrow Y)$  então  $(X \rightarrow Y)$ , mas sendo Z e Y atributos não pertencentes à chave, então a dependência  $(Z \rightarrow Y)$  não deve ocorrer.

# Terceira forma normal - 3FN

- Considere o seguinte esquema da tabela Empregado.
  - Empregado (id, nome, idCargo, descricaoCargo)
    - Chave primária é o campo id.
    - Os campos nome e idCargo dependem de id.
    - O campo descricaoCargo depende de idCargo (que não compõe a chave primária), por isso esta tabela não está na 3FN.

# Terceira forma normal - 3FN

- O atributo descricaoCargo depende exclusivamente de idCargo e, portanto, deve-se criar uma nova tabela com esses atributos para atender as restrições da 3FN.
  - Empregado (id, nome, idCargoFK)
    - idCargoFK é uma chave estrangeira que referencia o campo idCargo da tabela Cargo.
  - Cargo (idCargo, descricaoCargo)

# Exercícios

- 1) Normalize a tabela Funcionário(CPE, nome,{telefone}) de modo a atender os requisitos da 1FN.
- 2) Normalize a tabela Pedido(nroPedido, nroPeça, descriçãoPeça, qntComprada, dataPedido, valorTotal) de modo a atender os requisitos da 2FN.
- 3) Normalize a tabela Paciente(idPaciente, nomePaciente, endereçoPaciente, CRM, nomeMédico) de modo a atender os requisitos da 3FN.

# Referência

- R. Ramakrishnan e J. Gehrke, Database Management Systems, 3a Edição, McGraw-Hill, 2003.