**Normalização**

Serie para analisar tabelas e orientá-las de modo que sua estrutura seja simples, relacional e estável, para que o gerenciamento possa ser eficiente e seguro.

Os objetivos são evitar a perda e a repetição de informações e atingir uma forma de representação adequada para o que se deseja armazenar.

Técnica empregada para melhorar as estruturas de dados resultante de um projeto de dados do sistema.

Evita problemas de redundância e anomalia através da decomposição de uma **entidade** em uma ou mais entidades. As formas normais (**FN**), são as regras normalização e podem ocasionar uma proliferação de entidades que nem sempre é o ideal do ponto de vista de performance, devendo ser balanceadas as vantagens e desvantagens antes da efetivação dos resultados de uma **FN** (NF em inglês).

Assim, a normalização é um processo sistemático de geração de tabelas podendo ser aplicadas as regras que vão da 1° à 5° **FN.** Em grande partes dos casos, a aplicação da 3° forma normal, já produz um conjunto de tabelas que organiza satisfatoriamente um schema.

Para adequar uma nova tabela a uma FN deve-se redesenhar o seu formato. A normalização também pode ser utilizada, para projetar tabelas a partir de documentos, considerando-os como uma única tabela e aplicando as regras.

**Exemplo:** Tabela Desnormalizada

|  |  |
| --- | --- |
| |  | | --- | | **Nota fiscal** |  * N° Nota; * Nome do cliente; * Endereço; * Vendedor; * Data de Emissão; * Produto 1, 2, 3 … N. |

**1° FN**

Exige que uma tabela não contenha alinhamentos (atributos repetitivos). Realiza o que se chama de alinhamento. Gerando uma tabela com todos os dados coletados em atributos atômicos. A **PK** dessa tabela será formada pela composição da **PK** de cada tabela alinhada, além da **PK** da tabela externa. Caso existam atributos multivalorados, estes devem se tornar componentes da chave PK, uma vez que os seus valores estarão associados várias vezes com os demais dados da tabela. Se existia alinhamento, provavelmente a 1NF não é o estado final, uma vez que haverá redundância nos dados da tabela externa. Uma tabela só está na 1FN se nenhum dos seus atributos possui domínio mutivalorável, ou seja, devem ser eliminados registros de dados repetitivos, na Nota Fiscal vemos que os produtos são atributos repetidos.

|  |  |
| --- | --- |
| |  | | --- | | **Nota fiscal** |  * N° Nota; * Nome do cliente; * Endereço; * Vendedor; * Data de Emissão; * Valor total da nota. |

**1**

**|**

**/ | \**

**N**

|  |  |
| --- | --- |
| |  | | --- | | **Item Nota fiscal** |  * N° Nota; * CD\_Produto; * NM\_Produto. |

**2° FN**

Exige o entendimento do conceito de dependência funcional (DF). Existe DF em uma tabela sempre que um conjunto de um ou mais atributos determina o valor de outro conjunto de um ou mais atributos. A regra diz que uma tabela está na 2FN se e somente se, estiver na 1FN e todo atributo não chave depende funcionalmente de toda a PK e não apenas de parte dela (deve existir DF de toda a PK).

Nesse caso cada atributo não chave de uma tabela deve ser analisado. Caso seja verificado DF de parte da chave PK, gera-se uma tabela que tenha esse atributo não chave, e essa parte da PK (Que se torna agora PK completa). Se existirem outros atributos não chave, com DF dessa mesma parte da PK, esses serão inseridos nesta nova tabela. O resultado é um conjunto de tabelas onde só existem DFs da PK completa. Obviamente, tabelas com PK simples ou que não possui atributos não chave já estão na 2FN. Mesmo após a aplicação da 2FN podem ainda existir problemas de redundâncias e anomalias.

**Aplicação:**

A projeção do atributo determinante de produtos, mais os outros atributos que dependem apenas dele em outra tabela. A PK será o atributo determinante projetado.

**1 N**

**N**

**1**

|  |  |
| --- | --- |
| |  | | --- | | **Nota fiscal** |  * # N° Nota; * ID\_Cliente; * Endereço; * ID\_Vendedor; * Data\_Emissão; * Valor\_Total; |

|  |  |
| --- | --- |
| |  | | --- | | **Nota fiscal** |  * N° Nota; * CD\_Produto; * QT\_Produto; * VL\_Unitário; * VL\_Total; |

**N**

**1**

|  |  |
| --- | --- |
| |  | | --- | | **Produto** |  * CD\_Produto; * Descrição; * UM\_Medida; * VL\_Unitário; |

|  |  |
| --- | --- |
| |  | | --- | | **Vendedor** |  * ID\_Vendedor; * Nome; |

**Aula – 20/05/2025**

**Terceira Forma Normal**

A regra é clara, uma tabela está na **3FN**, se e somente se todo atributo não chave depende funcional e diretamente da **PK**, e não de um conjunto de um ou mais atributos não chaves. Obviamente, apenas tabelas com mais de um atributo não chave são consideras, para efeito de análise na **3FN**. Caso seja percebida uma **DF** indireta, uma nova tabela é gerada, onde os atributos não chave que determinam outros, tonam-se chave primária e os atributos com **DF** indireta migram para ela como atributos não chave. Na tabela antiga os atributos determinantes tornam-se chaves estrangeiras.

A tabela estará na **3FN** quando estiver na **2FN** que não há **DF** transitiva entre seus atributos.

**DFT -** Significa que um atributo não depende diretamente de um atributo determinante e sim de algum atributo, que por sua vez depende de determinante.

**Quarta Forma Normal**

Ocorre quando há tabelas que contem relacionamentos mantém relacionamentos ternários ou superiores e são detectadas dependências funcionais multivaloradas, ou seja, um ou mais atributos determinam vários valores de um outro atributo. São consideradas, para efeito de análise de **4FN**, ajudam tabelas com **PK** tripla (quádrupla e assim por diante) que tenham atributos não chave.

Normalmente detecta-se a necessidade da aplicação da **4FN** quando se tem mais de um atributo multivalorado em uma tabela não normalizada. Esses atributos acabam fazendo parte da **PK** nessa tabela na **1FN,** porém, isolados, eles não determinam nenhum atributo não chave. Isto resulta em uma tabela com **PK** tripla (no mínimo), onde um deles é o **ID** da tabela propriamente dito.

Uma tabela está na **4FN** se está na **3FN** e não apresenta mais um fato multiplicador em relação a uma entidade descrita na tabela.

O objetivo da **4FN** é evitar redundância em situações em que há fatos multivalorados em vez de **DF**.

Para isso, constroem-se tantas tabelas quanto fatos multivalorados há para certa entidade descrita na tabela. Esta entidade estará representada em todas as tabelas resultantes.

Esta regra também se aplica apenas a tabelas que representam relacionamentos ternários, sem atributos não chave. É dito que uma tabela está **5FN** se um relacionamento triplo puder ser decomposto em três tabelas de relacionamentos binários (**PK** é um par) sem que isso gere dados incorretos, quando eles forem combinados novamente em uma tabela de relacionamento triplo.

Essa garantia de consistência exige que uma premissa seja sempre verdadeira:

Toda vez que um dado **A** se relaciona com outros dois (**B** e **C**), e estes outros também, por sua vez (gerando 3 relações binárias (A e B) , (A e C) e (B e C);), então é verdade que ocorre na realidade os 3 dados combinados (A, B e C). Um exemplo é pensar em A como sendo um agente de vendas, B como sendo a empresa fabricante e C um produto.

A premissa seria: Se A representa a empresa B, A é especialista na venda do produto C e B produz o produto C, então é verdade que A vende o produto C para a empresa B. Essa **FN** elimina o relacionamento ternário e a redundância de pares de valores relacionados, como por exemplo, a redundância da afirmação de que o agente A representa a empresa B.

Uma tabela que está na **5FN** se está na **4FN** e seu conteúdo não pode ser reconstituído a partir de tabelas menores. O seu objetivo é impedir que a decomposição de tabelas gera informação inconsistente na junção.

Verifique sempre se há ou não a necessidade de conceber relacionamentos de grau maior que 2. Em todo caso, dificilmente um projetista aplicará a normalização segundo a **5FN,** porque qualquer tabela que não seja **5FN** geralmente falha em uma das **FN** anteriores.

**Atividade**

Um cliente entrou em contato, ele tem um formulário que precisa dos dados cadastrais do usuário, fazendo tudo manualmente. Agora ele quer um BD para armazenar e facilitar os dados.

**Dados cadastrais:**

Matrícula: Nome:

Nascimento: Nacionalidade: Sexo:

Estado Civil: RG: CPF:

Endereço: Data de Admissão:

Cargos ocupados

Cargo: Data de início: Data Fim:

**Repetições**

Departamento de Lotação

Departamento: Data Início: Data Fim:

**Repetições**

Dependentes

Nome: Data de Nascimento:

**Repetições**

Fazer o BD dessa estrutura de dados, com inserções de dados e consultas.