**Formas Normais**

A **normalização** é um processo de análise de esquemas das relações de dados.

A finalidade de uma normalização de dados é a minimização das redundâncias e a minimização das anomalias. O processo de normalização serve para evitar anomalias de atualização de dados, não análise de desempenho ou otimização de consultas. Assim, quanto mais normalizado for o banco de dados, mais junções entre tabelas existirão e como consequência, menos desempenho ele terá.

Um BD desnormalizado é moroso (lento) a sua consulta. **Um BD normalizado dentro do necessário para a organização é o ideal.** Um BD excessivamente normalizado também é moroso a sua consulta.

* **Primeira Forma Normal (1FN):** O objetivo é **retirar os atributos ou grupos repetitivos**. Uma tabela está na 1FN se, e somente se, todos os valores dos atributos forem atômicos (indivisíveis), ISTO É, **NÃO DEVEM EXISTIR ATRIBUTOS MULTIVALORADOS ou compostos. Cada campo de uma tabela precisa conter somente um único tipo de dado, e cada parcela de dado deve ser armazenada em somente um lugar**. Essa exigência é conhecida como **atomicidade de dados.**Na 1FN, há a **eliminação das tabelas aninhadas,**ou seja, tabelas dentro de tabelas.
* **Segunda Forma Normal (2FN):** A 2FN **é baseada no conceito de dependência funcional total.** O objetivo é **separar as dependências parciais**. Uma tabela está na 2FN se, e somente se, estiver na 1FN e **cada atributo não-chave for dependente da chave primária (ou candidata) inteira**, isto é, **não devem existir dependências parciais**. Em outras palavras, **todas as colunas devem ter dependência funcional com a totalidade de cada chave candidata e, não apenas, com parte dela**.
* **Terceira Forma Normal (3FN):** O objetivo é **eliminar dependências transitivas.** Uma tabela está na 3FN se, e somente se, estiver na 2FN e **cada atributo não-chave NÃO POSSUIR DEPENDÊNCIA TRANSITIVA PARA CADA CHAVE CANDIDATA**.
* **Forma Normal de Boyce-Codd (FNBC):** Uma tabela está na FNBC se, e somente se, estiver na 3FN e, para cada dependência x -> y, X deverá ser uma superchave, isto é, **todo determinante é uma chave candidata**. **Toda tabela que esteja na FNBC está na 3FN, mas nem toda tabela na 3FN está na FNBC.**
* **Quarta Forma Normal (4FN):** **Se baseia no conceito de dependência multivalorada**. Uma relação está na 4FN se, e somente se, estiver na 3FN e **não contiver dependências multivaloradas.**
* **Quinta Forma Normal (5FN)**: **Se baseia no conceito de dependência de junção (DJ).** Uma tabela está na 5FN se, e somente se, estiver na 4FN e **não existirem dependências de junções**. Uma relação está na 5FN quando o conteúdo do registro não pode ser mais reconstruído (efetuar join) a partir de outros registros menores extraídos desse registro considerado.

**Outra definição:**

* **Primeira Forma Normal (1FN):** O objetivo é **retirar os atributos ou grupos repetitivos**. Uma tabela está na 1FN se, e somente se, todos os valores dos atributos forem atômicos (indivisíveis), ISTO É, **NÃO DEVEM EXISTIR ATRIBUTOS MULTIVALORADOS ou compostos. Cada campo de uma tabela precisa conter somente um único tipo de dado, e cada parcela de dado deve ser armazenada em somente um lugar**. Essa exigência é conhecida como **atomicidade de dados.**Na 1FN, há a **eliminação das tabelas aninhadas,**ou seja, tabelas dentro de tabelas.
* **Segunda Forma Normal - u**ma relação está na 2FN se, e somente se, estiver na 1FN e **cada atributo não-chave for dependente da chave primária inteira**, isto é, cada atributo não-chave não poderá ser dependente de apenas parte da chave. 2º Forma Normal (2FN): toda relação deve estar na 1FN e devem-se eliminar dependências funcionais parciais, ou seja, todo atributo não chave deve ser totalmente dependente da chave primária.
* **Terceira Forma Normal (3FN):** Uma relação está em 3 FN se e somente se estiver em 2FN e nenhum atributo não chave for dependente não transitivo da chave primária, ou seja, garante que não haja dependência transitiva ou indiretas. **Dependência transitiva -**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. 3FN não deve haver dependência transitiva. Exemplo clássico é campos calculados.
* **FNBC**: Uma tabela está na FNBC se, e somente se, estiver na 3FN e, para cada dependência x -> y, X deverá ser uma super. chave, isto é, **todo determinante é uma chave candidata**. **Toda tabela que esteja na FNBC está na 3FN, mas nem toda tabela na 3FN está na FNBC. É** considerada uma variação da 3FN e não é necessário que os dados estejam normalizados. Então, é possível ter uma tabela não normalizada e ir para FNBC.
* **Quarta Forma Normal (4FN):** **Se baseia no conceito de dependência multivalorada**. Uma relação está na 4FN se, e somente se, estiver na 3FN e **não contiver dependências multivaloradas.**
* **Quinta Forma Normal (4FN):** ***Durante o projeto de um banco de dados, transformando-se as relações na quinta forma normal, garante-se que elas não terão anomalias que possam ser eliminadas usando-se projeções.*** A 5FN além de eliminar a dependência funcional de junção, também assegura que **nenhuma coluna estranha ao contexto foi criada.**A criação de colunas adicionais (que estão fora do contexto da aplicação) podem ser eliminadas utilizando a projeção. **Por isso, a 5FN garante que não existirão anomalias que possam ser eliminadas usando-se projeções.**

**Resumo**

* 1FN - não possui atributos multivalorados. Todas as colunas de uma tabela têm apenas valores atômicos.
* 2FN - não possui dependências parciais.
* 3FN - não possui dependências transitivas.
* FNBC (Forma Normal Boyce-Codd) - variação da 3FN. Se todo determinante é chave candidata.
* 4FN - elimina as dependências multivaloradas (DMV) não triviais de uma tabela.
* 5FN - combate a dependência de junção.

**Dependência funcional**: um atributo **Y**é funcionalmente dependente de um atributo **X**se e somente se cada valor de **X**tiver associado a ele um valor de **Y.** Simbolicamente:

**X ->Y = X** determina funcionalmente **Y -**  **Y**depende funcionalmente de **X.**

Uma chave primária em uma relação determina funcionalmente todos os outros atributos não chave na linha.

* **Dependência funcional total:**atributos não chave dependem como um todo da PK composta, e não somente parte dela.
* **Dependência funcional parcial:**Os atributos não chave não dependem funcionalmente de toda a PK quando está for composta.
* **Dependência funcional transitiva:**Um atributo não depende diretamente da PK, nem mesmo parcialmente, mas depende de um outro atributo não chave.
* **Dependência multivalorada:**Para cada valor de um atributo A, existe um conjunto de valores para outros atributos B e C que estão associados a ele, mas são independentes entre si.

**Dependência Funcional -** 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 linhas da tabela, para cada valor de**C1 que aparece na tabela, aparece o mesmo valor de C2.**

**J. C. Date** - Introdução a sistemas de bancos de dados página 16 - 8 a. Edição (os grifos são dele):

"Nota: Não queremos sugerir que *toda* redundância possa ou deva necessariamente ser eliminada. Às vezes, há motivos comerciais ou técnicos plausíveis para manter várias cópias distintas dos mesmos dados. Porém, queremos sugerir que toda redundância deve ser cuidadosamente **controlada**; isto é, o SGBD deve estar ciente dela (caso exista) e deve assumir a responsabilidade pela "propagação de atualizações"."