

8. Normalização

- Definições -

➤ Superchave:

- Uma superchave de um esquema de relação $R = \{A_1, A_2, \dots, A_n\}$ é um conjunto de atributos $S \subseteq R$ que contenha a propriedade na qual não haverá duas tuplas t_1 e t_2 , em qualquer estado da relação r de R , em que $t_1[S] = t_2[S]$.
- Uma chave K é uma superchave com a propriedade adicional de que a remoção de qualquer atributo J fará com que K não seja mais uma superchave.

➤ Atributo primário:

- Um atributo de um esquema de relação R é chamado atributo primário de R se for membro de alguma chave candidata de R . O atributo é dito não primário se não for um atributo primário – isto é, se não for membro de alguma chave candidata.

8. Normalização

- Dependência Funcional -

- É uma restrição de Integridade

Dada uma relação R com atributos

$A_1, \dots, A_n, B_1, \dots, B_m,$

Dizemos que

A_1, \dots, A_n determina funcionalmente B_1, \dots, B_m

$(A_1, \dots, A_n) \longrightarrow B_1, \dots, B_m$

Se sempre que duas tuplas tiverem os mesmos valores para A_1, \dots, A_n , então elas terão o mesmo valor para B_1, \dots, B_m .

8. Normalização

- Dependências Funcionais (DFs) -

MATRÍCULAS (curso, estudante, semestre, nota)

{Curso, estudante} $\not\rightarrow$ semestre, nota

{Curso, estudante, semestre} \rightarrow nota

| Estudante | Curso | Semestre | Nota |
|------------|------------------|----------|------|
| João Silva | Banco de Dados I | 2022.2 | 8,5 |
| João Silva | Banco de Dados I | 2023.1 | 7,5 |

- DFs (ou FDs, do inglês) ocorrem muito frequentemente:
- Os SGBDs permitem a manutenção de DFs.
- Durante o projeto do esquema, deve-se identificar todas as DFs, pois isto ajuda a identificar esquemas ruins.

8. Normalização

- Dependências Funcionais (DFs) -

Tipos de dependências funcionais em SGBDs:

- 1) Dependência funcional trivial
- 2) Dependência funcional não trivial
- 3) Dependência funcional multivalorada
- 4) Dependência funcional transitiva

8. Normalização

- Dependências Funcionais (DFs) -

Dependência Funcional Trivial:

- Na Dependência Funcional Trivial, um dependente é sempre um subconjunto do determinante.
- Ou seja, se $X \rightarrow Y$ e Y é o subconjunto de X , então esta DF é chamada de Dependência Funcional Trivial.
- P. ex.: $\{\text{roll_no}, \text{nome}\} \rightarrow \text{nome}$ é uma dependência funcional trivial, uma vez que o nome é um subconjunto do conjunto determinante $\{\text{roll_no}, \text{nome}\}$
- Da mesma forma, $\text{roll_no} \rightarrow \text{roll_no}$ é também um exemplo de dependência funcional trivial.

8. Normalização

- Dependências Funcionais (DFs) -

Dependência Funcional Não Trivial:

- Na dependência funcional não trivial, o dependente não é estritamente um subconjunto do determinante.
- Ou seja, se $X \rightarrow Y$ e Y não for um subconjunto de X , será chamado de dependência funcional não trivial.
 - $\text{roll_no} \rightarrow \text{nome}$ é uma dependência funcional não trivial, uma vez que o nome dependente não é um subconjunto do determinante roll_no
 - Da mesma forma, $\{\text{roll_no}, \text{nome}\} \rightarrow \text{idade}$ também é uma dependência funcional não trivial, uma vez que idade não é um subconjunto de $\{\text{roll_no}, \text{nome}\}$

8. Normalização

- Dependências Funcionais (DFs) -

Dependência Funcional *Multivalorada*:

- Na **dependência funcional de vários valores**, as entidades do conjunto dependente **não são dependentes umas das outras**.
- Ou seja, se $a \rightarrow \{b, c\}$ e não existe **dependência funcional** entre **b** e **c**, então esta DF é chamada de **dependência funcional multivalorada**.
- Por exemplo: $roll_no \rightarrow \{nome, idade\}$ é uma dependência funcional com vários valores (Multivalorada), uma vez que o nome e a idade do dependente não são dependentes um do outro (ou seja, **nome \rightarrow idade** ou **idade \rightarrow nome não existe!**)

8. Normalização

- Dependências Funcionais (DFs) -

Dependência Funcional *Transitiva*:

- Na **Dependência Funcional Transitiva**, dependente é indiretamente dependente do determinante.
- Ou seja, Se $a \rightarrow b$ & $b \rightarrow c$, então, de acordo com o axioma da transitividade, $a \rightarrow c$. Esta é uma **Dependência Funcional Transitiva**.
- Por exemplo: $\text{enrol_no} \rightarrow \text{dept}$ e $\text{dept} \rightarrow \text{building_no}$. Portanto, de acordo com o axioma da transitividade, $\text{enrol_no} \rightarrow \text{building_no}$ é uma dependência funcional válida. Esta é uma dependência funcional indireta, portanto chamada de dependência funcional transitiva.

8. Normalização

- Chave de Relação -

A chave de uma relação determina funcionalmente todos os atributos da relação

Suponha $R (A_1, A_2, \dots, A_n)$ e $X \subset \{A_1, A_2, \dots, A_n\}$.

- *X é uma superchave de R sss $X \longrightarrow \{A_1, A_2, \dots, A_n\}$*
- *X é uma chave de R se e somente se (sss):*
 - 1. $X \longrightarrow \{A_1, A_2, \dots, A_n\}$
 - 2. Não existe nenhum $Y \subset X$ tal que $Y \longrightarrow \{A_1, A_2, \dots, A_n\}$

8. Normalização

- Conceitos -

- A *normalização de dados* pode ser vista como o processo de análise de determinados esquemas de relações, com base em suas dependências funcionais e chaves primárias, para alcançar as propriedades desejáveis:
 - Minimização de redundâncias
 - Minimização de modificações
- Os esquemas de relações insatisfatórias, que não alcançam certas condições – os testes de forma normal, são decompostas em esquemas de relações menores que passam nos testes e, conseqüentemente, possuem as propriedades desejadas.

8. Normalização

- Conceitos -

- Entendendo as dependências funcionais, somos capazes de especificar aspectos semânticos dos esquemas de relação.
- Para tal, presumimos que:
 - Um conjunto de dependências funcionais seja específico para cada relação, e
 - Que cada relação tenha uma chave primária designada.

Essas informações, combinadas a testes para formas normais, direcionam o processo de normalização dos projetos de esquemas relacionais.

8. Normalização

- Conceitos -

- A *forma normal de uma relação* refere-se à condição da mais alta **forma normal** alcançada e, consequentemente, indica o grau no qual foi normalizada.
- A normalização sempre deve estar acompanhada da garantia de duas propriedades:
 - Junção sem perdas: que garante que o problema de geração de tuplas ilegítimas não ocorra; (propriedade crítica que deve sempre ser garantida)
 - Preservação de dependências: garante que cada dependência funcional será representada em alguma relação individual resultante da decomposição.

8. Normalização

- 3ª. Forma Normal (3FN) -

- A 3ª. forma normal prega que os atributos que não são chave devem ser independentes entre si e dependentes única e exclusivamente da chave primária da tabela.
- Não é permitido dependência transitiva.
- Adicionalmente, se a tabela está na 3ª. forma normal, automaticamente estará na 1ª. e na 2ª. formas normais também.

8. Normalização

- Formas Normais -

❑ Terceira Forma Normal (3FN)

↪ Dado um conjunto de dependências funcionais F , uma relação R está em 3FN, se todas dependências de F^+ da forma $\alpha \rightarrow \beta$, com $\beta \subseteq R$ e $\alpha \subseteq R$, pelo menos umas das seguintes condições é garantida

⇒ $\alpha \rightarrow \beta$ é uma dependência funcional trivial.

⇒ α é super chave de R

⇒ Não há atributos A fora da super chave com dependência funcional transitiva com esta

↪ Exemplo

⇒ Conta(num_conta, saldo, cod_cliente, cod_ag)

⇒ $\text{num_conta} \rightarrow \text{saldo}$

➤ (num_conta não é super chave, portanto \sim BCFN e 3FN)

Normalizando projeto para 3FN:

FUNC_DEP

| Fnome | <u>Cpf</u> | Datanasc | Endereco | Dnumero | Dnome | Cpf_gerente |
|-------|------------|----------|----------|---------|-------|-------------|
|-------|------------|----------|----------|---------|-------|-------------|



Normalizacao 3FN

DF1

| Fnome | <u>Cpf</u> | Datanasc | Endereco | Dnumero |
|-------|------------|----------|----------|---------|
|-------|------------|----------|----------|---------|



DF2

| <u>Dnumero</u> | Dnome | Cpf_gerente |
|----------------|-------|-------------|
|----------------|-------|-------------|



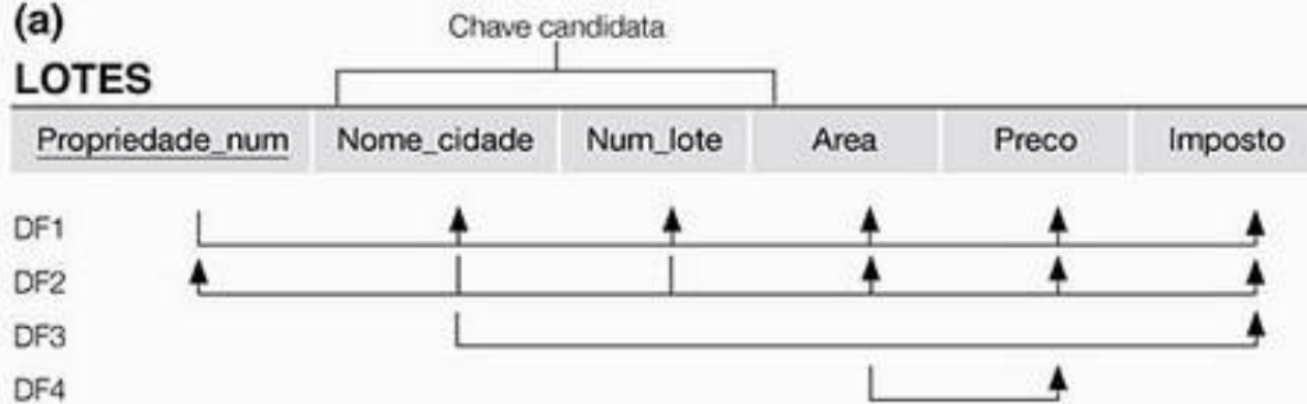
8. Normalização

- Exemplo de Normalização (2FN e 3FN) -

- Considere o esquema de relação LOTES mostrado na Figura a seguir, que descreve lotes de terreno à venda em diversas cidades de um estado.
- Suponha que existam duas chaves candidatas: `Propriedade_num` e `{Nome_cidade, Num_lote}`; ou seja, números de lote são únicos apenas dentro de cada cidade, mas números de propriedade (`Propriedade_num`) são únicos entre as cidades do estado inteiro.
- Suponha que as duas outras dependências funcionais se mantenham em LOTES:
 - DF3: `Nome_cidade` → Imposto
 - DF4: `Area` → Preço

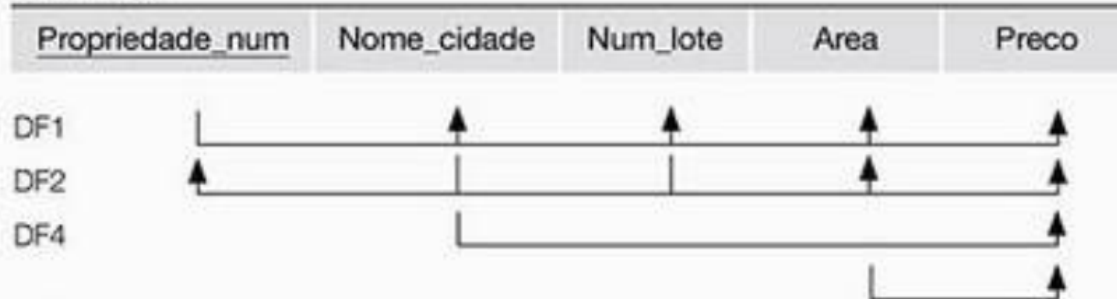
(a)

LOTES



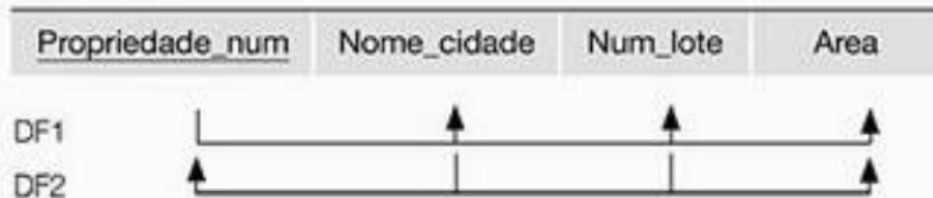
(b)

LOTES1

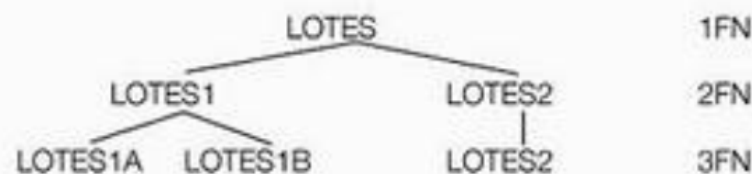


(c)

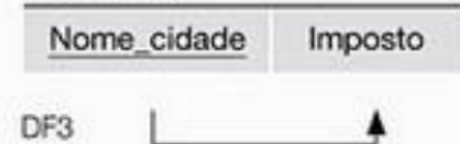
LOTES1A



(d)



LOTES2



LOTES1B

